

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Raquel Aparecida Pegoraro

**Métricas de avaliação para abordagens ágeis em
projetos de software**

Porto Alegre

2014

Raquel Aparecida Pegoraro

**Abordagens ágeis em projetos de software: um estudo exploratório das métricas de
avaliação**

Tese submetida ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul como requisito parcial à obtenção do título de Doutor em Engenharia, na área de concentração em Qualidade.

Orientador: Professor Istefani Carísio de Paula, Dr

Porto Alegre

2014

Raquel Aparecida Pegoraro

Abordagens ágeis em projetos de software: um estudo exploratório das métricas de avaliação

Esta tese foi julgada adequada para a obtenção do título de Doutor em Engenharia e aprovada em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal do Rio Grande do Sul.

Prof. Orientador Istefani Carísio de Paula,
Dr
Orientador PPGEP/UFRGS

Prof. José Luis Duarte Ribeiro, Dr
Coordenador PPGEP/UFRGS

Banca Examinadora:

Professor Daniel Capaldo Amaral, Dr (EESC/USP)

Professor Liane Werner, Dr (PPGEP/UFRGS)

Professor Marly Monteiro de Carvalho, Dr (POLI/USP)

Dedicatória

*Dedico este trabalho a minha filha (Luiza), ao
meu esposo (Josenir), ao meu pai (João em
memória) e a minha mãe (Iria).*

Amo vocês!

AGRADECIMENTOS

Agradeço à Deus pela luz em meu caminho, dando força e perseverança.

À minha filha maravilhosa que mesmo tão pequena foi sempre compreensiva nos muitos momentos que não pude dar a atenção merecida a ela. Ao meu esposo pelo amor, companheirismo e apoio. À minha mãezinha que sempre me apoiou em todos os momentos da minha vida.

Aos familiares e amigos que me apoiaram.

Agradeço de forma especial a minha orientadora Profa. Istefani Carísio de Paula pelo incentivo, ensinamentos e amizade. Foi realmente uma mestre e se tornou referência para minha vida.

Às professoras Marly Monteiro de Carvalho e Liane Werner pelas contribuições e conhecimentos repassados.

Ao professor Daniel Capaldo Amaral por ter aceitado fazer parte da minha banca.

À UFFS (Universidade Federal da Fronteira Sul) onde tenho o privilégio de trabalhar e aos professores do curso de Ciência da Computação em que atuo, pelo apoio e incentivo.

Às empresas e profissionais que permitiram realizar entrevistas e compartilharam o seu conhecimento.

Aos colegas do DINTER pela amizade.

À UFRGS e à Unochapecó pela oportunidade de concretizar este sonho.

RESUMO

A adoção de métodos ágeis é uma forma eficaz de reduzir o ciclo de entrega no desenvolvimento de software, fornecendo software de qualidade em curto espaço de tempo. Porém, a adoção desta nova abordagem de desenvolvimento de software torna necessário repensar a forma de medir e controlar os projetos. Os métodos ágeis não tratam claramente sobre os assuntos utilização de métricas e adoção de um processo de medição para projetos de softwares desta natureza, faltando estudos que tragam recomendações em como estabelecer métricas para projetos ágeis e como adotar um processo de medição compatível com esta abordagem. Visando contribuir neste sentido esta tese tem como objetivo definir um conjunto de métricas adequadas às necessidades de monitoramento e propor um processo de medição, compatível com a abordagem ágil de desenvolvimento de software. Como método de pesquisa foi realizado um trabalho exploratório através de revisão de literatura e de pesquisa de campo com entrevista em profundidade em empresas de desenvolvimento de software experientes em métodos ágeis. O primeiro resultado do trabalho é a apresentação de um conjunto de métricas consolidados para auxiliar na gestão de projetos ágeis de desenvolvimento de software nas fases de projeto/*releases*, iteração e diário. As métricas são especificadas detalhadamente contendo as informações necessárias para seu entendimento e aplicação. Posteriormente é proposto um processo de medição compatível com a abordagem ágil de desenvolvimento de software, visando apoiar as empresas que adotam métodos ágeis na definição de métricas adequadas para suas necessidades de medição e no monitoramento. O processo contempla as fases de planejamento de medição, monitoramento da iteração, ações da iteração, monitoramento do projeto/*releases*, ações sobre o projeto/*releases* e avaliação final, sendo que em cada fase do processo são apresentadas recomendações para a sua implantação. O processo está estruturado num ciclo de gestão baseado em etapas de planejar, executar, verificar, atuar, refletir e melhorar, respeitando as características dos projetos ágeis de desenvolvimento de software, e na proposição de um quadro visual de monitoramento que permita a gestão do processo de medição de forma visual. Além dos resultados apresentados foram deixadas hipóteses e recomendações para trabalhos futuros.

Palavras-chave: Métodos ágeis; Métricas; Processo de medição; Projetos de desenvolvimento de software.

ABSTRACT

The adoption of agile methods is effective way to reduce the delivery cycle on software development, providing quality software in a short time. However, the adoption of this new approach to software development is necessary rethink how to measure and control projects. Agile methods not explain about adoption metrics and measurement process for software projects of this approach, lacking studies providing recommendations on how to establish metrics for agile projects and how to adopt a process measurement compatible with this approach. Contributing this thesis goal produce a set of metrics adequate monitoring needs and propose a measurement process compatible with agile software development. Method of research was exploratory through literature review and field research with depth interviews in experienced software development companies in agile methods. The first result of this work is the presentation of a consolidated metrics set to help the management of agile development at the phases of project/releases, iteration and daily. The metrics are specified detailed containing the information necessary for their understanding and application. Later we propose a measurement process compatible with agile approach to software development, to support businesses that adopt agile methods in defining adequate metrics for your measurement needs and monitoring. The process include the steps of measurement planning, monitoring of the iteration, the iteration actions, monitoring project/releases, actions on the project/releases and final evaluation, in each stage of the process provides recommendations for implementation. The process is structured in a management cycle based on steps to plan, implement, check, act, reflect and improve, respecting the characteristics of agile software development projects and propose a visual tracking board that allows for the management of the measurement process. In addition to the results were allowed hypotheses and recommendations for future work.

Key words: Agile methods; Metrics; The measurement process; Software development projects.

LISTA DE FIGURAS

Figura 1. Etapas de pesquisa utilizadas na tese	18
Figura 2. Planejamento cebola baseado em várias fases de planejamento	33
Figura 3. Fases e subfases do ciclo de vida de um projeto ágil.....	34
Figura 4. Abordagem Gols, Questions, Metrics.	40
Figura 5. Número de métricas por fase da pesquisa	68
Figura 6. Exemplo de gráfico <i>burndown</i> da iteração.....	73
Figura 7. Exemplo de gráfico <i>burnup</i> da <i>release</i> /projeto	75
Figura 8. Exemplo de quadro de tarefas kanban	79
Figura 9. Exemplo diagrama de fluxo cumulativo	85
Figura 10. ‘ <i>Emoticons</i> ’ utilizados para pesquisa de motivação.....	94
Figura 11. Métricas por fase do projeto de análise.....	120
Figura 12. Relação entre as fases do processo e o ciclo lógico de gestão	122
Figura 13. Sinais visuais de alerta do quadro visual de monitoramento do projeto	125

LISTA DE QUADROS

Quadro 1. Principais diferenças entre o modelo tradicional (cascata) e o modelo ágil.....	28
Quadro 2. Categorização das métricas	43
Quadro 3. <i>Strings</i> de busca.....	44
Quadro 4. Estudos selecionados na revisão da literatura.....	46
Quadro 5. Síntese dos dados das empresas pesquisadas e de seus entrevistados.....	52
Quadro 6. Fases dos projetos e frequência de identificação por empresa que utiliza	59
Quadro 7: Funções envolvidas nos projetos por empresa conforme declarações da pesquisa de campo.....	62
Quadro 8: Síntese das atividades desempenhadas pelas funções dos envolvidos nos projetos ágeis	63
Quadro 9. Especificação da métrica: Acurácia das estimativas (tempo).....	70
Quadro 10. Especificação da métrica: <i>Burndown</i> da iteração	73
Quadro 11. Especificação da métrica: <i>burnup</i> da <i>release</i> /projeto.....	75
Quadro 12. Especificação da métrica: Cobertura de testes automatizados	77
Quadro 13. Especificação da métrica: <i>Cycle time</i> (tempo de ciclo para concluir uma tarefa). 79	
Quadro 14. Especificação da métrica: Defeitos em aberto (defeitos não solucionados).....	81
Quadro 15. Especificação da métrica: Defeitos encontrados pelo cliente.....	82
Quadro 16. Especificação da métrica: Densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto).....	83
Quadro 17. Especificação da métrica: Diagrama de fluxo cumulativo	85
Quadro 18. Especificação da métrica: Esforço de administração do projeto (% tempo despendido em atividades de administração do projeto).....	87
Quadro 19. Especificação da métrica: Funcionalidades Testadas e Entregues	88
Quadro 20. Especificação da métrica: Grau de atendimento aos requisitos funcionais	89
Quadro 21. Especificação da métrica: Índice de defeitos encontrados na fase de testes	90
Quadro 22. Especificação da métrica: Índice de histórias por iteração.....	92
Quadro 23. Especificação da métrica: <i>Lead time</i> (tempo de entrega das funcionalidades ao cliente)	93
Quadro 24. Exemplo de <i>Niko-Niko Calendar</i>	94
Quadro 25. Especificação da métrica: Média da motivação dos membros da equipe.....	95
Quadro 26. Especificação da métrica: Nível de satisfação do cliente	96

Quadro 27. Especificação da métrica: Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões).....	98
Quadro 28. Especificação da métrica: Número de tarefas não concluídas na iteração	99
Quadro 29. Especificação da métrica: Severidade média dos defeitos	100
Quadro 30. Especificação da métrica: Taxa de acerto na estimativa das tarefas	101
Quadro 31. Especificação da métrica: Taxa de dívida técnica (problemas técnicos a serem resolvidos)	102
Quadro 32. Especificação da métrica: Taxa de retrabalho	104
Quadro 33. Especificação da métrica: Tempo investido em mudanças	105
Quadro 34. Especificação da métrica: Tempo investido em tarefas não planejadas	106
Quadro 35. Especificação da métrica: Tempo médio de resolução dos impedimentos	107
Quadro 36. Especificação da métrica: <i>Throughput</i> (número de tarefas entregues por período de tempo)	109
Quadro 37. Especificação da métrica: Total de horas trabalhadas no projeto.....	110
Quadro 38. Especificação da métrica: Variação do número de defeitos entre iterações.....	111
Quadro 39. Especificação da métrica: Variação dos custos (entre estimado e realizado).....	113
Quadro 40. Especificação da métrica: Velocidade média diária da equipe.....	115
Quadro 41. Especificação da métrica: <i>Work in Progress</i> (WIP).....	116
Quadro 42. Relação entre as fases do processo e o momento em que deverá ser atualizado o quadro visual de monitoramento	124
Quadro 43. Síntese dos dados das empresas pesquisadas e do perfil dos entrevistados que participaram da avaliação do processo proposto	131
Quadro 44. Avaliação do processo pelos entrevistados	133

LISTA DE ABREVIATURAS

IR – Indicadores de resultado

IT – Indicadores de tendência

KLOC – Número de linhas de código

KPIs (*Key Performance Indicators*) – Indicadores chaves de desempenho

PO (*Product Owner*) – Proprietário do produto

ROI – Retorno sobre o investimento

RTF (*Running testing features*) – Funcionalidades testadas e entregues

WIP (*Work in Progress*) – Trabalhos em progresso

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	JUSTIFICATIVA E QUESTÃO DA PESQUISA	15
1.2	OBJETIVOS E QUESTÕES DE PESQUISA	16
1.3	MÉTODO DE TRABALHO	17
1.3.1	<i>Fase 1 - Investigação na literatura</i>	<i>19</i>
1.3.2	<i>Fase 2 - Pesquisa de campo</i>	<i>19</i>
1.3.3	<i>Fase 3 - Proposição do processo de medição</i>	<i>22</i>
1.3.4	<i>Fase 4 - Avaliação do processo de medição proposto.....</i>	<i>23</i>
1.4	DELIMITAÇÕES E LIMITAÇÕES DO TRABALHO	23
1.5	ESTRUTURA DO TRABALHO	24
2	REFERENCIAL TEÓRICO.....	25
2.1	MÉTODOS ÁGEIS, VALORES, PRINCÍPIOS E CONTEXTUALIZAÇÃO.....	26
2.1.1	<i>Fases de um projeto ágil</i>	<i>31</i>
2.1.2	<i>Interessados e envolvidos em projetos ágeis de software.....</i>	<i>34</i>
2.2	MEDIÇÃO DE DESEMPENHO DE PROJETOS DE SOFTWARE	36
2.2.1	<i>Fundamentos sobre medição e avaliação de desempenho</i>	<i>37</i>
2.2.2	<i>Planejamento de medição.....</i>	<i>38</i>
2.2.3	<i>Utilização de métricas nos métodos ágeis.....</i>	<i>40</i>
2.2.4	<i>Categorias de avaliação das métricas para projetos ágeis de software</i>	<i>42</i>
2.3	MÉTRICAS UTILIZADAS PARA PROJETOS DE SOFTWARE CITADAS NA LITERATURA.....	44
2.4	CONSIDERAÇÕES SOBRE O REFERENCIAL.....	50
3	PESQUISA DE CAMPO.....	51
3.1	CONTEXTO DAS ORGANIZAÇÕES ESTUDADAS E PERFIL DOS ENTREVISTADOS	51
3.2	MÉTRICAS UTILIZADAS PELAS EMPRESAS.....	52
3.3	FASES TÍPICAS DOS PROJETOS ÁGEIS.....	59
3.4	FUNÇÕES DOS ENVOLVIDOS NO PROJETO	61
3.5	CONSIDERAÇÕES SOBRE O CAPÍTULO 3	64
4	MÉTRICAS DE AVALIAÇÃO EM PROJETOS ÁGEIS DE SOFTWARE.....	66
4.1	CONJUNTO DE MÉTRICAS.....	66
4.1.1	<i>Acurácia das estimativas (tempo).....</i>	<i>69</i>
4.1.2	<i>Burndown da iteração</i>	<i>72</i>
4.1.3	<i>Burnup da release/projeto</i>	<i>74</i>
4.1.4	<i>Cobertura de testes automatizados.....</i>	<i>77</i>
4.1.5	<i>Cycle time (tempo de ciclo para concluir uma tarefa)</i>	<i>78</i>

4.1.6	<i>Defeitos em aberto (defeitos não solucionados)</i>	80
4.1.7	<i>Defeitos encontrados pelo cliente</i>	81
4.1.8	<i>Densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto)</i> ..	82
4.1.9	<i>Diagrama de fluxo cumulativo</i>	84
4.1.10	<i>Esforço de administração do projeto (% tempo despendido em atividades de administração do projeto)</i>	86
4.1.11	<i>Funcionalidades Testadas e Entregues (Running testing features - RTF)</i>	88
4.1.12	<i>Grau de atendimento aos requisitos funcionais</i>	89
4.1.13	<i>Índice de defeitos encontrados na fase de testes</i>	90
4.1.14	<i>Índice de histórias por iteração</i>	91
4.1.15	<i>Lead time (tempo de entrega das funcionalidades ao cliente)</i>	92
4.1.16	<i>Média da motivação dos membros da equipe</i>	94
4.1.17	<i>Nível de satisfação do cliente</i>	96
4.1.18	<i>Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões)</i>	97
4.1.19	<i>Número de tarefas não concluídas na iteração</i>	98
4.1.20	<i>Severidade média dos defeitos</i>	99
4.1.21	<i>Taxa de acerto na estimativa das tarefas</i>	101
4.1.22	<i>Taxa de dívida técnica (problemas técnicos a serem resolvidos)</i>	102
4.1.23	<i>Taxa de retrabalho</i>	103
4.1.24	<i>Tempo investido em mudanças</i>	104
4.1.25	<i>Tempo investido em tarefas não planejadas</i>	106
4.1.26	<i>Tempo médio de resolução dos impedimentos</i>	107
4.1.27	<i>Throughput (número de tarefas entregues por período de tempo)</i>	108
4.1.28	<i>Total de horas trabalhadas no projeto</i>	110
4.1.29	<i>Variação do número de defeitos entre iterações</i>	111
4.1.30	<i>Variação dos custos (entre estimado e realizado)</i>	112
4.1.31	<i>Velocidade média diária da equipe</i>	114
4.1.32	<i>Work in Progress (WIP)</i>	116
4.2	ANÁLISES REALIZADAS A PARTIR DO CONJUNTO DE MÉTRICAS	117
4.2.1	<i>Mapa mental</i>	117
4.2.2	<i>Agrupamento das métricas por fases</i>	120
4.3	PROPOSTA DE UM PROCESSO DE MEDIÇÃO PARA PROJETOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE	121
4.3.1	<i>Quadro visual de monitoramento</i>	123
4.3.2	<i>Fase 1 – Planejamento de medição e divulgação dos resultados</i>	125
4.3.3	<i>Fase 2 – Coleta de dados, monitoramento da iteração e divulgação dos resultados</i>	127
4.3.4	<i>Fase 3 – Ações da iteração e divulgação dos resultados</i>	128
4.3.5	<i>Fase 4 – Monitoramento do projeto/releases e divulgação dos resultados</i>	128
4.3.6	<i>Fase 5 – Ações sobre o projeto/releases e divulgação dos resultados</i>	129

4.3.7	<i>Fase 6 – Avaliação final e divulgação dos resultados</i>	129
4.4	AVALIAÇÃO DO PROCESSO PROPOSTO	130
4.5	CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO 4.....	135
4.5.1	<i>Análise do conjunto de métricas</i>	135
4.5.2	<i>Análise do processo de medição</i>	139
5	CONSIDERAÇÕES FINAIS	141
	REFERÊNCIAS	144
	APÊNDICES	149

1 INTRODUÇÃO

O Brasil possui uma posição estratégica no mercado mundial de software, os dados da Associação Brasileira das Empresas de Software (ABES, 2011) revela que o Brasil ocupa a 11^a no ranking internacional de mercados de software e serviços. Em 2011 o mercado de software no Brasil movimentou US\$ 21,4 bilhões, aumento de 12,4% em relação ao ano de 2010, quando comercializou US\$ 19,04 bilhões, o equivalente a 1% do PIB brasileiro naquele ano (ABES, 2011). O crescente mercado de software está relacionado com o significativo papel que a tecnologia de informação desempenha em todas as áreas da economia moderna, onde as exigências são exponencialmente maiores sobre a diversidade e qualidade do software a ser produzido. Neste contexto, o *time-to-market*¹ no lançamento de um produto software pode significar a diferença entre o sucesso e o fracasso da empresa (MOE, 2011).

Neste ambiente competitivo e acirrado em que se encontra o mercado de software, onde é necessário entregar produtos de alta qualidade, num curto espaço de tempo, em consonância com as necessidades dos clientes e mantendo um ambiente colaborativo, surgem os métodos ágeis como uma alternativa ao modelo cascata, também denominado de processo tradicional de desenvolvimento de software. Os métodos ágeis possuem uma abordagem iterativa e incremental de desenvolvimento, com equipes auto-organizadas, com constante colaboração do cliente que se ajusta dinamicamente às suas necessidades através da aceitação de mudanças de requisitos em qualquer momento do projeto, que busca a minimização de produtos de trabalho de engenharia de software e a simplicidade no desenvolvimento (ABBA, GRAVELL e WILLS, 2010; DYBA e DINGSOYR, 2008; NERUR, 2007).

Os métodos ágeis surgiram na década de 1990 (DYBA, 2000), mas foi com o advento do Manifesto Ágil em 2001, que ganhou força e trouxe mudanças sem precedentes para o campo da engenharia de software. Dingsoyr et al. (2012) avaliaram os primeiros 10 anos dos métodos ágeis e afirmam que esta foi uma das décadas que mais testemunhou a introdução de métodos de software, ferramentas, técnicas e melhores práticas. Porém, como em qualquer disciplina nascente, os primeiros anos do desenvolvimento ágil foram marcados pela exuberância de alguns e pelo ceticismo de outros. Surgiram uma série de métodos que aderem em diferentes graus aos princípios do manifesto (DINGSOYR et al., 2012). Entre os métodos

¹ *Time-to-market*: Tempo entre o projeto, concepção de um produto e sua disponibilização para a venda (COHEN et al., 1996).

criados, os mais utilizados são Scrum, *eXtreme Programming* (XP) e *Lean Software Development* (VERSIONONE, 2014). De um modo geral, todos estes métodos foram criados visando atender aos princípios fundamentais do manifesto. Embora este crescimento sem precedentes tenha sido prontamente aceito por muitos profissionais, muito trabalho ainda tem de ser empreendido para tornar coerente o discurso atual sobre a agilidade (DINGSOYR et al., 2012).

O *Standish Group* publicou em 2011 o relatório *Chaos Manifesto* que compara o desempenho entre projetos ágeis e tradicionais de software e apresenta fatores motivadores para adoção de métodos ágeis. Nos projetos tradicionais (modelo cascata) 14% tiveram sucesso, 57% foram contestados e 29% falharam, sendo que nos projetos ágeis 42% tiveram sucesso, 49% foram contestados e 9% falharam (STANDISH GROUP, 2012). Os dados apresentados na pesquisa mostram que os projetos realizados a partir da abordagem ágil tiveram mais sucesso que os tradicionais, porém a taxa de projetos que falharam ou foram contestados ainda é alta, o que demonstra sérios problemas de gestão e reforça a importância de um monitoramento e controle eficazes nos projetos desta natureza.

1.1 JUSTIFICATIVA

Mikulenias e Butleris (2010) afirmam que as pesquisas da área se concentram na apresentação de histórias de sucesso ou de lições aprendidas por organizações que adotaram métodos ágeis para projetos específicos, havendo uma falta de pesquisas de avaliação de adequação dos métodos ágeis, considerando diversas características ambientais em empresas de software, entre elas adoção de métricas de software. As métricas são essenciais para as empresas desenvolvedoras de software, pois ajudam na medição da qualidade, na estimativa dos recursos necessários e dos custos, no planejamento e controle do progresso de desenvolvimento de software (MISHRA, KUMAR e KUMAR, 2009).

Hartmann e Dymond (2006) afirmam que o aumento do uso dos métodos ágeis traz com ele os modos de avaliação incompatíveis com os valores e princípios ágeis. Poonacha e Bhattacharya (2012) argumentam que muitas organizações adotam métodos ágeis sem entender quais os fatores devem ser medidos e controlados.

Sob o ponto de vista prático existem comunidades muito ativas que tratam de questões ligadas aos métodos ágeis. Elas discutem e difundem os métodos ágeis e suas práticas,

motivando novas pessoas e empresas a adotá-las. Sob o ponto de vista acadêmico, existem poucas publicações específicas que tratam sobre a utilização de métricas e sobre adoção de processo de medição para projetos de softwares que utilizam métodos ágeis. Os próprios métodos ágeis não tratam claramente sobre o assunto, o que leva a uma falta de discernimento sobre o tema medição e métricas. Os profissionais frequentemente recorrem a relatos de experiência de outros profissionais e seguem muitas vezes realizando o processo de forma empírica.

1.2 OBJETIVOS E QUESTÕES DE PESQUISA

Partindo-se da premissa de que é possível desenvolver um processo de medição compatível com a abordagem ágil de desenvolvimento de software e, que a apresentação detalhada de um conjunto de métricas e suas representações irão auxiliar na gestão desses projetos, apresenta-se a seguinte questão de pesquisa:

Como contribuir com o esforço de definição das métricas para o monitoramento de projetos em empresas de desenvolvimento de software que utilizam métodos ágeis e com a proposição de um processo de medição compatível com esta abordagem?

O objetivo geral deste trabalho é definir um conjunto de métricas adequadas às necessidades de monitoramento e propor um processo de medição, compatível com a abordagem ágil de desenvolvimento de software, visando apoiar as empresas de software no monitoramento ao longo do projeto.

Para atingir estes resultados os seguintes objetivos específicos foram propostos:

- Identificar as métricas mais usadas e citadas tanto em empresas quanto na literatura, derivando os objetivos de medição atendidos por estas métricas;
- Definir diretrizes para definição de um processo de medição respeitando os valores e princípios ágeis;
- Propor uma forma de análise e monitoramento contínuo das métricas utilizadas na gestão do projeto de desenvolvimento de software que utilizam métodos ágeis respeitando as características que diferenciam esses projetos.

Após a definição dos objetivos foi desdobrada a questão principal de pesquisa em três questões secundárias descritas a seguir:

- QS1.** Quais são as métricas que atendem às necessidades de medição inerentes à natureza dos projetos ágeis de desenvolvimento de software, que são citadas pela literatura?
- QS2.** Quais são as métricas que atendem as necessidades de medição inerentes à natureza dos projetos ágeis de desenvolvimento de software, que são utilizadas pela empresa que os adotam?
- QS3.** Que elementos devem ser considerados para definir um processo de medição que atenda às necessidades de medição dos envolvidos em projetos de desenvolvimento de software que adotam métodos ágeis?

Os pesquisadores não têm a pretensão esgotar o estudo sobre o assunto, porém contribuir com o esclarecimento e a organização tanto de métricas quanto de processo de medição compatível com a abordagem ágil de desenvolvimento de software.

1.3 MÉTODO DE TRABALHO

Esta sessão descreve a estrutura e método de trabalho utilizado para alcançar os objetivos propostos na tese. Para responder cada questão secundária foi utilizado um procedimento metodológico específico. A Figura 1 apresenta a estrutura geral da pesquisa, a qual contém as fases da pesquisa.

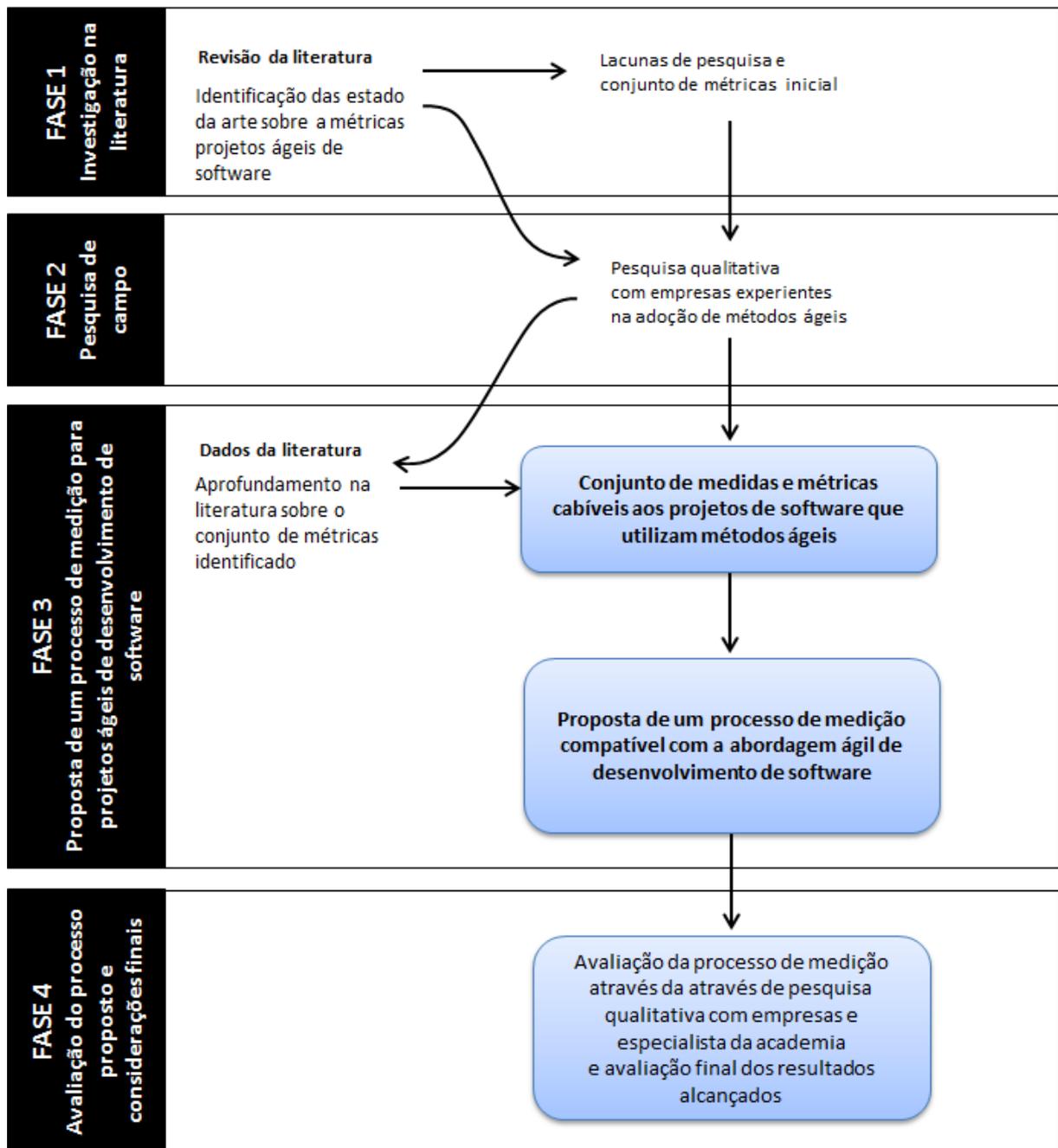


Figura 1. Etapas de pesquisa utilizadas na tese

Fonte: Elaborado pelo autor.

Nas próximas sessões são apresentadas cada umas das fases de pesquisas, descendo os objetivos específicos e os procedimentos metodológicos utilizados.

1.3.1 Fase 1 - Investigação na literatura

Nesta fase da pesquisa foi realizado o levantamento bibliográfico visando esclarecer o contexto sobre o desenvolvimento de software, as motivações que levam as empresas a adoram métodos ágeis, os fundamentos sobre métricas e processo de medição no desenvolvimento de software e sua relação com os métodos ágeis.

Nesta etapa da pesquisa também se buscou responder a questão de pesquisa QS1, para isso foi realizada uma busca sistematizada na literatura, a qual teve como objetivos: (i) investigar estudos que tratem especificamente do tema utilização de métricas em projetos de desenvolvimento de software que utilizem métodos ágeis; (ii) identificar métricas que permitam as empresas desenvolvedoras de software que utilizem métodos ágeis a controlar os seus projetos nas fases de release, iteração e diário. A partir das questões de pesquisa foi definido o protocolo de pesquisa que contém as bases eletrônicas a serem realizadas as pesquisas, identificadas às palavras-chaves, definidas as expressões de busca, restrições de ano de publicação e idioma. Também fez parte do protocolo de pesquisa as etapas a serem realizadas, o qual foi campos por 3 etapas: (i) pesquisa nas bases eletrônica e catalogação dos artigos; (ii) filtro através da leitura dos títulos e resumos; e (iii) aplicação dos critérios de exclusão.

Os resultados obtidos nesta fase da pesquisa são apresentados no capítulo 2.

1.3.2 Fase 2 - Pesquisa de campo

A opção metodológica adotada para responder a questão secundária de pesquisa QS2 foi entrevista qualitativa em profundidade, de caráter exploratório. As pesquisas exploratórias são orientadas para a descoberta, sem a intenção de testar ou validar hipóteses pré-estabelecidas (HAIR JR. et al., 2005; GRAY, 2012). Como há poucas informações disponíveis sobre a utilização das métricas em projetos de software que utilizam métodos ágeis e como é difícil identificar respondentes elegíveis, este trabalho apresenta características de uma pesquisa exploratória com a adoção de métodos de amostragem não probabilísticos e utilização de um critério de elegibilidade para selecionar as respostas válidas a partir das informações coletadas (HAIR JR. et al., 2005; MALHOTRA, 2012). Este estudo também se

caracteriza por um corte transversal, tendo como unidade de análise as organizações ou unidades de negócio de atuação dos respondentes (HAIR JR. et al., 2005).

O critério de elegibilidade da amostra deste estudo são os profissionais responsáveis pela tomada de decisão nas atividades relacionadas com a avaliação de desempenho dos projetos. Também se considerou como critério de elegibilidade a senioridade e o cargo dos respondentes, de tal forma que apenas os respondentes com cargos gerenciais foram considerados na amostra. Devido à dificuldade em ter acesso a tais indivíduos nas organizações, para a amostra inicial foi utilizado o método de amostragem por julgamento, expandida pelo método de amostragem que visa ampliar as unidades de amostra incentivando os respondentes a indicarem outros respondentes responsáveis pelo gerenciamento do processo de medição (HAIR Jr. et al., 2005). Para Hair Jr. et al. (2005), o tamanho da amostra para utilização de métodos não probabilísticos pode ser definido com base no julgamento ou percepção do pesquisador.

Os objetivos da entrevista em profundidade são:

- a) Identificar o contexto das empresas de software que utilizam métodos ágeis quanto a tempo de adoção dos métodos ágeis, métodos ágeis utilizados e tempo de experiência dos entrevistados;
- b) Identificar quais são as métricas utilizadas pelas empresas de software para controlar os projetos que utilizam métodos ágeis;
- c) Identificar quais são as fase típicas de um projeto ágil e o que é realizado em cada fase;
- d) Identificar quais são as funções envolvidas em cada fase do projeto e quais as suas responsabilidades nos projetos.

Para o planejamento das entrevistas foram utilizadas as etapas propostas por Ribeiro e Milan (2007) e Gray (2012), conforme descrito a seguir:

- a) **determinação da fonte dos dados e amostragem:** a amostra foi não probabilística e selecionou-se empresas que adotavam métodos ágeis a mais de 2 anos. A fonte de dados foi primária. As entrevistas foram realizadas com profissionais com mais de 3 anos de experiência na utilização de métodos ágeis e que atuassem em cargos de gestão (diretores, *scrum master*, líderes de equipe ou gerentes de projetos) totalizando 9 entrevistados. Para identificar vários contextos e necessidades buscou-se empresas

de diferentes portes e que desenvolvem softwares de diferentes características. Na literatura não existe um critério único para classificação do porte das empresas podendo variar tanto em número de funcionários quanto faturamento. Aqui nesta pesquisa foi utilizada a classificação proposta pelo SEBRAE que declara para as atividades de comércio e serviços, que uma micro empresa é aquela que possui até 9 pessoas, a pequena empresa de 10 a 49 pessoas, a média empresa de 50 a 99 pessoas, e a grande acima de 100 pessoas (SEBRAE, 2012). Esta escolha se deu em função da falta de acesso ao faturamento das empresas. No total foram estudadas 9 empresas sediadas no sul do Brasil (estados do Paraná, Santa Catarina e Rio Grande do Sul), sendo 3 de porte pequeno, 2 de porte médio e 4 de porte grande. Dessas, 8 possuem como atividade fim o desenvolvimento de software, e 1 empresa (Empresa D de porte grande) não tem o desenvolvimento de software como atividade fim, porém possui um setor de desenvolvimento de software interno composto por 35 pessoas, que é responsável pelo desenvolvimento dos softwares que atendam às suas necessidades organizacionais. Ela foi categorizada nesta pesquisa como empresa média, pelo porte da equipe de desenvolvimento. Os entrevistados das empresas C, E, F e I atuam também como *'agile coach'*². Pelo tempo de experiência e pela atuação dos entrevistados é possível observar que a maioria são especialistas em métodos ágeis e possuem uma visão da aplicação destes métodos não somente na empresa em que atuam mas também em outras.

- b) **definição da técnica de coleta de dados:** a técnica de coleta de dados foi a entrevista individual em profundidade, semiestruturadas, guiada por roteiro básico de questões, o qual apresenta a seguinte estrutura: (i) questões iniciais onde são levantadas informações sobre o contexto das organizações e perfil dos entrevistados, (ii) questões centrais onde são levantadas informações sobre as métricas utilizadas, fases do projeto e funções envolvidas nos projetos de software que utilizam métodos ágeis, e (iii) questões de fechamento onde se buscou identificar informações adicionais dos entrevistados. Foi realizado um pré-teste do roteiro junto a um especialista da área a fim de validar o mesmo. O roteiro de coleta de dados é apresentado na íntegra no Apêndice 1;

² *Agile Coach*, termo utilizado para denominar profissionais que atuam na área de desenvolvimento de software e devido ao seu conhecimento aprofundado sobre a prática dos métodos ágeis nas organizações são considerados especialistas sobre o assunto, e realizam palestras, treinamentos ou contribuem em *blogs* disseminando a cultura ágil entre profissionais da área de software. Também denominados de agilistas (<http://www.infoq.com/br/>).

- c) **coleta dos dados:** as entrevistas foram conduzidas de forma individual, gravadas e transcritas. Foram realizadas no período de agosto a outubro de 2013 e tiveram duração de 1 hora em média. Nas empresas D, E e I as entrevistas foram realizadas via Skype e as demais foram realizadas presencialmente;
- d) **análise e interpretação dos resultados:** o conteúdo das entrevistas foi transcrito e analisado através de comparação interna entre os respondentes; ordenamento por importância e comparação externa com a literatura. O objetivo desta análise foi o de construir um panorama do uso das métricas em empresas e, ao mesmo tempo, identificar aquelas mais utilizadas ou citadas pelos entrevistados.

Os resultados da pesquisa qualitativa estão descritos no capítulo 4. A seguir é apresentado o método de pesquisa da fase 3.

1.3.3 Fase 3 - Proposição do processo de medição

Para responder a questão secundária QS3 a estratégia de pesquisa adotada se subdividiu em 2 etapas. A primeira etapa consistiu em consolidar, a partir da pesquisa de campo e da revisão da literatura, um conjunto de métricas para projetos de software que utilizam métodos ágeis e teve como objetivos:

- a) definir um conjunto de características que ajudaram a descrever cada uma das métricas;
- b) compreender quais são os objetivos de medição que estão relacionados com cada métrica;
- c) elaborar um mapa mental procurando organizar essas métricas através de categorias de avaliação, tipos de indicador (tendência ou resultado) e dos seus objetivos de medição;
- d) agrupar as métricas por fase do projeto ágil em que serão analisadas.

A segunda etapa concentrou-se na proposição de um processo de medição compatível com as características dos projetos ágeis e um conjunto de procedimentos que orientam na sua utilização. Foi desenvolvido com base no ciclo lógico de gestão e das recomendações sobre processo de medição do modelo de maturidade MPS.BR (SOFTEX, 2011). Os seguintes passos foram realizados nesta etapa: (i) definição dos requisitos para construção do processo

de medição; (ii) definição das fases do processo de medição que permitam a uma equipe ágil planejar, acompanhar a execução, monitorar problemas e ações corretivas e fazer a avaliação do processo de medição de um projeto ágil de desenvolvimento de software; e (iii) proposição de um quadro visual de monitoramento que permita a gestão do processo de medição de forma visual.

1.3.4 Fase 4 – Avaliação do processo de medição proposto

A última etapa da pesquisa tratou da avaliação do conjunto de métricas e do processo de medição proposto. Este foi submetido à avaliação por profissionais que atuam em empresas de desenvolvimento de software que utilizam métodos ágeis e especialistas da academia sobre o assunto, sendo a amostra não probabilística e definida por conveniência. Esta etapa teve como objetivo avaliar o processo de medição proposto frente aos critérios definidos para sua construção e aos benefícios percebidos pelos avaliadores. A avaliação foi realizada através de entrevistas semi-estruturadas seguindo o roteiro e questionamentos apresentado no Apêndice 8, o qual é composto dos seguintes passos: (i) perguntas iniciais; (ii) apresentação do conjunto de métricas, suas representações e do processo de medição proposto; (iii) Avaliação do processo proposto frente aos requisitos previamente definidos; (iv) perguntas finais; (v) agradecimentos. As entrevistas foram realizadas através de visita presenciais e à distância, via Skype, respeitando a disponibilidade dos entrevistados. O resultado desta fase da pesquisa está apresentado no capítulo 4.

