ANA PAULA PEREIRA DE MORAES RESS

UM ESTUDO SOBRE A INTEGRAÇÃO CMMI E DESENVOLVIMENTO ÁGIL

São Paulo
2013

ANA PAULA PEREIRA DE MORAES RESS

UM ESTUDO SOBRE A INTEGRAÇÃO CMMI E DESENVOLVIMENTO ÁGIL

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências

Área de Concentração:
Engenharia de Produção

Orientador: Prof. Dr. Renato de Oliveira Moraes

São Paulo
2013

FICHA CATALOGRÁFICA

Ress, Ana Paula Pereira de Moraes
Um estudo sobre a integração CMMI e desenvolvimento ágil / A P.P.M.Ress. – São Paulo, 2013.
p. 130

Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Qualidade de software 2.Métodos ágeis 3.Métodos de desenvolvimento de software I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.

Aos meus pais, João Luiz e Cresci, pelo apoio aos estudos;
ao meu marido Weber e filhos, Marina e Ian, pelo carinho e
compreensão nos momentos de ausência, necessários
para a conclusão deste sonho.

AGRADECIMENTOS

Aprendi que a escrita de uma dissertação é um processo solitário a que todo pesquisador está destinado, porém é fruto de um trabalho que reúne a contribuição de diversas pessoas que colaboram na sua composição. Assim, dirijo meus agradecimentos:

ao Professor Doutor Renato de Oliveira Moraes, orientador da dissertação, pelo apoio, partilha do saber e pelas valiosas contribuições para o trabalho. Acima de tudo, obrigada pela sua disposição em apresentar-me ao campo da docência, experiência enriquecedora e muito gratificante;

aos professores doutores Fernando José Barbin Laurindo e Linda Lee Ho, pelo apoio aos processos administrativos necessários para conclusão deste curso;

aos professores do programa de pós-graduação em Engenharia de Produção, pela dedicação ao orientar e estimular o interesse pelo conhecimento e vida acadêmica;

aos responsáveis organizacionais que autorizaram a realização deste estudo, por dispor seu valioso tempo para responder ao questionário;

à revisora desta dissertação, Mônica de Moraes Oliveira, por conferir a este trabalho maior clareza, concisão e harmonia;

aos meus familiares, pelo incentivo recebido ao longo destes anos e pelo estímulo durante a jornada de estudos;

a todas as pessoas que contribuíram para a concretização desta dissertação.

RESUMO

É possível notar no trabalho diário das empresas que a pressão por diminuir o prazo de entrega dos *softwares* ainda é uma realidade. Neste contexto, empresas adotam técnicas de desenvolvimento ágil de *software*, implementando entregas constantes e estabelecendo ciclos iterativos e curtos no processo de desenvolvimento de *software*. Mas, se por um lado estas técnicas visam inovar e reduzir tempo de desenvolvimento, por outro, as empresas sentem-se pressionadas a adotar a forma tradicional de manter a qualidade de *software*. Neste ponto, o CMMI se apresenta como um modelo de maturidade de reconhecimento mundial e é classificado na abordagem tradicional por pregar grande formalismo e foco no processo. A abordagem ágil coloca que os indivíduos e suas interações têm maior importância que os processos e as ferramentas e que ser adaptável às mudanças é mais importante do que seguir a rigor o planejado. Ao integrar estes dois modelos de abordagens distintas espera-se que as práticas ágeis possam ajudar organizações maduras a se tornarem mais flexíveis. Com o intuito de averiguar esta integração, utilizou-se o método de pesquisa de estudo de caso, de modo a se aprofundar nas empresas que possuem certificação CMMI e decidiram adotar o modelo de desenvolvimento ágil. O objetivo é levantar como estas empresas resolveram os pontos de conflito da abordagem tradicional e ágil, de modo a se tornarem mais eficazes. O estudo ainda contém um quadro de integração CMMI *versus* desenvolvimento ágil de modo a elencar os pontos que merecem maior atenção e esforço de adaptação, visando a concretização da integração.

Palavras-chave: CMMI. Modelo de Maturidade. Desenvolvimento Ágil. Processo de Desenvolvimento de Software.

ABSTRACT

It’s possible to notice in the daily work of companies that pressure by decreasing the delivery of software is still a reality. It is in this contexto that companies adopt the techniques of agile software development, implementing and establishing constant deliveries and short iterative cycles in the processo f software development. But if on the onde hand these techniques aim to innovate and reduce development time, on the other, companies feel pressured to adopt the traditional way of maintaining the quality of software. At this point, the CMMI is presented as maturity model worldwide recognition, classified in the traditional approach for preaching extensive formalism and focus on the process, as the agile approach puts individuals and their interactions is more importante than processes and tools, whre being adaptable changes is more importante than following the strict what was planned. By integrating these two models of diferente approaches, it is expected that agile practices can help organizations mature to become more flexible. In order to investigate this integration is that this study uses the research method of case study so as to deepen the companies that have decided to adopt CMMI and agile development model. The goal is to raise as these companies solved the trouble spots of the traditional approach and agile in order to become mor effective. The study also contain a framework for integrating CMMI vs. Agile development in order to list the points that desserve more attention and effort to adapt in order to achieve integration.

Keywords: CMMI. Maturity Model. Agile Development. Software Development Process.

LISTA DE ILUSTRAÇÕES

Figura 1 – Melhoria contínua 21
Figura 2 – História CMMI 23
Figura 3 – Hierarquia dos princípios *Lean* 26
Figura 4 – Comparação do fluxo de projeto e desenvolvimento de produto 28
Figura 5 – Comparação de ciclo de vida de software 29
Figura 6 – Primeira Etapa da busca CMMI 33
Figura 7 – Terceira Etapa: distribuição por ano da busca CMMI 35
Figura 8 – Terceira Etapa: distribuição do assunto por ano da busca CMMI 35
Figura 9 – Primeira Etapa: busca por Lean Development 37
Figura 10 – Terceira Etapa: distribuição por ano da busca por *Lean Development* . 38
Figura 11 – Terceira Etapa: distribuição do assunto por ano da busca por *Lean Development* 39
Figura 12 – Primeira Etapa: resultado da busca por *Agile* 40
Figura 13 – Terceira Etapa: distribuição por ano da busca por *Agile* 43
Figura 14 – Terceira Etapa: distribuição de assuntos por anos por busca *Agile* 44
Figura 15 – *Framework* CMMI 48
Figura 16 – Estrutura básica do CMMI 49
Figura 17 – Representação contínua do CMMI 50
Figura 18 – Princípios da gestão ágil de projetos 56
Figura 19 – Decomposição do problema de adaptação parcial do método ágil 61
Figura 20 – Modelo AAIM (*Agile Adoption and Improvement Model*) 65
Figura 21 – Critérios para tipo de pesquisa 76
Figura 22 – Etapas da pesquisa 78
Figura 23 – Critérios de Qualidade dos projetos de pesquisa 79
Figura 24 – Evolução da metodologia de desenvolvimento da Empresa A 84
Figura 25 – Divisão de tarefas da Empresa A 86
Figura 26 – Empresa A: Aderência ao Scrum no início de 2008 92
Figura 27 – Empresa A: Aderência ao Scrum em setembro de 2009 93
Figura 28 – Empresa A: Defeitos de Homologação / Kloc 93
Figura 29 – Empresa A: Defeitos de Produção / Kloc 94

Figura 30 – Empresa A: Nível de satisfação 95
Figura 31 – Empresa B: Estrutura organizacional 96
Figura 32 – Empresa C: Mapa geral dos macro processos 104
Figura 33 – Empresa C: Processo..... 105

LISTA DE TABELAS

Tabela 1 – Hipótese básica 17
Tabela 2 – Fontes do CMMI 23
Tabela 3 – Comparação Manufatura Lean x Desenvolvimento Lean 27
Tabela 4 – Segunda Etapa para busca CMMI..... 33
Tabela 5 – Resultado da Segunda Etapa para busca CMMI 34
Tabela 6 – Segunda Etapa: classificação da busca por Lean Development 37
Tabela 7 – Segunda Etapa: resultado da busca por Lean Development 37
Tabela 8 – Segunda Etapa: classificação por assunto por busca Agile 40
Tabela 9 – Segunda Etapa: resultado da busca por Agile 42
Tabela 10 – Melhorias percebidas ao acrescentar práticas ágeis à abordagem tradicional 57
Tabela 11 – Comparação gestão ágil x tradicional..... 58
Tabela 12 – Subproblemas da adaptação parcial do método ágil 61
Tabela 13 – Parâmetros que afetam a agilidade..... 62
Tabela 14 – Quatro Dimensões 4-DAT 63
Tabela 15 – Quadro de integração CMMI x Desenvolvimento Ágil 70
Tabela 16 – Quadro de integração CMMI x Desenvolvimento Ágil 115
Tabela 17 – Objetivos genéricos e especifícos do CMMI..... 125

SUMÁRIO

1. INTRODUÇÃO……12
1.1. O PROBLEMA ESTUDADO……..15
1.2. HIPÓTESES……16
1.3. OBJETIVOS……. 18
1.4. RELEVÂNCIA DO TEMA E JUSTIFICATIVAS…….. 18
1.5. ESTRUTURA DO TRABALHO……. 20
2. CONTEXTO DO ESTUDO……21
2.1. MELHORIA CONTÍNUA……21
2.2. CMMI……22
2.3. LEAN DEVELOPMENT……..25
2.4. MÉTODOS ÁGEIS……..28
3. ANÁLISE BIBLIOMÉTRICA……..32
3.1. CMMI……32
3.1.1. Primeira Etapa: Uso da base Web of Science……32
3.1.2. Segunda Etapa: Classificação das publicações…….. 33
3.1.3. Terceira Etapa: Análise das Publicações……..34
3.2. LEAN DEVELOPMENT……..36
3.2.1. Primeira Etapa: Uso da base Web of Science……36
3.2.2. Segunda Etapa: Classificação das Publicações……..37
3.2.3. Terceira Etapa: Análise das Publicações……..38
3.3. MÉTODOS ÁGEIS……..39
3.3.1. Primeira Etapa: Uso da base Web of Science……40
3.3.2. Segunda Etapa: Classificação das Publicações……..40
3.3.3. Terceira Etapa: Análise das Publicações……..42
4. PESQUISA BIBLIOGRÁFICA……..46
4.1. CMMI……46
4.2. DESENVOLVIMENTO ÁGIL……..55
4.3. LEAN DEVELOPMENT……..65
4.4. QUADRO DE INTEGRAÇÃO CMMI VERSUS DESENVOLVIMENTO ÁGIL68
5. METODOLOGIA……76

5.1. PREPARAÇÃO…… 78
5.2. DESENVOLVIMENTO…… 79
5.2.1. Desenvolvimento do Questionário……..80
5.3. FINALIZAÇÃO……..82
6. ESTUDOS DE CASOS……. 83
6.1. EMPRESA A……83
6.1.1. Sobre a Empresa…….. 83
6.1.2. O Processo da Empresa……..85
6.1.3. Respostas do Questionário…… 87
6.1.4. Considerações do Caso……...91
6.2. EMPRESA B……95
6.2.1. Sobre a Empresa…….. 95
6.2.2. O Processo da Empresa……..96
6.2.3. Respostas do Questionário…… 98
6.2.4. Considerações do Caso……101
6.3. EMPRESA C……..102
6.3.1. Sobre a Empresa…… 102
6.3.2. O Processo da Empresa……103
6.3.3. Respostas do Questionário……...105
6.3.4. Considerações do Caso……109
6.4. SÍNTESE DOS ESTUDOS DE CASOS……110
7. CONSIDERAÇÕES FINAIS……... 112
REFERÊNCIAS BIBLIOGRÁFICAS……121
APÊNDICE 1 - CMMI-DEV V1.3……...125

1. INTRODUÇÃO

Segundo McMahon (2005), organizações que empregaram um grande esforço na melhoria dos seus processos calcados no modelo CMMI (*Capability Maturity Model Integration*) agora também acreditam que abordagens ágeis possam prover incrementos de melhorias. Turner e Jain (2002) acrescentam que, apesar da existência de características distintas entre os métodos ágeis e o modelo CMMI, ambos os métodos possuem planos específicos para o desenvolvimento de *software* e buscam o melhor para que a organização possa produzir *softwares* com qualidade. Lindvall et al. (2004) acrescentam que, quando os desenvolvedores são pressionados a entregar rapidamente um produto para capturar uma oportunidade de mercado, eles devem lutar duas batalhas ao mesmo tempo: uma para desenvolver um produto em certo tempo; outra para atingir os requisitos de qualidade do sistema. Encontrar formas alternativas para desenvolvimento de *software* mais rápido e mais flexível sem comprometer os padrões de qualidade das organizações tem se tornado essencial.

De acordo com Porter (1993), a competitividade depende da capacidade de a empresa inovar e melhorar. As empresas atingem alguma vantagem competitiva por meio das iniciativas de inovação no seu sentido mais amplo, abrangendo novas tecnologias e novas maneiras de execução. Segundo Megliorini e Guerreiro (2004), um dos requisitos de competividade é o baixo custo de produção, sendo a diretriz para uma gestão de custos eficiente e confiável, colaborando com a formação de um preço adequado, que seja competitivo e que apresente uma rentabilidade satisfatória para empresa.

Nesse ambiente competitivo, caracterizado por concorrência em nível global, com consumidores cada vez mais exigentes e sofisticados, e por mudanças tecnológicas mais rápidas, diversificadas e transformadas, as empresas estão sujeitas a pressões constantes para melhoria da qualidade dos produtos. Segundo Bartié (2002), o CMMI é um modelo que possui os elementos necessários para tornar um processo de desenvolvimento de *software* mais eficiente e controlado, construído para integrar diferentes modelos criados a partir do sucesso alcançado pelo SW-CMM.

Com intuito de descobrir quais seriam as motivações das organizações ao adotarem o CMMI, Staples e Niazi (2008) demonstraram, por análise de Pareto, que 80% das empresas estariam presentes em pelo menos uma das razões (em ordem de frequência), sendo que: 66% eram motivadas pela qualidade do *software*; 55% pelo tempo de desenvolvimento; 55% pelo custo do desenvolvimento; 41% pela produtividade; 30% pela visibilidade do processo; 21% pela demanda do cliente; 20% pela vantagem de mercado e 18% pela melhoria no processo de *software* (SPI). Por sua análise de alto nível, colocam que as razões mais frequentes para empresas adotarem o CMMI são: melhoria do produto e processo.

Ahern, Clouse e Turner (2008) acrescentam que o CMMI é utilizado como guia e suporte para atividades gerenciais e técnicas através dos objetivos citados: criar produtos e serviços de qualidade; enriquecer o dia a dia dos funcionários; aumentar a satisfação do cliente; implantar economia de custo e melhores práticas, além de encorajar a medição como ferramenta de gerência. Fora esses objetivos internos à organização, Hansson et al. (2006) ressaltam que, em alguns casos, a escolha pelo modelo CMMI pode ser imposta por clientes externos que, em sua maioria, procuram empresas fornecedoras de *software* que possuam certificação CMMI devido ao reconhecimento mundial. Para esse ponto, Abrahamsson et al. (2002) reforça a necessidade de a certificação ser realizada por um órgão regulamentador, de forma a assegurar legitimidade, credibilidade e confiabilidade de que o processo estabelecido, de fato, está submetido ao modelo de maturidade CMMI.

Uma desvantagem citada por Lazzari (2006) é o grande volume de documentação que o modelo CMMI exige, tendo como premissa de implementação o alto conhecimento técnico a respeito do modelo. Por este motivo, Kulpa e Johnson (2008) acrescentam que muitas empresas decidem investir na contratação de consultorias e em treinamento aos funcionários, tornando os custos de implementação do modelo impraticáveis para pequenas organizações. Outro ponto de desvantagem, destacado por Suikki, Tromsteedt e Haapasalo (2006), está nos ambientes caracterizados por incertezas e grandes desafios, as técnicas tradicionais de gestão de projetos têm apresentado limitações que, segundo Senhar (2001), estão mais presentes nas atividades de planejamento e controle de projeto.

Boehm e Turner (2005) notaram que, a partir de 2000, observou-se um ritmo acelerado de mudanças e inovações na tecnologia da informação nas organizações,

caracterizando-se, assim, ambientes com maiores incertezas e grandes desafios. Esse ritmo acelerado de mudanças tem causado frustrações crescentes com atualização de planos, especificações e documentações durante o processo de desenvolvimento que muitas vezes são mais decorrentes dos critérios de conformidade dos modelos de maturidade do que pela necessidade do processo de desenvolvimento de *software*. O desprendimento de esforço no cumprimento dessas atividades muitas vezes reflete no tempo de entrega do projeto e sua consequente postergação.

Por considerar que as metodologias tradicionais eram pouco adaptativas, em fevereiro de 2001, um grupo de desenvolvedores propôs uma metodologia alternativa para projetos de *software*, fundamentado em uma abordagem intitulada de "Gerenciamento Ágil de Projetos", também conhecida pela sigla APM (proveniente do termo inglês *Agile Project Management*). Trata-se de um documento publicado na Internet (http://agilemanifesto.org/) intitulado "Manifesto Ágil para Desenvolvimento de *Software*", no qual se questiona a eficácia das técnicas e métodos do gerenciamento tradicional de gestão aplicados a projetos que envolvem incertezas e estão sujeitos a alterações provenientes do negócio.

Segundo Mary e Tom Poppendieck (2003), as metodologias ágeis tiveram como origem os princípios do Desenvolvimento *Lean* que, por sua vez, é a aplicação dos princípios da *Toyota Product Development System*. O pensamento *Lean* tem como princípio atender bem o cliente, sem haver desperdício no processo, oferecendo entregas rápidas com qualidade, respeitando as pessoas que colaboram no processo e o compromisso de difundir o conhecimento, sendo que o todo deve ser oferecido no momento certo a volume de informações adequadas de forma a eliminar o desperdício e focar em sua otimização. Liker (2004) complementa que agilidade é frequentemente relacionada ao conceito *Lean* por possuir configurações e objetivos mútuos.

Segundo Highsmith (2004), as empresas precisam desenvolver uma cultura que promova a adaptação para absorver mudanças, ter algumas regras que encorajem a auto-organização, combinada com autogestão; colaboração intensa e iteração entre os membros da equipe. Qumer e Henderson-Sellers (2008) colocam a importância de tratar o desenvolvimento de *software* ágil como um dos pilares para a melhoria organizacional, o que pode ser reforçado pelo conceito de agilidade estabelecido por

Gould (1997) que define agilidade como uma habilidade de a empresa prosperar num ambiente de rápidas e imprevisíveis mudanças.

Inserido neste contexto de controvérsias e compatibilidades entre CMMI e o ambiente de desenvolvimento ágil, este trabalho se propõe a estudar como as empresas resolveram os pontos de conflito da abordagem tradicional e ágil, de modo a se tornarem mais eficazes.

1.1. O PROBLEMA ESTUDADO

Segundo Lycett et al. (2003), em 1968 e 1969, houve a conferência da Organização do Tratado do Atlântico Norte para tratar da "crise do *software*". Foram discutidos problemas como: aumento de tamanho e complexidade dos sistemas de *software*; dificuldade em estimar custos e gerenciar grande número de pessoas. Há evidências de que esses problemas ainda persistem, pois em suas pesquisas os autores evidenciaram que: 80% a 90% dos *softwares* não atingem o objetivo de desempenho; 80% dos sistemas são entregues tardios e acima do orçamento; em torno de 40% do desenvolvimento falhou ou foi abandonado; menos de 25% dos sistemas atingiram os objetivos de negócio e tecnologia; somente 10% a 20% foram considerados como de sucesso.

