



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

INDICADORES DE MEDIÇÃO DE TESTES EM PROJETOS ÁGEIS

Douglas da Silva Lima

Orientador: Gleison dos Santos Souza
Coorientador: Cristina Teles Cerdeiral

RIO DE JANEIRO, RJ - BRASIL

Setembro de 2017

INDICADORES DE MEDIÇÃO DE TESTES EM PROJETOS ÁGEIS

Douglas da Silva Lima

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

Prof. Gleison dos Santos Souza, D.Sc. - UNIRIO

Cristina Teles Cerdeiral, D.Sc. - UNIRIO

Prof. Toacy Cavalcante de Oliveira, D.Sc. - COPPE/UFRJ

Prof. Leonardo Guerreiro Azevedo, D.Sc. - UNIRIO

RIO DE JANEIRO, RJ - BRASIL

Setembro de 2017

L732 Lima, Douglas da Silva
 Indicadores de Medição de Testes em Projetos
 Ágeis / Douglas da Silva Lima. -- Rio de Janeiro, 2017.
 199 f.

 Orientador: Gleison dos Santos Souza.
 Coorientadora: Cristina Teles Cerdeiral.
 Dissertação (Mestrado) - Universidade Federal do Estado do Rio de
 Janeiro, Programa de Pós-Graduação em Informática, 2017.

 1. Indicadores. 2. Medidas. 3. Testes. 4. Medição de software. 5.
 Métodos ágeis. I. Souza, Gleison dos Santos, orient. II. Cerdeiral,
 Cristina Teles, coorient. III. Título.

AGRADECIMENTOS

Agradeço a Deus que me deu força para superar todas as dificuldades e concluir o curso de mestrado, por ter me concedido saúde, persistência e inteligência emocional para superar diversos momentos difíceis durante esta jornada.

Agradeço a minha esposa e filho que se privaram de vários momentos de lazer nesse período em prol do meu objetivo, e conviveram com meus momentos de angústia e incertezas nesse período.

Agradeço a minha amiga de longa data Larissa Araujo pela apresentação ao PPGI, foi o início de tudo há 3 anos atrás.

Agradeço a todos os colegas da Unirio de todas as disciplinas cursadas, com todos algum aprendizado eu tirei nesse período.

A todos os professores com quem tive contato cursando as disciplinas, meu muito obrigado.

A minha coorientadora, Cristina Cerdeiral, pelas valiosas contribuições dadas em nossas reuniões e revisões.

Aos meus colegas da UNIRIO, Yure Ouriques e Rafaela Sampaio pelo companheirismo e apoio durante esse período.

Aos professores Leonardo Guerreiro e Toacy de Oliveira por aceitarem participar da banca avaliadora

A todo corpo docente e funcionários da secretaria do PPGI.

Ao meu orientador, Gleison Santos, pessoa admirável, por toda paciência, atenção e apoio concedido, sem sua grandiosa colaboração eu não teria terminado esta jornada. Minha eterna gratidão e admiração!

LIMA, Douglas da Silva. **Indicadores de Medição de Testes em Projetos Ágeis**. UNIRIO, 2017. 199 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

As organizações de software buscam desenvolver software cada vez mais rápido, mantendo ou aumentando a qualidade das entregas. A fase de teste de software apoia na verificação e validação dos requisitos confirmando a qualidade dos produtos de software e apoiando a aceitação pelo cliente. A execução de casos de teste e consequente descoberta de defeitos ajuda a reduzir o impacto de problemas após a entrega do software. Metodologias ágeis focam em desenvolvimento direcionado a entregas em iterações curtas que permitem maior agilidade na liberação do software. Durante a execução da fase de testes é importante manter um processo de medição que apoie a coleta e análise de medidas e indicadores que ajudem na detecção de problemas e na melhoria do gerenciamento das atividades executadas. O objetivo deste trabalho é identificar medidas e indicadores utilizados em projetos de desenvolvimento de software aplicados às atividades de testes em projetos executados com metodologia ágil. A partir dos resultados de um mapeamento sistemático da literatura sobre medidas de teste foram propostos quinze indicadores, que foram descritos através de fichas de registro de indicadores utilizando o template chamado ASM.br. Um estudo de caso foi executado em uma organização que possui projetos de teste de software com o objetivo de avaliar as fichas de registro dos indicadores para verificar se o conteúdo e procedimentos podem apoiar ao gerenciamento dos testes. Como resultado, pudemos constatar evidências de facilidade de entendimento das fichas e procedimentos descritos, e também percepção de que as fichas podem ser úteis no apoio da gestão de projetos de teste.

Palavras-chave: métricas, indicadores, testes, medidas, medição de software, métodos ágeis.

ABSTRACT

Software organizations focus on delivering systems each time faster, keeping software quality and when possible increasing it. The testing phase supports verification and validation of requirements by assessing the quality of the software products and their acceptance by clients. Test case execution and defect identification during tests help to reduce the impact of problems after system deployment. During testing is important to use a measurement process to support to measures and indicators collection and analysis to help detect problems and to improve test management activities. The goal of this research is to identify and describe measures and indicators used during agile software development applied to testing activities. Using the results of a systematic mapping study in test measures, we propose fifteen indicators, that were detailed using a defined template named as ASM.br. We conducted a case study in an organization that has several testing projects with the objective to evaluate the information described through the ASM.br template and check if the contents and procedures can support test management. As a result, we found evidences of utility, easibility and feasibility that the proposed indicators can support test phase management.

Keywords: metrics, indicators, test, measure, software measurement, agile methodology.

ÍNDICE

1	Introdução	1
1.1.	Contexto e Motivação	1
1.2.	Objetivo e Questão de Pesquisa	4
1.3.	Abordagem Metodológica	5
1.4.	Organização da Dissertação	6
2	Medição de Software e Testes em Projetos Ágeis	8
2.1.	Medição de Software.....	8
2.2.	Criação de Planos de Medição	11
2.2.1.	Método GQM (Goal Question Metric)	11
2.2.2.	Método PSM (<i>Practical Software Measurement</i>)	12
2.3.	Template ASM.br	14
2.3.1.	Componente Perfil Organizacional	14
2.3.2.	Componente Mapeamento de Indicadores	15
2.3.3.	Componente Fichas	17
2.4.	Metodologias Ágeis	25
2.5.	Testes de Software em Projetos Ágeis	28
2.6.	Trabalhos relacionados.....	33
2.7.	Considerações Finais.....	37
3.	Indicadores de Testes em Projetos Ágeis.....	38
3.1.	Introdução	38
3.2.	Planejamento do Estudo	39
3.3.	Indicadores de testes em projetos ágeis	52
3.4.	Definição dos Objetivos e Necessidades de Informação	56
3.5.	Seleção dos Níveis e Processos do MR-MPS-SW	58
3.6.	Seleção dos Indicadores	58
3.7.	Mapa de relacionamentos.....	60
3.8.	Planilha de apoio a coleta de dados e geração dos gráficos para os indicadores ..	66
3.9.	Considerações Finais.....	66
4.	Avaliação dos Indicadores de Testes para Projetos Ágeis	69
4.1.	Introdução	69
4.2.	Planejamento	70
4.2.1.	Definição dos objetivos	70
4.2.2.	Caracterização da organização e participantes alvo	71
4.2.3	Definição do método de avaliação	71
4.2.4	Definição do método de análise dos resultados	73
4.3.	Execução do estudo de caso	73
4.3.1	Caracterização da organização.....	74
4.3.2	Caracterização dos participantes	74
4.3.3	Capacitação para utilização das fichas de registro dos indicadores	78
4.3.4	Avaliação do conteúdo das fichas de registro dos indicadores	79
4.3.5	Coleta do tempo de análise das fichas.....	83
4.3.6	Coleta das medidas e geração dos gráficos dos indicadores	84

4.4	Análise dos resultados	88
4.5	Ameças à validade e Limitações	90
4.6	Considerações finais.....	91
5	Conclusão	92
5.1	Considerações Finais.....	92
5.2	Contribuições	93
5.3	Limitações.....	93
5.4	Perspectivas Futuras.....	95
	Referências	97
	Apêndice I - Mapeamento Sistemático sobre Medidas e Indicadores de Testes de Software em Projetos Ágeis.....	101
	Apêndice II - Fichas do conjunto de Indicadores seguindo modelo ASM.br.....	157
	Apêndice III - Planilhas de apoio para coleta de medidas e geração do gráfico dos Indicadores.....	179

ÍNDICE DE FIGURAS

Figura 1 - Fases da abordagem metodológica proposta	5
Figura 2 - Paradigma GQM. Fonte: Adaptado de (Basili et al., 1994).....	12
Figura 3 - Identificador da Organização Fonte: (Bonelli, 2014)	15
Figura 4 - Mapa de Objetivos. Fonte: (Bonelli, 2014)	16
Figura 5 - Mapa de Indicadores. Fonte: (Bonelli, 2014)	17
Figura 6 - Exemplo de Ficha de Registro de Indicador. Fonte: (Bonelli, 2014)	18
Figura 7 - Template para Registro de Indicadores. Fonte: (Bonelli, 2014)	19
Figura 8 - Exemplo Indicador e Necessidade de Informação. Fonte (Bonelli, 2014)	19
Figura 9 - Exemplo preenchimento das categorias. Fonte (Bonelli, 2014)	20
Figura 10 - Exemplo preenchimento objetivos e tipos. Fonte (Bonelli, 2014)	21
Figura 11 - Exemplo preenchimento Nível do MR-MPS-SW. Fonte (Bonelli, 2014) ...	22
Figura 12 - Exemplo de representação gráfica de um indicador. Fonte (Bonelli, 2014)	22
Figura 13- Procedimento de Medição e Análise. Fonte (Bonelli, 2014)	23
Figura 14 - Procedimento de análise baseado em critérios. Fonte (Bonelli, 2014).....	24
Figura 15 - Procedimento de análise baseado em relacionamentos. Fonte (Bonelli, 2014)	25
Figura 16 - Ciclo de vida do SCRUM (Schwaber,1995).....	26
Figura 17 - Ciclo desenvolvimento tradicional e baseado em testes. Adaptado de (Cohn, 2014).....	31
Figura 18 - Fluxo de pesquisa e leitura dos artigos no Ciclo 2 do mapeamento.....	45
Figura 19 - Gráfico com a quantidade de medidas citadas por artigo e novas medidas.	51
Figura 20 - Gráfico com a quantidade de citações em artigos de cada medida.....	51
Figura 21 - Gráfico com a quantidade de artigos por ano de publicação	52
Figura 22 - Gráfico indicadores x medidas referenciadas nos artigos.....	55
Figura 23 – Mapa de objetivos e indicadores	61
Figura 24 – Mapa de relacionamento e indicadores	63
Figura 25 - Exemplo de ficha indicador TD – Taxa de defeitos.....	65
Figura 26 - Exemplo de planilha de medição: Taxa de defeitos.....	66
Figura 27 - Termo de consentimento para participação do estudo	75
Figura 28 - formulário de perfil dos participantes.....	76
Figura 29 - Gráfico de formação acadêmica dos participantes	77
Figura 30 - Gráfico de tempo de experiência dos participantes	77
Figura 31 - Ficha de avaliação dos indicadores.....	78
Figura 32 - Gráfico estudo de caso TACT.....	85

Figura 33 - Gráfico estudo de caso TD.....	86
Figura 34 - Gráfico estudo de caso TCEX	87
Figura 35 - Gráfico estudo de caso RR	88
Figura 36 - Fluxo de pesquisa e leitura dos artigos no Ciclo 2 do mapeamento	113
Figura 37 - Gráfico com a quantidade de medidas citadas por artigo e novas medidas.....	127
Figura 38 - Gráfico com a quantidade de citações de cada medida	127
Figura 39 - Gráfico com a quantidade de artigos por ano de publicação	128
Figura 40- Menu da planilha de coletas.....	179
Figura 41- Densidade de defeitos	180
Figura 42 - Esforço gasto para correção dos defeitos.....	180
Figura 43 - Esforço gasto nos testes	181
Figura 44 - Índice de cobertura de testes	181
Figura 45 - Quantidade de defeitos não cobertos por casos de teste	182
Figura 46 - Quantidade de defeitos reportados pelo cliente	182
Figura 47 - RR – Prontidão para Release	183
Figura 48 - Taxa de aumento de casos de testes entre releases	183
Figura 49 - Taxa de casos de testes executados em relação ao planejado	184
Figura 50 - Tempo para correção dos testes	184
Figura 51 - Taxa de casos de testes automatizados.....	185
Figura 52 - Taxa de defeitos	185
Figura 53 - Tempo entre detecção de defeitos pelo cliente	186
Figura 54 - Taxa de funcionalidades que passam nos testes de aceitação automatizados	186
Figura 55 - Tempo médio para correção de um defeito.....	187

ÍNDICE DE TABELAS

Tabela 1 - Critérios para uso das categorias selecionadas. Fonte (Bonelli, 2014).....	20
Tabela 2 - Notação para o grau de relacionamento. Fonte (Bonelli, 2014).....	20
Tabela 3 - Notação para descrever o Procedimento de Medição e Análise. Fonte (Bonelli, 2014).....	23
Tabela 4 - Lista de medidas de testes ágeis. Fonte: (Kupiainen et al., 2015).....	33
Tabela 5 - Medidas ágeis. Fonte: (Kupiainen et al., 2015).....	34
Tabela 6 - Lista Medidas. Fonte (Mota e Dias-Neto, 2014).....	36
Tabela 7 - Artigos de controle citados em (Mota e Dias-Neto, 2014).....	43
Tabela 8 – Exemplo de formulário de catalogação das publicações	44
Tabela 9 – artigos utilizados no mapeamento sistemático.....	45
Tabela 10 – Lista de medidas mapeadas após exclusão de medidas repetidas.....	47
Tabela 11 – Lista de Indicadores definidos com base na análise das medidas mapeadas	53
Tabela 12 - Lista das medidas descartadas e justificativa.	54
Tabela 13 - Objetivos propostos por BONELLI (2014) selecionados para o estudo	56
Tabela 14 - Objetivos de medição e Necessidades de Informação para o Objetivo Aumentar qualidade do software	57
Tabela 15 - Objetivos de medição e Necessidades de Informação para o Objetivo Diminuir o tempo de entrega do software	57
Tabela 16 - Objetivos de medição e Necessidades de Informação para o Objetivo Aumentar a eficácia da gestão	57
Tabela 17 - Processos do MR-MPS-SW selecionados	58
Tabela 18 - Medidas utilizadas para criação dos indicadores.....	58
Tabela 19 - Mapa de relacionamento entre indicadores e processos do MR-MPS-SW .	62
Tabela 20 - Estrutura GQM para avaliação da etapa execução	72
Tabela 21 - Métrica para respostas ao formulário de avaliação	73
Tabela 22 - Distribuição das fichas de indicadores para análise	79
Tabela 23 - Respostas da avaliação dos indicadores	80
Tabela 24 - Percentuais obtidos de cada questão.....	80
Tabela 25 - Tempo gasto na análise das fichas	84
Tabela 26 - Classificações das medidas. Fonte: (Mota e Dias-Neto, 2014)	109
Tabela 27 - Artigos de controle citados em (Mota e Dias-Neto, 2014).....	110
Tabela 28 - Medidas excluídas após revisão do mapeamento Ciclo 1	111
Tabela 29 - Publicações lidas e analisadas do mapeamento (Mota e Dias-Neto, 2014)	111

Tabela 30 - Publicações retornadas na execução do 2º ciclo (2014-2016).....	114
Tabela 31 - Publicações que contribuíram na identificação de medidas do 2º ciclo	121
Tabela 32 - Lista de publicações que contribuíram com medidas nos ciclos 1 e 2	121
Tabela 33 - Lista de medidas mapeadas após exclusão de medidas repetidas	123
Tabela 34 - Lista dos indicadores definidos e respectivas medidas.	129
Tabela 35 - Lista das medidas descartadas e justificativa	131
Tabela 36 - Lista de Indicadores definidos com base na análise das medidas mapeadas	132

1 Introdução

Este capítulo introdutório descreve a dissertação e esclarece em que cenário o resultado do trabalho se aplica e a quais problemas são focados. Desta forma, na Seção 1.1 é apresentado o contexto e motivação em que o trabalho se aplica. A Seção 1.2 descreve o objetivo e questão de pesquisa. Na Seção 1.3 é descrita a abordagem metodológica. Por fim, a Seção 1.4 apresenta a organização da dissertação.

1.1. Contexto e Motivação

Um dos grandes desafios enfrentados pelas organizações que desenvolvem software é encontrar formas de fornecer mais funcionalidades aos seus clientes, desenvolver os produtos e manter a qualidade do software (RATHOD, 2015). A presença de defeitos, indicativo de falta de qualidade, é um fenômeno comum que ocorre durante o processo de desenvolvimento de software. Defeitos levam a várias consequências afetando a entrega do produto, podendo causar ainda falha total no produto devido à utilização de técnicas de remoção de defeitos não apropriadas (POOJA, 2016).

Sistemas evoluem com o tempo e, em geral, são mantidos durante décadas. Sem a aplicação de ações efetivas, a qualidade do sistema cai gradualmente e o custo de manutenção aumenta. Para evitar a queda da qualidade, controle de qualidade contínuo é necessário durante o desenvolvimento e depois durante a manutenção (STEIDL *et al.*, 2014). Neste cenário, o time de garantia de qualidade tem grande responsabilidade em reproduzir a forma de uso dos clientes e, assim, minimizar a descoberta de surpresas desagradáveis (ROCHE, 2013).

Uma das formas de alcançar a visibilidade do andamento ou dos resultados de um projeto de software é por meio da medição de software, que pode ser definida como uma avaliação quantitativa de qualquer aspecto dos processos e produtos da Engenharia de Software (BASS, 1999). Segundo Vicente (2010), uma das práticas que apoia a melhoria contínua de processos de software é a coleta de dados sobre o andamento do projeto por

meio de medidas de software. A medição permite o melhor entendimento do que se produz e de como é produzido, com isso auxilia o planejamento, controle e melhoria dos projetos (BONELLII, 2014). Através de medições, pode-se medir o progresso e apontar melhorias e dificuldades durante todo o projeto. Também é possível acompanhar e gerenciar a qualidade do produto sendo desenvolvido.

Teste de software é o processo de execução de um programa com a finalidade de encontrar erros, no qual se aumenta a confiabilidade do software encontrando erros e possibilitando que sejam corrigidos (BORTOLUCI, 2015). Segundo o TMMI Foundation (TMMI, 2012), as atividades de teste de software geralmente são responsáveis por 30% a 40% dos custos de um projeto, o que motiva a aplicação de estratégias de medição nas fases de teste. Estas estratégias fornecem insumos para gerenciar os recursos dos projetos e garantir a qualidade e melhoria constante tanto do produto de software quanto do próprio processo de testes. Além disso, no contexto do processo de testes de software, as medidas também permitem fornecer um feedback instantâneo a respeito do software que está sendo desenvolvido e testado (CRISPIN, 2009).

No ambiente de negócios da indústria, a busca por eficiência nos processos de software está inserida em um contexto no qual ocorrem mudanças frequentes de prioridades, alteração do escopo de projetos e, demanda crescente por qualidade, instigando as equipes de desenvolvimento de software na formulação de estratégias para atender estas solicitações em um curto espaço de tempo, com isso as abordagens ágeis se tornam uma solução interessante para as organizações (LIMA, 2013).

Alguns dos métodos ágeis mais utilizados são Scrum (SCHWABER, 1995), Lean (POPPENDIECK, 2003), Kanban (OHONO, 1997) e XP (Extreme Programming) (BECK, 2004). Esses métodos recomendam que todas as pessoas de um projeto (programadores, gerentes, equipes de homologação e até clientes) trabalhem controlando a qualidade do produto todos os dias e a todo momento, pois acreditam que prevenir defeitos é mais fácil e barato que identificá-los e corrigi-los (BERNARDO, 2008). A Programação eXtrema, em particular, recomenda explicitamente testes automatizados para ajudar a garantir a qualidade dos sistemas de software. Vale ressaltar ainda, que os métodos ágeis não se opõem a quaisquer revisões adicionais que sejam feitas para aumentar a qualidade (BERNARDO, 2008).

Em geral as abordagens ágeis valorizam os testes de software como um dos caminhos para equilibrar a leveza dos respectivos processos e oferecer uma compensação para se alcançar um pouco mais de confiança no desenvolvimento ágil. Contudo, cada método apresenta diferentes níveis de detalhe sobre as atividades de teste. As práticas de software adotadas nos processos de desenvolvimento podem influenciar no processo de testes, mesmo aquelas práticas que não estão diretamente associadas com alguma atividade do processo de testes. Por exemplo, a implantação diária que é descrita como a prática de colocar código novo em produção todas as noites, não faz referência diretamente a qualquer atividade de teste. Contudo, não faria sentido colocar código em produção sem testar, pois, estaria contrariando os objetivos de qualidade almejados para o software desenvolvido com a metodologia ágil que recomendou a prática (ABRANTES, 2012).

Muitas pesquisas são relacionadas à medição no âmbito geral de gerenciamento de projetos de software não focando em medição relacionada a teste de software. Esta lacuna cria uma necessidade para equipes que fazem gestão de projetos de testes de software em possuir indicadores¹ mais apropriados ao gerenciamento dos testes e que apoiem na análise do progresso das atividades de testes executadas.

Existem vários frameworks (normas e modelos de maturidade) para apoio à implementação de processos de melhoria de software nas organizações que incluem práticas de medição, por exemplo, CMMI-DEV (CMMI Institute, 2011), MR-MPS-SW (SOFTEX, 2016), TMMI (TMMI, 2012). Estes modelos abordam a medição como prática a ser implantada, porém não citam quais medidas e indicadores a serem implementados. Também não detalham como aplicá-los especificamente à fase de teste de um processo de desenvolvimento de software.

Foram identificados trabalhos como (BORTOLUCI, 2015) cuja pesquisa teve intuito de verificar métodos ágeis, abordagens de teste e medidas de teste mais utilizados através de um *survey* aplicados a desenvolvedores e testadores em projetos de desenvolvimento de software. Esta pesquisa foi feita no âmbito de desenvolvimento e testes e identificou 6 medidas relacionadas a testes: quantidade de defeitos encontrados, tempo de execução de testes, funcionalidade testadas e entregues, quantidade de casos de teste e assertivas, fator de

¹ Um indicador pode ser definido como uma medida, ou combinação de medidas, normalmente apresentado na forma de gráfico ou tabela, que provê entendimento a respeito de uma questão ou conceito de software (ISO/IEC, 2002) (MCGARRY *et al.*, 2012).

teste² e cobertura de código. No *survey* foi apenas observado quais medidas são mais utilizadas, mas os autores não descrevem como coletá-las e calculá-las. Também não era objetivo do trabalho explorar a utilização das medidas em projetos.

Vicente (2010) caracteriza a atividade de teste de software identificando boas práticas e propôs adoção de um conjunto de medidas para facilitar o acompanhamento e melhoria constante dos testes. Entretanto, o foco do trabalho está em medidas relacionadas a código de software, como, por exemplo, cobertura de código. O autor identificou e implementou 14 medidas em uma ferramenta de automação de gerenciamento de iterações de desenvolvimento ágil. Com isso, as validações em tempo de teste foram focadas ao teste do código aplicado ao contexto de métodos ágeis.

Mota e Dias-Neto (2014) identificaram, por meio de um mapeamento sistemático da literatura, 24 medidas utilizadas em projetos de desenvolvimento de software com métodos ágeis e que podem ser aplicadas no processo de teste. Apesar de o trabalho listar medidas, não foram apresentados os detalhes referentes à coleta ou análise, nem foram identificadas medidas que apoiem a gestão de testes de software.

Por meio de disponibilização de um template para registro de indicadores Bonelli (2014) (BONELLI *et al.*, 2017) apoiou organizações na criação de seus planos de medição utilizando como referência os diferentes níveis do modelo MR-MPS-SW. O foco do trabalho foi apoiar o entendimento, identificação e padronização de indicadores em projetos de software de forma geral, focando fortemente em planejamento e custo nos projetos não incluindo medidas específicas para tratar os resultados das atividades de teste.

1.2. Objetivo e Questão de Pesquisa

O objetivo desta dissertação é descrever indicadores que apoiem as organizações nas fases de teste em projetos que utilizem metodologia ágil, com informações detalhadas que possibilitem a coleta e análise. Desta forma, tem-se a seguinte questão de pesquisa formulada:

(QP1) “Que indicadores relacionados a teste podem ser utilizados durante o desenvolvimento ágil de software?”

² Fator de teste serve para indicar a relação entre o tamanho do código de testes e tamanho do código em produção com o objetivo de verificar a evolução do código de teste e o esforço para criação de testes (BORTOLUCI, 2015).

Para atender a esse objetivo será criado um conjunto de fichas de indicadores de testes para projetos ágeis baseadas no *template* ASM.br (BONELLI, 2014; BONELLI *et al*, 2017) criadas a partir de um mapeamento sistemático da literatura e avaliado no contexto de um estudo de caso, conforme abordagem metodológica descrita na seção a seguir.

1.3. Abordagem Metodológica

A abordagem metodológica utilizada nesta dissertação pode ser vista na Figura 1.

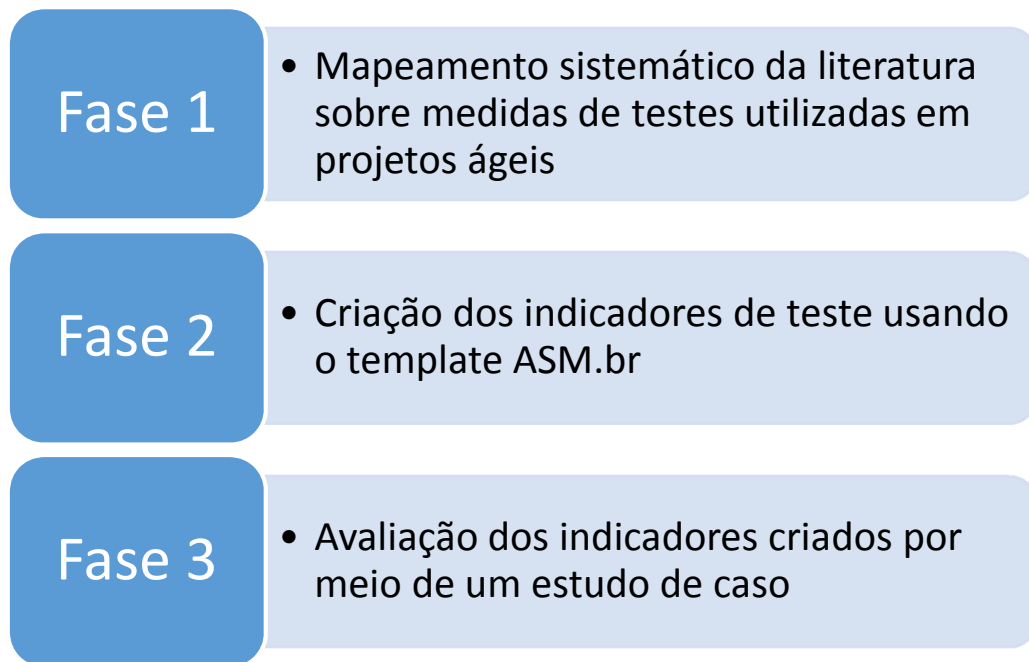


Figura 1 - Fases da abordagem metodológica proposta

A Fase 1 compreende 3 etapas:

- Replicação do mapeamento executado por Mota e Dias-Neto (2014) para identificar novas medidas relacionadas a testes em projetos ágil de desenvolvimento de software e validar os resultados do mapeamento descrito em (Mota e Dias-Neto, 2014).
- Atualização do mapeamento executado por Mota e Dias-Neto (2014) para incluir publicações entre os anos de 2014 e 2016.
- Revisão dos artigos identificados tanto na execução original do mapeamento sistemático, quanto na reexecução, para detalhar, sempre que possível, a forma de coleta das medidas, e procedimentos de análise.

A Fase 2 compreende a geração de uma lista de indicadores relacionados a testes oriundas do mapeamento sistemático da literatura. A partir dessa lista, serão criadas fichas baseadas no *template* ASM.br (BONELLI, 2014; BONELLI *et al.*, 2017) com informações de coleta e análise no contexto de projetos de desenvolvimento de software ágeis.

Na Fase 3, é realizada a avaliação das fichas dos indicadores por meio de um estudo de caso. Nesse estudo de caso, as fichas criadas serão utilizadas em uma organização que execute testes de software em projetos de desenvolvimento de software com metodologia ágil. Como resultado, espera-se que gerentes de projetos e líderes de projetos de testes avaliem o quanto os indicadores ajudam o gerenciamento das atividades de teste e se as recomendações presentes nas fichas são adequadas e relevantes.

1.4. Organização da Dissertação

Esta dissertação está organizada em 5 capítulos e 3 apêndices. O presente capítulo introdutório apresentou o contexto que motivou a elaboração deste trabalho, bem como objetivo e os procedimentos metodológicos utilizados. As próximas seções da dissertação estão organizadas da seguinte forma:

- **CAPÍTULO 2 – Medição de Software e Testes em Projetos Ágeis:** apresenta uma introdução sobre medição de software e atividades de testes de software no contexto de projetos ágeis.
- **CAPÍTULO 3 – Indicadores de Testes em Projetos Ágeis:** apresenta o desenvolvimento dos indicadores de teste em projetos ágeis.
- **CAPÍTULO 4 – Avaliação dos Indicadores de Testes para Projetos Ágeis:** apresenta o estudo de caso executado no contexto dessa dissertação.
- **CAPÍTULO 5 – Conclusão** – apresenta as considerações finais, contribuições e trabalhos futuros.
- **APÊNDICE I - Mapeamento Sistemático sobre Medidas e Indicadores de Testes de Software em Projetos Ágeis** – apresenta os resultados do mapeamento sistemático da literatura desenvolvido no contexto dessa dissertação.

- **APÊNDICE II – Fichas do conjunto de Indicadores seguindo modelo ASM.br:**
Apresenta o conjunto de fichas com o registro de todos os indicadores propostos nessa dissertação.
- **APÊNDICE III – Planilhas de apoio para coleta de medidas e geração do gráfico dos Indicadores:** Conjunto de planilhas criadas para coleta das medidas e cálculo dos indicadores e geração dos gráficos.

2 Medição de Software e Testes em Projetos Ágeis

Este capítulo descreve uma revisão da literatura sobre medição de software na Seção 2.1, na Seção 2.2 aborda sobre plano de medição, apresenta o template ASM.br na Seção 2.3, na Seção 2.4 descreve os métodos ágeis e na Seção 2.5 aborda tipos de testes de software e testes em projetos ágeis, na Seção 2.6 apresenta trabalhos que referenciam medidas em projetos ágeis e medidas de testes e finaliza na Seção 2.7 com considerações finais sobre o referencial teórico deste capítulo.

2.1. Medição de Software

Medição de software é uma avaliação quantitativa de qualquer aspecto dos processos e produtos da Engenharia de Software, que permite seu melhor entendimento e, com isso, auxilia o planejamento, controle e melhoria do que se produz e de como é produzido (BASS, 1999). A medição em um projeto de software pode apoiar estimativas, o controle de qualidade, a produtividade de equipe e o controle do projeto, além disso, um conjunto de medidas bem projetado pode ser utilizado para medir a qualidade de produtos de trabalho, dar apoio à tomada de decisão dos gerentes de projeto e aumentar o retorno de investimento do produto (VICENTE, 2010).

Medições devem ser planejadas e coletadas de forma organizada, para que possam ser monitoradas e ajustadas de acordo com os objetivos e necessidades da organização (MOTA e DIAS-NETO, 2014). No contexto da medição de software, um objetivo pode ser um Objetivo Estratégico, um Objetivo de Software ou um Objetivo de Medição. Um objetivo estratégico expressa a intenção pela qual ações estratégicas são planejadas e realizadas (BARCELLOS, 2009). Como exemplo, tem-se o objetivo “aumentar o número de clientes em 10%”. Um objetivo de software expressa a intenção pela qual ações relacionadas a área de software de uma organização são planejadas e realizadas. O objetivo “obter avaliação MPS.BR nível B” é um exemplo de objetivo de software. Por fim, um objetivo de medição expressa a intenção pela qual ações relacionadas a medição de software são planejadas e realizadas. Como exemplo, tem-se o objetivo “monitorar o desempenho dos processos

críticos”. Objetivos de software e objetivos de medição são definidos com base em objetivos estratégicos, sendo que objetivos de medição também podem ser definidos com base em objetivos de software (BARCELLOS, 2009).

Medidas podem ser utilizadas para indicar o alcance a objetivos. Quando isso ocorre, a medida desempenha o papel de indicador (BARCELLOS, 2009). Segundo McGarry *et al.* (2012) um indicador é uma medida ou uma combinação de medidas, normalmente apresentado na forma de gráfico ou Tabela, que provê entendimento a respeito de uma questão ou conceito de software. Um exemplo de um indicador seria um gráfico que apresenta o número de defeitos de código detectados em um determinado período de tempo. Os indicadores representam produtos de informação que atendem as necessidades de informação inicialmente identificadas e são registrados em um plano de medição (ROCHA *et al.*, 2012).

Os termos medida, métrica, medição e indicador, costumam causar dúvidas quanto à correta utilização. Em função disso, relacionamos abaixo as definições de cada um deles:

- Medida – é a variável para a qual o valor é atribuído como resultado de uma medição (ISO/IEC 15939, 2002).
- Métrica – é um atributo (propriedade ou característica) mensurável de uma entidade (produto ou processo). No caso de um projeto, uma métrica pode ser, por exemplo, o seu tamanho, uma vez que pode ser medido (MAIA, 2009).
- Medição – é o ato de medir, de determinar uma medida. É o conjunto de operações com o objetivo de determinar o valor de uma medida (ISO/IEC 15939, 2002).
- Indicador – é a medida utilizada para alcance de um objetivo (BARCELLOS, 2009). Informação relacionada a uma medida, métrica ou combinação de métricas, que pode ser utilizada para compreender a entidade que está sendo medida. Provê uma visão dos processos de desenvolvimento de software e das atividades para melhoria deste processo (BRAUN, 2007).

Segundo Farroha (2014), indicadores de desempenho são usados para avaliar o sucesso da nova cultura em organizações, por exemplo: Melhoria na qualidade de software, frequência de entrega de software, capacidade de resposta às necessidades do negócio, qualidade do código, o quanto rápido os defeitos são corrigidos, vulnerabilidade da estrutura a ataques.

O número de indicadores descritos na literatura é extenso e isso expõe as organizações de software ao risco de ter planos de medição com indicadores sem sinergia com seus objetivos ou com os demais projetos da organização. Essa falta de padrão pode gerar problemas de entendimento e a impossibilidade de acompanhar os resultados de todos os projetos de forma consistente. Por esse motivo, as organizações deveriam definir um conjunto predeterminado de indicadores para ajudar no planejamento e acompanhamento de suas medições de software (BONELLI, 2014).

Medição inclui monitoramento de alto nível das medidas de negócios com retorno de investimento, assim como o passo-a-passo das transações a cada minuto. Para isto é necessário a escolha de indicadores de desempenho e qualidade que possam, por exemplo, medir cobertura dos testes e reduzir o risco de erros do software que será entregue ao cliente (HUMBLE, 2011).

A análise dos resultados dos testes tem um importante papel no ciclo de desenvolvimento de software. Para o negócio que envolve software não há espaços para desculpas por falta de qualidade (RATHOD, 2015). Por esta razão as organizações utilizam diversos métodos para testes de software afim de atingir perfeição em seus produtos antes de entregar para produção (RATHOD, 2015). As organizações em geral possuem ferramentas que capturam medidas em tempo real, mas elas monitoram em silos e de maneira desconectada (SHARMA, 2015). Para a análise dos testes, os desenvolvedores precisam de relatórios de análise dos resultados e testes, que podem conter gráficos e relatórios mostrando as causas raízes dos problemas de desempenho ou qualidade (RATHOD, 2015).

Segundo Janus *et al.* (2002), em projetos que utilizam metodologias ágeis, a medição de software aponta para os principais focos do desenvolvimento ágil, como testes e integração contínua, e assim possibilitam a monitoração dos projetos. Como exemplo de indicadores relacionados pode-se citar quantidade de casos de testes, grau de cobertura de teste, taxa de crescimento de teste e implementações falhadas (JANUS *et al.*, 2012).

Segundo Ktata *et al.* (2010), é muito fácil admitir que a identificação de métricas para melhoria de software é muito chata, principalmente no contexto de ambientes ágeis. De fato, as métricas ágeis devem considerar não somente aspectos técnicos, mas também aspectos humanos. Heurísticas para uma boa medição ágil são descritas abaixo (KTATA *et al.*, 2010):

- Afirmar e reforçar os princípios do modelo Lean e princípios ágeis.
- Medir o resultado, não os números de saída.

- Seguir tendências e não números.
- Pertencer a um pequeno conjunto de métricas e diagnósticos.
- Ser fácil de coletar.
- Revelar, ao invés de ocultar, o contexto e o significado das variáveis.
- Fornecer insumos para uma conversa significativa sobre as medidas.
- Podem medir valor (produto) ou processo.
- Incentivar “ser bom o bastante” qualidade.

2.2. Criação de Planos de Medição

Organizações para serem competitivas necessitam planejar sua estratégia. Estratégia é um plano que visa dar à organização uma vantagem competitiva. Para isso a organização precisa entender o que ela faz, o que deseja se tornar e como fazer para alcançar essa meta. Uma estratégia deve identificar as metas de uma organização, a direção para organizar o seu trabalho e atingir as metas identificadas (ROCHA *et al.*, 2012). A definição da estratégia deve partir da missão da organização, seguir um processo bem definido, e chegar a um plano de ação possível de ser avaliado quanto ao desempenho (LUECKE, 2010).

O Plano de Medição define as medidas e indicadores que devem ser coletados, vincula-os aos objetivos estratégicos da organização, define como e quando os dados devem ser coletados, que deve coletar, interpretar e analisar cada medida e indicador e onde os resultados devem ser registrados.

Dois métodos são especialmente importantes para apoio ao planejamento de medições: GQM (*Goal_Question-Metric*) (BASILI *et al.*, 1994) e PSM (*Practical Software Measurement*) (MCGARRY *et al.*, 2012).

2.2.1. Método GQM (Goal Question Metric)

O método GQM (BASILI *et al.*, 1994) tem o objetivo de apoiar a definição de medidas alinhadas aos objetivos de negócio da organização. Baseia-se no entendimento de que, para realizar medições de forma adequada, uma organização necessita primeiramente definir os seus objetivos e os objetivos de seus projetos para depois relacioná-los aos dados necessários. Por fim, deve fornecer um framework para análise e interpretação dos dados com relação aos objetivos estabelecidos (ROCHA *et al.*, 2012). A abordagem GQM é um

método bastante aceito pela comunidade de engenharia de software para estudos experimentais e tem sido adotado para medir e melhorar a qualidade em organizações de desenvolvimento de software (VICENTE, 2010).

O resultado da aplicação do GQM é uma estrutura hierárquica com três níveis: 1- nível conceitual (Objetivo), 2- nível operacional (Questão), 3- nível quantitativo (Medida). Através de uma abordagem *top-down* e orientada a objetivos, são definidos objetivos que são refinados em questões. Medidas são, então, definidas de forma que sejam adequadas para responder às questões (ROCHA *et al.*, 2012):

- *Objetivo (conceitual)*: são identificados objetivos para o produto/processo/recurso.
- *Questão (operacional)*: determina questões que auxiliem na caracterização do objeto de estudo e com ele deve ser enxergado dentro do contexto de qualidade.
- *Medida (quantitativo)*: definição de métricas que irão fornecer uma resposta quantitativa de cada questão. Métricas podem ser objetivas ou subjetivas.

Na Figura 2 é exibido o paradigma GQM.

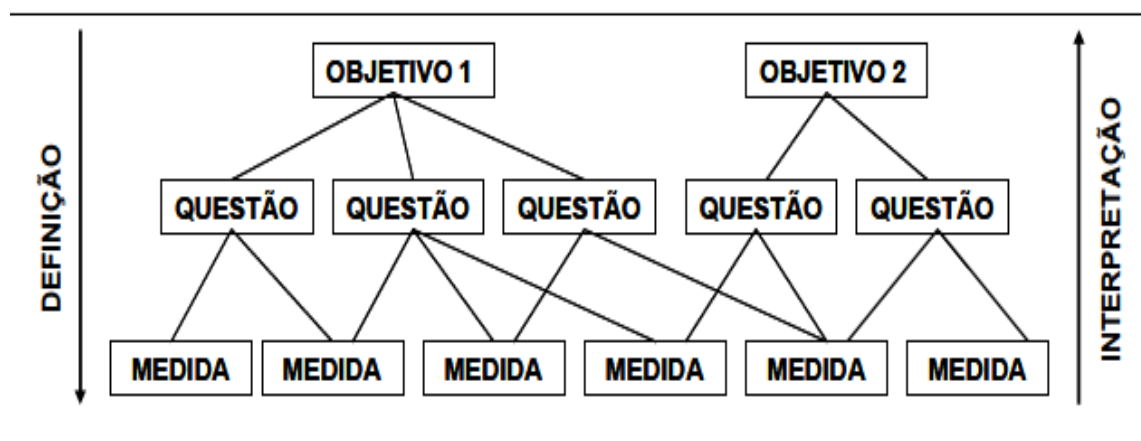


Figura 2 - Paradigma GQM. Fonte: Adaptado de (Basili *et al.*, 1994).

2.2.2. Método PSM (*Practical Software Measurement*)

O PSM – *Practical Software Measurement* (McGARRY *et al.*, 2012) é uma abordagem para medição de software orientada às necessidades de informação organizacionais aderente à ISO/IEC 15939 (ISO/IEC, 2002) e, como, ela possui dois componentes: um Modelo de Informação de Medição e um Processo de Medição (ROCHA *et al.*, 2012).

O Modelo de Informação de Medição, assim como na ISO/IEC 15939, tem como objetivo estabelecer a ligação entre as medidas definidas e as necessidades de informação identificadas (ROCHA *et al.*, 2012).

No PSM (MCGARRY *et al.*, 2012), as necessidades de informação relacionam diretamente tanto os objetivos estabelecidos como as áreas de interesse, identificando o que é necessário saber para tomar decisão e considerando os objetivos de negócio e os requisitos do cliente. Para cada necessidade de informação, pelo menos um conceito mensurável deve ser selecionado. A abordagem PSM provê um conjunto de categorias de informação com conceitos mensuráveis, que servem de ponto de partida para a definição e seleção de medidas. Também define modelos de análise, que exprimem o comportamento esperado das medidas ao longo do tempo e produzem estimativas e avaliações relevantes para as necessidades de informação. Estes modelos de análise possuem critérios de decisão associados, definidos a partir de valores limites, que determinam a necessidade de uma ação ou investigação mais detalhada e ajudam a interpretar os resultados das medições, estando baseados em dados históricos, planos ou heurísticas. Possui basicamente quatro atividades: estabelecer e manter compromisso; planejar a medição; executar a medição; e avaliar a medição (SOFTEX, 2016).

O PSM define que os 9 princípios básicos da medição são:

- Use questões e objetivos orientados aos requisitos de medição;
- Defina e colete medidas baseadas em técnica e processos de gerenciamento;
- Colete e analise dados ao nível de detalhamento capaz de identificar e isolar problemas;
- Implemente uma capacidade de análise independente;
- Use um processo de análise sistemático para rastrear a medida com a decisão;
- Interprete os resultados da medição no contexto de outros projetos de informação;
- Integre a medição ao processo de gerenciamento de projetos através do ciclo de vida;
- Use o processo de medição como uma base para os objetivos de comunicação;
- Foque inicialmente na análise ao nível de projeto.

Segundo o PSM (JONES, 2003), a medição em projetos de software deve servir como uma poderosa ferramenta aos gestores, e quando integrada a todos os processos que fazem parte do desenvolvimento do software ajuda os gerentes a identificarem riscos, problemas

específicos e o impacto desses problemas. As medidas coletadas são a base para a definição dos indicadores. Um indicador pode permitir identificar desvios no comportamento de um processo a partir de limites estabelecidos de acordo com o que está sendo analisado. Além disso, deve descrever como deve ser analisado para que seja possível realizar uma correta interpretação, além de identificar fatores que podem justificar determinados resultados inesperados (SILVA, 2013).

2.3. Template ASM.br

Embora o GQM ajude as organizações a definirem seus planos de medição, eles não definem como os indicadores devem ser especificados. O PSM, por sua vez, fornece além de um processo a ser seguido para realizar o planejamento de medição, um conjunto de medidas e indicadores que podem ser reutilizados por organizações. No entanto, informações importantes não estão disponíveis nas especificações de medidas e indicadores, tais como relações entre indicadores e entre eles e objetivos, bem como as relações podem ser usadas ao analisar dados coletados para a medida. Considerando a falta de uma maneira padronizada de registrar especificações de indicadores e a ausência de informações importantes em indicadores registrados na literatura, foi criado o ASM.br (*Assistance for Software Measurement based on relationships*, ou Apoio a Medição de Projetos de Software baseado em relacionamentos) (Bonelli, 2014) (Bonelli *et al.*, 2017).

O ASM.br é dividido em três partes, descritas a seguir: Perfil Organizacional, Mapeamento de Indicadores e Fichas de Indicadores.

2.3.1. Componente Perfil Organizacional

Segundo BONELLI (2014), o componente Perfil Organizacional é utilizado para indicar as características da organização que utilizará o AMP.br³ na representação de seus indicadores e é composto pelas seguintes seções: Nome da Organização, Repositório de arquivos e versão do Documento (vide figura 3). A finalidade do repositório é centralizar os documentos e dados que serão utilizados na composição dos indicadores, sejam os documentos, planilhas ou dados obtidos a partir de bancos de dados. A estruturação do

³ Deve-se notar que originalmente o template era chamado de AMP[br], conforme pode ser visto na parte esquerda do topo de algumas figuras desse capítulo

repositório de dados é diferente entre organizações, portanto o componente precisa armazenar qual o tipo de repositório é utilizado por uma determinada organização.

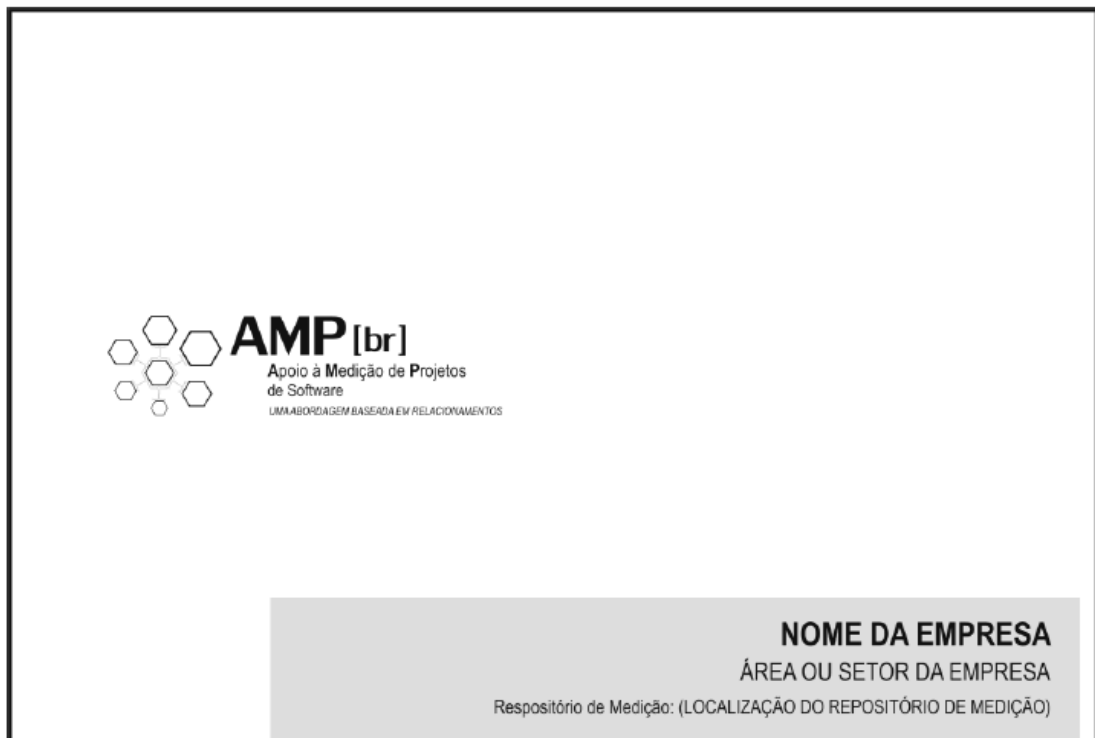


Figura 3 - Identificador da Organização Fonte: (Bonelli, 2014)

2.3.2. Componente Mapeamento de Indicadores

O componente seguinte, Mapeamento dos Indicadores, propõe a criação de três mapas: Mapa de Objetivos e Indicadores, Mapa para o MR-MPS-SW (ou CMMI) e Mapa de Indicadores. Os mapas, conforme BONELLI (2014), tem como objetivo auxiliar o gerente de projetos a selecionar os indicadores de acordo com os objetivos que devem ser monitorados, os níveis do MR-MPS-SW que devem ser contemplados e os indicadores e medidas necessários para realizar as análises.

O Mapa de Objetivos e Indicadores é responsável por fornecer uma visibilidade ao gerente de projeto sobre quais indicadores são importantes a fim de alcançar os objetivos da organização. O mapa é composto por objetivos estratégicos da organização, que por sua vez são a base da formação dos objetivos de medição. A partir desses últimos são identificadas as necessidades de informação e, por fim, os indicadores são formados a partir dessas necessidades. Logo, segundo BONELLI (2014), quando um indicador está associado a uma necessidade de informação, que por sua vez está associada a um objetivo de medição, pode-

se concluir que o indicador é utilizado para monitorar um objetivo de medição. Na Figura 4 pode ser observado um exemplo do Mapa de Objetivos preenchido.

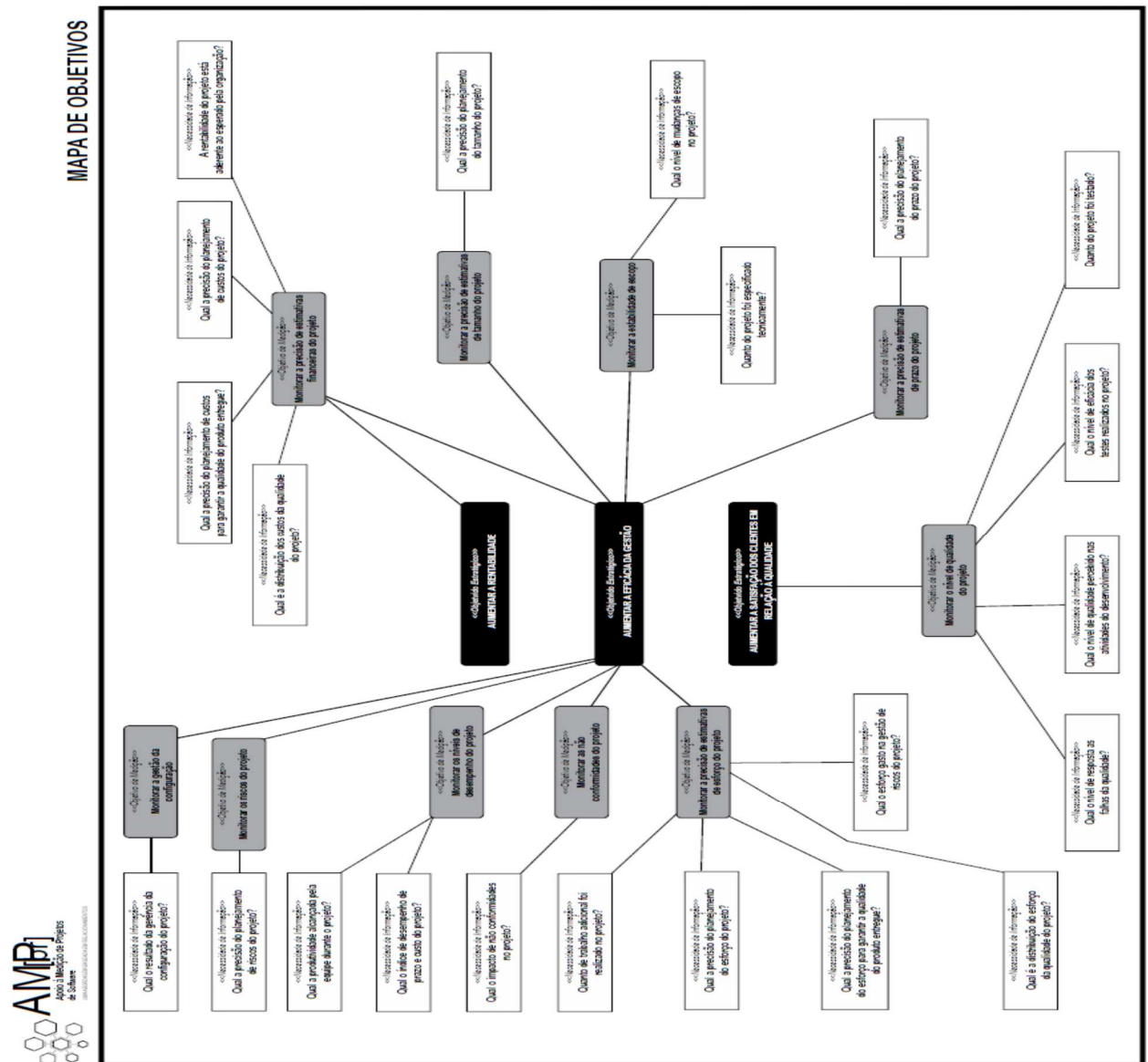


Figura 4 - Mapa de Objetivos. Fonte: (Bonelli, 2014)

O Mapa de Indicadores, conforme BONELLI (2014), tem como objetivo apresentar o relacionamento entre os indicadores considerando algumas categorias preenchidas. Foram consideradas para esse trabalho as seguintes categorias: Esforço, Escopo, Tempo, Custo, Qualidade e Produtividade. Cada indicador é representado em um retângulo, conforme mostra a Figura 5. Também é representado o relacionamento entre o indicador com uma de suas categorias, que será explorado mais à frente.

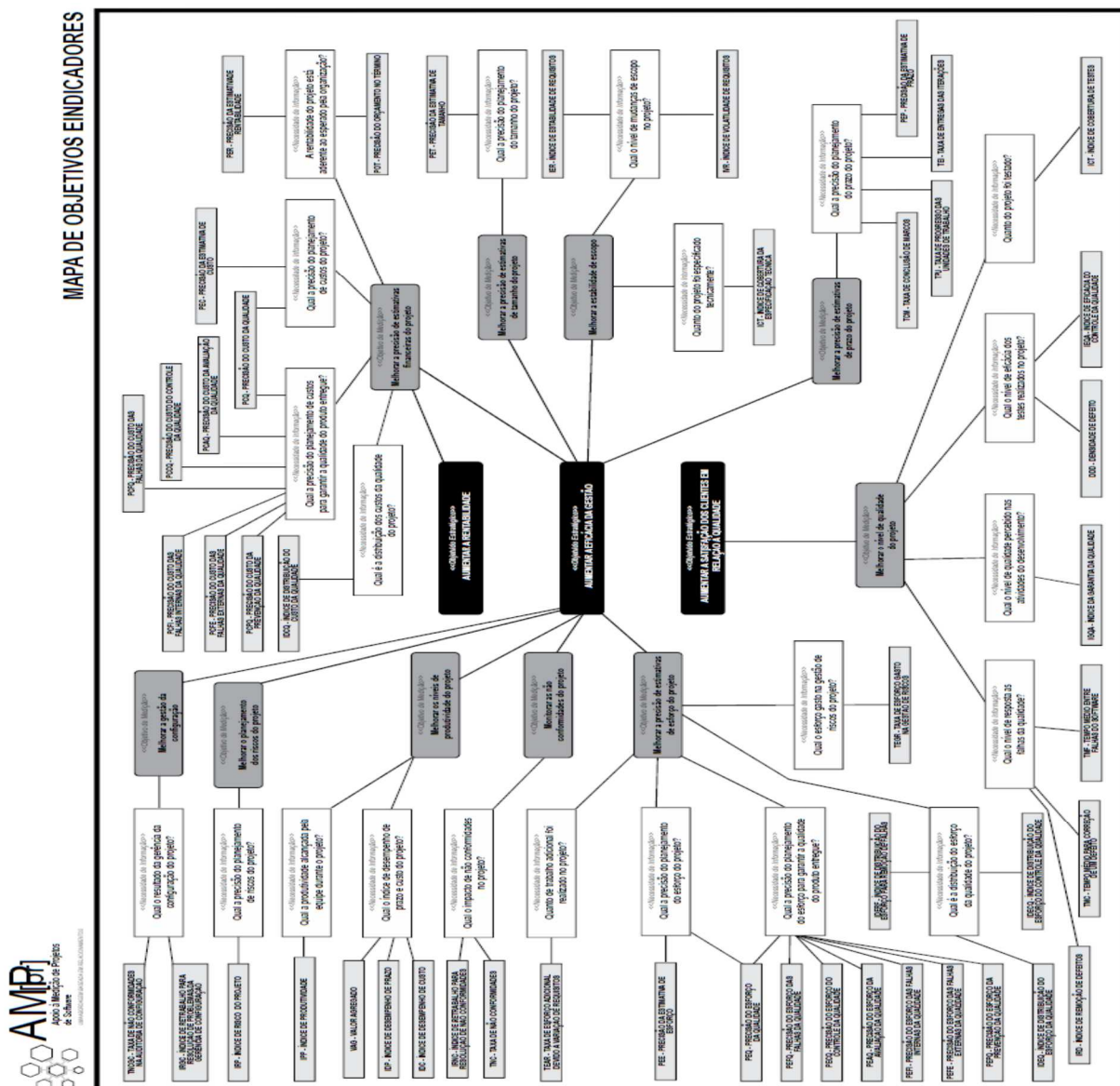


Figura 5 - Mapa de Indicadores. Fonte: (Bonelli, 2014)

2.3.3. Componente Fichas

O componente Fichas apresenta uma descrição detalhada dos indicadores, o objetivo é descrever cada indicador em uma única página (ficha), visando simplificar o acesso às informações de cada indicador. Cada ficha é chamada de Ficha de Registro de Indicador e é dividida em áreas específicas para armazenar as informações do indicador. A Figura 6 mostra um exemplo de ficha de registro de Indicadores.