1.4 DELIMITAÇÕES E LIMITAÇÕES DO TRABALHO

Esta pesquisa não tem como pretensão esgotar o conjunto de métricas úteis para controle o dos projetos ágeis, mas sim trazer ordem sobre o assunto. A intensão é investigar as métricas no que tange ao controle tático e operacional de projeto ágeis de desenvolvimento de software nas fases de *release*, iteração e diário. Desta forma foram consideradas estando fora do escopo deste trabalho a definição de métricas em nível de código (testes, complexidade do código ou de seus componentes, etc) ou definição de métricas ou indicadores de desempenho para decisão em nível estratégico.

Em relação à pesquisa de campo realizada, destacam-se as seguintes limitações: (a) os dados sobre as métricas foram coletados através de entrevista e o pesquisador não teve acesso aos documentos internos das empresas; (b) apesar de ter sido selecionado como perfil do entrevistado pessoas de nível tático (gerentes de projetos, diretores, etc.), alguns declararam não ter acesso às todas as informações sobre o projeto, principalmente as informações estratégicas; (c) a amostragem de 9 empresas de portes diferentes e maduras na utilização dos métodos ágeis, não assegura que a relação de métricas identificadas seja extensiva, podendo haver outras métricas que não identificadas nestas entrevistas; no entanto, observou-se uma repetição de dados no conjunto de entrevistados, o que sugere o esgotamento do tema; (d) não foi possível com o tamanho de amostra analisar em profundidade as diferenças entre os comportamentos de empresas certificadas e não certificadas, com respeito a avaliação de desempenho e métricas, mas alguns indicativos foram percebidos e discutidos no trabalho.

1.5 ESTRUTURA DO TRABALHO

A tese está estruturada em 5 capítulos. No primeiro capítulo foi apresentado o problema e motivação do estudo, questões e objetivos de pesquisa, justificativa, o método de pesquisa utilizado e as limitações da pesquisa. O capítulo 2 contemplou a revisão bibliográfica abordando os temas relacionados ao processo de desenvolvimento de software, desenvolvimento ágil de software, medição de software e utilização de métricas em projetos ágeis de software. No capítulo 3 foram descritos os dados obtidos na pesquisa de campo realizada com empresas desenvolvedoras de software experientes na adoção de métodos ágeis. No capítulo 4 foi apresentado um conjunto de métricas e suas representações para projetos ágeis de software que auxiliam na gestão de projetos, a proposta de processo de medição compatível com a abordagem ágil de desenvolvimento de software e a avaliação do processo proposto. No capítulo 5 foram apresentadas as considerações finais e sugestões de trabalhos futuros.

2 REFERENCIAL TEÓRICO

A área de estudo responsável pelas pesquisas sobre desenvolvimento de software é a engenharia de software, a qual é responsável por definir métodos, técnicas, processos e ferramentas para desenvolvimento de sistemas com qualidade (PRESSMANN, 2006). O processo de desenvolvimento de software mais conhecido e antigo é o processo linear e sequencial de desenvolvimento conhecido como modelo cascata ou modelo *waterfall*, também chamado de modelo tradicional de desenvolvimento de software. Ele organiza os projetos de software em quatro grandes fases realizadas sequencialmente: análise, projeto, codificação e testes, sendo que uma fase só inicia quando a anterior estiver completa. Ainda hoje, este é o modelo de desenvolvimento mais conhecido e amplamente utilizado (PRESSMAN, 2006). Modelos de desenvolvimento tradicionais são centrados no processo, no qual se acredita que sistemas são completamente especificáveis, previsíveis e podem ser construídos através de um planejamento meticuloso e extensivo. Este processo gera uma grande quantidade de documentação (NERUR, 2007).

Os métodos de desenvolvimento ágil de software surgiram como uma alternativa ao modelo cascata seguindo o modelo de processo de desenvolvimento de software iterativo e incremental, promovendo a auto-organização das equipes, buscando minimizar a documentação de engenharia de software, mantendo a constante colaboração do cliente e ajustando-se dinamicamente às suas necessidades através da aceitação de mudanças de requisitos em qualquer momento do projeto (ABBA, GRAVELL e WILLS 2008; DYBA e DINGSOYR, 2008; NERUR, 2007). Embora o modelo de desenvolvimento utilizado, os princípios individuais e as práticas do desenvolvimento proposto pelos métodos ágeis não sejam inteiramente novos para a comunidade de software (LARMAN e BASILI, 2003; ABBA, GRAVELL e WILLS 2008) a forma como foram reunidos num coerente ‘quadro teórico e prático’ foi certamente novo (WILLIAMS E COCKBURN, 2003). A sessão a seguir apresenta o que são os métodos ágeis, seus valores e princípios e contextualização da sua utilização e evolução.

2.1 MÉTODOS ÁGEIS, VALORES, PRINCÍPIOS E CONTEXTUALIZAÇÃO

O modelo cascata foi utilizado durante muitos anos como padrão no desenvolvimento de software, porém, percebendo o grande número de casos fracassos (STANDISH GROUP, 2012), alguns líderes experientes de desenvolvimento adotaram modos de trabalho que se opunham aos principais conceitos dos métodos tradicionais. Aos poucos, foram percebendo que suas formas de trabalho, apesar de não seguirem os paradigmas no mercado, eram eficientes. Aplicando-as em vários projetos, elas foram aprimoradas, e em alguns casos chegaram a se transformar em novas metodologias de desenvolvimento de software. Essas metodologias passaram a ser chamadas de leves por não utilizarem as formalidades que caracterizavam os processos tradicionais e por evitarem a burocracia imposta pela utilização excessiva de documentos. Com o tempo, algumas delas ganharam destaque nos ambientes empresarial e acadêmico, gerando grandes discussões, principalmente relacionadas à confiabilidade dos processos e à qualidade do software (PRESSMAN, 2011).

Em 2001 dezessete desses líderes que trabalhavam as chamadas abordagens leves, se reuniram para discutir sobre suas formas de trabalho e para chegar a uma nova metodologia de produção de software que pudesse ser usada por todos eles e em outras empresas, substituindo os modelos tradicionais de desenvolvimento (AGILE MANIFESTO, 2001). Após dois dias de debate, o grupo não chegou a uma metodologia: concluíram que desenvolver software é algo complexo demais para ser definido por um único processo. No entanto, o grupo chegou ao consenso de valores e princípios determinantes para a obtenção de bons resultados. Os quatro valores que o Manifesto Ágil definiu são (AGILE MANIFESTO, 2001):

- indivíduos e interação entre eles mais que processos e ferramentas;
- software em funcionamento mais que documentação abrangente;
- colaboração com o cliente mais que negociação de contratos;
- responder a mudanças mais que seguir um plano.

Ainda que haja valor nos termos à direita, devem ser valorizados mais os termos a esquerda nestas descrições. Conforme mencionado anteriormente, também foram definidos princípios que fundamentam o Manifesto Ágil que são (AGILE MANIFESTO, 2001):

1. A prioridade mais alta é satisfazer o cliente através de contínuas entregas de software de valor e mais cedo.
2. Mudanças de requisitos são bem vindas, mesmo que tardias no processo de desenvolvimento. Processos ágeis aproveitam as mudanças para obter vantagens competitivas para o cliente.
3. Entregas de software funcionando devem ser frequentes, de duas semanas a dois meses, com preferência para intervalos de tempo mais curtos.
4. O pessoal de negócios e os desenvolvedores devem trabalhar juntos diariamente ao longo do projeto.
5. Projetos devem ser construídos em torno de indivíduos motivados. Deve ser dado a eles o ambiente adequado e apoiadas as suas necessidades, além de confiar neles para realizar o trabalho.
6. O meio mais eficiente e eficaz de transmitir informação para dentro da equipe de desenvolvimento é a conversa face a face.
7. Software funcionando é a principal medida de progresso.
8. Processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. Atenção contínua à excelência técnica e ao bom projeto aumentam a agilidade.
10. A simplicidade, como arte de maximizar a quantidade de trabalho não executado é essencial.
11. As melhores arquiteturas, requisitos e projetos surgem das equipes auto-organizadas.
12. Em intervalos regulares, as equipes refletem em como se tornar mais eficientes, e sintonizam e ajustam seu comportamento adequadamente.

Segundo Williams e Cockburn (2003), os princípios não são uma definição formal de agilidade, mas são diretrizes para a entrega de software de alta qualidade de maneira ágil.

O advento do Manifesto ágil chamou a atenção da comunidade científica e da indústria de software sobre a sua utilização, e surgiu uma série de métodos aderindo aos princípios do manifesto, tais como: Scrum, XP (eXtreme Programming), *Lean Development Software*, Kanban, entre outros. De um modo geral, todos estes métodos se esforçaram para atender os valores fundamentais do manifesto (DINGSOYR et al., 2012):

- ênfase no desenvolvimento colaborativo onde as pessoas receberam privilégios sobre os processos que restringiam o seu trabalho;
- eliminação do trabalho desnecessário, especialmente no que diz respeito a criação de documentos desnecessário restringindo-se a documentar o que extremamente necessário e nada mais;
- os clientes e *stakeholders* deixaram de ser apenas um elemento isolado às margens do desenvolvimento de software, mas tornou-se papel fundamental que guia a evolução do produto a ser desenvolvido;
- houve aceitação do fato da incerteza fazer parte do processo de desenvolvimento de software e que há a necessidade de se controlar essas tendência de mudanças ao longo do projeto.

As diferenças entre métodos tradicionais e ágeis resultam em diversas práticas específicas de cada uma de como ocorre o planejamento e controle, o papel assumido entre os desenvolvedores, o papel dos clientes e o modo de condução do projeto. Eder et al. (2014) apresenta várias diferenças entre as abordagens tradicional e ágil de gerenciamento de projetos, as quais são apresentadas no Quadro 1.

Quadro 1. Principais diferenças entre o modelo tradicional (cascata) e o modelo ágil

	Abordagem de gerenciamento de projetos tradicional	Abordagem de gerenciamento ágil de projetos
1) A forma de elaboração do plano do projeto	Há um único plano de projeto, que abrange o tempo total do projeto e contém os produtos, entregas, pacotes de trabalho e atividades.	Há dois planos de projeto: a) um plano geral que considera o tempo total de duração do projeto, mas que contém apenas os produtos principais do projeto; b) um plano de curto prazo (iteração) que contém apenas as entregas e atividades referentes a uma fração de tempo do projeto.
2) A forma como se descreve o escopo do projeto	Descrição exata do resultado final por meio de texto, com normas do tipo contratuais, números objetivos e indicadores de desempenho.	Descrição do resultado final de maneira abrangente, desafiadora, ambígua e metafórica.

3) O nível de detalhe e padronização com que cada atividade do projeto é definida	As atividades são descritas de maneira padronizada e organizadas em listas do tipo WBS. Contêm códigos e são classificadas em conjuntos de pacotes de trabalho, entregas e produtos do projeto.	Não há um padrão para a descrição das atividades, que podem ser escritas na forma de estórias, problemas, ações ou entregas. E não há uma tentativa de organização, apenas a priorização do que deve ser executado no momento.
4) O horizonte de planejamento das atividades da equipe de projeto	As listas de atividades são válidas para o horizonte total do projeto.	As listas de atividades são válidas para uma iteração, que é definida como uma fração do tempo total do projeto.
5) A estratégia utilizada para o controle do tempo do projeto	Empregam-se relatórios com indicadores de desempenho, documentos escritos, auditorias e análises de transições de fase. As reuniões da equipe não são frequentes.	Empregam-se dispositivos visuais que indicam entregas físicas do resultado final (cartazes, autoadesivos etc.). As reuniões são curtas e frequentes.
6) A estratégia utilizada para a garantia do atingimento do escopo do projeto	O gerente de projeto avalia, prioriza, adiciona ou altera as atividades do projeto para que os resultados estejam em conformidade com o escopo do projeto assinado com o cliente.	O cliente avalia, prioriza, adiciona ou altera o produto final do projeto, conforme a experiência com os resultados alcançados. A equipe altera as atividades para obter os resultados propostos pelo cliente.

Fonte: Eder et al. (2014) .

A seguir é feita uma apresentação sucinta dos métodos ágeis Scrum, XP (*eXtreme Programming*), Lean/Kanban que são os métodos mais utilizados atualmente. Segundo pesquisa da VERSIONONE (2014) realizada com 3.501 participantes, observou-se que 55% utilizam Scrum, 11% Scrum e XP híbrido, 10% não se declaram usuários de um método específico, mas de um conjunto híbrido de práticas ágeis; 5% utilizam Kanban e 3% utilizam *Lean*. Nem todos os autores que tratam dos métodos ágeis apresentam de forma explícita como aplica-los, porém recomendam práticas a serem seguidas.

O Scrum³ foi desenvolvido por Ken Schwaber e Jeff Sutherland e em seu guia de implementação o descrevem como um framework para gerenciar o desenvolvimento de produtos complexos no qual emprega vários processos e técnicas. Consiste na definição de conjunto papéis, eventos, artefatos e as regras a serem seguidas. As equipes Scrum são compostas pelo *Product Owner*, ou dono do produto, responsável pela definição, priorização e aceite do produto, entre outras funções; o time de desenvolvimento responsável pelo

³ O guia com a descrição completa do Scrum desenvolvido pelos autores Ken Schwaber e Jeff Sutherland está disponível no site <http://www.scrum.org>

desenvolvimento propriamente dito do produto, entre outras funções; e o Scrum Master que atua como um líder de equipe e é responsável por garantir que o Scrum seja entendido e aplicado, entre outras funções. O processo de desenvolvimento ocorre de forma iterativa e incremental, sendo que em cada ciclo ocorre uma reunião de planejamento, a Sprint, revisão de Sprint e retrospectiva da Sprint (SCHWABER e SUTHERLAND, 2013). O Scrum não define técnicas específicas para projeto ou desenvolvimento do software, mas práticas de gerenciamento para as fases do seu processo.

O método XP (*eXtreme Programming*) foi desenvolvido por Kent Beck e propõe um conjunto de valores, princípios e práticas. Diferente do Scrum, enfatiza exclusivamente a atividade de desenvolvimento e visa o desenvolvimento rápido, com qualidade e satisfação do cliente, mesmo em situação de requisitos vagos ou em constante mudança. Possui como valores: simplicidade, comunicação, coragem, *feedback* constante e respeito, os quais influenciam na cultura e comportamento dos membros da equipe. Entre as práticas pode-se citar: programação em pares, simplicidade, desenvolvimento guiado a testes, integração contínua, cliente presente, entre outras (BECK e ANDRES, 2004).

Poppendieck e Poppendieck (2003) apresentam *Lean Software Development* como sete princípios *lean* (eliminar o desperdício, amplificar o aprendizado, adiar decisões, entregar o mais rápido possível, tornar a equipe responsável, construir com integridade e visualizar o todo), onde buscam traduzir esses princípios em práticas ágeis a serem seguidas no desenvolvimento de software.

Uma prática comum entre as equipes ágeis é a utilização de um quadro de tarefas kanban para visualizar as tarefas ao longo de processo de desenvolvimento do software. A sua adoção permitiu a utilização de um sistema puxado de produção de software com base na cultura da melhoria contínua, surgindo assim à adoção de novos conceitos *lean* para o desenvolvimento ágil de software, tais como, monitoramento do tempo de ciclo de desenvolvimento (*cycle time*), definir limite de trabalho por fase do processo (limitar o WIP – *Work in progress*), conhecer a capacidade de produção por período de tempo (*throughput*) e o conhecer o tempo de ciclo de entrega das funcionalidades desenvolvidas ao cliente (*lead time*), não mais realizando ciclos baseados em iterações, mas implantando um processo de fluxo contínuo de entregas ao cliente (BOEG, 2012). A descrição detalhada das métricas *cycle time*, *lead time*, *throughput* e *work in progress* é apresentada no capítulo 4 sessão 4.1.

2.1.1 Fases de um projeto ágil

Segundo Cohn (2005) o planejamento de um projeto ágil de software deve seguir um horizonte de visão que permita criar e revisar as metas do projeto. Também afirmam que não é possível planejar além deste horizonte devido a falta de visão a longo prazo do projeto, por isso é necessário haver uma elaboração progressiva do plano do projeto que está sendo executado. Um plano não garante um conjunto exato de funcionalidades a serem entregues numa data exata e num custo especificado. Um plano, no entanto, comunica e estabelece um conjunto de expectativas iniciais. Um bom plano é aquele em que as partes interessadas encontram-se suficientemente confiáveis e podem usá-lo como base para a tomada de decisões. No início de um projeto, isso pode significar que o plano informe que o produto pode ser lançado no terceiro trimestre, já no segundo momento, poderá conter um conjunto de características. Conforme ocorre o andamento do projeto o plano deverá ser mais preciso e fornecer informações específicas para tomada de decisão.

Ainda segundo Cohn (2005) o planejamento ágil consiste em:

- se concentrar mais no planejamento (atividade de planejar) do que no plano (documento);
- encorajar a mudança;
- resultar em planos que são facilmente alterados.

Segundo Cohn (2005) o desenvolvimento de projetos ágeis de software consiste em vários níveis de planejamento do projeto incluindo estratégia, portfólio, produtos, *releases*⁴, iterações e diário, o qual denominou de ‘planejamento cebola’, conforme apresentado na Figura 2. O planejamento da estratégia, portfólio e produto envolvem objetivos organizacionais e planejamento de longo prazo e deverão fornecer uma visão para o gerenciamento do desenvolvimento do produto, isto é, uma visão geral do produto para ser desenvolvido nas futuras *releases*. A maioria das equipes estão envolvidas nas fases de planejamento de *release*, iteração e diário, pois o planejamento do produto, portfólio e estratégico são de responsabilidade do proprietário do produto e envolvem questões estratégicas. De acordo com Cohn (2005) planejamentos longos são muito abstratos por isso recomenda o planejar em curtos períodos em releases, iterações e diário, recomendando a

⁴ *Release*: uma parte de um produto de software que é o resultado da implementação em vários ciclos de iteração com entregas incrementais (SLIGER e BRODERICK, 2008).

evolução progressiva do plano do projeto. O planejamento da *release* considera as histórias de usuários⁵ ou temas que serão desenvolvidos para uma nova *release* de um produto ou sistema, e seu objetivo é estabelecer uma resposta adequada às questões de escopo, cronograma e recursos para um projeto. O planejamento da *release* deve ocorrer no início do projeto, porém não deve ser um esforço isolado. Um bom plano de *release* é atualizado durante todo o projeto (geralmente no início de cada iteração) para que ele sempre reflita as expectativas atuais sobre o que será incluído na *release*. O próximo nível de planejamento é o da iteração, o que é realizado no início de cada iteração e deve ser realizado com base no trabalho realizado na iteração anterior.

O *Product Owner* (PO) deverá identificar quais as funcionalidades possuem maiores prioridades para serem realizados na nova iteração. Uma vez que se considere que num horizonte mais próximo do que o planejamento da *release*, o número de componentes a fazerem parte do plano devem ser menor. Durante o planejamento da iteração são descritas as tarefas que serão necessárias para transformar a solicitação de funcionalidades do PO em software funcional e devidamente testado. Finalmente, há o planejamento diário. As maiorias das equipes ágeis realizam uma reunião diária em pé para coordenar o trabalho e sincronizar esforço diário, sendo este um momento para as equipes avaliarem e revisarem os seus planos. Nesta reunião as equipes restringem o seu horizonte de planejamento até o dia seguinte, quando eles irão se reunir novamente. Devido a isso, eles se concentram no planejamento das tarefas e na coordenação das atividades individuais que conduzem à conclusão de cada tarefa. Pelo planejamento entre os três horizontes temporais *release*, iteração e diário, as equipes ágeis se concentram no que é visível e importante para o plano que eles estão criando (COHN, 2005).

⁵ História do usuário: Uma história do usuário é uma descrição concisa de uma necessidade do usuário, também podendo ser entendido como um requisito do usuário. Uma história do usuário busca descrever essa necessidade de uma forma simples e objetiva. Para atender a esta solicitação do cliente, escrita através de uma história, poderá ser necessário desenvolver várias tarefas pela equipe de desenvolvimento (KNIBERG, 2007; KNIBERG e SKARIN, 2009; Cohn, 2005).

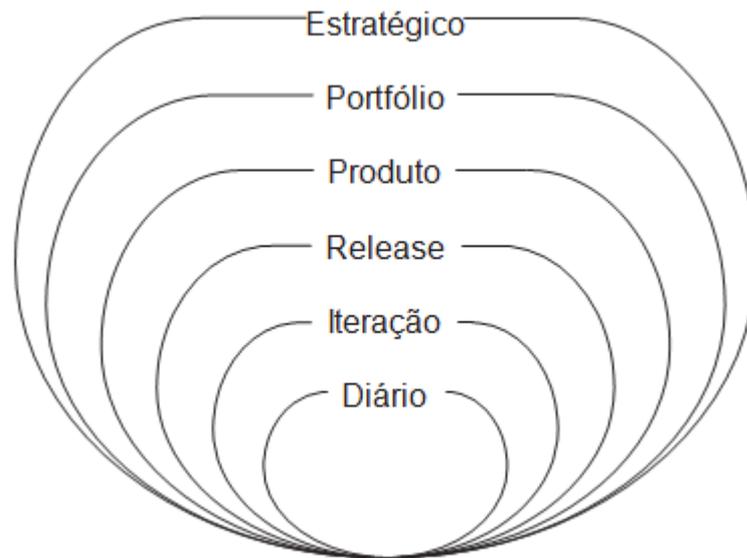


Figura 2. Planejamento cebola baseado em várias fases de planejamento

Fonte: Cohn(2005).

Sliger e Broderick (2008) apresenta um modelo de ciclo de vida de um projeto ágil baseado em fases e subfases conforme mostra a Figura 3. Inicialmente um projeto ágil possui uma reunião de planejamento onde é desenvolvida a visão do produto e um *roadmap*⁶ do produto. Este projeto deverá ser realizado através da subdivisão em múltiplas *releases* com um período de tempo definido, as *releases* através de subdivisão múltiplas iterações nas quais a equipe de desenvolvimento cria o produto propriamente dito, e diariamente a equipe avalia como foi seu trabalho. No início de cada fase deve ocorrer o planejamento, sendo este um processo chave para o sucesso. No final de cada fase deve ocorrer uma reunião de retrospectiva avaliando o que foi feito de bom, o que é necessário melhorar e quais ações serão tomadas em relação a equipe e em relação ao projeto.

⁶ *Roadmap*: É um mapa (plano) que descreve as macros funcionalidades do projeto, descrevendo como dever ser a evolução do produto ao longo do tempo. É uma representação de alto nível que contém as funcionalidades que devem conter em cada *release*, o cliente alvo, a arquitetura necessária e o valor de negócio esperado (Sliger e Broderick 2008).

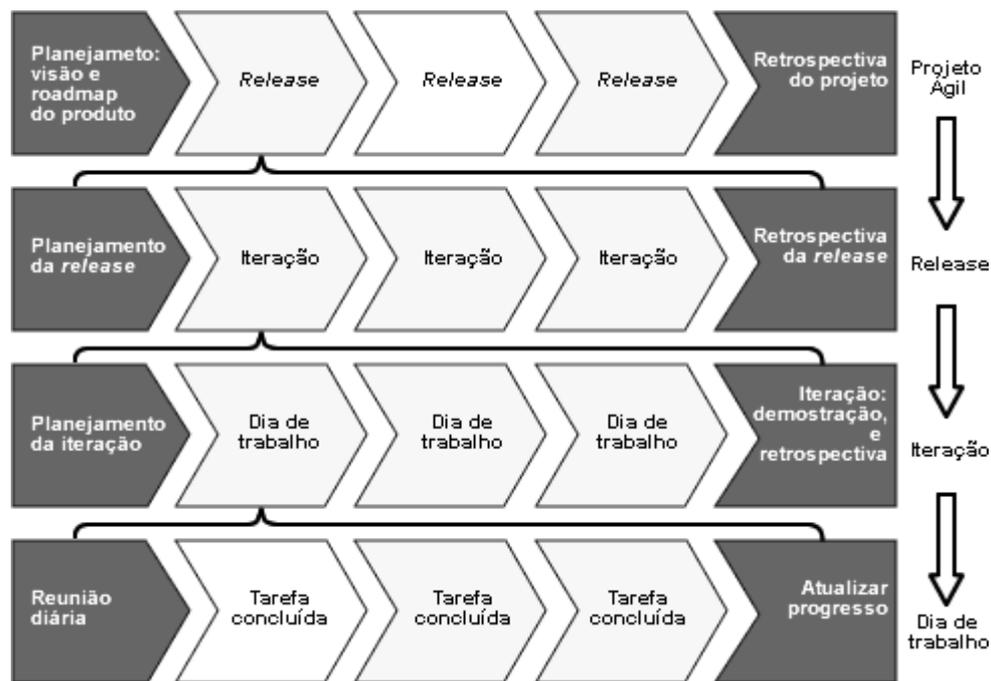


Figura 3. Fases e subfases do ciclo de vida de um projeto ágil

Fonte: Sliger e Broderick (2008).

Highsmith (2004) relata que algumas pessoas interpretam mal os princípios ágeis e acham que podem iniciar a codificação sem nenhum planejamento, sendo este um equívoco. Por outro lado, existe uma reação igualmente negativa aos projetos que investem meses de planejamento para especificação de requisitos e documentação de projeto. Sliger e Broderick (2008) enfatizam a importância de realizar o planejamento no início de cada fase e no seu final fazer reunião de retrospectiva discutindo os pontos-chaves sobre o andamento do projeto. As informações coletadas nestes momentos serão base para o processo de melhoria contínua. Highsmith (2004) afirma que deve haver um ponto de equilíbrio, sendo que para os projetos ágeis recomenda-se realizar uma Iteração 0 (zero). Sliger e Broderick (2008) relatam que a Iteração 0 (zero) serve para planejar o projeto e definir configurações de ambiente, discussão e definição sobre a arquitetura. Ambos os autores estão de comum consenso que deve ser o mais breve possível, pois o resultado de uma iteração implica na entrega de trabalho útil para o cliente, sendo que o resultado da iteração 0 é um trabalho útil apenas para a equipe de desenvolvimento.

2.1.2 Interessados e envolvidos em projetos ágeis de software

Ao iniciar em projetos de desenvolvimento com métodos ágeis deverá haver uma mudança significativa das funções e responsabilidades. Em um projeto tradicional as decisões são normalmente feitas de uma maneira hierárquica e os papéis são especificamente criados para apoiar esta estrutura, por isso manter os papéis tradicionais num projeto ágil poderá prejudicar esta mudança. Os projetos ágeis não possuem foco na hierarquia, mas sim em fazer as pessoas trabalharem juntos todos os dias (MOREIRA, 2013). Essa afirmação vem ao encontro dos valores e princípios do manifesto ágil os quais valorizam a interação e motivação dos indivíduos e enfatizam a importância da colaboração e satisfação do cliente (MANIFESTO ÁGIL, 2001).

Segundo Cohn (2005) um ponto crítico para o sucesso de um projeto ágil é que todos os participantes do projeto se vejam como uma equipe que vise um objetivo comum, sendo que não há espaço para ‘ficar em cima do muro’ no enfoque de um projeto ágil. Apesar de todos deverem trabalhar juntos como uma equipe, Cohn (2005) destaca que há uma série específica de funções entre os membros, sendo importante identificar e esclarecer as funções que cada membro desempenha no projeto ágil, a saber:

- Proprietário do produto (PO – *Product Owner*): certificar-se de que todos os membros da equipe possuem uma visão comum do projeto, estabelecer prioridades de modo a que a funcionalidade de valor mais elevado esteja sempre sendo trabalhado primeiro, tomar decisões que conduzam a um bom retorno sobre o investimento no projeto. Quando for um software comercial o PO poderá ser alguém da área de marketing ou de gerenciamento de produtos da empresa, se o software for para uso interno poderá ser um usuário, gerente, analista ou outra pessoa da empresa;
- Cliente: é a pessoa que tomou a decisão de comprar ou financiar o produto de software. Em um projeto de software para uso interno o cliente será normalmente um representante de grupo ou divisão. Nesses projetos o papel do PO e do cliente são muitas vezes combinados. Para um produto distribuído comercialmente, o cliente será a pessoa ou empresa que irá comprar o software;
- Desenvolvedor do software: este é um termo usado para denominar qualquer envolvido no desenvolvimento do software, isso inclui desenvolvedores, testadores, analistas de banco de dados, arquitetos, entre outros.
- Gerente de projetos: no desenvolvimento ágil de software o gerente de projetos mudou de papel, o qual deve ser concentrar mais na liderança do projeto do que no

gerenciamento propriamente dito. Em alguns projetos a pessoa que exerce a função de gerente de projetos poderá exercer outra função no mesmo projeto, como desenvolvedor ou PO.

O método Scrum define 3 tipos de papéis para equipes ágeis de software: o *Product Owner* (PO), o Time de Desenvolvimento e o Scrum Master. Os Times Scrum são auto-organizáveis e multifuncionais. Times auto-organizáveis escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros colaboradores de fora do Time. Times multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. O PO é o responsável por gerenciar o *Backlog*⁷ do Produto, priorizar os itens de *Backlog* para ser desenvolvido os itens de maior valor para o negócio. O Scrum Master é o profissional responsável pela liderança do time de desenvolvimento, garantindo que o time siga as regras do Scrum e apoiando-o em todos os aspectos para garantir o sucesso da Sprint (iteração). O time de desenvolvimento consiste de profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto 'Pronto' ao final de cada Sprint.

Pode-se perceber que existem vários grupos de *stakeholders* envolvidos nos projetos ágeis os quais possuem diferentes necessidades, interesses e responsabilidades. A maioria das literaturas cita os *stakeholders* como partes envolvidas que possuem interesses e responsabilidades nos projetos, porém as necessidades de métricas que cada perfil possui para fornecer informações que possibilitem melhor desempenhar o seu papel não são tratados e/ou discutidos, sendo esta uma lacuna a ser preenchida.

2.2 MEDIÇÃO DE DESEMPENHO DE PROJETOS DE SOFTWARE

Nesta sessão são apresentados os fundamentos na área de desenvolvimento de software sobre medição e avaliação de desempenho, descrição sobre o planejamento de medição e sobre a utilização de métricas nos métodos ágeis.

⁷ *Backlog* do produto: é uma lista contendo todas as funcionalidades desejadas para um produto. O conteúdo desta lista é definido pelo *Product Owner* e não precisa estar completo no início de um projeto. Pode-se começar com uma definição de visão do produto e sendo detalhada e ajustada conforme o software evolui (SCHWABER e SUTHERLAND, 2013).

2.2.1 Fundamentos sobre medição e avaliação de desempenho

A atividade da engenharia de software responsável pela análise de desempenho é a medição de software a qual desempenha um papel fundamental tanto para apoiar no controle de atividades básicas de controle do projeto quanto para monitorar planos de melhoria de processo de organizações de software (PRESSMAN, 2006). Trata-se de uma atividade utilizada para coletar dados necessários para compreender, gerenciar e melhorar os processos e produtos de software (BASILI, 1999). De um modo geral, a medição tem como objetivo quantificar atributos de entidades e de seus relacionamentos e então manipulá-los de maneira formal. Uma entidade pode ser uma pessoa, um objeto relacionado ao desenvolvimento de software ou um evento (KITCHENHAM, PFLEEGER e FENTON, 1995). A medição é o conjunto de operações com o objetivo de determinar um valor de uma medida (ISO/IEC, 2007), permitindo quantificar medidas de modo a manipulá-las e aprender mais a seu respeito. Uma medida é uma variável para a qual um valor é atribuído como resultado de uma medição (ISO/IEC, 2007). Uma métrica é um mapeamento de atributos de entidades do mundo real para entidades formais (valores da medição), de modo a descrevê-la de acordo com regras claramente definidas (BRIAND et al., 1996).

Ao coletar medidas e desenvolver métricas é possível obter indicadores. Um indicador é uma métrica ou combinação de métricas que proporcionam informações sobre o processo de software, em um projeto de software ou no próprio produto. Um indicador proporciona informações que o gerente de projetos ou engenheiros de software usam para ajustar o processo, o projeto ou o produto, de forma a incluir melhorias (PRESSMAN, 2011).

Ragland (1995) usa o termo indicador para referir-se à métrica que fornece informações úteis sobre o estado do processo e métrica como uma medida da extensão ou do grau a que um produto possui e exibe uma qualidade, uma propriedade ou um atributo.

Indicadores de desempenho, ou *Key Performance Indicators* (KPIs), medem a qualidade do processo para alcançar os resultados finais, avaliados por meio de critérios de sucesso previamente definidos (KEZNER, 2006). Os indicadores permitem que uma empresa possa obter informações importantes para a eficiência do processo. No entanto deve-se tomar cuidado quando se tratar de medições estratégicas, uma vez que uma escolha inadequada poderá levar a resultados errôneos. Kaplan e Norton (1997) classificam os indicadores de resultado (*lagginh indicators*) e indicadores de tendências (*leading indicators*). Os indicadores de resultado (*lagginh indicators*) tendem a ser indicadores de ocorrência “síntese

do passado” enquanto que os indicadores de tendências (*leading indicators*) mostram um “estado futuro”, permitindo interferências a fim de evitar que os resultados desejáveis sofram prejuízos. O equilíbrio entre indicadores de resultado (*lagging indicators*) e de tendência/direcionadores (*leading indicators*) fornece uma plataforma sólida para a definição e instituição de um sistema de monitoramento de desempenho (KAPLAN E NORTON, 1997).

A avaliação de desempenho se tornou um instrumento fundamental na gestão das organizações ante o ambiente de competitividade dinâmico no qual as empresas estão inseridas. Para Neely, Gregory e Platts (1995), a avaliação de desempenho pode ser definida como o processo de quantificar uma ação, no qual a mensuração é equacionada com a quantificação, e a ação é entendida como aquilo que leva ao desempenho. A avaliação de desempenho em empresas de desenvolvimento de software está associada à maturidade do processo de medição que será apresentado na próxima sessão.

2.2.2 Planejamento de medição

A adoção de uma estratégia de medição do comportamento dos processos relacionados ao desenvolvimento de software é apoiada pela implementação de um processo de medição. Possui como propósito coletar, armazenar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais (SOFTEX, 2011). Em geral, um processo de medição é implementado de forma evolutiva dentro da organização, sendo que, inicialmente, o controle do processo de desenvolvimento de software é exercido de forma reativa. As medições são difíceis de serem feitas e os dados são difíceis de coletar, como consequência do baixo nível de maturidade dos processos. À medida que os processos evoluem para níveis de maturidade mais elevados dos modelos de referência, o controle da execução do processo de desenvolvimento de software passa a ser pró-ativo baseado na previsão de desempenho dos componentes de processo.

A busca da melhoria do processo de software vem sendo discutida tanto na comunidade nacional quanto internacional. O CMMI é o modelo conhecido internacionalmente e desenvolvido pela *Software Engineering Institute* da *Carnegie Mellon University*. Nacionalmente, apoiado pelo governo brasileiro e outras entidades, foi

desenvolvido o MPS.BR (Melhoria de Processo do Software Brasileiro) o qual atende as mesmas normas técnicas do CMMI. A principal diferença entre os modelos é o número de níveis, sendo que o CMMI possui 5 níveis de maturidade e o MPS.BR possui 7 níveis de maturidade, alinhado com as necessidades das empresas de software brasileiras e permitindo as pequenas iniciarem na busca da melhoria do processo (SOFTEX, 2011; SEI, 2010). Tanto o CMMI quanto o MPS.BR não fornecem informações de como implementar a melhoria do processo, mas orientam qual é o propósito de cada processo e quais os resultados esperados. No MPS.BR a área de processo de medição possui como propósito coletar, armazenar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais. Os resultados esperados deste processo são:

- MED 1. Objetivos de medição são estabelecidos e mantidos a partir dos objetivos de negócio da organização e das necessidades de informação de processos técnicos e gerenciais;
- MED 2. Um conjunto adequado de medidas (métricas), orientado pelos objetivos de medição, é identificado e definido, priorizado, documentado, revisado e, quando pertinente, atualizado;
- MED 3. Os procedimentos para a coleta e o armazenamento de medidas são especificados;
- MED 4. Os procedimentos para a análise das medidas são especificados;
- MED 5. Os dados requeridos são coletados e analisados;
- MED 6. Os dados e os resultados das análises são armazenados;
- MED 7. Os dados e os resultados das análises são comunicados aos interessados e são utilizados para apoiar decisões.

Uma abordagem bastante utilizada para definição das métricas de software é a GQM (*Gols, Questions, Metrics*) a qual propõe a definição de métricas metas mensuráveis (BASILI et al., 1994). O modelo é uma estrutura hierárquica, como visto na Figura 4, sendo que o começa com a definição de um objetivo específico, após o objetivo é refinado em várias questões. Cada questão é então transformada em métricas as quais podem ser objetivas ou subjetivas. Uma mesma métrica pode ser utilizada para responder várias questões.

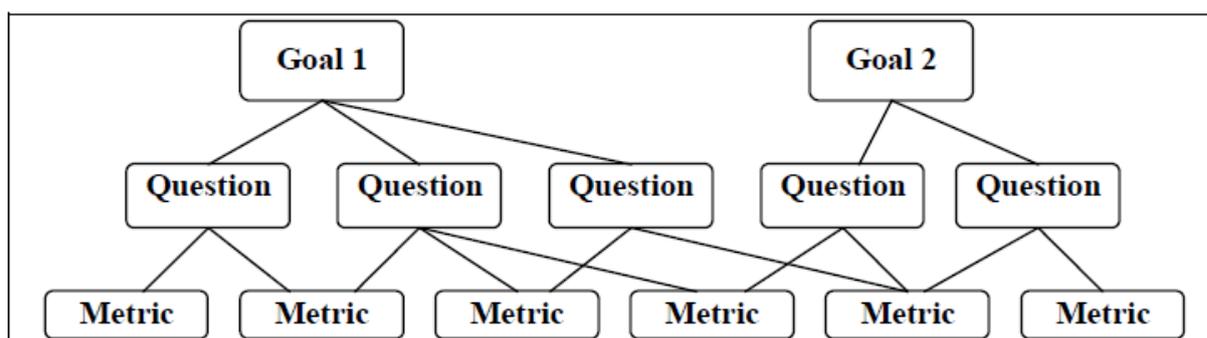


Figura 4. Abordagem Gols, Questions, Metrics.

Fonte: Basili et al., (1994).

Na próxima sessão é abordado sobre o tema métricas de software em projetos que utilizam métodos ágeis.