Lindvall et al. (2004) identificaram que, em grandes organizações, a equipe de desenvolvimento de *software* continua sendo o grande gargalo para aumento da produtividade no que tange à manutenção e melhoria da qualidade. Talvez seja por isso que muitas organizações estejam à procura de novas formas de trabalho para a equipe de desenvolvimento de *software*. Esses autores complementam que atualmente há muita controvérsia a respeito da utilização dos métodos ágeis, por remeter sua eficiência a determinados ambientes e para tipos específicos de projetos. Essa situação dificulta a utilização dessa abordagem em grandes organizações que geralmente procuram evidências de sua aplicação antes de adotá-la devido à complexidade de integrá-la a um ambiente complexo, caracterizado por tecnologia e processos existentes. As pequenas e médias organizações têm mostrado interesse pela abordagem ágil como alternativa à metodologia de

desenvolvimento tradicional, vista como ineficiente, burocrática e inflexível e esperam, assim, produzir mais a custos menores.

Lycett et al. (2003) observaram que as empresas têm investido significativamente em atividades de engenharia, como na série ISO 9000, modelo CMM; no uso de UML (*Unified Modeling Language*) e RUP (*Rational Unified Process*) como ferramentas de modelagem. Essas ações evidenciam a utilização da abordagem tradicional, que visa explicitamente processos definidos que possam ser padronizados pela organização. Contudo, o desenvolvimento ágil tem questionado essas ações por enfatizar o aspecto humano e iterações perante ferramentas e processos; o trabalho de *software* (codificação) sob a documentação; colaboração do cliente perante a negociação contratual e respostas rápidas às mudanças ao invés de seguir o planejado.

Turner e Jain (2002) acreditam que práticas ágeis possam contribuir com a melhoria de processos e sugerem que essas práticas sejam incorporadas ao modelo CMMI. Reforçando essa proposição, Port e Bui (2009) propõem que pontos fortes do desenvolvimento ágil e tradicional sejam combinados e sugerem uma nova estratégia de priorização de requisitos.

O princípio ágil, presente no Manifesto Ágil, é visto como um dos alicerces deste estudo. Os métodos ágeis (como, por exemplo, o XP, *Scrum*, ou *Crystal*), apesar de sua importância, na medida em que são materializações do princípio ágil, têm um papel apenas acessório. Esta dissertação tem seu foco na integração dos princípios ágeis e não dos métodos, com o modelo CMMI. Weber e Wild (2004) mencionam que os princípios ágeis são os mesmos princípios *Lean*, aplicados no campo da engenharia de *software* e gerenciamento de projetos, denominados como desenvolvimento ágil, enfatizados por Venugopal (2005), por expor que métodos ágeis ou *lean* representam a mais recente metodologia de desenvolvimento para gerenciar projetos de TI.

1.2. HIPÓTESES

A questão básica a ser respondida pela dissertação é: como as empresas que adotaram o CMMI e se interessam pela abordagem ágil estão empregando os princípios ágeis?

A hipótese básica a ser formulada é que as empresas devem ter desprendido um esforço maior em adaptar as atividades que exigem maior documentação, métrica e controle já que, por definição, a abordagem ágil prega menor burocratização e formalismo. Tendo em vista que o CMMI divide suas atividades em áreas de processos que possuem categorias (gerenciamento de projeto, gerenciamento de processo, engenharia e suporte), é possível elencar de quais áreas de processos se espera maior esforço de adaptação. Foi construída a Tabela 1, na qual, por categoria do CMMI, é exposto de quais áreas de processos se espera certo impacto (por exemplo: na categoria gerenciamento de processo o esperado é que todas as áreas de processos sejam impactadas, uma vez que os princípios ágeis não possuem foco no processo, mas, por outro lado, na categoria de engenharia, o esperado é não ter impacto). O intuito é averiguar, com as empresas que fazem parte deste estudo, se as áreas de processos elencados na Tabela 1 de fato constituem o maior esforço de adaptação.

Tabela 1 – Hipótese básica

Categorias	Áreas de Processo Impactadas
Gerenciamento de Projeto	Gestão de Contrato de Fornecedor Gestão de Risco e Gestão Quantitativa de Projeto
Gerenciamento de Processo	Todas
Engenharia de Sistemas	Nenhuma
Suporte	Garantia da Qualidade do Produto e Processo Análise e Métricas Resolução e Análise de Decisão Resolução e Análise Causal

Fonte: Produzida pela autora

As hipóteses complementares à inicialmente formulada são:

- As empresas presentes neste estudo foram motivadas a adotar a abordagem ágil por acreditarem que haveria melhoria de qualidade no produto.
- A percepção das empresas que integraram o CMMI e o desenvolvimento ágil é que o processo melhorou.
- Houve evolução nos projetos tidos como de sucesso, respeitando os critérios individuais das empresas quanto a qualidade, prazo ou custo.

1.3. OBJETIVOS

O objetivo primário da pesquisa é estudar como as empresas que possuem o CMMI e estão adotando o desenvolvimento ágil resolveram os pontos de conflito da abordagem tradicional e ágil, de modo a se tornarem mais eficazes. Os objetivos secundários são:

1. Fornecer uma percepção do esforço necessário de adaptação por nível de maturidade, tendo em vista que as áreas de processo do CMMI são divididas por nível de maturidade;
2. Averiguar se a integração CMMI e desenvolvimento ágil resultaram na melhoria do processo ou métricas de projetos, tidos como de sucesso, respeitando os critérios individuais das empresas quanto a qualidade, prazo e custo.

1.4. RELEVÂNCIA DO TEMA E JUSTIFICATIVAS

Segundo Baskerville et al. (2001), independente do nicho de mercado, as empresas têm sentido pressões para entregar mais rápido uma versão de *software*. O ciclo de desenvolvimento da década passada (com duração típica de 24 a 36 meses) foi reduzido (12 a 18 meses) para empresas que não têm internet como foco. Para empresas envolvidas em comércio eletrônico e criação e manutenção de portais *web*, o ciclo tem sido mais curto (3 a 6 meses de duração). Apesar de ter passado uma década desde essa observação, é possível notar, no trabalho diário das empresas, que a pressão por diminuir o prazo de entrega dos *softwares* ainda é uma realidade. Para atingir esses ciclos curtos, as empresas têm adotado técnicas do *lean production* com uso de ferramentas, métodos e gestão de projetos para eliminar ineficiência, desperdício, retrabalho em todas as fases de desenvolvimento. Mas, na luta para inovar e reduzir tempo de desenvolvimento, muitas empresas sentem-se pressionadas a adotar a forma tradicional de manter a qualidade de *software*. Esses autores averiguaram três importantes tendências no processo de desenvolvimento de *software*:

- Dirigido ao tempo: decisões de desenvolvimento são baseados na pressão de mercado;
- Qualidade dependente: indagação de como a prática e a qualidade podem conviver, ou seja, quão importante é a qualidade para o cliente e quão boa é para os desenvolvedores;
- Ajustes no processo: as empresas estão aprimorando seus processos para obter maior qualidade.

Para empresas que são dirigidas ao tempo, para serem competitivos, os produtos devem ir rapidamente ao mercado. Isto pode explicar por que pressões por tempo podem colocar atributos como desempenho, custo e qualidade em nível de prioridade menor. Essa ideia de ser a primeira a se lançar no mercado aplica-se não somente a novos sistemas, como também a novas características de produtos ou conceitos em sistemas já existentes.

Segundo Tonini, Carvalho e Spinola (2008), a maturidade é uma meta que só é atingida de forma gradual e persistente e pode significar que a organização está perfeitamente consolidada para gerenciar seus projetos, isto é, fazendo-os bem e de forma sistemática. O CMMI tem incorporado diversas melhorias tornando-se cada vez mais completo e consistente, sendo notório o aumento de empresas que têm procurado certificação em maturidade, o que fatalmente se transformará em condição essencial para participação no mercado global de *software*. A preocupação com certificação segundo modelos de maturidade está, via de regra, relacionada mais com ganhos de competividade que a empresa possa ter do que com questões de melhoria interna, mesmo que isso resulte em custo de desenvolvimento menor.

Os autores concluem que é possível admitir que buscar um modelo de excelência tendo como ponto de partida o desenvolvedor parece ser a maneira mais correta, plausível e eficaz. A formulação de um caminho singular para implementação de melhorias nos processos de desenvolvimento tem grandes chances de trazer benefícios mais duradouros para a organização, pois parte-se dos próprios elementos estruturais, organizacionais e culturais que precisam ser adequadamente abordados para que a empresa possa atingir uma efetiva maturidade organizacional.

1.5. ESTRUTURA DO TRABALHO

A dissertação é organizada em 7 capítulos:

- Capítulo 1: Definição da problemática abordada e dos objetivos;
- Capítulo 2: Contexto do estudo;
- Capítulo 3: Resultados da análise bibliométrica realizada para caracterizar as publicações acadêmicas sobre o tema da pesquisa;
- Capítulo 4: Pesquisa bibliográfica a respeito do CMMI, Desenvolvimento Ágil e *Lean* e proposição de um quadro de integração CMMI *versus* Desenvolvimento Ágil;
- Capítulo 5: Metodologia incluindo instrumento da coleta de dados e procedimentos de análise;
- Capítulo 6: Apresentação dos estudos de casos selecionados com intuito de analisar a aderência do quadro proposto de integração CMMI *versus* Desenvolvimento Ágil;
- Capítulo 7: Conclusões e limitações do estudo; indicação de perspectivas futuras.

2. CONTEXTO DO ESTUDO

Este capítulo dedica-se a contextualizar os pilares de sustentação da dissertação e, para tanto, posiciona o CMMI no contexto de melhoria contínua (Seção 2.1) para então apresentar seu surgimento (Seção 2.2), apresentar os princípios e conceitos do *Lean Development* (Seção 2.3), considerado grande influenciador do Desenvolvimento Ágil (Seção 2.4).

2.1. MELHORIA CONTÍNUA

Segundo Ahern, Clouse e Turner (2008), a excelência de processo requer identificação tanto de práticas de sucesso quanto de insucesso para usar em lições aprendidas. A melhor forma de derivar o valor destas e das práticas de sucesso não é simplesmente listando-as, mas integrando-as a processos e procedimentos padrões pelo quais o negócio é rotineiramente conduzido, sendo esta a atividade central de uma cultura de melhoria contínua. Para entender os efeitos da melhoria, o desempenho do processo deve ser medido e monitorado. A excelência do processo depende do gerenciamento de conhecimento, apresentado em três grandes pilares para melhoria contínua, conforme Figura 1.

Figura 1 – Melhoria contínua

Fonte: Modelo de Ahern, Clouse e Turner (2008)

Os três pilares são:

- CMMI: é um *framework* para coordenar esforços na melhoria do processo pela empresa (organização) podendo medir e monitorar o *status* deste esforço. Provê uma base analítica de comparação da "capacidade" do processo, de forma individual, e a "maturidade" da organização no esforço relacionado ao processo;
- *Lean*: é o resultado de "*Lean Advancement Initiative*" (LAI). Em 1990 iniciou-se o questionamento dos avanços da produção *lean* no Japão e se interessaram pela possibilidade de poder aplicar estes conceitos nos EUA. Então, em 1993 iniciou-se o LAI no MIT (*Massachusetts Institute of Technology*) com o objetivo de reduzir os custos da produção de avião e outros produtos aeroespaciais;
- Six Sigma: possui quatro objetivos principais (manter o controle do processo; melhoria constante; exceder as expectativas do cliente e adicionar tangibilidade para linha final da produção). Essa metodologia traz várias ferramentas como: análise de efeito e modos de falhas (FMEA); análise de regressão; simulação de processo e gráfico de controle.

Ahern, Clouse e Turner (2008) colocam a possibilidade de estabelecer um relacionamento entre CMMI, *Lean Engineering* e Six Sigma como sendo processos interconectados visando maturidade, eficiência e controle. Em termos gerais, modelos de maturidade do CMMI focam na maturidade e capacidade do processo; *Lean Engineering* foca na eficiência do processo e na remoção do desperdício e Six Sigma endereça o controle e a redução da variabilidade do processo.

2.2. CMMI

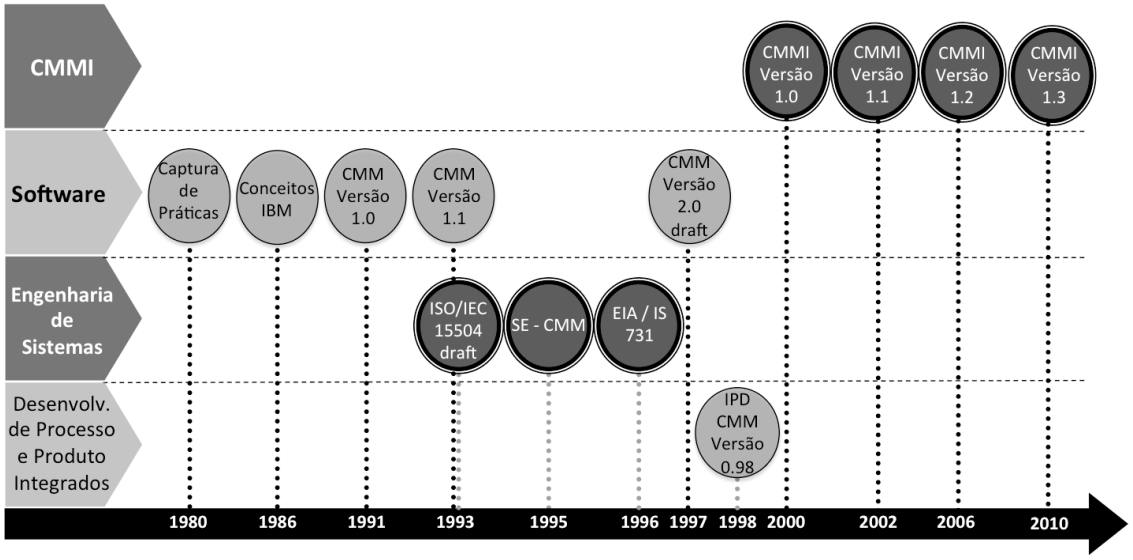
Ahern, Clouse e Turner (2008) apontam que os objetivos do CMMI são: eliminar inconsistências; reduzir duplicações; aumentar o entendimento; prover terminologia comum; prover estilo consistente; estabelecer regras de construção uniformes; manter componentes comuns e assegurar consistência com ISO/IEC 15504. Na Tabela 2 estão expostas as fontes que originaram o CMMI e na Figura 2 foi colocado o levantamento histórico extraído do CMMI-DEV v1.3 (2010).

Tabela 2 – Fontes do CMMI

Disciplina do Modelo	Modelo Fonte
Software	SW-CMM, versão draft 2.0
Engenharia de Sistemas	EIA/IS 731
Desenvolvimento de processo e produto integrados	IPD-CMM, versão 0.98

Fonte: Modelo extraído de Ahern, Clouse e Turner (2008)

Figura 2 – História CMMI



Fonte: CMMI-DEV, V1.3

Os eventos históricos estão listados em:

- *Software*

1980: Ron Radice e Richard Philips, no grupo *Watts Humphrey* da IBM, pensaram em capturar as práticas de sucesso do desenvolvimento de *software* de forma a organizá-las, tratá-las e melhorá-las. Observaram que a maior parte das práticas relatadas referiam-se à disciplina de gerenciamento de processo.

1986: *Watts Humphrey*, SEI (Instituto de Engenharia de *Software,* da Universidade de *Carnegie Mellon*) e a corporação Mitre responderam a uma requisição do governo federal dos EUA para criar uma forma de avaliar a capacidade de *software* de seus fornecedores. Esse grupo usou os conceitos da IBM para criar o *framework* de maturidade de *software*.

1991: SEI publicou "*The CMM for Software*", versão 1.0, um modelo que descreve os princípios e práticas sobre a maturidade do processo de *software*.

1993: SEI lançou nova versão do CMM para *software*, versão 1.1.

1997: Planejado o lançamento da versão 2.0 do CMM para *software*, porém não ocorreu, mas seu conteúdo foi utilizado no esforço de integração do CMM.

- Engenharia de Sistemas

A origem decorre de dois modelos de capacidade.

1993: Lançamento do draft ISO/IEC 15504 como fruto de um trabalho do projeto nomeado como SPICE (*Software Process Improvement and Capability Determination*).

1995: EPIC (*Enterprise Process Improvement Collaboration*), um grande grupo de indústria, academia e organizações governamentais, lançou o modelo de maturidade de capacidade de engenharia de sistemas (SE-CMM), o qual tinha como base o ISO/IEC 15504 que endereçava engenharia, projeto, processo e práticas organizacionais.

Paralelamente ao SE-CMM, INCOSE (*International Council on Systems Engineering*) criou um *checklist* para avaliar a capacidade de organizações de engenharia de sistemas, baseado em padrões de engenharia. Esse modelo ficou conhecido como SECAM (*System Engineering Capability Assessment*), que estendia os conceitos de SPICE e aplicava-os mais na prática de engenharia de *software* do que SE-CMM, usando o EIA 632 tendo o “modelo para os processos de engenharia” como referência principal.

1996: EPIC e INCOSE decidiram trabalhar juntos fazendo uma junção dos modelos de forma a enquadrar no padrão EIA e o resultado foi EIA/IS 731 “Modelo de Capacidade em Engenharia de Sistemas” (SECM).

- Desenvolvimento de processos e produtos integrados

1998: É originado de um draft conhecido como IPD CMM versão 0.98, ocorrendo quase na mesma época que o projeto CMMI começou seus trabalhos.

A equipe do CMMI queria incluir os conceitos de IPPD (*Integrated Product and Process Development*) no suíte de produtos do CMMI. Mas a definição de IPPD usada no documento de requerimentos do CMMI foi derivada da experiência do departamento de defesa americano com integrar operações de programas de aquisição do sistema do governo e a aquisição não era uma disciplina inicial para

CMMI. IPPD enfatizou o envolvimento de *stakeholders* em todas as funcionalidades de negócio e técnicas no decorrer do ciclo de vida do desenvolvimento do produto.

- CMMI

O desenvolvimento do projeto CMMI foi organizado em três grupos: grupo diretor, time de desenvolvimento de produto e grupo de *stakeholders*. Esses grupos eram representantes da indústria, governo e SEI. O grupo diretor produziu uma lista de requisitos para CMMI, a qual foi revista pelo grupo de *stakeholders* e subsequentemente utilizada pelo time de desenvolvimento do produto para guiar a criação dos produtos CMMI

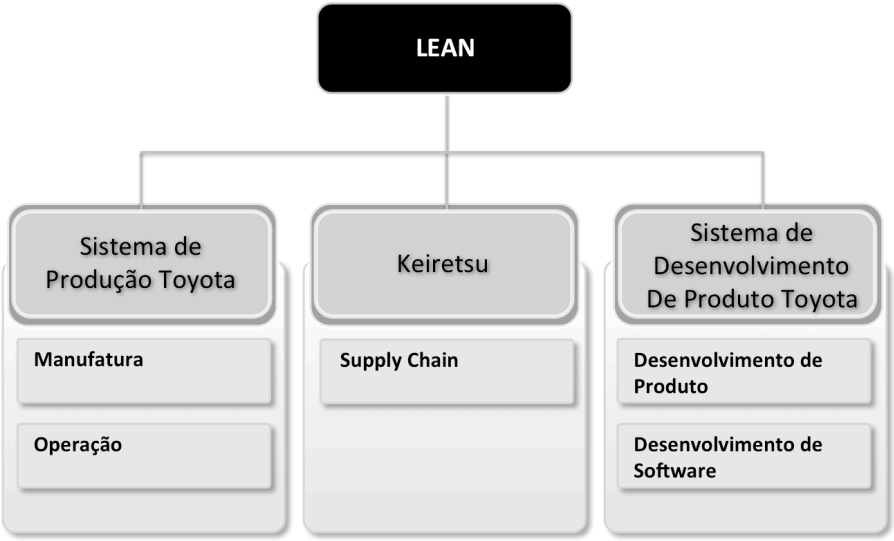
  - 2000: Primeira versão do CMMI, versão 1.0.
  - 2002: Lançamento da versão 1.1 do CMMI.
  - 2006: Lançamento da versão 1.2 do CMMI.
  - 2010: Lançamento da versão 1.3 do CMMI.