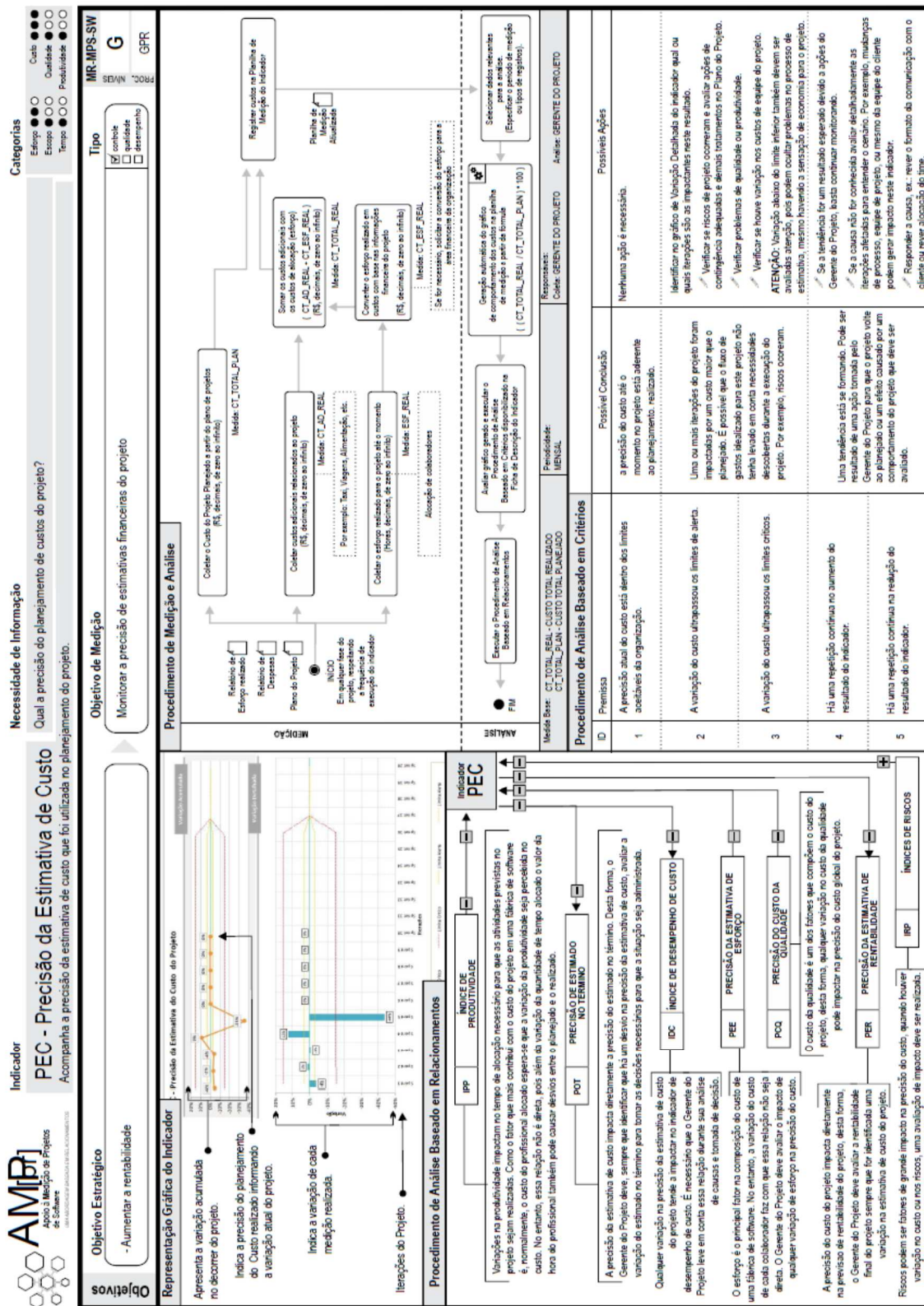


Figura 6 - Exemplo de Ficha de Registro de Indicador. Fonte: (Bonelli, 2014)

A Figura 7 apresenta um template da ficha e, em seguida, cada elemento presente nela é explicado.

The figure shows a complex form template for indicator registration. It includes sections for:

- A:** Nome e Descrição do Indicador (Indicator Name and Description)
- B:** Necessidade de Informação (Information Need)
- C:** Categorias (Categories)
- D:** Objetivos (Objectives)
- E:** Níveis do MR-MPS-SW (MR-MPS-SW Levels)
- F:** Representação Gráfica (Graphic Representation)
- G:** Procedimento de Medição e Análise (Measurement and Analysis Procedure)
- H:** Procedimento de Análise Baseado em Critérios (Criteria-Based Analysis Procedure)
- I:** Procedimento de Análise Baseado em Relacionamentos (Relationship-Based Analysis Procedure)

 The form also contains various sub-sections like 'Objetivos Estratégicos', 'Objetivos de Medição', 'Procedimento de Análise Baseado em Relacionamentos', and 'Procedimento de Análise Baseado em Critérios'.

Figura 7 - Template para Registro de Indicadores. Fonte: (Bonelli, 2014)

A. Indicador

Apresenta a sigla e o nome do indicador, bem como a sua descrição (vide Figura 8). As informações registradas devem ser concisas e diretas, visando facilitar o entendimento e a seleção do indicador pelo gerente de projetos. A sigla do indicador sempre deve formar uma combinação única de caracteres no conjunto de indicadores.

B. Necessidade de informação

A necessidade de informação atendida pelo indicador (Figura 8) representa o típico questionamento do gestor que será respondido para apoiar sua tomada de decisão.

Indicador	Necessidade de Informação
PEQ - Precisão do Esforço da Qualidade Acompanha o esforço associado com a obtenção e manutenção da qualidade em projetos.	Qual a precisão do planejamento do esforço para garantir a qualidade

Figura 8 - Exemplo Indicador e Necessidade de Informação. Fonte (Bonelli, 2014)

C. Categorias

Uma categoria é um agrupador lógico de indicadores e suas necessidades de informação. No template tem o objetivo de apresentar de forma simples e visual com quais categorias o indicador descrito se relaciona e o grau de relacionamento existente. O elemento que representa graficamente é apresentado na Figura 9.



Figura 9 - Exemplo preenchimento das categorias. Fonte (Bonelli, 2014)

A Tabela 1 apresenta os critérios para a classificação nas categorias.

Tabela 1 - Critérios para uso das categorias selecionadas. Fonte (Bonelli, 2014)

Categoria	Critérios
Produtividade	Valores reais da produtividade ou valores agregados produzidos
Tempo	Representam o prazo e estimativas indicam atrasos ou antecipações.
Custo	Representam valores monetários. Podem sinalizar variações de custo, receita, valor agregado e distribuição de investimentos.
Escopo	Representam os requisitos. Podem indicar distribuição por tipos, volatilidade ou variações de complexidade.
Qualidade	Representam os aspectos da qualidade de um projeto, tais como cobertura de testes, densidades de defeitos ou tempo para falha.
Esforço	Representam a quantidade de horas para executar o escopo

Um indicador pode estar presente em mais de uma categoria, tendo, ainda, graus de relacionamento diferentes com cada uma delas. A notação usada para indicar o grau de relacionamento encontra-se na Tabela 2.

Tabela 2 - Notação para o grau de relacionamento. Fonte (Bonelli, 2014)

Grau de relacionamento		Descrição do Relacionamento Indicador-Categoria
● ● ●	Impacto Direto	O objetivo primário do indicador está essencialmente relacionado a esta categoria.
● ● ○	Influência Direta	Não é a categoria primária do indicador, mas é perceptível a relação de causa e efeito.
● ○ ○	Influência Indireta	Não há uma relação de causa e efeito, mas é percebida uma relação conceitual ou que justifica a relação.
○ ○ ○	Sem influência	Não há uma relação conceitual ou de causa e efeito

D. Objetivos

São considerados dois tipos de objetivos: objetivos estratégicos e objetivos de medição. Um objetivo estratégico expressa a intenção com a qual ações estratégicas são planejadas e realizadas. Este elemento no template permite que haja um alinhamento entre os objetivos corporativos e os métodos de medição utilizados pelos gerentes de projetos.

Um objetivo de medição expressa a intenção pela qual ações relacionadas à medição de software são planejadas e realizadas deve ser registrado no template entre os tipos controle, qualidade e desempenho (Figura 10).

Objetivo Estratégico	Tipo
- Aumentar a eficácia da gestão	
Objetivo de Medição	☒ controle ☐ qualidade ☐ desempenho
Monitorar a precisão de estimativas de esforço do projeto	

Figura 10 - Exemplo preenchimento objetivos e tipos. Fonte (Bonelli, 2014)

E. Nível MR-MPS-SW

Este item do template tem a finalidade de apresentar para o gerente de projetos o nível MR-MPS-SW que possui resultados esperados que podem ser apoiados com o uso do indicador descrito (vide Figura 11).

A ficha também indica ao gerente de projetos como o indicador está associado ao MR-MPS-SW. Deve-se informar a sigla dos processos do MR-MPS-SW envolvidos (PROC), além de a que níveis eles pertencem (Níveis).

Na Figura 11, PROC. é a abreviação da palavra processo e identifica quais processos possuem resultados esperados que podem ser apoiados com o uso do indicador descrito. Sugere-se que sejam usadas as siglas utilizadas para identificar os processos do MR-MPS-SW, tais como: GPR – Gerência de Projetos, GRE – Gerência de Requisitos, GQA – Gerência da Qualidade, GCO – Gerência de Configuração, DRE – Desenvolvimento de Requisitos, PCP – Projeto e Construção do Produto, ITP – Integração do Produto, VER – Verificação, VAL – Validação, GRI – Gerência de Riscos, e assim por diante.

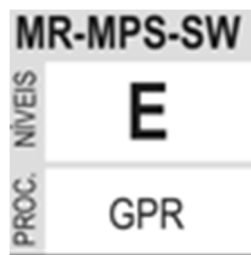


Figura 11 - Exemplo preenchimento Nível do MR-MPS-SW. Fonte (Bonelli, 2014)

F. Representação Gráfica

A representação de um indicador tipicamente é feita por meio de um gráfico, que apresenta os dados coletados facilitando a visualização e análise dos dados. Assim, o template prevê uma área específica para a apresentação de um gráfico que ilustre uma possível representação de dados para o indicador (Figura 12) auxiliando, assim, o gerente do projeto a avaliar se o indicador é ou não adequado à sua necessidade.

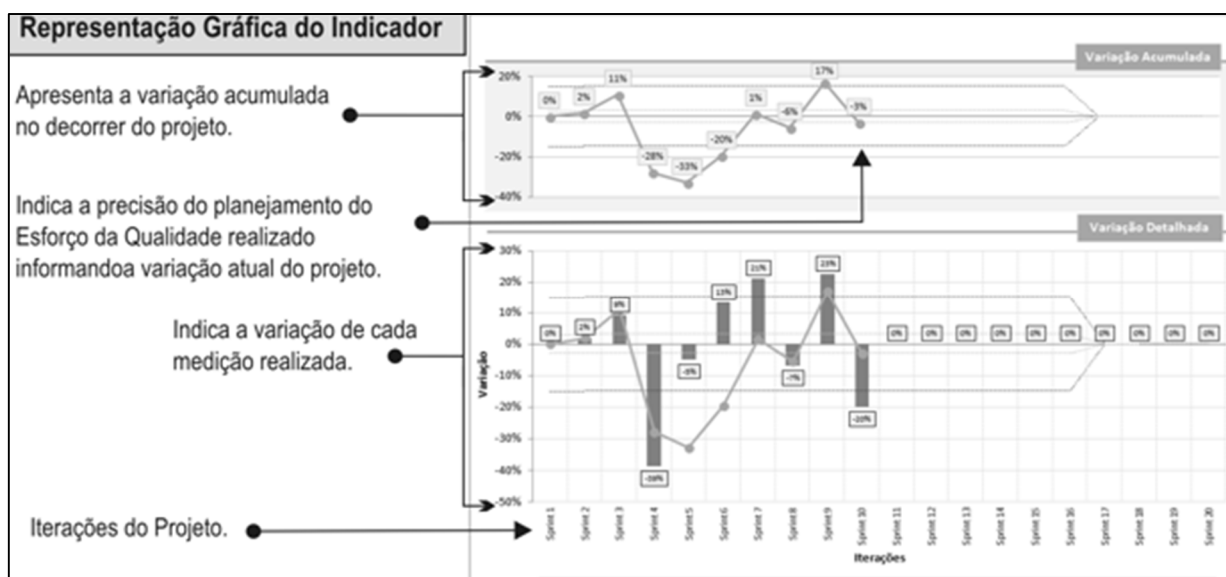


Figura 12 - Exemplo de representação gráfica de um indicador. Fonte (Bonelli, 2014)

G. Procedimento de Medição e Análise

Este elemento (Figura 13) tem o objetivo de apresentar o fluxo de atividades necessárias para que a medição, ou seja, a coleta dos dados (raia Medição), e a análise sejam realizadas adequadamente (raia Análise). A Medida Base indica as medidas necessárias para compor o indicador descrito. A Periodicidade representa a frequência temporal da execução,

por exemplo, mensal ou por iteração ou release. Os Responsáveis designam os papéis que têm a responsabilidade de realizar a medição e a análise do indicador.

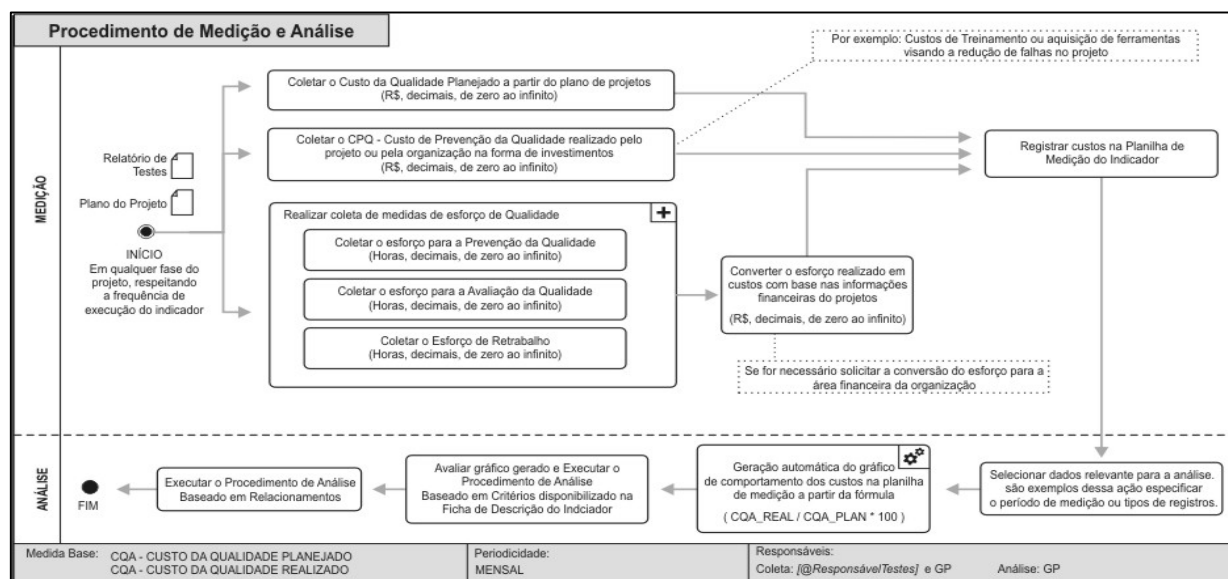


Figura 13- Procedimento de Medição e Análise. Fonte (Bonelli, 2014)

Para definir o fluxo que descreve o Procedimento de Medição e Análise, devem ser utilizados os elementos apresentados na Tabela 3 que foram baseados no BPMN (BPMN, 2014).

Tabela 3 - Notação para descrever o Procedimento de Medição e Análise. Fonte (Bonelli, 2014)

Elemento Visual	Definição
[Texto]	Exibidos próximos a outro elemento para definir alguma informação complementar.
●[Elemento Textual]	Todo procedimento deve iniciar com este elemento e deve haver um <i>Elemento Textual</i> com o nome da ação inicial do fluxo.
●	Pode haver mais de um final possível. É importante que fique claramente descrito no fluxo quais seus términos possíveis.
	Dentro deste elemento é representada a atividade que deve ser realizada em um determinado momento do procedimento.
	Quando necessário que a descrição de uma atividade inclua outras atividades, este elemento deve ser utilizado para agrupar todas as atividades.
	Atividades que sejam executadas de forma automática são representadas neste elemento.
	Indica que neste passo do procedimento deve ser tomada uma decisão. A decisão é descrita como um comentário associado a esse elemento.
	Informações complementares para esclarecer o procedimento (opcional). Utilizado pelo elemento <i>Decisão</i> .
←	Indica o fluxo a ser seguido após um elemento.
	Elemento visual que indica o que uma atividade deve receber como entrada ou ter como saída ao ser executada.

H. Procedimento de Análise baseado em critérios

Como definido na ontologia descrita em (Barcellos, 2009), um procedimento de análise dos dados coletados pode ser baseado em critérios. Dessa forma, premissas, possíveis conclusões e ações são descritas na ficha de registro dos indicadores (Figura 14). A ontologia foi adaptada em (Bonelli, 2014) para que o conceito Conclusão fosse substituído por “Possível Conclusão”, pois acredita-se que o líder de testes ou gerente de projetos possa realizar interpretações para o seu projeto. Foi incluída também a coluna “Possíveis Ações”, que indica sugestões de ações a serem tomadas no caso de a premissa ser verdadeira.

Procedimento de Análise Baseado em Critérios		
ID	Premissa	Possível Conclusão
1	A Variação Acumulada medida está dentro dos limites aceitáveis pela organização, ou seja, não atingiu os limites de Alerta ou Crítico.	O esforço da Qualidade acumulado até o momento no projeto está aderente ao planejamento realizado.
2	A Variação Acumulada do esforço da avaliação da qualidade ultrapassou os limites de alerta.	Uma ou mais iterações do projeto foram impactadas por um esforço da qualidade maior que o planejado. É possível que o fluxo de atividades idealizado para este projeto não tenha levado em conta necessidades descobertas durante a execução do projeto. Por exemplo, riscos ocorreram ou requisitos foram alterados.
3	A Variação Acumulada do esforço da avaliação da qualidade ultrapassou os limites críticos.	
4	Há uma repetição contínua no aumento da Variação Acumulada	Uma tendência está se formando. Pode ser resultado de uma ação tomada pelo Gerente do Projeto para que o projeto volte ao planejado ou um efeito causado por um comportamento do projeto que deve ser avaliado.
5	Há uma repetição contínua na diminuição da Variação Acumulada	

Possíveis Ações		
Nenhuma ação é necessária.		
Identificar no gráfico de Variação Detalhada do indicador qual ou quais iterações geraram impacto neste resultado.		
• Verificar se riscos de projeto ocorreram e avaliar ações de contingência adequadas e demais tratamentos no Plano do Projeto. • Verificar problemas nos requisitos e seu impacto na qualidade. • Verificar se houve variação na produtividade da equipe do projeto.		
ATENÇÃO: Variação abaixo do limite inferior também devem ser avaliadas atenção, pois podem ocultar problemas no processo de estimativa, mesmo havendo a sensação do projeto estar adiantado.		
• Se a tendência for um resultado esperado devido a ações do Gerente do Projeto, basta continuar monitorando. • Se a causa não for conhecida avaliar detalhadamente as iterações afetadas para entender o cenário. Por exemplo, mudanças de processo, equipe de projeto, ou mesmo da equipe do cliente podem gerar impacto neste indicador. • Responder a causa, ex.: rever o formato da comunicação com o cliente ou rever alocação do time.		

Figura 14 - Procedimento de análise baseado em critérios. Fonte (Bonelli, 2014)

I. Procedimento de Análise baseado em relacionamento

O Procedimento de Análise Baseado em Relacionamentos (Figura 15) foi incluído no template com o objetivo de complementar os procedimentos de análise considerados na Ontologia de Referência para Medição de Software (Barcellos, 2009) e no PSM (MCGARRY, 2012). Os elementos da notação são os mesmos usados para descrever o Procedimento de Medição e Análise, sendo acrescentadas as linhas bipartidas, que têm o objetivo de registrar as sugestões de análise para que o possível impacto dos relacionamentos em outros indicadores seja avaliado.

Os sinais de menos (-) e mais (+) nas extremidades de cada linha representam o comportamento esperado entre os indicadores devido as relações existentes. No exemplo da

Figura 15 espera-se que em caso de uma redução da PEQ (Precisão do Esforço da Qualidade) haja um aumento do IRP (Índice de Riscos do Projeto). No entanto, espera-se que os demais indicadores relacionados também sofram uma redução em seus resultados.

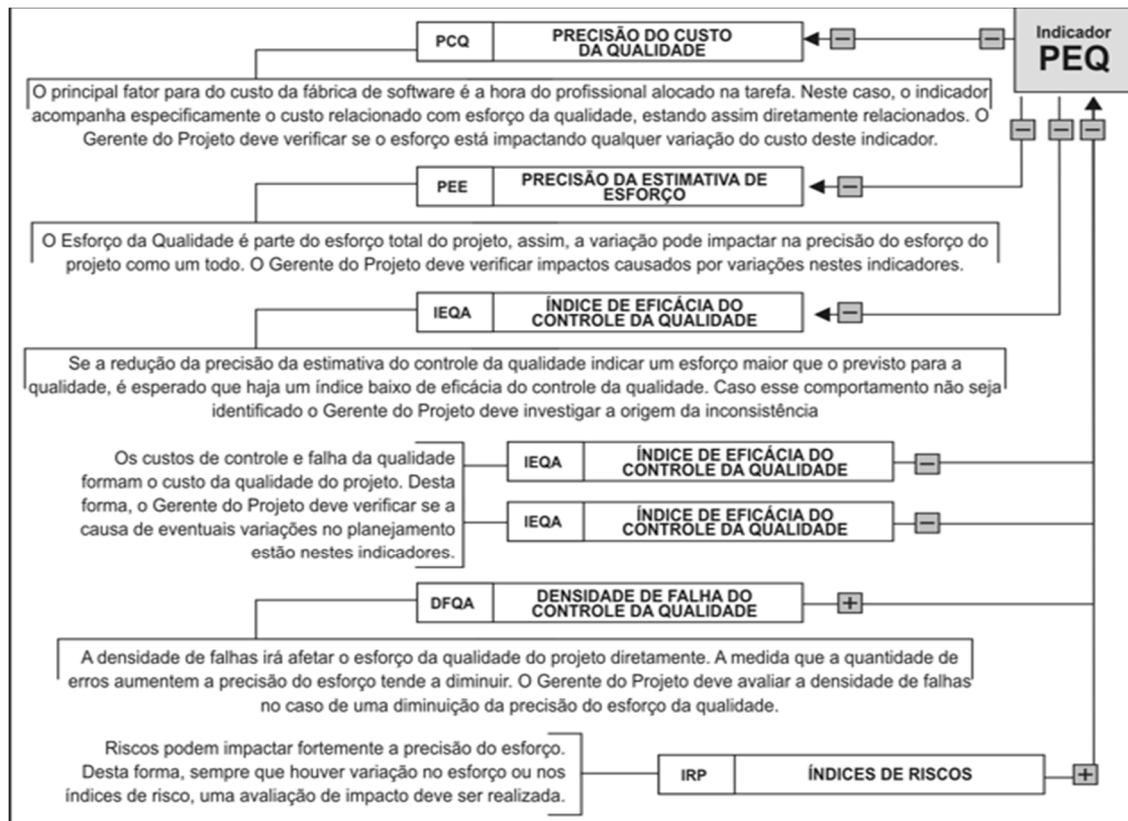


Figura 15 - Procedimento de análise baseado em relacionamentos. Fonte (Bonelli, 2014)

2.4. Metodologias Ágeis

Diversas metodologias ágeis foram desenvolvidas a fim de diminuir fraquezas encontradas na engenharia de software convencional (HENDRICKSON, 2004). As metodologias ágeis de desenvolvimento de software se destacam dos processos de desenvolvimento tradicionais devido, principalmente, ao fato de darem prioridade ao desenvolvimento de funcionalidades através de código executável ao invés da produção de extensa documentação escrita, e ainda, respostas rápidas às mudanças e colaboração com o cliente ao invés de seguir planos rígidos e negociações contratuais (LEAL 2009).

Baseado em um entendimento comum descrito pelo Manifesto Ágil (AGILE MANIFESTO, 2001), o ágil é aplicado pela introdução de algumas práticas que se aplicam a quaisquer projetos. Alguns dos Métodos ágeis mais utilizados são Scrum

(SCHWABER,1995), Lean (POPPENDIECK, 2003), Kanban (OHONO, 1997) e XP (Extreme Programming) (BECK, 2004). Esses métodos recomendam que todas as pessoas de um projeto (programadores, gerentes, equipes de homologação e até clientes) trabalhem controlando a qualidade do produto todos os dias e a todo momento, pois acreditam que prevenir defeitos é mais fácil e barato que identificá-los e corrigi-los (BERNARDO, 2008).

O Scrum (SCHWABER, 1995) é um processo de gerenciamento de projetos de software, que se concentra principalmente nos aspectos gerenciais em ambientes de negócio que mudam constantemente. Esse gerenciamento é feito por meio de iterações de duas semanas ou 30 dias, chamados Sprints, com monitoramento diário por intermédio de reuniões em pé (*stand-up meetings*) (VICENTE, 2010). O Scrum assume que o conhecimento vem da experiência e a tomada de decisões é feita com base no que é conhecido. O desenvolvimento do produto é feito de forma iterativa, em ciclos denominados sprints (COSTINHAS, 2014).

O coração do Scrum é a Sprint, um *time-box* de um mês ou menos, durante o qual um “Pronto”, versão incremental potencialmente utilizável do produto, é criado. Sprints têm durações coerentes em todo o esforço de desenvolvimento. Uma nova Sprint inicia imediatamente após a conclusão da Sprint anterior. A Figura 16 apresenta o ciclo de vida do Scrum (SOARES, 2015).

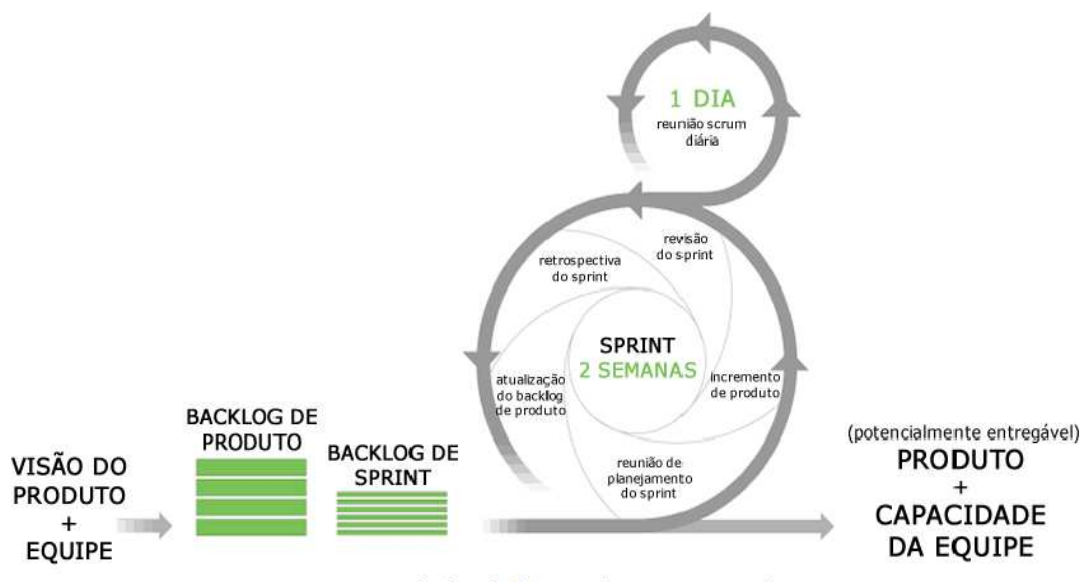


Figura 16 - Ciclo de vida do SCRUM (Schwaber,1995)

As sprints são compostas por uma reunião de planejamento da Sprint, reuniões diárias, o trabalho de desenvolvimento, uma revisão da Sprint e a retrospectiva da Sprint (SOARES, 2015). Durante a sprint: não são feitas mudanças que podem afetar o objetivo da Sprint; a composição da equipe de desenvolvimento permanece constante; as metas de qualidade não diminuem.

As funcionalidades a serem implementadas são mantidas em uma lista, chamada backlog do produto. No início de cada sprint, é feita uma reunião de planejamento onde é identificado seu escopo. A cada início de dia, o time de desenvolvimento realiza uma reunião breve, denominada reunião diária, para discutir o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia. Ao final do sprint, o time apresenta as funcionalidades implementadas em uma reunião de revisão e, por fim, há uma reunião de retrospectiva, para identificar o que funcionou bem e o que pode ser melhorado (COSTINHAS, 2014).

O desenvolvimento enxuto de software (do inglês, *lean software development*) (POPPENDIECK, 2003) é baseado nos conceitos de produção enxuta (cujo foco é a eliminação de desperdícios do processo de produção) e desenvolvimento enxuto de produtos. O desenvolvimento enxuto de produtos tem foco nas etapas de criação do produto, do conceito à entrega final, sendo mais próximo da engenharia de software. Os sete princípios adaptados ao desenvolvimento de software são: elimine desperdícios, estimule o aprendizado, tome decisões o mais tarde possível, entregue o produto o mais rápido possível, dê poder de decisão à equipe, inclua a qualidade no processo e otimize o todo (COSTINHAS, 2014).

Kanban (OHONO, 1997) é um termo japonês que significa sinal visual. Uma de suas grandes características é evidenciar os problemas existentes no processo. As equipes que o adotam precisam estar atentas ao processo aplicado para visualizar pontos de melhoria e adaptações, para que o processo possa fluir de forma satisfatória, o Kanban tem apenas três prescrições: visualize o fluxo de trabalho atual; limite o fluxo de trabalho; acompanhe e gerencie o fluxo de trabalho (COSTINHAS, 2014).

A programação Extrema (eXtreme Programming – XP) (BECK, 2004), é o método ágil mais conhecido e enfatiza a colaboração, criação rápida e precoce do software (LARMAN, 2003). Segundo Beck e Andres (2004), o XP é um método leve, que procura fazer o necessário para trazer valor para o cliente por meio de software funcionando. O

método XP adapta-se a requisitos vagos ou a constante mudança e também funciona em times de qualquer tamanho (BECK, 2004). XP (Extreme Programming) é uma metodologia iterativa e incremental, onde cada ciclo leva em média de 3 a 4 semanas. Possui um conjunto de práticas para desenvolvimento de software utilizando refatoramento, programação em pares, propriedade coletiva, ambiente único com desenvolvedores e clientes, elaboração de histórias e desenvolvimento dirigido por testes (*Test Driven Development*, ou *TDD*) (Leal, 2009). Parte importante do processo de desenvolvimento de software, o processo de qualidade envolvido requer planejamento e atenção, visto que garantir a qualidade do software desenvolvido é tão relevante quanto a sua codificação. Assim como a qualidade do produto deve ser alcançada em um meio de desenvolvimento clássico, ela também precisa ser levada em consideração nos ambientes ágeis (COLLINS, 2010).

2.5. Testes de Software em Projetos Ágeis

Define-se como teste de software o processo de execução de um programa com a finalidade de encontrar erros, no qual se aumenta a confiabilidade do software encontrando erros e corrigindo-os (BORTOLUCI, 2015). Segundo o TMMI Foundation (TMMI, 2012), as atividades de teste de software geralmente são responsáveis por 30% a 40% dos custos de um projeto. Os sistemas de informação estão se tornando cada vez mais complexos e, em função disso, a área de teste de software também está evoluindo de forma a buscar a qualidade de toda essa estrutura. O teste de software pode ser considerado como uma fase do processo de desenvolvimento de software, cujo objetivo é atingir um nível de qualidade de produto superior. Os testes antecipam a descoberta de falhas e incompatibilidades, reduzindo o custo do projeto.

Segundo Martins (2007), o teste de software faz parte de um aspecto mais amplo da garantia de qualidade no desenvolvimento de software, frequentemente referido como verificação e validação. A verificação busca avaliar se as funções do software foram implementadas corretamente, enquanto que a validação avalia se os requisitos informados pelo cliente foram implementados consistentemente no software.

Em geral as equipes de testes utilizam dos requisitos de negócios em conjunto com outros artefatos técnicos para definir os cenários e condições de testes necessários para verificação e validação do sistema. Em projetos que utilizam metodologia de desenvolvimento tradicional a criação dos casos de testes ocorre em paralelo ao

desenvolvimento, e a execução após a entrega do código para o ambiente de testes. O teste bem-sucedido é aquele que consegue determinar casos de teste para os quais o programa que está sendo testado falhe (VICENTE, 2010).

Em projetos tradicionais de desenvolvimento de software a atividade de teste ocorre basicamente por meio de duas estratégias: o sistema é testado somente quando ele está totalmente construído na esperança de encontrar erros ou utiliza-se a estratégia de teste incremental. A estratégia de teste incremental ocorre da seguinte forma (Vicente, 2010):

- *Teste de Unidade*: tem como foco as menores unidades de um programa, que podem ser funções, procedimentos, métodos ou classes. O objetivo desta fase é revelar defeitos de lógica e implementação em cada unidade, fornecendo evidências de que a unidade testada funciona adequadamente de forma isolada. Como cada unidade é testada separadamente, o teste de unidade pode ser aplicado à medida que ocorre a implementação das unidades e pelo próprio desenvolvedor, sem a necessidade de dispor-se do sistema totalmente finalizado.
- *Teste de Integração*: após serem testadas as unidades individualmente, o teste de integração deve ser realizado, sendo que a ênfase é dada na construção da estrutura do sistema. Nesta fase, as diversas partes do software são integradas e executadas, verificando se a integração entre elas funciona de maneira adequada e não leva a erros.
- *Teste de Sistema*: é iniciado após se ter o sistema completo, com todas as suas partes integradas. O objetivo é verificar se as funcionalidades especificadas nos documentos de requisitos estão todas implementadas, combinando-se todos os elementos do sistema.
- *Teste de regressão*: é realizado durante a manutenção do software. A cada modificação efetuado no sistema, seja no código, ambiente ou componente externo, após a sua liberação, corre-se o risco de que novos defeitos sejam introduzidos.

Diferentemente de métodos tradicionais nas quais as atividades de teste ocorrem mais tarde no processo de desenvolvimento, os testes ágeis devem ocorrer de forma frequente, procurando detectar defeitos o mais cedo possível em ciclos de desenvolvimento iterativos e curtos, com um constante feedback do cliente. O teste de regressão necessita essencialmente de ferramentas e de casos de testes automatizados.

Em projetos ágeis também há uma preocupação em testar sob o ponto de vista do cliente por meio de testes de aceitação. Para criação de testes de aceitação, há um esforço colaborativo entre um especialista do negócio ou domínio e o desenvolvedor, analista ou qualquer outro membro do time de desenvolvimento. Cada história do cliente deve ter um ou mais casos de teste de aceitação associados, que idealmente são baseados em exemplos fornecidos pelo cliente e além de ajudar na compreensão das histórias, validam o software desenvolvido (VICENTE, 2010). Em processos de desenvolvimento iterativos e incrementais, os testes de integração serão executados continuamente, sendo o plano de testes de integração particularmente importante nesses ambientes (ABRANTES, 2012).

Um aspecto enfatizado em projetos ágeis é abordagem do time completo (prática introduzida pela XP), no qual toda a equipe é responsável pela qualidade, e os testadores devem ser integrados à equipe. Neste sentido, é desejável que todos os integrantes da equipe reconheçam a importância dos testes, desde o nível de unidade a níveis mais altos (VICENTE, 2010).

Em vez de se testar a qualidade depois de um produto ter sido construído, ela deve ser incorporada ao processo e ao produto à medida que ele é desenvolvido (COHN, 2011). A qualidade de software possui diversas dimensões, cada uma requer uma diferente abordagem de testes. Neste contexto, pode-se apresentar uma divisão das atividades de testes em métodos ágeis em quatro categorias de testes (CRISPIN, 2009): (1) TDD (unidade e integração), (2) testes de negócio (*business/customer-facing tests*), (3) testes de negócio em relação ao produto desenvolvido (testes exploratórios, usabilidade, cenário, entre outros) e (4) testes relacionados à (desempenho, carga e segurança).

Exemplos de práticas ágeis nas quais os testes são parte fundamental são (Abrantes, 2012): (i) executar testes ao longo do ciclo de vida e (ii) desenvolvimento orientado a testes (TDD, ou *test-driven development*). Outras relacionadas à qualidade do código fonte incluem: (iii) refatoração; e (iv) código compartilhado.

A prática de teste ao longo do ciclo de vida parte da consideração de que com rígidas restrições de tempo, as atividades de teste tendem a ser negligenciadas se deixadas para o final do projeto. Com isso, componentes de software devem ser testados pelos desenvolvedores e usuários à medida em que são desenvolvidos. O teste deve ser incremental, e os testes de regressão devem ser particularmente enfatizados por causa do estilo evolucionário de desenvolvimento (ABRANTES, 2012).

A prática desenvolvimento orientado a testes recomenda que o desenvolvimento de software deve ser guiado por testes. O teste sempre vem primeiro, eliminando projetos complicados além do normal e possibilitando focar de fato na obtenção do trabalho feito. Testes de unidade são implementados antes do código e são executados continuamente e livre de falhas para que o desenvolvimento possa continuar (VICENTE, 2010). O ciclo do TDD segundo (MEYER, 2014) é: (i) Adicione um novo caso de teste; (ii) Execute os testes e confirme que o novo falhará; (iii) Faça a alteração no código; (iv) Execute todos os testes e confirme que agora todos passarão; (v) Refatorar para remover duplicações no código. Com relação à objeção de com que TDD escrever um teste primeiro deve demorar mais, há provas de que TDD realmente leva aproximadamente 15% há mais, mas em compensação há provas de que leva a menos defeitos (COHN, 2014).

A Figura 17 mostra uma comparação do modelo de desenvolvimento de testes tradicional e do TDD.

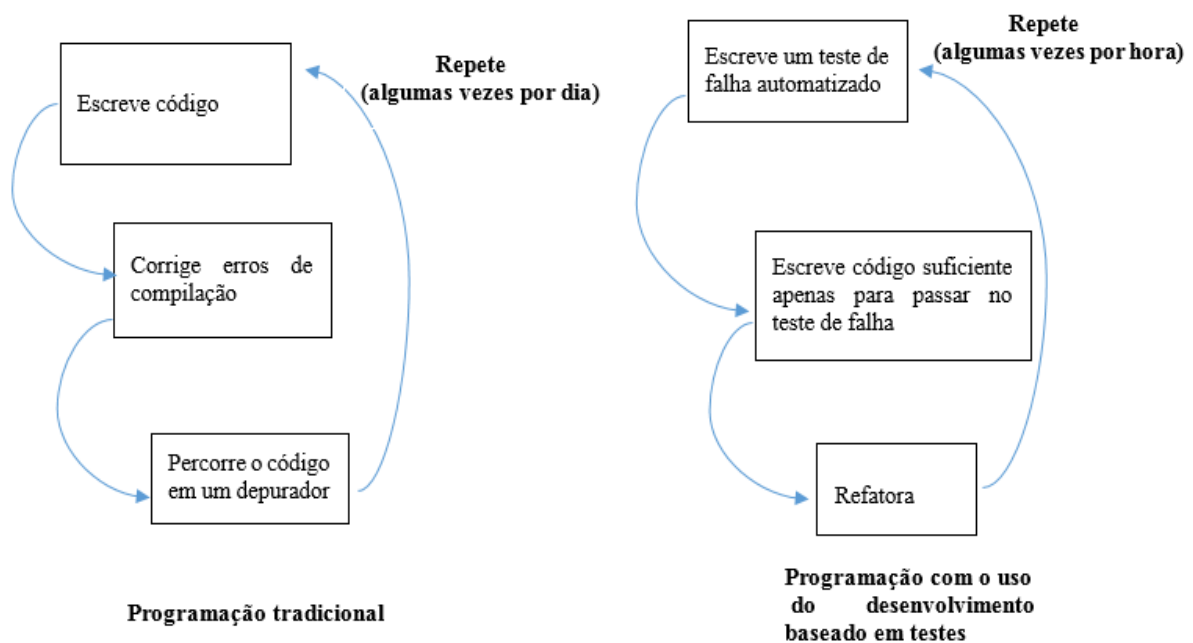


Figura 17 - Ciclo desenvolvimento tradicional e baseado em testes. Adaptado de (Cohn, 2014)

Com a prática refatoração, os programadores reestruturam o software sem modificar seu comportamento, para remover duplicações, melhorar comunicação, simplificar ou adicionar flexibilidade; todos os testes são mantidos rodando. Testes de unidade automatizados garantem que nada se quebra, permitindo aos programadores refatorar com confiança. Os programadores devem sempre deixar o código mais limpo do que o que eles

encontraram, melhorar sua performance e sua capacidade de reação rápida a mudanças (ABRANTES, 2012).

Ao adotar a prática código compartilhado, o código e seus testes associado são verificados e incluídos na base de código, que pode ser alterado por qualquer membro da equipe. Qualquer programador pode melhorar qualquer código em qualquer parte do software a qualquer tempo se ele vê esta oportunidade. Esta posse coletiva de código traz para cada membro da equipe o sentimento de ser o dono de toda a base de código e previne gargalos que poderiam ocorrer se o “dono” de um componente não estivesse disponível para realizar uma modificação necessária (ABRANTES, 2012).

Diferentes métodos ágeis, têm diferentes estratégias em relação à execução dos testes.

- No XP, todo o trecho de código tem um conjunto de casos de teste de unidade automatizados, entregue dentro do repositório de código-fonte. Os resultados dos testes servem como uma forma de feedback instantâneo, no qual o desenvolvedor pode detectar rapidamente se o método desenvolvido ainda precisa ser modificado ou refatorado (VICENTE, 2010).
- No método SCRUM, os testes são executados como tarefas em Sprints de desenvolvimento (incrementos do software), e pode ter como objetivo exclusivo a execução de tarefas relacionadas a testes, melhoria do código e correção de defeitos (VICENTE, 2010).
- No Lean, além da atividade de teste, são executadas inspeções de projeto e de código (LIMA 2013).
- O FDD (*Feature Driven Development*) (ABRAHAMSSON *et al.*, 2002) ou Desenvolvimento guiado por funcionalidade, é um processo de software de desenvolvimento iterativo e incremental. Assim como no Lean, são executadas inspeções de projeto e de código. O processo combina o desenvolvimento guiado por modelos (*model-driven*) e o desenvolvimento ágil, com ênfase em um modelo de objetos inicial, divisão do trabalho em características, e desenvolvimento iterativo para cada funcionalidade (LIMA, 2013).
- ASD (*Adaptative Software Development*), (ABRAHAMSSON *et al.*, 2002) ou Desenvolvimento Adaptativo de Software, encoraja o desenvolvimento iterativo e incremental, com constantes prototipações e a tentativa de promover um

paradigma adaptativo de software. Assim como no Lean, são executadas inspeções de projeto e de código (LIMA, 2013). Os testes fazem parte da construção concorrente dos componentes, com inspeção de código e revisões para melhorar a qualidade do produto (SOARES, 2015).

2.6. Trabalhos relacionados

Há poucas referências disponíveis referentes a medidas e/ou indicadores utilizados em projetos ágeis e, em especial, a medidas e/ou indicadores referentes a testes em projetos ágeis. No primeiro grupo, pode-se destacar o trabalho de Kupiainen *et al.* (2015) e, no segundo grupo, o trabalho de Mota e Dias-Neto (2014).

Kupiainen *et al.* (2015) apresentam uma revisão sistemática da literatura em estudos executados na indústria *Lean* sobre quais medidas são utilizadas em projetos de desenvolvimento de software que utilizaram metodologias ágil (*o que representa as questões de pesquisa principal dos autores - QP1*), o trabalho também busca entender as razões e efeitos do uso de métricas na indústria de desenvolvimento de software (QP2), e também quais métricas tiveram alta influência em projetos de desenvolvimento que utilizaram métodos ágeis (Q3).

Como resultado deste estudo 102 medidas foram identificadas em 30 diferentes estudos primários, respondendo assim à QP1 (*o que representa as questões de pesquisa principal dos autores*).

A Tabela 4 apresenta uma lista de medidas citadas na literatura ágil classificada por métodos ágil utilizando (Scrum, XP, Lean, Kanban) (KUPIAINEN *et al.*, 2015). Estão destacadas em negrito as medidas referentes a atividade de testes.

Tabela 4 - Lista de medidas de testes ágeis. Fonte: (Kupiainen et al., 2015)

Medida sugerida	Método
Esforço de estimativa	Scrum, XP, LeanSD
Velocidade	Scrum, XP, LeanSD
Casos de testes criados e passados	XP, LeanSD
Tempo atual de desenvolvimento	XP
Fator de carga	XP
Trabalho em progresso	Kanban
Tempo de espera	Kanban
Data de desempenho	Kanban
Fluxo de Tarefas	Kanban
Itens de trabalhos registrados e bloqueados	Kanban
Eficiência do fluxo	Kanban

Medida sugerida	Método
Qualidade Inicial	Kanban, LeanSD
Falha de carga	Kanban
Tempo do ciclo	LeanSD
Mapas de fluxo de valor	LeanSD
Número de Casos de testes criados e passados nos testes de aceitação por iteração	LeanSD

A Tabela 5 apresenta o conjunto de medidas identificadas pelos autores como resultado da revisão sistemática conduzida (KUPIAINEN *et al.*, 2015). As medidas em negrito foram as mapeadas como medidas de testes como contribuição do mapeamento sistemático deste trabalho e detalhado no Apêndice I.

Tabela 5 - Medidas ágeis. Fonte: (Kupiainen et al., 2015)

Medidas Utilizadas em Projetos Ágeis
Valor de negócio entregue, satisfação do cliente, total de defeitos após testes, quantidade de casos de teste, Quantidade de funcionalidade que passam nos testes de aceitação.
Velocidade, Progresso do trabalho.
Defeitos críticos enviados pelo cliente, defeitos abertos, taxa de teste falhados, Percentual de testes executados com sucesso, esforço para as tarefas restantes, efetividade da equipe.
Quadro de débito técnico, status da construção, esforço do débito técnico.
Burndown, check-ins por dia, quantidade de passos automatizados, falhas per iteração.
Velocidade, Estimativa de histórias.
Burndown, pontos de histórias, Quantidade de defeitos aberto, Quantidade de defeitos descobertos no teste de sistema, defeitos adiados, Net Promoter Score.
Pontos de história, esforço das tarefas, velocidade, operações de velocidade
Esforço estimado.
Quantidade de defeitos /velocidade.
Receita per cliente.
Data final esperada para a tarefas, Esforço estimado, Páginas web finalizadas, Tarefas finalizadas.
Tempo de correção para construções falhadas, percentual de fluxo de histórias, percentual de histórias preparadas para sprint, velocidade na elaboração das funcionalidades, velocidade na implementação de funcionalidades.
Status das construções, cobertura de teste, taxa de crescimento de teste, violações de análise estática de código, quantidade de testes unitários.
Esforço estimado.
Sprint burndown, release burndown, índice de desempenho de custo, índice de desempenho do prazo, velocidade planejada.
Ritmo de tempo comum, número de retornos, Tempo do ciclo, progresso do trabalho, satisfação do cliente (Kano analysis), estimativa de esforço.
Tempo de espera, tempo de processamento, tempo na fila.
Solicitação de mudança por requisito, falhas, requisitos implementados x desperdício, esforço de manutenção, Tempo de espera.
Número de solicitação dos clientes, inventário de requisitos atrasados.
Taxa de requisitos por fase, variação nas transferências, tipo de custo por requisitos.
Quantidade de requisitos por fase, tempo de espera.
Velocidade média / progresso do trabalho, Tempo do ciclo, pseudo velocidade.
Tempo de espera, progresso do trabalho
Indicador de tendência de defeitos, Quantidade de defeitos no backlog, Quantidade de defeitos previstos.
Throughput, fila.

Medidas Utilizadas em Projetos Ágeis
Burndown, check-ins por dia, número de passos automatizados , Quantidade de defeitos novos e abertos.
Burndown, número de passos automatizados , check-ins por dia.
Estimativa de história, percentual de histórias finalizadas
Progresso do código em trabalho.

As razões e efeitos levantados pela pesquisa foram categorizadas em Sprint e Planejamento de projeto, Sprint e Acompanhamento de progresso de projeto, Entendimento e melhorias de qualidade, Reparo de problemas no processo de software e Motivação das pessoas, o que responde a QP2. As respostas para a QP3 foram categorizadas em 3 níveis (alta, média e baixa) para avaliar o nível de influência das medidas.

Como conclusão os autores afirmam que a literatura ágil propõe a utilização de métricas, mas que poucas são conhecidas e divulgadas, e que muitas medidas são customizadas para atender à demanda dos projetos. Um viés mapeado pelos pesquisadores foi que 22% dos artigos mapeados foram publicações de uma única organização sueca do ramo de telecomunicações e 28% no total vieram da mesma indústria de telecomunicações, o que pode significar um risco para os resultados apresentados. Além disso, o trabalho não deixa claro a duração dos projetos ou iterações e nem quais medidas foram de fato coletadas, o que deixa vago quanto a real utilização de todas as medidas, tendo em vista que muitas medidas possuem nomes pouco mencionados em outros trabalhos.

O mapeamento sistemático realizado por Mota e Dias-Neto (2014) teve por objetivo principal identificar medidas aplicadas ao processo de teste em projetos de desenvolvimento ágil. O trabalho também mapeou e classificou as medidas com base em objetivo e características das mesmas. Este trabalho que foi finalizado em março de 2014 apresentou como resultado, 16 artigos no mapeamento e 24 medidas foram identificadas, como pode ser visto na Tabela 6.

Neste trabalho não foi detalhado formas de coletas das medidas, nem análise das mesmas, e também como citado pelos autores as medidas mapeadas não foram revisadas por outros pesquisados e profissionais da área de testes que atuem em projetos com desenvolvimento ágil, criando assim um ponto importante de ameaça à validade das medidas mapeadas.

Tabela 6 - Lista Medidas. Fonte (Mota e Dias-Neto, 2014)

SEQ	NOME MEDIDA
01	Total de defeitos encontrados nos testes
02	Tempo de retorno do defeito
03	Tempo gasto para correção do defeito
04	Tempo planejado para correção do defeito
05	Total de defeitos resolvidos
06	Porcentagem de tempo gasto na correção de defeitos
07	Densidade de falhas
08	Número de casos de teste automatizados
09	Número de casos de testes manuais
10	Número de testes de aceitação por estória
11	Tempo para corrigir os testes
12	Tempo de conclusão do projeto
13	Número de casos de teste concluídos no prazo correto
14	Pontos de estória concluídos por iteração
15	Planejamento do tamanho do backlog
16	Estórias de usuários não verificadas
17	Quantidade de trabalho restante (em horas) para cada tarefa
18	Quantidade de trabalho gasto (em horas) para cada tarefa
19	Número de casos de teste executados
20	Eficiência na remoção de defeitos
21	Estimativa de estórias de usuário (tempo)
22	Precisão de estimativa de teste
23	Percentual de estórias aprovadas
24	Percentual de cobertura de teste

Este trabalho foi utilizado como base para esta pesquisa pelo fato de ser o único identificado com foco em medidas de teste no desenvolvimento de software ágil. Porém para minimizar os pontos de ameaça, e por não termos acesso ao protocolo do mapeamento, todos os artigos citados no mapeamento de Mota e Dias-Neto (2014) foram analisados com leitura completa e re-analisados. Como resultado, dos 16 artigos citados por (Mota e Dias-Neto, 2014), 3 foram descartados por não atenderem aos critérios de inclusão definidos no mapeamento sistemático (a ser descrito no Capítulo 3 e apresentado em detalhes no Apêndice I). Desta forma 15 medidas anteriormente mapeadas foram descartadas, ou por não estar claro nos artigos que eram medidas relacionadas a teste, ou por não ser possível afirmar que foram utilizadas em projetos ágeis. Ainda como parte desta análise, 13 novas medidas foram mapeadas, medidas que não haviam sido consideradas anteriormente no mapeamento original. Com isso, o mapeamento realizado por realizado por Mota e Dias-Neto (2014) contribuiu, após replicação e validação do mapeamento conforme explicado no Capítulo 3, com 22 medidas do chamada de Ciclo 1 do mapeamento executado pelo autor dessa dissertação.

2.7. Considerações Finais

Este capítulo apresentou uma revisão da literatura sobre medição, plano de medição e apresentou o template ASM.br que será usado para detalhamento dos indicadores desta dissertação. Também apresentou alguns dos métodos ágeis mais utilizados na indústria, assim como os tipos de teste e testes ágeis. Por fim, foram apresentados trabalhos que pesquisaram referências sobre medidas em projetos de desenvolvimento ágil e medidas de testes em projetos ágeis.

O trabalho de Mota e Dia-Neto (2014) foi utilizado como referência para esta dissertação pela abordagem inicial de medidas de teste no contexto ágil, acrescido da proposta da criação de indicadores de testes que apoiem a gestão de teste nos projetos e a apresentação dos mesmos em forma de uma ficha única de uma página, o que pode facilitar a utilização no dia-a-dia dos projetos.

3. Indicadores de Testes em Projetos Ágeis

Este capítulo descreve o objetivo e questão de pesquisa na Seção 3.1, na Seção 3.2 aborda o planejamento do estudo do mapeamento sistemático sobre medidas de testes em projetos ágeis, na Seção 3.3 é abordado como as medidas foram identificadas no mapeamento sistemático, na Seção 3.4 aborda a definição dos objetivos e informação necessária para criação das fichas de registro de indicadores, os processos do MPS.BR utilizados são citados na Seção 3.5, na Seção 3.6 é detalhado a seleção final de indicadores criados a partir das medidas pesquisadas, na Seção 3.7 os mapas de indicadores gerados como parte do conjunto de fichas, na Seção 3.8 é explicado como as planilhas de coleta e medição foram definidas, o capítulo fecha na Seção 3.9 com as considerações finais.

3.1. Introdução

Uma das formas de alcançar a visibilidade do andamento ou dos resultados de um projeto de software é por meio da medição de software, que pode ser definida como uma avaliação quantitativa de qualquer aspecto dos processos e produtos da Engenharia de Software (BASS, 1999). Segundo Vicente (2010), uma das práticas que apoia a melhoria contínua de processos de software é a coleta de dados sobre o andamento do projeto por meio de medidas de software. A medição permite o melhor entendimento do que se produz e de como é produzido, com isso auxilia o planejamento, controle e melhoria dos projetos (BONELLI, 2014).

Medidas podem ser utilizadas para indicar o alcance a objetivos. Quando isso ocorre, a medida desempenha o papel de indicador (BARCELLOS, 2009). Segundo McGarry *et al.* (2012) um indicador é uma medida ou uma combinação de medidas, normalmente apresentado na forma de gráfico ou Tabela, que provê entendimento a respeito de uma questão ou conceito de software.

Nesse contexto, o objetivo desta dissertação é descrever uma lista indicadores que apoiem as organizações nas fases de teste em projetos de desenvolvimento que utilizem metodologia ágil com informações detalhadas que possibilitem a coleta das medidas, geração e análise dos indicadores. É esperado que os líderes de teste e os gerentes de projeto utilizem estas fichas como apoio a gestão do ciclo de testes e que também ajude na análise para possíveis problemas.

Conforme mencionado na revisão da literatura do capítulo 2, existem poucos trabalhos de pesquisa que citam ou sugerem medidas ou indicadores que apoiem a atividade de teste para projetos ágeis. Quando citam, na sua maioria são indicadores relacionados a outras atividades de gerenciamento de projeto de desenvolvimento ou citam apenas as medidas relacionadas a teste, sem detalhamento de coleta e análise de indicadores.

O fato das medidas serem citadas na literatura por si só não garante apoio no gerenciamento dos testes. Desta forma, esta pesquisa espera contribuir com uma lista de indicadores através da realização de uma pesquisa baseada em medidas identificadas na literatura.

Uma das primeiras atividades da dissertação foi realizar um mapeamento sistemático da literatura para investigar e identificar medidas e indicadores relacionados a testes em projetos ágeis de desenvolvimento de software.

Para delinear esta primeira atividade, a seguinte questão de pesquisa foi formulada:

(QP1) “Que indicadores relacionados a teste podem ser utilizados durante o desenvolvimento ágil de software?”

Para detalhar e descrever os indicadores definidos nesta pesquisa foi utilizado o template ASM.br (BONELLI, 2014; BONELLI *et al.*, 2017). A avaliação do conteúdo das fichas com os indicadores definidos nesta dissertação foi realizado através de um estudo de caso, que será descrito no capítulo 4.

3.2. Planejamento do Estudo

O planejamento deste estudo foi baseado nas diretrizes propostas por KITCHENHAM e CHARTERS (2007) e PETERSEN *et al.* (2015) para um mapeamento sistemático. Dessa forma, foi definido um protocolo baseado em mapeamento sistemático com o objetivo de pesquisar, avaliar e interpretar artigos relevantes para as questões de pesquisa.

Seguindo o paradigma Goal Question Metric (BASILI e ROMBACH, 1988), este estudo pode ser definido da seguinte forma: **analisar** relatos de experiência e publicações científicas sobre projetos ágeis de desenvolvimento **com o propósito de** identificar medidas e indicadores **com relação às** atividades relacionadas a testes **do ponto de vista de** pesquisadores **no contexto** acadêmico e industrial

Desta forma foram definidas duas questões de pesquisa:

- **QP1:** Que medidas relacionadas a teste durante o desenvolvimento ágil de software são citadas na literatura?
- **QP2:** Que indicadores relacionados a teste durante o desenvolvimento ágil de software podem ser derivados das medidas citadas na literatura?

No âmbito deste estudo foi adotado o seguinte escopo: (1) possuir um mecanismo de busca que permita o uso de expressões lógicas ou funcionalidade equivalente; (2) pertencer a uma das editoras listadas no Portal de Periódicos da CAPES; (3) incluir em sua base publicações da área de exatas ou correlatas que possuam relação direta com o tema a ser pesquisado; e (4) possuir mecanismos de busca que permitam a busca no texto completo das publicações.

A pesquisa está restrita à análise de publicações obtidas exclusivamente a partir das fontes selecionadas, a partir dos critérios supracitados. O estudo engloba os dados disponíveis nas fontes até dezembro de 2016. Os dados de 2004 até 2013 são derivados do trabalho de Mota e Dias-Neto (2014). Os dados de 2014 a 2016 são derivados da execução realizada exclusivamente no decorrer dessa dissertação. Importante notar que os trabalhos citados no mapeamento conduzido por Mota e Dias-Neto (2014) foram reavaliados e catalogados de acordo com os procedimentos descritos nesse protocolo.

O mapeamento sistemático de Mota e Dias-Neto (2014) foi utilizado como uma das referências desta pesquisa pelo fato deste trabalho ter uma proposta parcialmente semelhante à pesquisa atual. Neste trabalho que foi finalizado em março de 2014, os pesquisadores mapearam e caracterizaram medidas de testes utilizadas em projetos desenvolvidos com métodos ágeis aplicadas no contexto de teste de software.

As bases utilizadas para pesquisa foram Scopus e IEEE, e utilizaram a seguinte string de busca de artigos:

TITLE-ABS-KEY (("agile" OR "agile method" OR "agile methods" OR "agile methodology" OR "agile methodologies" OR "agile development" OR "agile process" OR "agile processes" OR "agile practice" OR "agile practices" OR "agile project" OR "agile projects" OR "agile lifecycle" OR "SCRUM" OR "extreme programming" OR "lean development" OR "feature driven development" OR "dynamic system development" OR "agile unified process") AND ("metric" OR "metrics" OR "measure" OR "measures" OR "measured" OR "measurement" OR "measurements" OR "indicator" OR "indicators" OR "evaluation") AND ("software project" OR "software projects" OR "software process" OR "software processes" OR "software product" OR "software products" OR "software production")) AND (LIMIT-TO(SUBJAREA,"COMP"))

Foram definidos os seguintes critérios de inclusão durante análise dos artigos retornados durante o mapeamento sistemático.

- **CI1:** Só considerar publicações que objetivam propor ou descrever a identificação de **medidas em projetos ágeis de desenvolvimento de software.**
- **CI2:** considerar publicações que objetivam propor ou descrever a identificação de **medidas de teste em projetos ágeis de desenvolvimento de software.**
- **CI3:** considerar publicações que objetivam propor ou descrever a identificação de **indicadores de teste em projetos ágeis de desenvolvimento de software.**

Nesta etapa foi ainda utilizado os seguintes critérios de exclusão:

- **CE1:** não considerar publicações que descrevem projetos de desenvolvimento de software, mas cujo o foco é apenas **em análise de desempenho de código ou gerenciamento de projetos ou não citam medidas ou indicadores de testes de software.**
- **CE2:** não considerar publicações que descrevem o conjunto de resumo de artigos de **conferências, capítulos de livros e artigos de seminários (proceedings).**
- **CE3:** não considerar publicações que descrevem a identificação de medidas e indicadores de **projetos não-ágeis de desenvolvimento de software.**

Durante a execução do mapeamento sistemático foram armazenadas informações sobre as publicações como:

- o Título; Autor(es)
- o Ano da publicação
- o Referência completa;
- o Resumo da publicação;

- o Medidas citadas no artigo (independentemente de ser de teste);
- o Contexto em que a medida é (ou pode ser) utilizada em projetos;
- o Exemplo e utilização;
- o Indícios de que as medidas podem ser utilizadas em outros projetos ágeis;
- o Evidência de uso efetivo;
- o Medidas relacionadas a testes;
- o Fórmulas para cálculo das medidas de teste.

As informações coletadas foram descritas no apêndice I, Seção I.8.

A análise dos dados é feita tanto de forma quantitativa como qualitativamente.

A análise quantitativa é realizada por meio da extração direta dos dados a partir do banco de dados com os registros dos itens retornados pela máquina de busca. Esta análise fornece o número de publicações selecionadas para fazerem parte do estudo.

A análise qualitativa deve utilizar como base os dados quantitativos e realizar considerações com o intuito de discutir os resultados da busca com relação às questões de pesquisa declaradas. Podemos citar as fases de testes que se pode coletar a medida, fase de planejamento ou execução, assim como classificar se a medida é: objetiva, direta, indireta, acompanhamento, desempenho, cronograma e progresso e qualidade do produto.