2.2.3 Utilização de métricas nos métodos ágeis

Uma das preocupações dos *stakeholders* envolvidos nos projetos ágeis é a dificuldade da definição de métricas que permitam visualizar a data de conclusão dos projetos, por isso é importante visualizar o progresso do projeto a cada iteração para não perder o controle do projeto (SCHATZ e ABDELSHAFI, 2005). As características que diferenciam os projetos tradicionais dos métodos ágeis de software, entre eles o processo do desenvolvimento do projeto que ocorre de forma emergente, iterativa e exploratória, com ciclos curtos, com escopo de projeto adaptável e reflexível (NERUR, 2007; MANIFESTO ÁGIL, 2001), torna necessário à definição de métricas e de um processo de medição adequado para esta abordagem.

Os métodos ágeis são vistos como um meio eficaz para encurtar os ciclos de entrega, adaptar-se as mudanças e entregar valor mais rapidamente aos clientes. Diante desta mudança de paradigma é necessário utilizar métricas que se adaptem a esta nova abordagem, auxiliando aos vários *stakeholders* envolvidos com informações úteis à tomada de decisões inerentes a sua função. Hayes et al., (2014) apontam diferenças importantes entre os processos tradicionais e ágeis que afetam as métricas:

- Utilização da abordagem de ciclos curtos com tempo fixo no lugar da abordagem de fases tradicionais para gerenciar o progresso. Isto significa que a equipe estará trabalhando dentro de um tempo limite estreitamente definido;

- Ao longo do projeto o calendário permanece fixo, porém o trabalho a ser realizado é variável.
- O cliente está envolvido em avaliações frequentes de entrega.

Com essas diferenças fundamentais é importante reconhecer a necessidade de métricas que atendam essas características. A revisão sistemática publicada por Dyba e Dingsoyr (2008) a qual faz um vasto levantamento sobre estudos empíricos utilizando métodos ágeis, não cita nenhum estudo sobre métricas de software, processo de medição ou indicadores de desempenho, constatando que nos primeiros anos após o advento do Manifesto Ágil os não estudos trataram destes temas. Os autores Dingsoyr et al., (2012) afirmam que entre os assuntos que ainda precisam ser evoluídos nos métodos ágeis está o gerenciamento desses projetos, especialmente sobre planejamento, controle, avaliação e estimativas.

Hartmann e Dymond (2006) afirmam que a medição nos métodos ágeis apoia a melhoria contínua do produto e do processo de desenvolvimento de software. Corroborando Gilb e Brodie (2007) reforçam que os métodos ágeis poderiam se beneficiar do uso de uma abordagem mais quantificada em todo o processo de implementação e os benefícios da adoção desta abordagem incluem a melhoria da comunicação dos requisitos, e melhor suporte para *feedback* e acompanhamento de progresso.

Segundo Gilb e Brodie (2007), os métodos ágeis têm foco insuficiente sobre os níveis de desempenho quantificados, ou seja, não fornecem métricas que indiquem a qualidade, que informe sobre economia de recursos ou capacidade de volume de trabalho dos softwares que estão sendo desenvolvidos. Os principais benefícios da sua adoção incluem a melhoria da comunicação dos requisitos, melhor suporte para *feedback* e acompanhamento de progresso, por isso é necessário realizar ao longo de toda a duração de um projeto ágil.

Hartmann e Dymond (2006) afirmam que não é possível aplicar os processos de medição e métricas utilizadas dos métodos tradicionais para os métodos ágeis. Uma vez que os métodos ágeis não tratam claramente sobre o processo de medição ou definição de planos de métricas, ao tentar utilizar métricas tradicionais podem talvez não se ajustar aos métodos ágeis. Hartmann e Dymond, (2006) afirmam que com o aumento da popularidade dos métodos ágeis, os profissionais mais tradicionais e preocupados com o monitoramento dos riscos estão sendo atraídos para os métodos ágeis, porém estão trazendo com eles os modos de avaliação incompatíveis com os valores e princípios ágeis. Hartmann e Dymond, (2006)

apresentam 11 atributos que boas métricas ou análises de diagnósticos nos métodos ágeis devem:

1. Afirmer e reforçar os princípios *lean* e ágeis;
2. Utilizar medidas de resultado e não medidas de saída;
3. Possibilitar observar tendência e não números;
4. Responder uma pergunta específica e para uma pessoa real;
5. Pertencer a um conjunto pequeno de métricas e diagnósticos;
6. Ser fácil de coletar;
7. Revelar ao invés de ocultar, com contexto e variáveis significativos;
8. Fornece “combustível” para conversas significativas;
9. Fornece informações de *feedback* com frequência e de forma regular;
10. Deve fornecer medidas de valor agregado do produto e sobre o processo;
11. Incentivar obter o nível de qualidade “o bom o suficiente” a partir de contexto do cliente de negócio e não no contexto do desenvolvedor.

Swaminathan e Jainm (2012), ao apresentar o caso de sucesso da adoção dos métodos ágeis, relataram que a equipe reagia conforme eram medidos, por isso foi fundamental a seleção de métricas corretas, utilizar poucas métricas, ser de fácil compreensão e que represente valor significativo para os interessados. Também relataram que a transparência adotada em todo o projeto entre a equipe e o cliente criou uma grande dose de confiança entre as partes, e que em qualquer momento o cliente sabia o *status* do projeto naquele exato momento. Trabalhando desta forma conseguiram fortalecer a cultura de confiança e coragem.

Atendo às questões de pesquisa secundária traçadas nesta pesquisa, buscou-se identificar as métricas para projetos ágeis de software, através de uma busca sistematizada na literatura. Esta revisão é apresentada na sequência.

2.2.4 Categorias de avaliação das métricas para projetos ágeis de software

O agrupamento das métricas em categorias de avaliação é importante para saber quais são os aspectos sobre o desenvolvimento de software estão sendo mais avaliados. Poucos estudos propõem categorização de métricas ou indicadores para projetos ágeis de software. Nesta revisão bibliográfica foram localizados os autores Hartmann e Dymond (2006), Cheng,

Jansen e Remmers (2009) e Poonacha e Bhattacharya (2012) que propõe formas de classificação.

Hartmann e Dymond (2006) propõem a classificação das métricas em organizacional e diagnóstico, sendo as métricas organizacionais abrangentes de longo prazo, retorno do negócio sobre o investimento e satisfação dos clientes, e de diagnóstico usadas para orientar a melhoria dos processos locais. A categoria diagnóstico proposta por Hartmann e Dymond (2006) é muito ampla e nela estariam aglomeradas a grande maioria das métricas, por este motivo optou em subdividi-la utilizando a classificação dos autores Cheng, Jansen e Remmers (2009) e Poonacha e Bhattacharya (2012). Cheng, Jansen e Remmers (2009) propõem a classificação em equipe/pessoas, tarefas do processo de desenvolvimento e qualidade. Poonacha e Bhattacharya (2012) propõem a classificação em pessoas, processo, cliente, qualidade e custos. As categorias a serem utilizadas para agrupamento das métricas serão: organizacional, equipe, qualidade e processo, conforme apresentado no Quadro 2.

Quadro 2. Categorização das métricas

	Organizacional	Equipe	Qualidade	Processo
Hartmann e Dymond (2006)	X			
Cheng, Jansen e Remmers (2009)	X	X	X	X
Poonacha e Bhattacharya (2012)	X	X	X	X

Fonte: Elaborado pelo autor.

A categoria Organizacional envolverá métricas que avaliam questões globais do negócio, financeiras, sobre a satisfação do cliente e decisões de nível gerencial; a categoria Equipe envolverá métricas que avaliam aspectos relacionados à equipe de desenvolvimento ou seus indivíduos especificamente; a categoria Processo envolverá métricas que avaliam aspectos relacionados ao processo de desenvolvimento iterativo incremental, gestão das tarefas ou controles necessários inerentes ao processo de desenvolvimento ágil; e a categoria Qualidade envolverá métricas que avaliam aspectos relacionados ao controle de qualidade do produto (HARTMANN e DYMOND, 2006; CHENG, JANSEN e REMMERS, 2009; POONACHA e BHATTACHARYA, 2012). Esta classificação será utilizada neste e nos demais capítulos para classificar as métricas pesquisadas.

2.3 MÉTRICAS UTILIZADAS PARA PROJETOS DE SOFTWARE CITADAS NA LITERATURA

A revisão sistematizada da literatura teve como objetivos: (i) investigar estudos que tratem especificamente do tema utilização de métricas em projetos de desenvolvimento de software que utilizem métodos ágeis; (ii) identificar métricas que permitam as empresas desenvolvedoras de software que utilizem métodos ágeis a controlar os seus projetos nas fases de *release*, iteração e diário. A partir destes objetivos foram identificadas as palavras-chaves, os termos de busca e a identificação das principais bases de dados eletrônicas relevantes para esta pesquisa. Esta análise foi realizada de forma cuidadosa para obter os artigos relevantes para responder às perguntas deste estudo e obter o maior número de estudos para análise.

A pesquisa foi realizada nas bases eletrônicas *IEEE Xplore* e *ISI Web of Science*. Estas bases foram escolhidas porque oferecerem publicações pertinentes à área de conhecimento. A base *IEEE Xplore* pela variedade de publicações de alto nível específicos na área de Ciência da Computação e a *ISI Web of Science* por indexar uma quantidade significativa de periódicos e congressos, além de ter acesso para recuperação de referências e na facilidade de recuperação do texto completo do artigo. O idioma escolhido para pesquisa foi o inglês, por ser padrão para publicação internacional. Foram considerados artigos publicados a partir de 2001, ano de decorrência do Manifesto Ágil, o qual foi um marco importante que impulsionou debates e pesquisas sobre o tema.

O Quadro 3 apresenta os *strings* de busca que foram utilizadas para pesquisa em cada base eletrônica, onde podem ser identificadas as palavras chaves utilizadas.

Quadro 3. *Strings* de busca

String utilizada na base ISI Web of Science
((("agile software") OR ("agile method*") OR ("agile development*") OR (scrum AND software) OR ("extreme programming") OR (lean AND software) OR ("agile project")) AND (indicator* OR metric* OR performance OR measure*))
String utilizada na base IEEE
((("agile AND software) OR ("agile method") OR ("agile methodology") OR ("agile development") OR (scrum AND software) OR ("extreme programming") OR (lean AND software) OR ("agile project")) AND (indicator OR metrics OR performance OR measurement OR measure))

Fonte: Elaborado pelo autor.

O procedimento para realização da seleção dos artigos foi realizado em 3 etapas:

1ª Etapa - Pesquisa nas bases eletrônica e catalogação dos artigos: nesta etapa foi realizada a pesquisa com a utilização da expressão de busca nas bases selecionadas. Todas as publicações retornadas foram catalogadas para análise posterior.

2ª Etapa - Filtro através da leitura dos títulos e resumos: nesta etapa foi realizado o filtro excluindo os trabalhos que não tratavam sobre métodos ágeis no desenvolvimento de software, duplicados, apenas o resumo disponível ou não estavam em inglês.

3ª Etapa – Aplicação dos critérios de exclusão: nesta etapa foi realizado o filtro dos artigos através da aplicação dos seguintes Critérios de Exclusão (CE):

CE1 – Estudos que não tratem especificamente do tema sobre métricas em projetos de desenvolvimento de software que utilizam métodos ágeis.

CE2 – Estudos que não tratem de métricas para controle do projeto nas fases de *release*, iteração e diário de um projeto ágil, desta forma foram descartados aqueles que tratavam de métricas de nível estratégico ou de controle das atividades de programação (ex. controle do código fonte, práticas de programação, refatoração).

CE3 – Em trabalhos semelhantes do mesmo autor, apenas o mais completo foi considerado.

Na situação em que um estudo atendeu parcialmente ao critério de exclusão CE2, fornecendo métricas para controle do projeto ágeis nas fases de *release*, iteração e diário e também apresente métricas fora deste escopo, neste caso o estudo foi selecionado, porém apenas as métricas que atendem aos objetivos da pesquisa foram utilizadas para análise.

Após a definição do protocolo de pesquisa foram realizados testes nas bases selecionadas com o objetivo de validar a expressão de busca. A pesquisa foi conduzida no período de junho a agosto de 2012. Na base *IEEE Xplore* foram recuperados 411 e na base *ISI Web of Science* 144 artigos, totalizando 555 artigos. Os artigos foram filtrados seguindo os critérios estabelecidos, sendo que ao final da análise resultaram 12 artigos aprovados pelos critérios estabelecidos.

Após a conclusão da análise dos artigos, também foi realizada pesquisa no Google acadêmico pelas palavras chaves onde foram localizados 2 trabalhos relevantes os quais foram adicionados a pesquisa, totalizando 14 artigos aprovados. No Quadro 4 são apresentados os estudos foram encontrados. Em alguns casos os trabalhos não tinham como

tema de pesquisa a utilização de métricas em projeto ágeis de software, porém abordavam com certa abrangência o assunto, por esse motivo foram mantidos.

Através da análise do Quadro 4 é possível observar dos 14 estudos identificados 8 (57%) não tratam exclusivamente do tema sobre utilização de métricas de software e apenas um apresenta a contextualização da utilização prática, detalhamento das métricas e especificação das fórmulas, mostrando uma carência de estudos aprofundados sobre o assunto.

Quadro 4. Estudos selecionados na revisão da literatura

Estudo	Referência	Estudo específico sobre métricas em projetos ágeis de software	Metodologia de estudo	Tipo de fonte da publicação	Contextualização da utilização prática	Detalhamento das métricas e especificação das fórmulas
E1	PETERSEN, K. WOHLIN, C. Measuring the flow in lean software development. Software: Practice and Experience. v. 41. n. 9, p. 975-996, 2011.	Sim	Estudo de caso	J	Sim	Sim
E2	FEYH, M.; PETERSEN, K. Lean Software Development Measures and Indicators. Lean Enterprise Software and Systems. p. 32-47, 2013.	Sim	Mapeamento sistemático	J	Parcial	Parcial
E3	TALBY, D.; DUBINSKY, Y. Governance of an agile software project. Software Development Governance, 2009. SDG '09. ICSE Workshop on. p.40-45.	Não	Estudo de caso	C	Sim	Não
E4	SWAMINATHAN, B.; JAIN, K. (2012). Implementing the Lean Concepts of Continuous Improvement and Flow on an Agile Software Development Project: An Industrial Case Study. AGILE India (AGILE INDIA), 2012. p. 10-19.	Não	Estudo de caso	C	Sim	Parcial
E5	ABBAS, N.; GRAVELL, N.; WILLS, G. (2010). The Impact of Organization, Project and Governance Variables on Software Quality and Project Success. Agile Conference (AGILE), 2010. p.77-86.	Não	Survey	C	Parcial	Não
E6	DUBINSKY, Yael et al. Agile metrics at the israeli air force. In: Agile Conference, 2005. Proceedings. IEEE, 2005. p. 12-19.	Sim	Estudo de caso	C	Sim	Não

E7	HARTMANN, D.; DYMOND, R. Appropriate agile measurement: Using metrics and diagnostics to deliver business value. In: Agile Conference, 2006. IEEE, 2006. p. 6 pp.-134.	Sim	Proposta	C	Sim	Não
E8	MIDDLETON, P.; JOYCE, D. Lean Software Management: BBC Worldwide Case Study. Engineering Management, IEEE Transactions on. v. 59, n.1, p.20-32, 2012.	Não	Estudo de caso	J	Sim	Parcial
E9	GREEN, P. Measuring the Impact of Scrum on Product Development at Adobe Systems. System Sciences (HICSS), 2011 44th Hawaii International Conference on. p. 1-10.	Não	Estudo de caso	C	Sim	Não
E10	JAVDANI, T. et al. On the Current Measurement Practices in Agile Software Development. International Journal of Computer Science Issues (IJCSI), v. 9, n. 4, 2012.	Sim	Revisão de literatura	J	Sim	Não
E11	LEE, G.; XIA, M. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. MIS Quarterly, v. 34, p. 87-114, 2010.	Não	Survey	J	Parcial	Não
E12	MISHRA, Deepti; BALCIOGLU, Eda; MISHRA, Alok. Measuring Project and Quality aspects in Agile Software Development. TTEM-Technics Technologies Education Management, v. 7, n. 1, 2012.	Não	Revisão de literatura	J	Parcial	Parcial
E13	SATO, D. T. Uso eficaz de métricas em métodos ágeis de desenvolvimento de software. 2007. Dissertação. Universidade de São Paulo, 2007.	Sim	Estudo de caso	D	Sim	Parcial
E14	GUSTAFSSON, J. Model of Agile Software Measurement: A Case Study. 2011. Dissertação. University of Gothenburg, 2011.	Não	Estudo de caso	D	Parcial	Não

Fonte: Elaborado pelo autor.

Legenda: C-Congresso, P-Periódico e D-Dissertação

A Tabela 1 apresenta a distribuição temporal dos estudos relevantes selecionados para análise por tipo de publicação e por ano. É possível notar que 10 (71%) dos artigos foram publicados nos últimos 4 anos, num período de investigação de 13 anos, o que demonstra que o tema de pesquisa é recente. Observou-se o aumento do número de publicações a partir do ano de 2010. Do total 6 (43%) foram publicados em periódicos, 6 (43%) em congressos e 2 (14%) são dissertações de mestrado.

Tabela 1. Resumo dos estudos por tipo de publicação e por ano

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Total
Congressos	-	-	-	-	1	1	-	-	1	1	1	1	-	6
Periódicos	-	-	-	-	-	-	-	-	-	1	1	3	1	6
Dissertações	-	-	-	-	-	-	1	-	-	-	1	-	-	2
Total	0	0	0	1	1	1	1	0	1	2	3	4	1	14
Percent.	0%	0%	0%	7%	7%	7%	7%	0%	7%	14%	21%	29%	7%	100%

Fonte: Elaborado pelo autor.

Na sequência foram analisados os 14 estudos aprovados e identificadas as métricas citadas pelos autores que atendem ao escopo da pesquisa, as quais estão apresentadas na Tabela 2. Depois de identificadas as métricas foram classificadas conforme de categorias provenientes da revisão bibliográfica apresentada na sessão 2.2.4. Optou-se neste capítulo em não descrever e contextualizar as métricas, pois esta informação é apresentada em detalhes no capítulo 4.

Tabela 2. Métricas identificadas na revisão de literatura

Nome	Cate- goria	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	T	% Freq
Retorno sobre o investimento	ORG					X	X				X		X	X	X	6	43%
Velocidade da equipe	EQP					X	X				X		X	X	X	6	43%
<i>Burndown</i> da iteração	PROC			X			X				X				X	4	29%
Cobertura de testes automatizados	QUAL			X	X		X								X	4	29%
<i>Cycle Time</i>	PROC		X						X					X		3	21%
Diagrama de fluxo cumulativo	PROC	X	X		X						X					3	21%
Fluxo das tarefas no quadro kanban / fila	PROC	X													X	3	21%
Funcionalidades Testadas e Entregues (<i>Running testing features-RTF</i>)	QUAL					X							X	X		3	21%
<i>Lead time</i>	PROC		X						X						X	3	21%
Defeitos encontrados na fase de testes	QUAL						X			X					X	3	21%
Número de integrações por dia	PROC			X			X									3	21%
Custos	ORG											X			X	2	14%
Defeitos encontrados pelo cliente	QUAL					X				X						2	14%
Total de horas consumidas no projeto	PROC								X			X				2	14%
Atividades que geraram retrabalho	QUAL										X					1	7%
Comparação do número de defeitos encontrados com outros ciclos do desenvolvimento	QUAL									X						1	7%
Dívida técnica	QUAL														X	1	7%
Defeitos em aberto	QUAL								X							1	7%
Frequência das <i>releases</i>	PROC								X							1	7%
Grau de atendimento aos requisitos	QUAL											X				1	7%

2.4 CONSIDERAÇÕES SOBRE O REFERENCIAL

Este capítulo apresentou a revisão de literatura deste trabalho onde foi possível entender o contexto sobre o desenvolvimento de software, as motivações que levam as empresas a adotarem métodos ágeis, os fundamentos sobre métricas e processo de medição no desenvolvimento de software e sua relação com os métodos ágeis. Não foi localizada na revisão de literatura a existência de um processo de medição adequado para projetos ágeis de desenvolvimento de software, sendo, assim confirmada a lacuna de pesquisa.

Quanto a utilização de métricas em projetos ágeis de software foi realizada uma busca sistematizada na literatura, a qual teve como objetivos: investigar estudos que tratem especificamente do tema utilização de métricas em projetos de desenvolvimento de software que utilizem métodos ágeis; e identificar métricas que permitam as empresas desenvolvedoras de software que utilizem métodos ágeis a controlar os seus projetos nas fases de *release*, iteração e diário. Como resultado foram localizados 14 estudos que pesquisaram sobre métricas em projetos de software que utilizam métodos ágeis e um conjunto de 30 métricas, classificadas em 4 categorias. No entanto, permanece a dúvida se tais métricas são empregadas na rotina de medição das empresas.

Desta forma, para atender às questões secundárias de pesquisa, o próximo passo será comparar os resultados obtidos sobre as métricas identificadas na revisão da literatura com a prática das empresas, utilizando-se para isso de uma pesquisa de campo, que será apresentada no capítulo 3.

3 PESQUISA DE CAMPO

Este capítulo apresenta a realização e os resultados da pesquisa de campo constituída com empresas desenvolvedoras de software, experientes na utilização de métodos ágeis, conforme planejamento apresentado no capítulo 1.

3.1 CONTEXTO DAS ORGANIZAÇÕES ESTUDADAS E PERFIL DOS ENTREVISTADOS

Conforme descrito na sessão 1.3.2 do método foram entrevistados profissionais de nove empresas, sendo que no Quadro 5, é apresentada uma síntese dos dados das empresas pesquisadas e de seus entrevistados. Dessas, sete empresas utilizam métodos ágeis em 100% dos seus projetos e duas analisam o contexto dos projetos para decidir qual o método utilizar, podendo haver uma mescla entre o método tradicional e o ágil. As empresas justificam suas escolhas pelo contexto da empresa e do mercado no qual estão inseridos, conforme demonstra os depoimentos a seguir:

“Não aceitamos trabalhar de outra forma, pois vender projeto com escopo fechado o risco é muito grande. Os clientes querem normalmente encher de funcionalidades que acabam prolongando muito o tempo dos projetos. Pelo fato da empresa ser pequena ela precisa ter muito controle senão pode ter prejuízo e isso é um risco para a sobrevivência da empresa” (Diretor da empresa G).

“Para projetos internos utilizamos os métodos ágeis em todos os projetos, porém para projetos externos fazemos uma mescla entre o tradicional e o ágil, pois os clientes normalmente não entendem a dinâmica dos métodos ágeis onde os projetos são orientados a entrega” (Diretor da empresa E).

Quadro 5. Síntese dos dados das empresas pesquisadas e de seus entrevistados

DADOS DAS EMPRESAS					DADOS DOS ENTREVISTADOS	
Nº	Empresa	Ano de início da adoção dos métodos ágeis	Porte	Utilizam métodos ágeis em todos os projetos	Perfil do entrevistado	Tempo experiência em ágil (ano de início)
1	Empresa A	2009	M	Depende do contexto, pode haver mescla entre o tradicional com ágil	Gerente de desenvolvimento	2009
2	Empresa B	2009	G	Sim	Diretor	2010
3	Empresa C	2008	G	Sim	Diretor	2008
4	Empresa D	2009	M	Sim	Scrum Master	2010
5	Empresa E	2010	P	Depende do contexto, pode haver mescla do tradicional com ágil	Diretor	2007
6	Empresa F	2007	P	Sim	Diretor	2003
7	Empresa G	2011	P	Sim	Diretor	2009
8	Empresa H	2007	G	Sim	Supervisor área de desenvolvimento	2007
9	Empresa I	2009	M	Sim	Diretor	2003

Fonte: Elaborado pelo autor com base nos dados da pesquisa de campo.

Todas as empresas possuem equipes de desenvolvimento subdivididas em grupos variando entre 3 e 10 pessoas por equipe. As empresas de portes médio e grande possuem equipes com maior número de pessoas (entre 5 e 10 pessoas); nas empresas menores as equipes variam de 3 a 6 pessoas. As empresas A, G e H adotam o método ágil Scrum em suas empresas percebendo-se a ênfase da adoção do ágil para gestão de projetos. No entanto, as empresas B, C, D, E, F e I, não adotam apenas um método específico, mas sim um conjunto de práticas ágeis onde é possível constatar a grande influência dos métodos Scrum, Lean e Kanban associados. Uma vez identificado o contexto das empresas e perfil dos entrevistados partiu-se então para a análise das informações sobre as métricas utilizadas pelas empresas para controlar o desempenho dos projetos ágeis de desenvolvimento de software.

3.2 MÉTRICAS UTILIZADAS PELAS EMPRESAS

As métricas citadas nas entrevistas são apresentadas na Tabela 3, consistindo num total de 26 métricas. A estas foram unidas às 14 métricas encontradas na revisão bibliográfica do item 2.3.3, totalizando 40 métricas. Foram incluídas neste quadro as métricas encontradas

na revisão bibliográfica visando verificar se existe relação entre as métricas citadas na pesquisa de campo e as métricas identificadas na pesquisa bibliográfica.

A Tabela 3 apresenta o nome das métricas e categorias de avaliação (Categ.) também provenientes da revisão bibliográfica sessão 2.2.4. As categorias descritas na revisão da literatura são: Equipe/indivíduos (EQP); Qualidade (QUAL); Organizacional (ORG); Processo (PROC). Optou-se neste quadro em apresentar apenas o nome da métrica, pois no capítulo 4 sessão 4.1 essas métricas serão contextualizadas em profundidade.

A Tabela 3 contém ainda a frequência absoluta de citações das métricas por empresa entrevistada (Fi_EC), a frequência relativa (EC%), ambas provenientes do estudo de campo e, por fim, a frequência absoluta de citações destas métricas na revisão bibliográfica (Fi_RB) e suas respectivas frequências relativas (RB%). O cálculo do EC% foi realizado através da relação entre o número Fi_EC e o número de empresas entrevistadas (9 empresas) e para o cálculo do RB% através da razão entre o Fi_RB e o número de estudos encontrados na revisão da literatura (14 estudos).

Observou-se que em algumas entrevistas eram utilizadas denominações diferentes para se referir à mesma métrica, e, nestas situações, foram escolhidas as denominações que melhor caracterizassem a cada situação da medição. Como exemplo é possível citar ‘acurácia das estimativas’ e ‘percentual de distorção entre a estimativa planejada e realizada’, optando-se, neste caso, em utilizar a nomenclatura ‘acurácia das estimativas’.

Referente às métricas utilizadas pelas empresas, percebeu-se que existem algumas métricas bastante comuns na sua prática como: Precisão na execução das tarefas, Fluxo das tarefas no quadro Kanban, Defeitos encontrados na fase de testes e o gráfico *Burndown* da iteração, citadas por pelo menos seis das nove empresas entrevistadas. Existem outras métricas que foram citadas esporadicamente, mas que cumprem objetivos de medição considerados relevantes nas empresas, tais como: Cobertura de testes automatizados, Dívida técnica, Motivação da equipe (*Happyness*) e Total de horas consumidas em atividades de administração do projeto. De uma forma geral, observou-se que todas as métricas citadas são consideradas úteis pelos entrevistados e que não existe um conjunto extenso de métricas por empresa, variando de 4 a 15 métricas.

Tabela 3. Relação entre as métricas identificadas nas entrevistas do estudo de campo com as métricas da revisão bibliográfica

MÉTRICAS E CATEGORIAS		EMPRESAS										FREQUÊNCIAS			
Métricas	Categ.	A	B	C	D	E	F	G	H	I	Fi_	EC%	Fi_RB	RB%	

											EC.			
Burndown da iteração	PROC	X	X	X	X	X	X	X	X	X	9	100%	4	29%
Fluxo das tarefas no quadro Kanban	PROC	X	X	X	X	X	X	X		X	8	89%	3	21%
Defeitos encontrados na fase de testes	QUAL	X		X	X		X		X	X	6	67%	3	21%
Precisão na execução das tarefas	PROC			X	X	X	X	X	X		6	67%	0	0%
Custos	ORG	X		X				X	X	X	5	56%	2	14%
Lead time	PROC		X	X			X	X		X	5	56%	3	21%
Acurácia das estimativas	PROC	X	X	X				X			4	45%	0	0%
Burnup da release/projeto	PROC		X	X	X			X			4	45%	0	0%
Defeitos encontrados pelo cliente	QUAL	X			X					X	3	33%	2	14%
Cycle time	PROC		X	X			X				3	33%	3	21%
Densidade dos defeitos	QUAL			X					X	X	3	33%	0	0%
Velocidade da equipe	EQP		X	X			X				3	33%	6	43%
Cobertura de testes automatizados	QUAL		X				X				2	22%	4	29%
Dívida técnica	QUAL			X		X					2	22%	1	7%
Motivação da equipe (<i>Happiness</i>)	EQP						X			X	2	22%	0	0%
Número de horas trabalhadas em atividades de desenvolvimento	ORG			X	X						2	22%	0	0%
Atividades que geraram retrabalho	QUAL			X					X		2	22%	1	7%
Retorno sobre o investimento	ORG						X				1	11%	6	43%
Total de horas consumidas no projeto	ORG								X		1	11%	2	14%
Impedimentos removidos	PROC				X						1	11%	1	7%
Grau de atendimento aos requisitos funcionais	QUAL								X		1	11%	1	7%
Throughput	PROC									X	1	11%	1	7%
Mudanças nos requisitos	PROC			X							1	11%	0	0%
Severidade dos defeitos por iteração	QUAL								X		1	11%	0	0%
Tarefas não planejadas	PROC				X						1	11%	0	0%
Total de horas consumidas em atividades de administração do projeto	ORG								X		1	11%	0	0%
Diagrama de fluxo cumulativo	PROC										0	0%	3	21%
Funcionalidades Testadas e Entregues (<i>Running testing features-RTF</i>)	QUAL										0	0%	3	21%
Número de integrações por dia	PROC										0	0%	2	14%
Work in Progress	PROC										0	0%	1	7%
Comparação do número de defeitos encontrados no ciclo atual com	QUAL										0	0%	1	7%

revisão da literatura e nenhuma vez na pesquisa de campo. Porém as empresas A, B, C e G relataram que fazem a avaliação da satisfação do cliente de forma subjetiva através das reuniões com o cliente.

Foi questionado aos entrevistados se eles utilizam sempre as mesmas métricas para todos os projetos ou se existem métricas diferentes para projetos de características diferentes. Os entrevistados das empresas A, D, E, F, G, H e I responderam que utilizam as mesmas métricas para todos os projetos, porém nas empresas B e C eles relataram que para realizar manutenção ou evolução dos softwares já desenvolvidos, não utilizam o conceito de iterações para realizar o planejamento e entrega das atividades, mas sim o conceito de fluxo de atividade contínua, e neste caso utilizam as práticas e métricas recomendadas pelos métodos *Lean/Kanban*.

Estas métricas também foram citadas na revisão da literatura observando-se uma convergência entre os dados. As métricas de origem dos métodos *Lean/Kanban* identificadas na pesquisa de campo e na revisão de literatura são apresentadas a seguir por ordem decrescente de citações: Fluxo das tarefas no quadro Kanban, *Lead time*, *Cycle time*, Diagrama de fluxo cumulativo, *Throughput*, Gargalos do fluxo de tarefas e *Work in Progress*.

A Tabela 4 apresenta uma síntese dos dados do Tabela 3 e o total de métricas por categorias de avaliação das métricas, onde é possível comparar o número e percentual das métricas e o número e percentual de citações, entre a pesquisa de campo e a revisão da literatura. Este agrupamento permitiu identificar que nestas empresas de desenvolvimento de software pesquisadas os aspectos mais avaliados tratam-se do monitoramento do processo (42% das métricas e 55% das citações) e do controle de qualidade (31% das métricas e 26% das citações). Também é possível observar que as categorias Organizacional (19% das métricas e 13% das citações) e Equipe (8% das métricas e 6% das citações) possuem um número de métricas limitado e o número de citações demonstra que estas categorias são menos monitoradas ao longo do projeto. Através da comparação entre as categorias citadas na revisão da literatura e na pesquisa de campo é possível observar que ambos possuem mesma ordem citação das categorias de avaliação e com percentual de citações ligeiramente diferentes.

Tabela 4. Métricas por categorias de avaliação e número de citações da pesquisa de campo e da revisão de literatura

TOTAL	PESQUISA DE CAMPO	REVISÃO DA LITERATURA
-------	-------------------	-----------------------

Categoria	Número de métricas	Número de métricas	% das métricas	Número de citações	% das citações	Número de métricas	% das métricas	Número de citações	% das citações
Equipe	2	2	8%	5	6%	1	3%	6	10%
Qualidade	12	8	31%	20	26%	10	33%	18	29%
Organizacional	6	5	19%	10	13%	4	13%	11	18%
Processo	20	11	42%	43	55%	15	50%	27	44%
	40	26	100%	78	100%	30	100%	62	100%

Fonte: Elaborado pelo autor com base nos dados da pesquisa de campo e da revisão da literatura.

Nas empresas A, B, F, G e H foi permitido visitar o ambiente de trabalho do time de desenvolvimento. Nessas empresas foi possível observar que existe uma preocupação com a transparência das informações sobre o projeto e ficam visualmente disponíveis para a equipe num espaço informativo através quadro Kanban e no gráfico *burndown*. No quadro Kanban fica evidente o controle das tarefas por fase do desenvolvimento, tarefas em impedimento, sinalização das tarefas com problemas nos testes, visualizar gargalos de tarefas. Entretanto, os dados sumarizados sobre as métricas utilizadas não ficam disponíveis neste espaço informativo sendo apresentados ao time de desenvolvimento nas reuniões de monitoramento do projeto e de retrospectiva. A empresa H dispõe as informações via software e não possui quadros visuais no ambiente de trabalho, sendo que todas as informações sobre o projeto estão disponíveis via software.

Entre as empresas entrevistadas 2 possuíam certificação CMMI nível 2 (empresas C e H) e 1 possui certificação MPS.BR nível G (empresa A), cujas certificações ocorreram com a utilização de métodos ágeis. Na opinião dos entrevistados dessas empresas, esta combinação entre adoção de métodos ágeis e a formalização dos processos devido aos modelos de maturidade permitiu gerenciar melhor os seus projetos, o que consequentemente tornou a empresa mais eficiente para execução dos seus processos, o que em última análise refletiu para eles uma maior satisfação dos clientes. Nessas empresas (empresas A, C e H) as métricas são formalizadas através de um plano de medição sendo este uma exigência desses modelos de maturidade, o qual é conhecido por todos e periodicamente revisado.

Foi relatado pela empresa G que a certificação surgiu como uma necessidade para o amadurecimento dos processos de desenvolvimento de software da empresa, pois com a adoção dos métodos ágeis vieram à tona os problemas, porém o método não forneceu subsídios para resolvê-los. Conforme depoimento:

“Um dos principais benefícios que o ágil trouxe para a empresa foi a visibilidade, principalmente em mostrar os problemas da área de desenvolvimento, porém não resolveu os nossos problemas. A empresa teve que correr atrás de alternativas em como resolvê-los, o que o CMMI possibilitou [...] Antes da definição formal das métricas a análise dos dados ocorria muito no ‘feeling’ do gerente de projetos, e o acompanhamento e análise do desempenho do projeto ocorria basicamente pela coleta dos dados das tarefas realizadas pelos membros do time. O CMMI trouxe a necessidade de definir um plano de medição, ter uma data de revisão deste plano, ter reuniões previamente agendadas, o que formalizou e fez a empresa evoluir muito neste sentido. O que trouxe maturidade na utilização das métricas foi o CMMI e não o SCRUM” (Supervisor da área de desenvolvimento da empresa G).

Ao perguntar para as empresas quais são os critérios utilizados para definição das métricas, apenas as empresas com CMMI ou MPS.BR (empresas A, C e H) responderam que foi a partir da análise das necessidades da organização sobre o controle dos projetos. A empresa B possui um histórico de boas práticas ágeis que obtiveram sucesso ao longo da adoção dos métodos ágeis inclusive referente as métricas a serem utilizadas. As empresas D, E, F e I estão baseadas nas experiências práticas dos profissionais e troca de experiência com a comunidade ágil. A empresa G admitiu que a sua forma de definição é empírica baseada na experimentação, conforme declaração a seguir:

“Um dos principais problemas que nós enfrentamos ao adotar os métodos ágeis é o fato de não ter métricas claras para monitorar os projetos. Desde o início estamos tentando aplicar algumas métricas para entender como as coisas estão andando. Atualmente temos algumas métricas que utilizamos, mas estamos sempre atentos às novidades referentes à adoção dos métodos ágeis, e procuramos testar novas práticas sempre. Após um tempo de experiência fazemos uma análise se continuamos ou não com aquela prática” (Diretor da empresa G).

Essas informações demonstram como diferentes contextos de empresas evoluem na utilização das métricas e como os gestores buscam alternativas para melhorar a forma de monitorar os seus projetos. Reflete também a falta de sistemática formal que sirvam de referência para o planejamento de métricas.

Algumas métricas citadas pelas empresas E e H não foram consideradas dentro do conjunto por essa pesquisa por serem métricas de controle de código e, na delimitação do estudo, foram consideradas fora do escopo desta pesquisa:

- percentual da cobertura do código através de auditoria pelo software Sonar (Empresa H);
- tamanho do produto medido em linhas pelo software Sonar (Empresa E).

Após analisar as métricas utilizadas para avaliar os projetos ágeis de software partiu-se para investigar quais são as funções envolvidas em cada fase do projeto e quais as suas responsabilidades nos projetos, conforme apresentado no próximo item.

3.3 FASES TÍPICAS DOS PROJETOS ÁGEIS

As fases típicas dos projetos mencionadas pelos entrevistados estão organizadas no Quadro 6. Alguns entrevistados relataram com detalhes as suas fases do projeto, porém outros foram sucintos nas suas respostas. Para conseguir observar a semelhança entre as fases utilizadas pelas empresas, foram elaborados dois quadros (Apêndice 2) e Quadro 6 que contêm as fases, a frequência de identificação por empresa que as utilizam e a correlação com a nomenclatura das fases utilizadas por Sliger e Broderick (2008) e Cohn (2005).

Analisando as fases citadas, é possível observar que todas correspondem a um projeto ágil com desenvolvimento iterativo incremental, semelhante ao proposto por Sliger e Broderick (2008) e Cohn (2005). Após iniciar a iteração todas as empresas realizam reuniões constantes de acompanhamento dos projetos. Nem todas as possuem as mesmas fases, destacando as empresas A, H e I que além das reuniões diárias realizam também reuniões semanais e mensais de monitoramento do projeto, reforçando a grande importância de monitorar o projeto.

Algumas empresas utilizam terminologias diferentes para tratar da mesma etapa, por isso o Quadro 6 apresenta as fases dos projetos com a denominação tratada por cada empresa e a terminologia utilizada por Sliger e Broderick (2008) para cada fase citada.