2.3. LEAN DEVELOPMENT

Em 1950, Taiichi Ohno, diretor da Toyota, seguindo as ideias de Kiichino Toyoda e recebendo apoio do consultor de qualidade, Shigeo Shino, implantou na fábrica o sistema de produção Toyota, com o objetivo de otimizar os processos e procedimentos pela redução contínua de desperdício, promovendo a melhoria contínua do processo produtivo, implantando a produção sem estoque e zero inspeção. Este foi o princípio do *Just in Time* e *Jidoka*.

Em 1990 o lançamento do livro *The machine that changed the world* tornou o termo *Lean Production* conhecido e passou a ser utilizado em *supply chain*, desenvolvimento de produto e desenvolvimento de *software*. Conforme mapeado por Mary e Tom Poppendieck (2003), é possível visualizar na Figura 3 quais são as bases fundamentais desses processos.

Figura 3 – Hierarquia dos princípios *Lean*



Fonte: Modelo de Mary e Tom Poppendieck (2003)

O desenvolvimento de *software Lean* originou-se da linha de pensamento do sistema de desenvolvimento de produto Toyota, o qual é caracterizado por quatro pilares:

- *Design* do sistema deve ser inovador, empreendedor;
- Uso de mão de obra de engenheiros experientes com intuito de orientar os iniciantes;
- Controle e planejamento orientado à responsabilidade;
- Configuração baseada em engenharia concorrente, explorar múltiplos espaços e converter em solução ótima, incentivando a criatividade por explorar múltiplos caminhos ao invés de rapidamente descartar ideias.

Mary e Tom Poppendieck (2003) expõem que o desenvolvimento é um processo de transformação das ideias em produtos e há duas escolas para definir o que é transformação: uma determinística, outra empírica. A determinística começa por criar uma definição completa do produto e cria a realização desta; a empírica começa por conceito de alto nível do produto e então estabelece ciclos de *feedback* para capturar definições que ajustem as atividades a criar uma interpretação ótima do conceito. Esses autores colocam que o desenvolvimento de produto Toyota se enquadra na escola empírica e acreditam que o processo de desenvolvimento que trata um ambiente em mudanças deveria ser um processo empírico porque provê a melhor abordagem para adaptação as mudanças. A Tabela 3 apresenta uma comparação entre os modelos.

Tabela 3 – Comparação Manufatura *Lean* x Desenvolvimento *Lean*

Manufatura *Lean*	**Desenvolvimento *Lean***
Frequentes mudanças de configuração	Frequentes mudanças de produto (versões de *software*)
Tempo de produção curto	Tempo de desenvolvimento curto
Redução da formalidade de trabalho efetuado nas etapas do processo	Redução de documentação entre as etapas de desenvolvimento
Transferência frequente de pequenas partes do trabalho entre as etapas da manufatura	Transferência frequente de informações preliminares entre as etapas de desenvolvimento
Reduzir recursos e maior fluxo de informação entre as etapas	Tempo de desenvolvimento reduzido requer menos recursos e fluxo de informação entre os estágios
Adaptabilidade a volume de mudanças, mix de produto e *design* de produto	Adaptabilidade a mudanças no *design* do produto, prazo e custo almejado
Alta atribuição de tarefas aos operários possibilita um aumento da produtividade	Ampla distribuição de tarefas a desenvolvedores aumenta produtividade
Foco na resolução rápida de problemas e na melhoria contínua de processos	Foco em inovações incrementais frequentes e continuidade do produto e melhoria do processo
Melhoria simultânea na qualidade, tempo de entrega e produtividade da manufatura	Melhoria simultânea em qualidade, tempo de desenvolvimento e produtividade do desenvolvimento

Fonte: Modelo de Mary e Tom Poppendieck (2003)

O desenvolvimento de *software lean* possui sete princípios:

1. Eliminar desperdícios: Taiichi Ohno definiu como "*overproduction*" a pior forma de desperdício na manufatura; em *software*, isso se daria pelo desenvolvimento de funcionalidades que nunca fossem utilizadas pelos usuários.

2. Construir com qualidade: construir código com qualidade desde o início do processo e não descobrir em seu final, na fase de testes. De acordo com Shigeo Shingo, há duas formas de inspeção: depois dos defeitos e para prevenir defeitos.

3. Criar conhecimento: o conhecimento é adquirido durante o processo de desenvolvimento. MacCormack definiu quatro práticas para conduzir um desenvolvimento de *software* de sucesso:

- Nas versões iniciais apresente um conjunto mínimo de características para avaliação e *feedback* do usuário;
- Construção diária e *feedback* rápido de testes integrados;
- Uma equipe ou líder com experiência e instinto para tomar boas decisões;

- Uma arquitetura modular que suporte facilmente novas características.

4. Adiar o compromisso: analise a situação e condições antes de retornar com a resposta, a ideia é fazer aquilo que se conhece.

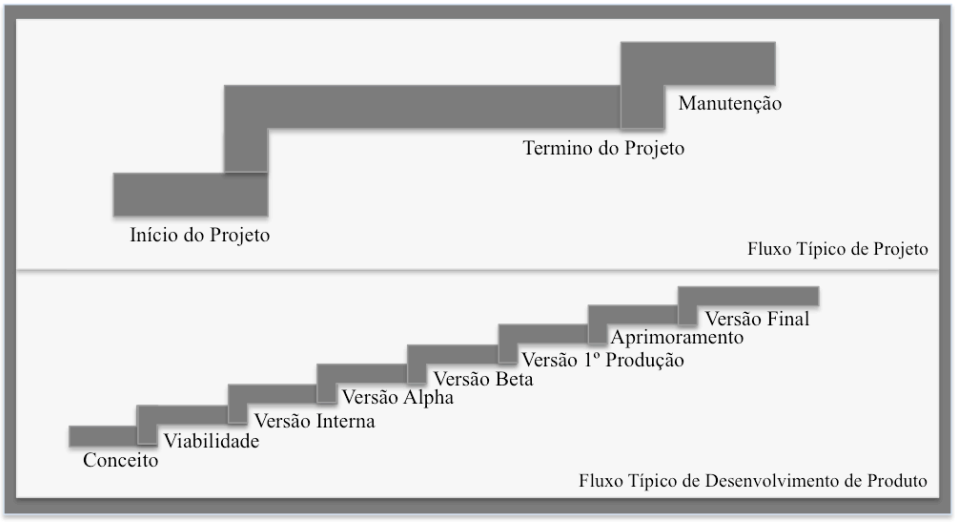
5. Entregar com rapidez: o princípio é que com entregas rápidas de *software* o cliente não tenha tempo de mudar de ideia, mas a entrega deve ser feita com qualidade.

6. Respeitar as pessoas: não existe a melhor forma de fazer, as pessoas podem colaborar para melhorar o processo, isso envolve respeito ao trabalho em equipe e democracia no trabalho.

7. Otimizar o todo: o importante é pensar no todo e otimizá-lo, de modo a construir com qualidade utilizando menos recursos e métricas de medição do processo.

Outro interessante ponto colocado por Mary e Tom Poppendieck (2003) é que o desenvolvimento de *software* estaria mais próximo de desenvolvimento de produto do que de projeto, pois o desenvolvimento é caracterizado por gerar várias versões do produto até atingir o ponto de produção.

Figura 4 – Comparação do fluxo de projeto e desenvolvimento de produto

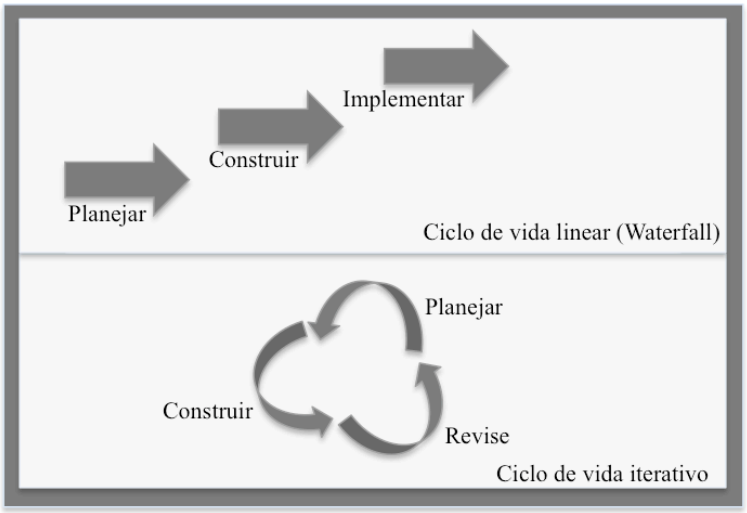


Fonte: Modelo de Mary e Tom Poppiedick (2003)

2.4. MÉTODOS ÁGEIS

Segundo Highsmith (2004), a evolução do ciclo de vida de *software* passou de linear, previsível para evolucionário, com intuito de diminuir a incerteza por dar pequenos passos e testar os resultados, conforme ilustrado na Figura 5.

Figura 5 – Comparação de ciclo de vida de *software*



Fonte: Modelo de Highsmith (2004)

A metodologia ágil surgiu da junção dos conceitos de *Lean* e desenvolvimento iterativo, o documento "Manifesto Ágil para Desenvolvimento de *Software*" foi elaborado por teóricos e profissionais da área de tecnologia da informação representantes das linhas:

- ASD (*Adaptive Software Development*): segundo Abrahamsson et al. (2002), foca principalmente em problemas presentes no desenvolvimento de sistemas complexos e grandes. O método encoraja o desenvolvimento iterativo, incremental com constante prototipação;
- *SCRUM*: segundo Swaber (2004), é um processo não prescritivo indicado para projetos que envolvem complexidade e inovação, no qual é impossível prever tudo o que acontecerá. As iterações são ciclos de desenvolvimento com duração média de 30 dias e o acompanhamento é efetuado com reuniões diárias rápidas para se discutir o progresso do trabalho, os desafios encontrados e o que esperar como resultado para a próxima reunião. No final da iteração é preciso ter alguma entrega, podendo ser uma nova funcionalidade, um documento, resultados de testes, etc.;
- *Crystal Methods*: faz parte de um conjunto de metodologias criado por Cockburn (2001), pregando que todo o projeto tem necessidades diferentes, seu funcionamento é influenciado por fatores humanos e quando os indivíduos produzem melhor há resultados positivos no projeto; comunicações e entregas frequentes reduzem a necessidade de construir produtos intermediários do processo. É um método direcionado para pequenos projetos

com equipes de até 6 desenvolvedores. Assim como o *SCRUM* tem ciclos/iterações de 1 mês;

- XP (*Extreme Programming*): segundo Decarlo (2004), o modelo é baseado em cinco fases (visão, especulação, inovação, reavaliação e disseminação) e está fundamentado em fatores críticos de sucesso considerados em cada fase do desenvolvimento do produto;
- DSDM (*Dynamic Systems Development Method*): segundo Abrahamsson et al. (2002), é a técnica de atribuir um conjunto de funcionalidades ao produto e então ajustar tempo e recurso para entregar essas funcionalidades; é um ciclo iterativo e incremental;
- FDD (*Feature-Driven Development*): segundo Abrahamsson et al. (2002), é um método que não cobre todo o processo de desenvolvimento de sistemas, mas foca nas fases de *design* e construção, realçando aspectos da qualidade (melhores práticas). É um processo iterativo com duração de 1 a 3 semanas.

Os profissionais e teóricos das linhas, acima citados, decidiram escrever o "Manifesto Ágil" contendo premissas comuns aplicadas ao desenvolvimento de *software,* cujos principais pontos são:

- Os indivíduos e suas interações têm maior importância que os processos e as ferramentas;
- A documentação é menos importante do que o executável do programa;
- A interação com o cliente tem maior valor do que o contrato assinado;
- Ser adaptável à mudança é mais importante do que seguir a rigor o que foi planejado.

Highsmith (2004) acrescenta que os principais objetivos dessa abordagem são:

- Inovação contínua: entregar produtos que atendam aos requisitos atuais dos clientes de maneira inovadora, que só seria possível em ambientes não autoritários e burocraticamente estruturados, ambientes ditos "adaptativos", nos quais a inovação e novas soluções são priorizadas;
- Adaptabilidade do produto: entregar produtos que atendam requisitos futuros dos clientes. As práticas ágeis aderem o custo da mudança como parte do desenvolvimento para atender mudanças de requisitos. Na abordagem tradicional, os requisitos são levantados no início do projeto e o produto é

entregue no final, durante esse processo, alterar uma característica do produto para atender um novo requisito pode ser impraticável. O resultado é que podem adiar demasiadamente as mudanças, gerando custos diretos mais elevados e, consequentemente, prejuízos para organização e cliente;

- Tempo de entrega reduzido: a ideia principal é a iteratividade, gerando entregas de curto prazo. Deve-se considerar quais características devem ser incluídas no projeto e criar, assim, um processo de desenvolvimento concentrado em atividades que agregam valor;
- Capacidade de adaptação do processo e das pessoas: o objetivo é que as pessoas não sejam resistentes às mudanças no decorrer do projeto, mas entendam que os produtos podem ser modificados ao longo do seu desenvolvimento. As equipes precisam ser receptivas a novas ideias;
- Resultados confiáveis: o objetivo é atender expectativas do cliente, de modo a analisar se a entrega está compatível com o resultado esperado pelo cliente.

Highsmith (2004) conclui que o desenvolvimento ágil visa agregar valor e procura gerar produtos adaptados às necessidades de modo a unir as pessoas em torno de um trabalho efetivamente colaborativo.

3. ANÁLISE BIBLIOMÉTRICA

Segundo Spinak (1996, p. 34), os primeiros registros da utilização da análise bibliométrica ocorreram no início do século XX e esta consiste na aplicação de estatísticas com intuito de estudar características e ocorrências de uso e criação de documentos. Este capítulo dedica-se a expor os resultados obtidos com a pesquisa, realizada em 2013, à procura de artigos acadêmicos no portal *Web of Science* publicados até dezembro de 2012. Foram considerados três termos presentes no tema desta dissertação: CMMI, *Lean Development* e Metodologia Ágil. Os resultados são apresentados para cada termo descrevendo três etapas da análise bibliométrica:

- Primeira Etapa: Uso de termos e critérios de refinamento para busca de artigos na base *Web of Sciente*;
- Segunda Etapa: Leitura de título, resumo e conteúdo completo (quando necessário) para classificação das publicações;
- Terceira Etapa: Análise das classificações e considerações da autora.

3.1. CMMI

A pesquisa na base *Web of Science* retornou 110 artigos, os quais foram classificados e os selecionados compõem a base teórica desta dissertação.

3.1.1. Primeira Etapa: Uso da base Web of Science

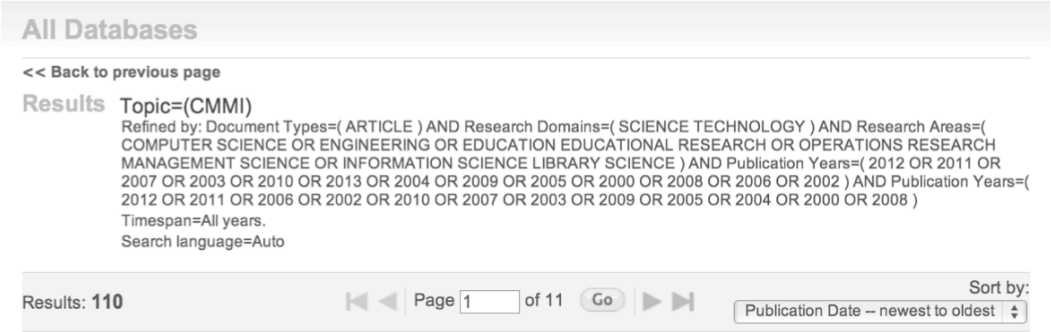
No critério de busca por tópico foi colocado o termo CMMI. A justificativa deste termo deve-se ao fato de o padrão CMMI versão 1.0 ter sido divulgado em 2000, sendo a junção dos conceitos SW-CMM, versão draft 2.0, EIA/IS 731 e IPD-CMM, versão 0.98, conforme exposto na seção *Contexto do Estudo* desta dissertação. Para os itens de refinamento da busca foram utilizadas as opções:

- *Document Types: article*;
- *Research Domains: Science Tecnology*;
- *Research Areas: Computer Science, Engineering, Operations Research Management Science, Information Science Library Science and Education Educationals Research*;

- *Publication Years*: de 2000 a 2012.

O resultado foram 110 artigos, conforme Figura 6:

Figura 6 – Primeira Etapa da busca CMMI



Fonte: Resultado da pesquisa da Web of Science

3.1.2. Segunda Etapa: Classificação das publicações

Os 110 artigos obtidos foram classificados pelo assunto seguindo a Tabela 4.

Tabela 4 – Segunda Etapa para busca CMMI

Classificação	Os artigos tratam de…
Análise dos processos	como implementar o modelo CMMI; custo da implantação e manutenção
Aplicação do CMMI	estudos de casos de empresas que aplicaram o CMMI; formas de implantação do CMMI
Aplicação em pequenas/médias empresas	estudos de casos de empresas classificadas como pequenas ou médias que aplicaram o CMMI
Compatibilidade com gestão ágil	estudos de casos de empresas que usavam o processo de desenvolvimento ágil e decidiram aplicar o CMMI; integração CMMI e Desenvolvimento ágil
Compatibilidade com ISO	estudos de casos que aplicaram o modelo CMMI associado ao padrão ISO; comparação entre CMMI e ISO
Compatibilidade com Pmbok	estudos de casos que aplicaram o modelo CMMI associado com Pmbok; integração CMMI e Pmbok
Compatibilidade com Six Sigma	estudos de casos que aplicaram o modelo CMMI associado ao Six Sigma; integração CMMI e Six Sigma
Definição/Classificação dos processos CMMI	qualidade; classificação das organizações pela maturidade; proposta de melhorias ao CMMI
Excluído	assuntos não aplicados ao processo de desenvolvimento de *software*, por isso foram retirados da análise
Técnicas de gestão de projetos	organização dos processos, qualidade, produtividade e métricas de progresso de projeto

Fonte: Produzida pela autora

O resultado obtido foi a Tabela 5.