Não foi realizado teste do protocolo uma vez que esta pesquisa é uma replicação de um mapeamento sistemático. No entanto, repetiu-se a execução da string de busca e aplicação dos critérios de inclusão e exclusão no mapeamento original para garantir que os resultados relatados em (Mota e Dias-Neto, 2014) fossem de fato replicados nesta pesquisa. Considerou-se a execução do Ciclo 1 como sendo o teste do protocolo já que o trabalho original de (Mota e Dias-Neto, 2014) era vago em relação aos procedimentos adotados.

Dessa forma, foram definidos como artigos de controle aqueles citados em (Mota e Dias-Neto, 2014), apresentados na Tabela 7. O ID do artigo foi criado para ajudar a identificar o artigo ao longo da pesquisa.

A replicação do mapeamento sistemático de (Mota e Dias-Neto, 2014) foi chamado de 1º ciclo de execução ou ciclo 1. Foram lidos todos os artigos do mapeamento sistemático original, aplicou-se os critérios de inclusão e exclusão definidos neste trabalho e também foi realizada a extração dos dados, seguindo o exemplo de formulário apresentado na Tabela 8. Neste ciclo houve 15 medidas listadas no mapeamento sistemático de (Mota e Dias-Neto, 2014) que não foram identificadas durante a replicação e revisão dos artigos, sendo assim

foram descartadas. Em contrapartida 13 medidas que não faziam parte da lista original, foram mapeadas durante a replicação e revisão e assim adicionadas a lista de medidas do ciclo 1, fazendo um total de 22 medidas como resultado final do ciclo 1 do mapeamento. Esta análise da replicação foi auditada por outro pesquisador que validou as medidas excluídas e as novas mapeadas.

Tabela 7 - Artigos de controle citados em (Mota e Dias-Neto, 2014)

ID	Título	Autor(es)	Ano
P01_01	Analyses of an agile methodology Implementation	Ilieva, S., Ivanov, P. and Stefanova, E.	2004
P01_02	Embedded Agile Project by the Numbers with Newbies	Schoonderwoert, N. V.	2006
P01_03	AM-QuICk: A Measurement-Based Framework for Agile Methods Customisation	Ayed, H., Habra, N. and Vanderose, B.	2013
P01_04	Measuring the Impact of Scrum on Product Development at Adobe Systems	Green, P.	2011
P01_05	Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A.	2011
P01_06	Metrics Functions for Kanban Guards	Heidenberg, J. and Porres, I.	2010
P01_07	A development of the effectiveness evaluation model for agile software development using the balanced scorecard	Atiwithayaporn, S. and Rivepiboon, W.	2013
P01_08	A metamodel-based approach for customizing and assessing agile methods	Ayed, H., Vanderose, B. and Habra, N.	2012
P01_09	Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process	Mahnic, V. and Vrana, I.	2007
P01_11	Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery	Brown A., Ambler S. and Royce W.	2013
P01_13	Refactoring the development process: experiences with the incremental adoption of agile practices	Hodgetts, P.	2004
P01_14	Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects	Racheva, Z., Daneva, M. and Buglione, L.	2008
P01_16	Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?	Ktata, O. and Levesque, G.	2010

Na parte chamada de ciclo 2, foi realizado uma extensão do mapeamento para o período de 2014 até 2016, uma vez que o mapeamento original de (Mota e Dias-Neto, 2014) foi finalizado em março de 2014. Neste ciclo foi executada a mesma string de busca utilizada no ciclo 1 e nas mesmas bases de pesquisa. Foram retornadas 139 publicações a partir do ano de 2014, todas as publicações tiveram seus títulos e resumos lidos e os critérios de inclusão considerados. Foram selecionados 22 artigos para leitura completa após aplicação dos critérios de inclusão e exclusão já mencionados. Como resultado, após leitura completa,

9 artigos foram considerados válidos e nestes artigos 58 medidas de teste foram mapeadas. Após exclusão de medidas duplicadas ou já mapeadas no ciclo 1, restaram 17 medidas. Estas 17 medidas foram consideradas novas, pois não haviam sido mapeadas antes no ciclo 1. A Figura 18 mostra graficamente a evolução do ciclo 2 do mapeamento.

Tabela 8 – Exemplo de formulário de catalogação das publicações

Dados da publicação	
Título:[2]	Embedded Agile Project by the Numbers with Newbies
Autor (es):	Schooenderwoert, N. V.
Ano da publicação:	2006
Referência completa:	Schooenderwoert, N. V. (2006) “Embedded Agile Project by the Numbers with Newbies”, Agile 2006 conference report.
Resumo da publicação	
Este artigo apresenta métricas detalhadas de um projeto de desenvolvimento de software com metodologia ágil onde uma equipe de desenvolvimento recém-formada desenvolveu um novo produto de software desde o início. O produto desenvolvido foi uma ferramenta para dispositivo móveis para uma indústria de produtos agrícolas para monitoramento de grãos. O artigo cita como a equipe de desenvolvimento era inexperiente no início do projeto e das dificuldades durante o desenvolvimento do produto a cada release.	
Medidas citadas no artigo (independente de ser de teste)	
Defect Rate – Taxa de Defeitos Defects to Customer – Defeitos do cliente Number of defects – Total de defeitos encontrados nos testes Defect removal efficiency – Eficiência na remoção de defeitos Lines of code – linhas de código Effort expended – Esforço gasto Size of software build – Tamanho de software construído Schedule compliance – Conformidade de cronograma Quality of the deliverable – Qualidade da entrega. Productivity– ESLOC/h (Effective source lines of code) – Linhas do código fonte efetivas.	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
Estas medidas foram coletadas pela equipe de desenvolvimento durante o projeto	
Exemplo de utilização	
N/A	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
O software citado foi desenvolvido para uma aplicação embarcada dentro de uma indústria específica. No artigo não fica claro se em outros contextos poderiam utilizar o mesmo software.	
Evidência de uso efetivo	
As medidas foram coletadas durante as fases de projeto e algumas informações com os dados são disponibilizadas.	
Medidas relacionadas a teste	
Defect Rate – Taxa de Defeitos Defects to Customer – Defeitos do cliente Number of defects – Total de defeitos encontrados nos testes Defect removal efficiency – Eficiência na remoção de defeitos Quality of the deliverable – Qualidade da entrega.	
Fórmulas para cálculo das medidas de teste	
Defect rate – Taxa de Defeitos = $DD = \text{TOT_DEF} / \text{ESLOC (effective source lines of code)} * 100$ Defects to Customer – Defeitos do cliente = $\text{Total FP} * \text{defects per FP} * (1.0 - \text{defect removal efficiency})$ Defect removal efficiency – Eficiência na remoção de defeitos = $\text{TOT_DEF REMOVED (BEFORE)} / \text{TOT_DEF} * 100$	

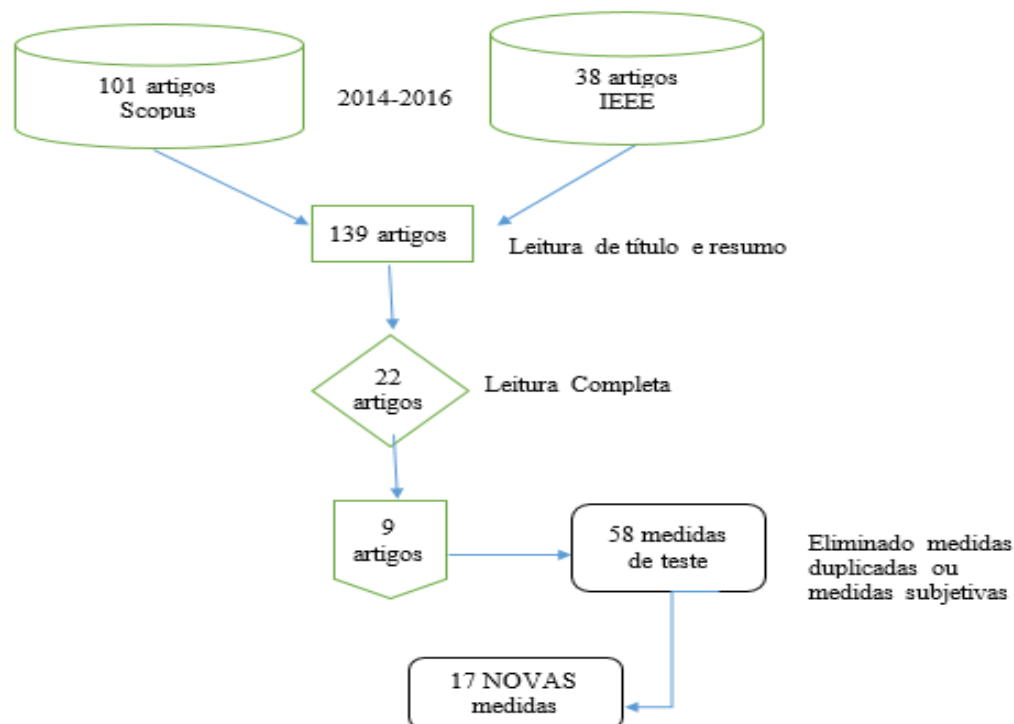


Figura 18 - Fluxo de pesquisa e leitura dos artigos no Ciclo 2 do mapeamento

Na Tabela 9 estão listados ID dos artigos, a referência completa dos artigos para os dois ciclos do mapeamento. A coluna ID é para identificar qual ciclo de mapeamento o artigo faz parte. Os artigos do Ciclo 1 tem o código iniciando com P01 e os do Ciclo 2 como P02.

Tabela 9 – artigos utilizados no mapeamento sistemático

ID	Referência Completa
P01_01	Ilieva, S., Ivanov, P. and Stefanova, E. (2004) “Analyses of an agile methodology Implementation”, Proceedings of the 30 th EUROMICRO conference. Pp. 326-333
P01_02	Schoonderwoert, N. V. (2006) “Embedded Agile Project by the Numbers with Newbies” Agile 2006 conference report.
P01_03	Ayed, H., Habra, N. and Vanderose, B. (2013) “AM-QuICk: A Measurement-Based Framework for Agile Methods Customisation”, In Proc. IWSM/Mensura 2013:71-80.
P01_04	Green, P. (2011) “Measuring the Impact of Scrum on Product Development at Adobe Systems”, In: 44 th Hawaii International Conference on System Sciences, p.1-10.
P01_05	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A., (2011) “Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs” Electronic, Robotics and Automotive Mechanics Conference (CERMA), 2011 IEEE, 421-426.
P01_06	Heidenberg, J. and Porres, I., (2010), “Metrics Functions for Kanban Guards”, In: IEEE International Conference and Workshops on Engineering of Computer-Based Systems. IEEE computer society.
P01_07	Atiwithayaporn, S. and Rivepiboon, W., (2013), “A development of the effectiveness evaluation model for agile software development using the balanced scorecard”, IMECS.
P01_08	Ayed, H., Vanderose, B. and Habra, N., (2012), “A metamodel-based approach for customizing and assessing agile methods”, In: The International Conference on the Quality of Information and Communications Technology (QUATIC).

ID	Referência Completa
P01_09	Mahnic, V. and Vrana, I., (2007), "Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process", Electrotechnical Review.
P01_11	Brown A., Ambler S. and Royce W., (2013), 'Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery', ICSE.
P01_13	Hodgetts, P., (2004), "Refactoring the development process: experiences with the incremental adoption of agile practices", IEEE Computer Society. In: Agile Development Conference.
P01_14	Racheva, Z., Daneva, M. and Buglione, L., (2008), "Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects", Proc. 34 th Euromicro Conf. Software Engineering and Advanced Applications, pp-457-464.
P01_16	Ktata, O. and Levesque, G., (2010), "Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?", Proceedings of the Third Conference on Computer Science and Software Engineering, p.101-107.
P02_01	Kayes I., Sarker M., Chakareski J., (2016), "Product backlog rating: a case study on measuring test quality in scrum", Innovation System software engineering
P02_02	Yamada S., Kii R., (2015), "Software quality analysis for agile development", 2015 4th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2015
P02_03	Kupiainen E., Mäntylä M.V., Itkonen J. (2015), "Using metrics in Agile and Lean software development - A systematic literature review of industrial studies", Information and Software Technology
P02_04	Staron M., Meding W., Palm K., (2015), "Release readiness indicator for mature agile and lean software development projects", Lecture Notes in Business Information Processing
P02_05	Staron M., Meding W., Hansson J., Höglund C., Niesel K., Bergmann V., (2014), "Dashboards for Continuous Monitoring of Quality for Software Product under Development", Relating System Quality and Software Architecture, pp-209-229.
P02_06	Willamy R., Nunesy J., Perkusichz M., Freirex A., Saraiva R., Almeida H., Perkusich A., (2016), "A method to build Bayesian networks based on artifacts and metrics to assess agile projects", Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE.
P02_07	Gupta R.K., Manikreddy P., Gv A., (2016), "Challenges in adapting agile testing in a legacy product", 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)
P02_08	Malhotra R., Chug A., (2016), "Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance", Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016
P02_09	X. Zhao; X. Xuan; A. Wang; D. Liu; L. Zheng , (2014), "Software Quality Control via Exit Criteria Methodology: An Industrial Experience Report", 2014 21st Asia-Pacific Software Engineering Conference

Desta forma, conforme detalhado na Seção 3.2, podemos afirmar que o ciclo 1 do mapeamento contribuiu com 22 medidas, e o ciclo 2 com 17 medidas, fazendo um total de 39 medidas identificadas após análise do ciclo 1 e ciclo 2 do mapeamento.

O resultado das execuções dos ciclos 1 e 2 do mapeamento permitiu responder à questão de pesquisa (QP1) **(Que medidas relacionadas a teste durante o desenvolvimento ágil de software são citadas na literatura?)** com total de 39 medidas de testes conforme Tabela 10.

Importante notar que os artigos no escopo do mapeamento, na sua grande maioria, apresentam medidas já bem definidas baseadas em execução dos testes e quantidade de

defeitos, porém o objetivo da pesquisa é com base nessa lista, propor indicadores que possam apoiar a gestão de testes no processo de garantia da qualidade dos testes e de software.

A execução e o resultado do mapeamento foram auditados por um outro pesquisador com grande experiência em dois momentos. Primeiro, foram avaliados os procedimentos de seleção dos trabalhos de acordo com o estipulado no protocolo. Durante esta revisão não foram identificadas grandes discrepâncias dos artigos selecionados nas seleções primárias e secundárias desta pesquisa no ciclo 2. Houve apenas quatro publicações onde os artigos passaram para leitura completa na segunda seleção, e o auditor achou desnecessário, porém estes mesmos artigos acabaram sendo descartados na seleção secundária durante leitura completa do artigo, com isso estes artigos não invalidam o resultado do mapeamento.

Em um segundo momento, todos os artigos no escopo do mapeamento foram lidos pelo 2º pesquisador e a catalogação de cada artigo foi revista e discutida. Como fruto dessa auditoria, chegou-se à lista final de medidas apresentada. Na Tabela 10 são listadas as medidas mapeadas como resultados do mapeamento, a classificação da medida, a atividade de coleta que representa em qual momento do ciclo de teste a medida é coletada, ID artigos onde as medidas foram citadas e o ID Med que é um sequencial criado para identificar cada medida.

Tabela 10 – Lista de medidas mapeadas após exclusão de medidas repetidas.

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED01	Avaliação do backlog do produto	Objetiva, Qualitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P02_01
MED02	Cobertura de Testes	Objetiva, Quantitativa, direta, Acompanhamento	Planejamento	P01_16 P02_03 P02_09 P02_07
MED03	Complexidade de defeitos	Objetiva, Qualitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P02_01 P02_05
MED04	Defeitos do cliente	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P01_02
MED05	Densidade de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P01_07
MED06	Densidade de falhas por linhas de código	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P01_08
MED07	Eficiência na remoção de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Desempenho do processo	Execução	P01_01 P01_02 P01_11

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED08	Esforço gasto nos testes de aceite	Objetiva, Quantitativa, direta, Acompanhamento, Cronograma e Progresso	Planejamento	P01_01
MED09	Funcionalidades testadas	Objetiva, Quantitativa, Direta, Acompanhamento, Cronograma e Progresso	Execução	P02_06
MED10	Histórias de usuário não verificadas	Objetiva, Quantitativa, Direta, Acompanhamento	Execução	P01_06
MED11	Indicador de Previsão de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Acompanhamento, Desempenho do Processo	Execução	P02_03
MED12	Indicador de tendência de defeitos	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P02_03
MED14	Número de casos de teste automatizados	Objetiva, Quantitativa, Direta, Acompanhamento, Qualidade do produto	Planejamento	P01_03 P02_03
MED15	Quantidade de defeitos a serem removidos (backlog da release)	Objetiva, Quantitativa, Direta, Acompanhamento, Cronograma e Progresso	Execução	P02_04
MED16	Número de defeitos em aberto (não resolvidos)	Objetiva, Quantitativa, Indireta, Acompanhamento, Desempenho do processo	Execução	P01_06 P02_06
MED18	Prontidão para release (Número de semanas para Release)	Objetiva, Quantitativa, Direta, Acompanhamento, Qualidade do produto	Planejamento	P02_04 P02_05
MED19	Número de testes de aceitação por história	Objetiva, Quantitativa, Direta, Acompanhamento, Qualidade do produto	Planejamento	P01_03
MED20	Onda de Impacto dos defeitos corrigidos	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_01
MED21	Percentual de cobertura de Teste	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P01_03
MED22	Porcentagem de esforço gasto para correção de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P01_01
MED23	Quantidade de casos de testes executados	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_02 P02_04 P02_05 P02_06 P01_07
MED24	Quantidade de casos de testes executados com sucesso	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_03 P02_04 P02_05
MED25	Quantidade de casos de testes falhados	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_04
MED26	Quantidade de casos de testes planejados	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Planejamento	P02_03 P02_04

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED27	Quantidade de defeitos /velocidade	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03
MED28	Quantidade de defeitos não cobertos pelos casos de testes	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_01
MED29	Quantidade de funcionalidade que passam nos testes de aceitação	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03 P02_04 P01_14 P01_16
MED30	Quantidade de testes unitários	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03
MED32	Taxa de aumento de casos de testes entre versões/releases	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03
MED33	Taxa de Defeitos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_01 P01_02 P02_03
MED34	Tempo (em dias) para detecção de defeitos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_08
MED35	Tempo para corrigir os testes	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_03
MED36	Tempo para correção do defeito	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_03 P01_06
MED37	Total de defeitos abertos em um dia específico da release.	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_04
MED38	Total de Defeitos analisados e não removidos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03 P02_04 P01_04
MED39	Total de defeitos encontrados nos testes	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_01 P01_02 P02_01 P02_03 P02_02 P02_04 P02_05 P02_07 P02_06 P02_08 P01_03 P01_05 P01_13
MED40	Total de defeitos removidos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_04 P01_07
MED41	Qualidade da entrega	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P01_02

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED42	Confiança nos testes	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P02_01
MED43	Débito de teste	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P02_07

Observa-se ainda que apesar do foco da pesquisa ser em projetos que utilizem métodos ágeis, é possível observar que existem medidas identificadas que também podem estar presentes em projetos de contexto “tradicional”, exemplo: MED 39. Outras medidas só se fazem presente em projetos que utilizem métodos ágeis, exemplo MED 15.

O objetivo desta pesquisa é focado em metodologia ágil, pela importância que estas metodologias têm nos projetos atuais, que buscam acelerar as entregas, mantendo a qualidade e focando em desenvolvimento contínuo e entregas constantes, desta forma decidiu-se que esse foco traria um conjunto maior de medidas que possivelmente não são mapeadas em projetos ditos tradicionais.

Na Figura 19 é exibida a quantidade de medidas relacionadas a testes identificadas por cada artigo nos ciclos 1 e 2. No gráfico são exibidos o total de medidas de testes citadas por artigo, o total de medidas novas identificadas que não tinham sido encontradas durante o ciclo 1 e o total de medidas mapeadas anteriormente no ciclo 1 para cada artigo (por essa razão não há valores para os artigos dos ID P02_01 até P02_09, pois os mesmos foram mapeados somente no ciclo 2).

O ID do artigo é exibido no eixo X conforme descrição apresentado nas Tabelas 9 e 10. Os artigos de ID P01_01 até P01_16 foram identificados no ciclo 1, os artigos de ID P02_01 até P02_09 foram identificados no ciclo 2. Após uma análise de medidas citadas mais de uma vez ou medidas com nomes diferentes mais que significavam a mesma coisa, chegou-se a uma lista final de 39 medidas.

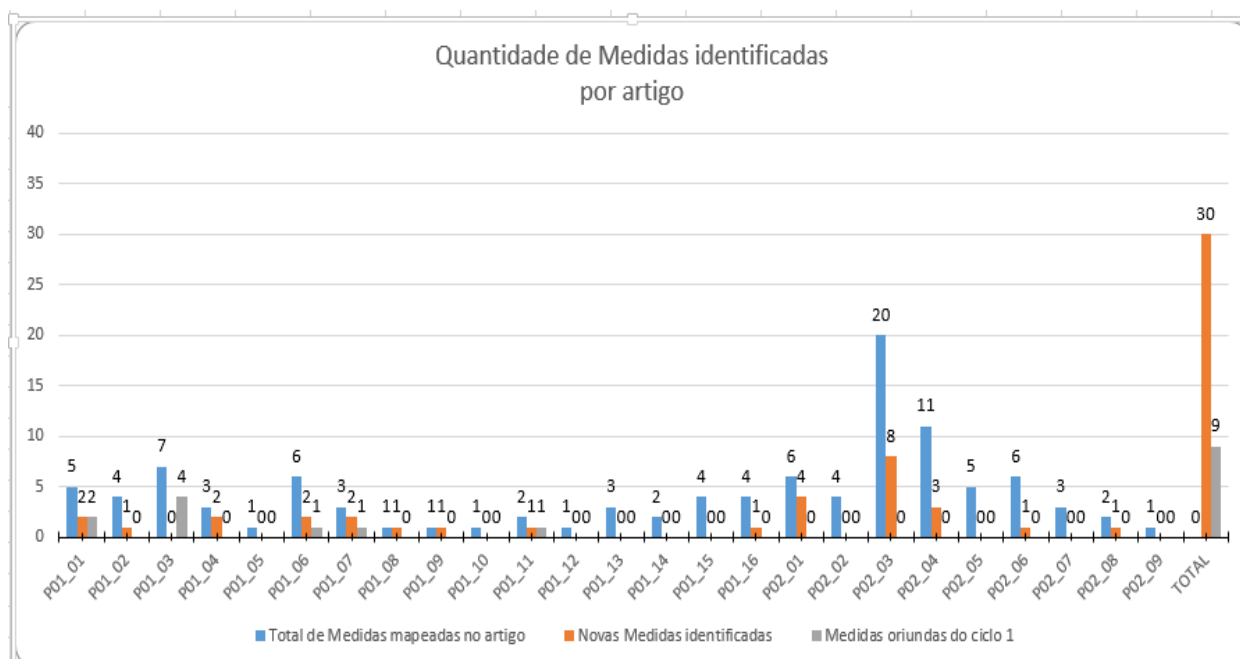


Figura 19 - Gráfico com a quantidade de medidas citadas por artigo e novas medidas

Na Figura 20, é exibida a quantidade de citações para cada medida nos artigos. As medidas são identificadas conforme descrito na Tabela 10. Como exemplo a medida MED39 - Total de defeitos encontrados nos testes, que foi citada em 24 artigos analisados durante o mapeamento sistemático.

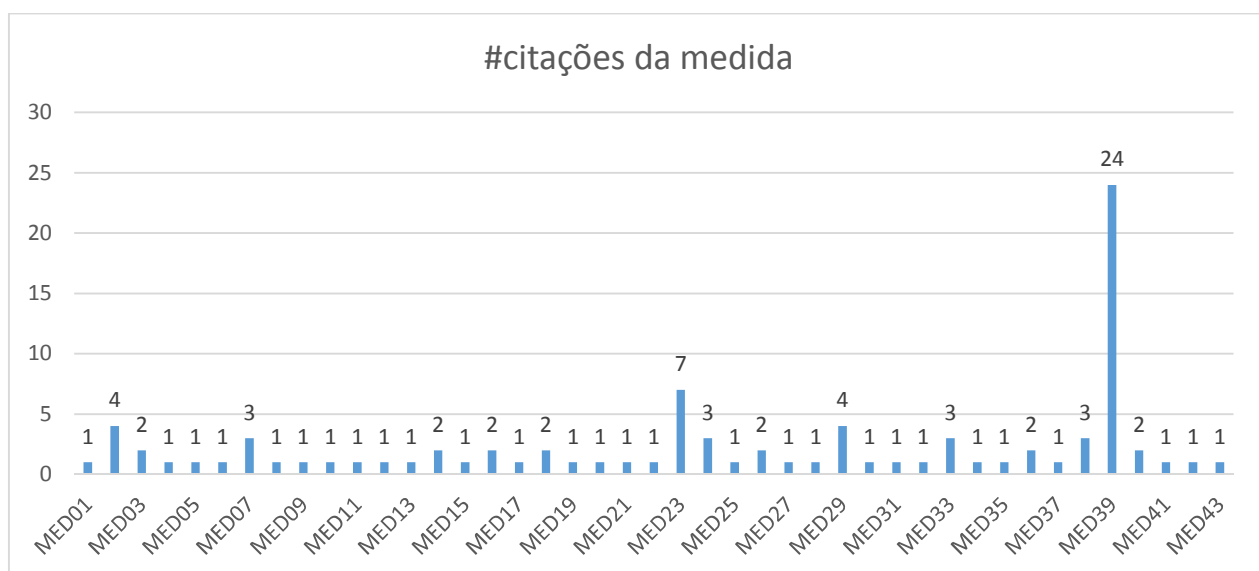


Figura 20 - Gráfico com a quantidade de citações em artigos de cada medida

Na Figura 21 é possível observar a quantidade de artigos mapeados com informação sobre medidas por cada ano de publicação, estes artigos listados são os 22 artigos que contribuíram de fato com medidas de testes para o mapeamento somando os ciclos 1 e 2.

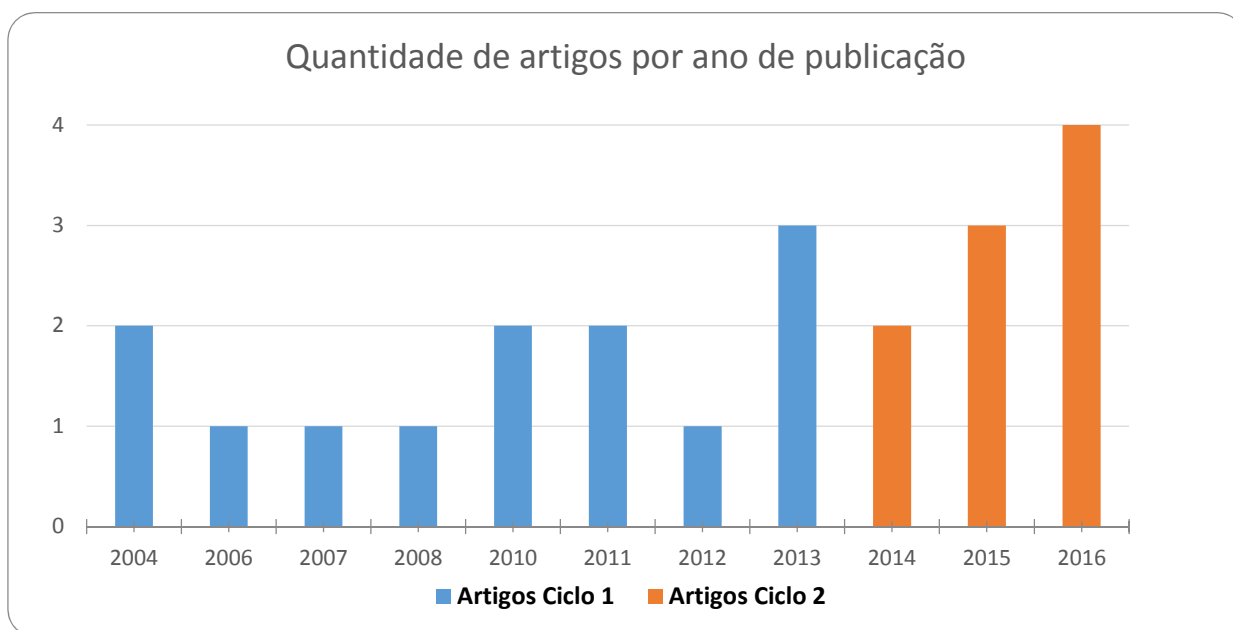


Figura 21 - Gráfico com a quantidade de artigos por ano de publicação

3.3. Indicadores de testes em projetos ágeis

Após a definição do conjunto de 39 medidas identificadas pelo mapeamento, o conjunto de medidas foi revisado por 2 especialistas em engenharia de software com alto grau de conhecimento para avaliar possíveis medidas duplicadas em razão de utilizarem nomes semelhantes na língua inglesa ou nomes diferentes executando medições semelhantes. Também foram removidas do escopo de análise para criação de indicadores algumas medidas que pela análise dos artigos o cálculo da medida era definido como sendo de forma subjetiva, e não sendo apresentado qualquer tipo de fórmula, não deixando claro como calcular. Com isso, foi possível responder à questão de pesquisa “**QP2:** Que indicadores relacionados a teste durante o desenvolvimento ágil de software que podem ser derivados das medidas citadas na literatura?”.

Desta forma, 15 indicadores foram propostos com base nas medidas identificadas. Estes indicadores são derivados das medidas após análise do contexto dos artigos em que são citados e das informações de como coletar ou calcular as mesmas. Além disso, houveram medidas que foram agrupadas em um único indicador por serem medidas semelhantes com

nomes diferentes, ex. a medida cobertura de teste e Percentual de cobertura de testes, foram agrupadas no indicador Índice de Cobertura de Testes.

A Tabela 11 apresenta a associação entre as medidas e os indicadores definidos.

O conjunto de 15 indicadores definidos são descritos na Tabela 11 com nome do indicador, sigla e descrição do indicador e ID sequencial criado para identificar o indicador.

Tabela 11 – Lista de Indicadores definidos com base na análise das medidas mapeadas

ID	Indicador resultante	Sigla	Descrição
01	Índice de Cobertura de Teste	ICT	Índice de cobertura de teste, considerando total e por tipo de teste (unidade, integração, funcional e aceitação). Este indicador será calculado de pelo total de requisitos já desenvolvidos que tenham ao menos um caso de teste associado sobre o total de requisitos do projeto já desenvolvidos do projeto/iteração.
02	Taxa de Defeitos	TD	Taxa de defeitos (abertos, analisados e não resolvidos, removidos) por iteração e por projeto. Total de defeitos (cada tipo) sobre o total de defeitos identificados no projeto/iteração.
03	Quantidade de defeitos reportados pelo cliente	QDRC	Quantidade de defeitos reportados pelo cliente (por severidade). Será calculado pelo somatório de defeitos acumulado por severidade por mês/período definido pelo projeto/iteração.
04	Densidade de Defeitos	DD	Densidade de defeitos (por projeto, por iteração, por complexidade, por severidade). Será calculado pela quantidade de defeitos por severidade ou complexidade acumulado por iteração sobre o tamanho do projeto conforme definição (linhas de código, pontos de função, histórias de usuários).
05	Taxa de funcionalidades que passam nos testes de aceitação automatizados	TFTA	Taxa de funcionalidades que foram implementadas e automatizadas. Calculado pela quantidade de funcionalidades aceitas nos testes automatizados sobre o total de funcionalidades implementadas no projeto.
06	Tempo entre detecção de defeitos	TDD	Tempo decorrido entre detecções consecutivas de defeitos por parte do cliente. Será calculado pelo número de dias em espera, até a correção do defeito para cada dia da iteração.
07	Taxa de aumento de casos de testes entre releases	TACT	Taxa de aumento de casos de testes em cada release. Será calculado pelo percentual de aumento de casos de testes entre releases.
08	Taxa de casos de testes automatizados	TCTA	Este indicador representa o número de casos de testes automatizados sobre o total de casos de testes existentes em cada iteração.
09	Prontidão para release	RR	Número de semanas previstas para disponibilização da release com base no número de defeitos identificados. O indicador sinalizará em quantas semanas a release estará liberada para implementação.
10	Tempo médio para correção de um defeito	TMCD	Será calculado para verificar qual o tempo médio de correção de defeitos por cada nível de severidade, considerando também o SLA definido pelo projeto.
11	Tempo para correção dos testes	TCT	Será calculado para verificar o tempo para correção dos defeitos identificados nos testes em cada rodada de testes ou iteração.
12	Taxa de casos de testes executados em relação ao planejado	TCEX	Taxa de casos de testes (não executados, executados com sucesso, executados com falha) em relação ao planejado. Será calculado para cada medida sobre o total de caso de teste planejados para a iteração/projeto.
13	Esforço gasto nos testes	EGT	Esforço gasto por tipo de teste (unidade, integração, funcional e aceitação). Será calculado pela porcentagem de horas gastas em testes sobre total de horas da gastas iteração.

ID	Indicador resultante	Sigla	Descrição
14	Quantidade de defeitos não cobertos por casos de testes	QDNC	Quantidade de defeitos identificados no software pela equipe não capturados pelos casos de teste.
15	Esforço gasto para correção de defeitos	EGCD	Esforço gasto para correção de defeitos por tipo de teste (unidade, integração, funcional e aceitação). Será calculado pela porcentagem de horas gastas na correção dos defeitos sobre total de horas gastas na iteração.

A Tabela 12 lista as medidas descartadas para criação de indicadores e a justificativa após análise junto aos especialistas.

Tabela 12 - Lista das medidas descartadas e justificativa.

ID Artigo	Medida	ID Medida	Justificativa
P02_01	Avaliação do backlog do produto	MED01	O indicador é calculado com base em medidas subjetivas e não quantitativas, os autores do artigo não proveem informações suficientes para auxiliar a análise do indicador, desta forma essa medida foi descartada.
P02_04	Quantidade de defeitos a serem removidos (backlog da release)	MED15	Não há clareza sobre como medir esse indicador com base no artigo.
P02_03	Número de defeitos previstos	MED11	Não há clareza sobre como medir esse indicador com base no artigo.
P02_03	Indicador de tendência de defeitos	MED12	Não há clareza sobre como medir esse indicador com base no artigo.
P02_01	Onda de Impacto dos defeitos corrigidos	MED20	A medida é muito específica, pois é parte do indicador "Avaliação do backlog do produto". Não será considerada para criação de indicadores.
P02_01	Confiança nos testes	MED42	A medida é muito específica, pois é parte do indicador "Avaliação do backlog do produto". Não será considerada para criação de indicadores.

A Figura 22 mostra a quantidade de artigos referenciados e quantidade de medidas utilizadas para criação de cada indicador. O eixo X mostra a sigla de cada um dos indicadores, nas barras temos a quantidade de artigos que foram referenciados com medidas para a definição do indicador seja com medidas repetidas ou com finalidades similares, e a outra coluna representa quantidade de medidas que foram citadas nos artigos e concatenadas para criação do indicador.

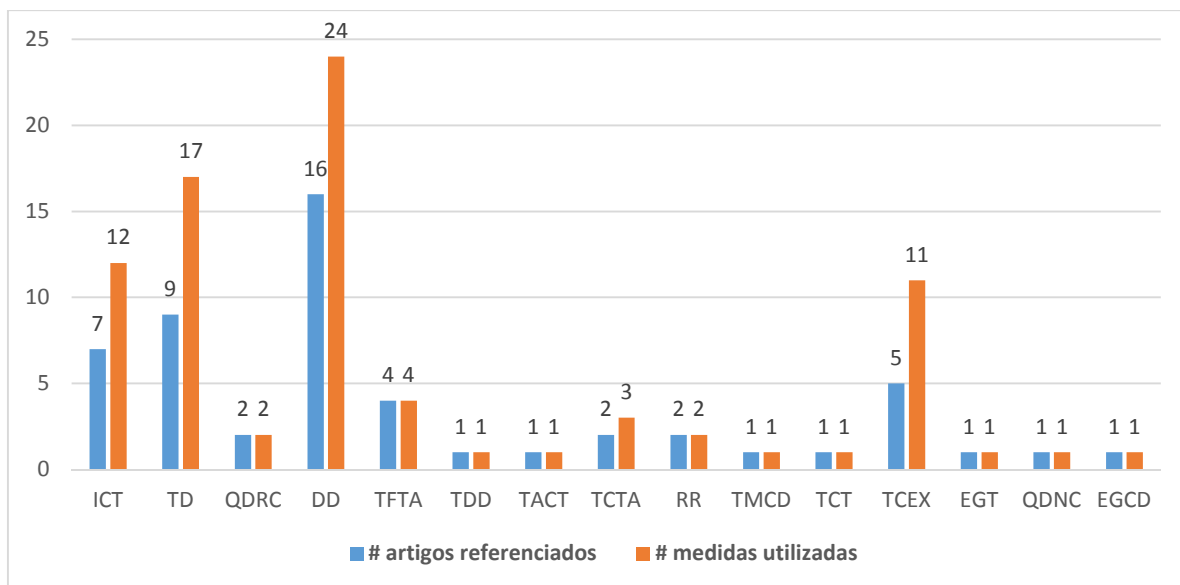


Figura 22 - Gráfico indicadores x medidas referenciadas nos artigos

Para os 15 indicadores definidos foram criadas fichas de registro com o conteúdo definido no template do ASM.br (descrito na Seção 2.3). As fichas de todos os indicadores definidos são apresentadas no Apêndice II. Além das fichas de registro de cada indicador descrito por esta pesquisa, outras fichas foram criadas como parte do template do ASM.br. Estas fichas são Mapa de Objetivos e Indicadores, Mapa para o MPS.br e Mapa de relacionamento dos indicadores.

Para identificação dos objetivos estratégicos, foram utilizados como base os objetivos estratégicos definidos por BONELLI (2014) e também a definição de cada indicador que foi definido como resultado do mapeamento sistemático. Depois foram também definidos objetivos de medição e necessidade de informação a serem atendidas para cada indicador. Embora o conjunto de indicadores não tenha sido definido para uma organização específica, espera-se que ele possa ser utilizado como base para líderes de testes gerenciarem seus projetos de testes e também como apoio para os gerentes elaborarem o plano de medição de seus projetos, uma vez que a definição dos indicadores foi baseado no mapeamento que identificou as medidas relacionadas a teste de software em projetos com desenvolvimento ágil.

O conjunto de indicadores criados nesta pesquisa não é um conjunto completo, capaz de atender todas as necessidades de organizações e projetos de desenvolvimento ágil no gerenciamento dos testes. O conjunto de indicadores proposto é um conjunto inicial, aderente à realidade de projetos de desenvolvimento ágil de software na fase de testes, que foi definido

com base nas medidas mapeadas no mapeamento sistemático e revisado por especialistas em engenharia de software.

Para selecionar os objetivos, estabelecer os relacionamentos entre os indicadores e os demais elementos, decisões foram tomadas e suas justificativas são apresentadas no decorrer deste capítulo.

Para definir o conjunto de indicadores foram seguidas quatro etapas: (i) Definição dos Objetivos e Necessidades de Informação, (ii) Seleção dos Níveis e Processos do MR-MPS-SW que se deseja atender, (iii) Seleção dos Indicadores e (iv) Elaboração dos Mapas de Relacionamento.

3.4. Definição dos Objetivos e Necessidades de Informação

Como dito anteriormente, os objetivos estratégicos definidos neste trabalho foram baseados nos utilizados em (BONELLI, 2014), que são: Aumentar Rentabilidade, Satisfação do Cliente, Aumentar Eficácia na Gestão. Esses objetivos são divididos em três perspectivas: Financeira, Clientes, Processos Internos. Os objetivos citados por BONELLI (2014) foram derivados a partir do mapeamento realizado por ASATO *et al.* (2006; 2011) com uma lista maior de objetivos definidos.

Para este trabalho foram adaptados os objetivos que são derivados dos objetivos originais do template ASM.br, a razão foi que desta forma os objetivos ficam mais direcionados ao propósito desta pesquisa que esta fortemente ligada a qualidade de software e também aos indicadores propostos.

Na Tabela 13 são apresentadas as justificativas para a inclusão dos objetivos definidos por Bonelli (2014) e o nome dos objetivos que foram re-definidos para que ficassem direcionados a proposta de gerenciamento de testes em projetos ágeis.

Tabela 13 - Objetivos propostos por BONELLI (2014) selecionados para o estudo

Objetivos do template ASM.br	Justificativa	Objetivos Estratégicos
Aumentar a Satisfação do Cliente	Os gerentes de projetos podem contribuir com o aumento da satisfação dos clientes por meio da qualidade das entregas realizadas.	Aumentar a qualidade do software
Aumentar a Satisfação do Cliente		Diminuir o tempo de entrega do software
Aumentar a eficácia da gestão	Este objetivo foi mantido, pois pode permitir que a gestão seja mais eficaz, por meio do monitoramento dos resultados alcançados.	Aumentar a eficácia da gestão

Após a definição dos objetivos estratégicos, foram definidos os objetivos de medição e identificadas as necessidades de informação. As necessidades de informação identificadas a partir desses objetivos de medição estão listadas nas Tabelas 14, 15 e 16.

Tabela 14 – Objetivos de medição e Necessidades de Informação para o Objetivo Aumentar qualidade do software

Objetivo estratégico Aumentar a qualidade do software	
Objetivos de Medição	Necessidade de Informação
Monitorar nível de cobertura de testes	Quanto dos requisitos estão sendo cobertos pelos casos de teste?
	Quantos casos de testes são implementados a cada release?
Monitorar o número de defeitos	Qual a taxa de defeitos do projeto?
	Qual a quantidade de defeitos externos?
	Qual é a densidade de defeitos do projeto?
	Qual o tempo decorrido entre diferentes detecções de defeito?
	Quantos defeitos não são capturados pelos casos de teste
Monitorar as funcionalidades verificadas	Qual a taxa de funcionalidades verificadas?
Monitorar a execução dos testes	Qual o status de execução dos casos de teste?
	Qual o grau de automação de caso de teste?

Tabela 15 – Objetivos de medição e Necessidades de Informação para o Objetivo Diminuir o tempo de entrega do software

Objetivo estratégico Diminuir o tempo de entrega do software	
Objetivos de Medição	Necessidade de Informação
Monitorar o tempo de liberação da versão para reteste	Qual a velocidade de correção dos defeitos?
	Quanto tempo demora para corrigir os defeitos identificados em uma rodada de teste?
Monitorar o tempo de entrega previsto de uma release	Para quando posso prever a disponibilização da release?

Tabela 16 – Objetivos de medição e Necessidades de Informação para o Objetivo Aumentar a eficácia da gestão

Objetivo estratégico Aumentar a eficácia da gestão	
Objetivos de Medição	Necessidade de Informação
Monitorar o tempo do projeto	Qual o esforço gasto nos testes?
	Qual o esforço gasto para correção dos defeitos?

3.5. Seleção dos Níveis e Processos do MR-MPS-SW

Para definição dos níveis de processos do MR-MPSPS-SW, foram analisados os objetivos de medição e estratégicos e assim definido que o nível de maturidade D seria o nível correto para todos os objetivos, uma vez que este nível é o que inclui as práticas de verificação e validação, que visam melhorar a produção de software e sua relação com a qualidade. Os processo selecionados foram Verificação (VER), Validação (VAL) e Integração do Produto (ITP). A Tabela 17 lista os processos selecionados e a justificativa para a seleção.

Tabela 17 – Processos do MR-MPS-SW selecionados

Processo	Justificativa
ITP – Integração do Produto	Esse processo foi selecionado, pois visa apoiar a integração de vários componentes para determinar tempo de entrega e controle do produto que está sendo desenvolvido.
VAL – Validação	Este processo foi selecionado, pois seu objetivo é confirmar que um produto atenderá a seu uso pretendido quando for colocado no ambiente para o qual foi desenvolvido.
VER – Verificação	O processo Verificação trata de como avaliar produtos de trabalho e serviços, garantindo que atendam a seus requisitos, por meio da identificação dos itens a serem verificados, do planejamento da verificação de cada um destes itens e da execução da verificação conforme planejado.

3.6. Seleção dos Indicadores

Os indicadores definidos a partir do mapeamento sistemático foram avaliados por outros dois pesquisadores, e a conclusão foi que seria indicado criar a ficha para todos, uma vez que cobrem pontos distintos da medição no gerenciamento de testes. A lista dos indicadores foi citada na Seção 3.3, Tabela 11.

Para definição dos indicadores as medidas utilizadas foram oriundas do mapeamento, porém para algumas medidas foi necessário criar medidas derivadas para a coleta dos dados. A Tabela 18 lista as medidas utilizadas para obtenção dos indicadores.

Tabela 18 – Medidas utilizadas para criação dos indicadores

Medida	Descrição	Indicadores
DTAD	Data de registro da ocorrência anterior de defeitos	TDD
DTDF	Data de registro dos defeitos	TDD
QCTTT	Quantidade de casos de teste por tipo de teste (unidade, integração, funcional, aceitação)	ICT
QDA	Quantidade de Defeitos Abertos	TD

Medida	Descrição	Indicadores
QDANR	Quantidade de Defeitos Analisados e Não Removidos	TD
QDR	Quantidade de Defeitos Removidos	TD
QDSV	Quantidade de defeitos por Severidade	QDRC
QFTS	Quantidade acumulada de funcionalidades aceitas nos testes automatizados	TFTA
QHCD	Quantidade de horas gastas para correção de defeitos (de cada tipo de teste)	EGCD
QTCT_ANTERIOR	Quantidade acumulada de casos de teste implementados na release anterior	TACT
QTCT_ATUAL	Quantidade acumulada de casos de teste implementados na release atual	TACT
QTCTA	Quantidade de casos de testes automatizados	TCTA
QTCTD	Quantidade de casos de teste desenvolvidos	TCTA
QTCTD	Quantidade de casos de testes planejados	TCEX
QTCTEX	Quantidade de casos de teste executados	RR
QTCTS	Quantidade de casos de teste com sucesso	RR
QTD	Quantidade de defeitos identificados	DD
QTD	Quantidade de defeitos únicos reportados pelo cliente	TDD
QTDCP	Quantidade de defeitos identificados por complexidade	DD
QTDF	Quantidade de defeitos identificados	RR
QTDFNC	Quantidade de defeitos não cobertos pelos casos de teste associados	QDNC
QTDFR	Quantidade de defeitos removidos	RR
QTDFSV	Quantidade de defeitos de severidade 1,2,3	TMCD
QTDFT	Quantidade total de defeitos da release	QDNC
QTDSV	Quantidade de defeitos identificados por severidade	DD
QTEXF	Quantidade de casos de testes executados com falha	TCEX
QTEXS	Quantidade de casos de testes executados com sucesso	TCEX
QTHGSV	Quantidade de horas gastas para correção dos defeitos de severidade 1,2,3	TMCD
QTHRC	Quantidade de horas gastas para correção dos testes em cada rodada de testes	TCT
QTHRIT	Quantidade de horas totais gastas na iteração	EGT
QTHRIT	Quantidade de horas totais gastas na iteração	EGCD
QTHRT	Quantidade de horas gastas nos testes (por tipo de teste)	EGT
QTI	Quantidade acumulada de funcionalidades implementadas	TFTA

Medida	Descrição	Indicadores
QTNEX	Quantidade de casos de testes não executados	TCEX
QTR	Quantidade acumulada de requisitos já desenvolvidos	ICT
QTRCTA	Quantidade acumulada de requisitos já desenvolvidos que tenham ao menos um caso de teste associado	ICT
SLAD	Limite máximo aceito de defeitos do projeto	QDRC
SLADFSEV	SLA de defeitos acordado para severidade	TMCD
TS	Tamanho do software (por exemplo, por 1000 linhas de código, pontos de função ou histórias de usuário).	DD

3.7. Mapa de relacionamentos

Os mapas de relacionamentos tem propósito em auxiliar a percepção das relações entre indicadores, objetivos e níveis de processos do MR-MPS-SW e, consequentemente apoiar os responsáveis pelos testes na seleção de indicadores. A Figura 23 apresenta o mapa de objetivos e indicadores relacionados.

MAPA DE OBJETIVOS

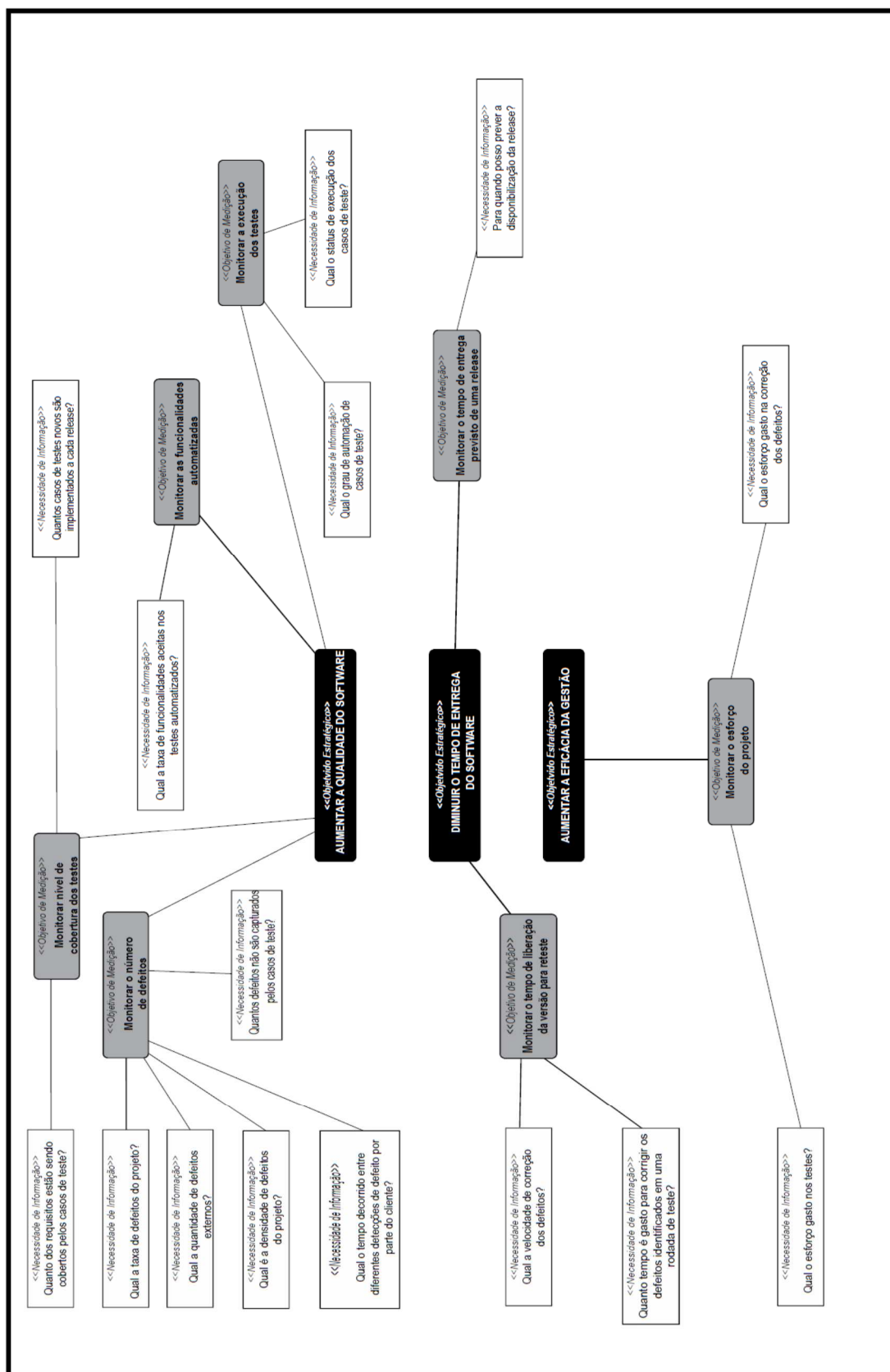


Figura 23 – Mapa de objetivos e indicadores

A Tabela 19 apresenta o Mapa para o MR-MPS-SW.

Tabela 19 – Mapa de relacionamento entre indicadores e processos do MR-MPS-SW



MAPA DE MR-MPS-SW

MAPA DE MR-MPS-SW					
INDICADORES		MR-MPW-SW			
		D			
SIGLA	NOME	VAL	VER	PI	
ICT	Índice de Cobertura de Teste	x	x		
TD	Taxa de Defeitos		x		
QDRC	Quantidade de defeitos reportados pelo cliente	x			
DD	Densidade de Defeitos		x		
TFTA	Taxa de funcionalidades que passam nos testes de aceitação	x			
TDD	Tempo entre detecção de defeitos	x			
TACT	Taxa de aumento de casos de testes entre releases		x		
TCTA	Taxa de casos de testes automatizados		x		
RR	Prontidão para release			x	
TMCD	Tempo médio para correção de um defeito		x		
TCT	Tempo para correção dos testes		x		
TCEX	Taxa de casos de testes executados em relação ao planejado		x		
EGT	Esforço gasto nos testes	x	x		
QDNC	Quantidade de defeitos não cobertos por casos de teste		x		
EGCD	Esforço gasto para correção de defeitos	x	x		

A Figura 24 apresenta o Mapa de relacionamento dos indicadores e as categorias relacionadas.



Para os 15 indicadores definidos, foram criadas fichas com o conteúdo definido no template do ASM.br (descrito na Seção 2.3) detalhadas no Apêndice II. A Figura 25 apresenta uma ficha criada para o indicador TD – Taxa de defeitos, conforme a seguir.

Representação gráfica do indicador – Exibe um gráfico que será alimentado através das planilhas de coleta de medidas e possibilitará a análise de evolução do indicador.

Procedimento de medição e análise – Detalha qual os artefatos, medidas necessárias para coleta e formula de geração do indicador, assim como informações de como analisar o indicador gerado.

Procedimento de análise baseado em critérios – Aponta premissas definidas com base nos possíveis valores e resultados para os indicadores, para cada premissa uma possível conclusão para a razão do indicador apurado é sugerida e possíveis ações também são descritas que possa apoiar o gestor responsável na resolução ou mitigação do problema.

Algumas das premissas possuem valores que devem ser definidos de acordo com a organização ou projeto, como por exemplo valores classificados como média, alta ou muito baixo. As premissas sugeridas como parte desta pesquisa poderão ser alteradas de acordo com as necessidades de cada organização para melhor ajuste aos processos organizacionais.

Em geral as ações de um nível inferior são aplicáveis aos níveis superiores tendo em vista que faixa do indicador poderá atingir o nível anterior em algum momento do monitoramento e a ação também ser recomendada para apoio na análise.

Procedimento de análise baseado em relacionamento – Apresenta o relacionamento do indicador que está sendo detalhado na ficha do ASM.br com outros indicadores que fazem parte da pesquisa, e possíveis influências positivas ou negativas destes relacionamentos, como possíveis ações para cada indicador relacionado.

As fichas para serem utilizadas efetivamente, precisam ser adaptadas para o contexto dos projetos desenvolvidos. Entende-se que o esforço inicial de adaptação envolve o entendimento de medição de software e dos objetivos estratégicos e de medição da organização de forma a defini-los usando o mapa de objetivos e definir os indicadores a serem criados. É necessário também a revisão das seções de coleta, medição e análise baseado nas particularidade dos projetos e da organização em questão. Para edição das fichas, é necessário um software para edição do template e o preenchimento das informações definidas, tendo a preocupação de não poluir visualmente a ficha, pois isso pode dificultar a leitura futura.

3.8. Planilha de apoio a coleta de dados e geração dos gráficos para os indicadores

Visando apoiar a coleta e representação de dados para análise, foram definidas planilhas para apoiar a coleta de dados e geração dos gráficos para os indicadores. Nesta Seção é apresentada, como exemplo, a planilha definida para apoiar a coleta e representação de dados do indicador taxa de defeitos. O conjunto completo de planilhas está disponível no Apêndice III.

Na planilha de coleta dos tipos de defeitos, o responsável pela coleta (líder de testes) deve preencher apenas as colunas em cor amarela na planilha conforme instrução de preenchimento coletando a quantidade de defeitos de cada tipo indicado. A planilha está configurada para realizar o cálculo da quantidade total de defeitos registrados na iteração e das taxas de defeitos para cada tipo (defeitos abertos, defeitos analisados e não removidos e defeitos removidos). Na Figura 26 é exibido um exemplo da planilha de coleta e cálculo do indicador taxa de defeitos (TD).

INDICADOR	Instruções de preenchimento:						
TD - Taxa de Defeitos	1-Preencher apenas as colunas em cor amarela na tabela de cada linha de iteração para as medidas: QDA - Quantidade de defeitos abertos QDANR - Quantidade de defeitos analisados e não removidos QDR - Quantidade de defeitos removidos						
Medidas ----->	QDA	QDANR	QDR		TAXA		
ITERAÇÕES	ABERTO	ANALISADOS e NAO REMOVIDOS	REMOVIDOS	TOTAL DEFEITOS	TD (Abertos)	TD(Analisados e Não Resolvidos)	TD (Removidos)
Iteração 1	10	1	5	16	63%	6%	31%
Iteração 2	6	3	5	14	43%	21%	36%
Iteração 3	2	4	6	12	17%	33%	50%
Iteração 4	1	1	10	12	8%	8%	83%
Iteração 5	0	1	16	17	0%	6%	94%

Figura 26 - Exemplo de planilha de medição: Taxa de defeitos

3.9. Considerações Finais

Neste capítulo foi abordado como as medidas de teste utilizadas em projetos de software com metodologia ágil foram identificadas por meio de um mapeamento sistemático que contribuiu com uma lista final de 39 medidas. A partir desta lista de medidas foram propostos 15 indicadores para apoio no gerenciamento dos testes. Para estes indicadores foram definidos critérios de avaliação e análise, que foram catalogados em fichas do template ASM.br (Bonelli, 2014) (Bonelli *et al.*, 2017).

Na conclusão do mapeamento sistemático da literatura, verifica-se a ausência de trabalhos que auxiliem a identificação de indicadores de testes que possam auxiliar na gerência de projetos de testes na tomada de decisão. Este fato motiva estudos mais aprofundados nesta área, para que as organizações consigam identificar as verdadeiras causas de seus problemas com testes e que possam assim mapear os problemas que podem afetar a conclusão dos testes. Alguns poucos indicadores identificados foram criados para projetos específicos com características próprias e que sua utilização se restringe ao contexto do artigo em questão.

A experiência do pesquisador como líder de testes e forte atuação na área de qualidade por vários anos, também é um ponto a ser ressaltado, pois ajudou na definição dos indicadores, além do conteúdo de medidas e indicadores mapeados.

As principais limitações do mapeamento sistemático apresentado são:

- Não ter o resultado de retorno da string e análise dos critérios de inclusão e exclusão no subconjunto de artigos do mapeamento de Mota e Dias-Neto, 2014 (Ciclo 1). Por este motivo foi decidido aplicar o critério de inclusão e exclusão nos 24 artigos selecionados para garantir que atendiam de fato o objetivo da pesquisa e confirmar se as medidas listadas realmente tinham origem dos artigos citados pelos autores.
- Não considerar outras bases, além das citadas no mapeamento Ciclo 1. Tendo em vista que o trabalho apresentado foi a replicação do mapeamento de Mota e Dias-Neto (2014) (Ciclo 1), apenas as duas bases de pesquisa utilizadas foram consideradas no Ciclo 2, com isso a ameaça seria de que realizando a busca em outras bases de pesquisa, poderíamos ter identificados medidas que não foram mapeadas nos Ciclos 1 e 2.
- Há possibilidade de ter-se catalogado medidas utilizadas em projetos tradicionais que sejam aplicáveis em projetos ágeis e não terem sido identificados, devido o foco da busca ter sido em projetos ágeis. Porém, com base no conjunto de medidas identificadas observa-se um conjunto extenso de medidas que são aplicáveis a projetos “tradicionais” e “ágeis”, minimizando levemente este risco da pesquisa.
- Grande parte dos artigos não detalham o que são as medidas, nem como calculá-las, por isso, muitas vezes, a interpretação dependeu do pesquisador que executou o mapeamento sistemático. Para tratar essa limitação, a definição final das medidas (referente à questão de pesquisa QP1) foi revista por outro pesquisador e a definição

dos indicadores (referente à questão de pesquisa QP2) foi revista por dois pesquisadores com conhecimento em Engenharia de Software.