Quadro 6. Fases dos projetos e frequência de identificação por empresa que utiliza

A	B	C	D	E	F	G	H	I	Sliger e Broderick (2008)
			Definição do produto e da estratégia de negócio						
Visão do produto	Iniciação	Preparação	Visão do produto	Visão do produto	Iniciação	Planning inicial	Iniciação	Reunião de <i>planning</i>	Planejamento do projeto:

									visão e roadmap do produto
				<i>kickoff</i>			<i>kickoff</i>		
	Planejamento da <i>release</i>							Planejamento da <i>release</i>	Planejamento da <i>release</i>
planejamento da iteração	planejamento da iteração	planejamento da iteração	Planejamento da sprint	Planejamento da iteração	Planejamento da iteração	Sprints <i>planning</i>	Planejamento da Sprint	Planejamento da iteração	Planejamento da Iteração
Iterações	Iterações	Sprints	Sprints	Iterações	Iterações	Sprints	Iterações	Iterações	Iterações
Reunião diária	Reunião diária	Reunião diária	Reuniões diárias	Reuniões diárias	Reuniões diárias	Reuniões diárias	Reuniões diárias	Reunião diária	Daily Stand-up
Reunião semanal							Reunião semanal	Reunião semanal	
Reunião de review	Reunião de entrega do produto da iteração	Review da Sprint	Review da sprint	Reunião de <i>review</i>	Reunião de entrega do produto da iteração	<i>Review</i> da Sprint	Reunião de <i>review</i>	Reunião de entrega do produto da iteração	Demonst ração e revisão da iteração
	retrospe ctiva da iteração		Retrospe ctiva da Sprint	Retrospe ctiva		Retrospe ctiva		Finalizaç ão da iteração	Retrospe ctiva da iteração
Reunião mensal							Reunião mensal		
							Reunião trimestra l ou de finalizaçã o da release		
							Homolog ação		
	Reunião de avaliação da <i>release</i>							Finalizaç ão da release	
	Finalizaç ão	Fechame nto		Fechame nto			Finalizaç ão		Retrospe ctiva do projeto

Fonte: Elaborado pelo autor com base nos dados da pesquisa de campo.

Apenas a empresa D relatou sobre a fase que antecede o desenvolvimento do software, onde a diretoria e departamento comercial da empresa definem qual o produto a ser desenvolvido e qual será sua estratégia comercial. Após definido qual será o novo produto a ser desenvolvido, em todas as empresas entrevistadas ocorre uma reunião chamada Visão de

produto ou Iniciação na qual a equipe de desenvolvimento contribui com esclarecimentos sobre como será a construção deste novo produto inclusive dando sugestões de melhoria na construção do produto. As empresas B e D relataram que na fase de Visão do Produto definem a MVP⁸ (*Minimum Viable Product*) que consideram ser esta uma boa estratégia para entregar valor ao cliente, o qual deseja obter rapidamente retorno sobre o investimento no software que está sendo desenvolvido. Outra constatação é que ao concluir os projetos algumas empresas não fazem finalização do projeto justificando que no final de cada iteração o produto sempre está em fase de entrega, ou seja, está em estado de ‘pronto’⁹ a ser entregue ao cliente.

Após identificar quais são as fases típicas de um projeto ágil conforme adotado pelas empresas partiu-se para investigar quais são as funções envolvidas em cada fase do projeto e quais as suas responsabilidades nos projetos, conforme apresentado no próximo item.

3.4 FUNÇÕES DOS ENVOLVIDOS NO PROJETO

As respostas dos entrevistados sobre as funções envolvidas nos projeto de software que utilizam métodos ágeis estão apresentado no Apêndice 3. O Quadro 7 apresenta a síntese das funções mencionadas pelos entrevistados, podendo-se observar que existe uma semelhança entre as várias respostas sendo que todas as empresas possuem um time de desenvolvimento, um líder da equipe ou *scrum master* e um representante do cliente. Algumas empresas possuem também o papel do gerente de projeto (empresas A, B, C, E, H e D). As empresas A, C e H justificam a necessidade deste profissional pela exigência do CMMI ou MPS.BR, a empresa B justifica a necessidade devido ao porte da empresa e tamanho dos projetos conforme depoimento do diretor da empresa B descrito abaixo. Porém nas empresas E, H e I os profissionais que exercem esta função também desempenham outros papéis dentro da organização.

⁸ Um MVP (*Minimum Viable Product*) é conjunto mínimo de funcionalidades que permite uma ação e aprendizado sobre os clientes ou usuários (MOREIRA, 2013).

⁹ A ‘definição de pronto’ é usado para assegurar que o trabalho está completado no incremento do produto. O propósito de cada Sprint é entregar incrementos de funcionalidades potencialmente utilizáveis que aderem à definição de ‘Pronto’ (SCHWABER e SUTHERLAND, 2013).

“Em projetos grandes é necessário ter também um gerente de projetos. Este profissional será responsável pelo controle dos recursos, controle dos riscos, e por monitorar e controlar o seu desempenho (Diretor da empresa B).”

O entrevistado da empresa F relatou a importância do líder da equipe, conforme declaração:

“É importante que os projetos sejam efetivamente liderados, que tenham um líder, pois o resultado do projeto não pode depender apenas do controle do time de desenvolvimento. Ele repassa a visão e os desafios, e os desenvolvedores devem achar o melhor caminho para alcançar aquele desafio (Diretor da empresa F)”.

Apenas as empresas C, D, G, H e I citaram como envolvido nos projetos a direção ou área comercial da empresa, mesmo alguns exercendo função de direção, percebe-se que isso se deve pelo fato deles estarem envolvidos diretamente com questões da execução do projeto, o que aproxima do dia a dia da organização, e ao responder as perguntas se colocavam na posição de membros da equipe do projeto.

Quadro 7: Funções envolvidas nos projetos por empresa conforme declarações da pesquisa de campo

A	B	C	D	E	F	G	H	I
PO interno	Business analytics	PO	PO	Representante do cliente	Cliente	Cliente/PO	PO	Cliente ou representante do cliente (PO)
BPO (Business PO ou PO da área de negócio)			BO (Business Owner)					
Time de desenvolvimento	Time de desenvolvimento	Time de desenvolvimento	Time de desenvolvimento	Time de desenvolvimento	Time de desenvolvimento	Time de desenvolvimento	Equipe de desenvolvimento	Time de desenvolvimento
Scrum master	Líder da equipe	Scrum Master	Scrum master	Scrum master	Líder da equipe	Gerente ágil de projetos	Scrum master	Gestor do projeto
							Scrum dos Scrum	
							Gerente	

							de medição	
Gerente de projetos	Gerente de projetos	Gerente de projetos		Gerente de projetos			Gerente de projetos	
		Diretor	Diretoria			Diretor de estratégia	Diretoria e área comercial	Diretoria

Fonte: Elaborado pelo autor com base nos dados da pesquisa de campo.

Referente às atividades desempenhadas por cada função envolvida, observou-se um alinhamento das respostas, por isso foram sintetizados esses dados no Quadro 8, o qual contém o nome da função envolvida, a descrição do perfil e atividades desempenhadas, conforme respostas dos entrevistados.

Quadro 8: Síntese das atividades desempenhadas pelas funções dos envolvidos nos projetos ágeis

Funções envolvidas	Descrição perfil e das suas atividades desempenhadas
Direção	Possui visão gerencial e não se envolve com questões cotidianas. Preocupa-se com questões financeiras, estratégias e prospecção de novos negócios/clientes
BPO ou BO (<i>Business PO</i> ou <i>Business Owner</i>)	Responsável pela visão estratégica do negócio e do produto. Comunica-se com o PO interno para conseguir alinhar visão de negócio com visão do produto que está sendo desenvolvido.
PO ou representante do cliente	Representa o cliente durante o projeto. Responsável pelo <i>backlog</i> do produto e deseja ser informado caso ocorram riscos ao projeto. Preocupa-se em saber se o time de desenvolvimento fez o que ele precisava (visão do produto conforme objetivos do negócio).
Gerente de projetos	Responsável pelo controle de projeto, monitoramento do projeto através da análise das métricas, controle dos recursos, controle dos riscos objetivos. Possui visão do projeto e dos objetivos organizacionais da empresa. Em projetos ágeis o gerente de projetos deve ter a postura de liderança e não de controle.
Scrum master ou Líder da equipe	Atua junto com o time e busca garantir o sucesso da iteração. Exerce o papel de líder do time de desenvolvimento. Responsável por verificar a demanda, coletar e analisar as métricas de desempenho do projeto. Interage com o gerente de projetos.
Time de desenvolvimento	Responsável pelo desenvolvimento do produto. É composto por programadores, analista de testes, analistas de negócios e outros profissionais técnicos que forem necessários. Preocupa-se com o monitoramento da iteração para entregar as tarefas dentro do prazo estabelecido. Sua preocupação de controle restringe-se a sua meta de trabalho.

Fonte: Elaborado pelo autor com base nos dados da pesquisa de campo.

É possível observar que existem muitos tipos de funções envolvidas num projeto de software ágil os quais são estruturados hierarquicamente dentro das organizações. Além dos membros internos também possui o envolvimento direto do cliente ou seu representante. Apesar de todos terem como objetivo concluir o projeto com sucesso, cada uma das funções envolvidas tem uma participação e um olhar diferente sobre o projeto e cada um possui uma visão deste prisma de acordo com seus responsabilidades e interesses. Essa constatação servirá de premissa para a próxima etapa da pesquisa apresentada no capítulo 5.

3.5 CONSIDERAÇÕES SOBRE O CAPÍTULO 3

Percebe-se que a grande maioria das empresas não possui um critério claro para definição de quais métricas serão utilizadas e, normalmente, a escolha ocorre de forma empírica, baseada na troca de experiência com profissionais da área. Através do agrupamento das métricas por categorias de avaliação e sua frequência de citação foi possível observar que a grande preocupação na gestão dos projetos ágeis está no controle do processo e da qualidade, sendo que o controle da equipe, projeto e organizacional possui um número inferior de métricas e de citações.

Referente às fases típicas dos projetos ágeis percebe-se que as empresas entrevistadas mencionaram fases que correspondem a um projeto ágil com desenvolvimento e planejamento do projeto em fases evolutivas atendendo ao modelo proposto por COHN (2005). Além disso, para cada fase ocorre uma reunião de planejamento inicial e no final da fase é realizada reunião de retrospectiva, onde são discutidos os pontos-chaves sobre o andamento do projeto. As informações coletadas são utilizadas melhoria contínua do processo de desenvolvimento, atendendo as recomendações de Sliger e Broderick (2008). Algumas possuem um monitoramento mais rigoroso do projeto, incluindo também reuniões semanais e mensais para discussão sobre o andamento do projeto, nas quais são relatados os fatos ocorridos e apresentadas as métricas que interessam, àqueles envolvidos naquela fase do projeto.

As funções dos profissionais que atuam num projeto ágil são comuns entre todas as empresas entrevistadas. Interessante observar que apesar das empresas apresentarem funções com responsabilidades semelhantes, elas possuem conjuntos de métricas diferentes. Considerando que a definição de uma métrica também visa, mas não somente, atender às necessidades de medição percebidas pelos envolvidos no projeto, torna-se necessário

investigar quais são os objetivos de medição cumpridos pelas métricas mais frequentemente usadas e se elas correspondem às expectativas dos profissionais envolvidos. Esta questão será tratada no capítulo 4.

Outro aspecto a ser considerado é o fato da pesquisa de campo ser realizada em uma amostra limitada, existindo assim a possibilidade das informações coletadas sobre as métricas e funções dos envolvidos não estarem completas, sendo necessário voltar na literatura e verificar se existe a necessidade de complementação. Por outro lado, observou-se uma falta de clareza ou padronização na forma como as empresas utilizam as métricas, dando margem a uma variedade de formas de aplicação. A literatura também não é profusa nestes detalhes, deixando dúvidas e lacunas a serem preenchidas. Este aprofundamento das observações de campo frente à teoria é apresentado no capítulo 4.

Assim, o capítulo 4 foi elaborado visando buscar resposta para as dúvidas suscitadas pelo estudo de campo e resposta para as questões secundárias de pesquisa propostas inicialmente. Ele apresenta a especificação detalhada de um conjunto de métricas, incluindo suas representações e a proposta de um processo de medição compatível com a abordagem ágil de desenvolvimento de software.

4 MÉTRICAS DE AVALIAÇÃO EM PROJETOS ÁGEIS DE SOFTWARE

Conforme comentado anteriormente, muitos são os desafios relacionados à adoção de métricas de software para projetos que utilizam métodos ágeis, entre eles, dificuldade de definir um processo de medição compatível com os projetos ágeis e a dificuldade na definição de métricas adequadas às necessidades específicas projetos ágeis. Com o objetivo de contribuir para resolver estes desafios, esta tese apresenta 2 produtos, descritos neste capítulo:

- um conjunto de métricas e suas representações para projetos ágeis de software que auxiliam na gestão de projetos;
- proposta de processo de medição, compatível com a abordagem ágil de desenvolvimento de software, visando apoiar as empresas de software que adotam métodos ágeis na definição de métricas adequadas para suas necessidade.

Este capítulo está estruturado da seguinte forma, na sessão 4.1 é apresentado o conjunto de métricas. Na sessão 4.2 são realizadas análises das representações referentes aos conjunto de métricas. Na sessão 4.3 é apresentada a proposta do processo de medição para métodos ágeis. Na sessão 4.4 é realizada a avaliação da abordagem proposta e apresentados os resultados obtidos. Finalmente, na sessão 4.5, são apresentadas as considerações finais do capítulo.

4.1 CONJUNTO DE MÉTRICAS

O conjunto detalhado de métricas foi considerado a partir da revisão da literatura apresentada no capítulo 2 e da pesquisa de campo apresentada no capítulo 3. Ele é composto de 32 métricas para controle de projetos ágeis de software, as quais são especificadas quanto à sua definição e forma de uso. Algumas nomenclaturas das métricas foram ajustadas conforme se julgou necessário, sendo que a relação entre as nomenclaturas está apresentada no Apêndice 9.

Durante a análise e compilação das métricas observou-se que algumas delas caracterizavam-se como medida, porém neste texto optou-se por utilizar apenas o termo métrica. Incluíram-se neste grupo representações gráficas de medidas ou métricas, também foram denominadas indistintamente no texto como ‘métricas’. Observou-se que algumas

medidas já estavam sendo contempladas no cálculo de outras métricas citadas nesta pesquisa, sendo desnecessário detalhar individualmente, são estas:

- fluxo das tarefas no quadro kanban que é apresentada na forma gráfica da métrica através do diagrama de fluxo cumulativo; e
- frequência das *releases* que está contemplada em *lead time*, pois ambas tratam sobre o tempo de entrega de funcionalidades ao cliente;
- gargalo que está contemplada no diagrama de fluxo cumulativo, através da análise das informações do gráfico;
- número de casos de testes que é uma medida base para cálculo da métrica índice de defeitos encontrados na fase de testes; total de esforço estimado que é uma medida de base para construção do gráfico *burnup*.

Observou-se que a métrica Retorno sobre o investimento não está enquadrada nos níveis *release*, iteração ou diário, e sim pertence ao nível organizacional, por isso foi removida. Além destes critérios de exclusão, na delimitação desta pesquisa restringiu-se o estudo às métricas para controle nos níveis da *release*, da iteração e diário de projetos ágeis de software. A métrica Número de horas efetivamente trabalhadas em atividade de desenvolvimento também foi removida por exigir o controle individual dos membros da equipe, desta forma prejudicando o princípio ágil de equipes auto-organizáveis e consequentemente podendo prejudicar a motivação dos indivíduos. Desta forma o conjunto foi reduzido de 40 para 32 métricas. Entende-se como limitação da pesquisa a possibilidade de existir métricas que não foram citadas neste conjunto e que são utilizadas por empresas que utilizam métodos ágeis. A Figura 5 apresenta o número de métricas identificadas por fase de pesquisa desta tese.

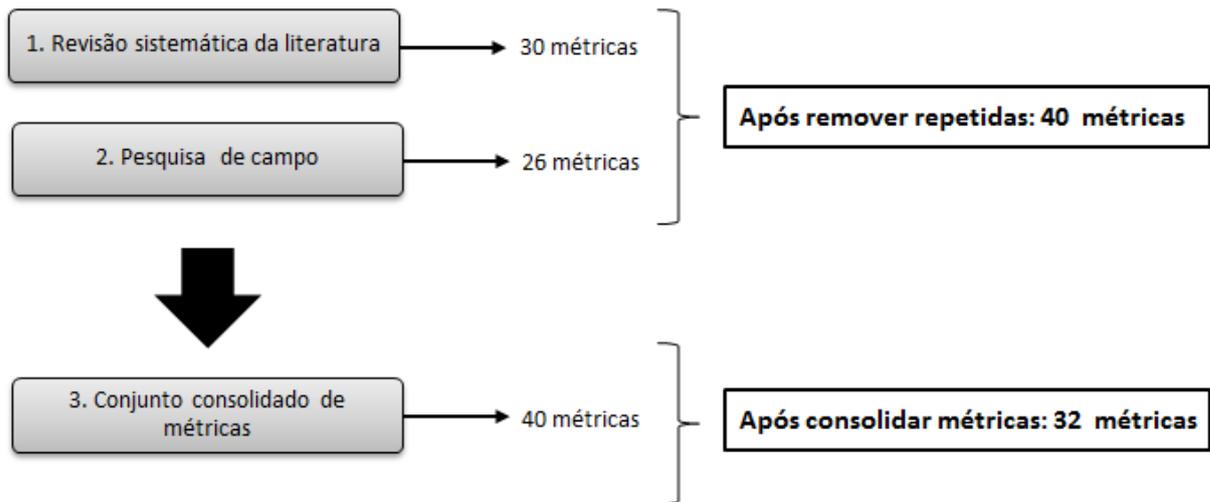


Figura 5. Número de métricas por fase da pesquisa

Fonte: Elaborado pelo autor.

Observou-se nos estudos sobre métricas que os objetivos de medição que tais métricas atendem não são claros. Por este motivo realizou-se o processo inverso de abstração dos objetivos a partir da análise da contextualização das métricas, utilizando-se o seguinte questionamento:

Quais são os objetivos de medição que são atendidos por esta métrica?

O conjunto de métricas é composto por 32 quadros descritivos, um para cada métrica. Para cada métrica haverá, inicialmente, um texto de contextualização da sua utilização, seguida de um quadro que contém as seguintes descrições:

- Nome da métrica: denominação dada à métrica;
- Categoria: serão apresentadas métricas das categorias Equipe (EQP), Qualidade (QUAL), Organizacional (ORG) e Processo (PROC), as quais foram classificadas conforme de avaliação provenientes da revisão bibliográfica sessão 2.2.4;
- Objetivo(s) de medição: benefício proporcionado pela métrica, conforme explicado anteriormente;
- Descrição do cálculo da métrica: descrição matemática da métrica;
- Equação: apresentação da fórmula de cálculo e descrição das suas variáveis;
- Intervalo de medição ou valor alvo: apresentam o valor inferior e superior que a métrica pode assumir ou valor alvo que a empresa define, quando for caso;
- Comportamento da métrica: trata-se do tipo de comportamento assumido para a métrica;

- Unidade de medida: descrição da unidade de medida da métrica;
- Fonte dos dados: origem a partir da qual os dados são coletados;
- Fase do projeto ágil que é analisada: fase do projeto em que a métrica é analisada podendo ser *release*, iteração ou diário;
- A quem interessa a métrica e/ou a analisa: descrição da(s) função(ões) que serão afetadas pela métrica ou irão tomar a decisão sobre o comportamento da métrica;
- Quem coleta os dados: quem operacionaliza a coleta dos dados que irão possibilitar o cálculo da métrica;
- Métrica exclusiva métodos ágeis: descreve se a métrica tem aplicação exclusivamente para projetos de software que utilizam métodos ágeis ou podem ser utilizados também para projetos tradicionais de software.

Para a utilização do conjunto de métricas apresentado a seguir em ordem alfabética, consideram-se os seguintes pressupostos:

- a empresa deve adotar métodos ágeis para controle dos seus projetos com ciclos de desenvolvimento iterativos e incrementais, com níveis de planejamento e controle subdivididos em *releases*, iteração e diário;
- os requisitos do cliente devem ser escritos em histórias e subdivididos em tarefas;
- deve existir um quadro de tarefas (ex. kanban) utilizado para acompanhar a execução das tarefas;
- devem ser realizadas estimativas do tempo das tarefas através de pontos ou horas.

Além desses pressupostos gerais sobre a utilização das métricas, para cada métrica apresentada a seguir, também serão apresentados os pressupostos individuais. A seguir são apresentadas as métricas e seus detalhamentos.

4.1.1 Acurácia das estimativas (tempo)

Acurácia das estimativas (tempo) é uma métrica para avaliar o grau de acerto entre o tempo de duração previsto para a execução das tarefas e o tempo de duração realizado. No início de cada nova iteração a equipe deve refletir sobre o motivo dos erros e acertos em relação às estimativas, e realizar ajustes. A precisão das estimativas é uma métrica que deve

ser discutida nesta reunião, pois interfere na decisão de quantas tarefas a equipe será capaz de realizar numa iteração. Esta reflexão possibilita revelar problemas referentes ao por que as estimativas não foram precisas, impulsionando a empresa a melhorar a sua forma de estimar (JAVDANI et. al, 2012; COHN, 2005). Espera-se que no decorrer das iterações que a equipe consiga ser mais assertiva nas estimativas realizadas, sendo que monitorar as distorções entre o previsto e o realizado será uma importante avaliação para reduzir o erro (COHN, 2005).

Esta métrica é útil para empresas que realizam a estimativa de duração das tarefas em horas, não devendo ser utilizado para controle individual dos membros da equipe, mas sim utilizar esta informação para reflexão e aprendizado sobre o processo de estimar. É pressuposto para a aplicação desta métrica a existência de um quadro de tarefas sendo necessário armazenar o tempo estimado e o tempo realizado para execução da tarefa (ex. quadro kanban). Recomenda-se realizar a análise ao final da iteração, quando a equipe tem acesso ao tempo de duração efetivamente realizado em todas as tarefas.

Para fins de tomada de decisão optou-se transformar a métrica numa expressão percentual, que tem como valor alvo 100% de acurácia. No caso, por exemplo, da acurácia da estimativa da iteração assumir um valor de 106%, significa que a equipe realizou em média 6% mais rápido do que o tempo estimado para executar as tarefas. É apresentada no Quadro 9 a especificação da métrica acurácia das estimativas (tempo).

Quadro 9. Especificação da métrica: Acurácia das estimativas (tempo)

Nome da métrica: Acurácia das estimativas (tempo)	
Categoria	Processo
Objetivo(s) de medição	Identificar distorções entre o tempo de duração previsto para a execução das tarefas e o tempo de duração realizado. Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos.
Descrição do cálculo da métrica	Duas etapas são necessárias para calcular esta métrica: (a) calcular a acurácia da estimativa de cada tarefa, que consiste na razão entre o tempo de duração em horas (h) realizado na execução da tarefa e o tempo de duração em horas (h) estimado para ela; (b) calcular a acurácia das estimativas da iteração que é a média geométrica das acurácias das estimativas das tarefas.

Equação	$(a) \text{ acE_TS}_k = \frac{tR_k}{tE_k}$ <p>Onde:</p> <p>acE_TS_k = Acurácia da estimativa da tarefa k</p> <p>tE_k = Tempo de duração (h) estimado para a tarefa k</p> <p>tR_k = Tempo de duração (h) realizado na execução da tarefa k</p> $(b) \text{ acE}_i = (\sqrt[n]{\text{acE_TS}_1 * \text{acE_TS}_2 * \dots * \text{acE_TS}_n}) * 100$ <p>Onde:</p> <p>acE_i = Acurácia das estimativas da iteração i</p> <p>acE_TS_k = Acurácia da estimativa da tarefa k</p> <p>n = Número de tarefas</p>
Intervalo de medição ou valor alvo	Valor meta igual a 100%
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Percentual de acurácia
Fonte dos dados	Quadro de tarefas e os tempos de duração das tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos e equipe de desenvolvimento
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

No cálculo da iteração utiliza-se a média geométrica, pois penaliza os valores extremos. As análises dos resultados de acurácia das distintas iterações permite que a equipe crie um gráfico, no qual possam visualizar o desempenho da métrica ao longo do tempo, indicando se a acurácia de estimativa está melhorando ou não.

4.1.2 *Burndown* da iteração

Trata-se da representação gráfica da medida de pontos (ou horas) de trabalho ao longo dos dias da iteração. O gráfico *burndown* é utilizado para medir o progresso das tarefas realizadas por uma equipe (TALBY e DUBINSKY, 2009; GUSTAFSSON, 2011). Apresenta uma diferença entre o número de pontos (ou horas) planejados e o número de pontos (ou horas) realizados, até um determinado momento. Neste trabalho, para explicação desta métrica, será adotada a unidade de medida de pontos. A métrica indica se a equipe está concluindo as tarefas mais rápidas ou lentamente do que o esperado. Se a diferença entre o número de pontos estimados e o realizado for grande, isso pode significar que houve um erro de estimativa e tal desvio deve ser monitorado ao longo do projeto. O gráfico *burndown* permite que as equipes possam monitorar seu progresso e visualizar tendências (GUSTAFSSON, 2011) permitindo saber quanto ainda falta ser desenvolvido para concluir a meta da iteração (TALBY e DUBINSKY, 2009; SCHWABER e SUTHERLAND, 2013; SUTHERLAND e SCHWABER, 2011). Também é possível utilizar o gráfico *burndown* para monitorar os pontos estimados e realizados ao longo das *releases* ou projeto (COHN, 2005).

A Figura 6 apresenta um exemplo do gráfico *burndown* onde no eixo horizontal é apresentado o número de dias da iteração e no eixo vertical é apresentado o número de pontos (ou horas) a serem desenvolvidos. O gráfico apresenta duas linhas uma contínua que representa o desempenho ideal, sendo que no final não irá restar nenhum trabalho a ser concluído; e outra linha que apresenta o desempenho real da equipe ao longo da iteração. A linha contínua do gráfico inicia com o número máximo de pontos previstos para a iteração e mostra um decréscimo regular desses pontos ao longo da iteração. A cada dia a equipe deve subtrair da linha que representa o desempenho real (total de pontos restantes) o número de pontos realizados no dia anterior, demonstrando o quanto ainda resta para atingir a meta, definida no início da iteração. Por esta ideia de subtração dos pontos (citada anteriormente) é utilizado o termo de número de pontos ‘queimados’ naquele dia da iteração. Espera-se que a equipe seja capaz de manter um ritmo constante de trabalho ao longo da iteração, realizando um mesmo número de pontos por dia, ou com variação mínima. Esta métrica explicita o princípio ágil que as equipes devem manter indefinidamente passos constantes.

Para construção deste gráfico o usuário deverá atender aos seguintes pressupostos: (a) a equipe deverá prever o número de pontos que, em conjunto, ela irá se comprometer em realizar diariamente. Este número de pontos é exatamente o valor que será subtraído

diariamente do total de pontos pendentes do dia anterior, durante a fase de construção da estimativa; (b) a equipe deverá prever no início da iteração qual a quantidade de trabalho que juntos irão se comprometer em entregar dentro do prazo da iteração; (c) é necessário de um quadro de acompanhamento diário de tarefas. Serão usadas no cálculo desta métrica, as tarefas que tiverem o status ‘concluída’ e, para fim de cálculo das tarefas de um determinado dia da iteração, consideram-se as tarefas que foram concluídas naquele dia.

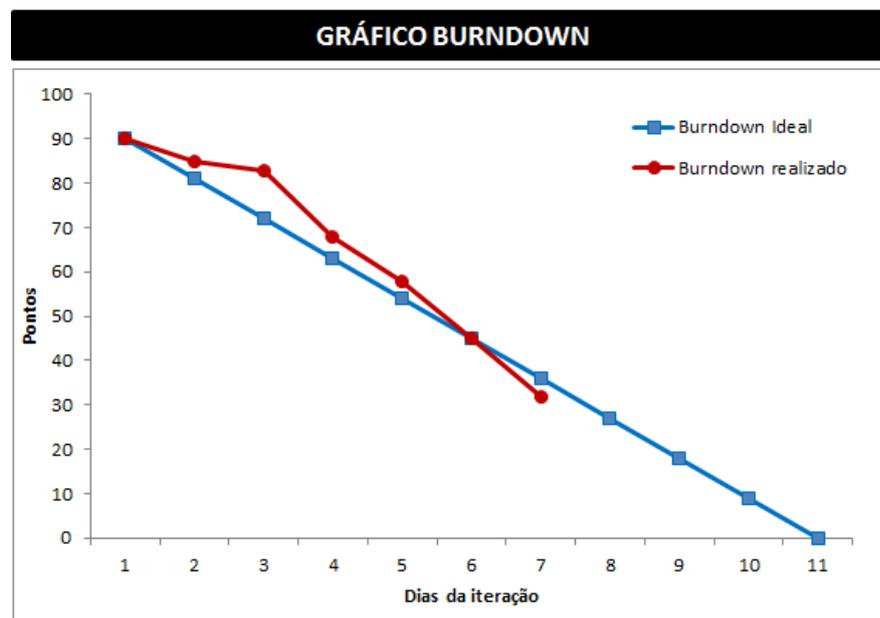


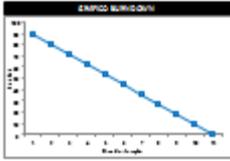
Figura 6. Exemplo de gráfico *burndown* da iteração

Fonte: Adaptado de Cohn(2005).

É apresentada no Quadro 10 a especificação da métrica *burndown* da iteração.

Quadro 10. Especificação da métrica: *Burndown* da iteração

Nome da métrica: <i>Burndown</i> da iteração	
Categoria	Processo
Objetivo(s) de medição	Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos. Visualizar o progresso do projeto.
Descrição do cálculo da métrica	Três etapas são necessárias para calcular esta métrica: (a) construir a linha de estimativa do gráfico que apresenta a quantidade de pontos planejados por dia da iteração; (b) somar os pontos das tarefas concluídas no dia d; (c) subtrair do total de pontos realizados no dia d-1 o total pontos realizados no dia d.

<p>Equação</p>	<div style="text-align: center;">  </div> <p>(a) Gráfico com linha estimativa de pontos/dia</p> <p>(b) $TT_PTS_R_d = \sum_{k=1}^n PTS_R_{d,k}$</p> <p>Onde:</p> <p>$n$ = Número de tarefas $PTS_R_{d,k}$ = Número de pontos realizados na tarefa k do dia d $TT_PTS_R_d$ = Total de pontos realizados no dia d</p> <p>(c) $Bw_R_{i,d} = (TT_PTS_R_{i,d-1} - TT_PTS_R_{i,d})$</p> <p>Onde:</p> <p>$Bw_R_{i,d}$ = Ponto da linha de desempenho real no gráfico <i>burndown</i> que apresenta os pontos do dia d na iteração i</p>
<p>Intervalo de medição ou valor alvo</p>	<p>Definido pela empresa</p>
<p>Comportamento da métrica</p>	<p>Alvo-é-melhor (número de pontos realizado por dia seja constante conforme planejado)</p>
<p>Unidade de medida</p>	<p>Pontos</p>
<p>Fonte dos dados</p>	<p>Quadro de tarefas</p>
<p>Fase do projeto ágil que é analisada</p>	<p>Iteração</p>
<p>A quem interessa a métrica e/ou analisa</p>	<p>Líder da equipe, gerente de projetos e equipe de desenvolvimento</p>
<p>Quem coleta dos dados</p>	<p>Equipe de desenvolvimento</p>
<p>Métrica exclusiva métodos ágeis?</p>	<p>Sim</p>

Fonte: Elaborado pelo autor.

4.1.3 *Burnup da release/projeto*

O gráfico *burnup* é formado por duas linhas. Uma linha apresenta a informação da soma total de trabalho realizado até o momento, normalmente medido em pontos. A outra

linha apresenta a informação da meta da *release* ou do projeto. A diferença entre as duas linhas permite a equipe identificar o quanto de trabalho ainda precisa a ser feito para concluir a meta da *release* ou do projeto. Desta forma é um gráfico útil nas reuniões de planejamento da iteração e da *release* e deverá ser atualizado ao final de cada iteração (JAVDANI et. al 2013, SABBAGH, 2013). Na Figura 7 é apresentado um exemplo do gráfico *burnup*.

Para construção deste gráfico deverá haver os seguintes pressupostos: (a) a equipe possui uma previsão inicial de total do trabalho a ser realizado na *release/projeto*; (b) a equipe deve possuir registros do número de pontos (trabalho) realizado por iteração.

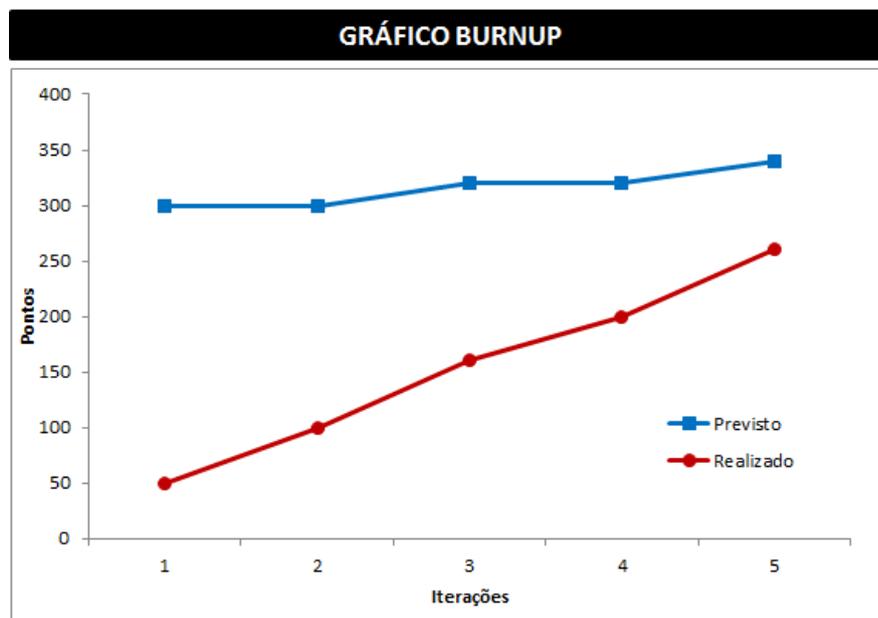


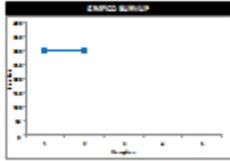
Figura 7. Exemplo de gráfico *burnup* da *release/projeto*

Fonte: Adaptado de Sabbagh (2013).

No Quadro 11 é apresenta a especificação da métrica *burnup* da *release/projeto*.

Quadro 11. Especificação da métrica: *burnup* da *release/projeto*

Nome da métrica: <i>Burnup</i> da <i>release/projeto</i>	
Categoria	Processo
Objetivo(s) de medição	Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos. Visualizar o progresso do projeto.
Descrição do cálculo da métrica	Duas etapas são necessárias para calcular esta métrica: (a) inserir no gráfico o dado que representa o total de pontos estimados para a <i>release</i> ou projeto; (b) somar os pontos realizados nas iterações; repetir as etapas a e b sucessivamente até a última iteração

Equação	 <p>(a) Gráfico do total de pontos estimados na <i>release</i> ou projeto</p> $(b) Bn_i = \sum_{i=1}^n TT_PTS_R_i$ <p>Onde:</p> <p>Bn_i = Ponto da linha do gráfico <i>burnup</i> que representa o trabalho realizado até a iteração i</p> <p>n = Número de iterações</p> <p>$TT_PTS_R_i$ = Total de pontos realizados na iteração i</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor (número de pontos realizado por iteração seja constante)
Unidade de medida	Pontos
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	<i>Release</i>
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos e equipe de desenvolvimento
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

A diferença entre o gráfico *burndown* e gráfico *burnup* é o que *burnup* é utilizado para prever a quantidade de escopo que a equipe poderá ser capaz de realizar num projeto e o *burndown* fornece a informação do quanto resta para atingir a meta de trabalho estabelecido naquela iteração, por este motivo a linha do *burnup* é crescente e não decrescente. Outra diferença é que na iteração existe uma velocidade alvo (meta de pontos) e um escopo claramente definido, já no nível da *release* não é conhecido o escopo total, além do fato de no decorrer do projeto haver a necessidade se adaptar as mudanças, caso seja necessário. Caso ocorram alterações é possível visualizar esta informação através do gráfico. A interpretação

do gráfico *burnup* permite à equipe saber o quanto de trabalho eles conseguem produzir ao longo do tempo (MOREIRA, 2013).

4.1.4 Cobertura de testes automatizados

A atividade de teste de software é uma atividade dinâmica e tem como objetivo demonstrar que o software funcionou conforme previsto (DELAMARO, MALDONADO e JINO, 2007), sendo considerado um fator decisivo para garantir a qualidade de um programa (PRESSMAN, 2006). Os testes podem ser realizados de forma manual ou podem ser realizados de forma automatizada. Testes automatizados são programas ou *scripts* simples que exercitam funcionalidades do sistema testando e que fazem verificações automáticas em busca de erros nos programas desenvolvidos. A grande vantagem desta abordagem é que rapidamente todos os casos de teste podem ser repetidos, com pouco esforço e executados sem a intervenção humana (BERNARDO e KONM, 2008). Os casos de testes são conjuntos de instruções utilizadas para testar um software onde são específicas as entradas que deverão ser fornecidas durante os testes, as ações a serem feitas e o resultado esperado. Se os resultados obtidos coincidem com os resultados esperados, então nenhum defeito foi identificado e pode ser afirmado que ‘o software passou no teste’ (SOMMERVILLE, 2011, PRESSMAN, 2011).

A cobertura de testes automatizados é uma métrica que busca comparar o quanto de testes automatizados foram realizados em relação ao número total de casos de testes (SWAMINATHAN e JAIN, 2012; TALBY e DUBINSKY 2009; DUBINSKY et al. 2005). Esta métrica é útil quando se deseja aumentar o percentual de cobertura de testes automatizados no sistema (GUSTAFSSON, 2011).

Para aplicação desta métrica deverá haver os seguintes pressupostos: (a) a equipe possui a prática de escrever casos de testes para validar os seus programas; (b) a equipe possui a prática de desenvolver testes automatizados para automatizar os seus casos de testes. No Quadro 12 é apresentada a especificação da métrica cobertura de testes automatizados.

Quadro 12. Especificação da métrica: Cobertura de testes automatizados

Nome da métrica: Cobertura de testes automatizados	
Categoria	Qualidade

Objetivo(s) de medição	Avaliar a qualidade do produto. Avaliar a qualidade do código desenvolvido.
Descrição do cálculo da métrica	Calcular a razão entre o número de testes automatizados na iteração e o número total de casos de testes na iteração, multiplicado por 100
Equação	$COB_TST_Aut_i = \left(\frac{TST_Aut_i}{CS_TST_i} \right) * 100$ <p>Onde:</p> <p>$COB_TST_Aut_i$ = Cobertura de testes automatizados da iteração i</p> <p>CS_TST_i = Número de casos de testes na iteração i</p> <p>TST_Aut_i = Número de testes automatizados TST na iteração i</p>
Intervalo de medição ou valor alvo	0 a 100%
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Percentual
Fonte dos dados	Casos de testes e número de testes automatizados
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.5 *Cycle time* (tempo de ciclo para concluir uma tarefa)

Cycle time é o tempo desde que uma tarefa inicia a ser trabalhada até o momento que esta esteja totalmente finalizada, devendo esta informação ser abstraída do quadro de tarefas kanban. Esta métrica está baseado no princípio *lean* de entregar um produto de trabalho em tempos de ciclos curtos, minimizando o tempo entre receber uma requisição e entregar uma funcionalidade de software conforme requisitada. Esta métrica revela a capacidade do processo de desenvolvimento ajudando a prever a quantidade de trabalho em andamento que será entregue (POPPENDIECK e POPPENDIECK, 2003; BOEG, 2012; KNIBERG e SKARIN, 2009). *Cycle time* pode ser entendido também como o tempo que uma tarefa

permanece como *WIP (work in progress)* (MIDDLETON e JOYCE, 2012), ou como o tempo que a tarefa demorou para atravessar o quadro kanban. A média do *cycle time* num determinado período de tempo permite prever a tendência do número de trabalhos que poderão ser entregues num próximo período de tempo. Para melhor representar as métricas que são extraídas a partir do quadro de tarefas kanban, foi desenvolvida a Figura 8, que mostra o quadro e a representação da métrica *cycle time*. Esta mesma figura pode ser utilizada para entender as métricas *Lead time*, *Work in progress* e *Throughput*, descritas a seguir neste trabalho.