Tabela 5 – Resultado da Segunda Etapa para busca CMMI

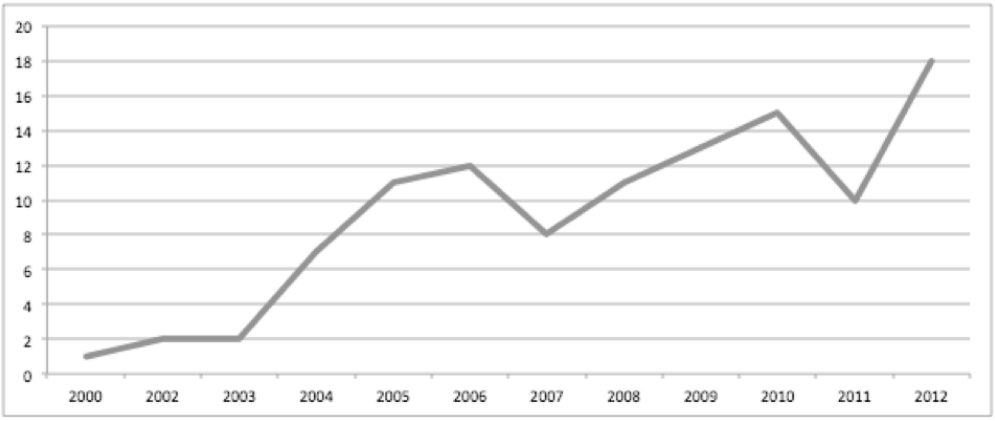
Classificação	2000	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	Total
Análise dos processos				1	3	2		3	3	3	1	3	19
Aplicação do CMMI	1		1		4		1			1	1	4	13
Aplicação em pequenas/médias empresas								1		1		2	4
Compatibilidade com gestão ágil				1			2		2	2		2	9
Compatibilidade com ISO				1		2			1	3	3	2	12
Compatibilidade com Pmbok				1	2	1				2			6
Compatibilidade com Six Sigma									1				1
Definição/Classificação dos processos CMMI				1		2	1	2	3		1		10
Excluído			1				2		1	2			6
Técnicas de gestão de projetos		2		2	2	5	2	5	2	1	4	5	30
Total	**1**	**2**	**2**	**7**	**11**	**12**	**8**	**11**	**13**	**15**	**10**	**18**	**110**

Fonte: Produzida pela autora

3.1.3. Terceira Etapa: Análise das Publicações

A distribuição por ano está apresentada na Figura 7:

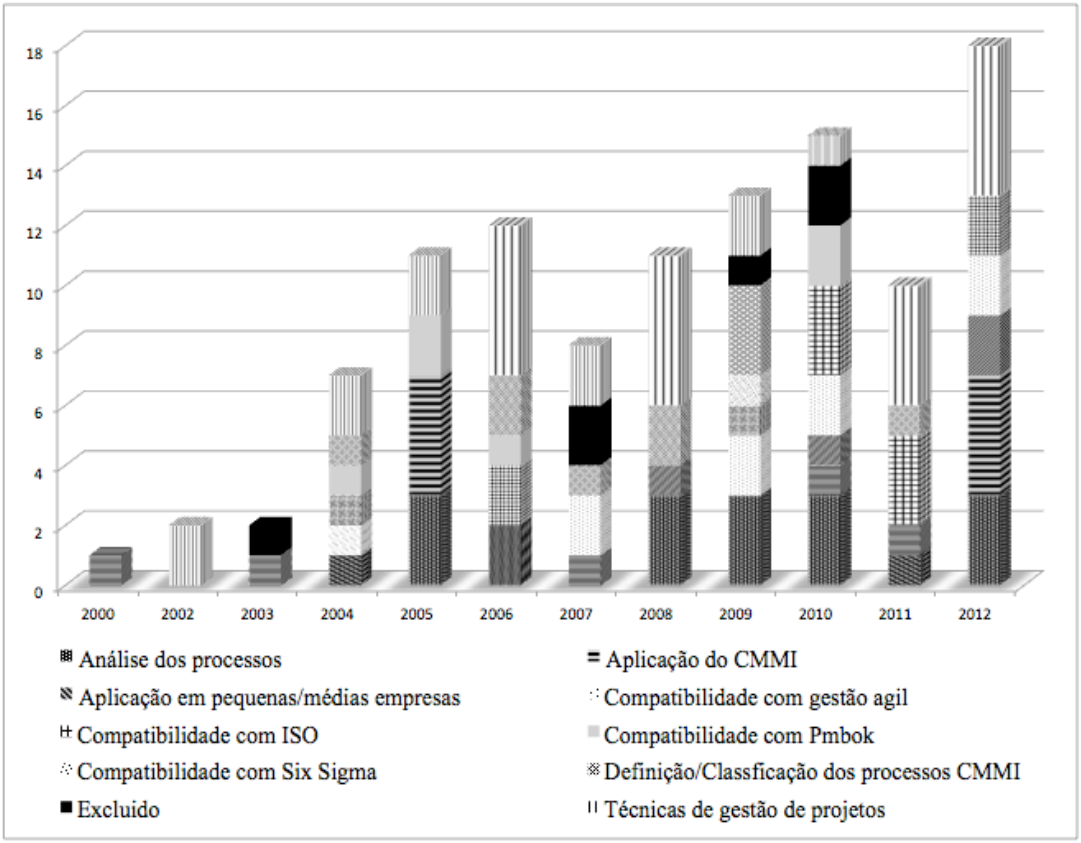
Figura 7 – Terceira Etapa: distribuição por ano da busca CMMI



Fonte: Produzida pela autora

A distribuição dos assuntos por ano está apresentada na Figura 8:

Figura 8 – Terceira Etapa: distribuição do assunto por ano da busca CMMI



Fonte: Produzida pela autora

Dado o resultado obtido com as classificações, é possível observar que, a partir de 2004, aparecem artigos propondo o estudo da integração CMMI com a abordagem ágil, o que é explicável já que, desde 2003, essa metodologia começou a ganhar repercussão com os estudos publicados por Beck. Em 2008 surgiram os primeiros trabalhos com divulgação de estudos de casos da implantação de CMMI em pequenas e médias empresas, o que permite concluir que a adoção desse modelo

seria possível para organizações que não dispunham de tantos recursos financeiros. Ainda poderia ser acrescido o fato de que as metodologias ágeis foram adotadas primeiramente por empresas deste porte o que motiva ainda mais o estudo da integração CMMI e desenvolvimento ágil.

As técnicas de gestão de projetos aparecem de forma constante entre os anos, mostrando uma preocupação quanto ao sucesso dos projetos, uma vez que estes estão submetidos ao modelo de processo do CMMI.

Um ponto que chamou a atenção foram artigos que discutiam sobre definição e classificação dos processos de CMMI, sendo que alguns artigos até sugeriam a possibilidade de adaptação e propostas de melhoria deste modelo.

Os artigos classificados em “Aplicação em pequenas/médias empresas”, “Compatibilidade com gestão ágil” e “Definição/classificação dos processos CMMI” foram considerados para etapa de pesquisa bibliográfica, totalizando 23 artigos.

3.2. LEAN DEVELOPMENT

A pesquisa na base *Web of Science* retornou 57 artigos, os quais foram classificados e os selecionados compõem a base teórica desta dissertação.

3.2.1. Primeira Etapa: Uso da base Web of Science

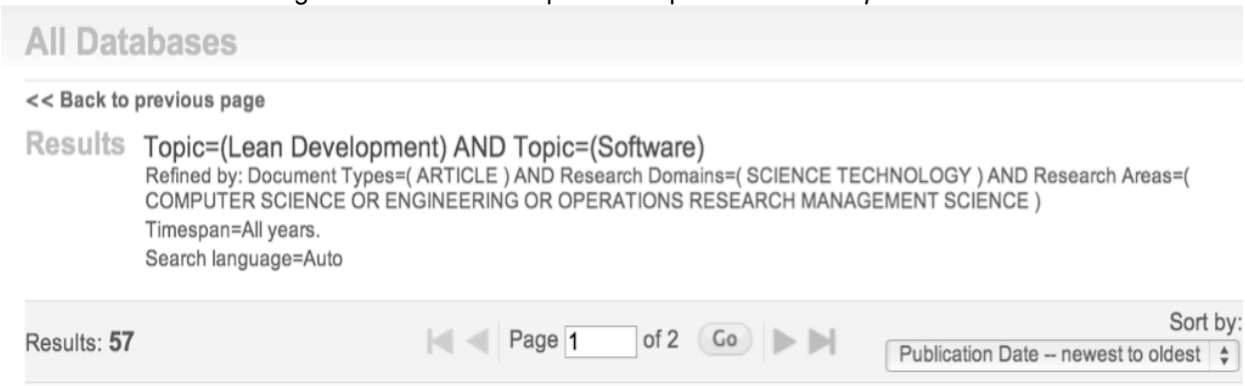
No critério de busca por tópico foi colocado o texto *Lean Development* e *Software*. Foi necessário acrescentar o termo *Software* porque apenas *Lean Development* retornava muitos artigos referindo a manufatura/indústria.

Para os itens de refinamento da busca foram utilizadas as opções:

- *Document Types: article;*
- *Research Domains: Science Tecnology;*
- *Research Areas: Computer Science, Engineering, Operations Research Management Science.*

O resultado foram 57 artigos, conforme Figura 9:

Figura 9 – Primeira Etapa: busca por *Lean Development*



Fonte: Resultado da pesquisa da Web of Science

3.2.2. Segunda Etapa: Classificação das Publicações

Os 57 artigos obtidos foram classificados pelo assunto seguindo a Tabela 6.

Tabela 6 – Segunda Etapa: classificação da busca por *Lean Development*

Classificação	Os artigos tratam de…
Desenvolvimento ágil	integração/história do Lean Development e metodologias ágeis
Excluído	assuntos não aplicados ao processo de desenvolvimento de *software*, por isso foram retirados da análise
Framework/ferramenta para manufatura/produção *lean*	manufatura *Lean*, propondo ferramentas, *framework*, sistemas de apoio
Lean development	estudos de casos de empresas que aplicaram os princípios *Lean* de desenvolvimento; discutem os conceitos Lean
Processos *Lean* - requisitos de infraestrutura	requisitos de infraestrutura necessários para suportar um processo *Lean Development*
Processos *Lean* - Uso de sistemas na aplicação	processo de desenvolvimento, propondo sistemas de apoio para difundir os princípios *Lean*
Processos *Lean* - gerenciamento de informações	informações a serem coletadas para confecção de relatórios gerenciais

Fonte: Produzida pela autora

O resultado obtido foi a Tabela 7.

Tabela 7 – Segunda Etapa: resultado da busca por *Lean Development*

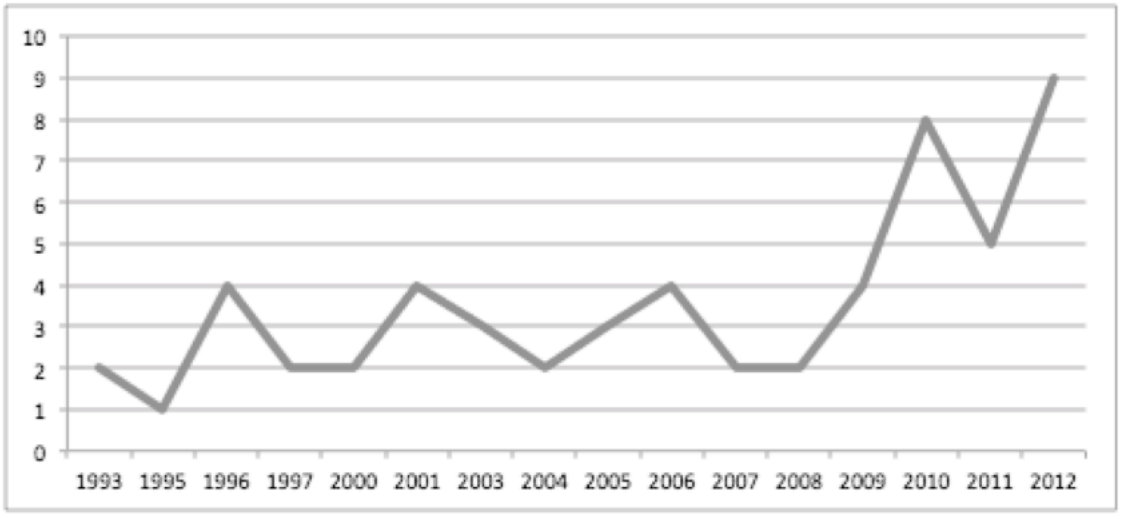
Classificação	1993	1995	1996	1997	2000	2001	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	Total
Desenvolvimento ágil										1				1		2	4
Excluído				1		1				1				1			4
Framework/ferramenta para manufatura/produção *lean*	1		2				1			1			3	2		1	11
Lean development					1	3	1	1	2		1		1	2	2	3	17
Processos *Lean* - gerenciamento de informações	1														1		2
Processos *Lean* - requisitos de infraestrutura		1													2	2	5
Processos *Lean* - Uso de sistemas na aplicação			2	1	1		1	1	1	1	1	2		2		1	14
TOTAL	**2**	**1**	**4**	**2**	**2**	**4**	**3**	**2**	**3**	**4**	**2**	**2**	**4**	**8**	**5**	**9**	**57**

Fonte: Produzida pela autora

3.2.3. Terceira Etapa: Análise das Publicações

A distribuição por ano está apresentada na Figura 10:

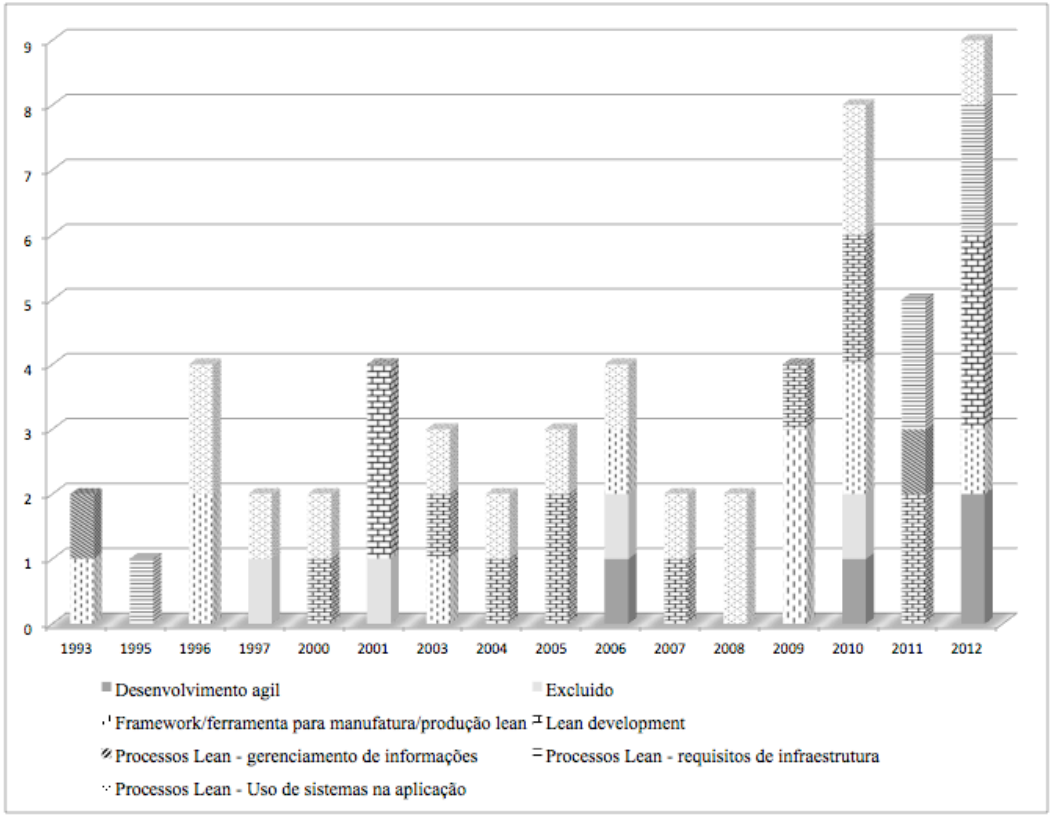
Figura 10 – Terceira Etapa: distribuição por ano da busca por *Lean Development*



Fonte: Produzida pela autora

A distribuição de assunto por anos está apresentada na Figura 11:

Figura 11 – Terceira Etapa: distribuição do assunto por ano da busca por *Lean Development*



Fonte: Produzida pela autora

Executada a busca, foi possível observar que o termo *Lean Development* é utilizado com maior frequência pela manufatura e, mesmo adicionando o termo *Software,* ainda resultou em 11 artigos que utilizam o sistema como apoio para criar ou mesmo aprimorar a agilidade da manufatura. Foram observados 4 artigos que se propunham a fazer uma análise sobre a relação com o desenvolvimento ágil referindo-se ao processo de desenvolvimento de *software*. Foi notado que 14 artigos utilizavam os sistemas de *software* como apoio à melhoria dos processos sob a óptica *Lean* (enxuto) e somente 17 artigos discutiam os conceitos ou faziam uma exposição dos casos de *Lean Development* no processo de desenvolvimento de *software*.

Os artigos classificados como “Desenvolvimento Ágil” e “*Lean Development*” foram considerados para a próxima etapa de Pesquisa Bibliográfica, totalizando 21 artigos.

3.3. MÉTODOS ÁGEIS

A pesquisa na base *Web of Science* retornou 238 artigos, os quais foram classificados e os selecionados compõem a base teórica desta dissertação.

3.3.1. Primeira Etapa: Uso da base Web of Science

No critério de busca por tópico foi colocado o texto "*Agile Software Development*" ou "*Agile Methods*".

Para os itens de refinamento da busca foram utilizadas as opções:

- *Document Types: article;*
- *Research Domains: Science Tecnology;*
- *Research Areas: Computer Science, Engineering*
- *Languages: English*
- *Publication Years: Exclude 2013*

O ano de 2013 foi excluído da busca, já que para os demais termos só foram considerados artigos publicados até 2012.

O resultado foram 238 artigos, conforme Figura 12:

Figura 12 – Primeira Etapa: resultado da busca por *Agile*

Fonte: Resultado da pesquisa da Web of Science

3.3.2. Segunda Etapa: Classificação das Publicações

Após a leitura dos 238 artigos, estes foram classificados por assunto (Tabela 8).

Tabela 8 – Segunda Etapa: classificação por assunto por busca *Agile*

Classificação	**Os artigos relacionados tratam de…**
Ensino da Metodologia Ágil	como abordar/ensinar a metodologia ágil nas universidades; como esta deve ser tratada na disciplina de engenharia de *software*
Melhoria de Processo	vantagens e contribuições da metodologia nos processos da organização
Desenvolvimento Distribuído	aplicação e proposta de *framework* para utilização da metodologia ágil no contexto de desenvolvimento distribuído
Em Grandes Empresas	estudos de casos e proposta de *framework* para abordagem ágil nas organizações classificadas como grandes
Aplicado a Sistemas *Web*	como a metodologia deve ser abordada e/ou aplicada para sistemas orientados ao *design* (*Web*)

Classificação	**Os artigos relacionados tratam de…**
Comparando com Modelo Tradicional	vantagens, diferenças, semelhanças com a metodologia tradicional
Foco na Equipe	exposição e preocupação com a satisfação da equipe no uso da metodologia; desafios da metodologia a profissionais inexperientes
Excluído	assuntos não aplicados ao processo de desenvolvimento de *software*, por isso foram retirados da análise
Fatores Críticos de Sucesso	exposição dos fatores críticos de sucesso para aplicação da metodologia
Técnicas de Gerenciamento	proposta de *framework* ou discussão sobre métricas de medição de progresso de projeto
Prática x Teoria	estudo da metodologia divulgada na teoria sobre o que está sendo praticado
Técnicas Aplicados ao Desenvolvimento	técnicas direcionadas ao desenvolvimento como detecção de erro, aplicação de TDD (*Test-Driven Development*)
Conhecimento Compartilhado	proposta de *framework* ou discussão sobre como o conhecimento pode ser compartilhado ou mesmo difundido já que a metodologia prega pouca documentação
Desenvolvimento de Novos Produtos	como a metodologia pode ser abordada para desenvolvimento de sistemas orientados ao mercado ou submetidos ao processo de linha de produção
Casos em Empresa/Área Específica	estudos de casos ou propostas de sistemas aplicados especificamente em algumas empresas e/ou áreas
Aplicado a Sistemas Críticos/*Embedded*	estudos de casos ou proposta de uso da metodologia a ser aplicada em sistemas críticos (não passíveis de falhas) ou a sistemas *embedded*
Lean Development	semelhanças com *Lean Development*
Foco no Usuário	discussão da importância do papel do usuário no processo submetido a esta metodologia
Cultura Organizacional	como a cultura organizacional pode dificultar ou promover a difusão da metodologia
CMMI	como a metodologia convive em organizações calcadas no modelo CMMI
Explicativo	exposição da metodologia: sua origem, características, princípios
Em Pequenas Empresas	estudos de casos e proposta de *framework* para abordagem ágil nas organizações classificadas como pequenas

Fonte: Produzida pela autora

O resultado obtido foi a Tabela 9.