4. Avaliação dos Indicadores de Testes para Projetos Ágeis

Este capítulo descreve o estudo de caso realizado em uma organização em projetos de teste de software, com o objetivo de verificar a análise das fichas que foram propostas nesta dissertação para o registro de indicadores de testes e mapeamento de seus relacionamentos. Na Seção 4.1 é apresentado o objetivo geral da avaliação. A Seção 4.2 detalha o planejamento do estudo de caso. A Seção 4.3 apresenta a execução do estudo de caso, onde os indicadores foram avaliados e algumas medidas foram coletadas pelos participantes com geração dos gráficos. Na Seção 4.4 a análise dos resultados do estudo de caso é realizada, na Seção 4.5 as considerações finais do capítulo são apresentadas.

4.1. Introdução

Após a execução do mapeamento sistemático para identificar medidas de testes e definição de indicadores de testes e a criação das fichas de registro com informações de coleta, cálculo, análise e ações para cada indicador foi realizado um estudo de caso para avaliar o uso do conteúdo proposto nas fichas de indicadores dentro do contexto de uma organização.

O estudo de caso descrito nas próximas seções visa avaliar o conteúdo das fichas de registro dos indicadores propostas no capítulo 3 em uma organização de desenvolvimento de software que utiliza metodologia ágil. A avaliação foi utilizada para identificar indícios de que o conteúdo das fichas e o formato das mesmas podem ser utilizados pela indústria. Para alcançar o objetivo do estudo de caso utilizou-se o modelo Technology Acceptance Model – TAM (DAVIS, 1986) (DAVIS et al., 1989) (BAGOZZI, 1992) (LAITENBERGER e DREYER, 2008), que procura determinar os aspectos de utilidade e facilidade de uso de

uma tecnologia segundo a percepção de seus usuários. Para organizar este estudo foi aplicado o paradigma Goal/Question/Metric – GQM (Basili *et al.*, 1994).

4.2. Planejamento

Foi realizado um planejamento que pudesse apoiar na definição e depois a execução do estudo de caso. As etapas definidas do planejamento foram:

- Definição dos objetivos da avaliação.
- Caracterização da organização e participantes alvo.
- Definição do método de avaliação.
- Definição do método de análise dos resultados.

Para execução do estudo de caso as seguintes etapas foram definidas:

- Caracterização da organização.
- Caracterização dos participantes.
- Capacitação para utilização das fichas e indicadores.
- Avaliação do conteúdo das fichas de registro dos indicadores.
- Coleta de medidas e geração dos gráficos com apoio das planilhas de coleta e análise dos resultados.
- Avaliação da percepção da equipe.
- Análise dos resultados.

4.2.1. Definição dos objetivos

O objetivo da avaliação é **analisar** o conteúdo das fichas de indicadores de testes para projetos ágeis com o **propósito** de avaliar sua utilidade, facilidade de uso e aplicabilidade **do ponto de vista** de líderes de teste e gerentes de projeto **no contexto de** projetos de desenvolvimento de software utilizando metodologias ágeis.

Para guiar a avaliação, o objetivo principal foi decomposto utilizando os dois conceitos básicos do TAM (DAVIS *et al.*, 1989): a percepção de utilidade e facilidade de uso. Para apoiar a análise dos dados coletados, foi incluído um objetivo adicional criado em (BONELLI, 2014) para identificar se o participante do estudo de caso, independente das respostas dadas às questões geradas a partir dos objetivos do TAM, demonstra interesse em utilizar ou não os indicadores no futuro. Assim os objetivos decompostos foram:

- Avaliar a facilidade de uso do conteúdo proposto nas fichas do ASM.br;

- Avaliar a utilidade dos procedimentos e análise dos indicadores, assim como clareza de entendimento.
- Identificar se os participantes do estudo demonstram interesse no uso futuro dos indicadores propostos.

4.2.2. Caracterização da organização e participantes alvo

Perfil da Organização:

Organizações que desenvolva software utilizando metodologias ágeis ou que executem testes utilizando práticas ágeis de gerenciamento. Para avaliação da ficha de indicadores, é indicado que seja aplicado em uma organização que possua projetos de testes gerenciados por gerentes de projetos ou líderes de testes, com possibilidade de coleta das medidas de testes sugeridas e aplicação dos indicadores nos períodos de iteração dos projetos. É importante que exista um líder ou gerente de projetos que possa analisar as fichas de indicadores propostas.

Participantes do estudo:

Líderes ou gerentes de projetos que atuem na gestão de projetos com testes há mais de 3 meses dentro da organização onde o estudo de caso está sendo realizado. Devido à complexidade dos projetos dentro da organização e período de adaptação dos novos colaboradores que ocorre para alguns projetos, foi decidido que apenas colaboradores com mais de 3 meses atuando na organização seriam convidados para participar do estudo. Desta forma já estariam habituados aos processos e projetos e, assim, podem de maneira mais consciente avaliar as fichas no contexto de seus projetos de teste dentro da organização.

Os participantes serão convidados para participarem do estudo pelo pesquisador de forma direta e pessoalmente. Neste momento o pesquisador fará uma breve explicação sobre a pesquisa, caso o participante concorde em participar, será agendada uma reunião para explicação detalhada da proposta da pesquisa e como analisar o conteúdo das fichas e realizar a coleta das medidas.

4.2.3 Definição do método de avaliação

A Tabela 20 apresenta a estrutura de objetivos e questões, com base no GQM (BASILI *et al.*, 1994), para a avaliação que será executada na etapa de execução. Os dois primeiros níveis da Figura apresentam os objetivos do estudo (identificados pela letra G).

Associadas a cada objetivo são apresentadas as questões (identificadas pela letra Q). As métricas para resposta as questões são listadas na Tabela 21. Os participantes devem responder um formulário ao final da avaliação com as questões abaixo e utilizar as métricas da Tabela 21 para responder as questões 1 até 13. As questões 14 e 15 serão questões com respostas subjetivas para que o respondente possa utilizar uma maior amplitude de comentários pessoais sobre sua análise. As perguntas foram adaptadas do trabalho de Bonelli (2014).

Tabela 20 - Estrutura GQM para avaliação da etapa execução

G0: Percebe-se facilidade de uso e aplicabilidade das fichas de indicadores do ponto de vista de líderes de teste e gerentes de projetos no contexto de projetos de desenvolvimento de software utilizando metodologias ágeis?		
G1:Avaliar a facilidade de uso das fichas do ASM.br.	G2:Avaliar a utilidade dos procedimentos e análise dos indicadores, assim como clareza de entendimento.	G3:Identificar se os participantes do estudo demonstram interesse no uso futuro dos indicadores propostos
Q1:Foi fácil entender as informações presentes nas fichas do ASM.br.	Q4:Considero as fichas úteis para monitorar os indicadores de teste.	Q11: Considero viável o uso das fichas na minha organização.
Q2: Os procedimentos de análise descritos nas fichas estão fáceis de serem entendidos.	Q5: Considero que o uso das fichas do ASM.br melhoraria o meu desempenho na análise dos resultados das medições.	Q12: Consideraria a utilização deste indicador no dia-a-dia do seu projeto de testes.
Q3:Considero o ASM.br fácil de utilizar.	Q6: Os indicadores sugeridos são aplicáveis dentro da minha organização.	Q13: Considero viável a coleta das medidas com base nas informações da ficha.
	Q7: As definições de análise e recomendações são úteis dentro do desenvolvimento de projeto.	Q14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando o esforço para coleta e análise dos dados?
	Q8: Os relacionamentos entre indicadores da ficha são pertinentes ao ponto de apoiarem na análise do indicador.	Q15: Você teria sugestões de melhoria para as fichas?
	Q9: Os indicadores sugeridos contribuem na tomada de ação dentro do projeto.	
	Q10: As fichas apresentam informações completas para análise do indicador.	

Tabela 21 - Métrica para respostas ao formulário de avaliação

Métrica	Descrição	Justificativa
1	Concordo totalmente	O participante concorda com a afirmação nos quesitos de facilidade, utilidade e interesse em utilização dos indicadores em projetos dentro da organização.
2	Concordo parcialmente	O participante tem dúvidas com relação a afirmação, ou entende que não são aplicáveis a todos os projetos da organização;
3	Indiferente	O participante não enxerga ganho ou perda nas afirmações em sua organização ou projetos de teste.
4	Discordo parcialmente	O participante não concorda com a afirmação, porém entende que possa ser utilizado em outras organizações.
5	Discordo totalmente	O participante entende que as afirmações não são aplicáveis a organização do estudo de caso.

4.2.4 Definição do método de análise dos resultados

A avaliação dos resultados será exibida em uma Tabela com as métricas de cada participante para cada questão respondida e também uma avaliação do percentual de respostas para cada questão realizada.

Será analisado também o tempo gasto por cada participante na avaliação das fichas de forma que possa perceber o esforço de tempo considerando uma primeira iteração dos respondentes com as fichas de indicadores.

4.3. Execução do estudo de caso

O fluxo para execução do estudo de caso foi definido da seguinte forma:

1 - Convite pessoal para participantes dentro do perfil definido. Oito profissionais que atenderiam o perfil foram convidados dentro da organização do estudo de caso, porém somente 5 confirmaram presença na reunião.

2- Reunião de capacitação para entendimento e análise das fichas. Os respondentes também entregam o termo de consentimento livre e esclarecimento (TCLE) assinados e o formulário de avaliação de perfil do participante.

3 – Distribuição das fichas para cada participante.

4 – Análise das fichas e entrega do formulário de avaliação preenchido pelos respondentes e o tempo gasto na análise das fichas (análise da informação do cabeçalho da ficha, procedimentos de coleta e medição, procedimentos baseados em critérios e procedimentos baseados em relacionamentos).

5 – Os participantes coletaram medidas de algum dos indicadores analisados, e com apoio das planilhas de medição geraram os gráficos para análise da ficha.

4.3.1 Caracterização da organização

A pesquisa foi realizada em uma organização da área de telecomunicações localizada no Rio de Janeiro, em projetos de teste de software que estão sob a gerência de qualidade ligada à diretoria de TI.

Este departamento possui um grupo de líderes de testes que atuam sob demanda nos projetos na fase de teste. Os líderes participam das fases de planejamento e execução de testes por projeto, assim como alinhamento e discussão de defeitos com as fábricas de desenvolvimento e com os líderes técnico durante a fase de execução dos testes.

Nesta organização, o desenvolvimento de software é realizado por diferentes fábricas de desenvolvimento (FD) (fornecedores). As fábricas utilizam metodologia ágil no desenvolvimento do software, porém a equipe de gestão de teste tem atuação limitada no processo e participação devido ao alto número de FDs e a grande quantidade de sistemas impactados nos projetos com diversas integrações entre as FDs.

Os líderes de testes (LDs) são responsáveis pela definição da estratégia de testes e contratação das fábricas de teste (FTs) durante a fase de planejamento do projeto. As FTs são responsáveis pelo desenvolvimento dos casos de teste e execução sob a gestão dos líderes de teste.

Durante a fase de execução, os líderes de teste são responsáveis pela gestão da execução dos casos de teste e defeitos que são reportados pelas FTs.

4.3.2 Caracterização dos participantes

Os LDs são responsáveis por reportar progresso das fases de teste, reportando desvios do plano seja por conta de problemas na atuação das FTs ou por conta de problemas causados pelas FDs. As fases de teste são planejadas em iterações que duram em média de 3 a 4 semanas conforme cronograma definido pelo departamento de release de TI. As releases com liberação do software para ambiente de produção ocorrem em média a cada 2 meses, após finalização das validações dos testes integrados e aceite da área funcional.

Neste estudo de caso, cinco líderes de testes participaram da avaliação das fichas dos indicadores.

A Figura 27 apresenta o TCLE – Termo de Consentimento e Livre Esclarecimento utilizado para formalizar a participação dos respondentes no estudo de caso. Este documento foi apresentado aos participantes ao final da reunião de capacitação.

	
TERMO DE CONSENTIMENTO LIVRE E ESCLARECIMENTO (TCLE) Universidade Federal do Estado do Rio de Janeiro, Centro de Ciências Exatas e Tecnologia. Linha de Pesquisa de Sistemas de Apoio a Negócios	
Pesquisador: Douglas Lima Orientadores: Gleison Santos, D.Sc. e Cristina Cerdeiral, D.Sc.	
Nome do Participante: _____	
Email do Participante: _____	
Descrição do Estudo	
Prezado(a) Senhor(a), Os alunos de mestrado da UNIRIO eventualmente realizam estudos experimentais para caracterizar/avaliar suas pesquisas. Este estudo é conduzido por Douglas Lima sob a orientação dos professores Gleison Santos e Cristina Cerdeiral. Você foi previamente selecionado pelo seu conhecimento/experiência e está sendo convidado a participar desta pesquisa. 1. Procedimentos: O estudo será realizado com data e hora marcada com os participantes pré-selecionados. Para participar do estudo normalmente será aplicado um formulário de caracterização de perfil, a fim de identificar seu nível de conhecimento/experiência. Em seguida, o estudo será executado de forma individual ou em grupos formados, seguindo sempre o planejamento do estudo feito pelo pesquisador(a) responsável. Caso seja necessário, ao final do estudo será solicitado ao participante que responda um questionário de avaliação sobre a tecnologia de software que está sendo caracterizada/avaliada. 2. Tratamento de possíveis riscos e desconfortos: Serão tomadas todas as providências durante a coleta de dados de forma a garantir a sua privacidade e seu anonimato. 3. Benefícios e Custos: Espera-se que, como resultado deste estudo, você possa aumentar seus conhecimentos, de maneira a contribuir para o aumento da qualidade das atividades com as quais você trabalhe ou possa vir a trabalhar. Este estudo também contribuirá com resultados importantes para a pesquisa que está sendo realizada. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à autorização dos seus dados na pesquisa. 4. Confidencialidade da Pesquisa: Toda informação coletada neste estudo é confidencial e seu nome não será identificado de modo algum, a não ser em caso de autorização explícita para este fim. Quando os dados forem coletados, seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados. 5. Participação: Sua participação neste estudo é muito importante e voluntária, pois requer a sua aprovação para utilização dos dados coletados neste estudo. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades. Em caso de você decidir se retirar do estudo, favor notificar o pesquisador(a) responsável (douglas.silvalima@uniriotec.br). Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas.. 6. Declaração de Consentimento: Declaro que li e estou de acordo com as informações contidas neste documento e que toda linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente, recebendo respostas para todas as minhas dúvidas. Confirmando também que recebi uma cópia deste Termo (TCLE), compreendo que sou livre para não autorizar a utilização dos meus dados neste estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e concordo de espontânea vontade em participar deste estudo.	
_____ Participante	_____ Douglas Lima

Figura 27 - Termo de consentimento para participação do estudo

Os participantes também responderam ao formulário de caracterização dos participantes para avaliar o nível de experiência dos mesmos dentro da área de TI e dentro da organização. A Figura 28 apresenta o formulário que foi preenchido após a reunião de capacitação.

FORMULÁRIO DE CARACTERIZAÇÃO DO PARTICIPANTE		
Nome:		
Qual sua atividade principal na empresa :		
Formação acadêmica ☐ Graduação ☐ Pós graduação / MBA ☐ Mestrado ☐ Doutorado		
Experiência na área de TI	Anos:	
Experiência na organização do estudo de caso	Anos:	Projetos: ☐ 1 ou 2 ☐ 2 a 5 ☐ mais de 5
Experiência na área de desenvolvimento	Anos:	Projetos: ☐ 1 ou 2 ☐ 2 a 5 ☐ mais de 5
Experiência na área de testes	Anos:	Projetos: ☐ 1 ou 2 ☐ 2 a 5 ☐ mais de 5
Experiência em gerenciamento de projetos	Anos:	Projetos: ☐ 1 ou 2 ☐ 2 a 5 ☐ mais de 5
Já executou ou executa atividades de medição	☐ Sim ☐ Não	Projetos: ☐ 1 ou 2 ☐ 2 a 5 ☐ mais de 5

Figura 28 - formulário de perfil dos participantes

Conforme resposta sobre o perfil dos participantes, constatou-se que todos os participantes possuíam no mínimo Graduação como formação acadêmica, sendo que um possuía Pós-graduação também, porém nenhum dos participantes possuía Mestrado. Todos já executaram atividades de medição em pelo menos um projeto. Na Figura 29 é apresentado o gráfico com nível de escolaridade dos participantes.

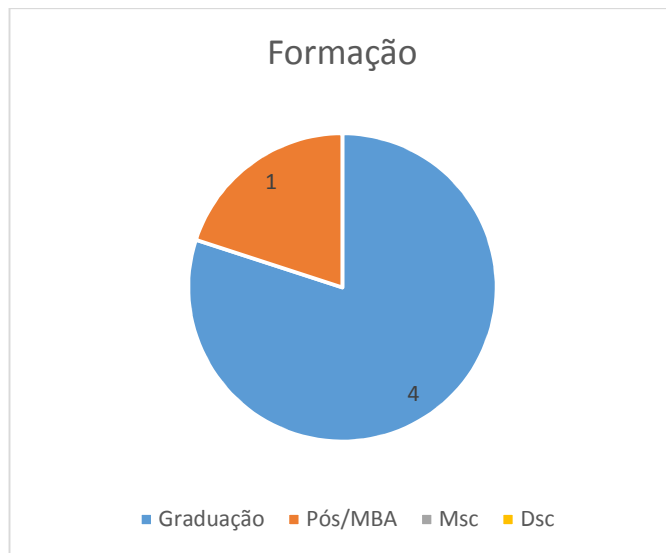


Figura 29 - Gráfico de formação acadêmica dos participantes

Na Figura 30 é exibido o gráfico da experiência dos participantes considerando anos de experiência e quantidade de projetos já participados. Todos os participantes (P[x]) possuem dez anos ou mais de experiência em TI, mais de um ano atuando dentro da organização do estudo e no mínimo dois anos na área de testes, sendo que dois possuíam mais de cinco anos na área de testes.

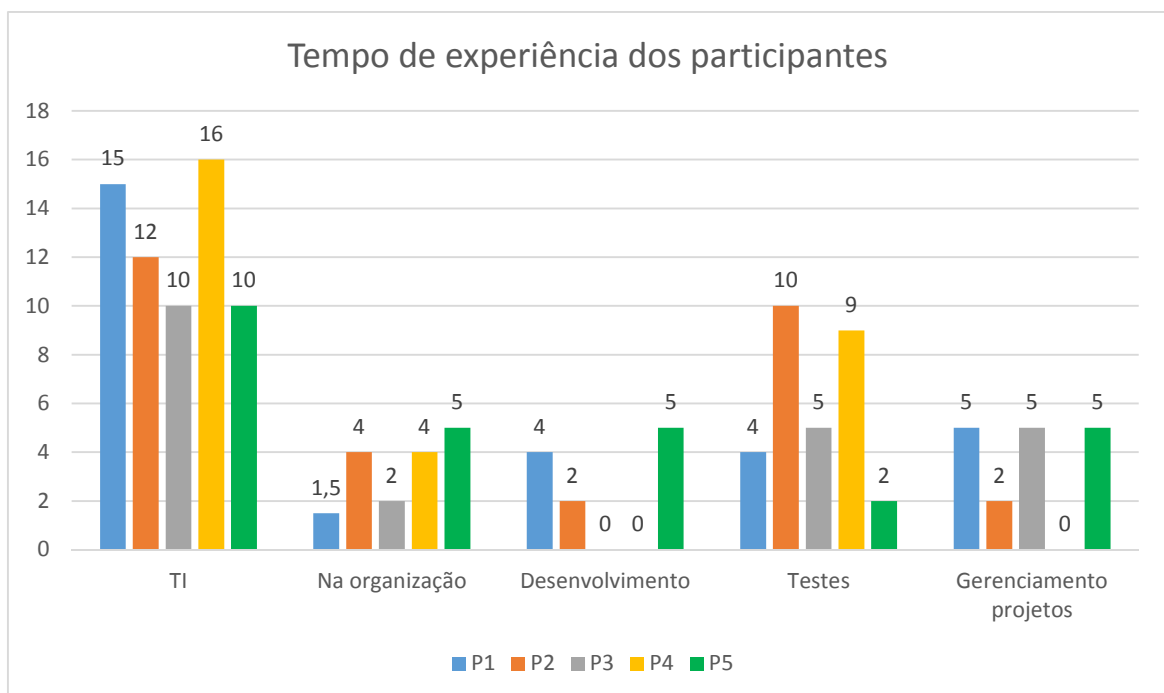


Figura 30 - Gráfico de tempo de experiência dos participantes

4.3.3 Capacitação para utilização das fichas de registro dos indicadores

A reunião de capacitação para entendimento e análise das fichas foi iniciada com agradecimento a todos os cinco participantes presentes e em seguida foi realizada uma apresentação que durou aproximadamente 40 minutos. Os respondentes também entregam o termo de consentimento livre e esclarecimento assinado e o formulário de avaliação de perfil do participante preenchido.

Foram esclarecidas dúvidas quanto à análise das fichas e preenchimento do formulário de avaliação que foi disponibilizado para cada participante. Na Figura 31 o formulário de avaliação é mostrado com as questões a serem respondidas ao final da avaliação das fichas de registro de indicadores.

 FORMULÁRIO DE AVALIAÇÃO DOS INDICADORES		
Pesquisador: Douglas Lima		
Orientadores: Gleison Santos, D.Sc. e Cristina Cerdeiral, D.Sc.		
Nome Participante:		
Data:		
Para as questões de 1 a 13 a seguir, resposta utilizando a escala abaixo: 1 = "Concordo Totalmente" 2 = "Concordo Parcialmente" 3 = "Indiferente" 4 = "Discordo Parcialmente" 5 = "Discordo Totalmente" <i>Observação:</i> Justifique as respostas que julgar necessário na folha em anexo. Para as questões 14 e 15, utilize a folha em anexo para responder.		
Questionário de avaliação:		
Análise	Questões	Resposta
Facilidade	Q1: Foi fácil entender as informações presentes nas fichas do ASM.br.	
	Q2: Os procedimentos de análise descritos nas fichas estão fáceis de serem entendidos.	
	Q3: Considero o ASM.br fácil de utilizar.	
Utilidade	Q4: Considero as fichas úteis para monitorar os indicadores de teste.	
	Q5: Considero que o uso das fichas do ASM.br melhoraria o meu desempenho na análise dos resultados das medições.	
	Q6: Os indicadores sugeridos são aplicáveis dentro da minha organização.	
	Q7: As definições de análise e recomendações são úteis dentro do desenvolvimento de projeto.	
	Q8: Os relacionamentos entre indicadores da ficha são pertinentes ao ponto de apoiarem na análise do indicador.	
	Q9: Os indicadores sugeridos contribuem na tomada de ação dentro do projeto.	
	Q10: As fichas apresentam informações completas para análise do indicador	
Interesse em usar	Q11: Considero viável o uso das fichas na minha organização.	
	Q12: Consideraria a utilização deste indicador no dia-a-dia do seu projeto de testes.	
	Q13: Considero viável a coleta das medidas com base nas informações da ficha.	
	Q14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando esforço para coleta e análise dos dados?	(use o verso)
	Q15: Você teria sugestões de melhoria para as fichas?	(use o verso)

Figura 31 - Ficha de avaliação dos indicadores

Após a capacitação, foram distribuídas três fichas para cada participante. Nesta pesquisa foram produzidas 15 fichas de indicadores, porém foi decidido distribuir apenas três fichas para cada participante devido à disponibilidade de tempo deles ser reduzida e ao risco de não terem tempo hábil de responder um número maior, uma vez que todos possuíam compromissos maiores no dia-a-dia da organização. Este risco foi levantado pelos próprios participantes durante a reunião de capacitação e o pesquisador decidiu então fazer a distribuição citada. Desta forma, cada participante recebeu um conjunto de três fichas de indicadores diferente, com isso foi possível avaliar todas as fichas durante o período do estudo de caso. A distribuição ocorreu de forma aleatória conforme Tabela 22, na coluna participante é exibido um código de identificação de cada participante.

Tabela 22 - Distribuição das fichas de indicadores para análise

Participante	Indicadores
P1	QDNC, TACT, TMCD
P2	TD,TDD,TFTA
P3	DD, TCEX, TCT
P4	EGT, ICT, RR,
P5	EGCD, QDRC, TCTA

4.3.4 Avaliação do conteúdo das fichas de registro dos indicadores

A avaliação dos resultados será exibida em uma Tabela com as métricas de cada resposta de cada participante para cada questão respondida com base na escala indicada. Os dados coletados nas questões são descritos, tanto em termos quantitativos (Tabela 23), quanto percentuais (Tabela 24) visando simplificar a análise dos resultados.

Na Tabela 23 são exibidas as 13 questões objetivas, e os participantes são identificados pelos códigos P1, P2, P3, P4 e P5 nas referidas colunas com os valores de cada resposta conforme escala citada na Tabela 20.

Tabela 23 - Respostas da avaliação dos indicadores

Questões	P1	P2	P3	P4	P5
Q1: Foi fácil entender as informações presentes nas fichas do ASM.br?	1	2	1	1	1
Q2: Os procedimentos de análise descritos nas fichas estão fáceis de serem entendidos?	1	2	1	1	1
Q3: Considero o ASM.br fácil de utilizar?	1	2	1	1	2
Q4: Considero as fichas úteis para monitorar os indicadores de teste?	1	1	1	1	1
Q5: Considero que o uso das fichas do ASM.br melhoraria o meu desempenho na análise dos resultados das medições?	1	1	1	1	1
Q6: Os indicadores sugeridos são aplicáveis dentro da minha organização?	2	2	1	2	2
Q7: As definições de análise e recomendações são úteis dentro do desenvolvimento de projeto?	1	1	1	2	1
Q8: Os relacionamentos entre indicadores da ficha são pertinentes ao ponto de apoiarem na análise do indicador?	1	1	1	2	1
Q9: Os indicadores sugeridos contribuem na tomada de ação dentro do projeto?	1	1	1	2	2
Q10: As fichas apresentam informações completas para análise do indicador?	1	2	1	1	1
Q11: Considero viável o uso das fichas na minha organização?	2	1	1	1	2
Q12: Consideraria a utilização deste indicador no dia-a-dia do seu projeto?	1	2	1	2	3
Q13: Considero viável a coleta das medidas com base nas informações da ficha?	2	1	1	3	3

A Tabela 24 mostra as 13 questões e as colunas para cada escala de resposta com os percentuais de resposta obtidos na avaliação.

Tabela 24 - Percentuais obtidos de cada questão

Questões	1	2	3	4	5
Q1: Foi fácil entender as informações presentes nas fichas do ASM.br?	80%	20%	0%	0%	0%
Q2: Os procedimentos de análise descritos nas fichas estão fáceis de serem entendidos?	80%	20%	0%	0%	0%
Q3: Considero o ASM.br fácil de utilizar?	60%	40%	0%	0%	0%
Q4: Considero as fichas úteis para monitorar os indicadores de teste?	100%	0%	0%	0%	0%
Q5: Considero que o uso das fichas do ASM.br melhoraria o meu desempenho na análise dos resultados das medições?	100%	0%	0%	0%	0%
Q6: Os indicadores sugeridos são aplicáveis dentro da minha organização?	80%	20%	0%	0%	0%
Q7: As definições de análise e recomendações são úteis dentro do desenvolvimento de projeto?	80%	20%	0%	0%	0%
Q8: Os relacionamentos entre indicadores da ficha são pertinentes ao ponto de apoiarem na análise do indicador?	80%	20%	0%	0%	0%
Q9: Os indicadores sugeridos contribuem na tomada de ação dentro do projeto?	60%	40%	0%	0%	0%
Q10: As fichas apresentam informações completas para análise do indicador?	80%	20%	0%	0%	0%
Q11: Considero viável o uso das fichas na minha organização?	60%	40%	0%	0%	0%
Q12: Consideraria a utilização deste indicador no dia-a-dia do seu projeto?	40%	40%	20%	0%	0%
Q13: Considero viável a coleta das medidas com base nas informações da ficha?	40%	20%	40%	0%	0%

Para as questões 14 e 15 do formulário de avaliação os participantes deram respostas subjetivas avaliando o conjunto de indicadores que cada um recebeu para análise. A seguir é apresentado o detalhamento das respostas e observações de cada participante.

Participante 1 – Avaliou as fichas dos indicadores QDNC, TACT, TMCD.

Questão 14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando o esforço para coleta e análise dos dados?

Resposta: TMCD, este indicador auxilia na tomada de decisão relacionada a um ponto que pode impactar diversas equipes do projeto como desenvolvimento e testes, correção de defeitos.

Questão 15: Você teria sugestões de melhoria para as fichas?

Resposta: Não

Participante 2 – Avaliou as fichas dos indicadores TD, TDD, TFTA.

Questão 14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando o esforço para coleta e análise dos dados?

Resposta: TD, pois acredito que pode gerar melhoria mais efetiva, na qualidade e no tempo da correção dos defeitos, ainda na execução dos testes.

Questão 15: Você teria sugestões de melhoria para as fichas?

Resposta: Como sugestão de melhoria, acrescentaria apenas um cabeçalho por área da ficha, com uma breve descrição do objetivo da Seção que será analisada.

Participante 3 – Avaliou as fichas dos indicadores DD, TCEX, TCT.

Questão 14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando o esforço para coleta e análise dos dados?

Resposta: Acredito que os 3 indicadores apresentados seriam facilmente utilizados em conjunto, proporcionando uma ótima ferramenta para aferição da qualidade do software e possíveis tomadas de decisão.

Questão 15: Você teria sugestões de melhoria para as fichas?

Resposta: O modelo da ficha atende perfeitamente.

Participante 4 – Avaliou as fichas dos indicadores EGT, ICT, RR.

Questão 14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando o esforço para coleta e análise dos dados?

Resposta: RR, esse indicador pode ser utilizado para monitorar e indicar se os testes vão finalizar dentro do previsto. Não apresenta dificuldade para a coleta de dados, desde que a empresa tenha ferramentas para o acompanhamento da execução e correção de defeitos.

Questão 15: Você teria sugestões de melhoria para as fichas?

Resposta: Na ficha RR, poderia considerar colocar a informação do previsto ou planejado para a execução dos testes para confrontar com os dados realizados que foram coletados.

Participante 5 – Avaliou as fichas dos indicadores EGCD, QDRC, TCTA.

Questão 14: Quais indicadores do conjunto apresentado são mais interessantes na sua opinião para uma melhor gestão dos testes considerando o esforço para coleta e análise dos dados?

Resposta: EGCD - Esforço gasto para correção de defeitos.

Questão 15: Você teria sugestões de melhoria para as fichas?

TCTA - No cabeçalho, no campo de Categorias, a “automação dos testes” que propõe o indicador ao meu ver está relacionado a **produtividade e tempo** além da qualidade.

TCTA - Procedimento de Análise Baseado em Critérios - Não está falando sobre o **desenvolvimento da automação**, creio que a análise seja feita junto a estratégia, e enquanto estiverem fazendo o plano já estarão desenvolvendo os casos de testes automatizados, que dão um trabalho maior de fazer.

TCTA - Procedimento de Análise Baseado em Relacionamentos - Não vi o relacionamento com algo para o **Reteste do Defeito** que possivelmente será feito de forma mais rápida caso o CT esteja automatizado.

QDRC – Procedimento de Análise Baseado em Critérios - Sugiro rever as possíveis conclusões no ID 3, pois para a faixa de 50% quando o SLAD for 10 não corresponde.

No item ID 4, nas possíveis ações, sugiro também adicionar ação onde o líder de testes deve rever com a equipe que realizou os testes porque foi o defeito não foi reportado durante a execução, em casos de não ter defeitos pendentes.

No gráfico achei estranho o SLA estar variando a cada iteração, geralmente é um valor fixo mesmo para projetos com métodos ágeis.

4.3.5 Coleta do tempo de análise das fichas

Foi coletado também por cada participante o tempo gasto na avaliação das fichas de forma que possa perceber o esforço de tempo considerando uma primeira iteração dos respondentes com as fichas de indicadores. A Tabela 25 mostra o tempo gasto pelos participantes na análise de cada parte da ficha de registro de indicadores, também é mostrado o tempo médio de análise de cada quadro da ficha e a média de tempo de análise completa das fichas.

Com base nos tempos coletados percebe-se que cada participante levou em média aproximadamente 14 minutos para análise completa de cada ficha, não se pode afirmar se esse tempo é bom ou ruim em termos de análise, porém o participante 3 comentou que acredita que com maior utilização das fichas e dos indicadores os tempos tendem a ser menores tendo em vista maior conhecimento dos procedimentos das fichas.

Observa-se também que o tempo de análise de cada quadro de informação da ficha foi bem aproximado para todos os participantes e em todas as fichas, ficando entre 2 e 6 minutos. A coleta de tempo não representa um ponto de alta relevância para a pesquisa, apenas pretende-se avaliar se o tempo gasto na avaliação das fichas poderia representar um ponto impeditivo para os projetos.

Tabela 25 - Tempo gasto na análise das fichas

Indicadores	Tempo para análise do cabeçalho e gráfico	Tempo para análise medição e coleta	Tempo para procedimentos baseados em critério	Tempo para procedimentos baseado em relacionamento	Tempo total ficha	ID Participante
DD - Densidade de defeitos	00:03:00	00:03:00	00:03:00	00:03:00	00:12:00	P3
EGCD - Esforço gasto para correção de defeitos	00:03:00	00:04:00	00:02:00	00:03:00	00:12:00	P5
EGT - Esforço gasto nos testes	00:02:00	00:05:00	00:06:00	00:05:00	00:18:00	P4
ICT - Índice de Cobertura de Testes	00:02:00	00:05:00	00:06:00	00:05:00	00:18:00	P4
QDNC - Quantidade de defeitos não cobertos por casos de teste	00:01:00	00:02:00	00:02:00	00:02:00	00:07:00	P1
QDRC - Quantidade de defeitos reportados pelo cliente	00:03:00	00:03:00	00:05:00	00:04:00	00:15:00	P5
RR - Prontidão para release	00:02:00	00:05:00	00:06:00	00:05:00	00:18:00	P4
TACT - Taxa de aumento de casos de testes entre releases	00:02:00	00:02:00	00:02:00	00:02:00	00:08:00	P1
TCEX - Taxa de casos de testes executados em relação ao planejado	00:03:00	00:03:00	00:03:00	00:03:00	00:12:00	P3
TCT - Tempo para correção dos testes	00:03:00	00:03:00	00:03:00	00:03:00	00:12:00	P3
TCTA - Taxa de casos de testes automatizados	00:02:00	00:03:00	00:04:00	00:04:00	00:13:00	P5
TD - Taxa de Defeitos	00:05:00	00:03:00	00:05:00	00:05:00	00:18:00	P2
TDD - Tempo entre detecção de defeitos pelo cliente	00:05:00	00:04:00	00:04:00	00:05:00	00:18:00	P2
TFTA - Taxa de funcionalidades que passam nos testes de aceitação automatizados	00:06:00	00:04:00	00:05:00	00:06:00	00:21:00	P2
TMCD - Tempo médio para correção de um defeito	00:02:00	00:04:00	00:03:00	00:03:00	00:12:00	P1
Tempo total do quadro----->	00:44:00	00:53:00	00:59:00	00:58:00	03:34:00	
Média de tempo gasto para cada quadrante da ficha----->	00:02:56	00:03:32	00:03:56	00:03:52		
Média de tempo para análise completa de cada ficha ----->					00:14:16	

4.3.6 Coleta das medidas e geração dos gráficos dos indicadores

Após análise das fichas, cada participante escolheu um indicador da lista apresentada para coletar as medidas de projetos em que tivesse participado ou ainda participando e com apoio das planilhas de medição calcular o indicador e gerar o gráfico de análise. Também foi solicitado que ao final da coleta e cálculo do indicador que o participante analisasse os procedimentos baseados em critério e relacionamento do indicador e fizesse comentários a respeito do resultado apurado.

Desta forma os seguintes indicadores tiveram as medidas coletadas e gráficos gerados conforme código do participante: P1 – TACT , P2 – TD, P3 TCEX e P4 – RR. O participante P5 alegou que não tinha informações de projeto com medidas possíveis que pudessem ser coletadas, com isso o mesmo não contribui com esta parte do estudo.

Participante P1 – Indicador TACT (taxa de aumento de casos de testes entre releases).

As medidas QTCT_ATUAL (Quantidade de casos de testes implementados na release atual) e QTCT_ANTERIOR (Quantidade de casos de testes implementados na release anterior) foram coletados com base em histórico do projeto e o gráfico mostrado na Figura 32 foi gerado. Este gráfico mostra a evolução do indicador de 4 releases finalizadas.

Foi questionado ao participante qual seria a análise possível para o gráfico gerado, com relação à análise do indicador com base em critérios nada fora do normal foi observado.

Porém, com relação ao relacionamento baseado nos relacionamentos, o participante observou que neste projeto não houve análise do crescimento dos casos de teste entre as release, e ocorreu impacto na execução por conta do crescimento de casos de teste não ter sido analisado o esforço. Com a utilização da ficha, seria possível analisar junto com o indicador EGT e executar alguma ação prévia.

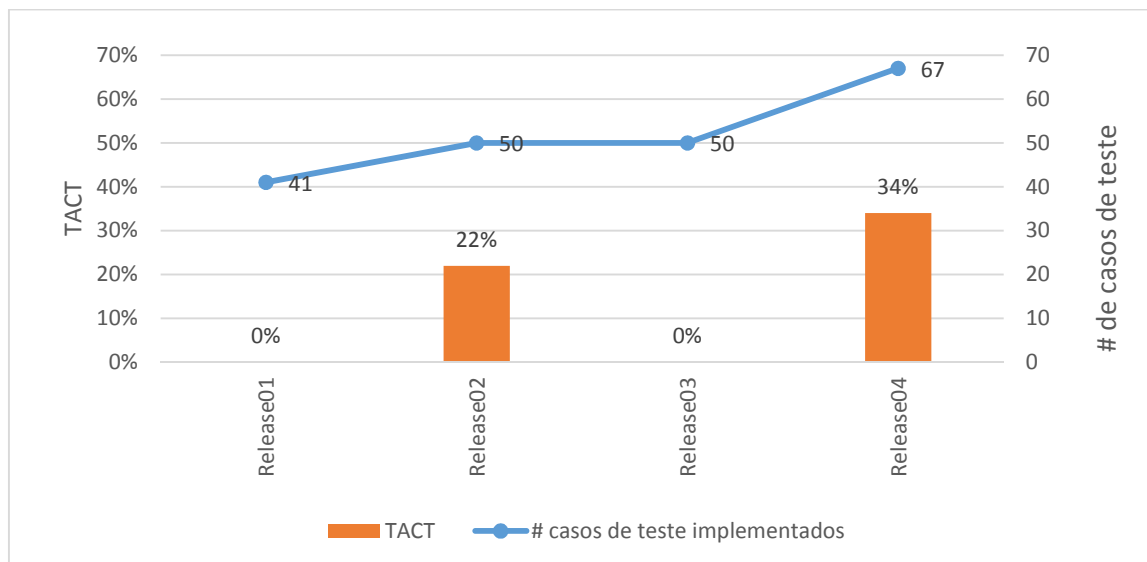


Figura 32 - Gráfico estudo de caso TACT

Participante P2 – Indicador TD (taxa de defeitos).

As medidas QDA (Quantidade de defeitos abertos) e QDANR (Quantidade de defeitos analisados e não removidos) e QDR (Quantidade de defeitos removidos) foram coletados com base em histórico do projeto e o gráfico mostrado na Figura 33 foi gerado. Este gráfico mostra a evolução do indicador de 3 iterações finalizadas mais 1 iteração em andamento (agosto).

Foi questionado ao participante qual seria a análise possível para o gráfico gerado, com relação à análise do indicador com base em critérios o participante observou que a taxa de defeitos abertos + analisados e não removidos ficou igual a 25% e que a ação tomada foi atuar junto ao gerente do projeto para priorizar a correção dos defeitos conforme recomendação da ficha. Com relação ao relacionamento, o EGT tem ligação direta, pois será necessário mais tempo para reteste dos defeitos. Também há relacionamento com RR, pois há risco de não finalizar na semana planejada que é a última semana da release.

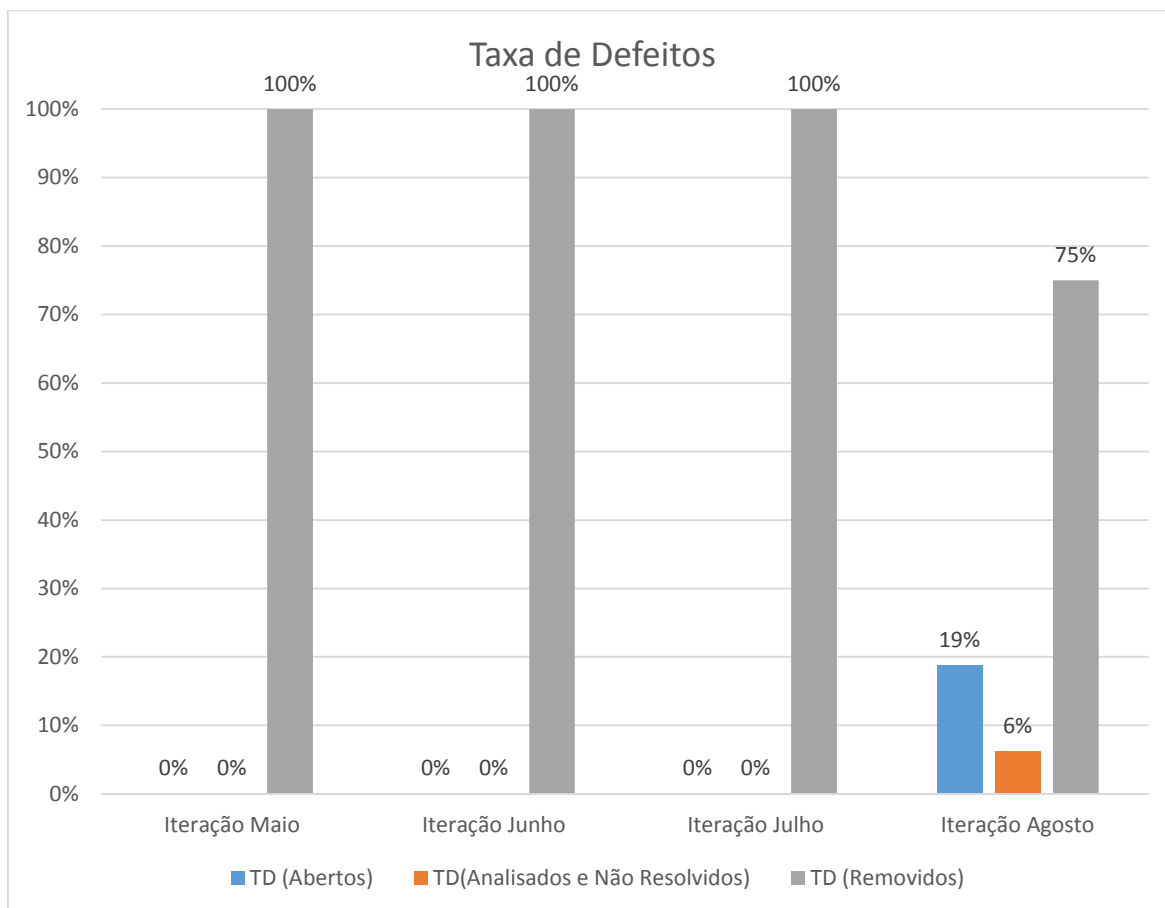


Figura 33 - Gráfico estudo de caso TD

Participante P3 – Indicador TCEX (taxa de casos de testes executados).

As medidas QTCTD (Quantidade de casos de testes planejados) e QTEXS (Quantidade de casos de testes executados com sucesso) e QTEXF (Quantidade de casos de testes com falha) foram coletados com em dados de projeto em andamento e o gráfico mostrado na Figura 34 foi gerado. Este gráfico mostra a evolução do indicador de 3 iterações.

Com relação a análise do indicador com base em critérios o participante observou que a taxa de casos de testes executados com sucesso atingiu 95% e que este indicador mostra que execução está acontecendo sem problemas. Com relação aos relacionamentos, a informação da ficha mostra a importância da automação dos casos de testes. Sendo assim, seria importante olhar para as ações de relacionamento do indicador TCTA, mesmo sabendo que no projeto atual não há possibilidade de realizar testes automatizados, pois não há contratação para esta modalidade.

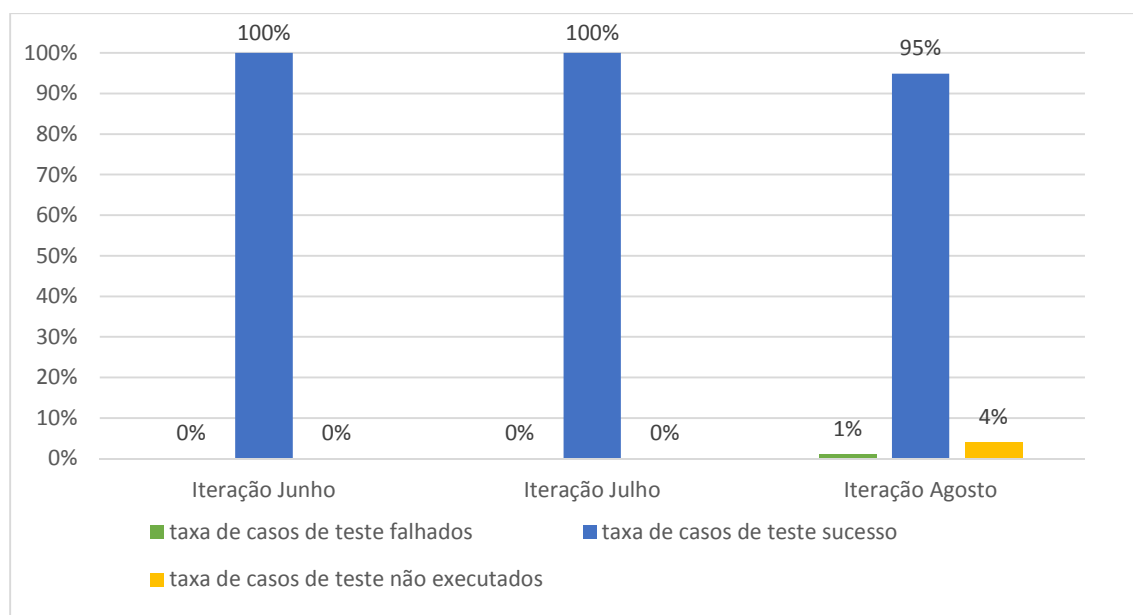


Figura 34 - Gráfico estudo de caso TCEX

Participante P4 – Indicador RR (prontidão para release).

As medidas QTDF (Quantidade de defeitos identificados), QTDFR (Quantidade de defeitos removidos), QTCTEX (Quantidade de casos de testes executados) e QTCTS (Quantidade de casos de teste executados com sucesso) foram coletadas em dados de projeto em andamento e o gráfico mostrado na Figura 35 foi gerado. Este gráfico mostra a evolução do semanalmente durante uma iteração do projeto.

Com relação à análise do indicador com base em critérios o participante observou que com a utilização do indicador fica claro que a data de entrega da release não será cumprida na semana 5. Também observou que as ações recomendadas de automação de casos de testes, que não ocorreram durante a iteração, além da maior cobrança do gerente de projetos na correção dos defeitos poderia ter ajudado a cumprir o planejado. Com relação aos relacionamentos, a informação da ficha mostra a importância da automação dos casos de testes, sendo assim seria importante olhar para as ações de relacionamento do indicador TCTA, além do EGCD para garantir agilidade na correção dos defeitos. No gráfico da Figura 35 é possível observar a inclusão da coluna **RR – planejado** que significa a quantidade de semanas de acordo com o planejamento da release, que seria a quantidade de semanas restante para a conclusão. Esta informação foi incluída com base na avaliação de um dos participantes durante o estudo de caso.

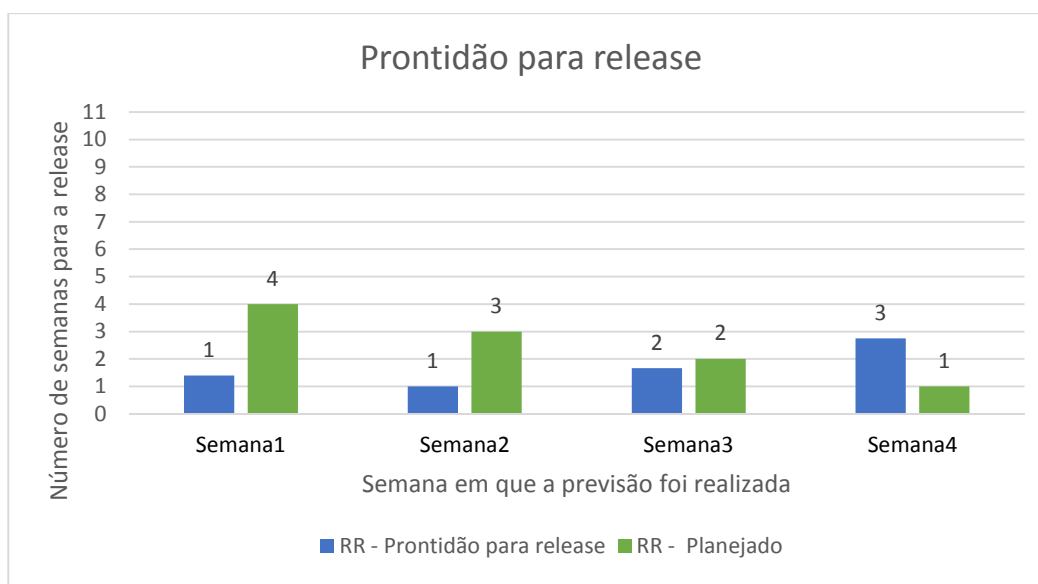


Figura 35 - Gráfico estudo de caso RR

4.4 Análise dos resultados

A avaliação indica uma grande aceitação pelos participantes da organização em relação aos indicadores propostos. Do total de 65 respostas obtidas dos participantes, 44 foram para a métrica 1, que significava “concordo totalmente”, onde as perguntas realizadas indicavam ao participante a percepção de facilidade no entendimento das informações e formato da ficha, além de confirmar utilidade de implementação das fichas, melhorias no desempenho e apoio no gerenciamento dos testes e interesse em utilizar as fichas dentro da organização do participante. Para a resposta 2 “concordo parcialmente” houve 18 respostas, e 3 respostas para a métrica 3 “Indiferente”, não houve respostas para as métricas 4 ou 5. As métricas para as respostas são mostradas na Tabela 21.

Para as 3 respostas “indiferentes”, não é possível afirmar que haja risco aos resultados de forma geral, uma vez que foram concentrados em apenas 2 perguntas.

Para os objetivos de facilidade e utilidade foi identificada a percepção de grande facilidade de entendimento das fichas e de seus conteúdos. Também percebeu-se a grande utilidade dos indicadores dentro da organização no dia-a-dia dos participantes e de suas tarefas.

Para o objetivo de interesse em usar as fichas é percebida uma pequena dificuldade de se viabilizar a utilização dos indicadores nos projetos da organização, talvez por alguma resistência de mudança nos procedimentos de coleta e análise de medidas dentro da

organização ou por falta de um apoio gerencial de nível superior para adoção deste novo processo de coleta e análise.

Como pontos fortes é notado pela avaliação que revelou 95% das respostas como valores “1” ou “2”, que os 3 objetivos determinados foram alcançados revelando facilidade, utilidade e interesse de todo o conteúdo das fichas de indicadores.

Como ponto fraco, pode-se indicar uma pequena dificuldade de alguns participantes pois as sugestões revelaram alguma confusão na análise dos procedimentos de relacionamentos. Uma possível explicação para isso, pode ser a tentativa de comparação com os procedimentos usados atualmente pelos participantes ou pela falta deles.

Com relação às questões objetivas, foi possível notar interesse dos participantes na melhoria dos indicadores tendo em vista os comentários e também a percepção de preferência por alguns indicadores que possivelmente estão ligados ao cotidiano dos participantes, como ao monitoramento de defeitos.

Com relação à questão 15, para a sugestão do participante 2, o pesquisador acredita não ser necessário incluir cabeçalho de informações, pois considerando a utilização constante das fichas e dos processos, uma vez após entendimento nas primeiras iterações o usuário da ficha terá esclarecimento pleno de todas as partes da ficha. Além disso, o template não dispõe de muito espaço para adição destas informações. Esta justificativa foi conversada com o participante que entendeu ser pertinente.

Para a sugestão do participante 4, foi incluída a informação da quantidade de semanas planejadas para finalização da release semanalmente, desta forma é possível comparar o RR atual com o RR planejado.

Para o participante 5, com relação ao SLA do indicador QDRD foi explicado que o valor variável foi apenas uma sugestão com base em relatos, mas que pode ser um SLA fixo adaptado de acordo com a necessidade do projeto ou organização. Com isso, essa informação não foi alterada na ficha neste momento. Foi revisado e melhorado o texto das premissas e conclusões do ID 3 e 4 na parte de análise de critérios. No TCTA, foi atualizada a categoria para considerar Produtividade além de qualidade. Com relação ao tempo, após nova explicação, o participante entendeu que não se aplicava realmente. Com relação à sugestão de citar o desenvolvimento da automação na análise baseada em critério, foi esclarecido que o plano de análise citado nas premissas 3 e 4 contempla essa parte de desenvolvimento da automação. Para o comentário relacionado ao “reteste dos defeitos”, o relacionamento com

o EGT citado na ficha está cobrindo esta parte.

O tempo de análise das fichas é um tempo baixo considerando que foi a primeira iteração dos participantes com o modelo de ficha apresentado e com o conteúdo de análise dos indicadores.

4.5 Ameças à validade e Limitações

Este estudo de caso possui algumas ameaças e limitações que podem afetar a validade dos resultados. Nesta Seção, essas ameaças são analisadas a partir das diretrizes propostas por RUNESON *et al.* (2012).

- **Ameaças à validade de *constructo*:** são eventos que questionam se os objetos e participantes do estudo estão aderentes com a questão que está sendo abordada.
 - a) Os indicadores sugeridos na pesquisa foram criados com base no conhecimento adquirido pelo pesquisador, desta forma havia o risco da dificuldade de avaliação de alguns indicadores pelos participantes pelo fato deles não executarem ou utilizarem algumas medidas propostas. Para minimizar este risco, durante a sessão de capacitação para avaliação das fichas dos indicadores, foi reforçado aos participantes o objetivo de analisar o indicador com foco nas ações, recomendações, fórmula de coleta e cálculo da medição e relacionamentos propostos.
- **Ameaças à validade interna:** são eventos não controlados pelo pesquisador que podem distorcer o resultado esperado. Neste estudo foram observadas as seguintes ameaças internas:
 - a) Cada ficha foi analisada apenas uma única vez durante o estudo. Com isso, não pode-se afirmar que as avaliações realizadas são suficientes para confirmação de utilidade das fichas, seria necessário outras rodadas de avaliação com os participantes avaliando todas as fichas, isto pode causar uma impressão errada quanto aos resultados positivos apurados através da avaliação.
 - b) A análise dos relacionamentos não foi realizada por completo, devido à limitação de apenas 3 fichas terem sido analisadas por cada participante. Os comentários relativos a esta parte da ficha foram baseados apenas nas informações descritas na parte de relacionamento da ficha principal, não houve avaliação da ficha dos indicadores relacionados para uma avaliação mais completa dos

relacionamentos. Assim não se pode afirmar que os relacionamentos são pertinentes ou necessitam de ajustes quanto as sugestões de ações.

- **Ameaças à validade externa:** são eventos externos que questionam a capacidade de generalizar os resultados do estudo para uma população maior que os participantes. Neste estudo foram observadas as seguintes ameaças externas.
 - a) O estudo envolveu apenas uma única organização. Desta forma seu resultado não é plenamente conclusivo, uma vez que pode significar que o template se aplica ao cenário desta organização e não à realidade de outras organizações que realizam testes de software. Para contornar esta limitação será necessário realizar este estudo, ou uma evolução, em um grupo mais abrangente de organizações.
- **Ameaças à confiabilidade:** são aspectos que se preocupam com o quão os dados e a análise dependem de pesquisadores específicos
 - a) Com relação à coleta das medidas, a impossibilidade de coleta de todas as medidas que compoem as fichas, não permitiu avaliar as planilhas de apoio para todos os indicadores nem avaliar se as premissas e conclusões na análise baseada em critérios são realmente pertinentes quando analisados dados reais de projeto, durante a monitoração dos testes.
 - b) As medidas coletadas foram na sua maioria oriundas de projetos já finalizados. Desta forma, apesar do feedback apresentado pelo participantes, não se pode afirmar que as ações sugeridas foram realmente executadas.

4.6 Considerações finais

Neste capítulo foi apresentada a avaliação do conjunto de fichas de registro de indicadores em uma organização que possui profissionais que atuam na gestão de projetos de testes. Os participantes puderam analisar o conteúdo das fichas e responder ao questionário com suas percepções.

Com base nessa percepção, pode-se supor que a aplicação e utilização dos indicadores poderá trazer benefícios para as organizações. Porém a utilização de todo o conjunto proposto pode não ser possível. Dessa forma, sugere-se a seleção daqueles que sejam mais indicados para a realidade de cada organização. Além disso, os procedimentos de coleta e os critérios de análise apresentados devem ser adaptados também à realidade de cada organização.

5 Conclusão

5.1 Considerações Finais

Medição de software é uma disciplina considerada importante para que as organizações de desenvolvimento de software possam ter as informações necessárias para tomar decisões no contexto organizacional e dos projetos. No entanto, medir também é considerada uma atividade não trivial e custosa. Contribuem para isso a falta de padronização dos termos utilizados na medição de software, a existência de muitas fontes que proveem informações sobre indicadores que podem ser utilizados em organizações de software e a pouca profundidade das informações sobre os relacionamentos existentes entre os indicadores, assim como detalhamento de coleta e medição de indicadores.

Considerando o contexto apresentado, esta dissertação propôs uma lista de indicadores de testes utilizando um template para registro dos indicadores, informações de coleta e medição, e análise dos indicadores baseados em critérios e seus relacionamentos. O template utilizado é composto por mapas que exploram os relacionamentos entre indicadores, bem como destes com as necessidades de informação, objetivos estratégicos e com os níveis/processos do MR-MPS-SW. Para avaliar o conteúdo das fichas foi realizado um estudo de caso em uma organização que executa testes de software com profissionais que atuam na gestão de projetos de teste. Pela análise das respostas dos participantes é notado que as fichas são fáceis de serem entendidas, demonstram ser úteis para apoio no gerenciamento e os participantes mostram interesse em usar as fichas.

Notou-se também que o tempo de análise da ficha não é longo, e não deve representar esforço adicional para o líder de testes. Para a adoção das fichas de indicadores sugeridas nesta pesquisa no dia-a-dia dos projetos de testes, não se faz necessário um processo complexo de operacionalização. Entende-se que basta os responsáveis pelos projetos indicarem a utilização das planilhas de apoio a coleta e geração dos indicadores e as fichas sirvam de apoio na análise e ações a serem tomadas. Caso haja necessidade de alterações, as fichas podem ser customizadas para atender novos parametros de premissas da organização.

Espera-se que o uso das fichas de indicadores definidos possa apoiar na gestão de projetos de testes de forma que líderes de testes e gerentes de projeto possam monitorar as medidas e indicadores e utilizar as ações sugeridas para embasar tomada de ações dentro dos projetos e junto as equipes de desenvolvimento de software.

5.2 Contribuições

O mapeamento sistemático da literatura produzido nesta pesquisa é uma das contribuições desta pesquisa, pois foi possível identificar e mapear medidas relacionadas a testes utilizadas em projetos de desenvolvimento ágil, estas medidas poderão servir de apoio em trabalhos futuros.

A principal contribuição deste trabalho foi a criação do conjunto de indicadores relacionados com os objetivos e necessidade de informação. Este conjunto de indicadores disponibiliza informações de coleta, análise baseada em critérios e baseada em relacionamento entre indicadores para que gestores possam selecionar alguns indicadores para utilizar em seus projetos desde as fases iniciais do projeto de teste e principalmente durante a execução da release ou iteração.

As fichas podem ser adaptadas a outras organizações com base em suas necessidades, pode ser considerado como ponto inicial para implementação de um plano de medição de entendimento mais simples e direto por conta da apresentação em modelo de fichas. As planilhas de coleta de medidas e geração dos gráficos também contribuem de forma a proporcionar um cálculo automático dos indicadores e dos gráficos após inserção das medidas necessárias.