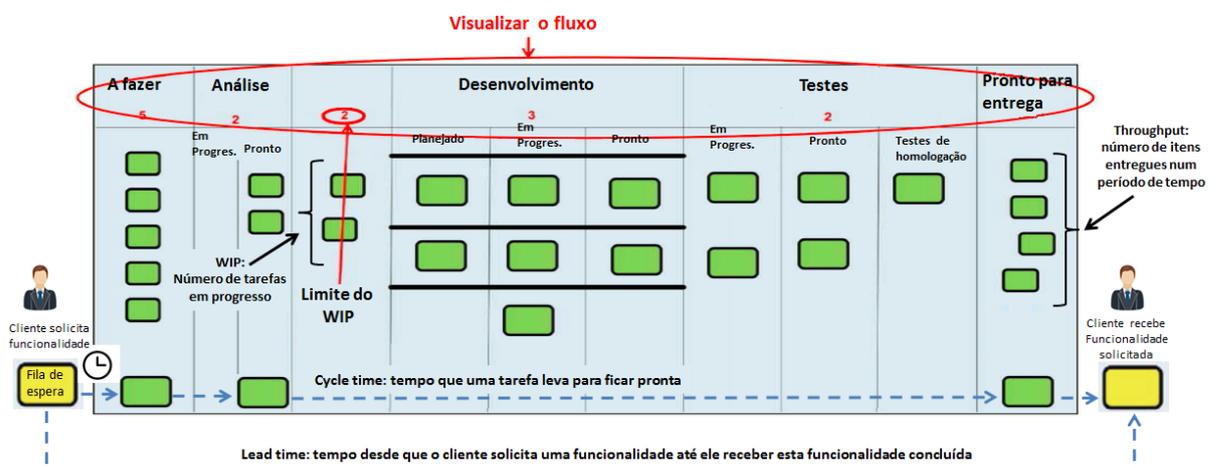


Figura 8. Exemplo de quadro de tarefas kanban

Fonte: Adaptado de Boeg (2012).

É pressuposto para a aplicação desta métrica a existência quadro de acompanhamento diário de tarefas onde é registrado para cada tarefa, o tempo que esta permaneceu em cada fase do quadro, até a sua conclusão final. No Quadro 13 é apresentada a especificação da métrica *cycle time* (tempo de ciclo para concluir uma tarefa).

Quadro 13. Especificação da métrica: *Cycle time* (tempo de ciclo para concluir uma tarefa)

Nome da métrica: <i>Cycle time</i> (tempo de ciclo para concluir uma tarefa)	
Categoria	Processo
Objetivo(s) de medição	Conhecer o ritmo de trabalho da equipe. Prever o tempo de entrega do produto.
Descrição do cálculo da métrica	Deve-se somar o tempo que a tarefa permaneceu em execução em cada fase do processo de desenvolvimento (fases do quadro kanban)

Equação	$c_{time_{TS}} = \sum_{f=1}^n t_{TS_ex_{TS,f}}$ <p>Onde:</p> <p>$c_{time_{TS}}$ = Cycle time da tarefa TS</p> <p>n = Número de fases</p> <p>$t_{TS_ex_{TS,f}}$ = Tempo que a tarefa TS permaneceu em execução na fase f</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Horas
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.6 Defeitos em aberto (defeitos não solucionados)

O número de defeitos em aberto é calculado pelo número de defeitos relatados pelos clientes na iteração, somado aos defeitos antigos ainda não resolvidos. Esta é uma métrica que possibilita interpretar a capacidade da equipe em resolver os defeitos encontrados pelo cliente (MIDDLETON e JOYCE, 2012). É pressuposto para a aplicação desta métrica a realização do monitoramento dos defeitos relatados pelo cliente, bem como o monitoramento da resolução desses defeitos. No Quadro 14 é apresentada a especificação da métrica defeitos em aberto (defeitos não solucionados).

Quadro 14. Especificação da métrica: Defeitos em aberto (defeitos não solucionados)

Nome da métrica: Defeitos em aberto (defeitos não solucionados)	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Deve-se somar o número de defeitos encontrados pelo cliente na iteração e o número de defeitos antigos ainda não resolvidos
Equação	$Df_{ab_i} = Df_{cli_i} + Df_N$ <p>Onde:</p> <p>Df_{ab_i} = Número de defeitos em aberto na iteração i</p> <p>Df_{cli_i} = Número de defeitos encontrado pelo cliente na iteração i</p> <p>Df_N = Número de defeitos antigos ainda não resolvidos</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Número de defeitos
Fonte dos dados	Defeitos relatados pelo cliente e monitoramento da resolução desses defeitos
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.7 Defeitos encontrados pelo cliente

Os defeitos que não são identificados na fase de testes poderão ser identificados pelo cliente quando estiverem efetivamente utilizando o produto de software que foi desenvolvido. A preocupação em controlar esses defeitos deve fazer parte do planejamento de qualidade das empresas. A medida do número de defeitos encontrados pelo cliente é uma informação que permite analisar o nível de qualidade do produto que está sendo entregue para o cliente (GREEN, 2011). É pressuposto para a aplicação desta métrica, a realização do monitoramento

dos defeitos relatados pelo cliente, bem como o monitoramento da resolução desses defeitos. No Quadro 15 é apresentada a especificação da métrica defeitos encontrados pelo cliente.

Quadro 15. Especificação da métrica: Defeitos encontrados pelo cliente

Nome da métrica: Defeitos encontrados pelo cliente	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Deve-se contar o número de defeitos encontrados pelo cliente
Variável	Df_{cli_r} <p>Onde:</p> $Df_{cli_r} = \text{Número de defeitos encontrados pelo cliente na release } r$
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Número de defeitos por <i>release</i>
Fonte dos dados	Defeitos relatados pelo cliente
Fase do projeto ágil que é analisada	Release
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.8 Densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto)

A densidade dos defeitos representa o número de defeitos em relação ao tamanho do produto (PRESSMAN, 2011). O tamanho do produto normalmente é obtido através da contagem do número de linhas de código (KLOC), porém o tamanho do produto também pode ser calculado pelo número de pontos, número de requisitos ou número de histórias, conforme padronização da empresa. Para este trabalho será utilizado para definir o tamanho

do produto a unidade de medida de número total de pontos realizados. Para aplicação desta métrica deverá haver os seguintes pressupostos: (a) a empresa monitora o tamanho dos seus produtos de software através de uma unidade de medida padrão; (b) a empresa monitora os defeitos encontrados ao longo do processo de desenvolvimento. No Quadro 16 é apresentada a especificação da métrica densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto).

Quadro 16. Especificação da métrica: Densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto)

Nome da métrica: Densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto)	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Trata-se da razão entre o número de defeitos encontrados na iteração <i>i</i> e o total de pontos realizados na iteração <i>i</i>
Equação	$dd_Df_i = \frac{Df_i}{PTS_R_i}$ <p>Onde:</p> <p>dd_Df_i = Densidade dos defeitos na iteração <i>i</i></p> <p>Df_i = Número de defeitos encontrados da iteração <i>i</i></p> <p>PTS_R_i = Total de pontos realizados na iteração <i>i</i></p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Defeitos por pontos
Fonte dos dados	Casos de testes
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.9 Diagrama de fluxo cumulativo

Segundo Petersen e Wohlin (2011) o diagrama de fluxo cumulativo apresenta quantas tarefas (ou requisitos) são desenvolvidas ao longo do processo de produção num período de tempo específico. O exemplo do diagrama de fluxo cumulativo é apresentado na Figura 9 onde o eixo x apresenta a linha do tempo e o eixo y apresenta o número cumulativo de tarefas concluídas (requisitos) nas diferentes fases de desenvolvimento. Nesta figura é possível observar que a linha do topo representa o número total de tarefas em desenvolvimento. A primeira linha do gráfico representa o número de tarefas em fase de análise (primeira fase do processo) sendo que quando terminadas serão entregues para a próxima fase de desenvolvimento. A área que está entre duas linhas do gráfico mostra o número de tarefas que estão naquela fase do processo. Olhando para o número de tarefas em diferentes fases, ao longo do tempo, é possível observar o número de tarefas que serão entregues de uma fase para outra. No final do processo encontram-se as tarefas concluídas. Um inventário é definido como o número de tarefas em uma fase em um ponto específico no tempo. Consequentemente, a diferença do número de tarefas (TS) em duas fases (f e $f+1$) representa o nível de inventário atual de um determinado ponto no tempo (t).

Na pesquisa de campo apresentada no capítulo 3, das 9 empresas entrevistadas 89% delas relataram que utilizam a métrica análise do fluxo das tarefas no quadro kanban, apesar de não terem citado o gráfico de fluxo cumulativo, porém entende-se que o gráfico é uma forma de representação visual da métrica, por isso foram consideradas sinônimos para tratar sobre a mesma métrica.

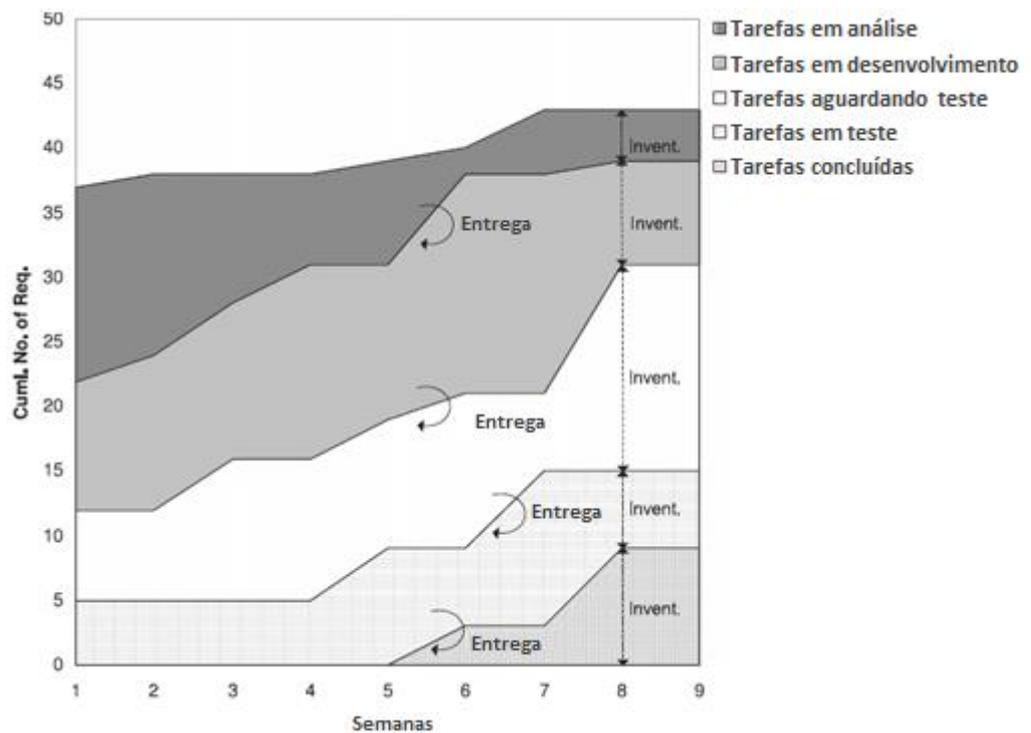


Figura 9. Exemplo diagrama de fluxo cumulativo

Fonte: Petersen e Wohlin (2011).

Para construção deste gráfico deverá haver os seguintes pressupostos: (a) a empresa deve possuir um quadro de monitoramento das tarefas no qual as fases são subdivididas, representando as etapas do processo de desenvolvimento, por exemplo (tarefas em análise, tarefas em desenvolvimento, tarefas aguardando teste, tarefas em teste e tarefas concluídas); (b) deve-se fazer um acompanhamento diário da execução das tarefas, onde é armazenado para cada tarefa o tempo e datas que estas permaneceram em cada fase do quadro até a sua conclusão. No Quadro 17 é apresentada a especificação da métrica diagrama de fluxo cumulativo.

Quadro 17. Especificação da métrica: Diagrama de fluxo cumulativo

Nome da métrica: Diagrama de fluxo cumulativo	
Categoria	Processo
Objetivo(s) de medição	<p>Conhecer o ritmo de trabalho da equipe</p> <p>Identificar acúmulo de trabalhos (gargalo) ao longo do processo de desenvolvimento</p> <p>Identificar o <i>status</i> atual do desenvolvimento das tarefas</p> <p>Visualizar progresso</p>

Descrição do cálculo da métrica	Calcular a diferença entre o número de tarefas (inventário) em uma fase em um ponto específico no tempo e o número de tarefas (inventário) da fase subsequente em um ponto específico no tempo (PETERSEN e WOHLIN, 2011)
Equação	$INV_f = TS_{f,t} - TS_{f+1,t}$ <p>Onde:</p> <p>INV_f = Inventário da fase f</p> <p>$TS_{f,t}$ = Número de tarefas (inventário) na uma fase f em um ponto específico no tempo t</p> <p>$TS_{f,t+1}$ = Número de tarefas (inventário) na uma fase f+1 em um ponto específico no tempo t</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Número de tarefas por fase
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.10 Esforço de administração do projeto (% tempo despendido em atividades de administração do projeto)

A administração do projeto é uma atividade que despende esforço dos seus membros para manter o controle sobre o projeto. A soma das horas consumidas nestas atividades (ex. reuniões de planejamento, reunião de retrospectiva, construção de documentos relacionados à gestão do projeto) permite saber o tempo utilizado para gerir o projeto (KULPA e JOHNSON, 2008). A razão entre quantidade total de horas consumidas em atividades desenvolvidas para gerir o projeto e o número de horas totais de trabalho realizado no projeto, permite saber o esforço despendido para administrar o projeto. É pressuposto para a aplicação desta métrica a

realização do controle das horas de atividades de administração dos projetos, obtendo informação da data e do tempo utilizado da realização de cada tarefa. É apresentada no Quadro 18 a especificação da métrica esforço de administração do projeto (% tempo despendido em atividades de administração do projeto).

Quadro 18. Especificação da métrica: Esforço de administração do projeto (% tempo despendido em atividades de administração do projeto)

Nome da métrica: Esforço de administração do projeto (% tempo despendido em atividades de administração do projeto)	
Categoria	Organizacional
Objetivo(s) de medição	Controlar o tempo utilizado em atividades de administração do projeto
Descrição do cálculo da métrica	Deve-se calcular a razão entre o número de horas trabalhadas em atividades administração e o número total de horas trabalhadas no projeto, multiplicado por 100.
Equações	$eAp = \left(\frac{hWAp}{TT_hW} \right) * 100$ <p>Onde:</p> <p>eAp = Esforço na administração do projeto</p> <p>$hWAp$ = Número de horas de trabalho em atividadesde administração do projeto</p> <p>TT_hW = Número de horas trabalhadas no projeto</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Percentual
Fonte dos dados	Controle das horas de trabalho do projeto Controle das tarefas de administração do projeto
Fase do projeto ágil que é analisada	Release
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento e líder da equipe
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.11 Funcionalidades Testadas e Entregues (*Running testing features - RTF*)

RTF representa quantas funcionalidades com qualidade (testadas) são fornecidas ao cliente em cada iteração (MISHRA, BALCIOGL e MISHRA, 2012). Cada uma das funcionalidades desenvolvidas deverá passar por um teste de aceitação¹⁰ do cliente antes de ser entregue (SATO, 2007). É pressuposto para a aplicação desta métrica que sejam realizados testes de aceitação antes de disponibilizar o software ao cliente, bem como o seu resultado de aceitação devem ser monitorados. No Quadro 19 é apresentada a especificação da métrica Funcionalidades Testadas e Entregues.

Quadro 19. Especificação da métrica: Funcionalidades Testadas e Entregues

Nome da métrica: Funcionalidades Testadas e Entregues	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Deve-se realizar a razão entre o número de funcionalidades aceitas e entregues na iteração e o número de funcionalidades para as quais foram realizados testes de aceitação na iteração, multiplicado por 100
Equação	$RTF_i = \frac{F_Ent_i}{F_TST_i} * 100$ <p>Onde:</p> <p>RTF_i = Funcionalidades Testadas e Entregues na iteração i</p> <p>F_Ent_i = Número de funcionalidades aceitas e entregues na iteração i</p> <p>F_TST_i = Número de funcionalidades para as quais foram realizados testes de aceitação na iteração i</p>
Intervalo de medição ou valor alvo	0 a 100%
Comportamento da métrica	Maior-é-melhor
Unidade de medida	Percentual
Fonte dos dados	Testes de aceitação
Fase do projeto ágil que é analisada	Iteração

¹⁰ Testes de aceitação: testes que são realizados pelo cliente antes de sua disponibilização. Tem como objetivo verificar se as funcionalidades desenvolvidas atendem aos requisitos iniciais estabelecidos. Normalmente é realizado num ambiente de trabalho similar ao que os usuários irão utilizar na organização (PRESSMAN, 2011).

A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.12 Grau de atendimento aos requisitos funcionais

Utilizado para identificar o índice que o software alcançou em relação aos objetivos dos requisitos funcionais (LEE e XIA, 2010), obtido através da razão entre o número de requisitos funcionais aceitos e o número total de requisitos funcionais. Nos métodos ágeis os requisitos funcionais são escritos em forma de história de usuário, por isso desse ser considerada essa informação para cálculo da métrica. É pressuposto para a aplicação desta métrica que os requisitos funcionais (histórias) sejam validados antes de sua entrega ao cliente. No Quadro 20 é apresentada a especificação da métrica grau de atendimento aos requisitos funcionais.

Quadro 20. Especificação da métrica: Grau de atendimento aos requisitos funcionais

Nome da métrica:	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Deve-se realizar a razão entre o número de requisitos funcionais (histórias) aceitos pelo cliente na iteração i e o número total de requisitos funcionais (histórias) da iteração i , multiplicado por 100
Equação	$G_{RF_i} = \frac{RF_{ac_i}}{TT_{RF_i}} * 100$ <p>Onde:</p> <p>G_{RF_i} = Grau de atendimento aos requisitos funcionais na iteração i</p> <p>TT_{RF_i} = Número total de requisitos funcionais (histórias) da iteração i</p> <p>RF_{ac_i} = Número de requisitos funcionais (histórias) aceitos pelo cliente na iteração i</p>
Intervalo de medição ou valor alvo	0 e 100%

Comportamento da métrica	Maior-é-melhor
Unidade de medida	Percentual
Fonte dos dados	Requisitos entregues ao cliente
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.13 Índice de defeitos encontrados na fase de testes

Controlar os defeitos encontrados na fase de testes permite avaliar a qualidade do código que está sendo entregue para o processo de teste do sistema (GREEN, 2011; DUBINSKY et al., 2005). O índice de defeitos encontrados na fase de testes é obtido pela razão entre o número de testes identificados com defeitos encontrados e o total de testes realizados. Essa informação poderá ser plotada num gráfico onde será permitido comparar o índice de defeitos encontrados entre as várias iterações do projeto. É pressuposto para a aplicação desta métrica que a empresa possua o registro do número de defeitos encontrados na fase de testes. No Quadro 21 é apresentada a especificação da métrica índice de defeitos encontrados na fase de testes.

Quadro 21. Especificação da métrica: Índice de defeitos encontrados na fase de testes

Nome da métrica: Índice de defeitos encontrados na fase de testes	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do código desenvolvido
Descrição do cálculo da métrica	Razão o número de testes que foram identificados com defeitos e o número de testes realizados na iteração

Equação	$In_Df_i = \frac{TST_Df_i}{TST_i}$ <p>Onde:</p> <p>In_Df_i = Índice de defeitos da iteração i</p> <p>TST_i = Número de testes realizados na iteração i</p> <p>TST_Df_i = Número de testes identificados com defeitos na iteração i</p>
Intervalo de medição ou valor alvo	0 a 1
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Número
Fonte dos dados	Casos de testes
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.14 Índice de histórias por iteração

Segundo Swaminathan e Jain (2012) índice de histórias por iteração é uma métrica que possibilita saber o fluxo de histórias num período de tempo. A comparação entre o índice de histórias entregues na iteração com outras iterações possibilita compreender a variação de produtividade da equipe (SWAMINATHAN e JAIN, 2012). É pressuposto para a aplicação desta métrica a existência de quadro de acompanhamento de tarefas, onde é armazenado para cada tarefa o número de pontos estimados e a data de início e fim da sua execução. É pressuposto para a aplicação desta métrica a realização do controle das histórias previstas e realizadas na iteração. No Quadro 22 é apresentada a especificação da métrica índice de histórias por iteração.

Quadro 22. Especificação da métrica: Índice de histórias por iteração

Nome da métrica: Índice de histórias por iteração	
Categoria	Processo
Objetivo(s) de medição	Conhecer o ritmo de trabalho da equipe. Visualizar o progresso do projeto.
Descrição do cálculo da métrica	Calcula-se através da razão entre o número de histórias entregues na iteração pelo número real de dias trabalhados na iteração Fonte: Swaminathan e Jain (2012).
Equação	$In_H_i = \frac{H_i}{d_i}$ <p>Onde:</p> <p>In_H_i = Índice de histórias por iteração</p> <p>H_i = Número de histórias entregues na iteração i</p> <p>d_i = Número real de dias trabalhados na iteração i</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Histórias por dia
Fonte dos dados	Histórias da iteração
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.15 *Lead time* (tempo de entrega das funcionalidades ao cliente)

Lead time é o tempo que uma funcionalidade (história/requisito) leva desde a sua solicitação até ser entregue ao cliente (MIDDLETON e JOYCE, 2012; POPPENDIECK e POPPENDIECK, 2003). Ao examinar previamente a métrica de *lead time* é possível obter previsibilidade de entrega de uma tarefa. Essas informações podem ser apresentadas num

gráfico a fim de visualizar a variância do tempo de entrega, sendo que num processo confiável a variância deve ser baixa (MIDDLETON e JOYCE, 2012). É pressuposto para a aplicação desta métrica a existência de um quadro de acompanhamento diário de tarefas, onde é armazenado para cada tarefa o tempo que esta permaneceu em cada fase do quadro, até a sua conclusão final. No Quadro 23 é apresentada a especificação da métrica *lead time*.

Quadro 23. Especificação da métrica: *Lead time* (tempo de entrega das funcionalidades ao cliente)

Nome da métrica: <i>Lead time</i> (tempo de entrega das funcionalidades ao cliente)	
Categoria	Processo
Objetivo(s) de medição	Prever o tempo de entrega do produto
Descrição do cálculo da métrica	Realizar a soma entre <i>cycle time</i> , o tempo de espera até entrar em produção e o tempo de espera após ser concluído e entregue ao cliente
Equação	$L_{time_{TS}} = c_{time_{TS}} + t_{eantes_{TS}} + t_{edepois_{TS}}$ <p>Onde:</p> <p>$L_{time_{TS}}$ = <i>Lead time</i> da tarefa TS</p> <p>$c_{time_{TS}}$ = <i>Cycle time</i> da tarefa TS</p> <p>$t_{eantes_{TS}}$ = Tempo de espera da tarefa TS até entrar em produção (antes de iniciar o processo de desenvolvimento)</p> <p>$t_{edepois_{TS}}$ = Tempo de espera após a tarefa TS ser concluída e ser entregue ao cliente (após concluir o processo de desenvolvimento)</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Horas
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.16 Média da motivação dos membros da equipe

Esta é uma métrica que indica a média da motivação ou satisfação (*happiness*) das pessoas que compõem a equipe (LAANTI, SALO e ABRAHAMSSON, 2011; WAN e WANG, 2010). Normalmente é realizada de forma pessoal pelo líder da equipe ou gerente de projetos, porém deve-se quantificar esta informação sendo recomendado usar uma escala ordinal, por exemplo, uma escala likert de 5 pontos, sendo 1 (insatisfeito), 2 (pouco satisfeito), 3 (neutro), 4 (satisfeito) e 5 (muito satisfeito). Pode-se usar também a técnica *Niko-Niko Calendar*, que consiste em colar na parede da sala um calendário com as datas e o nome das pessoas, sendo que no final do dia cada membro da equipe coloca um ‘*emoticon*’ com uma ‘carinha’ que representa a sua motivação. Assim é possível avaliar padrões ou mudança de humor da equipe (SATO et al., 2006; AKINORI, 2006). A Figura 10 contém a representação visual dos ‘*emoticons*’ que poderão ser utilizados para realizar a pesquisa. O Quadro 24 apresenta um exemplo de *Niko-Niko Calendar* que pode ser utilizado para coletar esta informação com os membros da equipe.



Figura 10. ‘*Emoticons*’ utilizados para pesquisa de motivação

Fonte: Adaptado de <http://eujogador-pt.blogspot.com.br/2011/02/satisfacaofaq.html>.

Quadro 24. Exemplo de *Niko-Niko Calendar*

Nome dos membros da equipe	Dias da iteração									
	1	2	3	4	5	6	7	8	9	10
Ana										
João										
Luiza										
Maria										
Pedro										

Fonte: Elaborado pelo autor.

No Quadro 25 é apresentada a especificação da métrica média de motivação dos membros da equipe.

Quadro 25. Especificação da métrica: Média da motivação dos membros da equipe

Nome da métrica: Média da motivação dos membros da equipe	
Categoria	Equipe
Objetivo(s) de medição	Avaliar a motivação da equipe
Descrição do cálculo da métrica	Duas etapas são necessárias para calcular esta métrica: (a) calcular a motivação diária dos membros da equipe através da média da avaliação da motivação dos indivíduos num determinado dia; (b) calcular a média de motivação dos membros da equipe na iteração que é a média da motivação dos indivíduos nos vários dias da iteração.
Equações	$(a)\overline{MOT_d_d} = \frac{\sum_{d=1}^n MOT_m_{d,m}}{Eq}$ <p>Onde:</p> <p>$\overline{MOT_d_d}$ = Média da motivação dos membros da equipe no dia d</p> <p>$MOT_m_{d,m}$ = Avaliação da motivação no dia d pelo membro m (de acordo com uma escala de 5 pontos)</p> <p>d = Número de dias da iteração i</p> <p>Eq = Número de membros na equipe</p> <p>n = Número de dias da iteração i</p> $(b)\overline{MOT_i} = \frac{\sum_{d=1}^n \overline{MOT_d_{i,d}}}{d}$ <p>Onde:</p> <p>$\overline{MOT_i}$ = Média da motivação dos membros da equipe na iteração i</p> <p>$\overline{MOT_d_d}$ = Média da motivação dos membros da equipe no dia d</p> <p>d = Número de dias da iteração i</p>
Intervalo de medição ou valor alvo	1 e 5
Comportamento da métrica	Maior-é-melhor
Unidade de medida	Valor da escala de satisfação
Fonte dos dados	Membros da equipe

Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe e gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento, líder da equipe
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.17 Nível de satisfação do cliente

O nível de satisfação do cliente é uma métrica que diz respeito à satisfação do cliente quanto ao produto desenvolvido (MONTEIRO, 2008). Segundo Gustafsson (2011) uma boa maneira de verificar se está sendo entregue valor ao cliente é perguntando para ele, devendo ser realizado através de pesquisa de satisfação após cada entrega (final da iteração). Recomenda-se utilizar uma escala ordinal, por exemplo, uma escala likert de 5 pontos, sendo 1 (insatisfeito), 2 (pouco satisfeito), 3 (neutro), 4 (satisfeito) e 5 (muito satisfeito). É pressuposto para aplicação desta métrica que exista uma pesquisa de satisfação com o cliente, após a reunião de entrega do produto, que ocorre no final da iteração. É apresentada no Quadro 26 a especificação da métrica nível de satisfação do cliente.

Quadro 26. Especificação da métrica: Nível de satisfação do cliente

Nome da métrica: Nível de satisfação do cliente	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a satisfação do cliente
Descrição do cálculo da métrica	Realizar a razão entre a soma das notas da pesquisa de satisfação nas iterações i e o número de iterações do projeto
Equação	$\bar{S} = \frac{\sum_{i=1}^n S_{-c_i}}{n}$ <p>Onde:</p> <p>\bar{S} = Nível de satisfação do cliente</p> <p>S_{-c_i} = Notas da avaliação de satisfação do cliente na iteração i (de acordo com uma escala de 5 pontos)</p> <p>n = Número de iterações do projeto</p>

Intervalo de medição ou valor alvo	1 a 5
Comportamento da métrica	Maior-é-melhor
Unidade de medida	Nota
Fonte dos dados	Pesquisa de satisfação com o cliente
Fase do projeto ágil que é analisada	Projeto
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos e equipe de desenvolvimento
Quem coleta dos dados	Líder da equipe, gerente de projetos
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.18 Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões)

O número de integrações por dia é medido pelo número de *check-in*¹¹ diários realizados no repositório central de armazenamento do software. Esta medida busca monitorar a prática de integração contínua. O objetivo da equipe deve ser manter um número constante (pulso) de integrações por dia. Essas informações podem ser plotadas num gráfico a fim comparar o número de integração entre os dias de trabalho na iteração (TALBY e DUBINSKY, 2009). Para aplicação desta métrica deverá haver os seguintes pressupostos: (a) a empresa utiliza software de controle de versão que permite realizar atividades de integração (*check-in*); (b) a equipe adota a prática de integração diária nas suas atividades de trabalho. No Quadro 27 é apresentada a especificação da métrica número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões).

¹¹ *Check-in*: Os sistemas de controles de versões buscam monitorar o histórico de alterações de um determinado item de trabalho ao longo do tempo. O comando *check-in* é utilizado para atualizar estes itens no repositório central mantendo seu histórico de alterações (SOMMERVILLE, 2011)

Quadro 27. Especificação da métrica: Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões)

Nome da métrica: Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões)	
Categoria	Processo
Objetivo(s) de medição	Conhecer o ritmo de trabalho da equipe. Visualizar o progresso do projeto.
Descrição do cálculo da métrica	Deve-se contar o número de integrações realizadas por dia de trabalho
Variável	$INT_{d,i}$ Onde: $INT_{d,i}$ = Número de integrações do dia d da iteração i
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Número de integrações por dia
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.19 Número de tarefas não concluídas na iteração

O número de tarefas (ou histórias) não concluídas são aquelas previstas para uma iteração, mas que não puderam ser implementadas, ou ficaram incompletas e que passaram para a próxima iteração. Esta é uma situação indesejada, pois irá comprometer o progresso geral do projeto (MISHRA, BALCIOGL e MISHRA, 2012). É pressuposto para a aplicação desta medida a existência de um quadro de acompanhamento de tarefas, onde é armazenada

para cada tarefa a data de início e de fim da conclusão das tarefas. No Quadro 28 é apresentada a especificação da métrica número de tarefas não concluídas na iteração.

Quadro 28. Especificação da métrica: Número de tarefas não concluídas na iteração

Nome da métrica: Número de tarefas não concluídas na iteração	
Categoria	Processo
Objetivo(s) de medição	Identificar o número de tarefas não concluídas na iteração
Descrição do cálculo da métrica	Deve-se contar o número de tarefas não concluídas na iteração
Variável	TS_{Nc_i} <p>Onde:</p> $TS_{Nc_i} = \text{Número de tarefas não concluídas na iteração } i$
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Número de tarefas
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.20 Severidade média dos defeitos

Segundo Bastos (2007) a severidade média dos defeitos representa o seu potencial impacto para o negócio do cliente, podendo ser classificado com uma escala ordinal, por exemplo, uma escala likert de 3 pontos, sendo 1 (baixo), 2 (grave) e 3 (crítico). Defeitos com severidade crítica têm sérios impactos na operação do negócio e da organização; com severidade grave produz um resultado incorreto ou para de funcionar; e severidade baixa possui menor impacto, não afetando diretamente os usuários do sistema (BASTOS, 2007). É

pressuposto para a aplicação desta métrica que a empresa possua o registro do número de defeitos relatados pelo cliente e realize avaliação da sua severidade. No Quadro 29 é apresentada a especificação da métrica severidade média dos defeitos.

Quadro 29. Especificação da métrica: Severidade média dos defeitos

Nome da métrica: Severidade dos defeitos	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Realizar a razão entre o somatório da avaliação da severidade do defeito e o número de defeitos relatados pelo cliente
Equação	$\overline{Sev_Df} = \frac{\sum_{Df=1}^n Sev_{Df}}{n_Df}$ <p>Onde:</p> <p>$\overline{Sev_Df}$ = Severidade média dos defeitos do projeto</p> <p>Sev_{Df} = Avaliação da severidade do defeito Df (de acordo com uma escala de 3 pontos)</p> <p>n_Df = Número de defeitos Df relatados pelo cliente</p> <p>n = Número de defeitos</p>
Intervalo de medição ou valor alvo	1 e 3
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Nível de severidade dos defeitos
Fonte dos dados	Defeitos relatados pelo cliente e monitoramento da sua severidade
Fase do projeto ágil que é analisada	<i>Release</i> , projeto
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.21 Taxa de acerto na estimativa das tarefas

Uma das questões fundamentais em um projeto de software é saber, antes de executá-lo, quanto de esforço será necessário para concluí-lo, informação que será a base para o planejamento do projeto (PRESSMAN, 2011). Segundo Cohn (2005) a realização de estimativas é fundamental em projetos ágeis, porém é uma tarefa difícil e propensa a erros. A taxa de acerto na estimativa das tarefas permite saber a fração das tarefas que foram entregues no prazo. É pressuposto para a aplicação desta métrica a existência do quadro de acompanhamento de tarefas, onde é armazenado para cada tarefa o tempo previsto para execução das tarefas e o tempo realizado. No Quadro 30 é apresentada a especificação da métrica taxa de acerto na estimativa das tarefas.

Quadro 30. Especificação da métrica: Taxa de acerto na estimativa das tarefas

Nome da métrica: Taxa de acerto na estimativa das tarefas	
Categoria	Processo
Objetivo(s) de medição	Identificar distorções entre o tempo de duração previsto para a execução das tarefas e o tempo de duração realizado. Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos.
Descrição do cálculo da métrica	Razão entre o número de tarefas desenvolvidas no prazo estimado e o número de tarefas da iteração, multiplicado por 100
Equação	$Tx_{TS_i} = \frac{TS_{prazo_i}}{TS_i} * 100$ <p>Onde:</p> <p>Tx_{TS_i} = Taxa de acerto na estimativa das tarefas na iteração i</p> <p>TS_{prazo_i} = Número de tarefas desenvolvidas no prazo estimado na iteração i</p> <p>TS_i = Número de tarefas da iteração i</p>
Intervalo de medição ou valor alvo	0 e 100%
Comportamento da métrica	Maior-é-melhor
Unidade de medida	Percentual
Fonte dos dados	Quadro de tarefas

Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe e gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

Observa-se pelo resultado da pesquisa de campo o alto interesse dos entrevistados em monitorar a sua eficiência em estimar corretamente (67%), as quais monitoram quantas tarefas foram entregues no prazo e fora do prazo.

4.1.22 Taxa de dívida técnica (problemas técnicos a serem resolvidos)

Dívida técnica, também chamada de débito técnico, é uma metáfora que se refere às consequências de desenvolver funcionalidades incompletas, artefatos imaturos ou inadequados no ciclo de vida de desenvolvimento de software, ou seja, com qualidade comprometida; sendo que estes problemas deverão ser resolvidos futuramente (SEAMAN e GUO, 2011). Por exemplo, um código escrito às pressas que tem necessidade de refatoração, consequentemente possui uma dívida técnica que deverá ser paga. É uma métrica importante para justificar a necessidade de se desenvolver software com qualidade. Para aplicação desta métrica deverá haver os seguintes pressupostos: (a) haver um controle das tarefas em dívida técnica; (b) deverá haver um controle das dívidas técnicas que foram pagas, ou seja, dos problemas referentes à dívida técnica que foram resolvidos. No Quadro 31 é apresentada a especificação da métrica Taxa de dívida técnica (problemas técnicos a serem resolvidos).

Quadro 31. Especificação da métrica: Taxa de dívida técnica (problemas técnicos a serem resolvidos)

Nome da métrica: Taxa de dívida técnica (problemas técnicos a serem resolvidos)	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do código desenvolvido
Descrição do cálculo da métrica	Calcular através da razão entre o número total de tarefas com dívida técnica na iteração e o número de tarefas da iteração i , multiplicado por 100

Equação	$Tx_{DvTc_i} = \frac{Dv_{Tc_i}}{TS_i} * 100$ <p>Onde:</p> <p>Tx_{DvTc_i} = Taxa de dívida técnica da iteração i</p> <p>Dv_{Tc_i} = Número de tarefas com dívida técnica na iteração i</p> <p>TS_i = Número de tarefas da iteração i</p>
Intervalo de medição ou valor alvo	0 a 100%
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Percentual
Fonte dos dados	Atividade de controle das tarefas com dívida técnica
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos, equipe de desenvolvimento
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.23 Taxa de retrabalho

Retrabalho é um indicador da qualidade do produto que está sendo entregue ao cliente. É calculado através soma do tempo em horas investidas corrigindo falhas e defeitos encontrados pelo cliente. Se comparado com o número de horas investidas para o desenvolvimento do software é possível obter a taxa de retrabalho em relação ao tempo total do projeto (JAVDANI et al., 2012; MONTEIRO, 2008). É pressuposto para definição desta métrica que a empresa possua o registro do número de defeitos encontrados na fase de testes e o registro do tempo investido para corrigi-los. No Quadro 32 é apresentada a especificação da métrica retrabalho.

Quadro 32. Especificação da métrica: Taxa de retrabalho

Nome da métrica: Taxa de retrabalho	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Deve-se realizar a razão entre o somatório do tempo investido na correção dos vários defeitos e o número total de horas de trabalhadas no projeto
Equação	$tx_ReW = \frac{\sum_{k=1}^n t_DF_{i,k}}{hW_TT}$ <p>Onde:</p> <p>tx_ReW = Taxa de retrabalho</p> <p>hW_TT = Total de horas de trabalhadas no projeto</p> <p>t_Df_k = Tempo investido na correção do defeito k da iteração i</p> <p>n = Número de defeitos</p>
Intervalo de medição ou valor alvo	0 a 1
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Adimensional
Fonte dos dados	Atividade de correções das tarefas com defeito
Fase do projeto ágil que é analisada	Projeto
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.24 Tempo investido em mudanças

Abraçar mudanças é um dos princípios dos métodos ágeis (MANIFESTO ÁGIL, 2001). Esta flexibilidade possibilita ao cliente a oportunidade de melhorar o produto ao longo do projeto, porém o excesso de mudanças pode revelar problemas na definição dos requisitos e gerar excesso de retrabalho. A medição do tempo investido em mudanças dentro de cada

iteração é útil para garantir que ele não fique fora de controle (RICHARD e JANE, 2005; KULPA e JOHNSON, 2008). É pressuposto para a aplicação desta métrica a realização do controle das alterações das tarefas, registrando as tarefas nas quais houve mudanças e o tempo investido em realizar as mudanças. No Quadro 33 é apresentada a especificação da métrica tempo investido em mudanças.