Tabela 9 – Segunda Etapa: resultado da busca por *Agile*

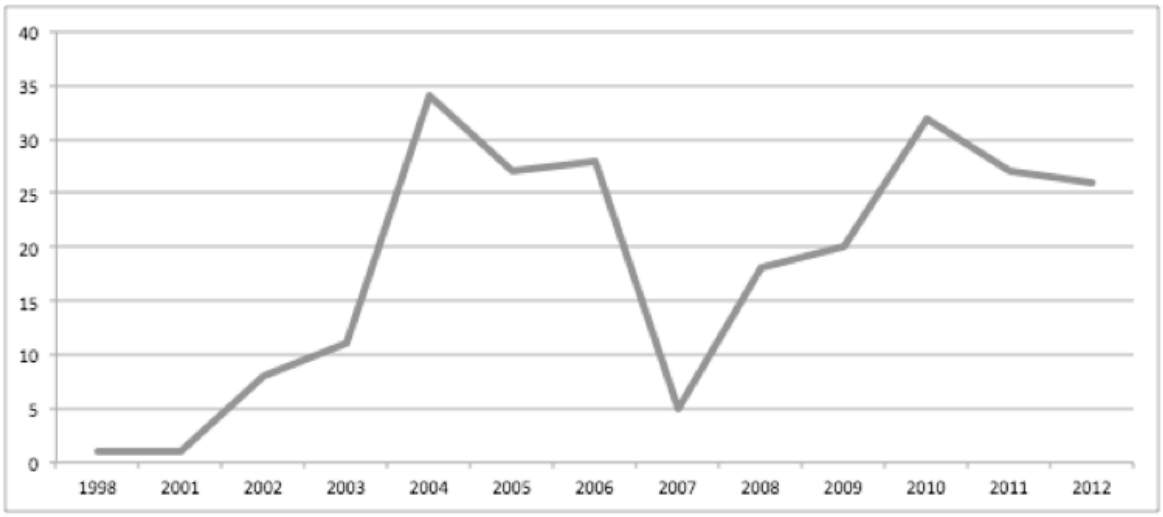
	1998	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**2012**	**Total**
Aplicado a Sistemas Críticos e *Embedded*					3	1	1		1	1		1	1	9
Aplicado a Sistemas *Web*					2		1		1	2				6
Casos em Empresa/Área Específica				1		2	2			2	2	1	1	11
CMMI			1		1									2
Comparando com Modelo Tradicional			1	2		3	4		4	1	1	2	2	20
Conhecimento Compartilhado			2		2	5	3	1	2	6	4		3	28
Cultura Organizacional			1									2	1	4
Desenvolvimento de Novos Produtos									2		2	2		6
Desenvolvimento Distribuído	1		2		2	1	1			1	7	2	1	18
Em Grandes Empresas				2	3	2	2							9
Em Pequenas Empresas							1				1			2
Ensino da Metodologia Ágil			1	2	1	2	1		1		1	1	2	12
Excluído					1	1		1			2	1		6
Explicativo				2	8	4	2	1			1	2	4	24
Fatores Críticos de Sucesso							2		1	2		1		6
Foco na Equipe				1	2	2	1				1	1	2	10
Foco no Usuário					2				1			2		5
Lean Development					1						1	2		4
Melhoria de Processo		1			3	3			1	1	3	2	1	15
Prática x Teoria					1		2				1	1		5
Técnicas Aplicados ao Desenvolvimento				1	1	1	1	1	2	2	3	1	3	16
Técnicas de Gerenciamento					1		4	1	2	2	2	3	5	20
Total	**1**	**1**	**8**	**11**	**34**	**27**	**28**	**5**	**18**	**20**	**32**	**27**	**26**	**238**

Fonte: Produzida pela autora

3.3.3. Terceira Etapa: Análise das Publicações

A distribuição por ano está apresentada na Figura 13.

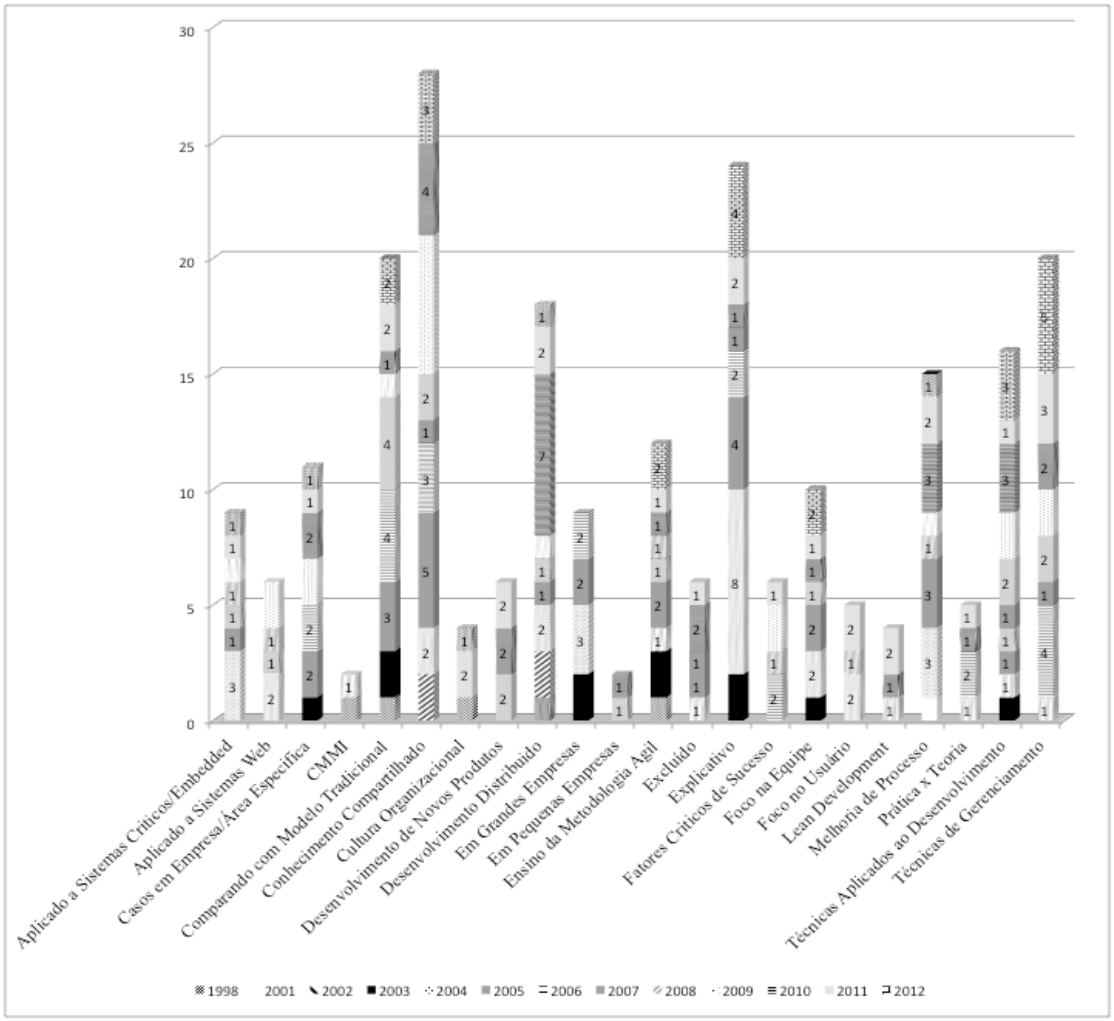
Figura 13 – Terceira Etapa: distribuição por ano da busca por *Agile*



Fonte: Produzida pela autora

A distribuição de assunto por anos está apresentada na Figura 14.

Figura 14 – Terceira Etapa: distribuição de assuntos por anos por busca *Agile*



Fonte: Produzida pela autora

De posse do resultado da pesquisa podemos observar que o ano de 2004 apresenta o maior número de publicações do período, o que pode ser justificado pelo fato de a metodologia ágil ter ganho maior destaque com a publicação dos estudos de Beck em 2003. O ano de 2004 apresentou a maior quantidade de artigos propondo explicar princípios e conceitos dos métodos ágeis. Outro ponto marcante foi identificar que 28 artigos mostram-se preocupados com a questão do conhecimento, ou seja, como difundir ou mesmo compartilhar informações quando o princípio ágil é produzir menos documentação.

Em 2010 houve uma grande incidência de artigos preocupados em estudar a aplicação dos princípios ágeis no ambiente de desenvolvimento distribuído, sendo 7 dos 18 artigos apresentados no período de 1998 a 2012.

Um ponto importante a observar é a quantidade de artigos preocupados com a questão do gerenciamento, ou seja, quais métricas ou mesmo como apurar o progresso de um projeto na metodologia ágil quando esta prega menos burocratização; outros fazem uma comparação com o modelo tradicional, propondo formas de integração. Desta forma, observamos alguns trabalhos interessados no estudo da integração do modelo tradicional e ágil, alguns expondo estudos de casos em que a metodologia ágil foi aplicada em grandes empresas e outros colocam como os princípios ágeis podem contribuir para a melhoria do processo de desenvolvimento de *software*.

Um ponto inesperado da pesquisa foi encontrar apenas 2 artigos (TURNER, 2002; KAHKONEN; ABRAHAMSSON, 2004) nos quais o foco era exatamente a integração do desenvolvimento ágil *versus* CMMI e outros 4 artigos (GREER; HAMON, 2011; PETERSEN, 2011; WEBER; WILD, 2004; WANG; GOBBO, 2010) que faziam a exposição dos processos de maturidade *versus* conceitos do *Lean Development.*

Os artigos que tratam dos conceitos *Lean versus* processos de maturidade são de Greer e Hamon (2011) e Petersen (2011) que colocam que não há um consenso a respeito da definição do desenvolvimento de *software* ágil, mas que certamente é a resposta para um desenvolvimento rápido de *software*, fortemente influenciado pelos princípios *Lean* de manufatura. Os autores acrescentam que os princípios de promoção da colaboração e autogestão podem ser absorvidos pelos processos de maturidade. Weber e Wild (2004) e Wang e Gobbo (2010) explanam como o

desenvolvimento *Lean* e valores ágeis podem ser aplicados para controlar a execução de processos do negócio e para melhorar sua eficiência e produtividade.

Os outros dois artigos que tratam do CMMI *versus* desenvolvimento ágil são de Turner (2002) que discute a relação entre eles, colocando que, se para alguns pode ser caracterizado como a relação entre óleo e água, para ele é mais uma relação de cooperação e sinergia, e sugere que a leitura do CMMI seja interpretada na sua essência e não de forma literal. Kahkonen e Abrahamsson (2004) acrescentam que o uso das práticas ágeis permite alcançar o nível 2 de maturidade. Os artigos classificados como “Em Grandes Empresas”, “Comparando com o modelo tradicional”, “*Lean Development*” e “CMMI” foram considerados para próxima etapa de Pesquisa Bibliográfica, totalizando 35 artigos.

4. PESQUISA BIBLIOGRÁFICA

No Capítulo 3 consta a análise bibliométrica realizada para esta dissertação e alguns artigos foram selecionados para compor a base teórica deste trabalho, compondo, assim, este capítulo.

4.1. CMMI

Segundo Garcia, Pacheco e Calvo-Manzano (2010) desde o início dos anos 80 a indústria de *software* tem tentado aumentar a qualidade e produtividade por aplicar novos métodos e técnicas e se reconhece que infelizmente o problema fundamental para muitas empresas tem sido a incapacidade de gerenciar o processo de *software*. Muitos métodos, técnicas e ferramentas têm sido desenvolvidos, contudo os produtos de *software* ainda sofrem com custos excessivos, atrasos na entrega e baixa qualidade. Ebert e De Man (2008) acrescentam que o desenvolvimento de sistemas e *software* requer profundos conhecimentos de produtos e tecnologia combinado com efetivo uso do trabalho de equipe, processos, método e ferramentas. Esses autores argumentam que empresas de *software* e TI são desafiadas por pressão de custo, diminuição do ciclo do tempo e evolução rápida de tecnologia, características e funcionalidades. Garcia e Pacheco (2010) acrescentam que o segmento de TI assume uma posição de destaque porque atualmente o *software* no contexto do entorno industrial representa a chave de produtividade e eficiência para as atividades de desenvolvimento e estratégia frente a competência.

Segundo Ebert e De Man (2008), a eficiência do ciclo de vida do produto é alcançada pela eficiência gerencial, excelência de processo e eficiência tecnológica. A eficiência gerencial almeja práticas de gerenciamento *lean* e assegura que as pessoas tenham concordância com o compromisso real e de entrega. Melhoria de processo endereça pesquisa e desenvolvimento e processo de gerenciamento de produto, almejando a melhora da capacidade do processo.

Garcia e Pacheco (2010) complementam que, para satisfazer os múltiplos requisitos do processo de *software*, tanto as grandes como as pequenas empresas têm

realizado um esforço para melhoria do processo de *software*. Um padrão conhecido internacionalmente é o CMMI, o problema é o seu alto custo de implantação.

Huang e Han (2006) expõem que o processo de desenvolvimento de *software* bem definido pode melhorar a qualidade do *software* e reduzir os riscos de projetos de desenvolvimento de *software*. Em 2001, o SEI (Instituto de Engenharia de *Software*) lançou o modelo CMMI com o objetivo de integrar os vários modelos de maturidade existentes e as principais vantagens foram: eliminação de inconsistências e duplicações e simplificação do processo de melhoria em toda a empresa; redução do custo e tempo associado com o processo de melhoria baseado em modelos; aumento do retorno de investimento da organização com os esforços de melhorias do processo de *software*.

Segundo o SEI, através do CMMI-DEV v1.3 (2010), a estrutura básica do CMMI é um *framework* com 22 áreas de processo distintas. Cada área de processo é relacionada a um dos 4 níveis de maturidade, sendo do nível 2 ao 5, já que o nível 1 não contém nenhuma área (vide Figura 15). Uma área de processo é definida como "um *cluster* de práticas relacionadas na área quando desempenhadas coletivamente, satisfazem a um conjunto de objetivos consolidados e importantes para fazer uma melhoria significativa na área". Os níveis de maturidade são chamados: iniciante, gerenciado, definido, quantitativamente gerenciado e otimizado.

- Nível 1: Iniciante, o foco é na competência das pessoas. Significa que o sucesso do projeto depende dos esforços de talentos individuais;
- Nível 2: Gerenciado, o gerenciamento do projeto é o mais importante do conjunto de áreas de processos que precisam ser estabelecidos;
- Nível 3: Definido, os processos são padronizados. Áreas de processo de engenharia são implementadas para assegurar saída de alta qualidade que convergem com as necessidades dos usuários. Processos são compartilhados no nível da organização e são pró-ativos, mas melhorias substanciais no processo ainda podem ser implementadas.
- Nível 4: Quantitativamente gerenciado, medida quantitativa dos processos estão disponíveis e processos são pró-ativamente controlados.
- Nível 5: Otimizado, melhorias substanciais do processo podem ser executadas baseadas no profundo conhecimento do comportamento do processo.

Maturidade do processo pode ser vista como “os degraus” onde o processo é explicitamente gerenciado, definido, gerencialmente quantificado e otimizado. As áreas de processo que compõem os níveis de maturidade são mostradas na Figura 15.

Figura 15 – *Framework* CMMI

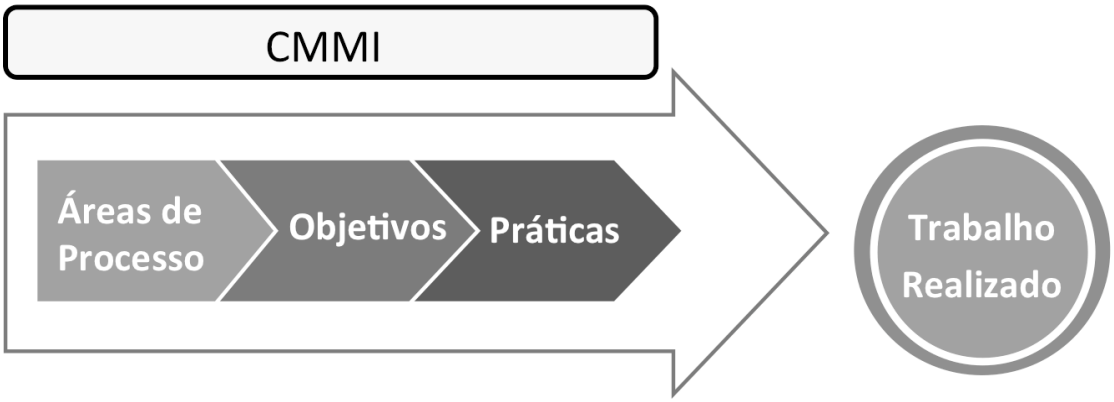
Fonte: Modelo adaptado de Veldman e Klingenberg (2009)

Cada área de processo possui um objetivo genérico, subdividido em objetivos específicos (vide Apêndice 1). Objetivos genéricos são anexados a cada nível de maturidade para guiar o progresso de um nível para outro. A ideia básica do nível de maturidade é que, quando processos tornam-se padronizados, podem ser controlados porque a variação é conhecida.

Hsueh (2008) argumenta que CMMI pode ser usado para melhoria do processo por meio de projetos, de uma divisão ou da organização. A abordagem é permitir que as empresas possam mapear seus processos seguindo os passos de definição de modelo de processo, modelo de verificação do processo, modelo de transformação, modelo de validação, processo de educação e execução do projeto piloto. Todas as áreas de processo CMMI podem ser contínuas ou faseadas.

Um estrutura básica do CMMI é mostrada na Figura 16.

Figura 16 – Estrutura básica do CMMI



Fonte: Modelo inspirado em Hsueh (2008)

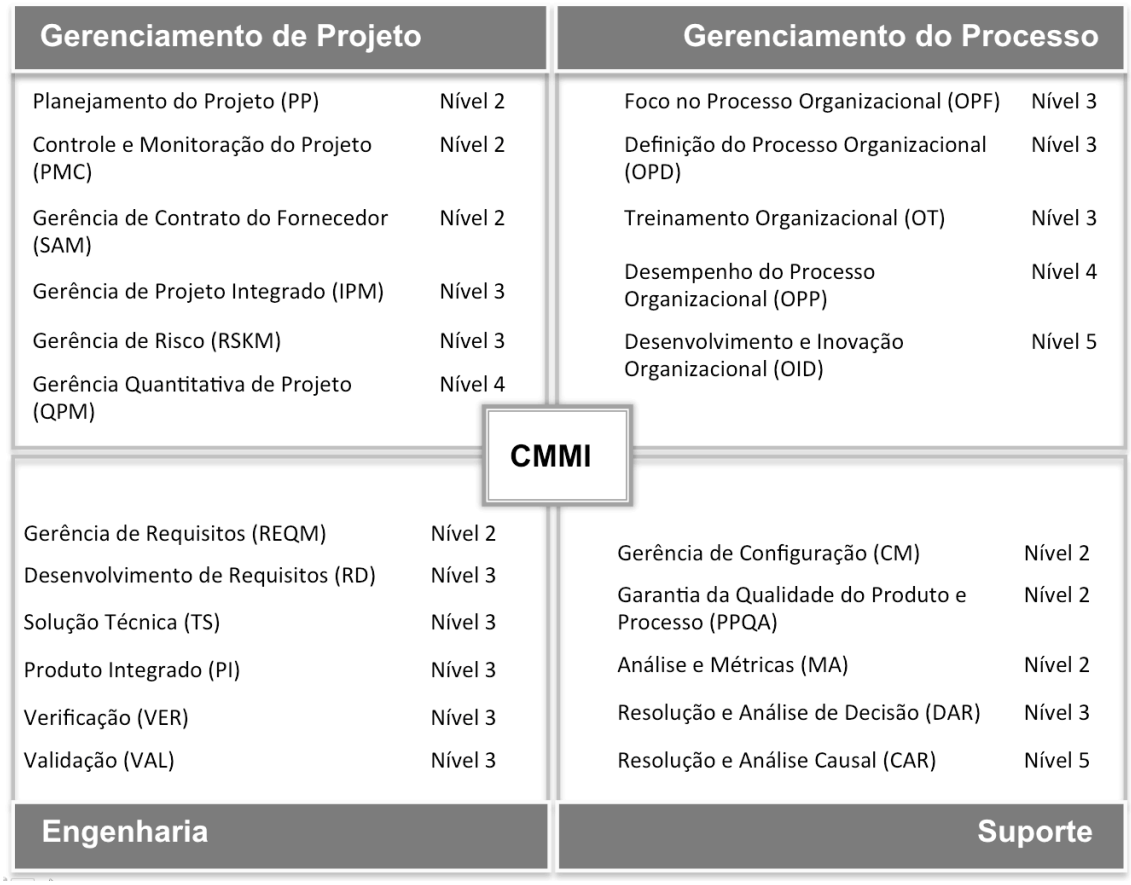
Uma área de processo possui muitos objetivos e cada objetivo é atingido pela execução de muitas práticas. O trabalho realizado é uma prática definida pela organização, que pode realizar uma prática CMMI e então satisfazer os objetivos CMMI.