5.3 Limitações

O mapeamento sistemático da literatura não considerar outras bases de pesquisa, além das citadas no mapeamento Ciclo 1. Tendo em vista que parte do trabalho apresentado foi a replicação do mapeamento de Mota e Dias-Neto (2014) (Ciclo 1), apenas as duas bases de pesquisa utilizadas foram consideradas no Ciclo 2, com isso a ameaça seria de que realizando a busca em outras bases de pesquisa, poderíamos ter identificados medidas que não foram mapeadas nos Ciclos 1 e 2.

Há possibilidade de ter mapeado medidas utilizadas em projetos tradicionais que sejam aplicáveis em projetos ágeis e não terem sido identificados, devido o foco da busca

ter sido em projetos ágeis, porém com base no conjunto de medidas identificadas observa-se um conjunto extenso de medidas que são aplicáveis a projetos “tradicionais” e “ágeis”, minimizando levemente este risco da pesquisa.

Grande parte dos artigos não detalham o que são as medidas, nem como calculá-las, por isso, muitas vezes, a interpretação dependeu do pesquisador que executou o mapeamento sistemático. Para tratar essa limitação, a definição final das medidas (referente à questão de pesquisa RQ1) foi revista por outro pesquisador e a definição dos indicadores (referente à questão de pesquisa RQ2) foi revista por dois pesquisadores com conhecimento em Engenharia de Software.

Apesar de as fichas de indicadores apresentadas no capítulo 3 terem sido baseadas em um mapeamento sistemático da literatura e tenha sido avaliada em uma organização que atua com testes de software em projetos que utilizam metodologia de desenvolvimento ágil, as mesmas não foram avaliadas em larga escala na indústria. Desta forma, é possível que as fichas apresentadas não sejam suficientes para apoio total à gestão de projetos de testes em outras organizações, para isto seria necessário adaptação dos indicadores para outras organizações.

Os valores e limites definidos para cada premissa na análise baseada em critérios foram definidos no contexto da pesquisa com base na experiência do pesquisador. Esses valores podem não atender a determinados projetos, porém para minimizar esta limitação os limites e faixas de análise foram avaliados por 2 especialistas em engenharia de software.

A análise de cada ficha de indicador teve apenas uma única avaliação, desta forma é possível que outros ajustes sejam necessários com base em revisões adicionais.

As fichas foram distribuídas para análise de forma aleatória para cada participante, desta forma não foi possível explorado alguma experiência prévia do participante com as medidas do indicador que avaliou, pois também não era alvo desta pesquisa.

Na fase de coleta de medidas e geração dos gráficos, apenas 4 indicadores tiveram medidas reais coletadas e os gráficos gerados, isto ocorreu pelo fato da organização não realizar coleta de todas as medidas que seriam necessárias para todas as quinze fichas de indicadores e pelo tempo limitado de dedicação de cada participante no estudo.

A maioria das medidas coletadas são de projetos já finalizados, as informações da ficha são indicadas para apoio durante as fases de teste, pois desta forma as ações e análise podem ser consideradas imediatamente em tempo de projeto ou iteração.

A coleta foi realizada em uma mesma organização, talvez a aplicação em outra organização contribuísssem para refinamento das instruções e identificação de outros relacionamentos.

A avaliação das fichas não ter sido realizada em ambiente controlado, poder ter gerado espaço para que durante a análise dos indicadores o participante tenha sido interrompido por fatores externos fazendo com que ele não dedicasse uma maior atenção na avaliação dos procedimentos.

5.4 Perspectivas Futuras

Alguns trabalhos futuros possíveis no contexto desta dissertação são:

- Avaliação das fichas de registro indicadores por um número maior de especialistas em medição para que possa aferir uma melhor escala de análise das premissas e ações.
- Avaliar a possibilidade de adicionar indicadores diferentes no mesmo gráfico e criar uma ficha composta de indicadores aumentando o nível de informação do gráfico.
- Avaliar as fichas em larga escala na indústria para atestar quais indicadores podem apoiar os projetos de testes de forma mais ampla e quais podem ser adaptados ao contexto da organização.
- Investigar outras bases científicas para identificação de possíveis medidas não citadas nesta pesquisa.
- Avaliar se para as medidas que foram excluídas na criação dos indicadores por serem consideradas subjetivas, se não poderiam ser adaptadas para algum indicador, fazendo algumas inferências para determinar como calcular o indicador e ações.
- Investigar uma forma de evitar que o mapa de relacionamentos entre indicadores tenha uma leitura difícil em casos de muitos relacionamentos a serem representados.
- Investigação de novos indicadores e relacionamentos.
- Criar um método de apoio para construção da ficha considerando uma avaliação dos impactos entre indicadores.
- Criar uma planilha única de coleta de medidas, ou consolidar o maior número possíveis de medidas para apoiar a geração dos gráficos e indicadores.

- Considerar a coleta das medidas a partir de ferramentas automatizadas de forma que as planilhas possam ser integradas dentro um ambiente como DevOps, ou ferramentas de apoio como, por exemplo, ⁴Sonarqube.
- Criar algum mecanismo que auxilie ao gerente de projeto na escolha de quais indicadores serão utilizados na criação de uma plano de medição.

⁴ <https://www.sonarqube.org/>

Referências

- ABRAHAMSSON, P., OUTI, S., RONKAINEN, J., WARSTA, J., (2002), “Agile Software Development Methods”, Espoo 2002, VTT publications
- ABRANTES, JOSE,F., (2012) “Estudos Experimentais sobre Agilidade no Desenvolvimento de Software e sua Utilização no Processo de Teste”, Tese de Doutorado, UFRJ, COOPE, Rio de Janeiro – Brasil.
- AGILE MANIFESTO (2001) Available at <http://agilemanifesto.org>.
- BARCELLOS, M., (2009) “Uma Estratégia para Medição de Software e Avaliação de Bases de Medidas para Controle Estatístico de Processos de Software em Organizações de alta maturidade”, Tese de Doutorado, Pós-Graduação de Engenharia de Sistemas e Computação, COOPE – UFRJ, Rio de Janeiro
- BASILI,V., ROMBACH,H., (1994) “Measurement, In: Encyclopedia of Software Engineering, v.1.
- BASILI, V., ROMBACH, H., (1988) “The Tame Project: Towards Improvement-Oriented Software Environments”, IEEE Transactions on Software Engineering”, v.14, n. 6, pp. 758-773.
- BASS, L., (1999) “Constructing Superior Software; Applying Proven Practices”, New Riders Publishing, Thousand Oaks, CA, USA.
- BECK, K., ANDRES, C., (2004) “Extreme programming explained: Embrace change. 2nd ed. Boston, MA, USA: Addison-Wesley.
- BERNARDO, P. (2008) “A importância dos Testes Automatizados” Artigo publicado na Engenharia de Software Magazine, pp.54-57.
- BIOCHINI, J., MIAN, P.G., NATALI, A.C.C., et al., (2005), “Systematic review in software engineering”, Programa de Engenharia de Sistema e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- BONELLI, S., Santos, G., Barcellos, M., (2017) “ASM.br: A Template for Specifying Indicators, Universidade Federal do Estado do Rio de Janeiro - UNIRIO, Rio de Janeiro, Brazil.
- BONELLI, S., (2014) “Uma abordagem para registro de indicadores de medição em organizações de desenvolvimento de software”, Dissertação de Mestrado, Universidade Federal do Estado do Rio de Janeiro - UNIRIO, Rio de Janeiro, Brazil.
- BORTOLUCI, R., DUDUCHI, M., (2015) “Indicadores para teste de software em desenvolvimento ágil”, X workshop de pós-graduação e pesquisa Centro Paula Souza, São Paulo, Brazil.
- BRAUN, L.,(2007) “ Modelo Para Medição do ROI em Processos de Teste de Software. 2007. “Centro Universitário Feevale, Instituto de Ciências Exatas e Tecnológicas, Novo Hamburgo – Brasil.
- CMMI Institute, (2011), SEI - CMMI for Development, Versão 1.3, Pittsburg, Software Engineering Institute, <https://www.sei.cmu.edu/cmmi/>

- COHN, M. (2011) “Desenvolvimento de Software com Scrum”, Editado por Aldir Jose Coelho Correa da Silva, Porto Alegre, Bookman.
- CONTE, T., 2009, Técnica de Inspeção de Usabilidade Baseada em Perspectivas de Projeto WEB. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- COSTINHAS, C., SANTOS, G. (2014) “Adoção de Práticas Ágeis em um Ambiente Tradicional: Um Estudo de Caso” In: XIII Simpósio Brasileiro de Qualidade de Software, pp. 239-246, Blumenau, Santa Catarina, Brasil.
- CRISPIN, L., GREGORY, J., (2009) “Agile testing: a practical guide for testes and agile teams”, Addison-Wesley Professional
- FARROHA, B.S., (2014), “A framework for managing mission needs, compliance and trust in the DevOps environment”, IEEE Military Communications Conference”, 2014.
- GOTTESHEIM, W., (2015), “Challenges, Benefits and Best Practices of Performance Focused DevOps”, Dynatrace Center of Excellence, 2015, Austria.
- HENDRICKSON, E. (2004) “Agility for Testers” In PNSQC - Pacific Northwest Software Quality Conference, October.
- ISO/IEC,15939 (2002), “Software Engineering – Software Measurement Process. International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.
- JANUS, A., SCHMIETENDORF, A., DUMKE, R., JAGER, J.(2012) “The 3C approach for Agile Quality Assurance”
- JONES, C. et al.(2003) “ Practical Software and Systems Measurement: A Foundation for Objective Project Management”. Department of Defense and US Army, USA.
- KAYES, I., SARKER, M., CHAKARESKEI, J., (2016). " Product backlog rating: a case study on measuring test quality in scrum." Computer Science and Engineering, University of South Florida, Tampa, FL, USA
- KITCHENHAM, B., CHARTERS, S., (2007), “Guidelines for Performing Systematic Literature Reviews in Software Engineering”, Tech. rep., Technical report, EBSE Technical Report EBSE-2007-01.
- KUPIAINEN, E., MANTYLÄ, M.V., ITKONEN, J., (2015). " Using metrics in Agile and Lean software development - A systematic literature review of industrial studies" Department of Computer Science and Engineering, Aalto University, Finland
- LARMAN,C.,(2003) “Agile and iterative development: A manager’s guide. Pearson Education.
- LEAL, I. (2009) “Requisitos de Metodologias de Testes de Software para Processos Ágeis”, dissertação de conclusão de curso – Departamento de Ciência e Computação – Universidade Federal de Minas Gerais, Belo Horizonte, Brasil
- LIMA, C.G.B., (2013) “Procedimento para Introdução de Agilidade em Testes de Software”, UFRJ – COOPE – Rio de Janeiro.
- LUECKE, R., (2010) “Estratégia, Harvard Business Essentials, Editora Record.
- MAIA, J. R. C., (2009) Use Métricas Adequadas. Garanta a Qualidade de Projeto Orientado a Objeto. 7 f. Euax – Gestão de Processos.

- MARTINS, J.C.C., (2007). “Técnicas para gerenciamento de projetos de software”. Rio de Janeiro: Brasport, 2007.
- MCGARRY, J., CARD, D., LAYMAN, B., CLARCK, E., DEAN, J., HALL, F., (2012) “Practical Software Measurement: objective information for decision makers. Prentice Hall
- MEYER, B. (2014) “Agile! The Good, the Hype and the Ugly”, Editora Springer, Zurich, Switzerland.
- MOTA, T., DIAS-NETO, A., (2014) “Mapeamento Sistemático sobre métricas no Contexto de Métodos Ágeis aplicadas a testes de software”, Universidade Federal do Amazonas, Manaus, Brazil.
- OHNO, T. (1997) “O Sistema Toyota de Produção: além da produção em larga escala”, In Bookman, Porto Alegre, Brazil.
- PETERSEN, K., VAKKALANKA, S., KUZNIARZ, L., (2015) “Guidelines for conducting systematic mapping studies in software engineering: An update”, Department of Software Engineering, Blekinge Institute of Technology, Sweden
- PETERSEN, K., FELDT, R., MUJTABA, S., MATTSSON, M., (2008), “Systematic mapping studies in software engineering”, in: 12th International Conference on Evaluation and Assessment in Software Engineering, vol. 17, p. 1.
- POOJA, J., PATNAIK, K.S, (2016), “Applicability of Process Capability Analysis in Metrifying Quality of Software”, Department of CSE, BIT, Mesra, Ranchi, 2016, India.
- POPPENDIECK, M.; POPPENDIECK, T., (2003) “Lean software development: An Agile toolkit”. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- RATHOD, N. (2015) “Test Orchestration, A framework for Continuous Integration and Continuous Deployment”, 2015 International Conference on Pervasive Computing (ICPC).
- ROCHA, A. R., BARCELLOS, M. P. and SANTOS, G. (2012) “Medição de Software e Controle Estatístico de Processos”, Ministério da Ciência, Tecnologia e Inovação - SEPIN - PBQP Software.
- ROCHE, J. (2013) “Adopting DevOps Practices in Quality Assurance”, 2013 Communications of the ACM Nov2013.
- RUNESON, P., HOST, M., RAINER, A., REGNELL, B., (2012) “Case Study Research in Software Engineering – Guidelines and Example”, John Wiley & Sons Inc.
- SANTOS, G., (2008) “Ambientes de Engenharia de Software Orientados a Corporação”, Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro.
- SILVA, F., GALARRAGA, O., (2013) “Proposta de Planejamento de Medições em Projetos de Software Utilizando um Ferramenta de Modelagem”, UNISINOS – São Leopoldo – Rio Grande do Sul – Brasil.
- SOFTEX, (2016), Guia Geral MPS de Software, SOFTEX - Associação para Promoção da Excelência do Software Brasileiro. Disponível em: <http://www.softex.br/mpsbr/guia-geral-de-software>.
- SCHWABER, K., (1995) The scrum development process. Proceedings of the 10th Annual

- ACM Conference on Object Oriented Programming Systems, Languages, and Applications, Austin, Texas, USA.
- SHARMA, S., COYNE, B., (2015) “DevOps for Dummies”, 2nd IBM limited edition.
- SILVA FILHO, R. C., 2006, Uma Abordagem para Avaliação de Propostas de Melhoria em Processos de Software. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- SOARES, H.F., (2015) “Estudos sobre o comportamento da dívida técnica de documentação ao se trabalhar com requisitos ágeis” UNIFACS – Salvador , Bahia Brasil.
- STARON M., MEDING, W., PALM, K. (2015). " Release readiness indicator for mature agile and lean software development projects." Software Centre, Computer Science and Engineering Chalmers / University of Gothenburg, Gothenburg, Sweden.
- STARON, M., MEDING, W., HANSSON, J., HOGLUND, C., NIESEL, K., BERGMANN, V.. (2014). " Dashboards for Continuous Monitoring of Quality for Software Product under Development " - University of Gothenburg – Sweden.
- STEIDL, D., DEISSENBOECK, F., POEHLAMN, M., HEINKE, R., UHINK-MERGETHALER, B. (2014) “Continuous Software Quality Control in Practice” 2014 International Conference on Software Maintenance and Evolution, Munich, Germany.
- TARHAN, A., YILMAZ, S.G., (2014). " Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process." Hacettepe University – Ankara, Turkey.
- TMMI Foundation (2012). Test Maturity Model Integration (TMMi). Release 1.0. Irlanda.
- TOMMY, R., MHAISEKAR, M., KALLEPALLY, S., VARGHESE, L., AHMED, S., SOMARAJU, M. (2015) " Dynamic Quality Control in Agile Methodology for Improving the Quality " ILP Innovations ,Tata Services, 2015 IEEE Conference on Computer Graphics(CGVIS) ,India.
- VICENTE, A.A, (2010) “Definição e gerenciamento de métricas de teste no contexto de métodos ágeis, Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – ICMC/USP, São Carlos, Brazil.
- YAMADA, S., KII, R.. (2015) " Software quality analysis for agile development" Department of Social Management Engineering, Graduate School of Engineering, Tottori University, Japan.
- YING, S., MING, L., STEVEN, A., SMIDTS, C., WALLER, J., EHMKE, N., HASSELBRING, W., (2016) “Metric-based software reliability prediction approach and its application” USA.

Apêndice I - Mapeamento Sistemático sobre Medidas e Indicadores de Testes de Software em Projetos Ágeis

I.1 Introdução

O termo “revisão sistemática” se refere a uma metodologia específica de pesquisa, desenvolvida para coletar e avaliar evidências disponíveis sobre determinado tópico (BIOLCHINI *et al.*, 2005). A partir da execução deste tipo de pesquisa, buscam-se resultados mais abrangentes e menos dependentes do conhecimento do pesquisador, permitindo que a pesquisa seja repetida por outros pesquisadores e os resultados possam ser comparados. O processo de pesquisa em uma revisão sistemática segue um conjunto de passos bem definidos em um protocolo, no qual se especifica a questão central da pesquisa, os critérios de inclusão e exclusão dos estudos e a estratégia de busca utilizada (SANTOS *et al.*, 2008; CONTE, 2009). Este apêndice descreve um mapeamento sistemático da literatura utilizando o mesmo formalismo usado em execução da revisão sistemática, conforme definido acima.

Estudos de mapeamento sistemático ou estudos de escopo são projetados para dar uma visão geral de uma área de pesquisa através da classificação e contando contribuições em relação às categorias dessa classificação (KITCHENHAM *et al.*, 2007). Envolve pesquisar a literatura para saber quais tópicos foram abordados na literatura e onde a literatura foi publicada (PETERSEN *et al.*, 2008). No entanto, um estudo de mapeamento sistemático e uma revisão sistemática da literatura compartilham alguns pontos comuns (por exemplo, em relação à busca e seleção de estudos), eles são diferentes em termos de objetivos e, portanto, abordagens para a análise de dados. Embora as revisões sistemáticas visem sintetizar a evidência, considerando também a força da evidência, os mapeamentos sistemáticos estão principalmente preocupados com a estruturação de uma área de pesquisa (PETERSEN *et al.*, 2015).

Parte dos resultados desta dissertação envolve identificar medidas e indicadores relacionados a testes em projetos ágeis de desenvolvimento de software. Estas medidas serão utilizadas para definição de indicadores que possam apoiar as equipes que gerenciam testes de software.

No contexto desta dissertação, um mapeamento sistemático foi conduzido com o objetivo de identificar trabalhos com propostas ou utilização de medidas e indicadores de testes em projetos ágeis. Este apêndice apresenta o planejamento e a execução deste estudo.

Com base nos resultados obtidos, espera-se obter informações relevantes para:

- (1) identificar as medidas e indicadores utilizados;
- (2) como é a aplicação destas medidas em processo de software na fase de teste;
- (3) criar indicadores com base nas medidas identificadas;
- (4) elaborar fichas no formato da abordagem AMP.br [Bonelli, 2014] com critérios de análise para os indicadores de testes.

Para condução do mapeamento sistemático, foi seguido o processo definido em (SILVA FILHO, 2006), composto pelas seguintes atividades:

- Definir Escopo e Estudos Preliminares
- Definir Protocolo
- Testar Protocolo
- Avaliar Protocolo
- Executar a Pesquisa
- Avaliar Resultados da Pesquisa
- Empacotar Resultados
- Publicar Resultados

A descrição de como cada uma das atividades deste processo foi conduzida é apresentada nas seções seguintes. As duas últimas atividades (Empacotar resultados e Publicar Resultados) não são descritas, pois se referem à disponibilização dos resultados para a comunidade científica, o que é realizado a partir da publicação do estudo neste trabalho.

Além da descrição das atividades do processo de execução do mapeamento sistemático, os dados das publicações selecionadas durante este estudo são apresentados nas últimas seções deste anexo.

1.2. Definição do Escopo e Estudos Preliminares

Nesta primeira atividade, foi realizada uma pesquisa informal sobre medidas e indicadores, objetivando obter conhecimento sobre o domínio. Em um primeiro momento, a pesquisa permaneceu focada em medidas de teste de software. Dentre os trabalhos

encontrados neste momento, destaca-se o trabalho de BORTOLUCI e DUDUCHI (2015) no qual uma observação sobre indicadores de testes de software que podem ser utilizados em projetos que utilizaram métodos ágeis de desenvolvimento.

Após analisar os trabalhos encontrados, verificou-se a importância da etapa de identificação das medidas e relatos de utilização na fase de testes em projetos de desenvolvimento de software. Além disso, verificou-se a carência de abordagens que lidassem com indicadores relacionados a testes.

Outros trabalhos que foram utilizados nesta etapa são: TOMMY *et al.* (2015), YING *et al.* (2016). Estes dois trabalhos foram utilizados para leitura inicial e entendimento sobre algumas medidas de testes. O trabalho de Mota e Dias-Neto (2014) foi utilizado como base para definição da *string* de busca, uma vez que teve o propósito de identificação de medidas relacionadas a teste com abordagem em métodos ágeis, porém sem abordar formas de coleta, análise e utilização de indicadores gerados a partir das medidas. Os autores identificaram e caracterizaram medidas utilizadas em projetos desenvolvidos com métodos ágeis aplicados ao contexto de teste de software. Decidiu-se replicar este estudo cuja realização foi finalizada em março de 2014 e verificar se novas medidas seriam identificadas após o período deste estudo citado. Os autores não disponibilizaram o protocolo completo do mapeamento sistemático original. Dessa forma, o protocolo presente nesse anexo é uma contribuição desta pesquisa.

A partir destas informações, o escopo do mapeamento sistemático foi executado.

I.3. Definição do Protocolo

O protocolo do estudo baseado em revisão sistemática possui o objetivo de guiar a execução do estudo. Este protocolo é composto pela descrição dos seguintes itens: contexto do estudo, objetivos a serem atingidos, questões de pesquisa, escopo delimitado, idiomas escolhidos, métodos de busca das publicações, procedimentos de seleção e critérios para inclusão das publicações no estudo, procedimentos para extração de dados e procedimentos para a análise dos resultados. Estes itens são apresentados nas seções a seguir, adotando a organização utilizada em (SANTOS, 2008).

I.3.1 Contexto

Há várias abordagens no contexto de medidas e indicadores, porém na sua grande parte focadas em gerenciamento de projetos. O foco deste trabalho é focado em teste de software e medidas e indicadores relacionados.

Além disso, espera-se obter informações no intuito de auxiliar: a identificação de dados relevantes para a coleta de medidas, definição de indicadores e análise dos mesmos.

I.3.2 Objetivo

O objetivo deste estudo foi delineado a partir do paradigma GQM (BASILII e ROMBACH, 1988) e consiste em:

Analisar relatos de experiência e publicações científicas sobre projetos ágeis de desenvolvimento

Com o propósito de identificar medidas e indicadores

Com relação às atividades relacionadas a testes

Do ponto de vista de pesquisadores

No contexto acadêmico e industrial

I.3.3 Questões de pesquisa

Para atender o objetivo descrito acima, foram definidas duas questões de pesquisa (QP) a seguir.

- **QP1:** Que medidas relacionadas a teste durante o desenvolvimento ágil de software são citadas na literatura?
- **QP2:** Que indicadores relacionados a teste durante o desenvolvimento ágil de software que podem ser derivados das medidas citadas na literatura?

I.3.4 Escopo

Para delinear o escopo da pesquisa, foram estabelecidos critérios para seleção das fontes de pesquisa. A pesquisa foi realizada a partir de bibliotecas digitais por meio de seus respectivos engenhos de busca.

Os seguintes critérios foram adotados para selecionar as fontes de pesquisa: (1) possuir um mecanismo de busca que permita o uso de expressões lógicas ou funcionalidade

equivalente; (2) pertencer a uma das editoras listadas no Portal de Periódicos da CAPES⁵; (3) incluir em sua base publicações da área de exatas ou correlatas que possuam relação direta com o tema a ser pesquisado; e (4) possuir mecanismos de busca que permitam a busca no texto completo das publicações.

A pesquisa está restrita à análise de publicações obtidas exclusivamente a partir das fontes selecionadas, a partir dos critérios supracitados. O estudo engloba os dados disponíveis nas fontes considerando o período de janeiro de 2004 até dezembro de 2016. Os dados de 2004 até 2013 são derivados do trabalho de Mota e Dias-Neto (2014). Os dados de 2014 a 2016 são derivados da execução realizada exclusivamente no decorrer dessa dissertação. Importante notar que os trabalhos citados no mapeamento conduzido por Mota e Dias-Neto (2014) foram reavaliados e catalogados de acordo com os procedimentos descritos nesse protocolo.

I.3.5 Idiomas

Para a realização desta pesquisa foi selecionado o idioma inglês. A escolha deste idioma deve-se à sua adoção pela grande maioria das conferências e periódicos internacionais relacionados com o tema de pesquisa, e por ser o idioma utilizado pela maioria das editoras relacionadas com o tema listadas no Portal de Periódicos da CAPES.

I.3.6 Métodos de busca de publicações

Assim como no mapeamento original, foram escolhidas as máquinas de busca Scopus (<https://www.sccopus.com/>) e IEEEExplore (<http://ieeexplore.ieee.org/Xplore/home.jsp>).

A máquina de busca será acessada via Web, por meio de expressão de busca pré-estabelecida, apresentada a seguir no formato da Scopus:

⁵ <http://www.periodicos.capes.gov.br>

TITLE-ABS-KEY (("agile" OR "agile method" OR "agile methods" OR "agile methodology" OR "agile methodologies" OR "agile development" OR "agile process" OR "agile processes" OR "agile practice" OR "agile practices" OR "agile project" OR "agile projects" OR "agile lifecycle" OR "SCRUM" OR "extreme programming" OR "lean development" OR "feature driven development" OR "dynamic system development" OR "agile unified process") AND ("metric" OR "metrics" OR "measure" OR "measures" OR "measured" OR "measurement" OR "measurements" OR "indicator" OR "indicators" OR "evaluation") AND ("software project" OR "software projects" OR "software process" OR "software processes" OR "software product" OR "software products" OR "software production")) AND (LIMIT-TO(SUBJAREA,"COMP"))

Para a plataforma IEEE as seguintes expressões foram utilizadas para replicar a string de busca utilizada no mapeamento sistemático original. Esta ação foi necessária devido à máquina de busca ter sofrido mudanças no seu processo de busca limitando a quantidade de palavras chaves. Assim, as strings abaixo foram utilizadas:

- *(agile OR scrum) AND (metric* OR measures OR measurements OR indicators) AND (software*)*
- *(agile OR scrum) AND (XP* OR lean*) AND (software*)*
- *(agile OR scrum) AND ("dynamic system development" OR evaluation) AND (software*)*

I.3.7 Procedimentos de seleção e critérios

Foram considerados dois ciclos de seleção de trabalhos:

- Ciclo 1: Trabalhos oriundos da replicação do mapeamento sistemático descrito em (Mota e Dias-Neto, 2014), considerando o período até 2013.
- Ciclo 2: Trabalhos publicados entre 2014 e 2016.

A seleção dos estudos foi feita em três etapas:

1ª Etapa: *seleção e catalogação preliminar dos estudos coletados:* a seleção preliminar das publicações foi feita a partir da aplicação da expressão de busca à fonte selecionada. Cada publicação foi catalogada e armazenada em uma planilha Excel para posterior análise;

2ª Etapa: *seleção dos estudos relevantes (1º filtro):* a seleção preliminar com o uso da expressão de busca não garante que todo o material coletado seja útil no contexto da pesquisa, pois a aplicação das expressões de busca é restrita ao aspecto sintático. Desta

forma, após a identificação das publicações por meio dos mecanismos de buscas, os títulos e resumos (*abstracts*) foram lidos e analisados seguindo os critérios de inclusão a seguir:

- **CI1:** a publicação objetiva propor ou descrever a identificação de **medidas em projetos ágeis de desenvolvimento de software**.
- **CI2:** a publicação objetiva propor ou descrever a identificação de **medidas de teste em projetos ágeis de desenvolvimento de software**.
- **CI3:** a publicação objetiva propor ou descrever a identificação de **indicadores de teste em projetos ágeis de desenvolvimento de software**.

Nesta etapa foi ainda utilizado os seguintes critérios de exclusão:

- **CE1:** a publicação objetiva propor ou descrever projetos de desenvolvimento de software, mas cujo o foco é apenas **em análise de desempenho de código ou gerenciamento de projetos ou não citam medidas ou indicadores de testes de software**.
- **CE2:** a publicação objetiva propor ou descrever o conjunto de resumo de artigos de **conferências, capítulos de livros e artigos de seminários (*proceedings*)**.
- **CE3:** a publicação objetiva propor ou descrever a identificação de medidas e indicadores de **projetos não-ágeis de desenvolvimento de software**.

3ª Etapa: seleção dos estudos relevantes (2º filtro): apesar de limitar o universo de busca, o filtro anterior empregado também não garante que todo o material coletado seja útil no contexto da pesquisa. Por isso, as publicações selecionadas na 2ª etapa devem ser lidas completamente, verificando se, de fato, atendem a, pelo menos um, dos critérios definidos.

I.3.8 Procedimentos para extração dos dados

De cada uma das publicações aprovadas pelo processo de seleção, os seguintes dados devem ser extraídos:

- Dados da publicação:
 - o Título
 - o Autor(es)
 - o Ano da publicação
 - o Referência completa;
 - o Resumo da publicação;

- o Medidas citadas no artigo (independentemente de ser de teste);
- o Contexto em que a medida é (ou pode ser) utilizada em projetos;
- o Exemplo e utilização;
- o Indícios de que as medidas podem ser utilizadas em outros projetos ágeis;
- o Evidência de uso efetivo;
- o Medidas relacionadas a testes;
- o Fórmulas para cálculo das medidas de teste.

I.3.9 Procedimentos para análise

A análise dos dados é feita tanto de forma quantitativa como qualitativamente.

A análise quantitativa é realizada por meio da extração direta dos dados a partir do banco de dados com os registros dos itens retornados pela máquina de busca. Esta análise fornece o número de publicações selecionadas para fazerem parte do estudo. Estas medidas são: quantidade medidas citadas, quantidade de medidas citadas por artigo, quantidade de citações de cada medida, quantidade de artigos por ano.

A análise qualitativa deve utilizar como base os dados quantitativos e realizar considerações com o intuito de discutir os resultados da busca com relação às questões de pesquisa declaradas. Podemos citar as fases de testes que se pode coletar a medida, fase de planejamento ou execução, assim como classificar se a medida é: objetiva, direta, indireta, acompanhamento, desempenho, cronograma e progresso e qualidade do produto.

Fase de planejamento significa o momento de definição do escopo de testes, estratégias de testes, definição dos cenários, casos de testes e passo-a-passo, ou seja, medidas que possam ser coletadas nesta fase. Fase de execução envolve o momento em que a equipe de testes monitora a execução dos testes e coleta as medidas relacionadas.

Para a classificação das medidas foi utilizada a mesma definida em [Mota e Dias-Neto, 2014] conforme Tabela 26.

Tabela 26 - Classificações das medidas. Fonte: (Mota e Dias-Neto, 2014)

Classificação	Descrição
Objetiva/Subjetiva	Medida objetiva depende do objeto de estudo. Medida subjetiva depende do ponto de vista de quem interpreta
Quantitativa/Qualitativa	Medida quantitativa pertence a um intervalo de certa magnitude e geralmente é calculada por um número. Medida qualitativa é representado por símbolos/artefatos, não numéricos.
Organizacional/Acompanhamento	Medidas organizacionais medem a quantidade de valor de negócio entregue ao cliente. Medidas de acompanhamento apoiam no entendimento e melhoria do processo que produz valor de negócio.
Direta/Indireta	Medidas diretas são medidas básicas e medidas indiretas são aquelas derivadas de outras medidas.
PSM (<i>Practical Software Measurement</i>) (Mcgarry et al., 2012)	• Cronograma e progresso: relacionado ao cumprimento de marcos de projeto e à conclusão de unidades de trabalho nos prazos previstos. • Qualidade do produto: relacionada à capacidade do software produzido de atender sem falhas às necessidades do usuário. • Desempenho do Processo: relacionada à capacidade do processo de atender às necessidades apresentadas em cada projeto.

I.4 Teste do Protocolo

Não foi realizado teste do protocolo uma vez que esta pesquisa é uma replicação de um mapeamento sistemático. No entanto, repetiu-se a execução da string de busca e aplicação dos critérios de inclusão e exclusão no mapeamento original para garantir que os resultados relatados em (Mota e Dias-Neto, 2014) fossem de fato replicados nesta pesquisa. Considerou-se a execução do Ciclo 1 como sendo o teste do protocolo já que o trabalho original era vago em relação aos procedimentos adotados.

Dessa forma, foram definidos como artigos de controle aqueles citados em (Mota e Dias-Neto, 2014), apresentados na Tabela 27. O ID do artigo foi criado para ajudar a identificar o artigo ao longo da pesquisa.

Caso a expressão de busca retornasse todos estes artigos, então se poderia afirmar que a expressão de busca estaria adequada. De fato, a expressão de busca foi capaz de retornar todos os artigos apresentados na Tabela 27.

Em decorrência dessa replicação, 3 artigos originalmente (artigo ID P01_10, P01_12 e P01_15) considerados no mapeamento foram excluídos conforme critério de exclusão CE3.

Tabela 27 - Artigos de controle citados em (Mota e Dias-Neto, 2014)

ID	Título	Autor(es)	Ano
P01_01	Analyses of an agile methodology Implementation	Ilieva, S., Ivanov, P. and Stefanova, E.	2004
P01_02	Embedded Agile Project by the Numbers with Newbies	Schoonderwoert, N. V.	2006
P01_03	AM-QuICk: A Measurement-Based Framework for Agile Methods Customisation	Ayed, H., Habra, N. and Vanderose, B.	2013
P01_04	Measuring the Impact of Scrum on Product Development at Adobe Systems	Green, P.	2011
P01_05	Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A.	2011
P01_06	Metrics Functions for Kanban Guards	Heidenberg, J. and Porres, I.	2010
P01_07	A development of the effectiveness evaluation model for agile software development using the balanced scorecard	Atiwithayaporn, S. and Rivepiboon, W.	2013
P01_08	A metamodel-based approach for customizing and assessing agile methods	Ayed, H., Vanderose, B. and Habra, N.	2012
P01_09	Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process	Mahnic, V. and Vrana, I.	2007
* P01_10	Collaboration model of software development	Kusumasari, T. F., Supriana, I., Surendro, K., Sastramihardja, H.	2011
P01_11	Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery	Brown A., Ambler S. and Royce W.	2013
* P01_12	Using the COSMIC method to estimate Agile user stories	Desharnais, J.-M., Kocaturk, B. and Abran, A.	2011
P01_13	Refactoring the development process: experiences with the incremental adoption of agile practices	Hodgetts, P.	2004
P01_14	Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects	Racheva, Z., Daneva, M. and Buglione, L.	2008
* P01_15	Measuring progress of Scrum-based software projects	Mahnic, V. and Zabkar, N.	2012
P01_16	Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?	Ktata, O. and Levesque, G.	2010

* Os artigos P01_10, P01_12 e P01_15 foram excluídos do escopo do mapeamento.

No mapeamento original de (Mota e Dias-Neto, 2014) chamado neste trabalho de Ciclo 1 foram identificados 24 medidas de teste em 16 artigos, durante a replicação e revisão deste mapeamento original, 3 artigos foram excluídos conforme critério de exclusão (CE3). Houve ainda 15 medidas listadas no mapeamento sistemático (Ciclo 1) que não foram identificadas durante a replicação e revisão dos artigos conforme descrito na Tabela 28. Nesta Tabela 28 existe uma coluna com o nome da medida descrita no mapeamento Ciclo 1 e o código do artigo que foi referenciado pelos autores.

Desta forma o total de medidas identificadas no Ciclo 1 foram de 13 (novas) mais 9 (existentes do mapeamento original), total de 22 medidas mapeadas.

Tabela 28 - Medidas excluídas após revisão do mapeamento Ciclo 1

Nome da Medida	Código do artigo
Tempo de retorno do defeito	P01_06
Tempo gasto para correção dos defeitos	P01_06
Tempo planejado para correção do defeito	P01_06
Porcentagem de tempo gasto na correção de defeitos	P01_01
Número de casos de testes manuais	P01_03
Tempo de conclusão do projeto	P01_11
Ponto de estória concluídos por iteração	P01_13, P01_14, P01_16
Planejamento do tamanho do backlog	P01_03, P01_13
Estórias de usuário não verificadas	P01_06, P01_15
Quantidades de trabalhos restante (em horas) para cada tarefa	P01_15
Quantidades de trabalhos gasto (em horas) para cada tarefa	P01_15
Número de casos de testes executados	P01_05
Estimativas de estórias aprovadas	P01_16
Precisão de estimativa de teste	P01_06
Percentual de estórias aprovadas	P01_07

I.5 Execução da Pesquisa

Após o estabelecimento e aprovação do protocolo de pesquisa, o mapeamento sistemático foi executado. A execução do protocolo foi finalizada em fevereiro de 2017.

I.5.1 – 1º ciclo da execução

Foram lidos todos os artigos do mapeamento sistemático original, aplicou-se os critérios de inclusão e exclusão definidos neste trabalho e também foi realizada a extração dos dados: Na Tabela 29 estão descritos ID criado para identificar artigo conforme Tabela 27, autores, ano de publicação do artigo, título do artigo e colunas para identificar se o artigo passou na 1ª seleção e 2ª seleção baseado nos critérios de inclusão e exclusão descritos na Seção 1.3.7. Foram aplicados estes critérios para o mapeamento original tendo em vista a necessidade de validar se os artigos do 1º ciclo identificaram de fato as medidas citadas nos artigos listados na Tabela 27.

Tabela 29 - Publicações lidas e analisadas do mapeamento (Mota e Dias-Neto, 2014)

ID	Autor(es)	Ano	Título	Passou 1ª seleção?	Passou 2ª seleção?
P01_01	Ilieva, S., Ivanov, P. and Stefanova, E.	2004	Analyses of an agile methodology Implementation	SIM	SIM
P01_02	Schooenderwoert, N. V.	2006	Embedded Agile Project by the Numbers with Newbies	SIM	SIM
P01_03	Ayed, H., Habra, N. and Vanderose, B.	2013	AM-QuICk: A Measurement-Based Framework for Agile Methods Customisation	SIM	SIM

ID	Autor(es)	Ano	Título	Passou 1ª seleção?	Passou 2ª seleção?
P01_04	Green, P.	2011	Measuring the Impact of Scrum on Product Development at Adobe Systems	SIM	SIM
P01_05	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A.	2011	Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs	SIM	SIM
P01_06	Heidenberg, J. and Porres, I.	2010	Metrics Functions for Kanban Guards	SIM	SIM
P01_07	Atiwithayaporn, S. and Rivepiboon, W.	2013	A development of the effectiveness evaluation model for agile software development using the balanced scorecard	SIM	SIM
P01_08	Ayed, H., Vanderose, B. and Habra, N.	2012	A metamodel-based approach for customizing and assessing agile methods	SIM	SIM
P01_09	Mahnic, V. and Vrana, I.	2007	Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process	SIM	SIM
* P01_10	Kusumasari, T. F., Supriana, I., Surendro, K., Sastramihardja, H.	2011	Collaboration model of software development	SIM	NÃO
P01_11	Brown A., Ambler S. and Royce W.	2013	Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery	SIM	SIM
* P01_12	Desharnais, J.-M., Kocaturk, B. and Abbran, A.	2011	Using the COSMIC method to estimate Agile user stories	SIM	NÃO
P01_13	Hodgetts, P.	2004	Refactoring the development process: experiences with the incremental adoption of agile practices	SIM	SIM
P01_14	Racheva, Z., Daneva, M. and Buglione, L.	2008	Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects	SIM	SIM
* P01_15	Mahnic, V. and Zabkar, N.	2012	Measuring progress of Scrum-based software projects	SIM	NÃO
P01_16	Ktata, O. and Levesque, G.	2010	Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?	SIM	SIM

* Os artigos P01_10, P01_12 e P01_15 foram excluídos do escopo do mapeamento.

I.5.2 – 2º ciclo da execução – 1ª seleção

A máquina de busca utilizada foi a Scopus e IEEE assim como no mapeamento original. Após a execução da expressão de busca, os seguintes resultados foram obtidos conforme Figura 36. Foram retornadas 139 publicações a partir do ano de 2014, todas as publicações tiveram seus títulos e resumos lidos de acordo com os critérios de inclusão definido. Foram selecionados 22 artigos para a 2ª seleção. Na Tabela 30 estão listados os títulos dos 139 artigos retornados, além de ano de publicação e autores. Além disso, existe duas colunas para identificar se o artigo após leitura do título e abstract passou na 1ª seleção (atendeu os critérios de inclusão) para leitura completa. A última coluna identifica se o artigo passou na 2ª seleção o que significa que ele contribui com informações sobre medidas de testes.

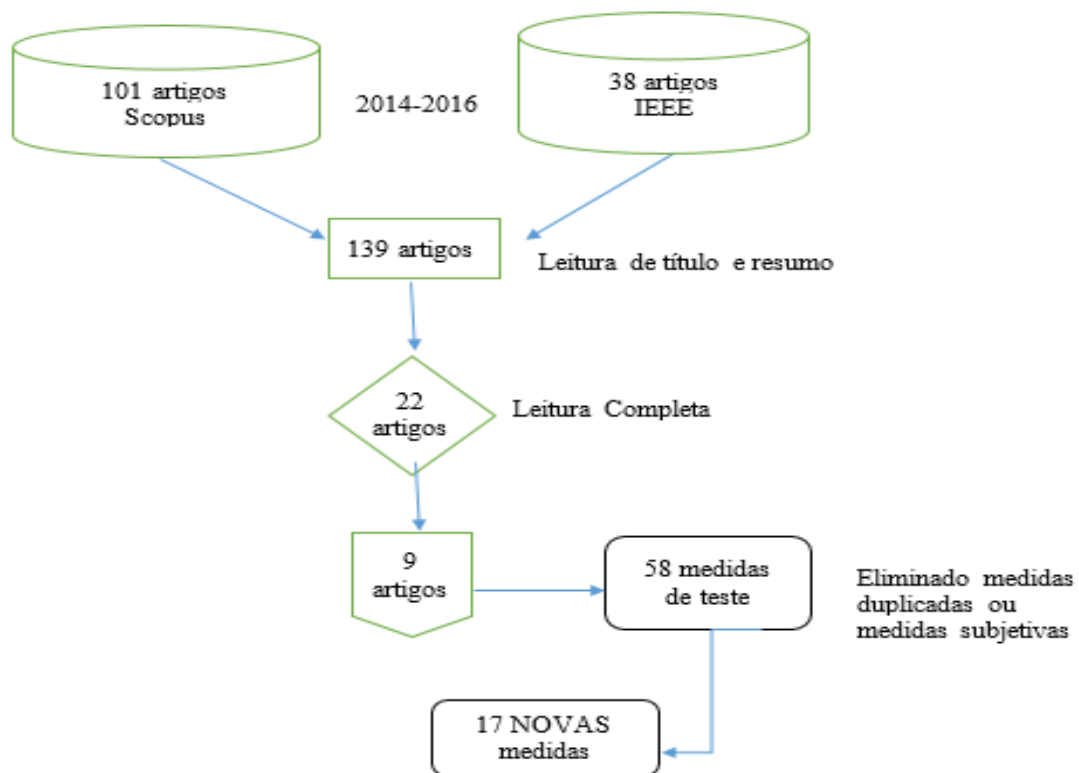


Figura 36 - Fluxo de pesquisa e leitura dos artigos no Ciclo 2 do mapeamento

Tabela 30 - Publicações retornadas na execução do 2º ciclo (2014-2016)

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
Huijgens H., Bruntink M., Van Deursen A., Van Der Storm T., Vogelesang F.	2016	An exploratory study on functional size measurement based on code	CE1	-
[No author name available]	2016	Proceedings - International Conference on Software and System Process, ICSSP 2016	CE2	-
Kayes I., Sarker M., Chakareski J.	2016	Product backlog rating: a case study on measuring test quality in scrum	CI2	SIM
Reddaiah B., Pradeep Kumar Reddy R., Nagaraju C., Harsha Sree V.	2016	A novel approach to adopt scrum by an enterprise	CE1	-
Colonese E.	2016	Agile: The human factors as the weakest link in the chain	CE1	-
Kołaczek G., Świątek P., Świątek J., Brzostowski K., Grzech A., Sławek A.	2016	Test scenarios and validation results of POP methodology	CI2	NÃO
Košinár M.A., Czopik J., Štolfa J.	2016	Formal Knowledge Framework for Software Processes Architecture	CE1	-
Mauro V., Messina A.	2016	AMINSEP-agile methodology implementation for a new software engineering paradigm definition. A research project proposal	CI2	NÃO
Kalinowski M., Felderer M., Conte T., Spínola R., Prikladnicki R., Winkler D., Fernández D.M., Wagner S.	2016	Preventing incomplete/hidden requirements: Reflections on survey data from Austria and Brazil	CE1	-
Raisian K., Yahaya J., Deraman A.	2016	Current challenges and conceptual model of green and sustainable software engineering	CE1	-
Basri S., Kama N., Haneem F., Ismail S.A.	2016	Predicting effort for requirement changes during software development	CE1	-
Maxim B.R., Kaur R., Apzynski C., Edwards D., Evans E.	2016	An agile software engineering process improvement game	CE1	-
Malhotra R., Chug A.	2016	Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance	CI2	SIM
Kastl P., Kiesmüller U., Romeike R.	2016	Starting out with projects - Experiences with agile software development in high schools	CE1	-
Gupta R.K., Manikreddy P., Gv A.	2016	Challenges in adapting agile testing in a legacy product	CI2	SIM
Kuhrmann M., Diebold P., Munch J., Tell P.	2016	How does software process improvement address global software engineering?	CE1	-
Barbosa R., Silva A.E.A., Moraes R.	2016	Use of Similarity Measure to Suggest the Existence of Duplicate User Stories in the Scrum Process	CE1	-
Voigt S., Von Garrel J., Müller J., Wirth D.	2016	A Study of Documentation in Agile Software Projects	CE1	-
[No author name available]	2016	International Symposium on Empirical Software Engineering and Measurement	CE2	-
Mansor Z., Arshad N.H., Yahya S., Razali R., Yahaya J.	2016	Ruler for effective cost management practices in agile software development projects	CE1	-

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
Martini A., Bosch J.	2016	A multiple case study of continuous architecting in large agile companies: current gaps and the CAFFEA framework	CE1	-
Alyahya S., Alqahtani M., Maddeh M.	2016	Evaluation and improvements for agile planning tools	CE1	-
Janjua J.I., Ali A., Chaudhry M.U.	2016	OPENTCQ: Towards change management in hybrid agile model	CE1	-
[No author name available]	2016	ACM International Conference Proceeding Series	CE2	-
Willamy R., Nunesy J., Perkusichz M., Freirex A., Saraiva R., Almeidaak H., Perkusich A.	2016	A method to build Bayesian networks based on artifacts and metrics to assess agile projects	CI2	SIM
Perkusich M., Gorgônio K.C., Almeida H., Perkusich A.	2016	Assisting the continuous improvement of Scrum projects using metrics and Bayesian networks	CE1	-
Rajkovic P., Petkovic I., Milenkovic A., Jankovic D.	2016	Combining agile and traditional methodologies in medical information systems development process	CE1	-
Alomar N., Almobarak N., Alkoblan S., Alhozaimy S., Alharbi S.	2016	Usability engineering of agile software project management tools	CE1	-
Maric M., Matkovic P., Tumbas P., Pavlicevic V.	2016	Documenting agile architecture: Practices and recommendations	CE1	-
[No author name available]	2016	Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE	CE2	-
[No author name available]	2016	CEUR Workshop Proceedings	CE2	-
[No author name available]	2016	16th International Conference on Software Process Improvement and Capability Determination, SPICE 2016	CE2	-
[No author name available]	2016	5th International Conference on Design, User Experience, and Usability, DUXU 2016 and held as Part of 18th International Conference on Human-Computer Interaction, HCI International 2016	CE2	-
[No author name available]	2016	ENASE 2016 - Proceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering	CE2	-
Fischer H., Senft B.	2016	Human-centered software engineering as a chance to ensure software quality within the digitization of human workflows	CE1	-
C. Matthies; T. Kowark; K. Richly; M. Uflacker; H. Plattner	2016	ScrumLint: Identifying Violations of Agile Practices Using Development Artifacts	CE1	-
H. Cao; C. Xue; B. Zhang	2016	Analysis and evaluation on influencing factors of enterprise information system innovation capacity	CE1	-
R. K. Gupta; P. Manikreddy; A. GV	2016	Challenges in Adapting Agile Testing in a Legacy Product	Duplicado (scopus)	
C. Matthies; T. Kowark; M. Uflacker; H. Plattner	2016	Agile metrics for a university software engineering course	CE1	-
V. Singh; L. L. Pollock; W. Snipes; N. A. Kraft	2016	A case study of program comprehension effort and technical debt estimations	CE1	-

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
R. Barbosa; A. E. A. Silva; R. Moraes	2016	Use of Similarity Measure to Suggest the Existence of Duplicate User Stories in the Scrum Process	Duplicado (scopus)	
K. M. Bumbarly	2016	Using Velocity, Acceleration, and Jerk to Manage Agile Schedule Risk	CE1	-
R. Niedermayr; E. Juergens; S. Wagner	2016	Will My Tests Tell Me If I Break This Code?	CI2	NÃO
D. R. Greening	2016	Agile Base Patterns in the Agile Canon	CI2	NÃO
M. Karakus; A. Duresi	2016	A Scalability Metric for Control Planes in Software Defined Networks (SDNs)	CE1	-
M. de los Angeles Martin; M. J. Divan	2016	Case based organizational memory for processing architecture based on measurement metadata	CE1	-
F. Furtado; A. Zisman	2016	Trace++: A Traceability Approach to Support Transitioning to Agile Software Engineering	CI2	NÃO
A. Marshall; R. F. Gamble; M. L. Hale	2016	Outcomes of Emotional Content from Agile Team Forum Posts	CE1	-
Yamada S., Kii R.	2015	Software quality analysis for agile development	CI2	SIM
Ramasubbu N., Bharadwaj A., Tayi G.K.	2015	Software process diversity: Conceptualization, measurement, and analysis of impact on project performance	CE1	-
Al-Elaimat A., Al-Ghuwairi A.-R.	2015	Procedural assessment process of software quality models using agility	CE1	-
Jiang Y., Zhang C., Wu D., Liu P.	2015	A Preliminary Analysis and Case Study of Feature-Based Software Customization (Extended Abstract)	CE1	-
Oliveira F., Goldman A., Santos V.	2015	Managing Technical Debt in Software Projects Using Scrum: An Action Research	CE1	-
Fehlmann S., Falkner K.	2015	A case study in agility and evolving the long-lived software system	CE1	-
Misra R., Panigrahi C.R., Panda B., Pati B.	2015	Software design	CE1	-
Rindell K., Hyrnsalmi S., Leppänen V.	2015	A comparison of security assurance support of agile software development methods	CE1	-
Khelladi D.-E., Bendraou R., Baarir S., Laurent Y., Gervais M.-P.	2015	A framework to formally verify conformance of a software process to a software method	CE1	-
Matta M., Marchesi M.	2015	Understanding approval rating of agile project management tools using twitter	CE1	-
Omar M., Syed Abdullah S.L.	2015	The impact of agile methodology on software team's work-related well-being	CE1	-
Suranto B.	2015	Exploratory software testing in agile project	CE1	-
Raschke W., Zilli M., Loinig J., Weiss R., Steger C., Kreiner C.	2015	Where does all this waste come from?	CI1	NÃO
Mohanarajah S., Jabar M.A.	2015	An improved adaptive and dynamic hybrid agile methodology to enhance software project success deliveries	CE1	-
[No author name available]	2015	15th International Conference on Software Process Improvement and Capability Determination, SPICE 2015	CE2	-
[No author name available]	2015	4th International Conference on Software Process Improvement, CIMPS 2015	CE2	-

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
Kupiainen E., Mäntylä M.V., Itkonen J.	2015	Using metrics in Agile and Lean software development - A systematic literature review of industrial studies	CI2	SIM
Gupta D., Dwivedi R.	2015	A framework to support evaluation of project in-hand and selection of software development method	CE1	-
Crawford B., Crawford K., Soto R., de la Barra C.L.	2015	Creativity in agile software development methods	CE1	-
Humayoun S.R., Dubinsky Y., Catarci T.	2015	User evaluation support through development environment for agile software teams	CE1	-
Misirli A.T., Verner J., Markkula J., Oivo M.	2015	Factors affecting team motivation: A survey of Finnish software engineers	CE1	-
Fleischmann A., Schmidt W., Stary C.	2015	Requirements specification as executable software design - A behavior perspective	CE1	-
Ali N.B., Petersen K., De França B.B.N.	2015	Evaluation of simulation-assisted value stream mapping for software product development: Two industrial cases	CE1	-
[No author name available]	2015	21st International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2015	CE2	-
[No author name available]	2015	9th International Conference on Evaluation of Novel Approaches to Software Engineering, ENASE 2014	CE2	-
[No author name available]	2015	16th International Conference on Product-Focused Software Process Improvement, PROFES 2015	CE2	-
Biró M., Messnarz R., Colomo-Palacios R.	2015	Process improvement approaches fertilised by advances in SPI	CE2	-
Magdaleno A.M., De Oliveira Barros M., Werner C.M.L., De Araujo R.M., Batista C.F.A.	2015	Collaboration optimization in software process composition	CE1	-
[No author name available]	2015	2015 IEEE 8th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2015 – Proceedings	CE2	-
[No author name available]	2015	23rd International Conference on Systems Engineering, ICSEng 2014	CE2	-
Staron M., Meding W., Palm K.	2015	Release readiness indicator for mature agile and lean software development projects	CI2	SIM
J. A. Krill; A. Darrin; C. J. Grant; A. Kedia	2015	Engineering innovation into systems engineering practices	CE1	-
K. Power; K. Conboy	2015	A Metric-Based Approach to Managing Architecture-Related Impediments in Product Development Flow: An Industry Case Study from Cisco	CE1	-
C. Kin Keong; K. Tieng Wei; A. A. Abd. Ghani; K. Y. Sharif	2015	Toward using software metrics as indicator to measure power consumption of mobile application: A case study	CE1	-
S. Morasca	2015	Classifying faulty modules with an extension of the H-index	CE1	-
J. Duenckel; R. Soileau; J. Pittman	2015	An electrical reliability metric for preventive maintenance: Mean time between failure plus finds	CE1	-

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
F. Oliveira; A. Goldman; V. Santos	2015	Managing Technical Debt in Software Projects Using Scrum: An Action Research	Duplicado (scopus)	
M. P. Boerman; Z. Lubsen; D. A. Tamburri; J. Visser	2015	Measuring and Monitoring Agile Development Status	CE1	
B. Celebic; R. Breu	2015	Using Green KPIs for Large IT Infrastructures' Energy and Cost Optimization	CE1	
S. Almugrin; A. Melton	2015	Indirect Package Coupling Based on Responsibility in an Agile, Object-Oriented Environment	CE1	
M. D. Tokcan; O. Ozturk; H. Tuna	2015	MetTest: A Test Automation Framework for Development of a Point-To-Multipoint Radio	CE1	
A. Marshall; R. Gamble	2015	Gauging influence in software development teams	CE1	
D. R. Greening	2015	Agile Enterprise Metrics	CI1	NÃO
Eloranta V.-P., van Heesch U., Avgeriou P., Harrison N., Koskimies K.	2014	Lightweight Evaluation of Software Architecture Decisions	CE1	-
Staron M., Meding W., Hansson J., Höglund C., Niesel K., Bergmann V.	2014	Dashboards for Continuous Monitoring of Quality for Software Product under Development	CI2	SIM
[No author name available]	2014	5th International Workshop on Emerging Trends in Software Metrics, WETSoM 2014 – Proceedings	CE2	-
Lehtinen T.O.A., Virtanen R., Viljanen J.O., Mäntylä M.V., Lassenius C.	2014	A tool supporting root cause analysis for synchronous retrospectives in distributed software teams	CE1	-
Ruhe G., Wohlin C.	2014	Software project management in a changing world	CE2	-
Castillo L.	2014	The use of SCRUM for laboratory sessions monitoring and evaluation in a university course enforcing transverse competencies	CE1	-
[No author name available]	2014	15th International Conference on Agile Software Development, XP 2014	CE2	-
Tarhan A., Yilmaz S.G.	2014	Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process	CE3	NÃO
Grzegorzóka P.	2014	Configuration management in agile software development	CE1	-
Agarwal A., Garg N.K., Jain A.	2014	Quality assurance for product development using Agile	CI1	NÃO
Sadiq M., Hassan T.	2014	An extended adaptive software development process model	CE1	-
Parrend P., Masai P., Zanni-merk C., Collet P.	2014	Swarm projects: Beyond the metaphor	CI1	NÃO
Verdugo J., Rodríguez M., Piattini M.	2014	Using agile methods to implement a laboratory for software product quality evaluation	CE1	-
Moe N.B., Šmite D., Šblis A., Börjesson A.-L., Andréasson P.	2014	Networking in a large-scale distributed agile project	CE1	-
Gazzan M., Shaikh A.	2014	Towards bridging the gap between CMMI and agile development methodologies	CE1	-

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
[No author name available]	2014	21st European Conference on Systems, Software and Services Process Improvement and Innovation, EuroSPI 2014	CE2	-
[No author name available]	2014	2014 International Conference on Software and Systems Process, ICSSP 2014 - Proceedings	CE2	-
Asta S., Özcan E., Siebers P.-O.	2014	An investigation on test driven discrete event simulation	CE1	-
Paz F., Zapata C., Pow-Sang J.A.	2014	An approach for effort estimation in incremental software development using cosmic function points	CE1	-
Dumke R., Neumann R., Schmietendorf A., Wille C.	2014	Empirical-based extension of the COSMIC FP method	CE1	-
Humayoun S.R., Dubinsky Y., Catarci T.	2014	User evaluation support through development environment for agile software teams	CE1	-
Maciaszek L.A., Sienkiewicz L.D.	2014	Complexity-aware software process management: A case of scrum in network organization	CE1	-
Dumas-Monette J.-F., Trudel S.	2014	Requirements engineering quality revealed through functional size measurement: An empirical study in an agile context	CI1	NÃO
[No author name available]	2014	36th International Conference on Software Engineering, ICSE Companion 2014 - Proceedings	CE2	-
Butt S.M., Onn A., Butt M.M., Tabassam N.	2014	Towards a model-based framework for integrating usability evaluation techniques in agile software model	CE1	-
[No author name available]	2014	48th Annual Convention of Computer Society of India, CSI 2013	CE2	-
[No author name available]	2014	10th International Conference on Predictive Models in Software Engineering, PROMISE 2014	CE2	-
Farahani F.F., Ramsin R.	2014	Methodologies for agile product line engineering: A survey and evaluation	CE1	-
Fehlmann T.M., Kranich E.	2014	Early software project estimation the six sigma way	CE1	-
Fitzgerald B., Musiał M., Stol K.-J.	2014	Evidence-based decision making in lean software project management	CI1	NÃO
Gómez M.N., Acuña S.T.	2014	A replicated quasi-experimental study on the influence of personality and team climate in software development	CE1	-
Fontana R.M., Fontana I.M., Da Rosa Garbuio P.A., Reinehr S., Malucelli A.	2014	Processes versus people: How should agile software development maturity be defined?	CE1	-
Rotella P., Chulani S.	2014	Predicting release quality	CI1	NÃO
[No author name available]	2014	Proceedings - 14th International Conference on Computational Science and Its Applications, ICCSA 2014	CE2	-
Shenvi A.A.	2014	Navigating the maze - Journey towards an optimal	CE1	-
D. Athanasiou; A. Nugroho; J. Visser; A. Zaidman	2014	Test Code Quality and Its Relation to Issue Handling Performance	CE1	-

Autor(es)	Ano	Título	1ª seleção?	2ª seleção?
Ä. OkÄYar	2014	A Bluetooth signal strength based indoor localization method	CE1	-
X. Zhao; X. Xuan; A. Wang; D. Liu; L. Zheng	2014	Software Quality Control via Exit Criteria Methodology: An Industrial Experience Report	CI2	SIM
A. Tahir; S. G. MacDonell; J. Buchan	2014	Understanding class-level testability through dynamic analysis	CE1	-
P. Mell; R. E. Harang	2014	Using Network Tainting to Bound the Scope of Network Ingress Attacks	CE1	-
M. Butkiewicz; H. V. Madhyastha; V. Sekar	2014	Characterizing Web Page Complexity and Its Impact	CE1	-
P. Rotella; S. Chulani	2014	Predicting Release Quality	Duplicado (scopus)	
A. Rastogi; A. Sureka	2014	What Community Contribution Pattern Says about Stability of Software Project?	CE1	-
M. Benner-Wickner; M. Book; T. BrÄ¼ckmann; V. Gruhn	2014	Examining Case Management Demand Using Event Log Complexity Metrics	CE1	-
V. Antinyan; M. Staron; W. Meding; P. Ä–sterstrÄ¼m; E. Wikstrom; J. Wrangler; A. Henriksson; J. Hansson	2014	Identifying risky areas of software code in Agile/Lean software development: An industrial experience report	CE1	-
B. Priyogi; B. A. Nan Cenka; A. A. G. Y. Paramartha; A. Rubhasy	2014	Work in progress - Open Education Metric (OEM) developing metric to measure open education service quality	CE1	-
L. A. B. Silva; L. Ribeiro; C. Costa; J. L. Oliveira; M. Santos	2014	Normalizing medical imaging archives for dose quality assurance and productivity auditing	CE1	-
R. Mishra; A. Sureka	2014	Mining Peer Code Review System for Computing Effort and Contribution Metrics for Patch Reviewers	CE1	-

Na Seção 1.7 serão discutidos os resultados da questão de pesquisa e na Seção 1.8 são descritos os formulários de extração dos dados dos artigos mapeados.