Quadro 33. Especificação da métrica: Tempo investido em mudanças

Nome da métrica: Tempo investido em mudanças	
Categoria	Processo
Objetivo(s) de medição	Controlar as alterações do projeto
Descrição do cálculo da métrica	Deve-se somar o tempo investido nas mudanças durante a iteração
Equação	$t_Mud_i = \sum_{TS=1}^n TS_Mud_{i,TS}$ <p>Onde:</p> <p>t_Mud_i = tempo investido em mudanças na iteração i</p> <p>$TS_Mud_{i,TS}$ = tempo investido com mudanças na tarefa TS na iteração i</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Horas investidas em mudança
Fonte dos dados	Registro das tarefas que tiveram mudanças
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.25 Tempo investido em tarefas não planejadas

Em projetos ágeis de software, no início de cada iteração, a equipe de desenvolvimento se reúne para planejar o próximo ciclo de trabalho. O ideal é serem identificadas todas as tarefas a serem desenvolvidas para entregar as funcionalidades solicitadas pelo cliente. Porém, no decorrer da iteração pode existir a necessidade de incluir itens não planejados os quais precisam ser identificados e monitorados, pois o tempo adicional dessas tarefas não previstas pode impactar no cumprimento da meta estabelecida no início da iteração (SCHWABER e SUTHERLAND, 2013; KNIBERG e SKARIN, 2009). É pressuposto para a aplicação desta métrica a realização do controle das tarefas não planejadas na iteração bem como o tempo investido na realização dessas tarefas. No Quadro 34 é apresentada a especificação da métrica tempo investido em tarefas não planejadas.

Quadro 34. Especificação da métrica: Tempo investido em tarefas não planejadas

Nome da métrica: Tempo investido em tarefas não planejadas	
Categoria	Processo
Objetivo(s) de medição	Identificar o tempo investido devido a problemas de planejamento na iteração
Descrição do cálculo da métrica	Deve somar o tempo investido na execução das tarefas não planejadas na iteração
Equação	$t_{TS_NPl_i} = \sum_{TS=1}^n TS_NPl_{i,TS}$ <p>Onde:</p> <p>$t_{TS_NPl_i}$ = Tempo investido nas tarefas não planejadas na iteração i</p> <p>$TS_NPl_{i,TS}$ = Tempo investido para execução da tarefa TS não planejada na iteração i</p> <p>n = Número de tarefas não planejadas</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Horas
Fonte dos dados	Quadro de tarefas

Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.26 Tempo médio de resolução dos impedimentos

Impedimentos são problemas que bloqueiam o desenvolvimento de uma tarefa ou impedem que a equipe produza software. Todos os impedimentos devem ser registrados de forma visual para a equipe, devendo conter informação de quais itens estão bloqueados e o motivo (SCHWABER e SUTHERLAND, 2013; BOEG, 2012). Na reunião diária deve ser utilizada para identificar e remover os impedimentos que estão atrapalhando o processo de desenvolvimento. A redução da média do tempo de remoção de impedimentos, indica que os problemas estão sendo removidos mais rapidamente (MIDDLETON e JOYCE, 2012). É pressuposto para a aplicação desta métrica a existência de um quadro de acompanhamento de tarefas onde sempre que alguma tarefa entra em impedimento o tempo de resolução do problema é armazenado. No Quadro 35 é apresentada a especificação da métrica tempo médio de resolução dos impedimentos.

Quadro 35. Especificação da métrica: Tempo médio de resolução dos impedimentos

Nome da métrica: Tempo médio de resolução dos impedimentos	
Categoria	Processo
Objetivo(s) de medição	Controlar a rapidez na resolução dos problemas que atrapalham o processo
Descrição do cálculo da métrica	Calcula-se através da razão entre o somatório do tempo que as tarefas permaneceram em impedimento na iteração e o número de tarefas que estiveram em impedimento

Equação	$\overline{t_{resol_Imp}_i} = \frac{\sum_{TS=1}^n t_{TS_Imp_{i,TS}}}{TS_Imp_i}$ <p>Onde:</p> <p>$\overline{t_{resol_Imp}_i}$ = Tempo médio da resolução dos impedimentos na iteração i</p> <p>$t_{TS_Imp_{i,TS}}$ = Tempo que a tarefa TS permaneceu em impedimento na iteração i</p> <p>TS_Imp_i = Número de tarefas que estiveram em impedimento na iteração i</p> <p>n = Número de tarefas em impedimento</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Horas por tarefa
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.27 *Throughput* (número de tarefas entregues por período de tempo)

O *Throughput* mede o número de tarefas entregues por período de tempo, ou seja, mede rendimento da equipe de desenvolvimento; definido também como a produção média num período de tempo (BOEG, 2012; MIDDLETON e JOYCE, 2012). O esforço de redução de variabilidade do número de tarefas entregues, possibilita a garantia de um rendimento igual (ou superior) com menos *Work in Progress* (a métrica WIP é descrita na sessão 4.1.32). O caminho para tornar um processo mais produtivo é assegurando que a quantidade de WIP não esteja acima da capacidade do sistema, desta forma é possível se concentrar na redução do

tempo de ciclo (*cycle time*) e conseqüentemente aumentar o *throughput* (MIDDLETON e JOYCE, 2012).

É pressuposto para a aplicação desta métrica a existência do quadro de acompanhamento diário de tarefas, onde é armazenado para cada tarefa o tempo que permaneceu em cada fase do quadro até a sua conclusão final. No Quadro 36 é apresentada a especificação da métrica *Throughput*.

Quadro 36. Especificação da métrica: *Throughput* (número de tarefas entregues por período de tempo)

Nome da métrica: <i>Throughput</i> (número de tarefas entregues por período de tempo)	
Categoria	Processo
Objetivo(s) de medição	Avaliar a produtividade da equipe. Prever o tempo de entrega do produto.
Descrição do cálculo da métrica	Calcula-se através da razão entre o <i>WIP</i> e o <i>cycle time</i> num período de tempo <i>t</i> . Fonte: Middleton e Joyce (2012).
Equação	$THPUT_t = \frac{WIP_t}{c_time_t}$ <p>Onde:</p> <p>$THPUT_t$ = Produção média num período de tempo <i>t</i> (<i>throughput</i>)</p> <p>WIP_t = <i>WIP</i> (<i>work in process</i>) num período de tempo <i>t</i> (ver métrica <i>WIP</i> na sessão 4.1.33 deste trabalho)</p> <p>$c_time_{TS,f}$ = <i>Cycle time</i> num período de tempo <i>t</i></p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Número de tarefas/hora
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.28 Total de horas trabalhadas no projeto

Middleton e Joyce (2012) definem o tempo de desenvolvimento como sendo o número de dias úteis de trabalho ao longo do projeto, porém na pesquisa de campo apresentada no capítulo 3, a empresa H relata o cálculo desta métrica através da soma das horas trabalhadas no projeto. Optou-se em utilizar a soma das horas trabalhadas devido ser esta uma informação mais significativa, para visualizar qual foi o esforço despendido para concluir o projeto, sendo que o número de dias de trabalho pode ser obtido pela soma dos dias das iterações. Para calcular o total de horas é necessário somar o número de horas das atividades de desenvolvimento e horas consumidas com atividades de administração do projeto, permitindo saber o tempo real de trabalho da iteração. Através do somatório das horas das iterações é possível saber as horas de trabalho das *releases*, somando as horas da release é possível saber o número de horas do projeto, sendo útil para cálculo de custos do projeto. Não deve ser utilizada para controlar individualmente os membros da equipe, desta forma prejudicando o princípio ágil de equipes auto-organizáveis e consequentemente podendo prejudicar a motivação dos indivíduos. É apresentada no quadro 37 a especificação da métrica total de horas trabalhadas no projeto.

Quadro 37. Especificação da métrica: Total de horas trabalhadas no projeto

Nome da métrica: Total de horas trabalhadas no projeto	
Categoria	Processo
Objetivo(s) de medição	Controlar o tempo total de horas trabalhadas no projeto
Descrição do cálculo da métrica	Deve-se somar o número de horas efetivamente trabalhadas com atividades de desenvolvimento e o número de horas consumidas em atividades de administração do projeto nas várias iterações
Equação	$hW_{TT} = \sum_{i=1}^n (hWD_i + hWA_i)$ <p>Onde:</p> <p>hW_{TT} = Total de horas de trabalhadas no projeto</p> <p>hWD_i = Número de horas efetivamente trabalhadas em atividades de desenvolvimento na iteração i</p> <p>$hWAp_i$ = Número de horas trabalhadas com atividades de administração na iteração i</p> <p>n = Número de iterações</p>

Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Horas
Fonte dos dados	Quadro de tarefas Tarefas de administração do projeto
Fase do projeto ágil que é analisada	<i>Release</i>
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento e líder da equipe
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.29 Variação do número de defeitos entre iterações

Apresenta a variação entre o número de defeitos encontrados na iteração atual e o número de defeitos encontrados nas iterações anteriores. A comparação do número de defeitos nas várias iterações permitirá avaliar se houve melhoria de qualidade do produto ao longo do projeto (GREEN, 2011). É pressuposto para a aplicação desta métrica que a empresa possua o registro do número de defeitos encontrados por iteração. No Quadro 38 é apresentada a especificação da métrica variação do número de defeitos entre iterações.

Quadro 38. Especificação da métrica: Variação do número de defeitos entre iterações

Nome da métrica: Variação do número de defeitos entre iterações	
Categoria	Qualidade
Objetivo(s) de medição	Avaliar a qualidade do produto
Descrição do cálculo da métrica	Diferença entre o número de defeitos identificados numa iteração ou release anterior e o número de defeitos identificados durante a iteração ou release atual

Equação	$\Delta Df_i = Df_{i-1,j} - Df_{i,j}$ <p>Onde:</p> <p>ΔDf_i = Variação do número de defeitos identificados durante a iteração i atual e a anterior i-1</p> <p>$DF_{i,j}$ = Número de defeitos j identificados na iteração i</p> <p>$DF_{i-1,j}$ = Número de defeitos j identificados na iteração i-1</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Menor-é-melhor
Unidade de medida	Número de defeitos
Fonte dos dados	Resultado dos defeitos encontrados na fase de testes
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.30 Variação dos custos (entre estimado e realizado)

A variação dos custos é uma métrica relacionada à avaliação do recurso financeiro despendido nas atividades do processo de desenvolvimento de software (MONTEIRO, 2008), sendo que um projeto deve respeitar as restrições de custos previamente acordadas (PMI, 2013). Sliger e Broderick (2008) afirmam que em projetos ágeis existem restrições orçamentárias e de tempo, porém com escopo negociável, sendo assim uma informação importante a ser controlada. A variação do custo pode ser calculada através da diferença entre o custo estimado para o projeto e o custo real do projeto (LEE e XIA, 2010). O custo real do projeto (ou custo real do trabalho realizado) são os custos realmente incorridos e registrados na realização do trabalho executado durante um determinado período de tempo, podendo às vezes ser representado apenas pelo número de horas de mão-de-obra direta (valor horem/hora) ou incluindo também os custos indiretos (ex. infraestrutura de TI, treinamentos, despesas

administrativas) (MONTEIRO, 2008; PMI, 2013). O somatório das horas de trabalho da iteração e dos custos indiretos naquele período permite saber o valor real da iteração. Através do somatório dos custos das iterações é possível saber o custo das *releases* e do projeto. É pressuposto para a aplicação desta métrica que a empresa possui o registro dos custos diretos e indiretos por iteração, sendo que no Quadro 39 a especificação da métrica variação dos custos (entre estimado e realizado).

Quadro 39. Especificação da métrica: Variação dos custos (entre estimado e realizado)

Nome da métrica: Variação dos custos (entre estimado e realizado)	
Categoria	Organizacional
Objetivo(s) de medição	Controlar os recursos financeiros do projeto
Descrição do cálculo da métrica	(a) calcular o custo real através do somatório entre o custo homem/hora de trabalho multiplicado pelo número de horas nos vários dias de trabalho da iteração mais os custos indiretos da iteração; (b) calcular a variação dos custos da iteração através da diferença entre custo estimado na iteração e o custo real da iteração.
Equações	$(a) CR_i = \left[\sum_{d=1}^n (HW_{i,d} * CH) \right] + CI_i$ <p>Onde:</p> <p>CR_i = Custo realizado na iteração i</p> <p>CI_i = Custos indiretos na iteração i</p> <p>CH = Custo médio da hora de trabalho dos membros da equipe (homem/hora)</p> <p>HW_d = Número de horas de trabalho dos colaboradores no dia d na iteração i</p> <p>n = Número de dias da iteração</p> $(b) \Delta C_i = CE_i - CR_i$ <p>Onde:</p> <p>ΔC_i = Variação do custo da iteração i</p> <p>CE_i = Custo estimado na iteração i</p> <p>CR_i = Custo realizado na iteração i</p>
Intervalo de medição ou valor alvo	Definido pela empresa

Comportamento da métrica	Menor-é-melhor
Unidade de medida	Valor monetário
Fonte dos dados	Horas de trabalho e valor dos gastos indiretos
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Líder da equipe, gerente de projetos
Quem coleta dos dados	Líder da equipe, gerente de projetos
Métrica exclusiva métodos ágeis?	Não

Fonte: Elaborado pelo autor.

4.1.31 Velocidade média diária da equipe

A velocidade representa a quantidade de trabalho que uma equipe de desenvolvimento é capaz de entregar numa iteração, podendo ser utilizadas as unidades de medida ‘pontos de história’ ou ‘horas ideais’ por iteração, sendo o ideal ser coletado por pontos (COHN, 2005; HARTMANN e DYMOND, 2005; SATO, 2007). Velocidade é uma medida muito útil no nível de controle do projeto, pois permite prever a quantidade de trabalho que eles poderão entregar baseados em dados das iterações anteriores. Por isso é considerada uma medida de capacidade sendo necessário coletar dados de várias iterações (HARTMANN e DYMOND, 2005).

Conhecendo a velocidade da equipe, é possível fazer a relação entre a quantidade de trabalho e o tempo que ainda falta para ser concluída a iteração (COHN, 2005; JAVDANI et. al, 2012; KNIBERG e SKARIN, 2009; KNIBERG, 2007). A velocidade de entrega normalmente é utilizada para alcançar a previsibilidade de cumprimento do prazo e estimar a capacidade de entrega, porém deve-se cuidar para o controle da velocidade não prejudicar a qualidade (GUSTAFSSON, 2011). Segundo Hartmann e Dymond (2005) não deve ser utilizada para medir a velocidade dos membros individualmente, mas a velocidade da equipe como um todo.

É proposto no Quadro 40 a métrica de velocidade média diária da equipe. Se a empresa desejar saber a velocidade total da equipe numa iteração deve-se multiplicar a velocidade média diária pelo número de dias da iteração. A informação da velocidade média

diária deverá ser utilizada para o planejamento das iterações seguintes (COHN, 2005). É pressuposto para a aplicação desta métrica a existência quadro de acompanhamento de tarefas, onde é armazenado para cada tarefa o número de pontos estimados e a data de início e fim da execução.

Quadro 40. Especificação da métrica: Velocidade média diária da equipe

Nome da métrica: Velocidade média diária da equipe na iteração	
Categoria	Equipe
Objetivo(s) de medição	Conhecer a capacidade de produção da equipe. Conhecer o ritmo de trabalho da equipe.
Descrição do cálculo da métrica	Calcular através razão entre o somatório dos pontos realizados nos vários dias da iteração e o número de dias da iteração
Equação	$\bar{V}_i = \frac{\sum_{j=1}^d PR_{i,j}}{d}$ <p>Onde:</p> <p>\bar{V}_i = velocidade média do time na iteração i</p> <p>d = número de dias da iteração i</p> <p>$PR_{i,j}$ = Número de pontos realizados no dia j da iteração i</p>
Intervalo de medição ou valor alvo	Definido pela empresa
Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Pontos por dia
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe e gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.1.32 Work in Progress (WIP)

WIP (*Work in Progress*) é o número de atividades em andamento no processo produtivo, ou seja, é o termo dado ao trabalho que foi iniciado, mas não foi concluído (GUSTAFSSON, 2011; KNIBERG e SKARIN, 2009). O WIP excessivo pode revelar vários problemas: capital de investimento sem nenhum retorno sobre o investimento, ou seja, dinheiro gasto sem retorno financeiro; esconder gargalos de atividades em processo deixando lento o fluxo geral; e representar risco sob potencial retrabalho, já que poderão ocorrer alterações nesses itens não aceitos. Devido a esses problemas as abordagens ágeis normalmente limitam o WIP. A maneira mais comum de limitar WIP em projeto ágeis é utilizando placas no quadro kanban, restringindo a quantidade de trabalho numa determinada fase do desenvolvimento, o que ajuda a assegurar que os limites não sejam excedidos (GRIFFITHS, 2011).

O monitoramento WIP em cada etapa do processo é um indicador do desempenho e de coordenação da equipe, sendo que um WIP baixo significa que a equipe trabalha muito bem em conjunto (GUSTAFSSON, 2011). No Quadro 41 é apresentada a especificação da métrica work in progress (WIP).

Quadro 41. Especificação da métrica: *Work in Progress* (WIP)

Nome da métrica: <i>Work in Progress</i> (WIP)	
Categoria	Processo
Objetivo(s) de medição	Identificar capacidade de produção
Descrição do cálculo da métrica	Calcular através do somatório do número de tarefas na várias fases do processo
Equação	$WIP_{TT} = \sum_{j=1}^n WIP_{f_j}$ <p>Onde:</p> <p>WIP_{TT} = WIP Total</p> <p>WIP_{f_j} = Número de tarefas na fase j</p> <p>n = Número de fases</p>
Intervalo de medição ou valor alvo	Definido pela empresa

Comportamento da métrica	Alvo-é-melhor
Unidade de medida	Número de tarefas
Fonte dos dados	Quadro de tarefas
Fase do projeto ágil que é analisada	Iteração
A quem interessa a métrica e/ou analisa	Equipe de desenvolvimento, líder da equipe, gerente de projetos
Quem coleta dos dados	Equipe de desenvolvimento
Métrica exclusiva métodos ágeis?	Sim

Fonte: Elaborado pelo autor.

4.2 ANÁLISES REALIZADAS A PARTIR DO CONJUNTO DE MÉTRICAS

Foram definidas formas de categorização das métricas o que permitiu gerar representações deste conjunto que são apresentadas a seguir:

- mapa mental onde é possível visualizar a distribuição das métricas primeiramente por ‘categoria’ (equipe, processo, qualidade e organizacional), na sequência distribuídas por ‘tipo de indicador’ (tendência ou resultado) e, por fim, distribuídas por ‘objetivos de medição’;
- sumarização das métricas por categoria e agrupadas por ‘exclusivas ou não dos métodos ágeis’;
- agrupamento das métricas por ‘fases do processo de desenvolvimento do software’.

4.2.1 Mapa mental

Foi desenvolvido um mapa mental¹² para visualizar graficamente o agrupamento das métricas (disponível no Apêndice 5) o qual consolidou 3 tipos de classificação dessas

¹² Mapa mental é uma técnica gráfica utilizada para representar visualmente fatos e ideias de forma estruturada. Possibilita organizar, classificar e sintetizar informações de forma hierárquica registrando as interligações entre os assuntos (BUZAN, 2005).

métricas: por categoria, tipo de indicador e por objetivo. No mapa mental, primeiramente as métricas foram agrupadas nas categorias equipe, processo, qualidade e organizacional. Observou-se que a maioria das métricas são relacionadas ao processo, seguida das categorias qualidade, organizacional e equipe. Na sequência, buscou-se entender o tipo de decisão a ser tomada a partir da análise das métricas e, baseado nos autores Kaplan e Norton (1996), agrupou-se as métricas de cada categoria em Indicadores de Tendência ou Indicadores de Resultado. Foi utilizada a expressão indicadores de tendência e resultado embora se tratem de métricas, com o objetivo de manter a nomenclatura utilizada pelos autores proponentes. Esta classificação permitiu observar que há uma maior concentração nas métricas de tendência nas métricas de processo, as quais estão associadas ao andamento do projeto. Ou seja, são métricas que seu monitoramento serve para direcionar os próximos passos a serem tomados no projeto sendo possível agir sobre o problema que venha a ocorrer. Este fenômeno se justifica pelo fato dos projetos ágeis ocorrerem de forma emergente, iterativa e exploratória, com ciclos curtos e com escopo de projeto adaptável (NERUR, 2007; MANIFESTO ÁGIL, 2001), o que explica os indicadores de tendência que permitem agir sobre ele. Na categoria de processo também é possível observar que concentram-se a maioria das métricas exclusivas dos métodos ágeis, sendo que várias destinam-se ao monitoramento do progresso das atividades do projeto.

Os indicadores de resultado apresentam-se nas categorias equipe, organizacional e qualidade, sendo estes indicadores de síntese e associados com o sucesso do projeto. Neste contexto, poderiam ser definidas dimensões de análise, e para cada dimensão um conjunto de indicadores.

Através do mapa mental também é possível observar a classificação das métricas por objetivos de medição. Ao analisar os objetivos de medição é possível observar a relação entre as métricas e os objetivos de medição que possuem em comum. Várias métricas possuem os mesmos objetivos, podendo a empresa escolher quais dessas deseja adotar em função da sua necessidade de medição, facilidade de uso ou outro critério de escolha.

A seguir são apresentados em ordem decrescente os objetivos de medição que tiveram maior concentração de métricas:

- a) avaliar a qualidade do produto: 9 métricas;
- b) conhecer o ritmo de trabalho da equipe: 5 métricas;
- c) visualizar o progresso do projeto: 5 métricas;

- d) Observar tendência/previsibilidade do cumprimento das entregas dentro dos prazos: 4 métricas;
- e) avaliar a qualidade do código desenvolvido: 3 métricas;
- f) prever o tempo de entrega do produto: 3 métricas.

Ao analisar os objetivos de medição mais citados observou-se que existe um número maior de métricas para avaliar a qualidade do produto, avaliar avanço do projeto através de conhecer o ritmo de trabalho da equipe, o monitorar o progresso do trabalho na iteração e observar previsibilidade do cumprimento das entregas dentro do prazo previsto, estando aderente ao princípio ágil de satisfazer através de contínuas entregas de software ao cliente.

Não se preocupou em distinguir métricas por tipo de método ágil já que existem empresas que adotam um conjunto de práticas ágeis e não um método específico, e pelo fato de diversas métricas poderem ser utilizadas por qualquer método ágil. As empresas deverão fazer a escolha de quais métricas utilizar levando em consideração as métricas mais aderentes com suas práticas de trabalho, as necessidades de controle dos diversos envolvidos e alinhadas aos objetivos organizacionais da empresa.

Através Tabela 5 a qual sumariza o total de métricas por categorias e se estas são exclusivas ou não dos métodos ágeis, é possível ver que 59% do total das métricas não são exclusivas dos métodos ágeis, ou seja, tiveram origem nos métodos tradicionais, porém são utilizadas para monitorar os projetos ágeis. Essa informação sugere que existem complementariedade entre os métodos e não uma antítese. O maior número de métricas não exclusivas dos métodos ágeis concentram-se nas categorias organizacional e qualidade. Referente às métricas exclusivas dos métodos ágeis, existe uma concentração maior na categoria processo.

Tabela 5. Sumarização do total de métricas por categorias e métricas exclusivas ou não de métodos ágeis

Categoria	Número de métricas	% das métricas	Exclusivo MA	% Exclusivo MA	Não exclusivo MA	% Não exclusivo MA
Equipe	2	6%	1	50%	1	50%
Organizacional	3	9%	0	0%	3	100%
Processo	15	47%	12	80%	3	20%
Qualidade	12	38%	0	0%	12	100%
Total	32	100%	13	41%	19	59%

Fonte: Elaborada pelo autor.

Na próxima sessão é apresentado o agrupamento das métricas por fase de análise.

4.2.2 Agrupamento das métricas por fases

A Figura 11 apresenta as métricas distribuídas por fases do projeto em que são analisadas e suas categorias. A forma de apresentação está baseada no modelo em camadas (*release*, iteração e diário) proposto por Cohn (2005). A fase que concentra o maior número de métricas é a iteração, as quais são analisadas ao longo e ao final da iteração, com predomínio de métricas de processo e qualidade. No nível diário ocorre a coleta dos dados das métricas.

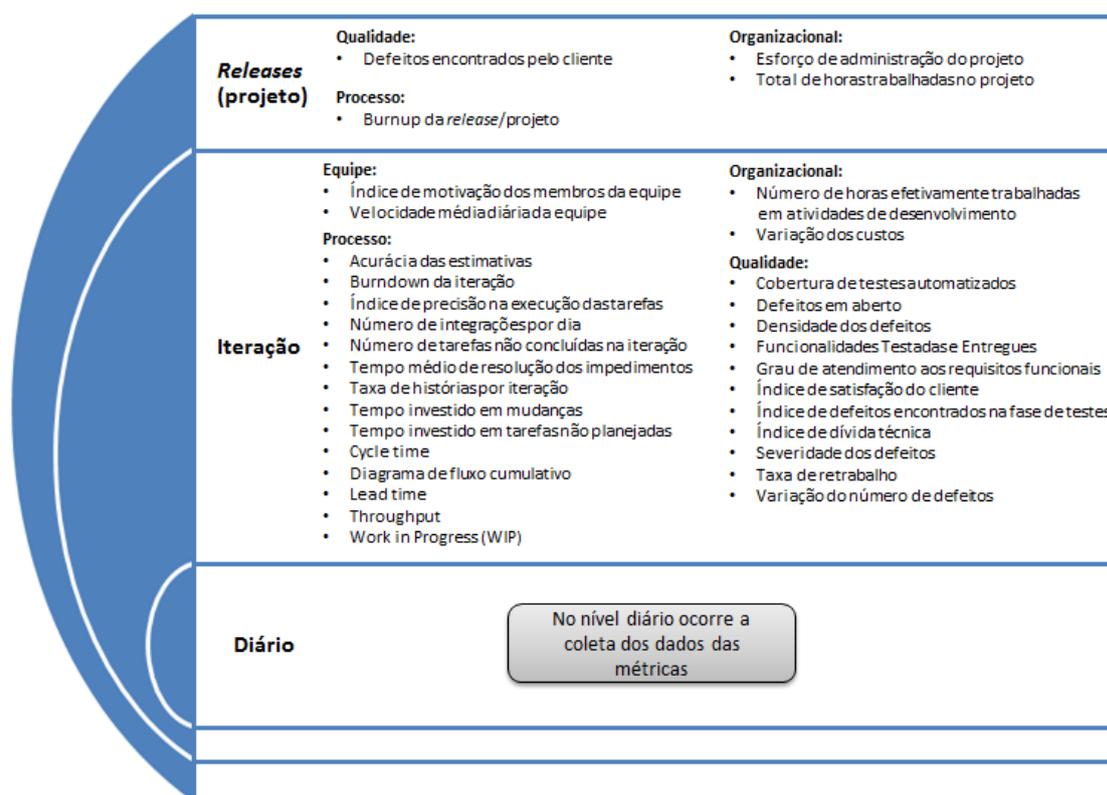


Figura 11. Métricas por fase do projeto de análise

Fonte: Elaborada pelo autor.

Na próxima sessão é apresentado o processo de medição proposto para projetos de software que utilizam métodos ágeis.

4.3 PROPOSTA DE UM PROCESSO DE MEDIÇÃO PARA PROJETOS ÁGEIS DE DESENVOLVIMENTO DE SOFTWARE

Esta sessão apresenta a proposta de um processo de medição compatível com as características dos projetos ágeis. Inicialmente são descritos os requisitos considerados para definição do processo, na sequência é apresentada a estrutura do processo e o detalhamento das fases, o quadro visual de monitoramento e, por fim, é apresentada a avaliação do processo e suas limitações.

Inicialmente foram definidos os seguintes requisitos para a definição do processo de medição, levando em consideração a própria natureza dos métodos ágeis:

- a) deve ser simples;
- b) não deve agregar novas etapas ao processo de desenvolvimento de software;
- c) deve fazer uso da estrutura organizacional existente na empresa o que implica em não criar novos cargos ou funções;
- d) fornecer orientações que permitam às empresas refletir sobre os objetivos de medição e necessidades de controle através de métricas;
- e) auxiliar no monitoramento das métricas ao longo do projeto (novo);
- f) permitir o acesso às informações a todos os interessados de forma rápida e simples.

O processo foi construído através da definição num ciclo lógico de gestão, apresentado através de uma sequência de fases que permita a uma equipe ágil planejar, executar, verificar, atuar, refletir e melhorar a forma de realizar a medição em seus projetos, respeitando as características dos projetos ágeis de desenvolvimento de software; e da proposição de um quadro de monitoramento que permita a gestão do processo de medição de forma visual.

As fases definidas para o processo são:

- Fase 1 – Planejamento de medição e divulgação dos resultados;
- Fase 2 – Coleta de dados, monitoramento da iteração e divulgação dos resultados;
- Fase 3 – Ações sobre a iteração e divulgação dos resultados;
- Fase 4 – Monitoramento do projeto/*releases* e divulgação dos resultados;
- Fase 5 – Ações sobre o projeto/*releases* e divulgação dos resultados;
- Fase 6 – Avaliação final e divulgação dos resultados.

A Figura 12 apresenta a representação gráfica do ciclo lógico de gestão e das fases do processo proposto. A discussão de cada fase do processo e do quadro visual de monitoramento é apresentada nas próximas sessões deste capítulo.

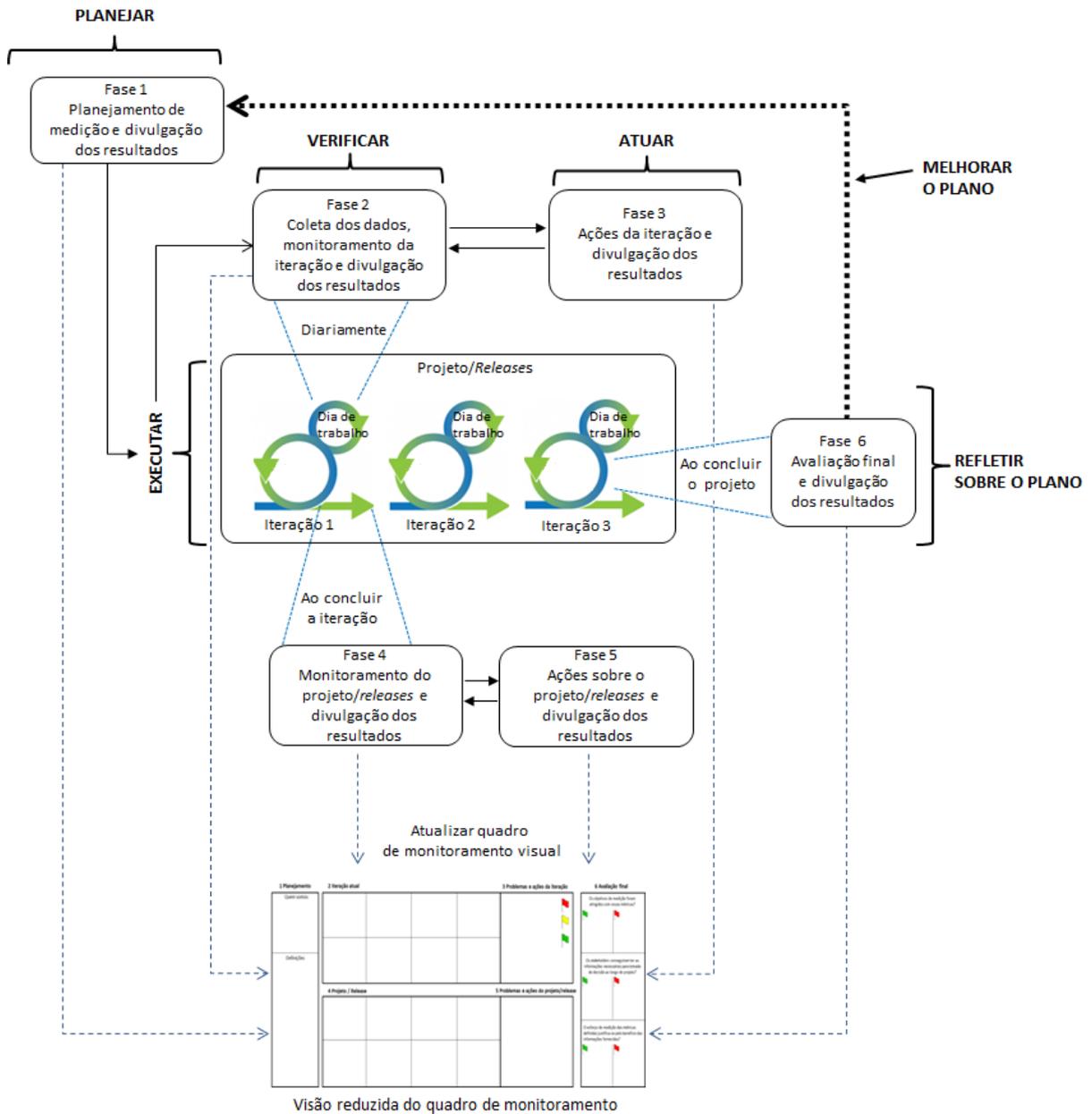


Figura 12. Relação entre as fases do processo e o ciclo lógico de gestão

Fonte: Elaborada pelo autor.

Propõe-se que o monitoramento do projeto ocorra de forma evolutiva em vários níveis e que esteja associado aos eventos já existentes num projeto ágil de software.

Uma vez que o desenvolvimento ágil está associado à incerteza, visto que no início possui um escopo não claramente definido, é necessário que os gestores e os membros da equipe monitorem a evolução do projeto e concomitantemente acertem o seu curso, conforme a visão vai ficando clara e as definições vão sendo estabelecidas. Por este motivo é recomendado ter duas visões de monitoramento das métricas:

- i. a visão global do projeto com valores acumulados, que apresentem a evolução das métricas e que possam permitir visualizar a evolução, tendências e distorções. Neste processo de medição esta visão é proporcionada pela fase 4 - Monitoramento do projeto/*releases* e divulgação dos resultados. Nesta fase, ao realizar a análise das métricas recomenda-se fazer a seguinte pergunta: estamos nos movendo para o caminho certo?
- ii. visão local da iteração atual do projeto para garantir que este ciclo seja entregue conforme planejado. Neste processo de medição esta visão é proporcionada pela fase 2 – Coleta de dados, monitoramento da iteração e divulgação dos resultados. A iteração pode ser considerada uma etapa prescritiva, já que possui um escopo de trabalho claramente definido, tarefas a serem desenvolvidas planejadas, com previsão de tempo estimado e com recursos claramente definidos. Por isso o objetivo dos gestores é conseguirem ser o mais assertivos possível quanto ao planejado e o realizado para a iteração. Nesta fase, ao realizar a análise das métricas recomenda-se fazer a seguinte pergunta: estamos executando as tarefas conforme o planejado?

Na próxima sessão será apresentando o quadro visual de monitoramento do projeto, o qual sintetiza o processo proposto e nas próximas sessões serão descritas as fases do processo.

4.3.1 Quadro visual de monitoramento

Para orientar as equipes de desenvolvimento na utilização do processo e auxiliar as equipes no monitoramento das métricas, foi elaborado um quadro visual que está apresentado nos apêndices 6 e 7. O quadro foi elaborado para possibilitar de forma visual planejar, acompanhar a execução, monitorar problemas e ações corretivas e fazer a avaliação do processo de medição de um projeto ágil. Este pode ser considerado um documento formal do

projeto, porém deve ser elaborado de forma participativa e comunicado entre os membros envolvidos no projeto.

O quadro visual está subdividido em 6 campos cada um relacionado às informações de uma fase específica do processo de medição, sendo a fase 2 destinada à atualização e divulgação aos membros da equipe. A relação entre as fases do processo e o momento em que deverá ser atualizado o quadro de monitoramento está descrito no Quadro 42.

Quadro 42. Relação entre as fases do processo e o momento em que deverá ser atualizado o quadro visual de monitoramento

Campo do quadro	Fase que o quadro visual é atualizado	Descrição do momento que deverá ser atualizar este campo
1 – Planejamento	Fase 1 – Planejamento e divulgação dos resultados	Atualizar o quadro ao concluir a fase 1 (realizar durante reunião de planejamento inicial do projeto)
2 – Iteração atual	Fase 2 – Coleta de dados, monitoramento da iteração e divulgação dos resultados	Atualizar este campo do quadro diariamente (realizar durante a reunião diária) ou conforme periodicidade de coleta estabelecida na fase 1. No final da iteração os dados deste campo 2 serão atualizados no campo 4 (Projeto/ <i>releases</i>). Ao iniciar uma nova iteração o campo 2 deverá estar limpo para receber os dados desta iteração (durante a reunião de retrospectiva da iteração).
3 – Problemas e ações da iteração	Fase 3 – Ações sobre a iteração e divulgação dos resultados	Atualizar este campo quando foram realizadas ações corretivas sobre a iteração. Da mesma forma que o campo 2, ao iniciar uma nova iteração deverá estar limpo para receber os dados desta iteração
4 – Projeto/ <i>Releases</i>	Fase 4 – Monitoramento do projeto/ <i>releases</i> e divulgação dos resultados	Atualizar este campo ao final de cada iteração (realizar durante a reunião de retrospectiva da iteração) com valores acumulados das métricas e/ou gráficos de monitoramento do desempenho das métricas ao longo das iterações
5 – Problemas e ações do projeto/ <i>releases</i>	Fase 5 – Ações sobre o projeto/ <i>releases</i> e divulgação dos resultados	Atualizar este campo quando foram realizadas ações corretivas sobre o projeto/ <i>releases</i>
6 – Avaliação final	Fase 6 – Avaliação final e divulgação dos resultados	Atualizar este campo no final do projeto (realizar durante reunião de finalização do projeto)

Fonte: Elaborado pelo autor.

É sugerido utilizar sinais visuais (indicadores) para controle das métricas ou das ações a serem tomadas. Estes sinais se apresentam em forma de bandeiras com cores específicas e quando se apresenta na cor verde indica que a situação está sob controle, quando se apresenta na cor amarela significa que alguma ação deve ser tomada, e quando a bandeira encontra-se na cor vermelha significa um sinal de alerta indicando que a situação está fora do controle e o projeto pode estar de alguma forma comprometido. A Figura 13 apresenta os sinais visuais utilizados pelo quadro visual de monitoramento do projeto.

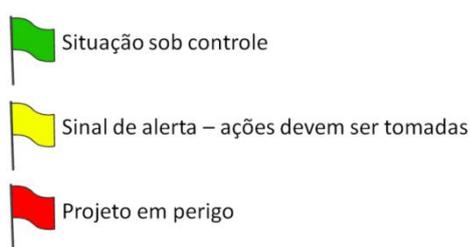


Figura 13. Sinais visuais de alerta do quadro visual de monitoramento do projeto

Fonte: Elaborada pelo autor.

No apêndice 6 é apresentado o quadro visual de monitoramento do projeto com descrição e informações sobre as etapas e com instruções de preenchimento; e no apêndice 7 está apresentado este mesmo quadro em branco, da forma que deve ser fornecido a equipe no início do projeto.