Huang e Han (2006) explicam que o CMMI provê duas opções de escolha: faseado ou representação contínua. A representação faseada provê um *framework* para organizar os passos evolucionários e cinco níveis são usados para medir a maturidade do processo de *software* da empresa.

SEI também adicionou a representação contínua para o CMMI com intuito de prover flexibilidade para capacitar as organizações de *software* a escolher o caminho para melhoria, que permitiu a comparação com o padrão ISO/IEC 15504. Quando se escolhe pela representação contínua o segredo é selecionar quais áreas de processo se enquadram ao contexto da empresa, já que o CMMI não provê um guia que representa a prioridade entre as áreas de processos.

Enquanto a representação faseada atribui a cada área de processo um nível de maturidade que varia de 2 a 5, a representação contínua utiliza as mesmas áreas de processo, porém sem atribuir nenhum nível de maturidade. Todas as áreas de processo são organizadas em quatro categorias: gerenciamento do processo, gerenciamento de projetos, engenharia e suporte, conforme ilustrado na Figura 17.

Figura 17 – Representação contínua do CMMI



Fonte: Modelo de Huang e Hang (2006)

A representação contínua dá à organização de *software* a flexibilidade para selecionar as áreas de processo que deseja melhorar, permitindo que possam ser ordenadas da melhor forma, de modo a convergir com os objetivos do negócio e redução de riscos. Muitas empresas têm escolhido esse modelo pela questão da flexibilidade, porém sentem dificuldades em estabelecer qual a melhor ordenação das áreas de processo para certo contexto.

Considerando as áreas de processo do CMMI, Von Wangenheim, Rossa Hauck e Von Wangenheim (2009) argumentam que o monitoramento e controle de projeto é fundamental para entendimento do progresso do projeto e gerentes de projetos geralmente capturam medidas atualizadas de parâmetros de projeto (como prazo, progresso, custo ou risco) e comparam com as estimativas do projeto feitas inicialmente. Eles observaram que o estabelecimento desses processos ajudam as empresas a melhorar o desempenho do projeto em termos de prazo, custo e esforço e também permite que os gerentes de projetos possam observar um desvio do plano de projeto mais rapidamente. Um resultado interessante foi que a visibilidade do

status do projeto em muitos casos levou a empresa a abortar um projeto de insucesso que não teria sido identificado anteriormente.

Dos Santos, De Oliveira e Da Silva (2009) aplicaram um *survey* no qual puderam observar que os clientes têm uma percepção significativamente inferior a suas expectativas em relação à qualidade do serviço de fornecedores de *software* que possuem CMMI. A pior qualidade de serviço percebida é relativa ao controle e monitoramento de projeto, processos relativos ao nível 2 do CMMI, e Solução técnica, integração do produto e gerenciamento de projeto integrado, processos presentes no nível 3. Outro ponto que pode ser observado é que clientes têm uma expectativa mais alta por serviços prestados por fornecedores que possuem CMMI principalmente quanto ao gerenciamento de requisitos. Os resultados sugerem que os clientes esperam por melhores serviços em relação às principais atividades de engenharia de *software* (como requisitos, *design*, implementação e seu gerenciamento) quando contratam um fornecedor de *software* com certificação CMMI.

Ter a certificação CMMI, acrescenta Liou (2011), impulsiona as empresas a buscar melhorias no gerenciamento do processo, necessárias para assegurar que o processo sirva ao seu propósito. Quando o processo está claramente definido, bem mensurado e sob controle, ele reflete o nível de maturidade da empresa. Esse autor ainda complementa que fatores humanos podem impactar no processo de maturidade, devido às seguintes questões:

- Divulgação de informações:
    - Negligência de gerenciamento de baixo nível;
    - Resistência da equipe;
    - Falta de comunicação entre as pessoas envolvidas no processo.
- Mudança da estrutura de desenvolvimento, já que a reorganização é um fenômeno muito comum na indústria de *software*;
- Mudança na liderança e estilo de gerenciamento afetam a eficiência da execução do processo.

Não somente fatores humanos podem tornar o processo de implantação do modelo CMMI mais moroso, mas como Larsson e Ekdahl (2004) complementam, há a necessidade de:

- Um forte comprometimento gerencial, que deve ter disposição e paciência para aguardar os resultados do esforço de melhoria no processo;
- Recursos para melhoria de desempenho, recursos financeiros e humanos, necessários para cobrir os custos de avaliação e fundos para atividades subsequentes de melhoria;
- Infraestrutura de melhoria com certo grau de estrutura organizacional para que efetivamente possa usufruir dos resultados como, por exemplo, geração de relatórios.

Ebert e De Man (2008) relatam que o SEI reporta no sumário anual a duração de mais de 3 anos como um tempo significante para evolução de maturidade de nível 1 do CMMI para nível 3, o qual requisita processo definido e conhecimento de engenharia compartilhado. No caso da empresa estudada (Alcatel-Lucent), foi possível alcançar em poucos meses, eles integraram e utilizaram o conhecimento de 3 dimensões: projeto, produto e processo.

- Conhecimento do projeto: é o conhecimento a respeito de recursos, funcionalidades e requisitos, produto de trabalho, orçamento, tempo, entregas, incrementos, qualidade almejada e parâmetros de desempenho. Conhecimento de projeto está fortemente relacionado com conhecimento de produto e processo.
- Conhecimento de produto: é o conhecimento a respeito das características do produto e como eles se relacionam com outros produtos, padrões e protocolos.
- Conhecimento de processo: é o conhecimento a respeito do processo de negócio, *workflow*, responsabilidades, tecnologia suportada e interfaces entre processos.

Compartilhar e reusar o conhecimento do processo é o mais difícil das três dimensões. Tendo um entendimento do processo organizacional, o próximo passo é entender como processos são transformados (*tailored*) em projetos ou linhas de produtos. Entender como os elementos de processo se relacionam e como são transformados é o próximo passo para integrá-los ao processo de desenvolvimento de *software*.

A qualidade do processo de *software* é importante e, como já visto, é possível definir, gerenciar, medir e melhorar. Por outro lado, a qualidade do *software* também é importante e fortemente dependente da qualidade do processo pelo qual ele é mantido e construído. Como colocado por Boehm e Turner (2003), a preocupação com qualidade do produto foi um dos propulsores para disseminação dos princípios ágeis de desenvolvimento.

Kahkone e Abrahamsson (2004) colocam que a abordagem de desenvolvimento ágil é uma opção viável para muitas empresas que produzem *software* no ambiente de negócio volátil e que tem gerado muito interesse no campo de engenharia de *software* nos últimos anos. Pino, Garcia e Piattini (2008) constataram um aumento desse interesse no estudo da melhoria do processo de *software* em pequenas empresas por parte da comunidade de engenharia de *software*. Em seus estudos, os autores verificaram que os esforços na melhoria do processo têm sido conduzidos em 47% por empresas que têm de 10 a 49 funcionários, 33% por empresas com menos de 10 funcionários, totalizando 80% das empresas. Somente 20% é reportado para empresas de tamanho médio (entre 50 a 249 funcionários). Concluem que as empresas de pequeno porte estão preocupadas em melhorar seus processos; outra descoberta relevante foi que o modelo de melhoria mais utilizado tem sido o CMMI.

Garcia e Pacheco (2010) acrescentam que o CMMI permite que as pequenas empresas entreguem de forma mais consistente melhores produtos e serviços no prazo, com maior qualidade e dentro dos custos previstos. As iniciativas de melhoria dos processos são afetadas por recursos financeiros, recursos humanos, objetivos de negócio, cultura da organização e visão. Em outro estudo Garcia, Pacheco e Calvo-Manzano (2010) procuraram demonstrar que pequenas empresas podem usar uma ferramenta *web* e o modelo CMMI como um *framework* para fortalecer as práticas de gerenciamento ágil, melhorando o desempenho de projetos e atingindo mais alto nível de capacidade CMMI. A ferramenta *web* seria importante para prover aos membros da equipe um local de fácil acesso, onde poderiam registrar o conhecimento adquirido, medidas de progresso e pontos de falhas e sucesso de projeto, permitindo que o gerente de projeto possa usar essa base como fonte para sua análise.

Lester et al. (2010) expõe que empresas menores geralmente possuem menos formalidades no estágio de planejamento já que sua força gerencial está compartilhada entre vários projetos. Complementa que os projetos são geralmente menores e podem ser devidamente realizados sem muito planejamento, favorecido pelo maior comprometimento, visto que geralmente há poucos *stakeholders* implicando em menor esforço de integração de comunicação. Em seu estudo, ao analisar seis empresas (quatro de tamanho pequeno e duas de tamanho médio), o autor conclui que em geral organizações de médio porte tendem a ter um desempenho melhor do que pequenas empresas e que claramente o CMMI pode contribuir com pequenas e médias empresas na questão de melhoria do planejamento.

Staples et al. (2007) acrescenta, no estudo efetuado por ele, classificado como exploratório, que as empresas não adotam o CMMI porque são geralmente pequenas, os serviços são dispendiosos e a organização não tem tempo de usar esses serviços, porque utilizam outra abordagem para melhoria do processo de *software* (como, por exemplo, o ágil). O autor complementa que outro motivo para a não adoção do CMMI por pequenas empresas poderia ser decorrente do orçamento necessário para sua implantação *versus* o tempo que elas teriam disponível para investir na sua aplicabilidade. Um ponto importante levantado foi a percepção negativa que as pequenas empresas tinham sobre os benefícios do CMMI, mesmo que nunca tivessem passado por uma experiência de adoção. O autor acredita que isso ocorra devido ao fato de que as áreas de processo e metas visam a garantia da qualidade do processo conflitando com a visão de pequenas empresas que questionam se o alto custo e tempo requerido para sua adoção de fato agregam valor às atividades que contribuem para a qualidade do produto. Outro ponto é que a adoção de metodologia ágil pelas pequenas empresas explica-se pelo fato de o objetivo ser a qualidade do produto ao invés de ser a garantia da qualidade do processo, por ter a percepção de que requer baixo custo e tempo para experimentar e adotar métodos ágeis.

Santos (2007) expõe que aproximadamente 80% das *software houses* ao redor do mundo são pequenas empresas. Comparadas às grandes empresas, as pequenas geralmente têm vantagens em termos de agilidade, desempenho, motivação e foco. O que elas não têm é a validação de que seu processo de desenvolvimento também

foca na qualidade de entrega do *software* – uma validação tipicamente reservada às grandes empresas que têm adotado modelos de maturidade como CMMI.

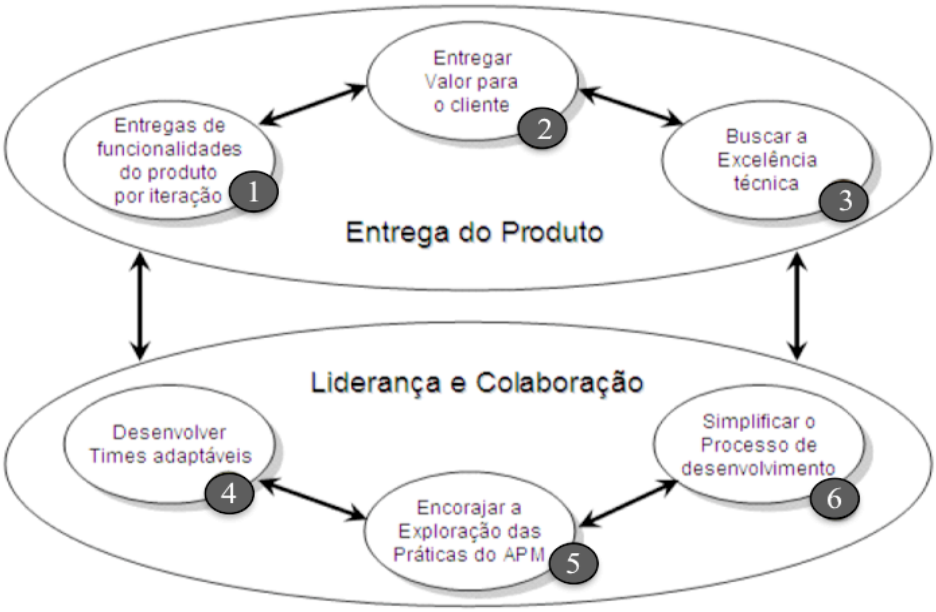
A empresa estudada por Santos (2007) aplicava o *SCRUM* (método ágil) para gerir os projetos e levou 14 meses para adaptar-se ao CMMI. Foi necessário incluir no projeto horas a mais que até então eram consideradas como administrativas (treinamento, estudo, pausas do trabalho, etc.). Essa ação foi considerada como uma abordagem não ágil, mas necessária para considerar o tempo decorrido do projeto e não somente das horas trabalhadas. O próximo passo foi documentar cada final da iteração do desenvolvimento e reservar um tempo para revisões necessárias ao processo. No final, o processo estava mais formal, porém sem perder as características da abordagem ágil. Na visão do autor, o CMMI não impõe nenhum método de trabalho, mas força as empresas a repensar sobre seu método de trabalho. O ponto positivo foi que a empresa passou a estar mais confiante sobre seu processo. O CMMI foi de grande ajuda ao possibilitar que a empresa pudesse conhecer o que estava fazendo e o que deveria estar de acordo com os seus próprios procedimentos.

4.2. DESENVOLVIMENTO ÁGIL

Boehm e Turner (2005) colocam que o gerenciamento ágil de projetos segue algumas premissas: uso de ciclos iterativos e curtos; presença ativa do cliente para definir, priorizar e verificar os requisitos do produto, desenvolvimento incremental; auto-organização e autodisciplina. O estudo de Misra et al. (2009) determinaram os fatores que levaram ao sucesso de projetos de desenvolvimento de *software* que adotaram as práticas do gerenciamento ágil: colaboração, comprometimento do cliente, tempo da tomada de decisão, cultura corporativa, controle, características pessoais, cultura social, treinamento, aprendizagem e satisfação do cliente.

Highsmith (2004) acrescenta que as empresas precisam desenvolver uma cultura que promova a adaptação para absorver mudanças; ter algumas regras que encoraje a auto-organização, combinada com autogestão; ter colaboração intensa e interação entre os membros da equipe. Disso resultam seis princípios que orientam o uso de APM (*Agile Project Management*), conforme Figura 18:

Figura 18 – Princípios da gestão ágil de projetos



Fonte: Modelo de Highsmith (2004)

Highsmith (2004) encoraja as entregas iterativas, com o objetivo de diminuir as incertezas e riscos ao longo do processo de desenvolvimento de produto; prega a simplificação do processo de gestão. Segundo o autor, as iterações são definidas com base nas palavras-chaves: iterativo, baseado em funcionalidades, prazo pré-definido (*timeboxed*) e incremental. A partir de uma construção inicial do produto essa visão pode ser estendida em ciclos sucessivos seguidos de revisões e adaptações. Lippert (2004) acrescenta que este é um importante conceito em projetos de desenvolvimento ágil pois fornece a ideia de integração contínua, significa que mudanças e melhorias são realizadas em pequenos ciclos.

Baskerville et al. (2001) argumentam que o tradicional desenvolvimento linear de *software* e com *overhead* de processo não consegue sobreviver numa cultura de desenvolvimento em que o pensamento é “primeiro ao mercado”. Outra característica que a pressão do tempo implica é o desenvolvimento orientado a versões, o que impulsiona o desenvolvimento paralelo por habilitar o desenvolvimento incremental. Uma vez que a maior porção do trabalho deve ser dedicada à primeira versão por conter as principais funcionalidades e características, as próximas versões deveriam ser dedicadas a implementações menores, correções de erros e *bugs*. As especificações também devem sofrer uma pressão por concentrar as funcionalidades e características mais importantes e relevantes para

que outras de menor prioridade fiquem para próximas versões. A especificação deverá ocorrer por ciclos, como o desenvolvimento.

Segundo Middleton (2001), a mudança de paradigma requerida para *software* é ver que especificação em papel e programas não finalizados é um efeito de estoque ou custo do processo. O real custo não é o armazenamento do papel; o custo real está em erros tardiamente descobertos na documentação que, com o passar do tempo, teve detalhes de situações que não foram descritos ou mesmo porque o projeto tenha mudado, tornando a correção destes erros muito mais cara.

Segundo Petersen (2011), há poucos estudos que focam na comparação direta dos modelos de processo em geral. A necessidade de clientes, em termos de funcionalidades e qualidade, geralmente envolvem a alta volatilidade dos requisitos, o que requer alta flexibilidade das empresas de *software*. Contudo, mais e mais empresas de *software* começam a adotar métodos ágeis e incrementais e o número de estudos recentes em métodos ágeis tem aumentado. De acordo com o estudo dirigido, os autores puderam perceber algumas melhorias ao adotar práticas de desenvolvimento ágil ao modelo tradicional, concluindo que um dos maiores problemas do modelo tradicional é o desperdício de requisitos; por sua vez, o desenvolvimento ágil trabalha com iterações, promove um escopo reduzido que ajuda na acurácia da estimativa permitindo que o escopo de requisitos seja apropriado ao projeto, conforme Tabela 10.

Tabela 10 – Melhorias percebidas ao acrescentar práticas ágeis à abordagem tradicional

Área de Processo	Descrição
Requisitos	Requisitos mais estáveis proporcionam menor retrabalho
Requisitos	Tudo o que iniciado foi implementado
Requisitos	Estimativas foram mais precisas
Verificação	*Feedback* de testes e detecção de falhas foram antecipadas
Verificação	O tempo ocioso da fase de testes foi reduzido
Gestão do Projeto	Maior colaboração das pessoas reduziu a quantidade de documentação não utilizada

Fonte: Petersen (2011)

O trabalho de Shenhar e Dvir (2007) faz uma comparação entre a gestão ágil e tradicional de projetos, conforme ilustrado na Tabela 11.

Tabela 11 – Comparação gestão ágil x tradicional

Abordagem	Tradicional	Ágil
Metas do projeto	Foco no tempo, custo e requisitos de qualidade	Foco no negócio, atingir múltiplos critérios de sucesso
Plano do Projeto	Conjunto de atividades a serem executas conforme o planejamento com o objetivo de atender custo, prazo e qualidade	Ciclo/processo com o objetivo de atender a meta esperada e resultado para o negócio
Planejamento	Realizado uma vez no início do projeto	Realizado no início e reavaliado sempre que necessário
Abordagem Gerencial	Rígida, com foco no plano inicial	Flexível, adaptável
Execução	Previsível, mensurável	Imprevisível, não mensurável
Influência da organização	Mínima, a partir do *Kick-off* do projeto	Impacto no projeto ao longo da execução
Controle de projeto	Identifica os desvios a partir do plano inicial e os corrige para seguir conforme o planejado	Identifica as mudanças e ajusta o plano
Aplicação de metodologia	Aplicação genérica de forma similar a todos os projetos	Adaptação do processo dependendo do projeto
Estilo de gestão	Um modelo atende todos os tipos de projetos	Abordagem adaptativa, um único modelo não atende todos os projetos.