I.6 – 2º ciclo da execução – 2ª seleção

Na segunda seleção foram lidos por completos os 22 artigos selecionados na 1ª seleção e identificadas 58 medidas de teste. Algumas medidas foram excluídas por terem cálculo de forma subjetiva nos artigos referenciados. Após a eliminação destas medidas e de medidas duplicadas ou já mapeadas em [Mota e Dias-Neto, 2014], foram identificadas 17 novas medidas. Na Tabela 31 estão listados ID artigo, Autores, Ano de publicação e título dos artigos que contribuíram com medidas neste ciclo de execução.

Tabela 31 - Publicações que contribuíram na identificação de medidas do 2º ciclo

ID	Autor(es)	Ano	Título
P02_01	Kayes I., Sarker M., Chakareski J.	2016	Product backlog rating: a case study on measuring test quality in scrum
P02_02	Yamada S., Kii R.	2015	Software quality analysis for agile development
P02_03	Kupiainen E., Mäntylä M.V., Itkonen J.	2015	Using metrics in Agile and Lean software development - A systematic literature review of industrial studies
P02_04	Staron M., Meding W., Palm K.	2015	Release readiness indicator for mature agile and lean software development projects
P02_05	Staron M., Meding W., Hansson J., Höglund C., Niesel K., Bergmann V.	2014	Dashboards for Continuous Monitoring of Quality for Software Product under Development
P02_06	Willamy R., Nunesy J., Perkusich M., Freirex A., Saraiva R., Almeida H., Perkusich A.	2016	A method to build Bayesian networks based on artifacts and metrics to assess agile projects
P02_07	Gupta R.K., Manikreddy P., Gv A.	2016	Challenges in adapting agile testing in a legacy product
P02_08	Malhotra R., Chug A.	2016	Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance
P02_09	X. Zhao; X. Xuan; A. Wang; D. Liu; L. Zheng	2014	Software Quality Control via Exit Criteria Methodology: An Industrial Experience Report

Na Tabela 32 estão listados ID dos artigos, a referência completa dos artigos para os dois ciclos do mapeamento. A coluna ID é para identificar qual ciclo de mapeamento o artigo faz parte. Os artigos do Ciclo 1 tem o código iniciando com P01 e os do Ciclo 2 como P02.

Tabela 32 - Lista de publicações que contribuíram com medidas nos ciclos 1 e 2

ID	Referência Completa
P01_01	Ilieva, S., Ivanov, P. and Stefanova, E. (2004) "Analyses of an agile methodology Implementation", Proceedings of the 30 th EUROMICRO conference. Pp. 326-333
P01_02	Schoonderwoert, N. V. (2006) "Embedded Agile Project by the Numbers with Newbies" Agile 2006 conference report.
P01_03	Ayed, H., Habra, N. and Vanderose, B. (2013) "AM-QuICK: A Measurement-Based Framework for Agile Methods Customisation", In Proc. IWSM/Mensura 2013:71-80.
P01_04	Green, P. (2011) "Measuring the Impact of Scrum on Product Development at Adobe Systems", In: 44 th Hawaii International Conference on System Sciences, p.1-10.
P01_05	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A., (2011) "Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs" Electronic, Robotics and Automotive Mechanics Conference (CERMA), 2011 IEEE, 421-426.
P01_06	Heidenberg, J. and Porres, I., (2010), "Metrics Functions for Kanban Guards", In: IEEE International Conference and Workshops on Engineering of Computer-Based Systems. IEEE computer society.
P01_07	Atiwithayaporn, S. and Rivepiboon, W., (2013), "A development of the effectiveness evaluation model for agile software development using the balanced scorecard", IMECS.

ID	Referência Completa
P01_08	Ayed, H., Vanderose, B. and Habra, N., (2012), "A metamodel-based approach for customizing and assessing agile methods", In: The International Conference on the Quality of Information and Communications Technology (QUATIC).
P01_09	Mahnic, V. and Vrana, I., (2007), "Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process", Electrotechnical Review.
P01_11	Brown A., Ambler S. and Royce W., (2013), 'Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery', ICSE.
P01_13	Hodgetts, P., (2004), "Refactoring the development process: experiences with the incremental adoption of agile practices", IEEE Computer Society. In: Agile Development Conference.
P01_14	Racheva, Z., Daneva, M. and Buglione, L., (2008), "Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects", Proc. 34 th Euromicro Conf. Software Engineering and Advanced Applications, pp-457-464.
P01_16	Ktata, O. and Levesque, G., (2010), "Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?", Proceedings of the Third Conference on Computer Science and Software Engineering, p.101-107.
P02_01	Kayes I., Sarker M., Chakareski J., (2016), "Product backlog rating: a case study on measuring test quality in scrum", Innovation System software engineering
P02_02	Yamada S., Kii R., (2015), "Software quality analysis for agile development", 2015 4th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2015
P02_03	Kupiainen E., Mäntylä M.V., Itkonen J. (2015), "Using metrics in Agile and Lean software development - A systematic literature review of industrial studies", Information and Software Technology
P02_04	Staron M., Meding W., Palm K., (2015), "Release readiness indicator for mature agile and lean software development projects", Lecture Notes in Business Information Processing
P02_05	Staron M., Meding W., Hansson J., Höglund C., Niesel K., Bergmann V., (2014), "Dashboards for Continuous Monitoring of Quality for Software Product under Development", Relating System Quality and Software Architecture, pp-209-229.
P02_06	Willamy R., Nunesy J., Perkusichz M., Freirex A., Saraiva R., Almeida H., Perkusich A., (2016), "A method to build Bayesian networks based on artifacts and metrics to assess agile projects", Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE.
P02_07	Gupta R.K., Manikreddy P., Gv A., (2016), "Challenges in adapting agile testing in a legacy product", 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)
P02_08	Malhotra R., Chug A., (2016), "Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance", Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016
P02_09	X. Zhao; X. Xuan; A. Wang; D. Liu; L. Zheng , (2014), "Software Quality Control via Exit Criteria Methodology: An Industrial Experience Report", 2014 21st Asia-Pacific Software Engineering Conference

I.7 Avaliação dos Resultados da Pesquisa

A partir das informações extraídas das publicações selecionadas para o estudo, foi possível responder, parcialmente, às questões de pesquisa formuladas na Seção I3.3.

I.7.1 - QP1: Que medidas relacionadas a teste durante o desenvolvimento ágil de software são citadas na literatura?

Em relação à questão de pesquisa QP1 ("Que medidas relacionadas a teste durante o desenvolvimento ágil de software são citadas na literatura?"), de uma forma geral, as

abordagens propostas nos trabalhos identificados permitiram identificação destas medidas e indicadores relacionados conforme Tabela 33, descrita a seguir.

Após a revisão do Ciclo 1, restaram 22 medidas de testes conforme listado na Tabela 33.

No mapeamento descrito como Ciclo 2 foram selecionados 9 artigos e 17 medidas novas, que não foram mapeadas no Ciclo 1. O conjunto total de medidas identificado é apresentado na Tabela 33. A coluna ID Med identifica um código atribuído a cada medida mapeada, note que esse código não é sequencial devido a algumas medidas terem sido excluídas durante a análise e revisão feita com especialistas. A coluna Medidas descreve o nome da Medida, também é descrito em qual fase/atividade de coleta a medida está associada e a classificação conforme descrito na Tabela 26. A coluna ID artigos identifica os artigos que citaram as medidas descritas baseado no ID artigos atribuídos na Tabela 32 para todos os artigos em que as medidas foram identificadas no 1º ciclo de execução por Mota e Dias-Neto (2014) ou no 2º ciclo da replicação do mapeamento sistemático.

A discussão dos resultados é feita em conjunto com as medidas identificadas nos Ciclos 1 e 2 do mapeamento sistemático nesta Seção. O mesmo é feito para os formulários de coleta descritos na Seção I.8.

Tabela 33 - Lista de medidas mapeadas após exclusão de medidas repetidas

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED01	Avaliação do backlog do produto	Objetiva, Qualitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P02_01
MED02	Cobertura de Testes	Objetiva, Quantitativa, direta, Acompanhamento	Planejamento	P01_16 P02_03 P02_09 P02_07
MED03	Complexidade de defeitos	Objetiva, Qualitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P02_01 P02_05
MED04	Defeitos do cliente	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P01_02
MED05	Densidade de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P01_07
MED06	Densidade de falhas por linhas de código	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do Produto	Execução	P01_08
MED07	Eficiência na remoção de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Desempenho do processo	Execução	P01_01 P01_02 P01_11

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED08	Esforço gasto nos testes de aceite	Objetiva, Quantitativa, direta, Acompanhamento, Cronograma e Progresso	Planejamento	P01_01
MED09	Funcionalidades testadas	Objetiva, Quantitativa, Direta, Acompanhamento, Cronograma e Progresso	Execução	P02_06
MED10	Histórias de usuário não verificadas	Objetiva, Quantitativa, Direta, Acompanhamento	Execução	P01_06
MED11	Indicador de Previsão de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Acompanhamento, Desempenho do Processo	Execução	P02_03
MED12	Indicador de tendência de defeitos	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P02_03
MED14	Número de casos de teste automatizados	Objetiva, Quantitativa, Direta, Acompanhamento, Qualidade do produto	Planejamento	P01_03 P02_03
MED15	Quantidade de defeitos a serem removidos (backlog da release)	Objetiva, Quantitativa, Direta, Acompanhamento, Cronograma e Progresso	Execução	P02_04
MED16	Numero de defeitos em aberto (não resolvidos)	Objetiva, Quantitativa, Indireta, Acompanhamento, Desempenho do processo	Execução	P01_06 P02_06
MED18	Prontidão para release (Número de semanas para Release)	Objetiva, Quantitativa, Direta, Acompanhamento, Qualidade do produto	Planejamento	P02_04 P02_05
MED19	Número de testes de aceitação por história	Objetiva, Quantitativa, Direta, Acompanhamento, Qualidade do produto	Planejamento	P01_03
MED20	Onda de Impacto dos defeitos corrigidos	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_01
MED21	Percentual de cobertura de Teste	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P01_03
MED22	Porcentagem de esforço gasto para correção de defeitos	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P01_01
MED23	Quantidade de casos de testes executados	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_02 P02_04 P02_05 P02_06 P01_07
MED24	Quantidade de casos de testes executados com sucesso	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_03 P02_04 P02_05
MED25	Quantidade de casos de testes falhados	Objetiva, Quantitativa, Indireta, Acompanhamento, Qualidade do produto	Execução	P02_04
MED26	Quantidade de casos de testes planejados	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Planejamento	P02_03 P02_04

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED27	Quantidade de defeitos /velocidade	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03
MED28	Quantidade de defeitos não cobertos pelos casos de testes	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_01
MED29	Quantidade de funcionalidade que passam nos testes de aceitação	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03 P02_04 P01_14 P01_16
MED30	Quantidade de testes unitários	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03
MED32	Taxa de aumento de casos de testes entre versões/releases	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03
MED33	Taxa de Defeitos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_01 P01_02 P02_03
MED34	Tempo (em dias) para detecção de defeitos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_08
MED35	Tempo para corrigir os testes	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_03
MED36	Tempo para correção do defeito	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_03 P01_06
MED37	Total de defeitos abertos em um dia específico da release.	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_04
MED38	Total de Defeitos analisados e não removidos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_03 P02_04 P01_04
MED39	Total de defeitos encontrados nos testes	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P01_01 P01_02 P02_01 P02_03 P02_02 P02_04 P02_05 P02_07 P02_06 P02_08 P01_03 P01_05 P01_13
MED40	Total de defeitos removidos	Objetiva, Quantitativa, Direta, Acompanhamento, Desempenho do processo	Execução	P02_04 P01_07
MED41	Qualidade da entrega	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P01_02

ID Med	Medidas	Classificação	Atividade de Coleta	ID Artigos
MED42	Confiança nos testes	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P02_01
MED43	Débito de teste	Objetiva, Qualitativa, Indireta, Acompanhamento, Desempenho do Processo	Execução	P02_07

Observa-se ainda que apesar do foco da pesquisa ser em projetos que utilizem métodos ágeis, é possível observar que existem medidas identificadas que também podem estar presentes em projetos de contexto “tradicional”, exemplo: MED 39. Outras medidas só se fazem presente em projetos que utilizem métodos ágeis, exemplo MED 15.

O objetivo desta pesquisa é focado em metodologia ágil, pela importância que estas metodologias têm nos projetos atuais, que buscam acelerar as entregas, mantendo a qualidade e focando em desenvolvimento contínuo e entregas constantes, desta forma decidiu-se que esse foco traria um conjunto maior de medidas que possivelmente não são mapeadas em projetos ditos tradicionais.

Na Figura 37 é exibida a quantidade de medidas relacionadas a testes identificadas por cada artigo nos Ciclos 1 e 2. No gráfico são exibidos o total de medidas de testes citadas por artigo, o total de medidas novas identificadas que não tinham sido encontradas durante Ciclo 1 e o total de medidas mapeadas anteriormente no Ciclo 1 para cada artigo (por essa razão não há valores para os artigos dos ID P02_01 até P02_09, pois os mesmos foram mapeados no Ciclo 2). O ID do artigo é exibido no eixo X conforme descrição apresentada na Tabela 32. Os artigos de ID P01_01 até P01_16 foram identificados no Ciclo 1, os artigos de ID P02_01 até P02_09 foram identificados no Ciclo 2. No total, 106 menções a medidas foram identificadas. Após uma análise de medidas citadas mais de uma vez ou medidas com nomes diferentes mais que significavam a mesma coisa, chegou-se a uma lista final de 39 medidas.

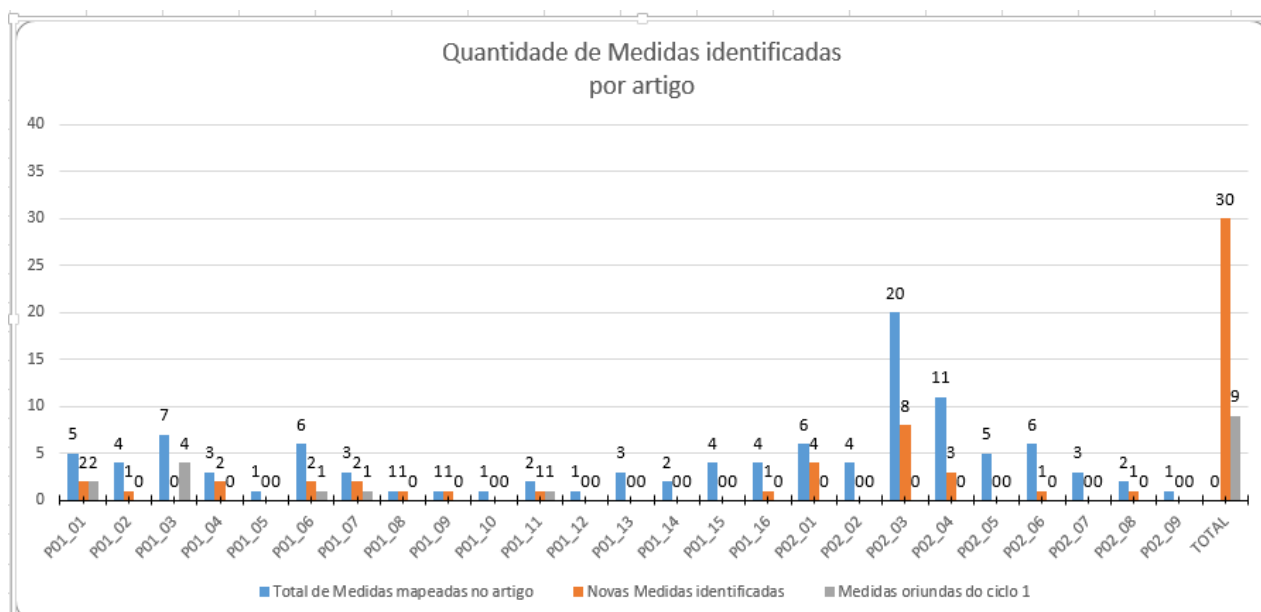


Figura 37 - Gráfico com a quantidade de medidas citadas por artigo e novas medidas

Na Figura 38, é exibida a quantidade de citações para cada medida nos artigos. As medidas são identificadas conforme descrito na Tabela 33.

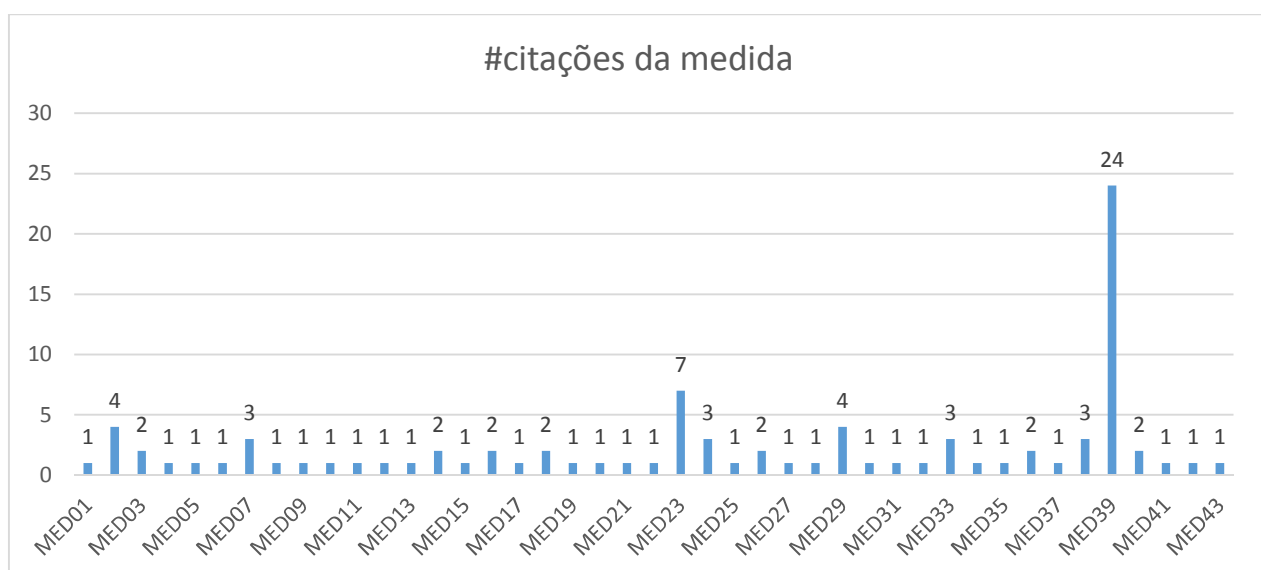


Figura 38 - Gráfico com a quantidade de citações de cada medida

Na Figura 39 é possível observar a quantidade de artigos mapeados com informação sobre medidas por cada ano de publicação, estes artigos listados são os 22 artigos que contribuíram de fato com medidas de testes para o mapeamento somando os Ciclos 1 e 2.

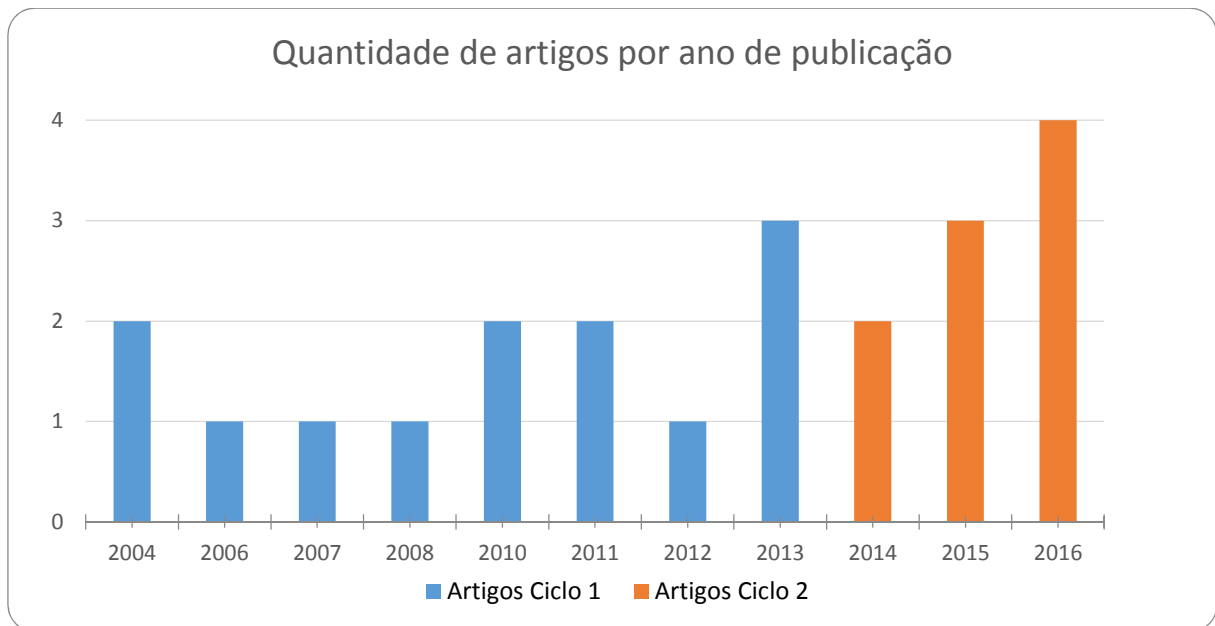


Figura 39 - Gráfico com a quantidade de artigos por ano de publicação

Importante notar que os artigos no escopo do mapeamento, na sua grande maioria, apresentam medidas já bem definidas baseadas em execução dos testes e quantidade de defeitos, porém o objetivo da pesquisa é com base nessa lista, propor indicadores que possam apoiar a gestão de testes no processo de garantia da qualidade dos testes e de software.

I.7.2. QP2: Que indicadores relacionados a teste durante o desenvolvimento ágil de software que podem ser derivados das medidas citadas na literatura

Após a identificação do conjunto de 39 medidas identificadas em ambos os mapeamentos, o conjunto de medidas foi revisado por 2 especialistas em engenharia de software com alto grau de conhecimento para avaliar possíveis medidas duplicadas em razão de utilizarem nomes semelhantes na língua inglesa ou nomes diferentes executando medições semelhantes. Também foram removidas do escopo de análise para criação de indicadores algumas medidas que pela análise dos artigos o cálculo da medida era definido como sendo de forma subjetiva, e não sendo apresentado qualquer tipo de fórmula, não deixando claro como calcular. A Tabela 34 apresenta a associação entre as medidas e os indicadores definidos. Um ID “-“ indica que a medida não foi considerada para compor nenhum indicador na versão final.

Tabela 34 - Lista dos indicadores definidos e respectivas medidas.

ID Artigo	Medida	ID Medida	IDIndicador	Indicador Resultante
P02_01	Avaliação do backlog do produto	MED01	-	
P01_03	Número de testes de aceitação por história	MED19	ID_01	Índice de Cobertura de Testes
P01_06	Histórias de usuário não verificadas	MED10		
P01_16	Cobertura de testes	MED02		
P01_03	Percentual de cobertura de Teste	MED21		
P02_03	Cobertura de testes	MED02		
P02_07	Funcionalidades testadas	MED09		
P02_07	Cobertura de testes de integração	MED23		
P02_07	Cobertura de testes funcionais	MED23		
P02_07	Cobertura de testes unitários	MED23		
P02_08	Cobertura de testes	MED02		
P02_08	Débito de teste	MED43		
P02_10	Cobertura de testes	MED02		
P01_01	Eficiência na remoção de defeitos	MED07	ID_02	Taxa de Defeitos
P01_02	Eficiência na remoção de defeitos	MED07		
P01_02	Qualidade da entrega	MED41		
P01_04	Total de defeitos abertos em um dia específico da release	MED37		
P01_06	Número de defeitos em aberto (não resolvidos)	MED16		
P01_07	Número de defeitos resolvidos	MED40		
P01_11	Eficiência na remoção de defeitos	MED07		
P02_03	Quantidade de defeitos analisados e não resolvidos	MED38		
P02_03	Defeitos abertos	MED39		
P02_03	Quantidade de defeitos abertos	MED39		
P02_03	Quantidade de defeitos no backlog	MED39		
P01_04	Total de defeitos analisados e não removidos	MED38		
P02_03	Quantidade de defeitos novos e abertos	MED39		
P02_04	Quantidade de defeitos analisados e não resolvidos	MED38		
P02_04	Total de defeitos removidos	MED40		
P02_07	Quantidade de bugs abertos	MED16		
P02_08	Defeitos não resolvidos	MED39		
P01_02	Defeitos do cliente	MED04	ID_03	Quantidade de defeitos reportados pelo cliente
P02_03	Quantidade de Defeitos críticos enviados pelo cliente	MED39		
P01_01	Taxa de Defeitos	MED33	ID_04	Densidade de Defeitos
P01_01	Número de defeitos encontrados nos testes	MED39		
P01_02	Taxa de Defeitos	MED33		
P01_02	Total de defeitos encontrados nos testes	MED39		
P01_08	Densidade de defeitos	MED05		
P01_03	Número de defeitos	MED39		
P01_04	Total de defeitos encontrados nos testes	MED39		
P01_05	Total de defeitos encontrados nos testes	MED39		

ID Artigo	Medida	ID Medida	IDIndicador	Indicador Resultante
P01_09	Densidade de falhas por linhas de código	MED06		
P01_09	Número de erros	MED39		
P01_13	Total de defeitos encontrados nos testes	MED39		
P02_01	Severidade dos defeitos	MED03		
P02_01	Quantidade de defeitos	MED39		
P02_02	Total de Defeitos encontrados nos testes	MED39		
P02_03	Quantidade de defeitos/velocidade	MED27		
P02_03	Falhas por iteração	MED39		
P02_03	Número Total de Defeitos após testes	MED39		
P02_03	Quantidade de defeitos descobertos no teste sistema	MED39		
P02_04	Quantidade de defeitos encontrados nos testes	MED39		
P02_05	Complexidade de defeitos	MED03		
P02_05	Número de Defeitos	MED39		
P02_07	Quantidade de defeitos	MED39		
P02_08	Percentual de detecção de defeitos	MED39		
P02_09	Total de Defeitos encontrados nos testes	MED39		
P01_14	Quantidade de funcionalidade que passam nos testes de aceitação	MED29	ID_05	Taxa de funcionalidades que passam nos testes de aceitação automatizados
P01_16	Quantidade de funcionalidade que passam nos testes de aceitação	MED29		
P02_03	Quantidade de funcionalidade que passam nos testes de aceitação	MED29		
P02_04	Quantidade de funcionalidade que passam nos testes de aceitação	MED29		
P02_09	Tempo (em dias) para detecção de defeitos	MED34	ID_06	Tempo entre detecção de defeitos
P02_03	Taxa de aumento de casos de testes entre versões/releases	MED32	ID_07	Taxa de aumento de casos de testes entre releases
P02_04	Quantidade de defeitos a serem removidos (backlog da release)	MED15	-	
P01_03	Número de Casos de testes automatizados	MED14		
P02_03	Número de passos de teste automatizados executados com sucesso	MED14	ID_08	Taxa de casos de testes automatizados
P02_03	Quantidade de testes unitários	MED30		
P02_03	Número de defeitos previstos	MED11	-	
P02_03	Indicador de tendência de defeitos	MED12	-	
P02_04	Prontidão para release (Número de semanas para Release)	MED18	ID_09	Prontidão para release
P02_05	Prontidão para release (Número de semanas para Release)	MED18		
P02_01	Onda de Impacto dos defeitos corrigidos	MED20	-	
P01_06	Tempo para correção do defeito	MED36	ID_10	Tempo médio para correção de um defeito

ID Artigo	Medida	ID Medida	IDIndicador	Indicador Resultante
P01_03	Tempo para corrigir os testes	MED36	ID_11	Tempo para correção dos testes
P01_07	Número de casos de testes concluídos	MED23	ID_12	Taxa de casos de testes executados em relação ao planejado
P02_02	Número de casos de testes executados na fase de testes de sistema na última fase de cada iteração	MED23		
P02_04	Quantidade de casos de testes executados com sucesso	MED24		
P02_03	Percentual de testes executados com sucesso	MED24		
P02_03	Quantidade de casos de testes planejados	MED26		
P02_03	Taxa de testes com falha	MED33		
P02_04	Quantidade de casos de testes executados	MED23		
P02_04	Quantidade de casos de testes falhados	MED25		
P02_04	Quantidade de casos de testes planejados	MED26		
P02_05	Número de casos de testes executados	MED23		
P02_05	Porcentagem de casos de testes executados com sucesso da fase de testes	MED24		
P01_01	Esforço gasto nos testes de aceite	MED08	ID_13	Esforço gasto nos testes
P02_01	Quantidade de defeitos não cobertos pelos casos de testes	MED28	ID_14	Quantidade de defeitos não cobertos por casos de teste
P01_01	Porcentagem de esforço gasto para correção de defeitos	MED22	ID_15	Esforço gasto para correção de defeitos
P02_01	Confiança nos testes	MED42	-	

Na Tabela 35 estão listadas as medidas descartadas para criação de indicadores e a justificativa após análise junto aos especialistas.

Tabela 35 - Lista das medidas descartadas e justificativa

ID Artigo	Medida	ID Medida	Justificativa
P02_01	Avaliação do backlog do produto	MED01	O indicador é calculado com base em medidas subjetivas e não quantitativas, os autores do artigo não proveem informações suficientes para auxiliar a análise do indicador, desta forma essa medida foi descartada.
P02_04	Quantidade de defeitos a serem removidos (backlog da release)	MED15	Não há clareza sobre como medir esse indicador com base no artigo.
P02_03	Número de defeitos previstos	MED11	Não há clareza sobre como medir esse indicador com base no artigo.
P02_03	Indicador de tendência de defeitos	MED12	Não há clareza sobre como medir esse indicador com base no artigo.

ID Artigo	Medida	ID Medida	Justificativa
P02_01	Avaliação do backlog do produto	MED01	O indicador é calculado com base em medidas subjetivas e não quantitativas, os autores do artigo não proveem informações suficientes para auxiliar a análise do indicador, desta forma essa medida foi descartada.
P02_04	Quantidade de defeitos a serem removidos (backlog da release)	MED15	Não há clareza sobre como medir esse indicador com base no artigo.
P02_03	Número de defeitos previstos	MED11	Não há clareza sobre como medir esse indicador com base no artigo.
P02_01	Onda de Impacto dos defeitos corrigidos	MED20	A medida é muito específica, pois é parte do indicador "Avaliação do backlog do produto". Não será considerada para criação de indicadores.
P02_01	Confiança nos testes	MED42	A medida é muito específica, pois é parte do indicador "Avaliação do backlog do produto". Não será considerada para criação de indicadores.

Desta forma, um conjunto de 15 indicadores foram definidos, na Tabela 36 os indicadores são descritos, foi criado na Tabela uma coluna ID para identificar cada indicador sequencialmente, coluna Indicador resultante com o nome do indicador, coluna Sigla com a sigla do indicador, e a coluna Descrição, com um resumo de utilização do indicador.

Tabela 36 - Lista de Indicadores definidos com base na análise das medidas mapeadas

ID	Indicador resultante	Sigla	Descrição
01	Índice de Cobertura de Teste	ICT	Índice de cobertura de teste, considerando total e por tipo de teste (unidade, integração, funcional e aceitação). Este indicador será calculado de forma que o total de requisitos com pelo menos 1 caso de testes sobre o total de requisitos do projeto.
02	Taxa de Defeitos	TD	Taxa de defeitos (abertos, analisados e não resolvidos, removidos) por iteração e por projeto. Total de defeitos (cada tipo) sobre o total de defeitos identificados no projeto/iteração
03	Quantidade de defeitos reportados pelo cliente	QDRC	Quantidade de defeitos reportados pelo cliente (por severidade). Será calculado pelo total de defeitos por severidade por mês/período definido pelo projeto
04	Densidade de Defeitos	DD	Densidade de defeitos (por projeto, por iteração, por complexidade, por severidade). Será calculado pela quantidade de defeitos por severidade ou complexidade acumulado por iteração
05	Taxa de funcionalidades que passam nos testes de aceitação automatizados	TFTA	Taxa de funcionalidades que foram implementadas e verificadas. Calculado pela quantidade de funcionalidades com testes de 0aceitação com sucesso sobre o total de funcionalidades do projeto
06	Tempo entre detecção de defeitos	TDD	Tempo decorrido entre detecções consecutivas de defeitos. Será calculado pelo número de dias até a correção do defeito para cada dia da iteração.
07	Taxa de aumento de casos de testes entre releases	TACT	Taxa de aumento de casos de testes em cada release. Será calculado pelo percentual de aumento de casos de testes entre releases.
08	Taxa de casos de testes automatizados	TCTA	Este indicador representa o número de casos de testes automatizados sobre o total de casos de testes existentes

ID	Indicador resultante	Sigla	Descrição
09	Prontidão para release	RR	Número de semanas previstas para disponibilização da release com base no número de defeitos identificados. O indicador sinalizará em quantas semanas a release estará liberada para implementação
10	Tempo médio para correção de um defeito	TMCD	Este Indicador será calculado para verificar qual o tempo médio de correção de defeitos por cada severidade, considerando ainda o SLA definido pelo projeto
11	Tempo para correção dos testes	TCT	Será calculado para verificar a média de tempo para correção dos defeitos identificados nos testes em cada rodada de testes ou iteração
12	Taxa de casos de testes executados em relação ao planejado	TCEX	Taxa de casos de testes (não executados, executados com sucesso, executados com falha) em relação ao planejado. Este indicador será calculado para cada medida sobre o total de caso de teste planejados para a iteração ou projeto]
13	Esforço gasto nos testes	EGT	Esforço gasto por tipo de teste (unidade, integração, funcional e aceitação). Será calculado pela porcentagem de horas gastas em testes sobre total de horas da iteração
14	Quantidade de defeitos não cobertos por casos de testes	QDNC	Quantidade de defeitos identificados no software pela equipe não capturados pelos casos de teste. Total de defeitos não cobertos por casos de teste
15	Esforço gasto para correção de defeitos	EGCD	Esforço gasto para correção de defeitos por tipo de teste (unidade, integração, funcional e aceitação). Será calculado pela porcentagem de horas gastas na correção dos defeitos sobre total de horas da iteração

1.7.3 Discussão Final e Limitações

Como conclusão deste mapeamento sistemático da literatura, verifica-se a ausência de trabalhos que auxiliem a identificação de indicadores de testes que possam auxiliar na gerência de projetos de testes na tomada de decisão. Este fato motiva estudos mais aprofundados nesta área, para que as organizações consigam identificar as verdadeiras causas de seus problemas com testes e que possam assim mapear os problemas que podem afetar a conclusão dos testes. Alguns poucos indicadores identificados foram criados para projetos específicos com características próprias e que sua utilização se restringe ao contexto do artigo em questão.

A execução e o resultado do mapeamento foram auditados por um outro pesquisador com grande experiência em dois momentos. Primeiro, foram avaliados os procedimentos de seleção dos trabalhos de acordo com o estipulado no protocolo. Durante esta revisão não foram identificadas grandes discrepâncias dos artigos selecionados nas seleções primárias e secundárias desta pesquisa no ciclo 2. Houve apenas quatro publicações onde os artigos passaram para leitura completa na segunda seleção, e o auditor achou desnecessário, porém estes mesmos artigos acabaram sendo descartados na seleção secundária durante leitura completa do artigo, com isso estes artigos não invalidam o resultado do mapeamento. Em um segundo momento, todos os artigos no escopo do mapeamento foram lidos pelo 2º

pesquisador e a catalogação de cada artigo foi revista e discutida. Como fruto dessa auditoria, chegou-se à lista final de medidas apresentada.

As principais limitações deste mapeamento são:

- Não ter o resultado de retorno da string e análise dos critérios de inclusão e exclusão no subconjunto de artigos do mapeamento de Mota e Dias-Neto, 2014 (Ciclo 1). Por este motivo decidimos aplicar o critério de inclusão e exclusão nos 24 artigos selecionados para garantir que os mesmos atendiam de fato o objetivo da pesquisa e confirmar se as medidas listadas realmente tinham origem dos artigos citados pelos autores.
- Não considerar outras bases, além das citadas no mapeamento Ciclo 1. Tendo em vista que o trabalho apresentado foi a replicação do mapeamento de Mota e Dias-Neto (2014) (Ciclo 1), apenas as duas bases de pesquisa utilizadas foram consideradas no Ciclo 2, com isso a ameaça seria de que realizando a busca em outras bases de pesquisa, poderíamos ter identificados medidas que não foram mapeadas nos Ciclos 1 e 2.
- Conforme citado no item I.7.1 há possibilidade de termos medidas utilizadas em projetos tradicionais que sejam aplicáveis em projetos ágeis e não terem sido identificados, devido o foco da busca ter sido em projetos ágeis, porém com base no conjunto de medidas identificadas observa-se um conjunto extenso de medidas que são aplicáveis a projetos “tradicionais” e “ágeis”, minimizando levemente este risco da pesquisa.
- Grande parte dos artigos não detalham o que são as medidas, nem como calculá-las, por isso, muitas vezes, a interpretação dependeu do pesquisador que executou o mapeamento sistemático. Para tratar essa limitação, a definição final das medidas (referente à questão de pesquisa RQ1) foi revista por outro pesquisador e a definição dos indicadores (referente à questão de pesquisa RQ2) foi revista por dois pesquisadores com conhecimento em Engenharia de Software.

I.8 Informações Extraídas das Publicações Seleccionadas

Nesta seção, são apresentadas as informações extraídas das 9 publicações seleccionadas através da execução busca no período de 2014 até dezembro de 2016 (Ciclo 2) e dos 13 artigos do mapeamento de Mota e Dias-Neto (2014) (Ciclo 1).

I.8.1 Publicações ciclo 1

Dados da publicação	
Título: [1]	Analyses of an agile methodology implementation
Autor (es):	Ilieva, S., Ivanov, P. and Stefanova, E.
Ano da publicação:	2004
Referência completa:	Ilieva, S., Ivanov, P. and Stefanova, E. (2004) "Analyses of an agile methodology implementation", Proceedings of the 30th EUROMICRO Conference, pp. 326-333.
Resumo da publicação	
O artigo fala sobre uma nova abordagem chamada eXPERT que combina princípios da metodologia ágil XP (eXtreme Programming) com PSP (Personal Software Process) para gerenciamento de projetos de desenvolvimento. A abordagem eXPERT possui os seguintes objetivos de negócios definidos: • Aumentar produtividade em aproximadamente em 20% • Reduzir taxa de defeitos em até 30% • Reduzir execuções extras do projeto em até 15% Esta abordagem eXPERT combina práticas de XP e PSP, e é descrita através dos processos abaixo: • Gerência de requisitos de cliente • Gerência de projeto • Desenho/Projeto • Código • Teste Para cada processo, são descritos objetivos, entradas, atividades, saídas dos processos, critérios de finalização e Medição. Inicialmente foi feito um "gap analysis" para avaliar e adequar a nova abordagem ao processo corrente da organização baseado na normal internacional (ISO 9001). Foi gerado um guia com as adaptações e treinamento provido aos membros do projeto. Um piloto foi realizado utilizando a nova abordagem.	
Medidas citadas no artigo (independente de ser de teste)	
Defect rate – Taxa de defeitos Number of defects – Número de defeitos encontrados nos testes Effort spent on Acceptance Testing – Esforço gasto nos testes de aceite Defect removal efficiency – Eficiência na remoção de defeitos Percentage of the effort spent for bug fixing - Porcentagem de esforço gasto para correção de defeitos Productivity Metric – Métrica de produtividade Relative schedule deviation – Desvio relativo de cronograma Project cost change – Mudança nos custos do projeto Customer and developer satisfaction – Satisfação do cliente e desenvolvedores Relative Cost Deviation - Desvio relativo de custo	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
O artigo descreve um dos sete pilotos realizado com diferentes SMEs (Subject Matter Experts) em projetos de desenvolvimento de software em uma empresa da Bulgária. SMEs – Profissionais especialista no assunto em discussão.	
Exemplo de utilização	
Foi feito apenas no projeto piloto que tinha característica de ter uma baixa taxa de defeitos em um curto espaço de tempo. Mostrou que é relativamente simples de ser aplicado em times e organizações pequenas ajudados pela comunicação direta (que é relatado como sendo fator muito importante para esta abordagem).	

Este piloto foi realizado em um projeto corrente e comparado com outro projeto de característica similar que utilizou processo baseado na norma internacional ISO 9001, porém não há detalhes das comparações realizadas no artigo, apenas a citação que houve.
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis
Em projetos que utilizem metodologia ágil, mas que possuam equipes bem disciplinadas de forma profissional para executar o proposto na abordagem.
Evidência de uso efetivo
No projeto citado a abordagem resultou em aumento da produtividade (41%), porém houve um aumento de código novo expressivo para este projeto, sem reuso de código. Este fator elevou a produtividade além do esperado (20%). Outro ponto avaliado foi a redução de defeitos, que atingiu quase metade do esperado (30%). A redução foi em 13%. Quanto à redução de esforço nas reexecuções, foi atingido 11.5%, que é bem próximo aos 15% propostos na abordagem.
Medidas relacionadas a teste
Defect rate – Taxa de defeitos Number of defects – Número de defeitos encontrados nos testes Effort spent on Acceptance Testing – Esforço gasto nos testes de aceite Defect removal efficiency – Eficiência na remoção de defeitos Percentage of the effort spent for bug fixing - Porcentagem de esforço gasto para correção de defeitos
Fórmulas para cálculo das medidas de teste
Defect rate = (Effort spent for bug fixing / Effort) x 100

Dados da publicação	
Título:[2]	Embedded Agile Project by the Numbers with Newbies
Autor (es):	Schooenderwoert, N. V.
Ano da publicação:	2006
Referência completa:	Schooenderwoert, N. V. (2006) “Embedded Agile Project by the Numbers with Newbies”, Agile 2006 conference report.
Resumo da publicação	
Este artigo apresenta métricas detalhadas de um projeto de desenvolvimento de software com metodologia ágil onde uma equipe de desenvolvimento recém-formada desenvolveu um novo produto de software desde o início. O produto desenvolvido foi uma ferramenta para dispositivo móveis para uma indústria de produtos agrícolas para monitoramento de grãos. O artigo cita como a equipe de desenvolvimento era inexperiente no início do projeto e das dificuldades durante o desenvolvimento do produto a cada release.	
Medidas citadas no artigo (independente de ser de teste)	
Defect Rate – Taxa de Defeitos Defects to Customer – Defeitos do cliente Number of defects – Total de defeitos encontrados nos testes Defect removal efficiency – Eficiência na remoção de defeitos Lines of code – linhas de código Effort expended – Esforço gasto Size of software build – Tamanho de software construído Schedule compliance – Conformidade de cronograma Quality of the deliverable – Qualidade da entrega. Productivity– ESLOC/h (Effective source lines of code) – Linhas do código fonte efetivas.	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
Estas medidas foram coletadas pela equipe de desenvolvimento durante o projeto	
Exemplo de utilização	
N/A	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
O software citado foi desenvolvido para uma aplicação embarcada dentro de uma indústria específica. No artigo não fica claro se em outros contextos poderiam utilizar o mesmo software.	
Evidência de uso efetivo	
As medidas foram coletadas durante as fases de projeto e algumas informações com os dados são disponibilizadas.	

Medidas relacionadas a teste	
Defect Rate – Taxa de Defeitos	
Defects to Customer – Defeitos do cliente	
Number of defects – Total de defeitos encontrados nos testes	
Defect removal efficiency – Eficiência na remoção de defeitos	
Quality of the deliverable – Qualidade da entrega.	
Fórmulas para cálculo das medidas de teste	
Defect rate – Taxa de Defeitos = $DD = \text{TOT_DEF} / \text{ESLOC (effective source lines of code)} * 100$	
Defects to Customer – Defeitos do cliente = $\text{Total FP} * \text{defects per FP} * (1.0 - \text{defect removal efficiency})$	
Defect removal efficiency – Eficiência na remoção de defeitos = $\text{TOT_DEF REMOVED (BEFORE)} / \text{TOT_DEF} * 100$	

Dados da publicação	
Título:[3]	AM-QuICk: A Measurement-Based Framework for Agile Methods Customisation
Autor (es):	Ayed, H., Habra, N. and Vanderose, B.
Ano da publicação:	2013
Referência completa:	Ayed, H., Habra, N. and Vanderose, B. (2013) “AM-QuICk: A Measurement-Based Framework for Agile Methods Customisation”, In Proc. IWSM/Mensura 2013: 71-80.

Resumo da publicação	
O artigo descreve um framework criado para apoiar projetos com metodologia ágil na adoção da metodologia e customização. O nome do framework é AM-QuICk (Agile Methods Quality-Integrated Customisation Framework). O framework é baseado nos modelos SME (Situation Method Engineering) e QIP (Quality Improvement Paradigm) para medição de conceitos e apoio a tomada de decisões durante o processo de adoção e evolução do modelo. O modelo SME foca em reusar partes do modelo atual para construção de novos modelos customizados para a organização. QIP foca na melhoria e qualidade do processo. O artigo cita as dificuldades de customização de modelo, e reforça que o modelo AM-QuICk foca em reutilização de experiências passadas para melhoria do processo. Os passos deste modelo são: 1-Estudo do contexto 2-Desenho do método ágil customizado 3-Implementar e avaliar 4-Capitação e utilização em novos projetos do modelo customizado Inclui medição nos processos na parte de gerenciamento, execução e planejamento	

Medidas citadas no artigo (independente de ser de teste)	
Code dependencies – Dependências de código	
Abstractness – Abstrações	
Automation Tests – Testes automatizados	
Cyclomatic complexity – Complexidade ciclomática.	
Coding standards violation – Violação dos padrões de codificação	
Number of refactors – Número de refatorações	
Code coverage – Cobertura de código	
Code duplication – Duplicação de código	
Percentage of test coverage - Percentual de cobertura de Teste	
Number of automated test cases - Número de Casos de testes automatizados	
Number of acceptance tests per story - Números de testes de aceitação por história	
Number of defects – Número de defeitos	
Backlog size - Planejamento do tamanho do Backlog	
Time to fix tests - Tempo para corrigir os testes	
Product ownership health – Saúde da propriedade do produto	
Team technical health – Saúde da equipe técnica	
Agile Knowledge degree – Nivel de conhecimento ágil	
Actual velocity – Velocidade Real	
Planned velocity – Velocidade planejada	
Number of stories planned – Número de histórias planejadas	

Number of stories accepted- Número de histórias aceitas
Release Progress Charts – Gráficos de progresso
Value Delivered – Valor entregue
On-time delivery – Entregas no prazo
Feature Business value estimation – Estimativa do valor das características do negócio
Work Unit estimation – Estimativa de unidade de trabalho
Contexto em que a medida é (ou pode ser) utilizada em projetos
Estas medidas foram utilizadas na definição do novo modelo customizados para coleta das métricas do framework
Exemplo de utilização
Foi implementado em um projeto de TI dentro de uma organização pública da Bélgica
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis
Em projetos de TI que necessitam de customização dos processos para metodologia ágil
Evidência de uso efetivo
Não fica claro no artigo uma vez que o artigo cita a necessidade de enriquecem o meta-modelo ágil.
Medidas relacionadas a teste
Percentage of test coverage - Percentual de cobertura de Teste
Number of automated test cases - Número de Casos de testes automatizados
Number of acceptance tests per story - Números de testes de aceitação por história
Number of defects – Número de defeitos
Time to fix tests - Tempo para Corrigir os testes
Fórmulas para cálculo das medidas de teste
N/A

Dados da publicação	
Título:[4]	Measuring the Impact of Scrum on Product Development at Adobe Systems
Autor (es):	Green, P.
Ano da publicação:	2011
Referência completa:	Green, P. (2011) “Measuring the Impact of Scrum on Product Development at Adobe Systems”, In: 44th Hawaii International Conference on System Sciences, p.1-10, 2011.
Resumo da publicação	
O artigo cita a implementação do modelo ágil SCRUM dentro da organização Adobe Systems. Cita 3 medidas utilizadas nos projetos: 1-medidas subjetivas coletadas dos praticantes de SCRUM 2- contagem de defeitos como número de defeitos, taxa de defeitos, curva de informação de defeitos ciclo à ciclo, 3-Cartão de avaliação baseado em pesquisas feita com os clientes. Foi identificado que em cada tipo de medida coleta há falta de informação para ajudar na tomada de decisão. Parte da pesquisa descrita foi para avaliar como as equipes utilizam o SCRUM. Esta avaliação foi feita através de um formulário desenvolvido dentro da própria Adobe, a primeira parte da pesquisa foi para avaliar a efetividade do SCRUM e a segunda um cartão de avaliação de várias áreas de processo do SCRUM. Também foram coletadas informações sobre defeitos de diferentes sistemas, notou-se que 2/3 das equipes participantes não possuíam esta informação por não possuírem histórico de projetos iniciados, ou porque não configuraram seus projetos para coletar esta informação. Na última parte foi avaliado a satisfação dos clientes por meio de questionário de avaliação.	
Medidas citadas no artigo (independente de ser de teste)	
Total Defects Found - Total de defeitos encontrados nos testes Number Defects Deferred – Total de defeitos analisados e não removidos Number of open defects on any given day of the release – Total de defeitos abertos em um dia específico da release.	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
Avaliação de vários projetos coletando medidas sobre defeitos e avaliando o progresso	
Exemplo de utilização	
Quantidade de defeitos descobertos e defeitos não resolvidos pelas equipes	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Há indícios que as medidas citadas possam ser aplicadas em outros projetos por serem medidas básicas para defeitos.	
Evidência de uso efetivo	

Sim, os valores em percentual dos defeitos são descritos no artigo
Medidas relacionadas a teste
Total Defects Found - Total de defeitos encontrados nos testes Number Defects Deferred – Total de defeitos analisados e não removidos Number of open defects on any given day of the release – Total de defeitos abertos em um dia específico da release.
Fórmulas para cálculo das medidas de teste
N/A

Dados da publicação	
Título:[5]	Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs
Autor (es):	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A.
Ano da publicação:	2011
Referência completa:	Ruiz, J.C., Osorio, Z.B., Mejia, J., Munoz, M., Chavez, A.M. and Olivares, B.A. (2011) “Definition of a hybrid measurement process for the models ISO/IEC 15504-ISO/IEC 12207:2008 AND CMMI-DEV1.3, SMEs”, Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2011 IEEE, 421-426..
Resumo da publicação	
O artigo descreve uma proposta de medição híbrida para os modelos ISO/IEC 15504-ISO/IEC 12207:2008 e CMI Dev 1.3 direcionados a pequenas e médias empresas. Inicialmente foi feito uma rastreabilidade entre os dois modelos da seguinte forma: 1-Seleção do modelo e padrões a serem analisados. 2-Escolher um modelo como referência da norma internacional (ISO) – as normas escolhidas foram ISO/IEC 15504-ISO/IEC 12207:2008 3-Escolher qual processo será analisado dos modelos, processos equivalentes 4-Definir nível de detalhamento que será utilizado de cada processo 5-Identificar as similaridades entre os modelos 6-Exibir resultados Para definir como usar cada processo foi analisado o modelo SCRUM por serem mais fáceis de adoção e entendimento. Ao final o novo modelo exhibe quais práticas foram definidas e objetivos e processos.	
Medidas citadas no artigo (independente de ser de teste)	
Number of daily meeting issues - Número de problemas durante reuniões diárias Team member satisfaction - Satisfação da equipe Code Size - Tamanho do código Performed tasks - Tarefas realizadas Number of defects during review meeting - Total de defeitos encontrados nos testes	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas foram escolhidas para fazerem parte do processo de medição no novo modelo.	
Exemplo de utilização	
Não há exemplo do piloto, mas no artigo é descrito os processos e práticas utilizadas.	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Qualquer projeto que utilize modelos ISO baseado nas normas ISO/IEC 15504-ISO/IEC 12207:2008 ou CMMI podem experimentar este modelo	
Evidência de uso efetivo	
O modelo híbrido foi utilizado como piloto em uma organização e os resultados iniciais motivaram a organização em manter a linha de pesquisa e desenvolver mais o modelo	
Medidas relacionadas a teste	
Number of defects during review meeting - Total de defeitos encontrados nos testes	
Fórmulas para cálculo das medidas de teste	
N/A	

Dados da publicação	
Título:[6]	Metrics Functions for Kanban Guards
Autor (es):	Heidenberg, J. and Porres, I.
Ano da publicação:	2010
Referência completa:	Heidenberg, J. and Porres, I. (2010) “Metrics Functions for Kanban Guards”, In: IEEE International Conference and Workshops on Engineering of Computer-Based Systems. IEEE Computer Society.
Resumo da publicação	
O artigo descreve métricas para desenvolvimento de software para que as equipes ágeis possam tomar decisões sobre a qualidade do produto de software. É proposto uma ferramenta conceitual chamada de “kanban guards” que coleta e analisa as métricas em projetos com métodos ágeis/lean e gera avisos de como agir em cada situação que não esteja de acordo com o planejado. A metodologia lean foi criada nos anos 80 dentro da Toyota construtora de veículos, também chamado de produção just-in-time, onde as partes/peças são produzidas conforme necessidade das linhas de montagem. O fluxo básico seria: 1- Medição 2- Análise ou interpretação 3- Aviso ou comunicação É proposto uma função de medição kanban que analisa cada ponto e divide por setores (verde, vermelho e laranja). Os avisos podem ser recomendações de como tratar um problema ou priorizar resolução de partes de código. Se algo receber pontuação vermelha, todo o processo deve ser parado.	
Medidas citadas no artigo (independente de ser de teste)	
Lines of code- Contador de linhas de código Lack of Cohesion - Falta de coesão Defect turn-around time - Tempo para correção do defeito cycle time – Tempo do ciclo Unverified user stories - Histórias de usuário não verificadas Accuracy of estimation - Precisão da estimativa Number of open bug reports – Numero de defeitos em aberto (não resolvidos) Throughput – Fluxo de tarefas	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas fazem parte de um processo de medição para avaliar o desenvolvimento de um projeto de software, como exemplo de medidas que podem ser extraídas.	
Exemplo de utilização	
N/A	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Para apoio a qualquer projeto com metodologia ágil lean que queira adequar as funções guardians, para sinalizar problemas projeto.	
Evidência de uso efetivo	
Existe somente a definição dos conceitos sobre a função “Kanban Guards”	
Medidas relacionadas a teste	
Defect turn-around time - Tempo para correção do defeito Unverified user stories - Histórias de usuário não verificadas Number of open bug reports – Numero de defeitos em aberto (não resolvidos)	
Fórmulas para cálculo das medidas de teste	
Não há.	

Dados da publicação	
Título:[7]	A development of the effectiveness evaluation model for agile software development using the balanced scorecard
Autor (es):	Atiwithayaporn, S. and Rivepiboon, W.
Ano da publicação:	2013
Referência completa:	Atiwithayaporn, S. and Rivepiboon, W. (2013) “A development of the effectiveness evaluation model for agile software development using the balanced scorecard”, IMECS.
Resumo da publicação	

O artigo fala sobre um modelo baseado na norma internacional ISO/IEC12207 e no framework chamado Balanced Scorecard (BSC) para criar um plano de ação. Este plano de ação foi adicionado ao framework do BSC composto de objetivos, medidas e processos. O BSC é uma ferramenta para avaliar alinhamento de projetos com alinhamento organizacional/objetivo de negócio.	
Medidas citadas no artigo (independente de ser de teste)	
Time from start to end a system process – Tempo de resposta do sistema Average time from start system to downtime – Disponibilidade média do sistema Tasks finished within time – Tarefas finalizadas no prazo Survey the satisfy customer – Pesquisa de satisfação do cliente Number of Bugs resolved – Número de defeitos Resolvidos Number of test cases executed – Número de casos de testes concluídos Percentage of functions follow requirements - Percentual de funcionalidades compatíveis aos requisitos Bugs Resolved Rate- Taxa de remoção de defeitos Tasks finished per a service – Tarefas finalizadas por serviço	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas descritas fazem parte do cálculo da medida <i>bugs resolved</i> que faz parte do plano scorecard balanced	
Exemplo de utilização	
Não há.	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Em organização que utilizem o balanced scorecard e que tenham foco em transforma as estratégias de negócios em plano de ação	
Evidência de uso efetivo	
Os autores não apresentaram exemplo de uso efetivo.	
Medidas relacionadas a teste	
Number of Bugs resolved – Número de defeitos Resolvidos Number of test cases executed – Número de casos de testes concluídos Bugs Resolved Rate- Taxa de remoção de defeitos	
Fórmulas para cálculo das medidas de teste	
$\text{Bugs resolved rate} = \text{Taxa de defeitos resolvidos} = \frac{\text{Number of bugs resolved} * 100}{\text{Number of teste cases executed}}$	

Dados da publicação	
Título:[8]	A metamodel-based approach for customizing and assessing agile methods
Autor (es):	Ayed, H., Vanderose, B. and Habra, N.
Ano da publicação:	2012
Referência completa:	Ayed, H., Vanderose, B. and Habra, N. (2012) “A metamodel-based approach for customizing and assessing agile methods”, In: The International Conference on the Quality of Information and Communications Technology (QUATIC).
Resumo da publicação	
O artigo descreve como customizar modelos ágeis para problemas específicos de uma organização que utiliza os métodos XP e SCRUM nos seus projetos. Para esta abordagem foi utilizado o paradigma SME (Situational Method Engineer) que foca em conceitos de medição e gerenciamento para construção de software com métodos ágeis. Foi construído ao final um meta-modelo com classes e elementos com possíveis ações.	
Medidas citadas no artigo (independente de ser de teste)	
Release Schedule – Cronograma de entrega Release Design Instability – Instabilidade do desenho da entrega Defect Density – Densidade de defeitos	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas são descritas como parte do processo de medição do meta-modelo desenvolvido	
Exemplo de utilização	
Não há	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Este modelo foi desenvolvido para um cenário de organização específico, porém acredita-se que podem ser utilizados em outras organizações que usem métodos XP ou SCRUM .	

Evidência de uso efetivo
Não
Medidas relacionadas a teste
Defect Density – Densidade de defeitos
Fórmulas para cálculo das medidas de teste
Não há

Dados da publicação	
Título:[9]	Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process
Autor (es):	Mahnic, V. and Vrana, I.
Ano da publicação:	2007
Referência completa:	Mahnic, V. and Vrana, I. (2007) “Using stakeholder driven process performance measurement for monitoring the performance of a Scrum based software development process”, Electrotechnical Review.

Resumo da publicação
O artigo especifica um plano de medição que foca em monitorar e melhorar o processo de desenvolvimento de software considerando a visão dos patrocinadores (stakeholders). Foram descritos Indicadores para alguns objetivos definidos como: Informação sobre desempenho focado no projeto e orçamento financeiro. E também em melhorias na qualidade de desenvolvimento do software.