A presença física de um quadro kanban é uma prática bastante utilizada nos projetos ágeis de software, por isso recomenda-se o quadro visual de monitoramento esteja localizado ao seu lado ou o mais próximo possível. Normalmente as equipes realizam reuniões diárias com os membros da equipe posicionados em frente ao quadro kanban, estando o quadro visual ao seu lado, se tornará fácil incluir a sua análise neste momento de discussão.

Nas próximas sessões serão detalhadas as fases do processo proposto.

4.3.2 Fase 1 – Planejamento de medição e divulgação dos resultados

Inicialmente deverá ser realizado o planejamento de medição buscando identificar as necessidades de informação dos envolvidos no projeto e da organização. Essas informações deverão ser utilizadas para definir os objetivos de medição e quais métricas que atendam aos

objetivos. Como ponto de partida para os membros da equipe refletir sobre suas necessidades de medição e sobre as métricas a serem adotadas, poderão ser utilizados:

- o conjunto de métricas apresentado na sessão 4.1;
- o mapa mental apresentado no Apêndice 4;
- o agrupamento das métricas por fases do projeto apresentado na sessão 4.2.2.

Nesta etapa devem ser realizadas as seguintes análises:

- i. Quais são os envolvidos no projeto e suas necessidades de informação para controle das suas atividades?
- ii. Quais são as métricas que fornecem as informações necessárias e quais são os objetivos de medição?
- iii. Quais os valores a serem alcançados para as métricas (caso houver) e quais os valores que representam risco ao projeto?
- iv. Qual a periodicidade de coleta e atualização da métrica?
- v. Quais os responsáveis pela coleta dos dados e atualização de cada métrica no quadro visual?
- vi. Existem considerações de projetos anteriores que poderão ser utilizados como suporte para melhorar o plano de medição? (resultado da Fase 6 Avaliação final dos projetos anteriores, se houver)

A decisão das métricas a serem adotadas não se restringe ao conjunto apresentado nesta tese, podendo ser possível serem adotadas outras métricas que não estejam aqui contempladas.

Apesar desta tese não tratar diretamente sobre indicadores de desempenho, recomenda-se que sejam estabelecidas metas de valores a serem alcançadas com as métricas e quais os valores que representam risco ao projeto, devendo ser adotado de acordo com a realidade de cada empresa.

No quadro visual de monitoramento do projeto (Apêndices 6 e 7) a área destinada para ser preenchida nesta fase possui o título ‘1 Planejamento’ o qual está subdividida em 2 subetapas:

- a) quem somos: deverá conter o nome e função todas as pessoas que participaram da discussão e preenchimento deste quadro;

- b) definições: deverá conter a lista das métricas que foram definidas para serem utilizadas devendo descrever o nome da métrica, objetivo de medição, limite de controle (caso houver) e o nome do responsável pela medição, sendo que cada métrica poderá ter um responsável de medição diferente.

Para a fase de planejamento de medição recomenda-se:

- não destacar nomes ou cargos funcionais de pessoas que tomam decisões ou realizam ações sobre as métricas. Essa recomendação se faz necessária para respeitar os princípios ágeis de manter equipes auto-organizáveis onde os membros refletem em como se tornar mais efetivos incentivando a discussão e responsabilidade coletiva;
- iniciar com um conjunto pequeno de métricas selecionado os objetivos de medição essenciais necessários para obter controle sobre o projeto. Essa recomendação se faz para não ‘pesar’ o processo e inviabilizar o seu uso. Se o esforço de medição for maior que os benefícios das informações fornecidas, todo o processo de planejamento e monitoramento de medição pode ser comprometido, pois poderá gerar resistência entre os membros da equipe;
- eleger métricas que estejam alinhadas aos objetivos organizacionais;
- ser adotado por empresas experientes em métodos ágeis.

Na próxima sessão é apresentada a fase de coleta de dados, monitoramento da iteração e divulgação dos resultados.

4.3.3 Fase 2 – Coleta de dados, monitoramento da iteração e divulgação dos resultados

O próximo passo após o planejamento é a realização da implementação e do monitoramento das métricas. No quadro visual de monitoramento do projeto (Apêndice 6) a área destinada para isso possui o título ‘2 Iteração atual’ a qual está subdividida em várias áreas menores. Cada um desses espaços deverá ser utilizado para monitorar uma métrica.

Ao iniciar a iteração todas as métricas irão apresentar o sinal de bandeira verde representando que a métrica está sob controle. Na fase 1 foram estabelecidos valores a serem alcançados para as métricas e os valores que representam risco ao projeto. Essas informações

devem ser utilizadas pela equipe para observar o momento em que o sinal de alerta deve ser acionado e ações corretivas devem ser tomadas.

Na próxima sessão é apresentada a fase de ações da iteração e divulgação dos resultados.

4.3.4 Fase 3 – Ações da iteração e divulgação dos resultados

Conforme mencionado anteriormente, a atualização das métricas ao longo da iteração deverá ser realizada nas reuniões diárias, porém neste momento poderão ser identificados problemas sendo necessário acionar um sinal de alerta. Um sinal de alerta significa que um problema está ocorrendo e uma ação deverá ser tomada. O objetivo da fase 3 é monitorar os problemas que ocorreram e propor ações a serem tomadas. Recomenda-se registrar os seguintes dados neste quadro:

- problema detectado e sua data da ocorrência. Neste momento a bandeira amarela ou vermelha deverá ser sinalizada conforme o grau de gravidade do problema;
- ações realizadas, data da ação e responsável. Devem ser registradas as ações que estão sendo realizadas para corrigir o problema e a data. Neste momento a bandeira poderá mudar de cor, conforme o status atual da métrica;
- resultado alcançado e data conclusão. Quando o problema for resolvido deverá ser descrito o resultado alcançado e a data de conclusão. Neste momento a cor da bandeira poderá mudar para verde.

Na próxima sessão é apresentada a fase de monitoramento das *releases* e projeto e divulgação dos resultados.

4.3.5 Fase 4 – Monitoramento do projeto/*releases* e divulgação dos resultados

Ao concluir uma iteração os dados das métricas deverão ser atualizados no quadro visual de monitoramento do projeto/*releases* na área que possui o título ‘4 Projeto/*Releases*’. Conforme mencionado anteriormente, neste momento deverá ser mantido o valor acumulado das métricas permitindo observar a evolução dos dados, visualizar tendências e observar

distorções. Preferencialmente, as informações serem consolidadas em forma de gráfico ou representações que permitam compreender o comportamento das métricas ao longo das iterações. Da mesma forma que a fase 2, esta área está subdividida em várias áreas menores, sendo que cada um dessas áreas deverá ser utilizado por uma métrica específica. O momento de atualização desses dados é na reunião de retrospectiva da iteração, no qual os dados a avaliação do desempenho da iteração são avaliados. Na próxima sessão é apresentada a fase de ações do projeto/*releases* e divulgação dos resultados.

4.3.6 Fase 5 – Ações sobre o projeto/*releases* e divulgação dos resultados

Da mesma forma que ocorre com as iterações, está previsto um espaço de monitoramento para os problemas e ações corretivas a serem tomadas no decorrer do projeto/*releases* ou referente ao projeto como um todo. Sinais de alerta utilizando as bandeiras devem ser utilizados para sinalizar o *status* e gravidade do problema.

Conforme mencionado anteriormente, a atualização das métricas ao longo da iteração deverá ser realizada nas reuniões diárias, porém neste momento poderão ser identificados problemas sendo necessário acionar um sinal de alerta. Um sinal de alerta significa que um problema está ocorrendo e uma ação deverá ser tomada. O objetivo da fase 5 é monitorar os problemas que ocorreram e ações que foram tomadas. Recomenda-se registrar os seguintes dados neste quadro o problema e sua data da ocorrência, as ações realizadas e data da ação e ao concluir, o resultado alcançado e data de conclusão.

4.3.7 Fase 6 – Avaliação final e divulgação dos resultados

Os objetivos de medição e necessidades de controle poderão mudar ao longo do tempo, por esse motivo o plano de medição deverá ser constantemente revisado. Recomenda-se que essa análise seja realizada no final do projeto durante a reunião de retrospectiva final, nos casos de projetos de longa duração esta avaliação poderá ser feita após a conclusão de uma *release* que se julgue estar num estado avançado do projeto e que permita esta análise ser feita. Nesta etapa devem ser realizadas as seguintes análises:

- i. Os objetivos de medição foram atingidos com essas métricas?
- ii. Os *stakeholders* conseguiram ter as informações necessárias para tomada de decisão ao longo do projeto?
- iii. O esforço de medição das métricas definidas justifica-se pelo benefício das informações fornecidas?

O quadro visual de monitoramento do projeto (Apêndices 6 e 7) propõe uma área subdivida em 2 partes para destacar os pontos positivos e pontos negativos observados ao longo do processo de medição. Recomenda-se que os membros da equipe possam ter a liberdade no decorrer do projeto colar *post-it* nestes espaços indicando o que achou bom ou ruim. Na reunião final esses pontos deverão ser discutidos entre os membros da equipe. Essas informações devem ser mantidas como histórico de lições aprendidas sobre o processo de medição do projeto e servirão de base para a discussão na próxima momento que ocorrer uma reunião da fase 1.

Na próxima sessão é realizada a avaliação do processo e discutido sobre os resultados alcançados.

4.4 AVALIAÇÃO DO PROCESSO PROPOSTO

Conforme descrito na sessão 1.3.3 o processo proposto foi submetido à avaliação de especialistas em métodos ágeis da academia e do mercado. O objetivo desta etapa foi submeter a proposta a avaliação frente aos critérios definidos para sua construção e aos benefícios percebidos pelos avaliadores. A seguir serão apresentados os resultados das referidas entrevistas. A avaliação foi realizada através de entrevistas semi-estruturadas seguindo o roteiro e questionamentos apresentado no Apêndice 8. No Quadro 43 é apresentada uma síntese dos dados das empresas pesquisadas e do perfil dos entrevistados que participaram da avaliação do processo proposto.

Quadro 43. Síntese dos dados das empresas pesquisadas e do perfil dos entrevistados que participaram da avaliação do processo proposto

Entrevistado	Perfil do entrevistado	Descrição da empresa	Descrição do entrevistado
1	Especialista da academia	-	Pesquisador sobre métodos ágeis. Professor universitário em área específica do tema da pesquisa. Doutorado com tema em métodos ágeis.
2	Especialista do mercado	-	Consultor ágil há 6 anos. Implementador e consultor MPS.BR e CMMI há 7 anos. Docente de ensino superior. Possui mestrado com tema de pesquisa em métodos ágeis.
3	Especialista do mercado	Empresa de médio porte experiente na utilização de métodos ágeis	Atua em cargo de gestão. Possui 5 anos de experiência em métodos ágeis. Responsável pela definição das métricas e análise de desempenho dos projetos.
4	Especialista do mercado	Empresa de grande porte experiente na utilização de métodos ágeis	Atua como engenheiro de sistemas e <i>agile coach</i> . Possui 6 anos de experiência em métodos ágeis. Possui mestrado com tema de pesquisa em métodos ágeis.

Fonte: Elaborado pelo autor.

Após contextualização da pesquisa, apresentação do processo proposto, do conjunto de métricas e suas representações, foram realizadas avaliação do processo quanto aos requisitos previamente definidos na sessão 4.3.

Em relação às perguntas iniciais sobre a existência de um processo de medição formalizado nas empresas entrevistadas, nenhuma possui um processo de medição formalizado. Todos os entrevistados desconhecem um processo de medição específico para métodos ágeis de desenvolvimento de software.

A análise foi realizada através uma escala ordinal de 5 pontos, sendo que os seus limites são: 1 (discordo totalmente) e 5 (concordo totalmente). Apesar da avaliação ter sido feita com um número reduzido de entrevistados, o alto grau de especialização desses profissionais sobre o assunto permite obter informações críticas sobre o processo. A Tabela 6 apresenta a média das avaliações realizadas pelos entrevistados.

Tabela 6. Média das respostas dos entrevistados sobre os requisitos para a definição do processo de medição

PERGUNTA	MÉDIA DAS RESPOSTAS	DESVIO PADRÃO
Quanto ao uso do processo, ele não exige a criação de novos cargos ou funções	5,0	0,0
Quanto aos benefícios do processo, permite monitorar as métricas ao longo do projeto	5,0	0,0
Quanto à comunicação, o processo permite acesso às informações para todos os interessados de forma rápida e simples	4,8	0,5
Quanto à compreensão do processo, você o considera simples de ser entendido	4,8	0,5
Quanto aos benefícios do processo, fornece orientações que permitem à empresa refletir sobre os seus objetivos de medição e necessidades de controle através do uso de métricas	4,5	0,6
Quanto ao uso do processo, ele não agrega novas etapas ao processo de desenvolvimento do software na sua empresa	4,3	0,5
Quanto ao alinhamento com a abordagem ágil, respeita os valores e princípios ágeis	4,3	0,5
Quanto à implantação do processo, você o considera fácil de ser implantado	3,5	0,6

Através das respostas pode ser considerado que os requisitos para a definição do processo de medição foram atingidos, sendo que a maioria obteve uma média superior a 4. A nota 3,5 para implementação foi subjetiva, uma vez que não foi implementado, mas possivelmente a dificuldade de implementação estaria ligada ao fato de agregar mais uma tarefa ao trabalho da equipe.

Também foram realizadas perguntas permitindo os entrevistados relatarem suas impressões sobre o processo proposto. No Quadro 44 são apresentado os depoimentos dos entrevistados e as perguntas realizadas.

Quadro 44. Avaliação do processo pelos entrevistados

Entrevistado	Qual a sua opinião sobre o processo de medição proposto?	Qual a sua opinião sobre a especificação das métricas e instrumentos de apoio	Qual seria a utilidade do processo proposto para as empresas
1	<p><i>“Realmente esta é uma lacuna que nós temos, esse processo irá ajudar muito as empresas que adotam métodos ágeis... O processo apresenta orientações claras ajudando na sua adoção ... A forma visual de comunicação visual será muito útil para as equipe, permitindo visualmente saber como está o andamento do projeto tanto a nível de iteração e quanto no projeto como um todo ... Ensina que as pessoas comecem aos poucos e evoluam conforme seu aprendizado ... Outro ponto positivo que eu enxerguei é o fato de trabalhar com a equipe inteira pois são esses que realmente sabem das necessidades reais da equipe”.</i></p>	<p><i>“Os documentos ajudam a compreender quais métricas eu posso usar e pra que usar. Isso irá ajudar muito, pois quando uma empresa inicia é difícil para fazer esta definição, pois normalmente não se tem uma relação abrangente de quais métricas podem ser utilizadas, podendo ser útil neste momento de definição”.</i></p>	<p><i>“Sendo este um problema real que as empresas enfrentam, tanto o processo quanto a documentação das métricas irão ajudar muito as empresas refletirem sobre o assunto”.</i></p>
2	<p><i>“Existe uma pressão por parte da alta gerência por uma maior visibilidade dos resultados dos projetos ágeis. Com o processo aqui definido, é possível entender as necessidades dos diversos envolvidos e, com a visualização no quadro a comunicação é muito mais eficaz. O processo definido é de fácil compreensão. A dificuldade na sua implantação estaria relacionada à cultura organizacional e ao entendimento dos objetivos estratégicos da organização e dos objetivos de medição relacionados. Saber o que medir e o porquê disso é o grande desafio.”</i></p>	<p><i>“Os instrumentos são bastante úteis como uma orientação para que a própria empresa possa analisar suas necessidades e a sua capacidade em obter determinada métrica. A dificuldade aqui fica por conta da própria filosofia do ágil que trabalha bastante com análises mais subjetivas. Acredito que o desafio para a implantação nas empresas fique por conta da seleção de métricas que não gerem custos excessivos para o projeto e também que não induzam ao individualismo, em contraponto ao espírito de equipe valorizada no ágil.”</i></p>	<p><i>“Enxergo uma grande utilidade do processo nas empresas para as quais presto consultoria, exatamente por possibilitar o monitoramento dos projetos ágeis com o mesmo nível de confiabilidade que se imagina ter nos projetos tradicionais. Diminuiria em muito a lacuna encontrada e as solicitações por indicadores por parte da gerência.”</i></p>

3	<p><i>“O processo é de fácil entendimento. Demonstra uma aderência grande na abordagem dos métodos ágil. Este processo vem a contribuir com um gap do processo ágil. Pois o processo ágil não possui ferramentas de medições e nem um processo estabelecido.”</i></p>	<p><i>“O mapa mental apresenta uma quantidade de informações bastante detalhada, precisando entender sua organização para compreendê-lo. O quadro de métricas por fase deixa claro as timebox do projeto. Assim como demonstra a forma como será dividido/organizado o projeto. A estrutura da descrição técnica é simples e de fácil entendimento, possui todas as informações necessárias para a aplicação das métricas. Gostei bastante da proposição do quadro visual, desde a colocação próxima ao kanban, como a organização do mesmo. Tem marcadores visíveis, facilitando observar pontos de atenção de longe (bandeira vermelha). Sendo possível também mensurar métricas da iteração e da release.”</i></p>	<p><i>“A utilização desse processo dentro da empresa seria de grande valia. Dentro da empresa já utilizamos algumas métricas, mas não possui a estruturação clara do processo. Logo agregaria bastante desde o planejamento/entendimento, execução/acompanhamento e fechamento do projeto.”</i></p>
4	<p><i>“Gosto das vantagens deste modelo que se aproxima do dia-a-dia de um projeto Scrum. Isso é de fato positivo e creio que pode auxiliar as equipes a prestarem atenção a possíveis métricas a serem utilizadas no projeto”.</i></p>	<p><i>“Acho o quadro visual valioso, devendo ser um ponto de partida para a equipe discutir sobre as métricas ... não deve ser rígida a definição em todo o projeto ... a apresentação das métricas num ambiente visual pode ser útil para a equipe, porém deve ter cuidado para não ter poluição visual ... O mapa mental possui muitas informações que inicialmente podem parecer confusas devendo haver uma explicação”.</i></p>	<p><i>“Creio que principalmente didática. Acredito que o modelo ajudaria para melhorar a capacidade da equipe em definir métricas com base em seu contexto ... Creio que aprender a criar/selecionar métricas de acordo com a necessidade do projeto é um hábito que pode ser aprendido e transformado como inerente à cultura da equipe ou organização”.</i></p>

Fonte: elaborado pelo autor com base nas respostas dos entrevistados.

Através da análise dos depoimentos dos entrevistados é possível observar que tanto o modelo proposto quanto à especificação detalhada do conjunto de métricas será útil para as empresas. Também foi possível confirmar o preenchimento da lacuna de pesquisa.

As limitações desta etapa da pesquisa foram: o fato de algumas serem realizadas via Skype, o número de entrevistados e não ter aplicado numa empresa real para avaliar os resultados.

4.5 CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO 4

Este item será dividido em duas partes, uma dedicada à análise do conjunto de métricas proposto e outra à análise do processo de medição.

4.5.1 Análise do conjunto de métricas

Este capítulo apresentou um conjunto de métricas para projetos ágeis de software que auxiliam na gestão de projetos, as quais são especificadas quanto à sua definição e forma de uso. Procurou-se trazer ordem e clareza sobre informações relativas às métricas incluindo: quais métricas têm sido utilizadas em projetos ágeis de software, quais os benefícios proporcionados pela métrica, apresentados através de especificação dos objetivos de medição, quais fases de um projeto ágil podem ser analisadas, quais os envolvidos no projeto poderão ter interesse, os pressupostos para utilização, equação de cálculo e comportamento das métricas. A busca por essas informações foi difícil em função da literatura ser escassa e pouco aprofundada, muitas vezes devendo ser proposta uma solução pelo próprio autor, a partir das descrições encontradas.

Procurando atender aos requisitos de boas métricas definido por Nelly et al. (1997) buscou-se definir um título claro para as métricas encontradas na literatura e nas empresas. Muitas vezes os títulos foram adequados a partir dos dados levantados na literatura ou estabelecidos a partir dos objetivos de medição. Em outros casos, uma explicação adicional foi necessária e colocada entre parênteses, como pode se ver em: defeitos em aberto (defeitos não solucionados); *cycle time* (tempo de ciclo para concluir uma tarefa); densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto) e outras.

Após a especificação detalhada das métricas procurou-se explorar não somente o objetivo que a métrica atenderia, mas também qual o tipo de comportamento ela desencadearia na equipe ou interessados. Esta análise foi importante e permitiu, mesmo que em nível teórico, prever impactos positivos ou negativos sobre o projeto do software. Estes impactos serão discutidos a seguir, em blocos.

Métricas cujos resultados podem desencadear aceleração da equipe, como: *Taxa de acerto na estimativa das tarefas*, *Burnup da release/projeto*; *Cycle time*, *Número de integrações por dia*, *Número de tarefas não concluídas na iteração* e *Throughput*, poderão ter um impacto positivo caso a equipe esteja consciente dos valores e princípios ágeis que preconizam manter passos constantes. A equipe poderá replanejar as entregas se for necessário, quebrar tarefas em pacotes menores e gerenciáveis, colocando mais pessoas para realizar a tarefa ou adotando outra estratégia. Do contrário, os resultados da métrica poderão causar um desvio da forma como a equipe conduz projeto ágil. Tal análise pode desencadear a seguinte hipótese (H1): métricas que desencadeiam a aceleração da equipe podem desviar a equipe de um ritmo sustentável do projeto ágil.

Métricas cujos resultados podem desencadear desconforto da equipe, podendo ser entendidas como um excessivo controle, como: *Número de horas efetivamente trabalhadas em atividades de desenvolvimento* e *Total de horas trabalhadas no projeto* poderão ter um impacto positivo caso exista uma justificativa clara para o seu uso, como reduzir custos, identificar causas de atrasos, ou outras. Do contrário, os resultados da métrica poderão causar desmotivação da equipe. Observa-se um aspecto positivo que estas métricas medem o grupo e não o indivíduo. Esta análise pode desencadear a seguinte hipótese (H2): Métricas que exercem controle rígido sobre as horas de trabalho podem ter impacto negativo sobre a produtividade da equipe.

Métricas cujos resultados podem desencadear ações de melhoria do fluxo de tarefas, como: *Diagrama de fluxo cumulativo da tarefas* e *Work in Progress* poderão ter um impacto positivo caso sejam realizadas ações, ainda durante o projeto, para reduzir problemas de filas, de gargalos, reduzindo possíveis perdas ou atrasos. Do contrário poderão representar desgaste de coleta e análise dos dados, sem ser útil. Esta análise pode desencadear as seguintes hipóteses (H3): As métricas que desencadeiam ações de melhoria do fluxo de tarefas auxiliarão na redução de perdas; e (H4) As métricas que desencadeiam ações de melhoria do fluxo de tarefas são mais indicadas para projetos de médio e grande portes.

Métricas cujos resultados podem desencadear ações de minimização do surgimento de defeitos, como: *Defeitos em aberto (defeitos não solucionados)* e *Defeitos encontrados pelo cliente* poderão ter um impacto positivo caso a equipe se aprimore em encontrar defeitos do projeto ainda na fase de testes e que não deixe acumular para fases subsequentes. Do contrário poderão afetar a qualidade do produto entregue, desviando a equipe de atender o princípio de

contínua atenção à excelência técnica e ao bom projeto aumentam a agilidade. Entende-se que a área ‘métricas para qualidade do produto entregue’ merece um estudo específico.

Métricas cujos resultados podem desencadear ações de gerar aprendizado referente o custo benefício do projeto, como: *Tempo investido em tarefas não planejadas*, *Variação do número de defeitos entre iterações* e *Variação dos custos (entre estimado e realizado)* poderão ter um impacto positivo caso a equipe utilize esta informação para compreender os seus problemas e tomar ações corretivas. Do contrário poderão representar desgaste de coleta e análise dos dados, sem ser útil. Esta análise pode desencadear a seguinte hipótese (H5): Deve ser estabelecido um plano de ação para agir sobre métricas que envolvem custos ou desperdício dos recursos.

Métricas cujos resultados podem desencadear ações de desenvolver funcionalidades que atendam melhor as necessidades do cliente, como: *Funcionalidades Testadas e Entregues* e *Grau de atendimento aos requisitos funcionais* poderão ter um impacto positivo caso a equipe aprimore a forma de planejar as funcionalidades a serem desenvolvidas, devendo ter foco no atendimento às necessidades do cliente. Do contrário poderão representar desgaste de coleta e análise dos dados, sem ser útil. Essas métricas fazem parte da ‘métricas para qualidade do produto entregue’ merece um estudo específico.

Foi observado que existe um conjunto de métricas que não são exclusivas dos métodos ágeis e que são utilizadas pelas empresas. Elas têm contribuído para as equipes ágeis a obter melhor controles sobre suas atividades, entre elas:

- Acurácia das estimativas,
- Funcionalidades testadas e entregues,
- Índice de defeitos encontrados na fase de testes,
- Nível de satisfação do cliente e
- Índice de motivação dos membros da equipe.

Entende-se que estas métricas poderão e que reforçam a filosofia ágil e poderá desencadear um comportamento positivo sobre os membros da equipe. Se a equipe vier a substituir induzir a equipe a substituir as práticas administrativas por práticas de registro, comunicação e gestão mais ágeis

Observa-se também que algumas métricas que não são exclusivas dos métodos ágeis e que não possuem um alinhamento com princípios ou práticas dos métodos ágeis, entre elas Total de horas trabalhadas no projeto e Número de horas efetivamente trabalhadas no projeto,

sendo estas métricas que poderão trazer aspectos negativos no comportamento da equipe. Caso uma empresa necessite adotar essas métricas para obter controle sobre os recursos do projeto recomenda-se que não sejam medidos os indivíduos e sim um acumulado da equipe, amenizando o aspecto negativo junto ao grupo.

O conjunto de métricas apresentado é amplo atende a diversas necessidades de informação e controle para projetos ágeis de desenvolvimento de software. As empresas precisarão escolher quais métricas serão úteis de acordo com suas necessidades específicas. Algumas métricas possuem objetivos similares e que são intercambiáveis, podendo ser escolhidas entre essas aquelas qual mais se enquadra com suas necessidades de medição da organização. Entre elas pode-se citar Funcionalidades Testadas e Entregues e Grau de atendimento aos requisitos funcionais, Acurácia das estimativas e Taxa de acerto na estimativa das tarefas.

Por fim, observa-se, que outras métricas poderão ser desenhadas nas empresas para atender aos mesmos objetivos encontrados no estudo, ou mesmo atender outros objetivos organizacionais, não mencionados. Guardadas as limitações desta tese e considerando o interesse das empresas em utilizar o conjunto aqui apresentado, ressalta-se que cada métrica deverá ser testada/validada futuramente nos contextos das empresas. Justifica-se esta validação, conforme observado em Neely et al. (1997), pelo fato de que as características de uma métrica, desde seu nome, fórmula, metas ou outras serem susceptíveis de alteração em função do contexto de aplicação. Ainda no contexto de uma futura validação, esta também poderá ter como objetivo verificar quais subconjuntos de métricas são mais adequados para as diferentes fases do projeto (diário, iteração, release), mais aderentes aos objetivos organizacionais, quais são as limitações e dificuldades de uso, coleta, interpretação e proposição de ações.

Em função da delimitação desta pesquisa, não foram incluídas métricas de nível estratégico. Da mesma forma, não foi considerada na análise as possíveis interações existentes entre as métricas, elas foram tratadas como informações independentes.

4.5.2 Análise do processo de medição

Visando analisar o processo de medição proposto foram eleitos os critérios adotados por Folan e Browne (2005): especificidade, perspectivas de avaliação, estrutura procedimental, outros apoios à medição, dinamismo da proposta.

Em relação à especificidade, ela foi desenhada para monitorar o desenvolvimento de software pela metodologia ágil, e não se espera que seja utilizada necessariamente em outras situações.

Quanto às perspectivas de avaliação, pode-se considerar que as dimensões de análise abrangidas pelo método, até o momento, são a organizacional, a equipe, a qualidade do produto e a qualidade do processo. Elas coincidem com as categorias de agrupamento das métricas, visto que a delimitação do estudo se circunscreveu aos níveis tático e operacional. Não se encontram, por exemplo, uma categoria Financeira ou de inovação e aprendizagem, a exemplo do BSC, que teriam um viés estratégico. No entanto, a proposta de medição é flexível e dependente das categorias de avaliação das métricas que serão usadas. Neste sentido, se no futuro novas categorias de análise forem incorporadas, o método continuará acomodando-as. Recomenda-se de qualquer forma, que se busque um balanceamento entre as métricas e entre categorias, financeiras e não financeiras, métricas de tendência e resultado, atendimento às perspectivas interna e externa, entre objetivos de curto e longo prazos, o que no momento não se alcançou plenamente.

A estrutura procedimental proposta é intuitiva, seguindo um ciclo de gestão de planejar, executar, verificar, atuar, refletir e melhorar a forma de realizar a medição em seus projetos, respeitando as características dos projetos ágeis de desenvolvimento de software. Ela contempla (i) um procedimento lógico com passos a serem seguidos, (ii) leva em considerações aspectos organizacionais como responsáveis (equipe) e reuniões e, (iii) inclui um conjunto de métricas que servem de inspiração para os gestores. Desta forma, a proposta é tão abrangente quanto propostas sistêmicas de medição existentes na literatura (ver FOLAN e BROWNE, 2005). Importante mencionar que a etapa de planejamento é fundamental para o sucesso da implementação, mas não foi detalhada. Questionamentos deixados pelos autores guiarão, como um *check list* a realização das atividades da fase de planejamento. Em trabalhos futuros, a etapa de planejamento poderá ser aprofundada, assegurando que seja o elo de ligação com o nível estratégico, visto que ali serão declarados os objetivos de negócio e as métricas necessárias ao acompanhamento dos resultados organizacionais.

Em relação a outros apoios que a proposta deixa à medição, podem se incluir os materiais gráficos como o quadro de acompanhamento visual, o mapa mental e o conjunto de métricas mencionados anteriormente. As propriedades de comunicação e vínculo podem ser proporcionadas por estes recursos visuais e pelas recomendações incluídas para uso deste processo, como: definição e avaliação das métricas de forma participativa, entender as necessidades de informação dos envolvidos e objetivos de medição antes de definir quais métricas serão utilizadas, eger métricas que estejam alinhadas com os objetivos organizacionais, iniciar com um conjunto mínimo de métricas, ser adotado por empresas experientes em métodos ágeis, entre outras.

O dinamismo da proposta é assegurado pelo planejamento, uso e acompanhamento do quadro visual nas reuniões regulares, já pertencentes ao processo de desenvolvimento de software pelos métodos ágeis. A proposta permite, portanto, traduzir a visão dos gestores em relação à análise e monitoramento das métricas de projeto, ao longo dos níveis tático e operacional. Embora não tenha sido desenvolvido no trabalho, maior dinamismo ainda poderá ser dado ao se desenvolver o quadro de acompanhamento em formato digital. Este último será fácil de ser preenchido e monitorado pela equipe e gestores. Importante comentar que o processo no formato que foi desenhado é aderente às recomendações sobre processo de medição do modelo de maturidade MPS.BR.

Por fim, pode-se mencionar que o aprendizado e *feedback* também foram contemplados na proposta, visto que o quadro visual prevê um espaço para planejamento de ações que venham a reverter os valores indesejados das medidas e métricas.

Ao realizar a avaliação do processo com especialistas foi evidenciado o processo proposto atende aos requisitos previamente definidos. Os especialistas consideraram tanto o processo proposto quanto a especificação do conjunto de métricas úteis para as empresas, por serem juntos uma solução para um problema real, que as empresas que utilizam métodos ágeis enfrentam.

5 CONSIDERAÇÕES FINAIS

Esta tese apresentou um conjunto de métricas consolidado para auxiliar na gestão de projetos ágeis de desenvolvimento de software nas fases de projeto/releases, iteração e diário. Também propõe de um processo de medição compatível com a abordagem ágil de desenvolvimento de software, visando apoiar as empresas que adotam métodos ágeis na definição de métricas adequadas para suas necessidades de medição e no monitoramento.

A primeira questão que pesquisa QS1 era ‘Quais são as métricas que atendem às necessidades de medição inerentes à natureza dos projetos ágeis de desenvolvimento de software, que são citadas pela literatura?’. Para responder esta questão foi realizada uma pesquisa sistematizada na literatura, a qual teve como objetivos de investigar estudos que tratem especificamente do tema utilização de métricas em projetos de desenvolvimento de software que utilizem métodos ágeis e identificar métricas que permitam as empresas desenvolvedoras de software que utilizem métodos ágeis a controlar os seus projetos nas fases de *release*, iteração e diário. A pesquisa foi realizada nas bases eletrônicas *IEEE Xplore* e *ISI Web of Science* e como resultado foram localizados 14 estudos que pesquisaram sobre métricas em projetos de software que utilizam métodos ágeis e um conjunto de 30 métricas. O levantamento revelou uma carência de estudos que tratem do assunto em profundidade, o que se justifica pelo fato de ser um tema de pesquisa recente. A revisão de métricas realizada nesta pesquisa é uma contribuição relevante para a área de conhecimento. Trabalhos futuros poderão ampliar a pesquisa para verificar se existem métricas não identificadas. Os resultados desta questão de pesquisa estão apresentados no capítulo 2.

Para responder a questão QS2 ‘Quais são as métricas que atendem as necessidades de medição inerentes à natureza dos projetos ágeis de desenvolvimento de software, que são utilizadas pela empresa que os adotam?’ foi desenvolvida uma pesquisa de campo realizadas através de entrevistas com profissionais de 9 empresas experiência na utilização de métodos ágeis. Vários das métricas identificadas na pesquisa de campo já haviam sido identificadas na revisão da literatura, sendo que no total obteve-se um conjunto de 40 métricas. A pesquisa foi realizada com empresas da região sul do Brasil, podendo ser estendida no futuro para outras empresas brasileiras e estrangeiras. Eventualmente, tal ampliação poderá detectar um

conjunto maior de métricas além daquelas mencionadas nesta pesquisa. Os resultados desta questão de pesquisa estão apresentados no capítulo 3.

A terceira questão que pesquisa QS3 era ‘Que elementos devem ser considerados para definir um processo de medição que atenda às necessidades de medição dos envolvidos em projetos de desenvolvimento de software que adotam métodos ágeis?’ Para responder questão primeiramente foram propostos requisitos para a definição do processo de medição para projetos ágeis de software, levando em consideração a própria natureza dos métodos ágeis. A partir deste conjunto de requisitos foi proposto de processo de medição, compatível com a abordagem ágil de desenvolvimento de software, visando apoiar as empresas de software que adotam métodos ágeis na definição de métricas adequadas para suas necessidade. Para atender a esta questão de pesquisa também foi apresentado um conjunto de métricas consolidados para auxiliar na gestão de projetos de desenvolvimento de software que utilizam métodos ágeis, as quais são especificadas detalhadamente contendo as informações necessárias para seu entendimento e aplicação. Os resultados desta esta da pesquisa foram avaliados junto a especialistas da academia e do mercado os quais consideraram uma contribuição relevante para um problema real que as empresas de software que utilizam métodos ágeis enfrentam, comprovando a contribuição prática e teórica do processo proposto apresentado. Quanto a contribuição acadêmica esta tese está o conjunto de métricas consolidado para projetos ágeis de software e a definição de um processo de medição compatível com os métodos ágeis. Quanto a contribuição para a prática nas empresas de desenvolvimento de software, desca-se a abordagem didática do processo proposto, a gestão visual do processo através do quadro visual de monitoramento, as recomendações a serem seguidas em cada etapa do processo auxiliando as decisões ao adotá-lo, na maneira evolutiva de adotar o processo e de monitorar os projetos, e por clarear quais métricas podem ser utilizadas na prática e como utilizá-las. Percebe-se que a proposição do conjunto de métricas e do processo de medição deverão ser colocados à prova para verificar a sua validade através da aplicação prática em empresas reais. Desta forma poderá ter análise mais detalhadas dos seus pontos positivos e negativos. A partir desses dados poderão então se realizar melhoria no trabalho desenvolvido.

Considera-se que os objetivos propostos nesta tese foram atendidos, uma vez que foi apresentado como resultados um conjunto de métricas consolidados para auxiliar na gestão de projetos ágeis de desenvolvimento de software; e um processo de medição compatível com a abordagem ágil de desenvolvimento de software. Conforme mencionado na introdução, esta tese não teve como pretensão esgotar o estudo sobre o assunto, porém pretendia contribuir

trazendo ordem e clareza sobre tema em questão, o que foi possível com os resultados desta tese.

O caráter exploratório trabalho permitiu observar oportunidades de pesquisa e 5 hipóteses a serem testadas em trabalhos futuros, como citado anteriormente. Dentre as oportunidades futuras de pesquisa podem-se citar:

- avaliar o processo proposto através da aplicação prática em projetos reais, onde também será possível validar as métricas e suas diferentes representações;
- explorar a relação de interdependência entre essas métricas;
- propor indicadores de desempenho por categorias de avaliação permitindo gerar informações para decisão em nível estratégico e nas quais se considerem as interdependências entre as métricas;
- entender aspectos financeiros que envolvem os métodos ágeis, ampliando as métricas relacionadas a custos e relacionando-as com os objetivos estratégicos;
- estudar o comportamento da utilização das métricas em projetos de desenvolvimento de software que utilizam métodos ágeis em ambientes de que possuem certificação CMMI ou MPS.BR e que não possuem certificação;
- desenvolver um sistema que permita automatizar a coleta, armazenamento e processar os dados das métricas, mantendo assim o registro histórico dos dados do projeto e facilitando a implementação do processo.

REFERÊNCIAS

ABBAS, N.; GRAVELL, A.; WILLS, G. The Impact of Organization, Project and Governance Variables on Software Quality and Project Success. In: AGILE CONFERENCE, 2010, Washington, DC. **Proceedings...** IEEE Computer Society, Washington, DC, USA, p.77-86, 2010.

ASSOCIAÇÃO Brasileira das Empresas de Software. **Mercado Brasileiro de Software: panorama e tendências**, 2011. Disponível em <http://www.abes.org.br/UserFiles/Image/PDFs/Mercado_BR2011.pdf>. Acesso em: 08 mai. 2012.

AGILE Manifesto. **Manifesto for Agile Software Development**, 2001. Disponível em <<http://agilemanifesto.org>>. Acesso em: 25 jul. 2012.

AKINORI, S. **Niko-niko Calendar**. Janeiro, 2006. Disponível em <http://www.geocities.jp/nikonikocalendar/index_en.html>. Acesso: 10 jan. 2014.

BASILI, V. R. **Lecture material of the course on A Quantitative Approach to Software Management and Engineering**, Department of Computer Science, University of Maryland, 1999.

BASILI, V. et al. Goal question metric paradigm. **Encyclopedia of software engineering**. In: MARCINIAK, J.J. (Ed.), Encyclopedia of Software Engineering. John Wiley and Sons, pp. 528–532.

BASTOS, A. **Base de conhecimento em teste de software**. São Paulo: Martins Fontes, 2007.

BECK, K. B.; ANDRES, C. **Extreme Programming Explained: Embrace Change**. 2nd ed. [S.I.]: Addison-Wesley Professional, 2004.

BERNARDO, P. C.; KON, F. A importância dos testes automatizados. **Engenharia de Software Magazine**, v. 1, n. 3, p. 54-57, 2008.

BOEG, J. **Kanban em 10 passos** . InfoQ Brasil, 2012.