Fonte: Modelo de Shenhar e Dvir (2007)

Segundo Lippert et al. (2003), o maior problema com o processo de desenvolvimento tradicional é que ele deixa uma grande lacuna entre clientes e desenvolvedores, já os métodos ágeis, com intuito de prevenir desentendimentos a respeito dos objetivos e resultados do projeto, envolvem o cliente desde o início do projeto tornando-o um parceiro do processo de desenvolvimento.

Talby et al. (2006) colocam que, no projeto sob abordagem tradicional, a fase de testes ocorre após a conclusão do desenvolvimento do sistema, para o qual é enviada uma versão para o grupo de QA (*Quality Assurance*) e este devolve grande volume de defeitos que geralmente excedem os recursos disponíveis para corrigi-los. É necessária a execução de um longo processo de priorização dos defeitos e dificilmente todos serão sanados até a entrega final do sistema. Os projetos ágeis substituem esse processo por uma simples regra: corrigir cada defeito assim que é descoberto. Os autores concluem que corrigir defeitos o quanto antes possui algumas vantagens como: requerer menos tempo para corrigir o defeito; o desenvolvedor pode trabalhar numa base de código mais estável, promovendo

velocidade no desenvolvimento e evitando desperdiçar tempo com o planejamento; priorização dos defeitos a serem corrigidos.

Fogelstrom et al. (2010) ressaltam um importante princípio do desenvolvimento ágil: a simplificação. Os requisitos devem ser priorizados com o intuito de produzir versões e o desenvolvimento pode ser iniciado mesmo que todos os requisitos não tenham sido devidamente especificados, iniciando por maior prioridade, consistindo no planejamento da versão e seus refinamentos.

Um ponto levantado por Lindvall et al. (2004) é que as empresas identificam a necessidade de implementar processos mais flexíveis capazes de adaptar-se à volatilidade dos requisitos quando ocorre uma mudança repentina de requisitos e deparam-se com uma especificação mal definida, executada em alto nível e que muitas vezes encontra-se obsoleta, decorrente do tempo consumido para sua elaboração. Port e Bui (2009) argumentam que é altamente reconhecido que em muitos projetos de desenvolvimento de sistemas de informação nem todos os requisitos são implementados e que, de alguma forma, a priorização de requisitos é fundamental. Com expectativas de clientes cada vez maiores, tempo de entrega reduzido e recursos limitados, a priorização tem sido uma estratégia comum para limitar o escopo e entregar funcionalidades mais essenciais o quanto antes possível. Eles puderam notar, em suas pesquisas, que a priorização de requisitos é uma atividade custosa e muito complexa tanto na abordagem tradicional quanto ágil.

Os autores sugerem que os pontos fortes do desenvolvimento ágil e tradicional sejam combinados a fim de propor uma nova estratégia de priorização. Em seus estudos agregaram atividades do tradicional à abordagem ágil, denominada AG2, e agregaram atividades ágeis na abordagem tradicional, nomeando como HY. Por simulações, comprovaram que ambas tiveram resultado superior ao método ágil ou tradicional, ou seja, concluem ser arriscado praticar estritamente a estratégia tradicional ou ágil. Por esta razão, os autores inferem que, na prática, a estratégia de combinar os métodos é a mais acertada e acrescentam que, ao adicionar atividades da abordagem tradicional ao ágil, como avaliação de custos e priorização por ordenação de Pareto, o desempenho não é tão bom quanto considerar a abordagem tradicional, como majoritária, e acrescentar práticas ágeis, porém mostrou-se menos custosa. Como sugerido na simulação tanto AG2 quanto HY se adaptaram facilmente ao dinamismo e a estratégia HY parece ser, de modo geral, de melhor

desempenho, mas o preço de implantação dessa abordagem é o mais caro. Apesar de AG2 não ter um desempenho tão bom quanto HY, é significantemente melhor que o puramente tradicional ou ágil.

Lycett et al. (2003) colocam que a abordagem ágil é fortemente a favor da comunicação humana e colaboração ao invés de atividades repetidas e definidas como mecanismo para desenvolver *software* de qualidade. Concluem que, na prática, a dificuldade que muitas organizações enfrentam com a abordagem ágil reside na mudança cultural necessária para convencer a gerência dos benefícios e a viabilidade no contexto de pressão do mercado global e práticas regulatórias.

Lee e Yong (2010) reforçam que os princípios ágeis enfatizam a comunicação face a face, times auto-organizáveis, colaboração, mudanças de requisitos, entregas frequentes, excelência técnica, entre outros. Mas de todas essas características, o elemento mais crítico é a comunicação cliente com membros da equipe.

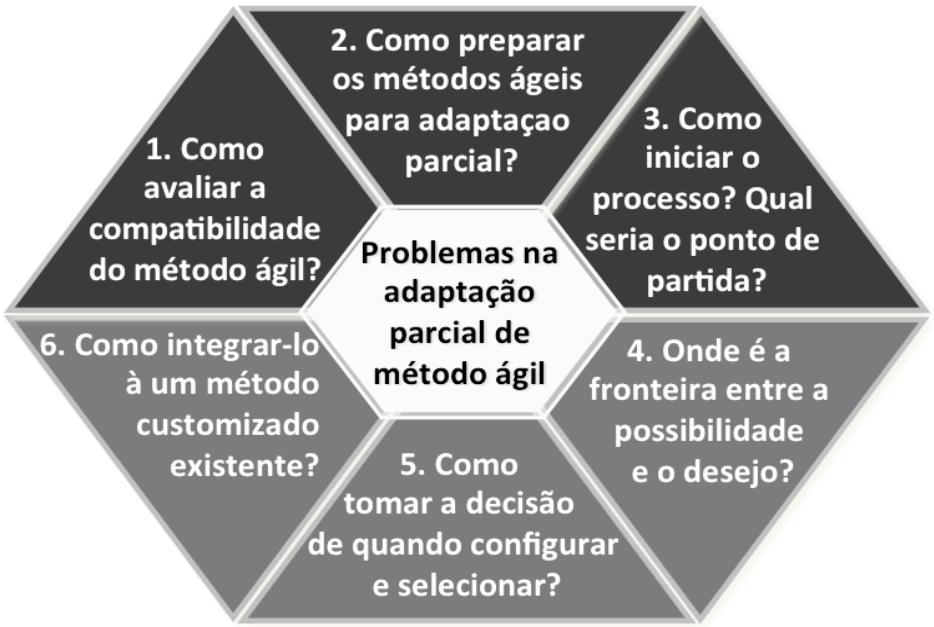
Diante de todas essas características do desenvolvimento ágil, Nielsen e Mcmunn (2005) enfatizam a importância de implantar esses conceitos de projeto em projeto, diluindo o impacto e os riscos das mudanças ao longo do tempo. Dessa forma, os usuários também são introduzidos gradualmente e podem perceber, ao longo do tempo, a importância da participação ativa no projeto. Essa iteração da área de TI e negócio deve ser apoiada pela área gerencial da empresa.

Nessa linha de introduzir os conceitos ágeis gradualmente, Mikulenas, Butleris e Nemuraite (2011) argumentam que passados somente 8 anos desde a 1º publicação do Manifesto Ágil, o conceito de desenvolvimento ágil ganhou forte destaque no campo do desenvolvimento de sistemas de informação. Mas muitas organizações receiam mudar suas metodologias de trabalho de forma drástica e ter que assumir riscos pela escolha, ainda mais se considerar o último relatório CHAOS publicado em 2011 pela *Standish Group,* no qual é apresentado que somente 32% dos projetos de *software* podem ser classificados como sendo de sucesso, por terem atingido a meta de custo e tempo. As organizações geralmente não preferem reconstruir seus métodos e processos de imediato, em sua maioria optam por fazer por projeto ou mesmo estender o método existente para agregar algumas partes do método ágil. O problema é que os métodos ágeis são apresentados, atualmente, como uma solução monolítica sem um *roadmap* formal de como customizar e

configurá-lo para uma adaptação parcial, fora que há uma variedade de métodos ágeis para escolher.

Os autores propõem uma visão do problema, generalizado em seis subproblemas, de adaptar parcialmente os métodos ágeis, conforme ilustrado na Figura 19.

Figura 19 – Decomposição do problema de adaptação parcial do método ágil



Fonte: Mikulenas, Butleris e Nemuraite (2011)

Os autores expõem uma breve explicação sobre os subproblemas (Tabela 12).

Tabela 12 – Subproblemas da adaptação parcial do método ágil

Subproblema	Descrição
1. Como avaliar a compatibilidade do método ágil?	Métodos ágeis são frequentemente utilizados como alternativa ao modelo tradicional. Há restrições tanto no nível do projeto quanto organizacional que devem ser ajustados para sua adoção por TI.
2. Como preparar os métodos ágeis para adaptação parcial?	A demanda atual é estender o método existente na empresa por adicionar algumas práticas ágeis, há necessidade de identificar quais práticas serão utilizadas no contexto atual.
3. Como iniciar o processo? Qual é o ponto de partida?	A atualização de um método é uma atividade que gera alto custo para TI tanto pelo risco da implantação quanto de treinamento da equipe que associado a pressão de mercado direciona a buscar o momento mais propicio para início da atualização.

Subproblema	Descrição
4. Onde é a fronteira entre a possibilidade e o desejo?	Dependendo do método existente na empresa pode haver espaço para agregar muitas práticas ágeis, podendo elevar e muito o esforço de adaptação, sendo assim, deve-se balancear os elementos que são possíveis ou desejáveis de serem utilizados.
5. Como tomar a decisão de quando configurar ou selecionar?	Envolvem questões de como selecionar e construir fragmentos de um método ágil concreto.
6. Como integrá-lo a um método customizado existente?	Depois de estender o método existente com práticas ágeis deve-se tomar a decisão de implementá-lo.

Fonte: Mikulenas, Butleris e Nemuraite (2011)

Mikulenas, Butleris e Nemuraite (2011) concluem que não há um consenso universal de que os métodos ágeis são facilmente adaptáveis, há um conjunto de características de ambiente tanto da organização quanto do projeto que facilitam sua utilização, o qual os autores denominam como requisitos de agilidade descrito como constituído de dois parâmetros: tipo de medida (quantitativa ou qualitativa) e tipo de conteúdo (técnico, social, negócio, psicológico, etc.).

Já o estudo de Mafakheriab et al. (2008) propõem indicadores de agilidade para identificar as empresas que adotam o desenvolvimento ágil que seriam compostos pelos parâmetros apresentados na Tabela 13.

Tabela 13 – Parâmetros que afetam a agilidade

N.	Parâmetro	Objetivo
1	Dinamismo	Habilidade de mudar os requisitos Entregas frequentes de *softwares* funcionais
2	Tamanho da equipe	Ter um time pequeno
3	Comunicação	Ter a proximidade do cliente Redução da documentação
4	Teste	Ser capaz de testar os resultados mais frequentemente
5	Perfil e conhecimento dos desenvolvedores	Não ter nenhum pessoal no nível 1B (aprendiz) ter pelo menos 30% no nível 2 e 3 (pleno e sênior)
6	Cultura	Ter alto grau de liberdade para qualquer um envolvido na equipe

Fonte: Modelo de Mafakheriab et al. (2008)

Ao estudar os indicadores de agilidade, Qumer e Henderson-Sellers (2008) definem que o método ágil é uma função envolvendo agilidade, abstração, pessoas,

processo, produto, ferramentas, conhecimento e governança. Os autores realizaram uma pesquisa na qual puderam identificar o impacto e a importância da governança para o sucesso da operacionalização da iniciativa ágil em grandes organizações. De fato foi reconhecido por 77% dos participantes que a abordagem ágil é produtiva em projetos pequenos e médios, mas ao adicionar elementos de governança seria possível escalar para projetos grandes e complexos. Os autores propõem uma forma para calcular a agilidade envolvendo 4 dimensões denominado 4-DAT, que permite uma avaliação numérica do grau de agilidade no nível de processo e práticas de modo a considerar se o método utilizado de fato é ágil. Neste sentido construíram um questionário, ilustrado na Tabela 14.

Tabela 14 – Quatro Dimensões 4-DAT

Dimensão 1: Escopo do Método	
Escopo	**Descrição**
1. Tamanho do projeto	O método é suportado para pequenos, médios ou grandes projetos?
2. Tamanho da equipe	O método é suportado por equipes grandes ou pequenas? Mesmo para múltiplas equipes?
3. Estilo de desenvolvimento	Qual o estilo de desenvolvimento que o método suporta (iterativo, rápido)?
4. Estilo de código	O método especifica o estilo de código (simples ou complexo)?
5. Ambiente tecnológico	Qual ambiente tecnológico (ferramentas, compiladores) que o método utiliza?
6. Ambiente físico	Qual o ambiente físico (localizados no mesmo local ou distribuídos) que o método suporta?
7. Cultura do negócio	Qual é o tipo de cultura do negócio (colaborativo, cooperativo ou não-colaborativo)?
8. Mecanismo de abstração	Qual é o mecanismo de abstração (orientado a objetos, orientada a agentes) do método?
Dimensão 2: Caracterização da Agilidade	
Características	**Descrição**
1. Flexibilidade	O método acomoda mudanças esperadas e não esperadas?
2. Velocidade	O método produz resultados rapidamente?
3. Enxuto	O método segue a linha de produzir no menor tempo possível, uso econômico, simplicidade e uso de instrumentos de qualidade?
4. Aprendizado	O método emprega o uso do conhecimento e a experiência para criar um ambiente de aprendizado?
5. Capacidade de resposta	O método apresenta sensibilidade?

Dimensão 3: Caracterização do Valor Ágil	
Valores Ágeis	**Descrição**
1. Indivíduos e interações sob processos e ferramentas	Qual valor prático de pessoas e interações sob processos e ferramentas?
2. Codificação sob documentação	Qual o valor prático de codificação sob documentação?
3. Colaboração com cliente sob negociação de contrato	Qual o valor prático de colaboração com cliente sob negociação de contrato?
4. Respostas as mudanças sob seguir o planejamento	Qual o valor prático de respostas as mudanças sob seguir o planejamento?
5. Manter o processo ágil	Qual o valor prático de ajudar a manter o processo ágil?
6. Manter o custo efetivo do processo	Qual o valor prático de ajudar a manter o custo efetivo do processo?

Dimensão 4: Caracterização do Processo de *Software*	
Processo	**Descrição**
1. Processo de desenvolvimento	Quais práticas cobrem o processo de ciclo de vida e testes (garantia da qualidade)?
2. Processo de gerenciamento de projeto	Quais práticas estão presentes no gerenciamento do projeto?
3. Processo de suporte e controle de configuração	Quais práticas estão presentes no processo que possibilita o gerenciamento de configuração?
4. Processo de gerenciamento de processo	Quais práticas cobrem o processo que é requerido para gerenciar o próprio projeto?

Fonte: Qumer e Henderson-Sellers (2008)

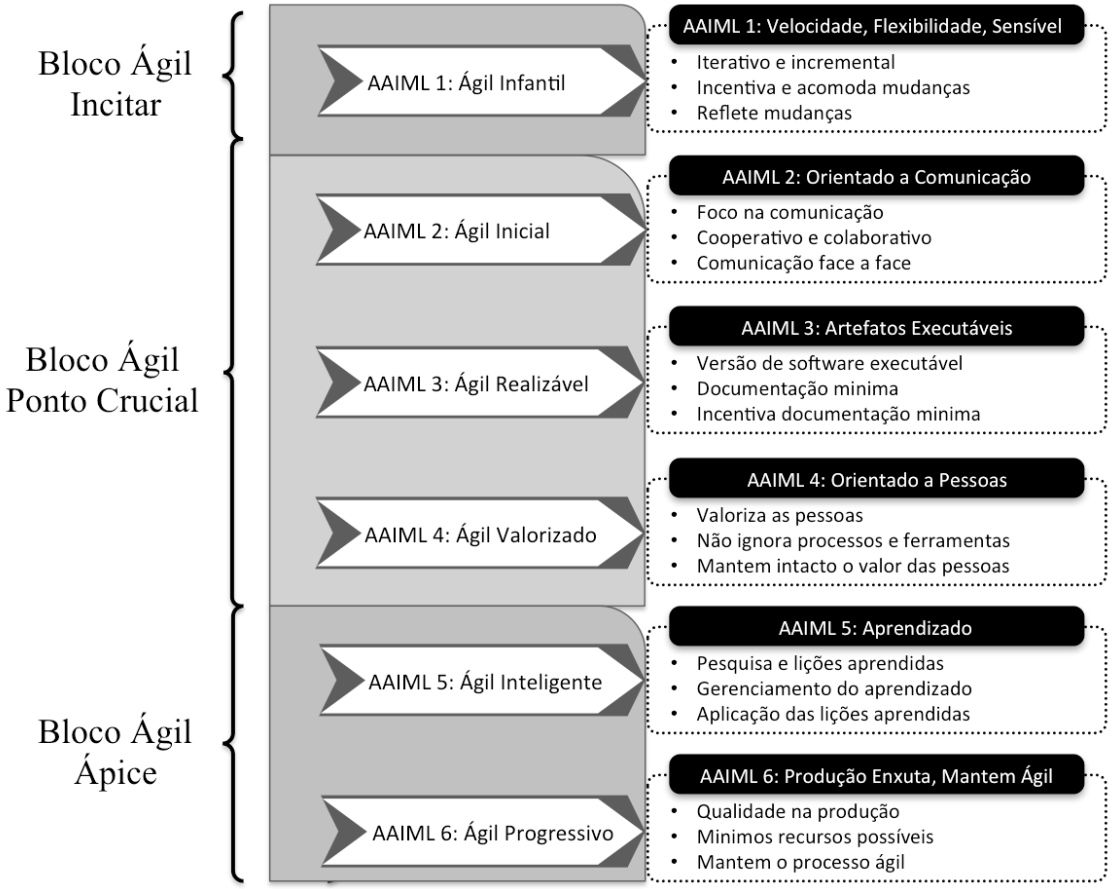
Qumer e Henderson-Sellers (2008) utilizaram o conceito envolvido no modelo 4-DAT para derivar a definição de método ágil como:

Um método de desenvolvimento de *software* é dito como sendo um método de desenvolvimento ágil quando o método é focado em pessoas, orientado a comunicação, flexível (pronto para adotar mudanças esperadas ou não a qualquer tempo), velocidade (encoraja o desenvolvimento rápido e iterativo de produtos em pequenas versões), *lean* (foca no prazo e custo reduzido e melhoria da qualidade), capacidade de resposta (reage apropriadamente a mudanças esperadas ou não) e aprendizado (foca na melhoria durante e após o desenvolvimento do produto).

Qumer e Henderson-Sellers (2008) utilizaram a ferramenta 4-DAT para construir um modelo composto por 6 estágios de agilidade denominado AAIM (*Agile Adoption and Improvement Model*) subdivididos em três grandes blocos: 1. Incitar, 2. Ponto

Crucial, 3. Ápice. Os autores acrescentam que é um modelo que pode guiar uma organização de desenvolvimento de *software* a adotar e melhorar práticas ágeis em uma situação ou projeto específico.