Medidas citadas no artigo (independente de ser de teste)
Error Density – Densidade de falhas por linhas de código Size of code – Tamanho do código Costs of rework – Custo do retrabalho Work spent on day – Trabalho feito no dia The length of the sprint – Tamanho da sprint Number of errors – Número de erros SPI - Schedule Performance Index – Índice de desempenho de prazo CPI - Cost Performance Index – Índice de desempenho de custo Amount of rework – Retrabalho Ratio between the work spent and the decrement– Relação entre o trabalho gasto e o decréscimo Fulfillment of Scope – Cumprimento do escopo The average amount of overtime – Tempo médio de atraso The average number of projects the employees work in parallel – Media de projeto com recursos trabalhando em paralelo. Qualitative evaluation of working – Avaliação qualitativa do trabalho Qualitative evaluation of customer satisfaction - Avaliação qualitativa da satisfação do cliente.

Contexto em que a medida é (ou pode ser) utilizada em projetos
As medidas foram citadas como parte do plano de medição sugerido

Exemplo de utilização
Não há descrição

Indícios de que as medidas podem ser utilizadas em outros projetos ágeis
Projetos que visam melhoria contínua no seu processo de desenvolvimento software

Evidência de uso efetivo
O artigo cita que as medidas foram coletadas em reuniões com as equipes, porém não há números descritos.

Medidas relacionadas a teste
Error Density – Densidade de falhas por linhas de código
Number of errors – Número de erros

Fórmulas para cálculo das medidas de teste
Error Density = Number of error / Kloc

Dados da publicação	
Título:[11]	Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery
Autor (es):	Brown A., Ambler S. and Royce W
Ano da publicação:	2013
Referência completa:	Brown A., Ambler S. and Royce W. (2013) "Agility in Scale: Economic Governance, Measured Improvement, and Disciplined Delivery", ICSE.
Resumo da publicação	
O artigo trata de como melhorar a adoção de agilidade em projetos de desenvolvimento de software. O artigo cita 3 princípios fundamentais: governança econômica, melhoria baseada em medições, e entrega disciplina de agilidade O artigo sugere algumas práticas para melhoria destes princípios. Algumas das melhorias propostas são: a) Teste de integração deve preceder testes unitários – Na prática seria em paralelo para ganhar tempo na agilidade b) Agilidade e controle devem ser complementares – cita a necessidade de integração entre agilidade e governança.	
Medidas citadas no artigo (independente de ser de teste)	
Defect removal efficiency- Eficiência na remoção de defeitos Test time - Tempo de teste Resources Estimates – Estimativa de recursos Late projects – Projetos em atraso Staff Morale – Moral da equipe Client Satisfaction – Satisfação do cliente Overhead – Gasto extra Change Efficiency – Eficiência de mudança Requirements Churn – Mudança de requisitos On-time projects – Projetos no prazo Cancelled projects – Projetos cancelados Integration Stability – Estabilidade de integração	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas fazem parte do framework no contexto de medição	
Exemplo de utilização	
Não há, apenas um modelo descrevendo os princípios citados	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Projetos que utilizem metodologia ágil	
Evidência de uso efetivo	
Não identificado	
Medidas relacionadas a teste	
Defect removal efficiency- Eficiência na remoção de defeitos Test time - Tempo de teste	
Fórmulas para cálculo das medidas de teste	
Não há.	

Dados da publicação	
Título:[13]	Refactoring the development process: experiences with the incremental adoption of agile practices
Autor (es):	Hodgetts, P.
Ano da publicação:	2004
Referência completa:	Hodgetts, P. (2004); Refactoring the development process: experiences with the incremental adoption of agile practices. IEEE Computer Society. In: Agile Development Conference.
Resumo da publicação	
O artigo cita como projetos sem abordagem de processos, mas com taxa de sucesso baixa.	

Cita que com adoção de processos incrementais a experiência foi bem melhor em resultados. A abordagem começou com adoção da metodologia ágil XP, de forma incremental, utilizado TDD (test driven development) para fase de avaliação da qualidade (QA), Pair Programming (programação em pares) e refatoração para avaliação do código.	
Medidas citadas no artigo (independente de ser de teste)	
Defects counts - Total de defeitos encontrados nos testes Size of the backlog - Tamanho do Backlog Iteration Velocity – Pontos de história concluídos por Iteração	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas foram coletadas durante as fases de testes na adoção da abordagem proposta	
Exemplo de utilização	
Não há	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Na adoção da metodologia ágil em desenvolvimento de software	
Evidência de uso efetivo	
Não há, apenas recomendações de utilização em outros projetos	
Medidas relacionadas a teste	
Defects counts - Total de defeitos encontrados nos testes	
Fórmulas para cálculo das medidas de teste	
Não há	

Dados da publicação	
Título:[14]	Complementing Measurements and Real Options Concepts to Support Iteration Decision-Making in Agile Projects
Autor (es):	Hodgetts, P.
Ano da publicação:	2008
Referência completa:	Racheva, Z., Daneva, M. and Buglione, L. (2008) "Complementing Measurements and Real Options Concepts to Support Iteration Decision-Making in Agile Projects," Proc. 34th Euromicro Conf. Software Engineering and Advanced Applications, pp. 457-464.
Resumo da publicação	
O trabalho apresenta uma proposta complementar de utilização de técnicas e conceitos reais de medição para apoio a tomada de decisões dos clientes na avaliação de requisitos.	
Medidas citadas no artigo (independente de ser de teste)	
System Design Instability – Instabilidade do design do sistema Earned Business Value – Valor do negócio agregado Whole ammount of work – Quantidade de trabalho Story based level - Pontos de História concluídos por iteração Product Size Unit – Unidade de tamanho de projeto Function Point Productivity – Produtividade de pontos de função Drag Factor Trend – Tendência do fator de tração Running Tested Features – Quantidade de funcionalidade que passam nos testes de aceitação	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas são aplicadas de forma conceitual na avaliação da proposta	
Exemplo de utilização	
Não há	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Abordagem para apoio na validação dos requisitos de projeto pelos clientes	
Evidência de uso efetivo	
Não há evidência descrita.	
Medidas relacionadas a teste	
Running Tested Features – Quantidade de funcionalidade que passam nos testes de aceitação	
Fórmulas para cálculo das medidas de teste	
N/A	

Dados da publicação	
Título:[16]	Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?
Autor (es):	Ktata, O. and Levesque, G.
Ano da publicação:	2010
Referência completa:	Ktata, O. and Levesque, G. (2010) "Designing and implementing a measurement program for Scrum teams: what do agile developers really need and want?", Proceedings of the Third Conference on Computer Science and Software Engineering, p.101-107.
Resumo da publicação	
O artigo critica que Velocity (velocidade) é uma das medidas mais utilizadas em projetos com metodologia ágil e alerta para outras medidas que podem trazer um ganho de desempenho no projeto de desenvolvimento de software. Cita que as medidas mais visualizadas são Value Delivered e Technical Debt. O artigo propõe na sua abordagem um programa de medição que ajuda na identificação de indicadores e objetivos para melhoria do gerenciamento do projeto. É demonstrado como foram coletadas as medidas e gerado alguns indicadores.	
Medidas citadas no artigo (independente de ser de teste)	
Earned Value – Valor agregado Running tested Features – Quantidade de Funcionalidades que passam nos testes de aceitação Number of tasks forgotten during planning session-Tarefas esquecidas durante o planejamento Test coverage – Cobertura de testes Real size at completion of user story – Tamanho real da história User story cycle time – Tempo entre identificação e implementação da história Visibility of Debt – Visibilidade do débito técnico Transparency on collaboration issues – Transparência em questões de colaboração Team efficiency – Eficiência da equipe Individual performance – Desempenho Individual Team and individual motivational level variations – Variação nos níveis motivacionais da equipe e individual Team performance: - Desempenho da equipe Maturity-Maturidade Estimation of user stories– Estimativas de histórias de usuários Estimation of tasks decomposition – Estimativa das tarefas de decomposição Adherence to the process and good practices – Aderência aos processo e boas práticas Work in progress– Progresso do trabalho Indicator about risky user stories – Indicador de risco das histórias de usuário Visibility of Business Value – Visibilidade de valor do negócio Visibility on support performance – Visibilidade no desempenho do suporte Financial aspects of projects, ROI – Aspectos financeiros dos projetos Customer satisfaction indicator – Indicador de satisfação do cliente	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
Na avaliação de uma abordagem diferente para criar indicadores de monitoramento.	
Exemplo de utilização	
O programa de medição descreve como cada indicador proposto foi avaliado pelos representantes do projeto	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Qualquer projeto de metodologia ágil	
Evidência de uso efetivo	
Apenas as informações exibidas referentes ao programada de medição	
Medidas relacionadas a teste	
Test coverage – Cobertura de testes Running tested Features – Quantidade de Funcionalidades que passam nos testes de aceitação	
Fórmulas para cálculo das medidas de teste	
Não há	

I.8.2 Publicações Ciclo 2

Dados da publicação	
Título: [1]	Product backlog rating: a case study on measuring test quality in scrum
Autor (es):	Kayes I., Sarker M., Chakareski J.
Ano da publicação:	2016
Referência completa:	Kayes I., Sarker M., Chakareski J.. (2016). " Product backlog rating: a case study on measuring test quality in scrum." Computer Science and Engineering, University of South Florida, Tampa, FL, USA
Resumo da publicação	
O artigo apresenta métricas que são utilizadas para definir um Indicador que mede a qualidade do processo de testes em um projeto com metodologia ágil SCRUM, o nome do indicador proposto para avaliar o processo de testes é PBR (Product Backlog Rating). O artigo não está avaliando o processo SCRUM, a intenção é avaliar o processo de testes em um projeto que utilize SCRUM. A avaliação foi realizada em uma empresa provedora de soluções localizada na Dinamarca chamada SoftwarePeople. O PBR considerou na avaliação dois parâmetros: nível de complexidade do produto (PCL-Product Complexity Level) e Taxa de avaliação de teste (TAR-Test Assessment Rating) baseado em valores numéricos definidos pelos autores de forma subjetiva. O artigo explica ainda como funciona o processo SCRUM, geração de artefatos, backlog do projeto, posições de trabalho e fases do projeto SCRUM, detalha como PCL e TAR é extraído do Product Backlog (PBL). O PBL é uma lista de requerimentos priorizados do produto. Os autores definem subjetivamente os índices, chamados de PBR score: 5 – Excelente, 4- Bom, 3-Moderado, 2-Ruim, 1-Pior. Os autores citam a dificuldade de definição do PBR, uma vez que foram utilizadas medidas não fáceis de serem quantificadas. Citam no final que as medidas deveriam idealmente ser calculadas de forma objetiva e não de forma subjetiva.	
Medidas citadas no artigo (independente de ser de teste)	
• Velocity – Velocidade de realizar as tarefas • Number of completed story points per sprint - Número de histórias completas por sprint • Implemented development size for each iteration. – Tamanho de cada implementação de desenvolvimento • Product Backlog Rating – PBR – Avaliação do backlog do produto (mede a qualidade do processo de teste) • Severity of the bugs – Severidade dos defeitos • Bug fixing ripple impact - Onda de Impacto dos defeitos corrigidos • Numbers of Bugs missed by test case - Defeitos não cobertos pelos casos de teste • Requirements change – Mudança de requisitos • Bugs count – Quantidade de defeitos • Test Confidence – Confiança nos testes	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
Os autores citam a definição do PBR, com base em dois critérios de avaliação (PCL e TAR) que são medidas não fáceis de quantificar. A fórmula para o PBR foi definida e aplicada a um projeto dentro de uma única organização seguindo o processo SCRUM definido.	
Exemplo de utilização	
Os autores apresentam uma Tabela com dados retirados de um projeto e aplica a formula do PBR, e definem valores de interpretação assim como um limite para qualificar o projeto.	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Pelo texto, é possível entender que qualquer tipo de projeto de desenvolvimento com metodologia ágil SCRUM poderá utilizar estas medidas, porém dificuldades poderão surgir uma vez que as características podem ser diferentes entre organizações, assim como dados de projetos.	
Evidência de uso efetivo	
As medidas e fórmulas foram apresentados usando dados empíricos.	
Medidas relacionadas a teste	
1- Product Backlog Rating – PBR – Avaliação do backlog do produto (mede a qualidade do processo de teste) 2-Severity of the bugs – Quantidade de defeitos por Severidade. 3-Bug fixing ripple impact – Onda de Impacto dos defeitos corrigidos 4- Numbers of Bugs missed by test case – Quantidade de defeitos não cobertos pelos casos de testes 5- Bugs count – Quantidade de defeitos	

6- Test Confidence – Confiança nos testes

Fórmulas para cálculo das medidas de teste

$$PBR = \frac{\sum_{i=1}^{i \in N} PCL_i * TAR_i}{\sum_{i=1}^{i \in N} PCL_i}$$

PCL -Product Complexity Level – Nível de complexidade do produto – É baseado nos seguintes fatores:

- CBLP – Complexity of business logic of the Product - PBL - Complexidade da lógica de negócio do produto
- NTD- Necessity of test data – Necessidade de dados de teste
- TE- Test estimation – Estimativa de testes
- PID- PBL item inter-dependency –dependências internas do backlog do produto
- PED- PBL item external-dependency –dependências externas do backlog do produto
- TEC- Test execution complexity- Complexidade de execução do teste
- LGUI - Logic on graphical user interface (GUI) – Lógica na interface gráfica
- BA - Bug assumptions – Suposições de defeitos

$$PCL_k = \frac{\sum_{i=1}^{i \in P} FP_i * W_i}{\sum_{i=1}^{i \in P} W_i},$$

TAR - Test Assessment Rating - Taxa de avaliação de teste - É baseado nos seguintes fatores:

- BC- Bugs count – Contador de defeitos
- SB - Severity of the bugs – Severidade dos defeitos
- BFRI - Bug fixing ripple impact – Onda de Impacto dos defeitos corrigidos
- NBMTTC - Numbers of Bugs missed by test case – Defeitos não cobertos pelos casos de teste
- RC - Requirements change – Mudança de requisitos

$$TAR_k = \sum_{i=1}^{i \in T} \frac{FT_i}{T}$$

Os valores para PCL e TAR foram definidos entre 1-5 pelos autores.

O PCL é definido após definição das tarefas e durante a release do projeto.

O TAR é definido após a execução dos testes.

Dados da publicação	
Título: [2]	Software quality analysis for agile development
Autor (es):	Yamada S., Kii R.
Ano da publicação:	2015
Referência completa:	Yamada S., Kii R.. (2015). " Software quality analysis for agile development." Department of Social Management Engineering, Graduate School of Engineering, Tottori University, Japan
Resumo da publicação	
O artigo propõe uma forma de avaliação quantitativa para identificar confiabilidade/qualidade de produtos de software durante o desenvolvimento de software com métodos ágeis. Os autores discutem sobre um modelo de confiabilidade de software baseado no modelo não-homogêneo e geométrico Poisson (NHPP- nonhomogeneous Poisson process). Cita que a confiabilidade cresce à medida que a relação entre o número de iterações e a o número acumulado de defeitos termina ao final de cada iteração. Através de modelo NHPP, uma série de formulas são exemplificadas para obter detecção de falhas, e número de falhas latentes esperadas. O autor exemplifica ainda a fórmula para detecção de falhas através de um conjunto de dados do projeto e mostra valores das medidas listadas no artigo como medidas básicas. Depois de realizada a comparação dos valores obtidos usando alguns modelos de confiabilidade existente chamados (SRGM's- Software Reliability Growth Model), utiliza também gráficos estatísticos mostrando como o nível de confiança evolui ao longo da release. A conclusão é que o número de detecção de falhas tende a crescer nos estágio iniciais e gradualmente estabilizam, melhorando assim a qualidade do produto.	

Número de casos de testes e tamanho de desenvolvimento para cada fase são as medidas que podem trazer maiores benefícios de acordo com o artigo, porém sem exemplos práticos listados.	
Medidas citadas no artigo (independente de ser de teste)	
• Total number of iterations – Numero total de iterações • Number of test-cases executed by the system testing in the last phase of each iteration. – Número de casos de testes executados na fase de testes de sistema na última fase de cada iteração • Implemented development size for each iteration – Tamanho da implementação de desenvolvimento de cada iteração • Development effort spent on implementing for each iteration.- Esforço de desenvolvimento de cada iteração • Number of performed reviews for each iterations.- Número de revisões realizadas em cada iteração • Total number of defects – Total de Defeitos encontrados nos testes • Expected total number of latent faults prior testing. – Número total de falhas latentes esperadas antes de iniciar os testes • Expected Cumulative number of faults detected in the given number of iterations – Número acumulado esperado de faltas detectadas até a n-ésima iteração	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
1. Os autores citam as medidas como sendo elementos importantes para análise dos resultados de cada iteração. Não substanciam essa opinião com nenhum dado ou referência, mas as utilizam como entradas para a criação da medida E(n). 2. Medida proposta pelos autores para calcular a confiabilidade de software desenvolvido utilizando métodos ágeis.	
Exemplo de utilização	
1. Os autores apresentam um exemplo de uso do indicador gerado, utilizando as medidas base. Através de gráficos estatísticos chegam à conclusão que o número de detecção de falhas tende a crescer nos estágios iniciais e gradualmente estabilizam, melhorando assim a qualidade do produto.	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Qualquer tipo de projeto de desenvolvimento com metodologia ágil. Número de casos de testes e tamanho de desenvolvimento para cada fase são as que podem trazer maiores benefícios de acordo com o artigo, porém sem exemplos práticos.	
Evidência de uso efetivo	
As medidas e fórmulas foram aplicadas em datasets extraídos de um projeto de desenvolvimento ágil.	
Medidas relacionadas a teste	
1- Number of test-cases executed by the system testing in the last phase of each iteration. – Número de casos de testes executados na fase de testes de sistema na última fase de cada iteração 2 - Total number of defects – Total de Defeitos encontrados nos testes	
Fórmulas para cálculo das medidas de teste	
$E(n) = \sum_{i=1}^n \lambda k^{i-1} = \frac{\lambda(1-k^n)}{(1-k)},$ $E(\infty) = \lim_{n \rightarrow \infty} E(n) = \frac{\lambda}{1-k}.$ i = average number of faults detected during the first interval [0,T] k= the decrease ratio of the number of faults detected in each iteration	

Dados da publicação	
Título: [3]	Using metrics in Agile and Lean software development - A systematic literature review of industrial studies
Autor (es):	Kupiainen E., Mäntylä M.V., Itkonen J.
Ano da publicação:	2015
Referência completa:	Kupiainen E., Mäntylä M.V., Itkonen J.. (2015). " Using metrics in Agile and Lean software development - A systematic literature review of industrial studies" Department of Computer Science and Engineering, Aalto University, Finland
Resumo da publicação	
O artigo descreve através de uma revisão sistemática da literatura a utilização de métricas em projetos de desenvolvimento de software com metodologia ágil-lean. Afirma que a literatura ágil propõe a utilização de métricas,	

mas que poucas são conhecidas e divulgadas, e que muitas medidas são customizadas para atender à demanda dos projetos.

O artigo analisa as motivações reportadas, benefícios das métricas aplicadas e os efeitos práticos da utilização de métricas em projetos ágeis. Uma lista de medidas é listada e catalogada por métodos ágil utilizado (Scrum, XP, Lean, Kanban). As razões para utilização de cada medida em cada fase de projeto também são apresentadas.

Medidas citadas no artigo (independente de ser de teste)

[S1] Business value delivered, customer satisfaction, defect count after testing, number of test cases, running tested features.

[S1] Valor de negócio entregue, satisfação do cliente, total de defeitos após testes, quantidade de casos de teste, Quantidade de funcionalidade que passam nos testes de aceitação.

[S2] Velocity, Work in progress.

[S2] Velocidade, Progresso do trabalho.

[S3] Critical defects sent by customer, open defects, test failure rate, test success rate, remaining task effort, team effectiveness.

[S3] Defeitos críticos enviados pelo cliente, defeitos abertos, taxa de teste falhados, Percentual de testes executados com sucesso, esforço para as tarefas restantes, efetividade da equipe.

[S4] Technical debt board, build status, technical debt in effort

[S4] Quadro de debito técnico, status da construção, esforço do débito técnico.

[S5] Burndown, check-ins per day, number of automated passing test steps, faults per iteration

[S5] Burndown, check-ins por dia, quantidade de passos automatizados, falhas por iteração.

[S6] Velocity, story estimates.

[S6] Velocidade, Estimativa de histórias.

[S7] Burndown, story points, # of open defects, # of defects found in system test, defects deferred, Net Promoter Score.

[S7] Burndown, pontos de histórias, Quantidade de defeitos aberto, Quantidade de defeitos descobertos no teste de sistema, defeitos adiados, Net Promoter Score.

[S8] Story points, task effort, velocity, operations' velocity.

[S8] Pontos de história, esforço das tarefas, velocidade, operações de velocidade .

[S9] Effort estimate.

[S9] Esforço estimado.

[S10] # of defects/velocity.

[S10] Quantidade de defeitos /velocidade.

[S11] Revenue per customer.

[S11] Receita per cliente.

[S12] Task's expected end date, effort estimate, completed web pages, task done.

[S12] Data final esperada para a tarefas, Esforço estimado, Paginas web finalizadas, Tarefas finalizadas.

[S13] Fix time of failed build, story flow percentage, percentage of stories prepared for sprint, velocity of elaborating features, velocity of implementing features.

[S13] Tempo de correção para construções falhadas, percentual de fluxo de histórias, percentual de histórias preparadas para sprint, velocidade na elaboração das funcionalidade, velocidade na implementação de funcionalidades.

[S14] Build status, test coverage, test growth ratio, violations of static code analysis, # of unit tests.

[S14] status das construções, cobertura de teste, taxa de crescimento de teste, violações de análise estática de código, quantidade de testes unitários. [S15] Effort estimate. [S15] Esforço estimado. [S16] Sprint burndown, release burndown, cost performance index, schedule performance index, planned velocity. [S16] Sprint burndown, release burndown, índice de desempenho de custo, índice de desempenho do prazo, velocidade planejada. [S17] Common tempo time, number of bounce backs, cycle time, work in progress, customer satisfaction (Kano analysis), effort estimate kits. [S17] Ritmo de tempo comum, número de retornos, Tempo do ciclo, progresso do trabalho, satisfação do cliente (Kano analysis), estimativa de esforço. [S18] Lead time, processing time, queue time. [S18] Tempo de espera, tempo de processamento, tempo na fila. [S19] Change requests per requirement, fault slips, implemented vs wasted requirements, maintenance effort, lead time. [S19] Solicitação de mudança por requisito, falhas, requisitos implementados x desperdício, esforço de manutenção, Tempo de espera. [S20] Number of requests from customers, inventory of requirements over time [S20] Número de solicitação dos clientes, inventário de requisitos atrasados. [S21] Rate of requirements per phase, variance in handovers, requirement's cost types. [S21] Taxa de requisitos por fase, variação nas transferências, tipo de custo por requisitos. [S22] # of requirements per phase, lead time. [S22] Quantidade de requisitos por fase, tempo de espera. [S23] Average velocity/ work in progress, cycle time, pseudo velocity. [S23] Velocidade média / progresso do trabalho, Tempo do ciclo , pseudo velocidade. [S24] Lead time, work in progress. [S24] Tempo de espera, progresso do trabalho [S25] Defect trend indicator, # of defects in backlog, predicted # of defects [S25] Indicador de tendência de defeitos, Quantidade de defeitos no backlog, Quantidade de defeitos previstos. [S26] Throughput, queue. [S26] Throughput, fila. [S27] Burndown, check-ins per day, number of automated passing test steps, number of new and open defects [S27] Burndown, check-ins por dia, número de passos automatizados, Quantidade de defeitos novos e abertos. [S28] Burndown, number of automated passing test steps, check-ins per day . [S28] Burndown, número de passos automatizados, check-ins por dia. [S29] Story estimate, story complete percentage [S29] Estimativa de história, percentual de histórias finalizadas [S30] Progress as working code. [S30] Progresso do código em trabalho.
Contexto em que a medida é (ou pode ser) utilizada em projetos

O contexto é baseado na revisão sistemática, durante uma pesquisa sobre quais medidas são utilizadas, e quais benefícios e fases dos projetos ágeis são mais usadas.
Exemplo de utilização
Os autores apenas listam as medidas de acordo com as questões de pesquisa.
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis
Qualquer tipo de projeto de desenvolvimento com metodologia ágil as medidas listadas podem ser aplicadas na minha visão.
Evidência de uso efetivo
Alguns gráficos são mostrados com números de utilização das medidas em projetos, como: fontes e razões, como resolução de problemas, motivação de pessoas, planejamento de projetos, etc.
Medidas relacionadas a teste
1-Open defects – Defeitos abertos 2-Test failure rate - Taxa de testes com falha 3-Faults per iteration – Falhas por iteração 4-Number of automated passing test steps - Número de passos de testes automatizados executados com sucesso 5-Critical defects sent by customer - Quantidade de Defeitos críticos enviados pelo cliente 6-Defect trend indicator - Indicador de tendência de defeitos 7-Predicted number of defects - Indicador de Previsão de defeitos 8-Test growth ratio - Taxa de aumento de casos de testes entre versões/releases 9-Defect count after testing – Número Total de Defeitos após testes 10-Number of test cases – Quantidade de casos de testes planejados 11-Test success rate - Percentual de testes executados com sucesso 12- Defects deferred - Quantidade de defeitos analisados e não resolvidos 13-Test coverage - Cobertura de Testes 14 - Running Tested Features - Quantidade de funcionalidade que passam nos testes de aceitação 15 - # of defects/velocity - Quantidade de defeitos /velocidade 16 - # of open defect. – Quantidade de defeitos abertos 17 - # of defects found in system test – Quantidade de defeitos descobertos no teste sistema 18 - # of unit tests - Quantidade de testes unitários 19 - # of defects in backlog – Quantidade de defeitos no backlog 20 - number of new and open defects – Quantidade de defeitos novos e abertos
Fórmulas para cálculo das medidas de teste
Não há formulas dos indicadores descritos.

Dados da publicação	
Título: [4]	Release readiness indicator for mature agile and lean software development projects
Autor (es):	Staron M., Meding W., Palm K.
Ano da publicação:	2015
Referência completa:	Staron M., Meding W., Palm K. (2012). " Release readiness indicator for mature agile and lean software development projects." Software Centre, Computer Science and Engineering Chalmers / University of Gothenburg, Gothenburg, Sweden.
Resumo da publicação	
O artigo apresenta uma pesquisa-ação que resultou na definição de um indicador que pudesse prever o tempo em semanas para que uma release fosse liberada para entrega em ambiente de produção. O indicador RR (Release Readiness) se mostrou muito importante. Foi utilizado como indicador de desempenho e como controlador de desenvolvimento de produto e melhoria para desempenho organizacional. Este indicador pode ser utilizado por diferentes membros de projeto como: Gerente de projetos, product owner e para os responsáveis de releases que devem sinalizar quando a release está liberada para o ambiente do cliente.	
Medidas citadas no artigo (independente de ser de teste)	
• RTF - Running Tested Features - Quantidade de funcionalidade que passam nos testes de aceitação • RR Release Readiness –Prontidão para release (Número de semanas para Release) • Number of executed test cases – Número de casos de testes executados • Number of planned test cases (per priority and type) – Número de casos de testes planejados • Number of failed test cases – Número de casos de testes falhados	

• Number of passed test cases – Número de casos de testes executados com sucesso • Number of defects discovered per test case – Número de defeitos descobertos po casos de teste • Number of defects discovered – Número de defeitos descobertos • Number of defects to be removed (defect backlog) – Quantidade de defeitos a serem removidos do backlog • Number of defects removed – Número de defeitos removidos • Number of defects verified – Número de defeitos verificados • Number of features integrated into the main code branch (work package integration pace) – Número de funcionalidades integrada no repositório de código. • Number of empty integration spots (LSV usage) – Número de integrações em branco	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
O indicador RR foi desenvolvido e aplicado em um projeto de desenvolvimento de software ágil, em uma empresa de telecomunicações.	
Exemplo de utilização	
O artigo apresentou um gráfico baseado no cálculo do indicador RR durante 11 semanas, porém não mostrou os números coletados nas medidas usadas de cálculo. Com isso, fica inconclusivo se outros números serão aceitos pela formula do indicador	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Foi aplicado em um projeto ágil, utilizando o formato Lean. Porém baseado no resultado da pesquisa, percebe-se que este indicador pode ser aplicado a outros projetos com métodos ágil e não ágil, uma vez que as medidas necessárias são passíveis de serem coletadas nos projetos.	
Evidência de uso efetivo	
Durante o desenvolvimento, o indicador RR foi aplicado em dados históricos de projetos passados. Depois foi implementado em um projeto com uma média de 100 desenvolvedores de software.	
Medidas relacionadas a teste	
1-RR Release Readiness –Prontidão para release (Número de semanas para Release) 2-Number of executed test cases – Quantidade de casos de testes executados 3-Number of planned test cases (per priority and type) – Quantidade de casos de testes planejados 4-Number of failed test cases – Quantidade de casos de testes falhados 5-Number of passed test cases – Quantidade de casos de testes executados com sucesso 6-Number of defects discovered per test case – Quantidade de defeitos por caso de testes 7-Number of defects discovered – Quantidade de defeitos encontrados nos testes 8-Number of defects to be removed (defect backlog) – Quantidade de defeitos a serem removidos (backlog da release) 9-Number of defects removed – Quantidade de defeitos removidos 10-Number of defects verified – Quantidade de defeitos analisados e não resolvidos 11- RTF - Running Tested Features - Quantidade de funcionalidade que passam nos testes de aceitação	
Fórmula apresentada pelos autores	
$RR = \left(\frac{\# defects}{defect_removal_rate - (test_execution_rate - test_pass_rate)} \right)$ #defects = Number of defects discovered (Quantidade de Defeitos encontrados nos testes) defect_removal_rate= media da Quantidade de defeitos resolvidos durante 4 semanas de testes test_execution_rate = média da Quantidade de casos de testes executados durante 4 semanas test_pass_rate = média da Quantidade de casos de testes passados durante 4 semanas	

Dados da publicação	
Título: [5]	Dashboards for Continuous Monitoring of Quality for Software Product under Development
Autor (es):	Staron M., Meding W., Hansson J., Höglund C., Niesel K., Bergmann V.
Ano da publicação:	2014
Referência completa:	Staron M., Meding W., Hansson J., Höglund C., Niesel K., Bergmann V.. (2014). " Dashboards for Continuous Monitoring of Quality for Software Product under Development " - University of Gothenburg – Sweden.
Resumo da publicação	

O artigo cita a importância de monitoração de projetos de desenvolvimento de software, cita a qualidade de software como um fator que diferencia projetos de excelência de projetos comuns. Cita também que o monitoramento de qualidade precisa ser feito gerenciando o número de defeitos e agilidade da organização que faz o desenvolvimento de software, monitorando quando e como a qualidade é obtida. Estes novos desafios requerem novas formas de pensar e a construção de medição, requerem um grande número de medidas exibidas em um grande quadro de informações (dashboard).

O framework utilizado como apoio para medição de sistemas é a norma ISO/IEC 15939 e GQM – Global-Question-Metric.

Medidas citadas no artigo (independente de ser de teste)

- Release Readiness –Prontidão para release (Número de semanas para Release)
- Number of model revisions – Número de modelos revisados
- Complexity of defects – Complexidade de defeitos
- Number of defects – Número de Defeitos
- Number of requirements in requirements' backlog for the requirements phase – Quantidade de requisitos no backlog por fase de requisitos
- Number of model/code check-ins for the development phase – Número de check-ins de modelos/código na fase de desenvolvimento
- Number of violations of architectural rules in models/code – Quantidade de violações das regras de arquitetura no modelo/código
- Percentage of passed test cases for the test phase -Porcentagem de casos de testes executados com sucesso da fase de testes
- Test Execution Progress – Número de casos de testes executados

Contexto em que a medida é (ou pode ser) utilizada em projetos

1 - Os autores citam as medidas bases, medidas derivadas e indicadores que compõem um quadro de informações para monitoramento de projetos com metodologia ágil-lean.

Medidas bases – Ex. Testes passados na semana

Medidas Derivadas: Taxa de testes passadas nas últimas 4 semanas.

Indicadores – Release Readiness

2 - Definição de quais medidas e indicadores devem ser usados para monitoramento do progresso.

Exemplo de utilização

1. Os autores apresentam um exemplo de uso dos indicadores como parte de um dashboard de projeto na avaliação do processo de desenvolvimento e teste.

Indícios de que as medidas podem ser utilizadas em outros projetos ágeis

Qualquer tipo de projeto de desenvolvimento com metodologia ágil. No monitoramento de projetos de desenvolvimento.

Evidência de uso efetivo

O artigo cita quais indicadores foram utilizados em dashboard em 3 diferentes empresas e também mostra recomendações para a construção de dashboards como escolher os indicadores que trarão informações relevantes aos stakeholders, limitar o número de indicadores evitando informação desnecessária, envolver equipe de validação e verificação. Também lista recomendações para escolha dos indicadores e medidas.

Porém não há exemplos de um dashboard com as medidas e indicadores.

Medidas relacionadas a teste

- 1-Release Readiness Indicator-Número de semanas para Release
- 2- Complexity of defects – Complexidade de defeitos
- 3- Number of defects – Número de Defeitos
- 4-Percentage of passed test cases for the test phase -Porcentagem de casos de testes executados com sucesso da fase de testes
- 5- Test Execution Progress – Número de casos de testes executados

Fórmulas para cálculo das medidas de teste

$$RR = \left(\frac{\# defects}{defect_removal_rate - (test_execution_rate - test_pass_rate)} \right)$$

#defects = Number of defects discovered (Quantidade de Defeitos encontrados nos testes)

defect._removal_rate= media da Quantidade de defeitos resolvidos durante 4 semanas de testes

test_execution_rate = média da Quantidade de casos de testes executados durante 4 semanas

test_pass_rate = média da Quantidade de casos de testes passados durante 4 semanas

Dados da publicação	
Título: [6]	A method to build Bayesian networks based on artifacts and metrics to assess agile projects
Autor (es):	Willamy R., Nunesy J., Perkusichz M., Freirex A., Saraiva R., Almeida H., Perkusich A.
Ano da publicação:	2016
Referência completa:	Willamy R., Nunesy J., Perkusichz M., Freirex A., Saraiva R., Almeida H., Perkusich A. (2016). " A method to build Bayesian networks based on artifacts and metrics to assess agile projects ". Federal University of Campina Grande, Campina Grande, Brazil
Resumo da publicação	
Este artigo propõe um método para construção de artefatos e modelos base de medição para avaliar projetos. Nestes métodos foi aplicado o GQM "Goal Question Metric", para definição das medidas. Após as definições das medidas foi criado uma rede Bayesiana para definição do modelo proposto.	
Medidas citadas no artigo (independente de ser de teste)	
• Number of Defects– Quantidade de defeitos • Project Velocity– Velocidade de execução do projeto • Backlog itens Dependency – Dependência dos itens do backlog • Risks - Riscos • Estimation Conformity – Estimativa de conformidade • Business Value – Valor de negócio • Number of open bugs - Quantidade de bugs abertos. • Tested Feature - Funcionalidades testadas • Unit tests coverage - Cobertura de testes unitários • Item detail level – Nível de detalhamento do item • Dynamics of the estimation – Dinamica da estimativa • Static Analysis Warning – Aviso da análise estática • Integration Test Coverage – Cobertura de testes de integração • Functional Test Coverage – Cobertura de testes funcionais	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas foram usadas para definição e aplicação no método proposto.	
Exemplo de utilização	
As medidas foram apresentadas no modelo de rede Bayesian proposto no artigo.	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Foi aplicado em um projeto ágil, utilizando XP. As medidas são passíveis de serem aplicadas em outros projetos de contexto ágil.	
Evidência de uso efetivo	
Não, apenas simulação conforme citado no artigo	
Medidas relacionadas a teste	
1- Number of Defects– Quantidade de defeitos 2- Number of open bugs - Quantidade de bugs abertos 3- Tested Feature - Funcionalidades testadas 4- Unit tests coverage - Cobertura de testes unitários 5- Integration Test Coverage – Cobertura de testes de integração 6- Functional Test Coverage – Cobertura de testes funcionais	
Fórmulas para cálculo das medidas de teste	
Não há	

Dados da publicação	
Título: [7]	Challenges in adapting agile testing in a legacy product
Autor (es):	Gupta R.K., Manikreddy P., Gv A.
Ano da publicação:	2016
Referência completa:	Gupta R.K., Manikreddy P., Gv A. (2016). " Challenges in adapting agile testing in a legacy product ". Siemens Techonology and Services, Bangalore, India

Resumo da publicação	
Este artigo descreve, em forma de estudo de caso, a adoção de métodos ágeis em um projeto de desenvolvimento de software que utilizava metodologia “waterfall” durante longo tempo. É descrito as dificuldades encontradas pela equipe de testes, como falta de skill e prática em modelo ágil SCRUM. Ao final do artigo é feito uma comparação utilizando algumas medidas sobre o antes e depois da implementação do modelo ágil dentro da equipe. Uma das métricas revelou que testes automatizados nunca eram utilizados anteriormente.	
Medidas citadas no artigo (independente de ser de teste)	
• Open Defects–Defeitos não resolvidos • Test Debt– Débito de teste • Test Coverage –Cobertura de testes • Defect Detection Percentage – Percentual de detecção de defeitos	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas foram usadas para avaliar a aplicação do modelo SCRUM na equipe de testes e evolução do projeto.	
Exemplo de utilização	
Foram coletados dados antes e depois da implementação do SCRUM	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
Foi aplicado em um projeto ágil, utilizando SCRUM. As medidas são passíveis de serem aplicadas em outros projetos de contexto ágil.	
Evidência de uso efetivo	
Sim, pela comparação de dados e medidas do projeto.	
Medidas relacionadas a teste	
1- Open Defects–Defeitos não resolvidos 2 - Test Coverage –Cobertura de testes 3- Defect Detection Percentage – Percentual de detecção de defeitos 4 – Test Debt– Débito de teste	
Fórmulas para cálculo das medidas de teste	
Não há	

Dados da publicação	
Título: [8]	Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance
Autor (es):	Malhotra R., Chug A.
Ano da publicação:	2016
Referência completa:	Malhotra R., Chug A. (2016). " Comparative analysis of agile methods and iterative enhancement model in assessment of software maintenance ". University School Of Information and Communication Technology, New Dheli, India
Resumo da publicação	
Este artigo realiza uma comparação entre metodologia ágil, através do modelo SCRUM com o Modelo de Melhoria Continua -IEM (Iterative Enhancement Model). Através de desenvolvimento de um software duas equipes utilizaram os mesmos requisitos, cada uma usando um modelo (SCRUM x IEM). Ao final, 4 medidas foram coletadas e as comparações realizadas.	
Medidas citadas no artigo (independente de ser de teste)	
• Number of Defects– Total de Defeitos encontrados nos testes • Number of Change Requests (CR) Received – Quantidade de Mudanças de requisitos recebidos • Time of Error Detected - Tempo (em Dias) para detecção dos defeitos • Features Rolled Out With Respect Time – Quantidade de funcionalidades liberados para cliente ao final de cada iteração	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
As medidas citadas foram usadas para avaliação dos modelos utilizados.	
Exemplo de utilização	
Coletas durante o projeto e analisados ao final para comparação	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	

Foi aplicado em um projeto ágil, utilizando SCRUM. As medidas são passíveis de serem aplicadas em outros projetos de contexto ágil.
Evidência de uso efetivo
Sim, pois as mesmas foram coletas durante o projeto.
Medidas relacionadas a teste
1- Number of Defects– Total de Defeitos encontrados nos testes 2- Time of Error Detected - Tempo (em dias) para detecção dos defeitos
Fórmulas para cálculo das medidas de teste
Não há

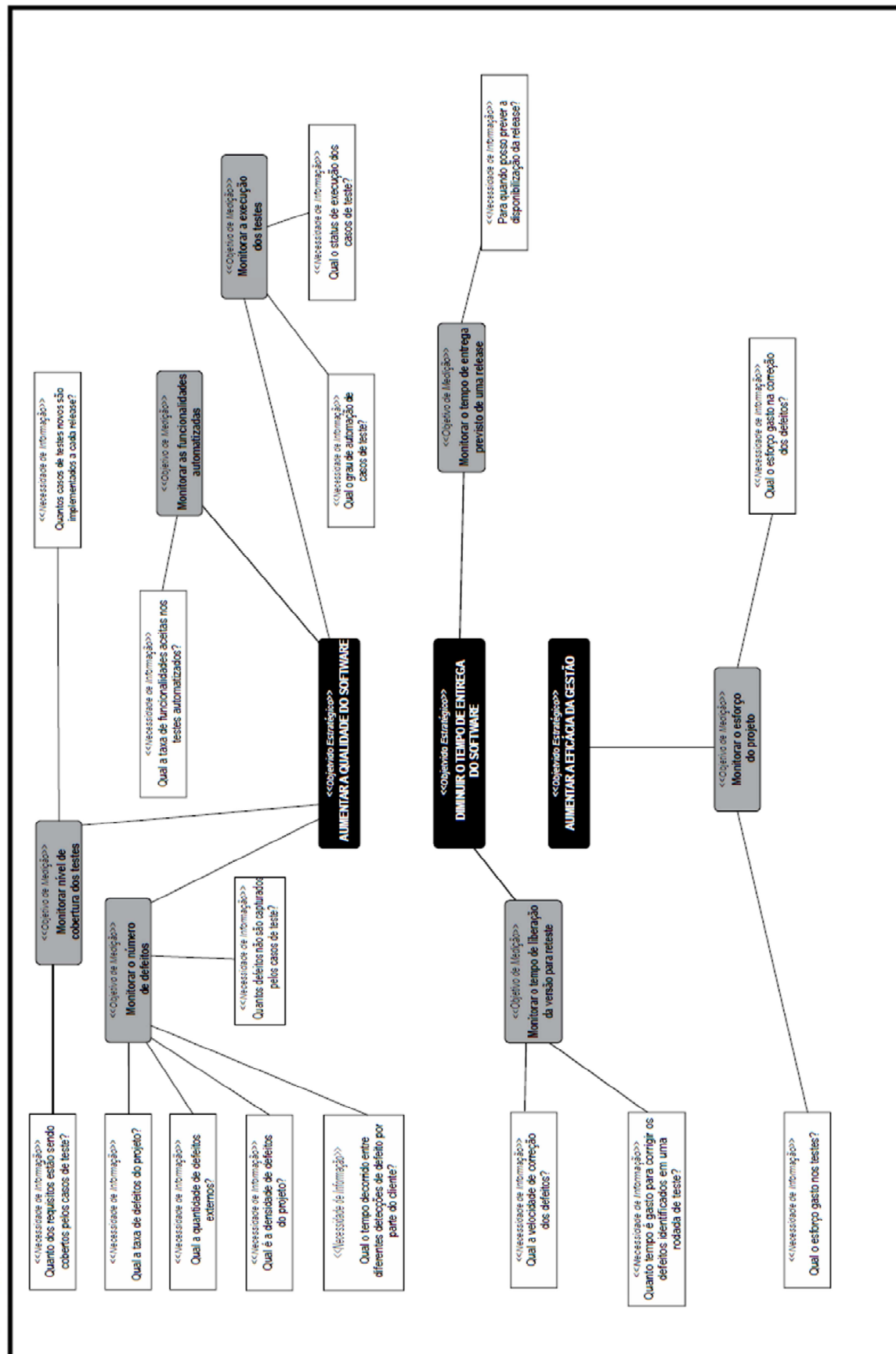
Dados da publicação	
Título: [9]	Software Quality Control via Exit Criteria Methodology: An Industrial Experience Report
Autor (es):	X. Zhao; X. Xuan; A. Wang; D. Liu; L. Zheng
Ano da publicação:	2014 - IEEE
Referência completa:	X. Zhao; X. Xuan; A. Wang; D. Liu; L. Zheng. (2014). " Software Quality Control via Exit Criteria Methodology: An Industrial Experience Report ". Shangai, China.
Resumo da publicação	
O artigo descreve a experiência prática e organizacional da utilização da metodologia critério de saída usada para controle de qualidade de software no desenvolvimento de uma grande aplicação web. O projeto utilizou metodologia SCRUM e o plano com critérios de saída foi utilizado para avaliação da qualidade da entrega do sistema em produção.	
Medidas citadas no artigo (independente de ser de teste)	
Test Coverage - Cobertura de testes	
Contexto em que a medida é (ou pode ser) utilizada em projetos	
Em um projeto que utilizou metodologia SCRUM na implementação de um modelo de validação da qualidade do desenvolvimento de software	
Exemplo de utilização	
Não foi descrito	
Indícios de que as medidas podem ser utilizadas em outros projetos ágeis	
As medidas identificadas são aplicadas em outros projetos que decidam avaliar informação semelhante	
Evidência de uso efetivo	
A medida citada foi coletada como parte do plano desenvolvido para avaliação da qualidade do software	
Medidas relacionadas a teste	
Test Coverage - Cobertura de testes	
Fórmulas para cálculo das medidas de teste	
Não há.	

Apêndice II - Fichas do conjunto de Indicadores seguindo modelo ASM.br

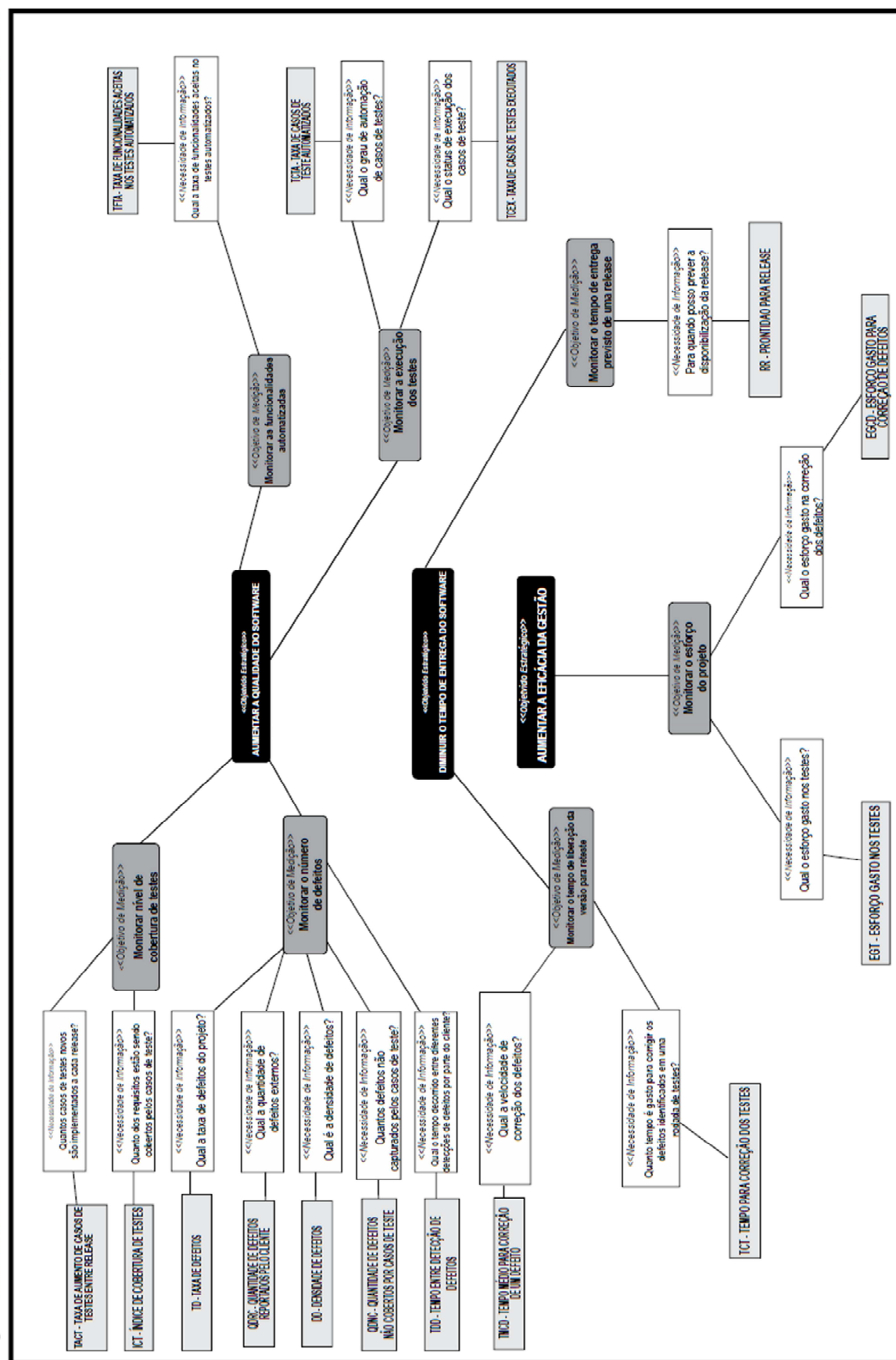
Repositório de Medição e informações da organização

 ASM[.br] Assistance for Software Measurement based on relationships UMA ABORDAGEM BASEADA EM RELACIONAMENTOS	NOME DA EMPRESA ÁREA OU SETOR DA EMPRESA Respositório de Medição: (LOCALIZAÇÃO DO REPOSITÓRIO DE MEDIÇÃO)
---	--

MAPA DE OBJETIVOS

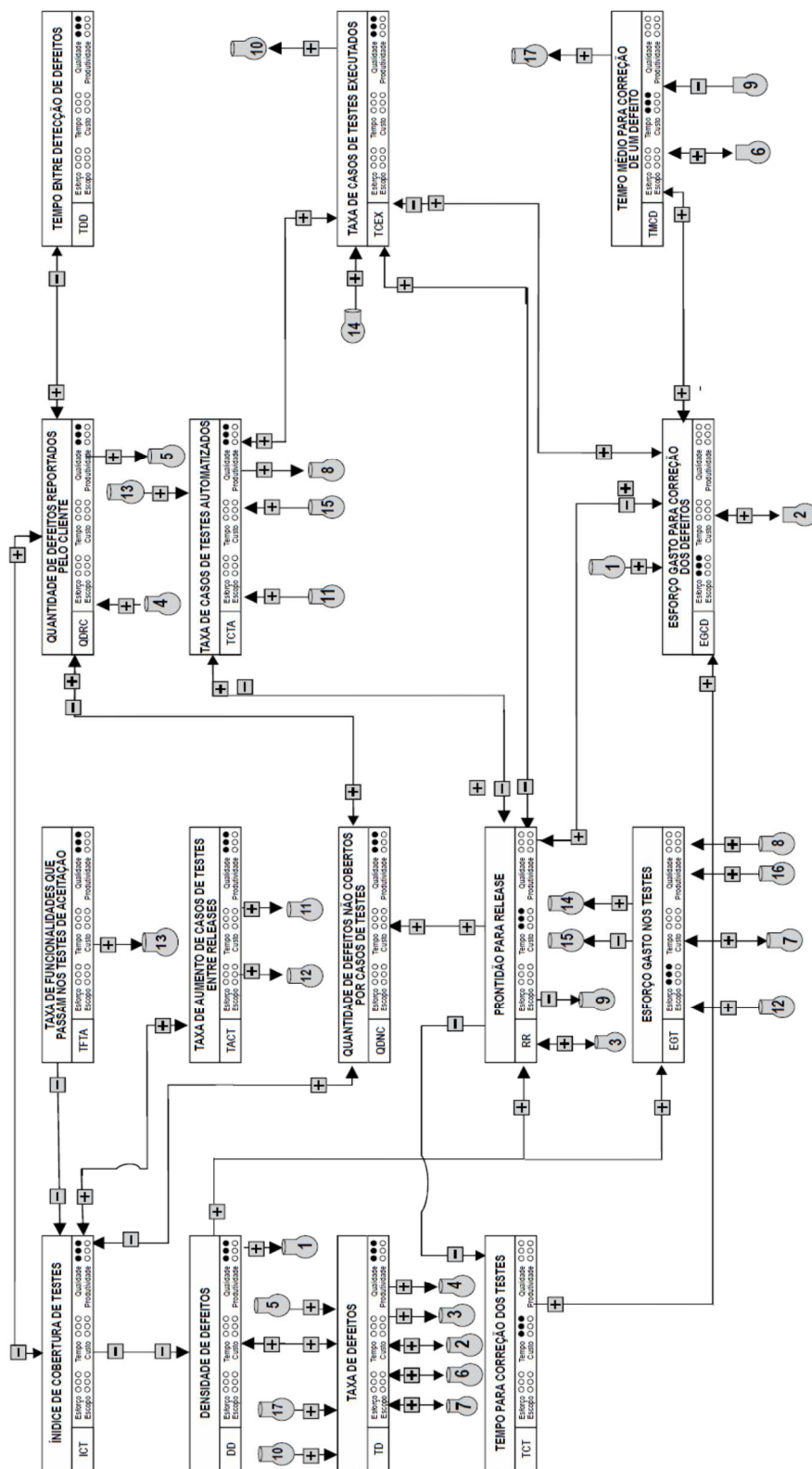


MAPA DE OBJETIVOS E INDICADORES

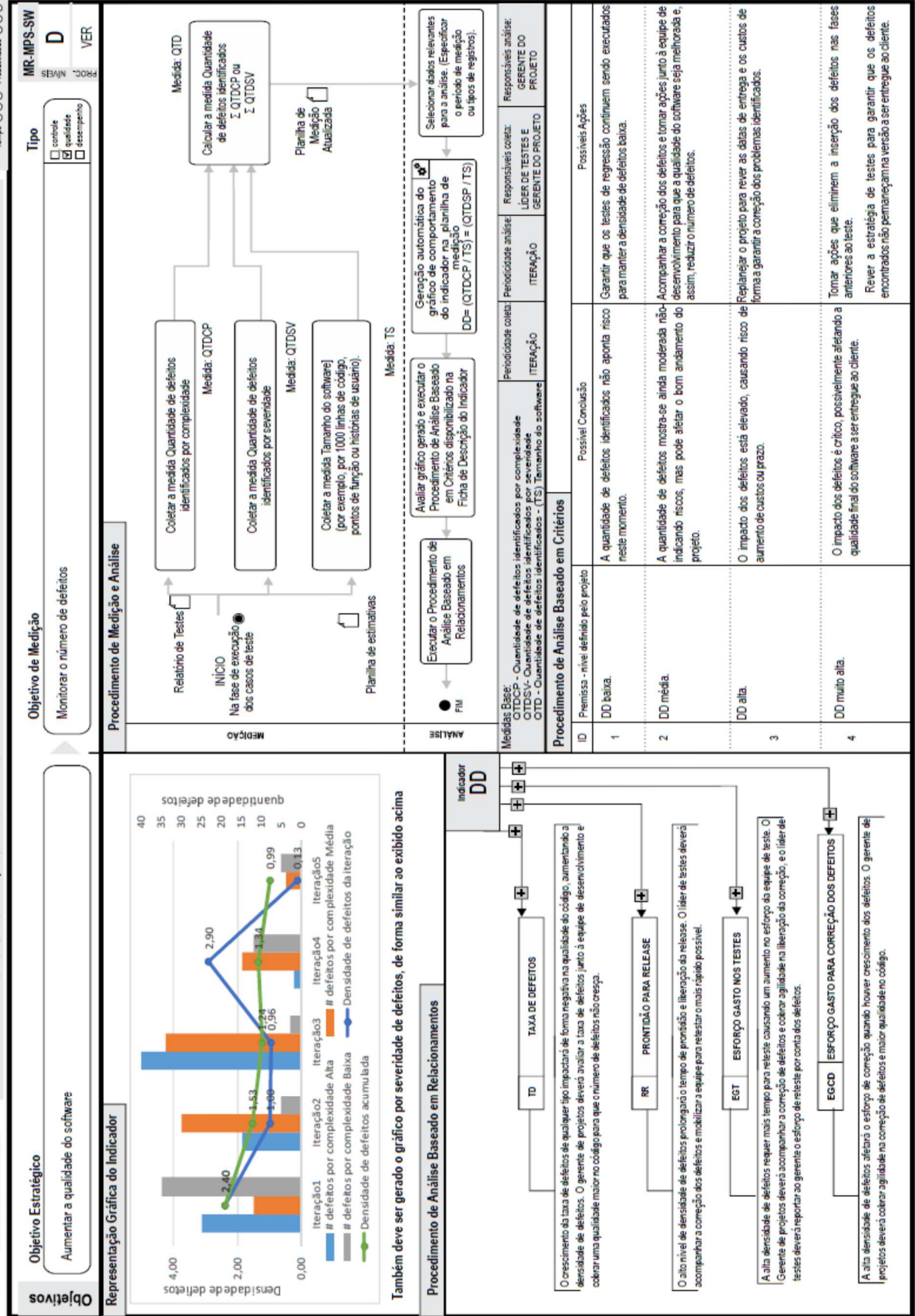


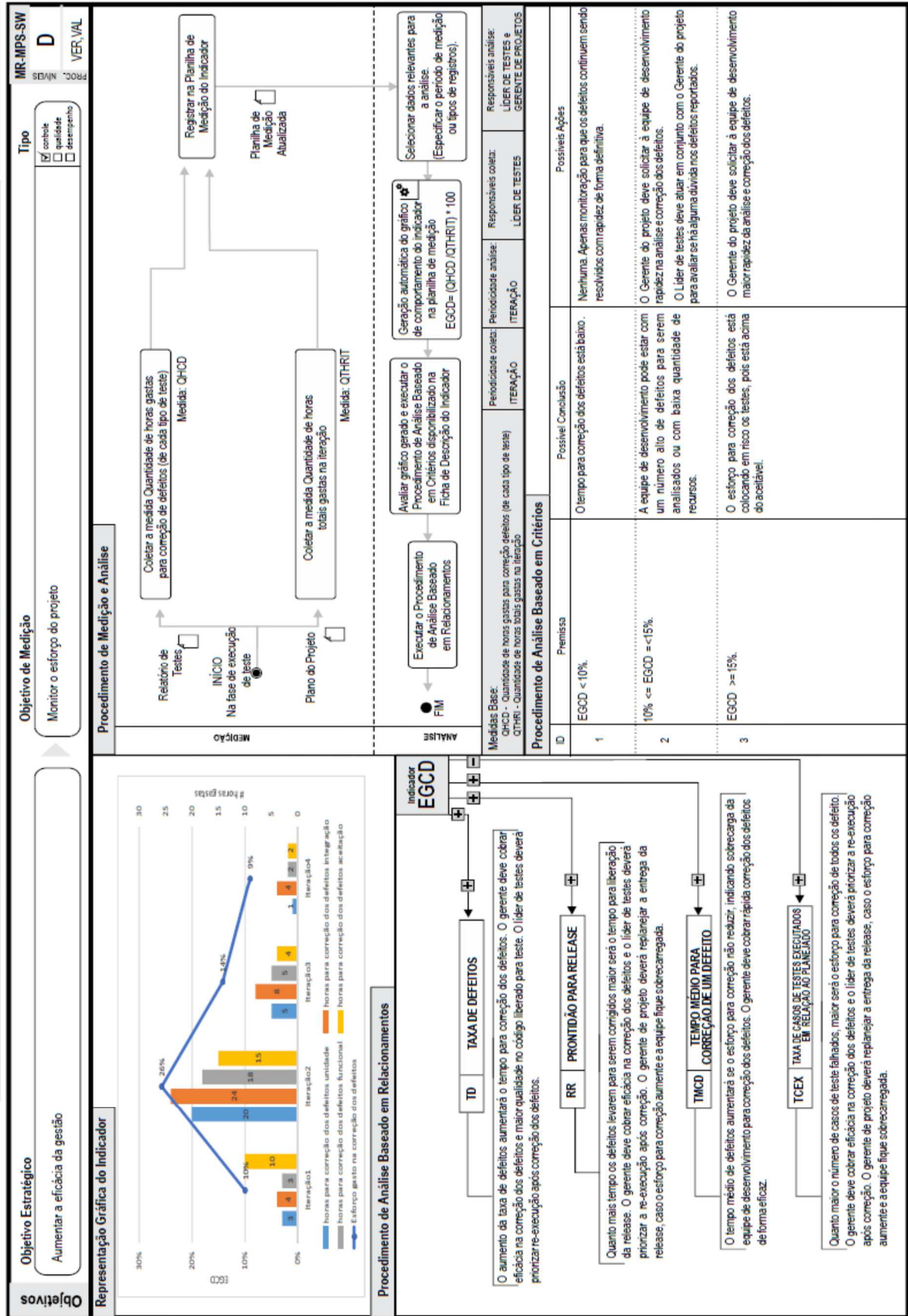
MAPA DE MR-MPS-SW				
INDICADORES		MR-MPW-SW		
		D		
SIGLA	NOME	VAL	VER	PI
ICT	Índice de Cobertura de Teste	X	X	
TD	Taxa de Defeitos		X	
QDRC	Quantidade de defeitos reportados pelo cliente	X		
DD	Densidade de Defeitos		X	
TFTA	Taxa de funcionalidades que passam nos testes de aceitação	X		
TDD	Tempo entre detecção de defeitos	X		
TACT	Taxa de aumento de casos de testes entre releases		X	
TCTA	Taxa de casos de testes automatizados		X	
RR	Prontidão para release			X
TMCD	Tempo médio para correção de um defeito		X	
TCT	Tempo para correção dos testes		X	
TCEX	Taxa de casos de testes executados em relação ao planejado		X	
EGT	Esforço gasto nos testes	X	X	
QDNC	Quantidade de defeitos não cobertos por casos de teste		X	
EGCD	Esforço gasto para correção de defeitos	X	X	