BRIAND, L. C. et al. Practical Guidelines for Measurement-Based Process Improvement. **Software Process**, v. 2, n. 4, dez. 1996.

BUZAN, T. **Mapas mentais e sua elaboração**. São Paulo: Cultrix, 2005.

CHENG, T.; JANSEN, S.; REMMERS, M. Controlling and monitoring agile software development in three dutch product software companies. In: 2009 ICSE WORKSHOP ON SOFTWARE DEVELOPMENT GOVERNANCE, 2009, Washington, DC. **Proceeding...** Washington, DC, USA, p. 29-35, 2009.

CHIKOFFSKY, E.; RUBIN, H. A. Using metrics to justify investment in IT. **IT professional**, v. 1, n. 2, p. 75-77, 1999.

COHEN, M. A.; ELIASBERG, J.; HO, T.J. New product development: The performance and time-to-market tradeoff. **Management Science**, vol. 42, n.2, p. 173-186, 1996.

COHN, M. **Agile Estimating and Planning**. New York: Prentice Hall, 2005.

DELAMARO, M. E.; MALDONADO, J.; JINO, M. **Introdução ao teste de software**. Rio de Janeiro, RJ: Elsevier, 2007.

DINGSOYR, T. et al. A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, v. 85, n. 6, p. 1213–1221, jun. 2012.

DUBINSKY, Y. et al. Agile metrics at the israeli air force. In: AGILE CONFERENCE, 2015. **Proceedings...** IEEE , 2005. p. 12-19.

DYBA, T.; DINGSOYR, T. Empirical studies of agile software development: A systematic review. **Information and Software Technology**, Amsterdam, v. 50, n. 9-10, p.833-859, ago. 2008.

FEYH, M.; PETERSEN, K. Lean Software Development Measures and Indicators. **Lean Enterprise Software and Systems**. p. 32-47, 2013.

FOLAN, P.; BROWNE, J. A review of performance measurement: Towards performance management. **Computers in Industry**, v. 56, n. 7, p. 663-680, 2005.

GILB, T.; BRODIE, L. What's Wrong with Agile Methods? Some Principles and Values to Encourage Quantification. **Agile Software Development Quality Assurance**, Information Science Reference, p. 56-69, 2007.

GRAY. D. E. **Pesquisa no mundo real**. 2. ed. Porto Alegre: Penso, 2012.

GREEN, P. Measuring the Impact of Scrum on Product Development at Adobe Systems. **System Sciences (HICSS)**, 2011 44th Hawaii International Conference on, 2011. p. 1-10.

GUSTAFSSON, J. **Model of Agile Software Measurement: A Case Study**. 2011. Dissertação. University of Gothenburg, 2011.

KEZNER, H. **Project management best practices: achieving global excellence**. Hoboken, NJ, USA: John Wiley & Sons Inc, 2006.

KNIBERG, H.; SKARIN, M. **Kanban e Scrum - obtendo o melhor de ambos**. InfoQ Brasil, 2009.

KNIBERG, H. **Scrum e XP direto das Trincheiras**. InfoQ Brasil, 2007.

KULPA, M. K.; JOHNSON, K. A. **Interpreting the CMMI (R): A Process Improvement Approach**. CRC Press, 2004.

HAIR JR. et al. **Fundamentos de métodos de pesquisa em administração**. Porto Alegre: Bookman, 2005.

HARTMANN, D.; DYMOND, R. Appropriate agile measurement: Using metrics and diagnostics to deliver business value. In: AGILE CONFERENCE, 2006, Washington, DC. **Proceedings...** IEEE Computer Society, Washington, DC, USA, p.6pp-134, 2006.

HIGHSMITH, J. **Agile Project Management**: creating innovative products. Boston: Addison-Wesley, 2004.

INTERNATIONAL Organization for Standardization/ International Electrotechnical Commission. **ISO/IEC 15939**: Software Engineering – Software Measurement Process: ISO, 2007.

JAVDANI, T. et al. On the Current Measurement Practices in Agile Software Development. **International Journal of Computer Science Issues (IJCSI)**, v. 9, n. 4, 2012.

KAPLAN, R. S. ; NORTON, D. P. **A estratégia em ação**: Balanced Scorecard. Rio de Janeiro: Campus, 1997.

KITCHENHAM., B., PFLEEGER, S. L., FENTON, N. Towards a Framework for Software Measurement Validation. **IEEE Transactions on Software Engineering**, vol. 21, n. 12, dez. 1995.

LAANTI, M.; SALO, O.; ABRAHAMSSON, P. Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. **Information and Software Technology**, v. 53, n. 3, p. 276-290, 2011.

LARMAN, C.; BASILI, V. Iterative and incremental developments a brief history. **Computer**, v. 36, 47-56. 2003.

LEE, G.; XIA, M. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. **MIS Quarterly**, v. 34, p. 87–114.

MALHOTRA, N.K. **Pesquisa de Marketing**: uma orientação aplicada. 6. edição. Porto Alegre: Bookman, 2012.

MIDDLETON, P.; JOYCE, D. Lean Software Management: BBC Worldwide Case Study. Engineering Management, **IEEE Transactions on**. v. 59, n.1, p.20-32, 2012.

MISHRA, D.; BALCIOGLU, E.; MISHRA, A. Measuring Project and Quality aspects in Agile Software Development. **TTEM-Technics Technologies Education Management**, v. 7, n. 1, 2012.

MOE, N. B. **From Improving Processes to Improving Practice**. 2011. 183 f. Phd Thesis - Norwegian University of Science and Technology, Trondheim, 2011. Disponível em: <<http://ntnu.diva-portal.org/smash/get/diva2:438637/FULLTEXT02>>. Acesso em: 02 jul. 2012.

MOREIRA, M. **Being Agile**: Your Roadmap to Successful Adoption of Agile. [S.I.]: Apress, 2013.

NEELY, A.; GREGORY, M.; PLATTS, K. Performance measurement system design: a literature review and research agenda. **International Journal of Operations & Production Management**, vol. 15, n. 4, p. 80-116, 1995.

- NEELY, A. et al. Designing performance measures: a structured approach. **International Journal of Operations & Production Management**, vol. 17, n. 11, p. 1131-1152, 1997.
- NERUR, S.; BALIJEPALLY, V. Theoretical reflections on agile development methodologies. **Communications of the ACM**, v. 50, n. 3, 79-83, mar. 2007.
- PETERSEN, K.; WOHLIN, C. Measuring the flow in lean software development. **Software: Practice and Experience**, v. 41, n. 9, p. 975-996, 2011.
- PRESSMAN, R.S. **Engenharia de Software**. 6. ed. Porto Alegre: Mc Graw Hill, 2006.
- PRESSMAN, R.S. **Engenharia de Software**. 7. ed. Porto Alegre: Mc Graw Hill, 2011.
- PROJECT Management Institute . **PMBOK Guide**. A Guide to the Project Management Body of Knowledge. Project Management Institute, 5. ed. Português, 2013.
- POONACHA, K. M.; BHATTACHARYA, S. Towards a Framework for Assessing Agility. System Science (HICSS), In: 45TH HAWAII INTERNATIONAL CONFERENCE, 2012. **Proceedings...** p. 5329-5338, 2012.
- POPPENDIECK, M., POPPENDIECK, T.: **Lean Software Development: an Agile Toolkit for Software Development Managers**. 1st ed. Boston: Addison-Wesley Professional, 2003.
- RAGLAND, B. Measure or indicator: what's the difference? **Crosstalk**, v. 8, n 3, mar, 1995.
- RICHARD, E.; JANE, M. Iterative rework: the good, the bad, and the ugly. **Computer**, vol. 38, p. 34-41, 2005.
- SABBAGH, R. **Scrum: Gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.
- SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
- EDER, S. et al. Diferenciando as abordagens tradicional e ágil de gerenciamento de projetos. **Production**. vol. 18, 2014. No prelo.
- SATO, D. T. **Uso eficaz de métricas em métodos ágeis de desenvolvimento de software**. 2007. Dissertação. Universidade de São Paulo, 2007.
- SCHWABER, K.; SUTHERLAND, J. **O guia Scrum**. Scrum.org, 2013.
- SEAMAN, C.; GUO, Y. Measuring and Monitoring Technical Debt, **Advances in Computers**, v. 82, p. 25-46, 2011.
- SEBRAE. **Anuário do trabalho na micro e pequena empresa**. 5. ed. 2012. Disponível em <<http://www.sebrae.com.br/customizado/estudos-e-pesquisas/temas-estrategicos/emprego/anuario-do-trabalho-na-micro-e-pequena-empresa-1>>. Acesso em: Ago. 2013.
- SOFTWARE Engineering Institute. **CMMI® for Development V1.3**. Pittsburgh, 2010.

SLIGER, M.; BRODERICK, S. **The Software Project Manager's Bridge to Agility**. Boston: Addison-Wesley Professional, 2008.

SOFTEX. **Melhoria de Processo do Software Brasileiro: Guia Geral**. Campinas: SOFTEX, 2011.

SWAMINATHAN, B.; JAIN, K. Implementing the Lean Concepts of Continuous Improvement and Flow on an Agile Software Development Project: An Industrial Case Study. **AGILE India** (AGILE INDIA), 2012. p. 10-19.

RIBEIRO, J.L.D; MILAN, G.S. **Entrevistas individuais: teorias e aplicações**. 2. ed. Porto Alegre: FEENG/UFRGS, 2007.

SCHATZ, B.; ABDELSHAFI, I. Primavera Gets Agile. **IEEE software**, v. 3, p. 7, 2005.

STANDISH Group. **Chaos Manifesto**. 2012. The Standish Group International. Relatório. Disponível em: <<https://secure.standishgroup.com/reports/reports.php>> Acesso em: 06 dez. 2012.

TALBY, D.; DUBINSKY, Y. Governance of an agile software project. In: 2009 ICSE WORKSHOP ON SOFTWARE DEVELOPMENT GOVERNANCE, 2009, Washington, DC. **Proceeding...** Washington, DC, USA, p. 40-45, 2009.

VERSIONONE. **8th State of Agile Survey**. Version One Inc, 2014. Disponível em <<http://www.versionone.com>>. Acesso em: 20 mar. 2014.

WAN, J.; WANG, R. Empirical Research on Critical Success Factors of Agile Software Process Improvement. **Journal of Software Engineering & Applications**, v. 3, n. 12, 2010.

WILLIAMS, L.; COCKBURN, A. Agile software development: it's about feedback and change. **Computer**, v. 36, p. 39-43, 2003.

APÊNDICES

Apêndice 1 – Roteiro de questões do estudo de campo

Apresentação:

Esta entrevista faz parte da pesquisa de doutorado sobre utilização de métricas em projetos de software que utilizam métodos ágeis do programa de Engenharia de Produção da UFRGS. Gostaríamos da sua participação respondendo às questões a seguir. Asseguramos total sigilo das informações repassadas. Assim que concluída a pesquisa, caso haja interesse, serão repassados os resultados para a empresa.

Empresa: _____ Data da entrevista: _____

Entrevistado: _____ Número total de funcionários: _____

Número de pessoas que trabalham especificamente no desenvolvimento de software: _____

QUESTÕES INICIAIS

i) Informações sobre o contexto das organizações e perfil dos entrevistados

Sobre a organização:

Qual o método ágil que adota?

Utiliza métodos ágeis em todos os projetos da empresa?

Sobre o entrevistado:

Qual sua função exercida durante o desenvolvimento dos projetos ágeis?

Qual o seu tempo de experiência com desenvolvimento ágil?

QUESTÕES CENTRAIS

ii) Informações sobre as métricas utilizadas, fases do projeto e funções dos envolvidos nos projetos de software que utilizam métodos ágeis

Quais são as métricas utilizadas para monitorar o desempenho dos projetos que utilizam?

Utilizam as mesmas métricas para todos os projetos ou existem métricas diferentes para projetos de características diferentes?

Quais são as fases típicas de um projeto ágil e o que é realizado em cada fase?

Quais são as funções dos vários envolvidos num projeto ágil de software?

QUESTÃO DE FECHAMENTO

Você tem algo a acrescentar?

Agradecimentos

Apêndice 2 – Respostas da pesquisa de campo sobre as fases dos projetos ágeis

EMPRESA	FASES
A	<ol style="list-style-type: none"> 1) Visão do produto, definição dos requisitos e estimativas iniciais 2) Reunião de planejamento e estimativas da iteração 3) Iterações 4) Reuniões diárias do time para discutir sobre o andamento da iteração 5) Reuniões semanais de acompanhamento da iteração com scrum master, time de desenvolvimento e gerente de projetos 6) Reuniões mensais de monitoramento dos projetos entre todos os envolvidos 7) Reunião de review
B	<ol style="list-style-type: none"> 1) Iniciação (inception): momento de definição do que será o produto, onde faz elaboração de cenários, definição das pessoas envolvidos e função a ser exercida, composição do time de desenvolvimento, verificam se existe alguma data fixa a ser cumprida, definem a MVP (minimum viable product) onde é possível definir uma visão do tamanho e tempo do projeto, e planeja as <i>releases</i> (normalmente de 3 meses). Ou seja, é a fase concepção do produto e do projeto. Após ocorre a definição da infraestrutura e quais ferramentas serão utilizadas. 2) Planejamento da <i>release</i> 3) Reunião de planejamento inicial da iteração 4) Iterações: Inicia-se o processo de desenvolvimento através de iterações (normalmente de 2 semanas). 5) Reuniões diárias da equipe 6) Entrega do produto desenvolvido na iteração. 7) Reunião de retrospectiva da iteração 8) Reunião de avaliação da <i>release</i> 9) Finalização: coloca-se o produto em produção, ou seja, entrega final do produto ao cliente e finalização do projeto.
C	<ol style="list-style-type: none"> 1. Preparação: fase de iniciação onde é construído um mapa geral do projeto 2. Planejamento inicial da iteração: onde é construído o <i>backlog</i> do produto 3. Desenvolvimento: fase das sprints 4. Reunião diária 5. Reviews da Sprint: reunião com o cliente 6. Fechamento: encerramento do projeto
D	<ol style="list-style-type: none"> 1. Definição do produto e sua estratégia (diretoria) 2. Reunião com todos os envolvidos no projeto para definição da visão do produto: time de desenvolvimento, scrum máster, PO e outros interessados no projeto. Nesta reunião o time vai contribuir na definição do produto e na construção das histórias. São definidos além das histórias o seu critério de aceitação e o MVP (Minimum Viable Product). Inicia-se o ciclo tradicional do Scrum 3. Planejamento da sprint 4. Sprints 5. Reuniões diárias 6. Review da sprint 7. Retrospectiva da Sprint <p>Obs. Não fazem reunião de finalização do projeto apenas encerram o projeto.</p>

E	<ol style="list-style-type: none"> 1. Visão do produto e definição dos pacotes de <i>backlog</i> 2. Reunião de <i>kickoff</i> com todos os interessados: momento em que apresentam para o cliente a sua estratégia para garantir a entrega do produto e os riscos do projeto. Apresentam um conjunto de informação ao cliente sobre o projeto e uma visão da arquitetura do software em alto nível, como as mudanças irão ser tratadas, como irá ocorrer a priorização dos requisitos/funcionalidades. Ou seja, nesta reunião são apresentadas as regras do jogo. 3. Planejamento da iteração 4. Iterações 5. Reuniões diárias 6. Reunião de <i>review</i> 7. Retrospectiva: é estabelecido um plano de ação para a próxima iteração o qual é registrado num documento. Internamente essa retrospectiva poderia ser mais informal, mas para os clientes é importante demonstrar o esforço para resolver problemas e identificar a causa raiz dos problemas. 8. Fechamento do projeto
F	<ol style="list-style-type: none"> 1. Iniciação: Reunião de iniciação do projeto, objetivos do produto, principais riscos do produto, o que é desejado, o que não é desejado, ou seja, identificar elementos para compor a visão do produto. Nesta etapa também se faz um <i>backlog</i> inicial, se prioriza este <i>backlog</i>. Também se faz nesta etapa uma visão de arquitetura, onde é necessário desenvolver o ambiente de desenvolvimento e arquitetura, se valida a arquitetura e se prepara o ambiente de desenvolvimento. Após esta fase se começa o desenvolvimento. 2. Planejamento da iteração: definição de metas e informações sobre a sprint 3. Iteração 4. Reunião diária 5. Entrega do produto da iteração (não há finalização do projeto, afirma que o projeto está sempre sendo documento e entregue, então na conclusão do projeto é necessário apenas encerrar o projeto)
G	<p>As fases são as mesmas do Scrum:</p> <ol style="list-style-type: none"> 1. <i>Planning</i> inicial para definição da visão do produto 2. Sprints <i>planning</i> 3. Sprints 4. Reuniões diárias 5. <i>Review</i> da Sprint 6. Retrospectiva
H	<ol style="list-style-type: none"> 1. Iniciação: Fase em que os POs (<i>Product Owner</i>) apresenta para as equipes as necessidades através de histórias escritas por eles. A equipe de desenvolvimento irá refinar e se necessário solicitam mais informações sobre as histórias. Nesta fase de iniciação a equipe faz a análise das histórias, quebra em tarefas e faz as estimativas. 2. Reunião de planejamento com todas as equipes que farão parte do projeto: reunião de apresentação (<i>kickoff</i>) com todas as equipes que farão parte do projeto. Essa reunião é necessária, pois as aplicações são muito grandes e cada equipe trabalha em alguns módulos da aplicação. 3. Reunião de planejamento da Sprint por equipe 4. Iterações: desenvolvimento do produto 5. Reunião diária 6. Reunião de <i>review</i>: Entrega do produto desenvolvido na iteração.

	<p>7. Reuniões de acompanhamento do projeto: semanal, mensal e trimestral. Nas reuniões são apresentados os dados sobre o andamento do projeto e informações sobre o desempenho das métricas.</p> <p>8. Homologação: Após concluir o desenvolvimento é realizada a homologação do produto. Nesta fase existe uma grande preocupação com os testes. Até se estabilizar bem o produto e entregar definitivamente para o cliente</p> <p>9. Reunião de finalização do projeto: avaliação do planejado e realizado, da densidade dos defeitos e suas causas, e outras informações do projeto</p>
E9	<p>1. Reunião de planning: é escrita a visão do produto em <i>roadmaps</i> trimestrais (pré-jogo). A partir dessas definições iniciais irão haver discussões sobre negócio, arquitetura, requisitos não funcionais.</p> <p>2. As <i>releases</i> são planejadas mensalmente com a descrição da visão do produto.</p> <p>3. Reunião de planejamento da iteração onde ocorre a divisão e estimativas das tarefas</p> <p>4. Iterações</p> <p>5. Reunião diária para controle das tarefas.</p> <p>6. Reunião semanal</p> <p>7. Reunião de entrega do produto da iteração</p> <p>8. Reuniões ao finalizar a iteração onde é revisado o <i>planning</i> inicial do projeto.</p> <p>9. Reunião de finalização da release</p> <p>Não ocorre fase de finalização.</p>

Apêndice 3 – Respostas da pesquisa de campo sobre as funções envolvidas nos projetos ágeis

EMPRESA	ENVOLVIDOS E RESPONSABILIDADES
A	<p>PO interno: que representa os clientes durante os projetos. Este papel foi necessário ser criado devido os clientes não poderem estar participando diariamente nos projetos.</p> <p>BPO (Business PO ou PO da área de negócio): representante do cliente que é responsável pela visão estratégica do negócio e do produto. Se comunica com o PO para conseguir alinhar visão de negócio com visão do produto que está sendo desenvolvido.</p> <p>Time de desenvolvimento: responsável pela construção do produto propriamente dito e no monitoramento da <i>sprint</i>.</p> <p>Scrum master: interage com o gerente de projetos, verificar a demanda, fazer a classificação e define a estimativa dos projetos. Para realizar as estimativas a empresa adota uma métrica própria de baseado no histórico de projetos anteriores, tipo da tarefa e complexidade.</p> <p>Gerente de projetos: responsável pelo controle de projeto e no monitoramento das métricas, e realização de intervenções caso ocorram problemas ou o tempo estimado esteja fora do tempo realizado.</p>
B	<p>Time de desenvolvimento: responsável pelo desenvolvimento do produto, o qual é composto por programadores, analista de testes (escrever casos de testes, testar e fazer automatização de testes), analistas de negócios e outros profissionais técnicos que forem necessários. O time não se envolve com a parte comercial e ou financeira. Fora isso a responsabilidade é de todo mundo. Ou seja, se um dos papéis que estão envolvidos no projeto não cumprir com a sua responsabilidade irá ocorrer problemas no projeto.</p> <p>Líder da equipe: em um projeto grande irá trabalhar na função de gerente de projetos mais interno, e que irá trabalhar diretamente com o time durante as iterações. Atua como membro do time.</p> <p>Gerente de projetos: em projetos grandes é necessário ter também um gerente de projetos. É responsável pelo controle dos recursos, controle dos riscos, e por monitorar e controlar o seu desempenho.</p> <p>Business analytics: responsável por escrever e analisar as histórias (papel similar ao PO)</p>
C	<p>PO representando o cliente, o qual quer saber se o prometido foi cumprido e também deseja ser informado caso ocorram riscos ao projeto.</p> <p>Scrum Master que atua como líder da equipe do time de desenvolvimento. O líder da equipe é responsável pela coleta das métricas.</p> <p>Time de desenvolvimento: responsável pela construção do produto e monitoramento da iteração.</p>

	<p>Além disso, todo projeto possui um gerente de projeto, e um diretor responsável pelo acompanhamento. O gerente de projetos atua como consultor da direção e acompanha o dia a dia da equipe de desenvolvimento. Possui visão gerencial e organizacional e preocupa-se com o atendimento dos objetivos, análise dos riscos e monitoramento das métricas.</p> <p>O diretor possui a visão gerencial e não se envolve com questões cotidianas.</p>
D	<p>Todos os membros recomendados pelo Scrum: Scrum master, time de desenvolvimento e PO. Também definem o BO (Business Owner) que é o dono do produto e irá se preocupar com questões de estratégias de negócio.</p> <p>Diretoria: possui interesse financeiro.</p> <p>O BO e a diretoria não possuem a visão do projeto de forma ágil e sim de forma tradicional. Neste nível perde-se os princípios de agilidade.</p>
E	<p>Representante do cliente: responsável pelo <i>backlog</i> do produto</p> <p>Scrum master: atua junto com a equipe e buscar garantir o sucesso da iteração.</p> <p>Gerente de projetos: responsável pelo projeto</p> <p>Time de desenvolvimento: responsável pelo desenvolvimento e pela qualidade do produto. Possuem a restrição de tempo, porém na visão de uma meta de trabalho e não com a preocupação de custos.</p>
F	<p>Cliente.</p> <p>Time de desenvolvimento.</p> <p>Líder da equipe.</p>
G	<p>Gerente ágil de projetos que atua também como gerente de portfólio, pois ele participa de todos os projetos. Preocupa-se com as métricas e faz as cobranças necessárias aos envolvidos. Sua principal preocupação é gerenciar/monitorar riscos. Possui uma postura de liderança e não de controle.</p> <p>Time de desenvolvimento.</p> <p>Cliente/PO</p> <p>Diretor de estratégia: responsável pelo contato com o cliente.</p>
H	<p>Diretoria e área comercial: possui a visão cliente/negócio e se preocupa cumprimento do prometido e realizado de forma geral, monitoramento dos custos e prazo. Não vê o <i>burndown</i>, mas sim os indicadores de negócio do BSC.</p> <p>Gerente de projetos que é também atua como Scrum dos Scrums, pois são muitas equipes envolvidas num mesmo projeto. Faz reuniões semanais só com os Scrums master para saber como está a semana. Se preocupa tanto com a equipe, scrum master e PO, e se preocupa em relação aos objetivos organizacionais da empresa, com os riscos do projeto e com o resultado das medições.</p> <p>Scrum master do projeto: se preocupa se a iteração está no prazo e a questão da qualidade, pois isto impacta na equipe deles.</p> <p>POs - que são membros da empresa, porém com grande conhecimento de negócio e também conhecem sobre desenvolvimento. Se preocupa em saber se o time de desenvolvimento fez o que ele precisava (visão do produto conforme objetivos do negócio).</p> <p>Equipe de desenvolvimento: se preocupa com o cumprimento do desenvolvimento da tarefa no prazo estabelecido.</p> <p>Gerente de medição: se preocupa com o plano de medição e a eficaz utilização das métricas. Foi adotado em função do CMMI.</p>

I	Time de desenvolvimento Gestor do projeto Diretoria Cliente ou representante do cliente (PO)
---	---

Apêndice 4 – Padrão das legendas utilizadas para definição das fórmulas das métricas

A=administração

ab = aberto

ac=acurácia

Bn = *Burnup*

Bw = *Burndown*

c=concluídas

cli=cliente

COB = cobertura

C=custo

CS_TST = casos de testes

D=desenvolvimento

d=dias / diário

dd=densidade

Df = defeitos

Dv=dívida

e= esforço

ex=execução

E=estimado

Eq = Equipe

f=fase

F=funcionalidade(s)

F_Ent = Funcionalidades entregues

G=grau

H = histórias

h = horas

I=impedimentos

In=índice

i=iteração

MOT = Motivação

N = número

P = pontos

Pl = planejada

p=projeto

R=realizado

r=requisito

RC = receita

RF=requisitos funcionais

S=satisfação

t=tempo

Tc=técnica(o)

TS=tarefas

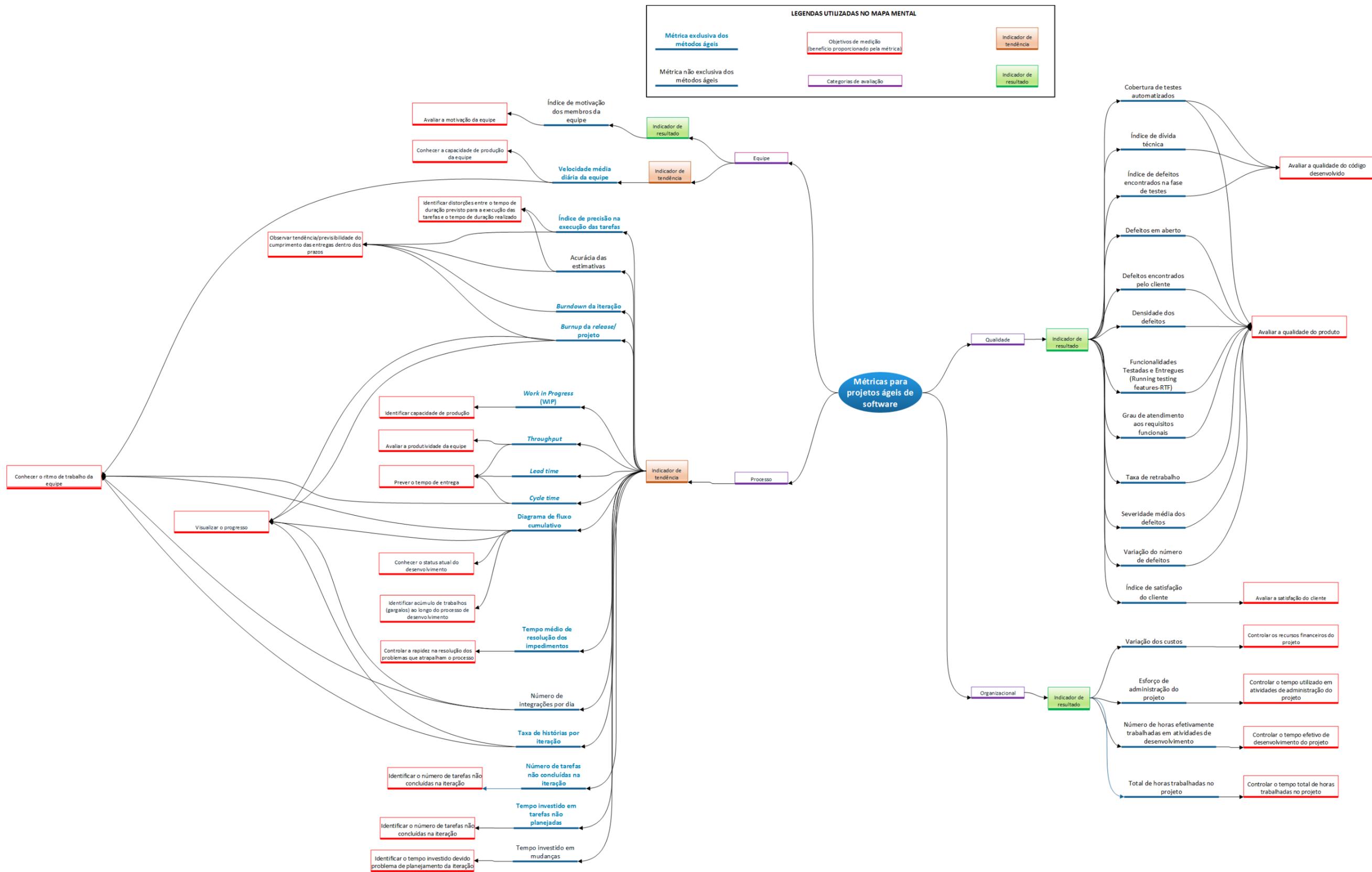
TST = testes

TST_A = testes automatizados

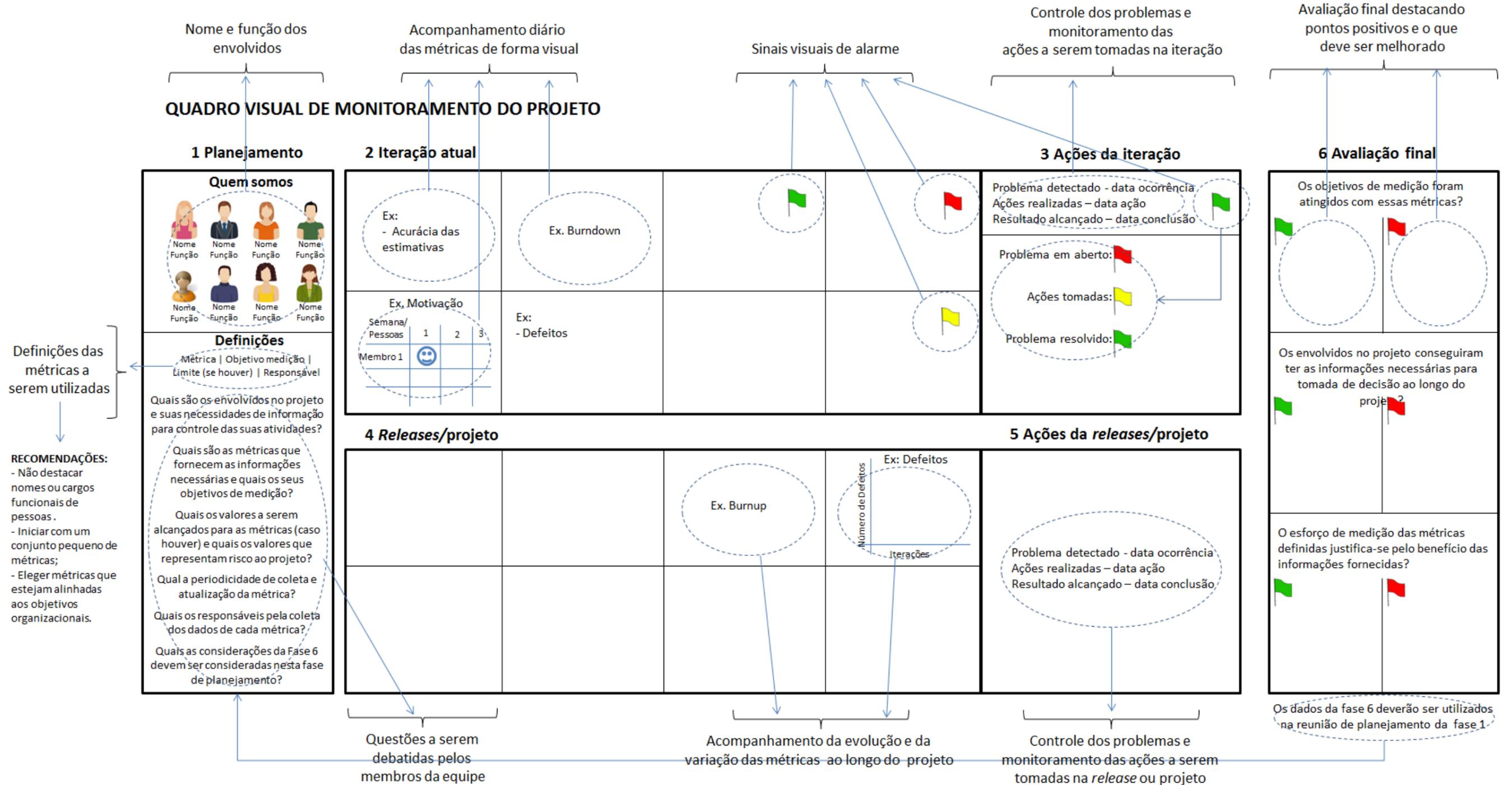
TT=total

W=trabalho

Apêndice 5 – Mapa mental



Apêndice 6 – Quadro visual de monitoramento do projeto – Com informações sobre as etapas e com instruções de preenchimento



Apêndice 7 – Quadro visual de monitoramento do projeto – Em branco pronto para uso

QUADRO DE MONITORAMENTO DO PROJETO

1 Planejamento	2 Iteração atual				3 Problemas e ações da iteração	6 Avaliação final	
Quem somos						Os objetivos de medição foram atingidos com essas métricas?  	
Definições						Os <i>stakeholders</i> conseguiram ter as informações necessárias para tomada de decisão ao longo do projeto?  	
<div style="display: flex; justify-content: space-between;"> 4 Projeto / Release 5 Problemas e ações do projeto/release </div>							
						O esforço de medição das métricas definidas justifica-se pelo benefício das informações fornecidas?  	

Apêndice 8 – Roteiro de questões para avaliação do processo proposto

Empresa: _____ Data da entrevista: _____

Entrevistado: _____ Número total de funcionários: _____

Cargo exercido: _____

1. Perguntas iniciais

Para especialistas do mercado:

Existe na empresa um processo de medição formalizado?

Quais são os seus pontos fortes e fracos?

Para especialistas da academia:

Você tem conhecimento de processo de medição específico para métodos ágeis de desenvolvimento de software?

Se sim, quais são os seus pontos fortes e fracos?

2. Roteiro da apresentação do processo de medição proposto

Apresentação dos problemas e dificuldades observados sobre definição de métricas e de processo de medição em empresas de desenvolvimento de software que utilizam métodos ágeis.

Critérios adotados para estabelecer um processo de medição compatível com a abordagem ágil de desenvolvimento de software.

Apresentação do processo proposto, do conjunto de métricas e suas representações.

3. Avaliação do processo proposto frente aos requisitos previamente definidos

Avalie o processo de medição proposto frente aos seguintes critérios					
	Discordo totalmente				Concordo totalmente
	1	2	3	4	5
Quanto à compreensão do processo, você o considera simples de ser entendido					
	1	2	3	4	5
Quanto à implantação do processo, você o considera fácil de ser implantado					
	1	2	3	4	5

Quanto ao uso do processo, ele não agrega novas etapas ao processo de desenvolvimento do software na sua empresa					
	1	2	3	4	5
Quanto ao uso do processo, ele não exige a criação de novos cargos ou funções					
	1	2	3	4	5
Quanto aos benefícios do processo, fornece orientações que permitem à empresa refletir sobre os seus objetivos de medição e necessidades de controle através do uso de métricas					
	1	2	3	4	5
Quanto aos benefícios do processo, permite monitorar as métricas ao longo do projeto					
	1	2	3	4	5
Quanto à comunicação, o processo permite acesso às informações para todos os interessados de forma rápida e simples					
	1	2	3	4	5
Quanto ao alinhamento com a abordagem ágil, respeita os valores e princípios ágeis					

4. Perguntas finais

Qual a sua opinião sobre o processo de medição proposto?

Qual a sua opinião sobre a especificação das métricas e instrumentos de apoio (mapa mental, quadro métricas por fase, descrição técnica das métricas, quadro visual)?

Qual seria a utilidade do processo proposto para as empresas?

5. Agradecimentos

Apêndice 9 – Nomenclatura das métricas apresentadas da revisão da literatura/pesquisa de campo e nomenclatura consolidada

Nomenclatura da revisão da literatura	Nomenclatura consolidada
Acurácia (precisão) das estimativas	Acurácia das estimativas (tempo)
<i>Burndown</i> da iteração	<i>Burndown</i> da iteração
<i>Burnup</i> da <i>release</i> /projeto	<i>Burnup</i> da <i>release</i> /projeto
Cobertura de testes automatizados	Cobertura de testes automatizados
Comparação do número de defeitos encontrados no ciclo atual com outros ciclos do desenvolvimento	Variação do número de defeitos entre iterações
Custos	Variação dos custos (entre estimado e realizado)
<i>Cycle time</i>	<i>Cycle time</i> (tempo de ciclo para concluir uma tarefa)
Defeitos em aberto	Defeitos em aberto (defeitos não solucionados)
Defeitos encontrados na fase de testes	Índice de defeitos encontrados na fase de testes
Defeitos encontrados pelo cliente	Defeitos encontrados pelo cliente
Densidade dos defeitos	Densidade dos defeitos (proporção do número de defeitos em relação ao tamanho do projeto)
Diagrama de fluxo cumulativo	Diagrama de fluxo cumulativo
Dívida técnica	Taxa de dívida técnica (problemas técnicos a serem resolvidos)
Eficiência na resolução dos impedimentos (impedimentos removidos)	Tempo médio de resolução dos impedimentos
Fluxo das tarefas no quadro Kanban	
Frequência das <i>releases</i>	
Funcionalidades Testadas e Entregues (<i>Running testing features-RTF</i>)	Funcionalidades Testadas e Entregues (<i>Running testing features-RTF</i>)
Gargalos do fluxo de tarefas	
Grau de atendimento aos requisitos funcionais	Grau de atendimento aos requisitos funcionais
<i>Lead time</i>	<i>Lead time</i> (tempo de entrega das funcionalidades ao cliente)
Motivação da equipe (<i>Happiness</i>)	Índice de motivação dos membros da equipe
Mudanças nos requisitos	Tempo investido em mudanças
Número de casos de teste	
Número de histórias não concluídas na iteração	Número de tarefas não concluídas na iteração
Número de horas trabalhadas em atividades de	Número de horas efetivamente trabalhadas em

desenvolvimento	atividades de desenvolvimento
Número de integrações por dia	Número de integrações por dia (tarefas concluídas e integradas no sistema de controle de versões)
Pesquisa de satisfação com o cliente	Nível de satisfação do cliente
Precisão na execução das tarefas	Taxa de acerto na estimativa das tarefas
Retrabalho com erros por iteração	Taxa de retrabalho
Severidade dos defeitos por iteração	Severidade média dos defeitos
Tarefas não planejadas	Tempo investido em tarefas não planejadas
Taxa de histórias por iteração	Índice de histórias por iteração
Total de horas consumidas no projeto	Total de horas trabalhadas no projeto
<i>Throughput</i>	<i>Throughput</i> (número de tarefas entregues por período de tempo)
Total de esforço estimado	
Total de horas consumidas em atividades de administração do projeto	Esforço de administração do projeto (% tempo despendido em atividades de administração do projeto)
Retorno sobre o investimento	
Velocidade da equipe	Velocidade média diária da equipe
<i>Work in Progress</i>	<i>Work in Progress</i>