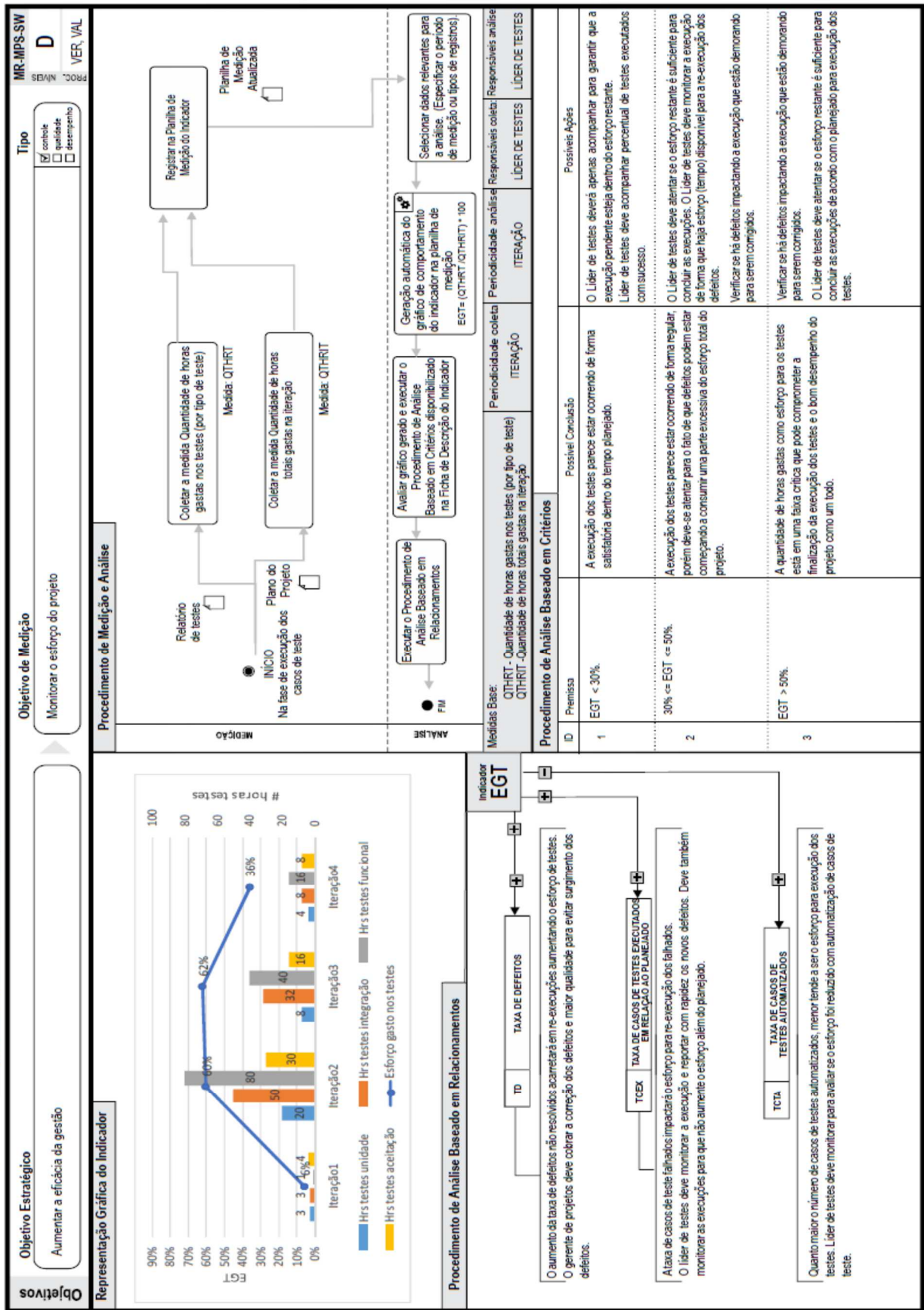
MAPA DE RELACIONAMENTO DE INDICADORES

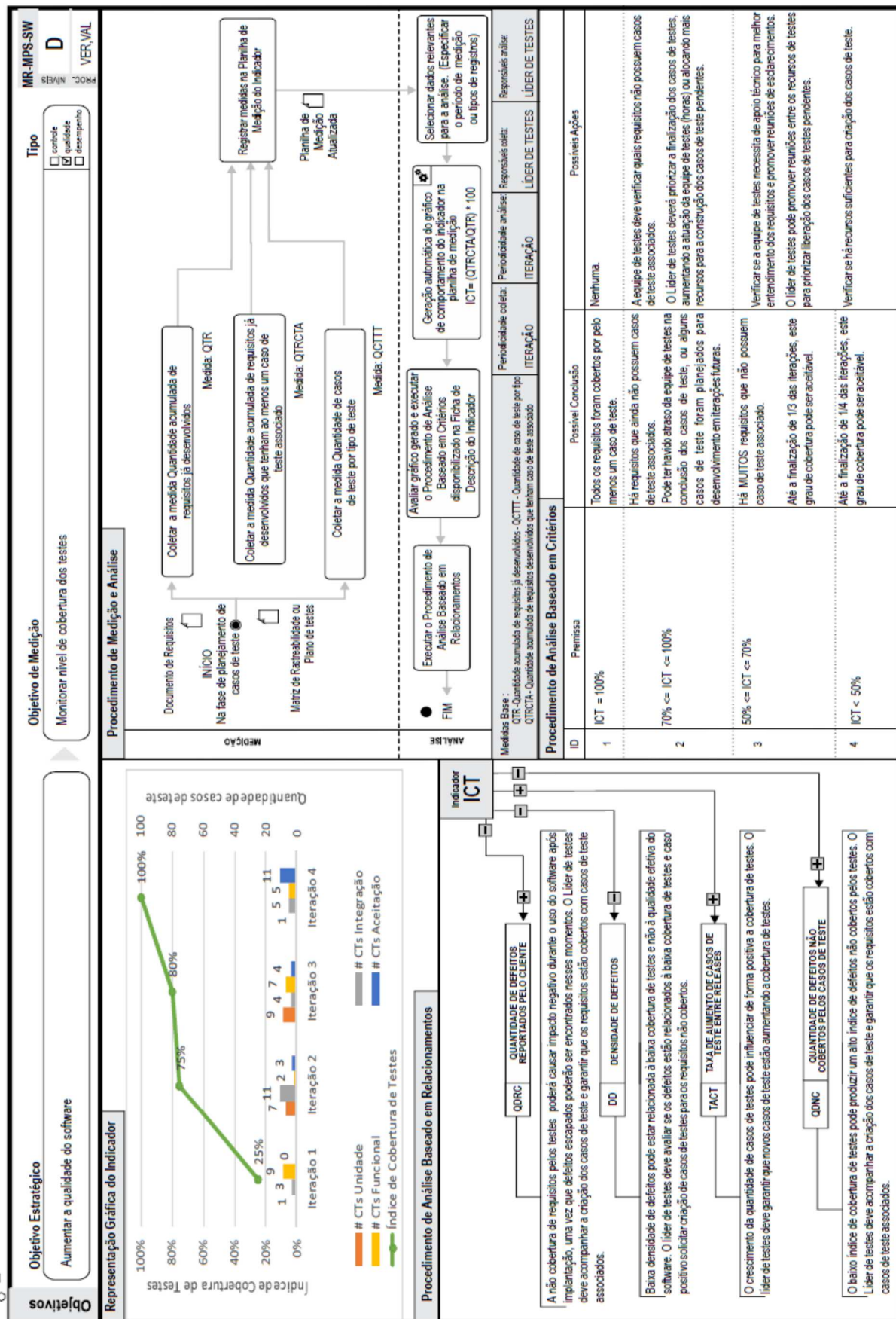


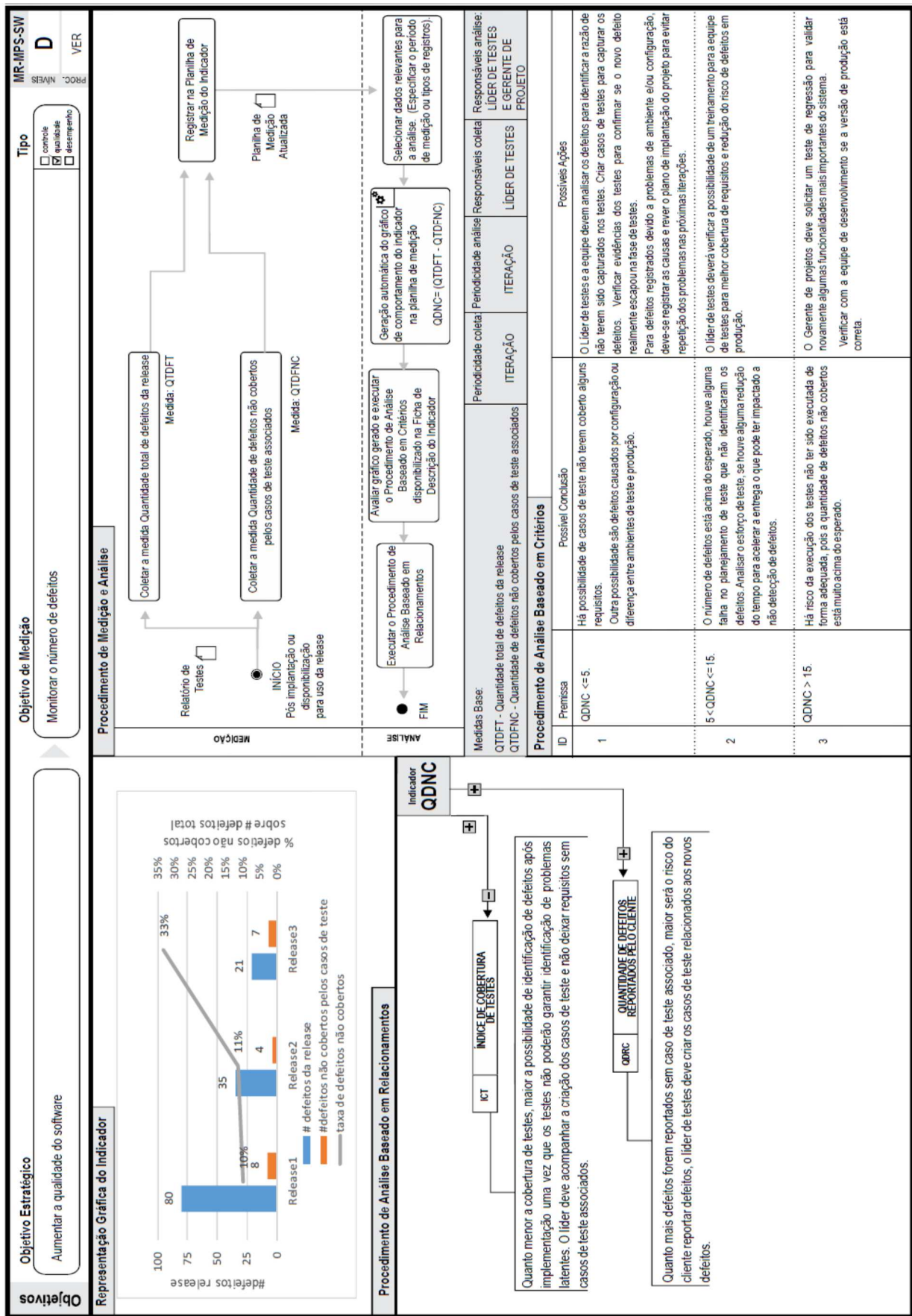
Fichas de registro do Indicadores

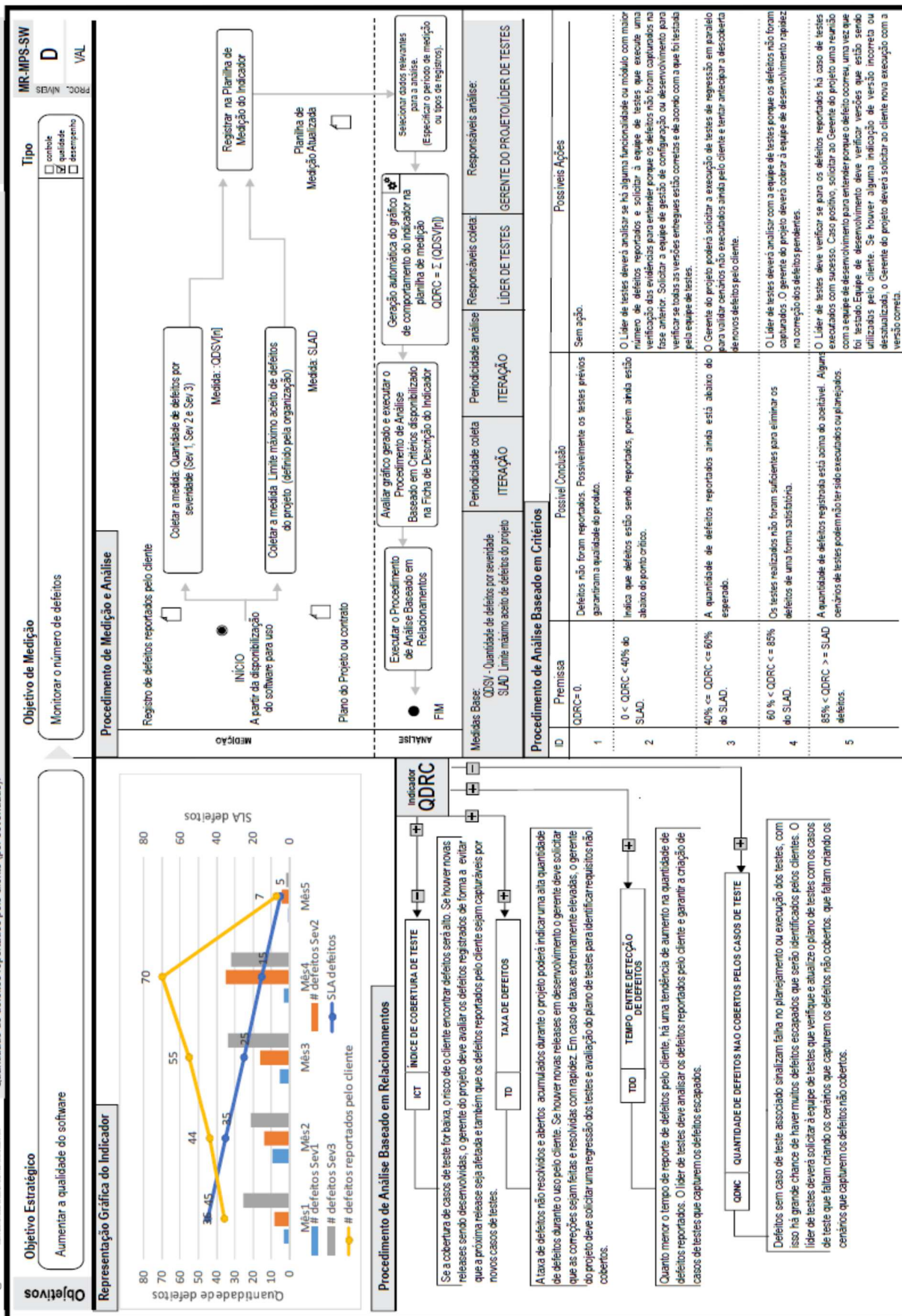


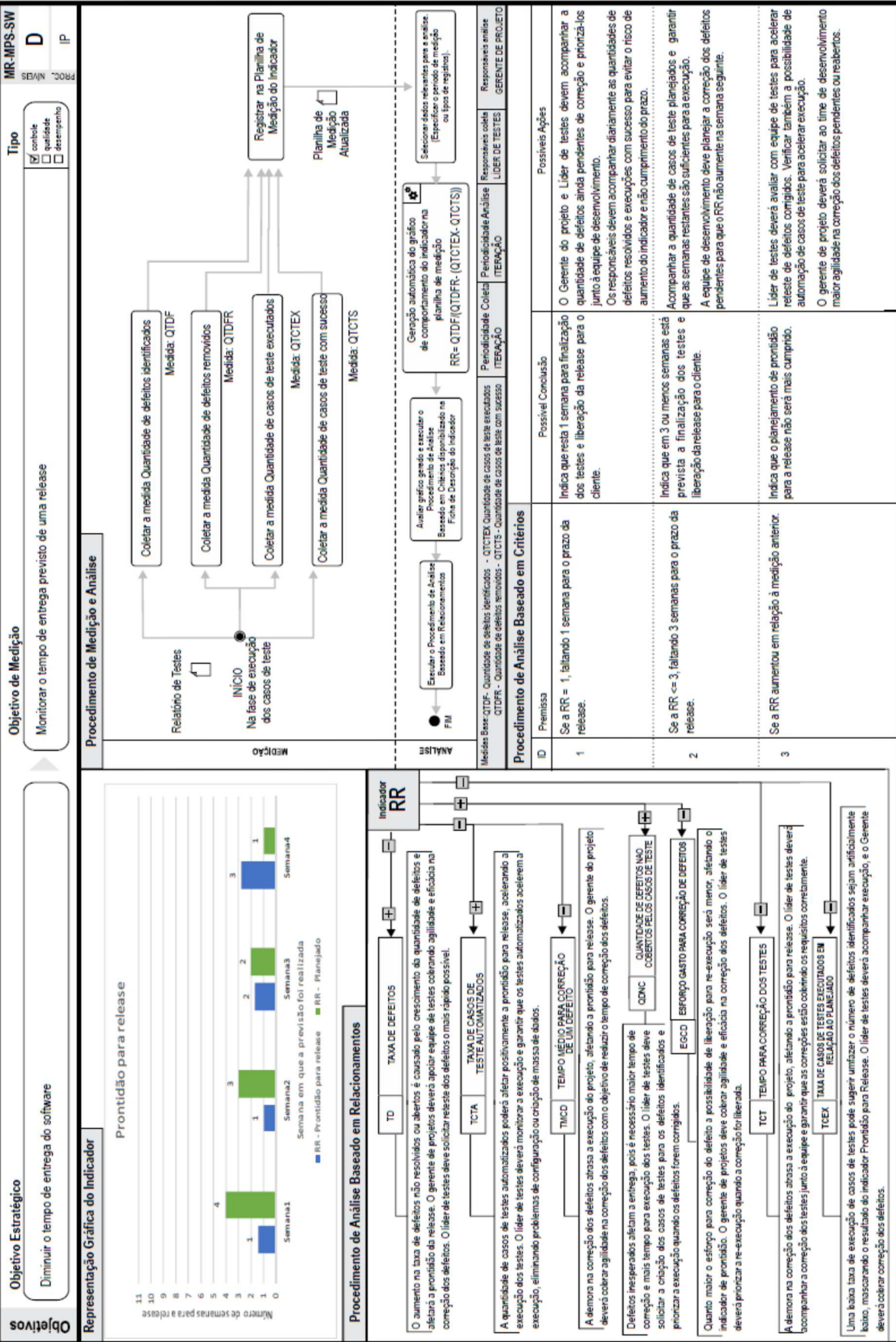


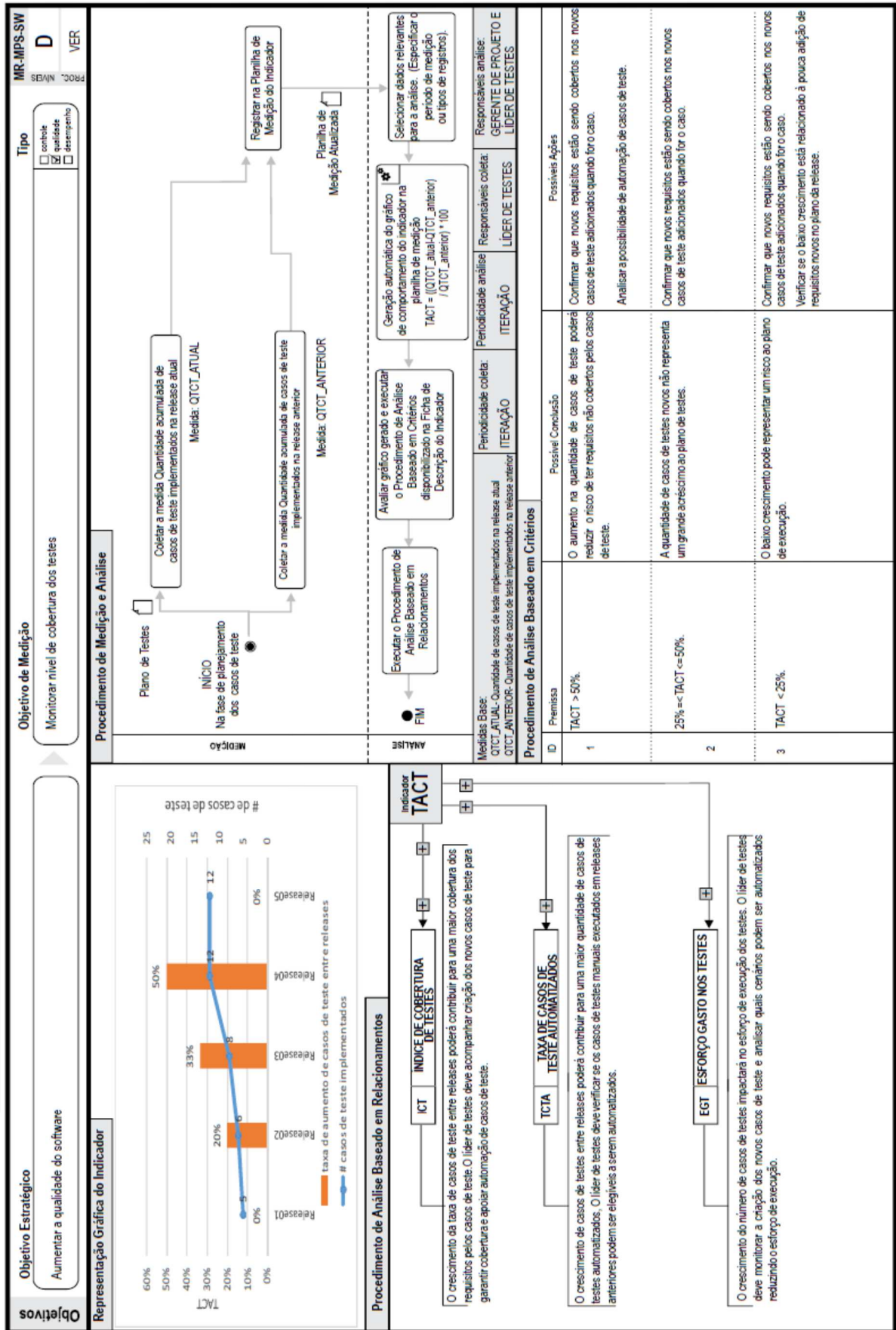












Qual o status de execução dos casos de teste?

TCEX - Taxa de casos de testes executados em relação ao planejado

Taxa de casos de testes (não executados, executados com sucesso, executados com falha) em relação ao planejado

Esforço ○○○○ Custo ○○○○
Escopo ○○○○ Qualidade ○○○○
Tempo ○○○○ Produtividade ○○○○

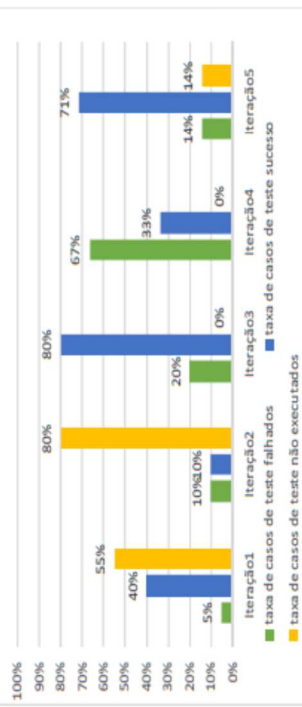
Objetivos

Objetivo de Medição

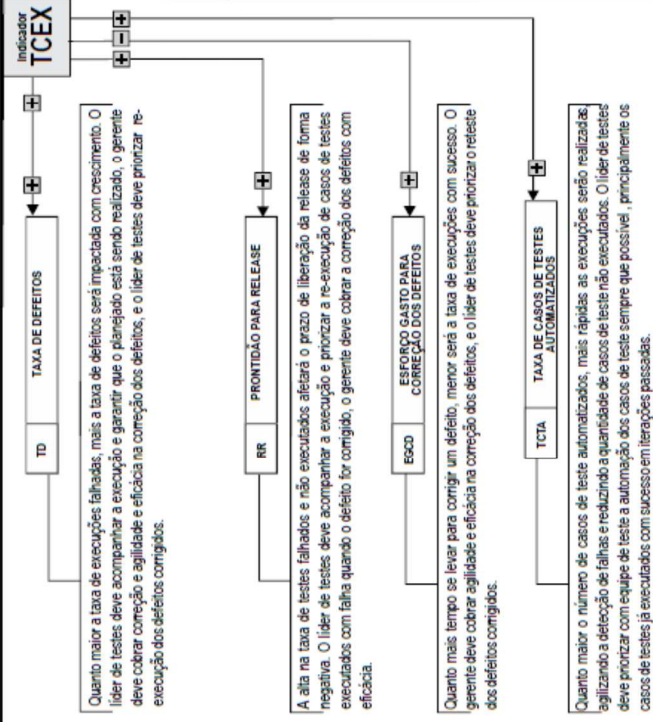
Aumentar a qualidade do software

Monitorar a execução dos testes

Representação Gráfica do Indicador



Procedimento de Análise Baseado em Relacionamentos



Indicador TCEX

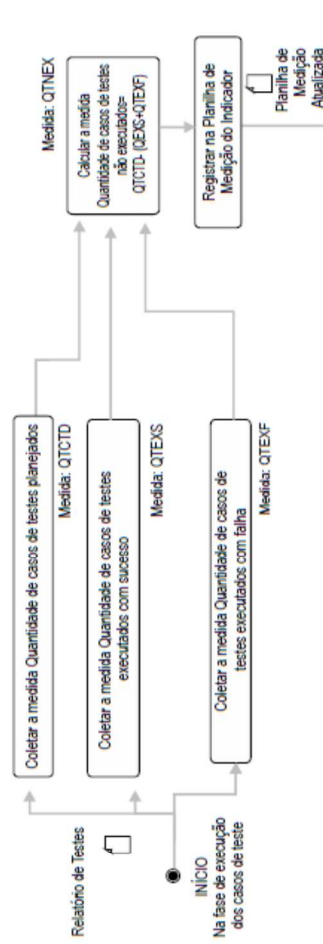
TD TAXA DE DEFETOS

RR PRONTIDÃO PARA RELEASE

ESGD ESFORÇO GASTO PARA CORREÇÃO DOS DEFETOS

TCTA TAXA DE CASOS DE TESTES AUTOMATIZADOS

Procedimento de Medição e Análise



Procedimento de Análise Baseado em Critérios

ID	Premissa	Possível Conclusão	Periodicidade coleta: ITERAÇÃO	Periodicidade análise: ITERAÇÃO	Responsáveis coleta: LÍDER DE TESTES	Responsáveis análise: LÍDER DE TESTES E GERENTE DO PROJETO
1	TCNEX < 50%	A taxa de não execução mostra que as execuções estão em andamento, porém ao menos nas últimas iterações a taxa deverá aproximar de 0%.				
2	TCNEX >= 50%	Caso de testes não executados podem ocultar defeitos de diversas severidades, inclusive os mais complexos que demandam mais tempo para correção.				
3	50% <= TCSEF <= 70%	Provavelmente há ainda defeitos que impactam a execução.				
4	TCSEF > 70%	HÁ MUITOS defeitos impactando a execução dos testes, com isso a taxa de casos de testes falhados está acima do previsto.				
5	TCSEF > 70%	A taxa de execução mostra que as execuções estão em andamento, porém ao menos nas últimas iterações a taxa deverá aproximar de 100%.				
6	40% <= TCSEF <= 70%	A quantidade de casos de teste falhados ou não executados ainda está impactando na taxa de sucesso de forma negativa.				
7	TCSEF <= 40%	As execuções com sucesso estão baixas, possivelmente há uma quantidade alta de execuções falhadas ou não executadas. Pode ser observável nas primeiras iterações.				

TCT - Tempo para correção dos testes

Indica o tempo para correção dos defeitos identificados nos testes em cada rodada de testes ou iteração.

Quanto tempo é gasto para corrigir os defeitos identificados em uma rodada de teste?

Esforço ○○○○ Custo ○○○○
Escopo ○○○○ Qualidade ○○○○
Tempo ●●●● Produtividade ○○○○

Objetivos

Objetivo Estratégico

Diminuir o tempo de entrega do software

Objetivo de Medição

Monitorar o tempo de liberação da versão para reteste

MR-MPS-SW

SE/MS

D

VER

Tipo

☒ controle
☐ qualidade
☐ desempenho

Representação Gráfica do Indicador

Iteração	Tempo de correção dos testes rodados testes 1	Tempo de correção dos testes rodados testes 2	Tempo de correção dos testes rodados testes 3	Total de tempo gasto para correção dos testes
Iteração01	16	9	5	30
Iteração02	8	7	15	30
Iteração03	3	24	25	52

Procedimento de Análise Baseado em Relacionamentos

Indicador TCT

TD

TAXA DE DEFEITOS

ESFORÇO GASTO PARA CORREÇÃO DOS DEFEITOS

Quanto maior a taxa de defeitos maior será o número de defeitos, causando aumento no tempo de correção dos testes. O gerente deve cobrar agilidade para correção dos defeitos e maior qualidade no código entregue para testes.

O aumento no tempo para correção dos defeitos, poderá impactar no esforço maior para correção dos defeitos. O gerente de projetos deve cobrar agilidade e maior eficácia na correção dos defeitos.

Procedimento de Medição e Análise

Medição

Relatório de Testes

INÍCIO

Na fase de execução dos casos de teste

Coletar a medida Quantidade de horas gastas para correção dos testes em cada rodada de testes

Medida: QTHRC

Registrar na Planilha de Medição do Indicador

Planilha de Medição Atualizada

Análise

Executar o Procedimento de Análise Baseado em Relacionamentos

Avaliar gráfico gerado e executar o Procedimento de Análise Baseado em Critérios disponibilizado na Ficha de Descrição do Indicador

Selecionar dados relevantes para a análise. (Especificar o período de medição ou tipos de registros).

Gerar o gráfico de comportamento do indicador na planilha de medição

TCT= Σ (QTHRC)

Medidas Base: QTHRC - Quantidade de horas gastas para correção dos testes em cada rodada de testes

Procedimento de Análise Baseado em Critérios

Medidas Base: QTHRC - Quantidade de horas gastas para correção dos testes em cada rodada de testes

Periodicidade coleta: ITERAÇÃO

Periodicidade análise: ITERAÇÃO

Responsáveis coleta: GERENTE DE PROJETOS

Responsáveis análise: GERENTE DE PROJETOS

Possível Conclusão

Possíveis Ações

1

TCT < 16 horas.

As correções estão sendo realizadas dentro de um tempo razoável.

Nenhuma ação é necessária.

2

16 < TCT <= 30 horas.

Provavelmente há muitas correções para serem realizadas, demandando um tempo acima do esperado.

Alvar com equipe de testes para analisar demora na correção dos testes.
Caso seja necessário solicitar apoio da equipe técnica para entendimento das correções necessárias.

3

TCT > 30 horas.

A quantidade de correções exigida pode ser superior à capacidade da equipe de fazê-las dentro do tempo esperado.

Providenciar mais recursos para agilizar a correção dos casos de teste.

TCTA - Taxa de casos de testes automatizados

Representa o número de casos de teste automatizados sobre o total de casos de teste existentes em cada iteração.

Qual o grau de automação de casos de teste?

Quais os casos de teste existentes em cada iteração.

Esforço ○○○○ Custo ○○○○
Escopo ○○○○ Qualidade ●●●●
Tempo ○○○○ Produtividade ●●○○

Objetivos

Objetivo Estratégico

Aumentar a qualidade do software

Objetivo de Medição

Monitorar a execução dos testes

Representação Gráfica do Indicador

Iteração	Taxa de casos de testes automatizados (%)	# casos de teste desenvolvidos	# casos de teste automatizados
Iteração1	10%	25	2
Iteração2	33%	20	7
Iteração3	40%	16	6
Iteração4	60%	12	7
Iteração5	80%	8	6
Iteração6	93%	1	24

Procedimento de Análise Baseado em Relacionamentos

A automação de caso de teste reduzirá o tempo de execução e detecção de defeitos, possibilitando antecipação ou redução do número de semanas para implementação. O líder de testes deverá acompanhar a execução para realizar o planejamento e monitorar a correção dos defeitos de forma que o reteste seja realizado o mais rápido possível. O gerente deverá apoiar cobrando agilidade na correção dos defeitos e maior qualidade no código.

RR PRONTIDÃO PARA RELEASE

TCEX TAXA DE CASOS DE TESTE EXECUTADOS EM RELAÇÃO AO PLANEJADO

EGT ESFORÇO GASTO NO TESTE

A automação proporcionará uma produtividade maior da equipe com uma quantidade maior de casos de testes executados com sucesso ou falhados. O líder de testes deverá acompanhar a execução para garantir que os casos de testes que foram automatizados estão sendo executados com sucesso e revisando e corrigindo com rapidez qualquer problema na automação.

O crescimento de casos de teste automatizados poderá afetar o esforço, uma vez que a tendência é que a execução ocorra de forma mais veloz. O líder de testes deve monitorar a equipe para que os testes automatizados sejam executados o quanto antes após a criação.

Procedimento de Medição e Análise

Medição

Relatório de Testes

INÍCIO

Na fase de planejamento de automação dos casos de teste

Coletar a medida Quantidade de casos de teste desenvolvidos

Medida: QTCTD

Registrar na Planilha de Medição do Indicador

Planilha de Medição Atualizada

Coletar a medida Quantidade de casos de testes automatizados

Medida: QTCTA

Análise

Executar o Procedimento de Análise Baseado em Relacionamentos

FM

Atualizar gráfico geral e executar o Procedimento de Análise Baseado em Critérios disponibilizado na Fone de Descrição do Indicador

Selecionar dados relevantes para a análise. (Especificar o período de medição ou tipos de registros).

Gerar gráfico de gráfico de relacionamento na planilha de medição

$TCTA = (QTCTA / QTCTD) * 100$

Responsáveis análise: LIDER DE TESTES

Responsáveis coleta: LIDER DE TESTES

Procedimento de Análise Baseado em Critérios

ID	Premissa	Possível Conclusão	Possíveis Ações
1	TCTA = 100%	Todos os casos de testes foram automatizados com sucesso.	Nenhuma.
2	TCTA > 70%	Quanto mais casos de teste automatizados, maior poderá ser a eficiência da execução dos testes. Idealmente este índice deverá ser alcançado ao menos nas últimas iterações.	Acompanhar junto com equipe de automação para continuar definindo mais testes automatizados.
3	40% <= TCTA <= 70%	Existe uma quantidade razoável de casos de teste que ainda não foram automatizados. Geralmente isso ocorre nas iterações iniciais, mas não é desejável nas iterações mais próximas do final do projeto.	Mapear casos de testes não automatizados e definir junto à equipe de testes e automação um plano de análise para aumentar a automação.
4	TCTA < 40%	Há MUITOS casos de teste que ainda não foram automatizados. O índice baixo é esperado nas iterações iniciais, mas deve aumentar com o progresso do projeto.	Mapear casos de teste e definir junto à equipe de testes e automação um plano de análise para aumentar a automação. O líder de testes deve sugerir ao Gerente do projeto a automação dos casos de teste, verificando quais ferramentas e recursos estão disponíveis.

MR-MPS-SW

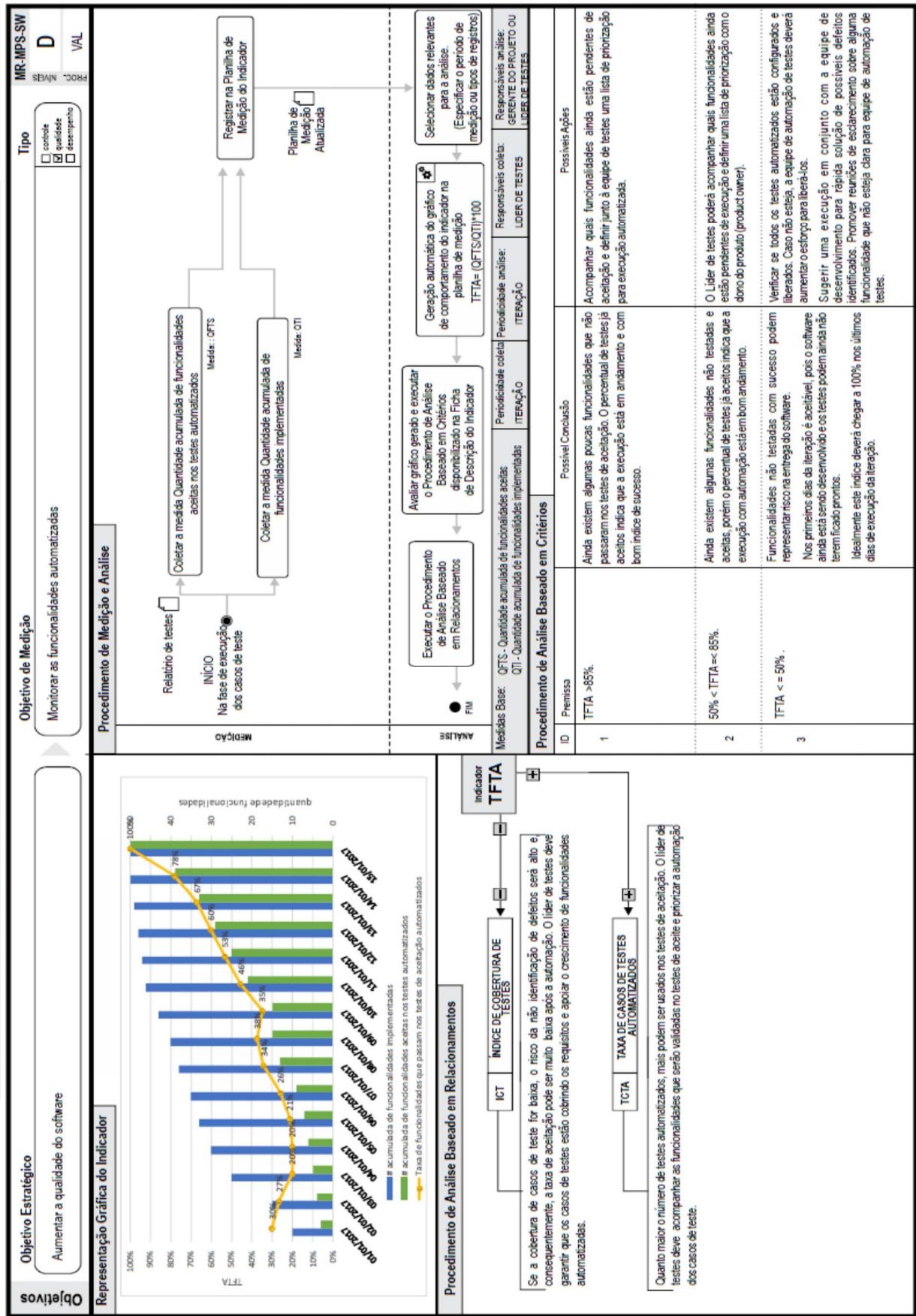
Proc. NÍVEL

D

VER

Tipo

☐ versão
☐ qualidade
☐ desempenho



Objetivos Objetivo Estratégico Diminuir o tempo de entrega do software		Objetivo de Medição Monitorar o tempo de liberação da versão para reteste		MR-MPS-SW Tipo ☐ controle ☒ qualidade ☐ desempenho D VER	
Representação Gráfica do Indicador		Procedimento de Medição e Análise Medição Relatório de testes INÍCIO Na fase de execução dos casos de teste Planos do Projeto Coletar a medida Quantidade de horas gastas para correção dos defeitos de severidade 1, severidade 2, severidade 3 Medida: QTHGSV Coletar a medida Quantidade de defeitos de severidade 1, severidade 2, severidade 3 Medida: QTDSEV Coletar a medida SLA de defeitos acordado para severidade 1, severidade 2, severidade 3 Medida: SLADFSEV Análise Executar o Procedimento de Análise Baseado em Relacionamento FIM Avaliar gráfico gerado e executar o Procedimento de Análise Baseado em Critérios disponibilizado na Ficha de Descrição do Indicador TMCDSEV = (QTHGSV/QTDSEV) Selecionar dados relevantes para a análise. (Especificar o período de medição ou tipos de registros).		Procedimento de Análise Baseado em Relacionamentos Indicador TMCD TD TAXA DE DEFEITOS EGCD ESFORÇO GASTO NA CORREÇÃO DOS DEFEITOS Quanto maior a taxa de defeitos não resolvidos maior tende a ser o tempo médio de correção devido ao volume de trabalho. O gerente deve cobrar agilidade e eficácia na correção dos defeitos e cobrar maior qualidade da equipe de desenvolvimento. Caso o esforço para remoção dos defeitos aumente, o tempo médio de resolução poderá crescer além do planejado no SLA do projeto. O gerente de projetos deve acompanhar o tempo e cobrar agilidade na correção dos defeitos com eficácia e melhor qualidade no código entregue para testes.	
Medidas Base: QTHGSV - Quantidade de horas gastas para correção de defeitos QTDSEV - Quantidade de defeitos de severidade - SLADFSEV - SLA de defeitos		Periodicidade coleta: ITERAÇÃO		Responsáveis análise: GERENTE DO PROJETO	
Procedimento de Análise Baseado em Critérios		Periodicidade coleta: ITERAÇÃO		Responsáveis coleta: LÍDER DE TESTES e GERENTE DO PROJETO	
ID 1		Possível Conclusão Indica que o tempo médio de correção está dentro de uma margem de controle, não representando risco no momento.		Possíveis Ações O gerente de projetos deve monitorar o tempo de correção dos defeitos para evitar o tempo médio ultrapassar o SLA de defeitos.	
2		Indica que o tempo médio de resolução está em ponto de atenção máxima, qualquer queda no desempenho acarretará quebra do SLA.		O gerente deverá mobilizar a equipe para ações rápidas de correção de defeitos e comunicá-la quanto às possíveis penalizações por razão de SLA perdido. Acrescentar mais recursos para análise de defeitos.	
3		Indica que o tempo médio de resolução está em ponto de quebra do SLA.		Acrescentar mais recursos para análise de defeitos. Líder de testes deve continuar monitorando o tempo de correção dos defeitos e reportar ao Gerente do projeto o tempo médio de correção para evitar quebra do SLA de defeitos.	
4		O tempo médio de correção está acima do SLA do projeto.		O Gerente do projeto deverá solicitar à equipe de desenvolvimento uma análise da demora na resolução e um plano de ação para resolução dos defeitos com tempo médio abaixo do SLA de defeitos. Acrescentar mais recursos para análise de defeitos ou negociar com o cliente o SLA acordado para ganhar mais tempo.	
				O gerente deverá mobilizar a equipe para ações rápidas de correção de defeitos e comunicar quanto às possíveis penalizações por razão de SLA de defeitos pendido.	

Apêndice III - Planilhas de apoio para coleta de medidas e geração do gráfico dos Indicadores

Para apoio na coleta das medidas, cálculo dos indicadores e geração dos gráficos, foram criadas planilhas que pudessem calcular automaticamente através das formulas citadas nas fichas e geração dos gráficos.

Foi criada uma planilha contendo 16 abas, uma aba foi criada para ser utilizada como menu de atalho para todos os outros indicadores e também para receber informações básicas sobre o projeto de testes como nome do projeto, responsáveis pela coleta e pela análise e período que está sendo feito a coleta das medidas. Também é listado o nome dos indicadores com o link para a aba do indicador e uma descrição sobre o indicador. Na Figura 40 é detalhado as informações do menu da planilha de coletas.

PLANILHA DE COLETA DE DADOS	
NOME DO PROJETO	XYZ
RESPONSÁVEL PELA COLETA	FULANO DE TAL
RESPONSÁVEL PELA ANÁLISE	BELTRANO
Período das Iterações/coleta:	18.17-19.08.17
Atalho para indicadores	Descrição do Indicador
DD - Densidade de defeitos	Densidade de defeitos (por projeto, por iteração, por complexidade, por severidade). Será calculado pela quantidade de defeitos por severidade ou complexidade acumulado por iteração sobre o tamanho do projeto conforme definição (linhas de código, pontos de função, histórias de usuários).
EGCD - Esforço gasto para correção de defeitos	Esforço gasto para correção de defeitos por tipo de teste (unidade, integração, funcional e aceitação). Será calculado pela porcentagem de horas gastas na correção dos defeitos sobre total de horas gastas na iteração.
EGT - Esforço gasto nos testes	Esforço gasto por tipo de teste (unidade, integração, funcional e aceitação). Será calculado pela porcentagem de horas gastas em testes sobre total de horas da gastas iteração
ICT - Índice de Cobertura de Testes	Índice de cobertura de teste, considerando total e por tipo de teste (unidade, integração, funcional e aceitação). Este indicador será calculado de pelo total de requisitos já desenvolvidos que tenham ao menos um caso de teste associado sobre o total de requisitos do projeto já desenvolvidos do projeto/iteração.
QDNC - Quantidade de defeitos não cobertos por casos de teste	Quantidade de defeitos identificados no software pela equipe não capturados pelos casos de teste.
QDRC - Quantidade de defeitos reportados pelo cliente	Quantidade de defeitos reportados pelo cliente (por severidade). Será calculado pelo somatório de defeitos acumulado por severidade por mês/período definido pelo projeto/iteração.
RR - Pronto para release	Número de semanas previstas para disponibilização da release com base no número de defeitos identificados. O indicador sinalizará em quantas semanas a release estará liberada para implementação.
TACT - Taxa de aumento de casos de testes entre releases	Taxa de aumento de casos de testes em cada release. Será calculado pelo percentual de aumento de casos de testes entre releases.
TCEX - Taxa de casos de testes executados em relação ao planejado	Taxa de casos de testes (não executados, executados com sucesso, executados com falha) em relação ao planejado. Será calculado para cada medida sobre o total de caso de teste planejados para a iteração/projeto.
TCT - Tempo para correção dos testes	Será calculado para verificar o tempo para correção dos defeitos identificados nos testes em cada rodada de testes ou iteração.
TCTA - Taxa de casos de testes automatizados	Este indicador representa o número de casos de testes automatizados sobre o total de casos de testes existentes em cada iteração.
TD - Taxa de Defeitos	Taxa de defeitos (abertos, analisados e não resolvidos, removidos) por iteração e por projeto. Total de defeitos (cada tipo) sobre o total de defeitos identificados no projeto/iteração
TDD - Tempo entre detecção de defeitos pelo cliente	Tempo decorrido entre detecções consecutivas de defeitos por parte do cliente. Será calculado pelo número de dias em espera, até a correção do defeito para cada dia da iteração.
TETA - Taxa de funcionalidades que passaram nos testes de aceitação automatizados	Taxa de funcionalidades que foram implementadas e automatizadas. Calculado pela quantidade de funcionalidades aceitas nos testes automatizados sobre o total de funcionalidades implementadas no projeto.
TMD - Tempo médio para correção de um defeito	Será calculado para verificar qual o tempo médio de correção de defeitos por cada nível de severidade, considerando também o SLA definido pelo projeto.

Figura 40- Menu da planilha de coletas

As abas subsequentes são utilizadas uma para cada indicador, com instruções de coleta, cálculo do indicador e geração do gráfico de análise. Nas figuras abaixo são descritas cada planilha de coleta de cada indicador criado na dissertação.

DD - Densidade de defeitos

INDICADOR:	Instruções de preenchimento:									
DD - Densidade de defeitos	1- Preencher apenas as colunas em cor amarela nas tabelas de cada iteração com quantidade de defeitos de cada tipo (complexidade e severidade) 2 - A coluna TS deverá ser preenchida com o tamanho do software (por exemplo, por 1000 linhas de código, pontos de função ou histórias de usuário), como definido na planilha de estimativa do projeto 3-A equipe de projeto deverá definir os limites para a densidade de defeitos (baixa, médio, alta, muito alta) para o projeto. Ex. baixa<10, média>30, alta>50, muito alta>70									
Medidas ----->	QTDCP									TS
	# defeitos por complexidade		DD	Densidade de defeitos acumulada	Tamanho da iteração em linhas de código					
Iterações	Alta	Média	Baixa							
Iteração1				#DIV/0!	#DIV/0!					
Iteração2				#DIV/0!	#DIV/0!					
Iteração3				#DIV/0!	#DIV/0!					
Iteração4				#DIV/0!	#DIV/0!					
Iteração5				#DIV/0!	#DIV/0!					

Figura 41- Densidade de defeitos

EGCD – Esforço gasto para correção dos defeitos

INDICADOR:	Instruções de preenchimento:									
EGCD - Esforço gasto para correção de defeitos	1-Preencher apenas as colunas em cor amarela na tabela									
Medidas ----->	QHCD									QTHRIT
	horas para correção dos defeitos unidade	horas para correção dos defeitos integração	horas para correção dos defeitos funcional	horas para correção dos defeitos aceitação	EGCD		total de horas de correção		total de horas gastas na iteração	
Iteração1					#DIV/0!		0			
Iteração2					#DIV/0!		0			
Iteração3					#DIV/0!		0			
Iteração4					#DIV/0!		0			

Figura 42 - Esforço gasto para correção dos defeitos

EGT – Esforço gasto nos testes

INDICADOR:	Instruções de preenchimento:					
EGT- Esforço gasto nos testes	1-Preencher apenas as colunas em cor amarela na tabela					
MEDIDAS:	QTHRT					QTHRIT
	Hrs testes unidade	Hrs testes integração	Hrs testes funcional	Hrs testes aceitação	EGT	total de horas gastas na execução
Iteração1					#DIV/0!	0
Iteração2					#DIV/0!	0
Iteração3					#DIV/0!	0
Iteração4					#DIV/0!	0

Figura 43 - Esforço gasto nos testes

ICT – Índice de Cobertura de testes

INDICADOR:		ICT - Índice de Cobertura de Testes					
Instruções de preenchimento:							
1- A planilha do Indicador é calculada automaticamente pelas fórmulas na tabela do INDICADOR							
2-Para cada iteração deverá ser incluído o quadro com a quantidade de linhas para cada requisito (RQx)							
3- Adicionar a nova iteração no quadro principal (primeira tabela) e adicionar no gráfico do ICT							
4-A coluna de requisitos pode ser nomeada com o nome principal do requisito ou código adotado no projeto							
5-Preencher apenas as colunas em cor amarela nas tabelas de cada iteração							
TABELA DO INDICADOR		CALCULADO AUTOMATICAMENTE					
ITERAÇÕES		# CTs Unidade	# CTs Integração	# CTs Funcional	# CTs Aceitação	Índice de Cobertura de Testes	
Iteração 1		0	0	0	0	0%	
Iteração 2		7	11	2	3	75%	
Iteração 3		9	4	7	4	80%	
Iteração 4		1	5	5	11	100%	
Preencher as colunas em amarelo com quantidade de casos de teste de cada tipo criados para os requisitos (RQx)							
ITERAÇÃO 1		Quantidade de Casos de Testes por tipo de Teste				QCTTT	
REQUISITOS	Unidade	Integração	Funcional	Aceitação	QTRCTA	ICT	
RQ1					0		
RQ2					0		
RQ3					0		
RQ4					0		
TOTAL REQUISITOS (QTR)		4			0	0%	
#CASOS DE TESTE		0	0	0	0		

Figura 44 - Índice de cobertura de testes

QDNC- Quantidade de defeitos não cobertos por casos de teste

INDICADOR:	Instruções de preenchimento:				
QDNC- Quantidade de defeitos não cobertos por casos de teste	1-Preencher apenas as colunas em cor amarela na tabela				
MEDIDAS:	QTDFT	QTDfNC	TDNC		
		#defeitos não cobertos pelos casos da release de teste	taxa de defeitos não cobertos		
Release1			#DIV/0!		
Release2			#DIV/0!		
Release3			#DIV/0!		

Figura 45 - Quantidade de defeitos não cobertos por casos de teste

QDRC - Quantidade de defeitos reportados pelo cliente

INDICADOR:	Instruções de preenchimento:				
QDRC - Quantidade de defeitos reportados pelo cliente	1-Preencher apenas as colunas em cor amarela na tabela				
	2-A coluna SLA de defeitos deverá ser informada de acordo com o planejado pela organização				
Medidas -----	QDSV[n]			QDRC	SLAD
				# defeitos reportados pelo cliente	SLA defeitos
Período	# defeitos Sev1	# defeitos Sev2	# defeitos Sev3		
Mês1				0	
Mês2				0	
Mês3				0	
Mês4				0	
Mês5				0	
Mês6					
Mês7					
Mês8					
Mês9					
Mês10					
Mês11					
Mês12					

Figura 46 - Quantidade de defeitos reportados pelo cliente

RR – Prontidão para Release

INDICADOR:	Instruções de preenchimento:					
RR - Prontidão para release	1-Preencher apenas as colunas em cor amarela na tabela					
Medidas -----	QTDF	QTDFR	QTCTEX	QTCTS		
	# de casos # defeitos de testes # testes com #defeitos removidos executados sucesso				RR - Prontidão para release	RR - Planejado
Semana1					#DIV/0!	4
Semana2					#DIV/0!	3
Semana3					#DIV/0!	2
Semana4					#DIV/0!	1
Total	0	0	0	0		

Figura 47 - RR – Prontidão para Release

Taxa de aumento de casos de testes entre releases

INDICADOR:	Instruções de preenchimento:					
TACT - Taxa de aumento de casos de teste entre releases	1-Preencher apenas as colunas em cor amarela na tabela					
Medidas	QTCT_ATUAL e QTCT_ANTERIOR					
Release	# casos de teste implementados		TACT			
Release01			0%			
Release02			#DIV/0!			
Release03			#DIV/0!			
Release04			#DIV/0!			
Release05			#DIV/0!			

Figura 48 - Taxa de aumento de casos de testes entre releases

TCEX - Taxa de casos de testes executados em relação ao planejado

INDICADOR:	Instruções de preenchimento:						
TCEX - Taxa de casos de testes executados em relação ao planejado	1-Preencher apenas as colunas em cor amarela na tabela						
MEDIDAS:				QTCTD	QTEXF	QTEXS	QTNEX
TCEX--->	TCEXF	TCEXS	TCNEX	Preencer apenas as colunas em amarelo			
ITERAÇÕES	taxa de casos de teste falhados	taxa de casos de teste sucesso	taxa de casos de teste não executados	#casos de teste planejados	#executados com falha	#executados com sucesso	# não executados
Iteração Junho	#DIV/0!	#DIV/0!	#DIV/0!				0
Iteração Julho	#DIV/0!	#DIV/0!	#DIV/0!				0
Iteração Agosto	#DIV/0!	#DIV/0!	#DIV/0!				0

Figura 49 - Taxa de casos de testes executados em relação ao planejado

TCT - Tempo para correção dos testes

INDICADOR:	Instruções de preenchimento:						
TCT - Tempo para correção dos testes	1-Preencher apenas as colunas em cor amarela na tabela						
MEDIDAS:	QTHRC		QTHRC		QTHRC		
Preencer apenas as colunas em amarelo							
	Tempo de correção dos testes						
ITERAÇÃO	rodada testes 1	total defeitos rod1	rodada testes 2	total defeitos rod2	rodada testes 3	total defeitos rod3	TCT
Iteração1							0
Iteração2							0
Iteração3							0
Iteração4							
Iteração5							

Figura 50 - Tempo para correção dos testes

Taxa de casos de testes automatizados

INDICADOR:	Instruções de preenchimento:		
TCTA - Taxa de casos de testes automatizados	1-Preencher apenas as colunas em cor amarela na tabela		
Medidas -----	QTCTD	QTCTA	
Iteração	# casos de teste desenvolvidos	# casos de teste automatizados	TCTA
Iteração1			#DIV/0!
Iteração2			#DIV/0!
Iteração3			#DIV/0!
Iteração4			#DIV/0!
Iteração5			#DIV/0!
Iteração6			#DIV/0!

Figura 51 - Taxa de casos de testes automatizados

Taxa de defeitos

INDICADOR	Instruções de preenchimento:						
TD - Taxa de Defeitos	1-Preencher apenas as colunas em cor amarela na tabela de cada linha de iteração para as medidas:						
	QDA - Quantidade de defeitos abertos						
	QDANR - Quantidade de defeitos analisados e não removidos						
	QDR - Quantidade de defeitos removidos						
Medidas ----->	QDA	QDANR	QDR		TAXA		
ITERAÇÕES	ABERTO	ANALISADOS e NÃO REMOVIDOS	REMOVIDOS	TOTAL DEFEITOS	TD (Abertos)	TD(Analisados e Não Resolvidos)	TD (Removidos)
Iteração 1				0	#DIV/0!	#DIV/0!	#DIV/0!
Iteração 2				0	#DIV/0!	#DIV/0!	#DIV/0!
Iteração 3				0	#DIV/0!	#DIV/0!	#DIV/0!
Iteração 4				0	#DIV/0!	#DIV/0!	#DIV/0!
Iteração 5				0	#DIV/0!	#DIV/0!	#DIV/0!

Figura 52 - Taxa de defeitos

TDD - Tempo entre detecção de defeitos pelo cliente

INDICADOR:	Instruções de preenchimento:		
TDD - Tempo entre detecção de defeitos pelo cliente	1- Preencher apenas as colunas em cor amarela 2 - Preencher com a data de identificação do defeito e quantidade de defeitos identificados no dia		
Medidas--> DTDF	QTD		
Data - DTDF	Quantidade de defeitos detectados no dia	TDD	
		0	
		0	
		0	
		0	
		0	
		0	
		0	
		0	
		0	

Figura 53 - Tempo entre detecção de defeitos pelo cliente

TFTA - Taxa de funcionalidades que passam nos testes de aceitação automatizados

INDICADOR:	Instruções de preenchimento:		
TFTA - Taxa de funcionalidades que passam nos testes de aceitação automatizados	1-Preencher apenas as colunas em cor amarela para cada data 2- Incluir novas datas na planilha quando novos testes forem realizados 3-Preencher com quantidade de funcionalidade implementadas e automatizadas na iteração		
Medidas ----->	QFTS	QTI	
Datas	# acumulada de funcionalidades aceitas nos testes automatizados	# acumulada de funcionalidades implementadas	TFTA
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!
			#DIV/0!

Figura 54 - Taxa de funcionalidades que passam nos testes de aceitação automatizados

TMCD - Tempo médio para correção de um defeito

INDICADOR:	Instruções de preenchimento:											
TMCD - Tempo médio para correção de um defeito	1-Preencher apenas as colunas em cor amarela na tabela											
MEDIDAS:	QTHGSV1	QTDFSV1	QTHGSV2	QTDFSV2	QTHGSV3	QTDFSV3	TMCDV1	TMCDV2	TMCDV3	SLADFSEV1	SLADFSEV2	SLADFSEV3
	Tempo de correção do defeito						tempo médio de correção defeito			SLA de defeitos		
	#horas gastas sev1	# defeitos sev1	#horas gastas sev2	# defeitos sev2	#horas gastas sev3	# defeitos sev3	sev1	sev2	sev3	sev1	sev2	sev3
Iteração1							#DIV/0!	#DIV/0!	#DIV/0!			
Iteração2							#DIV/0!	#DIV/0!	#DIV/0!			
Iteração3							#DIV/0!	#DIV/0!	#DIV/0!			
Iteração4												
Iteração5												

Figura 55 - Tempo médio para correção de um defeito