Figura 20 – Modelo AAIM (*Agile Adoption and Improvement Model*)



Fonte: Qumer e Henderson-Sellers (2008)

Com base na experiência das organizações estudadas, Lindvall et al. (2004) acreditam que as práticas ágeis cobrem as necessidades das grandes organizações e especialmente das pequenas. O desafio não é aplicar práticas ágeis ao projeto, mas sim como difundi-las pela organização, evitando duplo trabalho da equipe causado pelo conflito das práticas ágeis e tradicionais.

4.3. LEAN DEVELOPMENT

Segundo Middleton (2001) a abordagem *lean* é holística e pode abranger o processo de *software*. Técnicas *lean* trabalham para rapidamente identificar por que e como uma pessoa está cometendo erros. Os dados necessários são providos para assistir

uma mudança necessária na infraestrutura da organização. O aprendizado organizacional, alinhamento das pessoas, processos e objetivos permitem que as pessoas produzam o trabalho com a qualidade requerida. Produção *Lean* está presente como um sistema de gerenciamento completo dedicado a eliminar desperdícios.

Desenvolvimento de *software lean* indica que problemas na qualidade de *software* estão frequentemente relacionados à organização, seus hábitos de recrutamento, retenção e motivação. Qualidade de *software* pobre é geralmente uma manifestação de profundos problemas dentro da organização. Para resolver esses problemas é necessário haver uma mudança organizacional que geralmente requer rápidos resultados de ações de baixo custo, mas tende a ser um ciclo vicioso porque mudanças requerem motivação, desencadeada e sustentada por resultados.

Segundo Middleton, Flaxel e Cookson (2005) *software lean* é significante porque muitas das grandes organizações estão usando técnicas *lean* em suas operações de manufatura e estão descobrindo que o *software* é uma parte substancial de seus produtos, mas não é gerenciado usando os princípios *lean*. Petersen e Wohlin (2010) acrescentam que o objetivo geral do desenvolvimento *lean* é atingir um fluxo de produção contínua e suave com máxima flexibilidade e mínimo desperdício de processo. Toda atividade e produto de trabalho que não contribui com o valor do cliente deve ser considerado desperdício. Considerando o *feedback* da indústria, parece promissor uma abordagem para continuamente melhorar o processo de *software* para tornar-se mais *lean* (enxuto). Middleton (2001) reforça este conceito colocando que os métodos *lean* provem *feedback*, o qual motiva a equipe e prevê evidências claras de desempenho diário, os problemas são identificados e então colocados sob a responsabilidade do gerente para resolução.

Os princípios *Lean* não são prescritivos tanto que cada implementação poderá ser diferente dependendo do contexto e restrições da organização. Os pequenos ciclos de *feedback* aceleram o aprendizado organizacional e habilita a redução contínua no processo e variação do produto proporcionando ganho de qualidade e produtividade.

Segundo Petersen e Wohlin (2010) desenvolvimento *lean* de *software* compartilha princípios com o ágil em relação ao gerenciamento e liderança de pessoas, o foco na qualidade e excelência técnica e foco em frequentes e rápidas entregas de valor

ao cliente. O que distingue o *Lean* do ágil é o foco na perspectiva final de todo o fluxo de desenvolvimento. Para suportar este foco, ao final, o *lean* propõe um número de ferramentas para analisar e melhorar o mapeamento de fluxo, gerenciamento de estoque (documentação e código) e sugere utilização de sistemas como *Kanban*. Os autores complementam que as métricas obtidas com ferramentas e princípios *Lean* podem afetar a tomada de decisão devido a:

- Priorização de requisitos é suportada;
- As métricas ajudam a alocar a equipe;
- As métricas provêm transparência para equipe e gerente de projeto de que trabalho deve ser feito no futuro e o que foi terminado;
- Melhoria dos indicadores do processo de *software* propicia o uso de métricas para identificar problemas e alcançar melhorias de longa perspectiva.

Secundariamente ocorre:

- Aumento no foco do desenvolvimento contínuo por limitar o número permitido de requisitos em estoque;
- Integrações mais frequentes e antecipadas e testes do sistema para aumentar a qualidade.

A organização deve prover dados, informações e conhecimentos para tomada de decisão e, como visto, os princípios *Lean* podem auxiliar nesta coleta de métricas. Mas é necessário que o processo seja maduro com o objetivo de prover estas métricas de forma contínua. Pensando em qualidade no processo, Veldman e Klingenberg (2009) realizaram um estudo de caso em empresa de engenharia, caracterizado por assumir as melhores práticas do *Lean Production.* Concluíram que é possível aplicar práticas do CMMI, acrescentando que é o *framework* mais conhecido para atingir a maturidade de processo. Puderam observar algumas vantagens ao implantar CMMI na empresa estudada que podem ser assim generalizadas:

- *Framework* de maturidade reduz a incerteza de tarefas e ajuda a gerenciar iterações complexas entre atores, tarefas e processo, levando eventualmente à redução de defeitos e de retrabalho.
- *Framework* de maturidade provê guia substancial para integração de processo e experiência de produto envolvendo *design* e processo.

- Integração de fornecedores pode ser aprimorada pelas áreas de processo de gerenciamento de contrato de fornecedores (nível 2) e integrados com o processo de gerenciamento de fornecedores (nível 3). Estas áreas de processo estressam o relacionamento formal, ainda que o relacionamento seja de longo prazo baseado na negociação e coordenação mútua de interesses.

4.4. QUADRO DE INTEGRAÇÃO CMMI VERSUS DESENVOLVIMENTO ÁGIL

Segundo Lycett et al. (2003) apesar de, na teoria, a abordagem de engenharia e desenvolvimento possuírem o mesmo princípio (produzir o *software* observando as necessidades internas a um custo mínimo de forma a atingir as necessidades do cliente com qualidade e flexibilidade), na prática, há abordagens com conceitos diferentes quanto à percepção de qualidade. Da perspectiva de engenharia, qualidade está fortemente relacionada ao uso de padrões de qualidade como ISO 9000 e CMM que seguem a filosofia “diga o que faz e então faça o que você diz que faz”. Na prática a certificação requer uma auditoria e assume que a qualidade do processo equivalerá a qualidade do produto. Nesta mentalidade de auditoria há necessidade de artefatos tangíveis que tendem a dificultar o desenvolvimento por produzir documentação excessiva, reduzir as atividades orientadas à comunicação e aumentar a disjunção entre codificação e documentação.

Na perspectiva ágil há maior ênfase na visão de qualidade do produto. A qualidade do processo pode direcionar a qualidade do produto, mas um processo bom não é necessariamente repetitivo ou padronizado. Na perspectiva ágil há uma distinção entre consistência e repetitividade. Saídas consistentes se utilizam de experiências similares em projetos e experiência orientada através de práticas reflexivas que são antagônicas a “seguir um processo”. Situações emergentes não podem ser tratadas como repetíveis ainda mais no contexto de desenvolvimento de *software* onde informações, restrições e requisitos sofrem mudanças repentinas.

No estudo de Lycett et al. (2003), a institucionalização da qualidade, com a implantação do CMMI, foi importante para empresa tanto no aspecto de *marketing* quanto na melhoria das práticas de desenvolvimento de *software*. Contudo, a

organização teve a impressão de que o conceito “uma solução resolve tudo” não funcionava e por isso agregou algumas atividades às práticas de CMMI de forma a acomodar as consistências, permitindo que a experiência orientada possa diferenciar algumas atividades sem comprometer as saídas tangíveis que permitem serem auditadas.

Essa abordagem não viola os princípios do CMMI, mas busca substituir a forte noção de repetível pela noção da abordagem da consistência e tomada de decisão. O processo ágil torna as práticas de engenharia mais enxutas e bem focadas. Lindvall et al. (2004) acrescentam que processos maduros requerem atividades como gerenciamento de qualidade, documentação e medição, e práticas ágeis podem ajudar organizações maduras a tornarem-se mais flexíveis.

Visando o estudo da integração do CMMI e o desenvolvimento ágil, o propósito deste capítulo é prover um quadro que possibilite visualizar quais áreas de processo do CMMI terão maior esforço de adaptação. É utilizada a seguinte nomenclatura:

- Suportado: com pouco esforço é possível atender os objetivos propostos na área de processo CMMI a partir do desenvolvimento ágil;
- Parcialmente Suportado: será necessário fazer algumas adaptações e/ou melhorias no processo de desenvolvimento ágil para suportar os objetivos da área de processo do CMMI;
- Não Suportado: há conflito entre as abordagens de desenvolvimento ágil e a área de processo do CMMI, ou seja, será necessário acrescentar atividades no desenvolvimento ágil que divergem dos princípios ágeis.

Para construção deste quadro foi utilizada como base a tabela de objetivos genéricos e específicos do CMMI, conforme Apêndice 1. O quadro está organizado de acordo com as categorias do CMMI (gerenciamento de projeto, gerenciamento de processo, engenharia e sue) e é composto pelos seguintes campos:

- Áreas de Processo: corresponde à área de processo do CMMI;
- Nível: corresponde ao nível de maturidade que a área de processo está classificado no CMMI;
- Critério: indica o esforço de adaptação;
- Explicação: expõe argumentos para a classificação atribuída ao campo Critério.

Tabela 15 – Quadro de integração CMMI x Desenvolvimento Ágil

Gerenciamento de Projeto		Esforço de Adaptação	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Planejamento do Projeto (PP)	2	Suportado	As estimativas são efetuadas por iterações e podem ser corrigidas durante o projeto. A diferença do *Lean* para ágil é que o *Lean* estabelece um plano até o final do projeto enquanto o ágil é por iterações por isso planos de longo prazo são mais vagos e os de curto prazo mais detalhados. Os riscos e necessidades de treinamento também fazem parte do plano do projeto por iterações. O compromisso e responsabilidades de cada membro da equipe, inclusive cliente, são também estabelecidos por iterações.
Controle e Monitoração do Projeto (PMC)	2	Suportado	Com as iterações curtas permite-se o monitoramento diário. O cronograma e estimativas são efetuadas por iteração, por isso que o monitoramento também é efetuado por ciclo de desenvolvimento e não do projeto todo. A intensa comunicação entre cliente e membros da equipe ajuda a difundir estas informações. As ações corretivas também são efetuadas por iterações e tendem a ser informais.
Gestão de Contrato do Fornecedor (SAM)	2	Não Suportado	No desenvolvimento ágil não há nenhuma abordagem específica para fornecedores, já que as iterações são planejadas para serem realizadas pela equipe e não por terceiros. O fato de as iterações serem curtas, a entrega de um componente de produto pelo fornecedor poderia ser considerada crítica e dificultar as atividades de integração.
Gestão de Projeto Integrado (IPM)	3	Suportado	No desenvolvimento ágil há um processo definido de projeto, o qual é direcionado por iterações, sendo estabelecido o plano, a equipe e a forma de trabalho. O envolvimento do cliente é estimulado a todo o instante com intuito de propiciar intensa comunicação de cliente e membros de equipe.
Gestão de Risco (RSKM)	3	Parcialmente Suportado	Não há uma forma ou recomendação explicita de gestão de riscos para abordagem ágil de como identificar, analisar, priorizar ou mesmo mitigar. Geralmente esta atividade fica sob os critérios do líder de projeto. Sendo assim, há necessidade de maior formalismo quanto à forma. Um ponto positivo é que as iterações curtas permitem que a identificação, análise e mitigação sejam aplicadas rapidamente.

Gerenciamento de Projeto		Esforço de Adaptação	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Gestão Quantitativa de Projeto (QPM)	4	Não Suportado	Os objetivos e gestão de desempenho de projeto são estabelecidos, porém para conseguirmos um efeito estatístico é necessário certa homogeneidade nos dados, o que geralmente não ocorre nos projetos ágeis devido ao fato de o foco estar nas pessoas e não no processo. Sendo assim, cada projeto tende a ser único na sua forma de condução.

Gerenciamento de Processo		Esforço de Adaptação	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Foco no Processo Organizacional (OPF)	3	Não Suportado	O desenvolvimento ágil tem maior foco no produto e não no processo. Os princípios e conceitos ágeis são geralmente aplicados aos projetos e não à organização. Há um estimulo quanto ao desenvolvimento do conhecimento tácito o que pode ser aproveitado entre projetos, pelo aproveitamento dos recursos humanos, mas dificulta a disseminação deste conhecimento para a organização.
Definição do Processo Organizacional (OPD)	3	Parcialmente Suportado	No desenvolvimento ágil há regras e diretrizes para a equipe, definição do ciclo de vida e critérios para forma de trabalho. A grande questão é que estes conceitos geralmente ficam na fronteira dos projetos e deve-se propiciar uma forma de esses princípios permearem pela organização.
Treinamento Organizacional (OT)	3	Parcialmente Suportado	O plano e a capacidade de treinamento limita-se às fronteiras do projeto, o que muitas vezes acaba sendo executado de modo informal, não havendo explicitamente métricas para avaliar a efetivamente deste treinamento, se não a própria evolução das iterações. Mais uma vez, há necessidade de ampliar esta fronteira do projeto para organização.
Desempenho do Processo Organizacional (OPP)	4	Parcialmente Suportado	Modelos e linhas de controle de desempenho são estabelecidas ao projeto, mas não externalizadas a organização.
Gestão do Desempenho Organizacional (OPM)	5	Parcialmente Suportado	A gestão do desempenho é efetuado no contexto do projeto e não da organização.

Engenharia		Esforço de Adaptação	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Gestão de Requisitos (REQM)	2	Suportado	A compreensão dos requisitos ocorre por intensiva comunicação do cliente com a equipe. O compromisso dos membros do projeto com os requisitos ocorre na fase de planejamento de cada iteração (ciclos de desenvolvimento) e mudanças de requisitos são rapidamente absorvidas pelo projeto. Um ponto de atenção é quanto à rastreabilidade devido ao desenvolvimento ágil pregar menos documentação.
Desenvolvimento de Requisitos (RD)	3	Suportado	O desenvolvimento ágil possui algumas técnicas para desenvolvimento de requisitos do cliente e produto como a intensa comunicação entre as partes (proximidade), programação em pares, *Test-Driven Development* e entregas rápidas de partes do sistema que já podem ser avaliadas pelo cliente (análise e validação). Outro ponto importante é quanto à priorização dos requisitos que tendem a concentrar as principais partes do sistema nas primeiras entregas.
Solução Técnica (TS)	3	Suportado	O desenvolvimento ágil possui a característica de foco no código fonte e para tanto abordam soluções alternativas, técnicas de refatoração (significa melhoria do código, mais enxuto e melhor desempenho, sem perder a funcionalidade prevista, o requisito funcional).
Integração de Produto (PI)	3	Suportado	As iterações (ciclos de desenvolvimento) é uma forma de integração contínua por assegurar a compatibilidade com versões anteriores com o objetivo de entregar o produto completo. A cada iteração este produto de trabalho é submetido à avaliação do cliente, colaborando com a avaliação da montagem de componentes do produto.
Verificação (VER)	3	Suportado	A verificação é alcançada principalmente por técnicas de teste como *Test-Driven Development* (primeiro escreve-se o teste para depois o código fonte), focando no requisito funcional. O processo de desenvolvimento tem a característica de ter a democratização do código fonte pela equipe, que muitas vezes já aplica o desenvolvimento ou a revisão por pares, até mesmo por questões de refatoração.

Engenharia		Esforço de Adaptação	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Validação (VAL)	3	Suportado	A validação do produto ou componentes do produto é efetuada em cada iteração e o principal critério de validação é a aceitação do cliente. Outro ponto importante é que o cliente é estimulado a ficar próximo da equipe de desenvolvimento permitindo uma validação contínua do produto do trabalho.

Suporte		Esforço de Adaptação	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Gestão de Configuração (CM)	2	Suportado	As linhas de bases são estabelecidas por iteração e geralmente através de testes funcionais, já que existe um compromisso de entregas curtas de executáveis do sistema (partes funcionais). As mudanças são bem aceitas durante o projeto e rapidamente absorvidas e integradas, este é um ponto que merece atenção por ter uma tendência de informalidade.
Garantia da Qualidade do Produto e Processo (PPQA)	2	Parcialmente Suportado	No desenvolvimento ágil a preocupação maior é quanto ao produto e menos ao processo. Sendo assim, há uma necessidade quanto à melhora da percepção, ao processo e, consequentemente, à coleta de dados.
Análise e Métricas (MA)	2	Parcialmente Suportado	Durante as iterações é possível coletar vários dados e estabelecer métricas, mas deve haver disciplina, pois os ciclos de desenvolvimento mais curtos propiciam maior informalidade. Outro ponto importante a considerar é quanto à medição do progresso que tende a ser por ciclo e não do projeto.
Resolução e Análise de Decisão (DAR)	3	Parcialmente Suportado	O processo de avaliação de alternativas existe, porém não é executado de modo formal com critérios e métodos de avaliação.
Resolução e Análise Causal (CAR)	5	Parcialmente Suportado	A identificação das causas e as ações necessárias para correção são endereçadas, porém de modo informal e implementadas rapidamente.

Fonte: Produzida pela autora

Tendo em vista o exposto, é possível notar que o esforço de adaptação requerido por nível de maturidade seria:

Nível 2: Há duas áreas de processo parcialmente suportados, Garantia da Qualidade de Produto e Processo (PPQA) e Análise e Métrica (MA). Mas a adaptação tende a não ser tão dispendiosa porque os objetivos específicos destas áreas são atendidos pelo produto, sendo necessário estender a fronteira até o processo, com o cuidado de aumentar a formalização, porém, sem exageros. O ponto de atenção é quanto à área Gestão de Contrato do Fornecedor (SAM), identificada como “não suportado”, e isso ocorre porque o desenvolvimento ágil foi idealizado para equipes pequenas e autogerenciáveis. Para esse ponto é necessário certo esforço de gestão.

Nível 3: Há quatro áreas de processo parcialmente suportados: Gestão de Risco (RSKM), Definição do Processo Organizacional (OPD), Treinamento Organizacional (OT) e Resolução e Análise de Decisão (DAR). A recomendação seria a mesma do nível 2, ou seja, preocupação quanto ao processo e formalismo, acrescido pelo tempo necessário à adaptação, já que há mais áreas de processo não conformes. Há uma área não suportada, Foco no Processo Organizacional (OPF), e para este ponto é necessário ter maior atenção, visto que os princípios ágeis não contemplam o contexto da organização, apenas o projeto focado no produto. De forma geral, é possível atingir este nível de maturidade sem violar por demais os princípios ágeis, mas provavelmente consumirá maior tempo de adaptação.

Nível 4: Há um processo parcialmente suportado Desempenho do Processo Organizacional (OPP) e, se considerar o esforço que seria necessário para adaptar-se, a área Foco no Processo Organizacional (OPF) do nível 3. É possível notar que mencionar o desempenho do processo organizacional seria desviar dos princípios ágeis propostos pelo manifesto ágil. Há ainda uma área não suportada, Gestão Quantitativa de Projeto (QPM), já que para coleta estatística é necessária homogeneidade entre os projetos e, na abordagem ágil, o foco é o produto e as pessoas envolvidas no projeto, não no processo, assim, a condução de cada projeto tende a ser única. Por esses motivos, não seria possível atingir o nível 4 de maturidade sem distorcer os princípios ágeis propostos. Essas limitações poderiam explicar por que (Capítulo 2, *Análise Bibliométrica*) foram identificados 28 artigos preocupados com a difusão do conhecimento e outros 36 artigos que se propõem a discutir sobre técnicas de gestão de projetos empregadas na abordagem ágil, incluindo técnicas, métricas e formas de monitoração.

Nível 5: Há dois processos parcialmente suportados: Gestão do Desempenho Organizacional (OPM), os motivos do alto esforço de adaptação descritos no nível 4 se mantém; Resolução e Análise Causal (CAR) que, na abordagem ágil, é efetuada de modo informal e, ao longo do projeto, seguindo o princípio *Lean* de evitar desperdício. O nível 5 de maturidade não seria possível sem distorcer os princípios ágeis propostos.

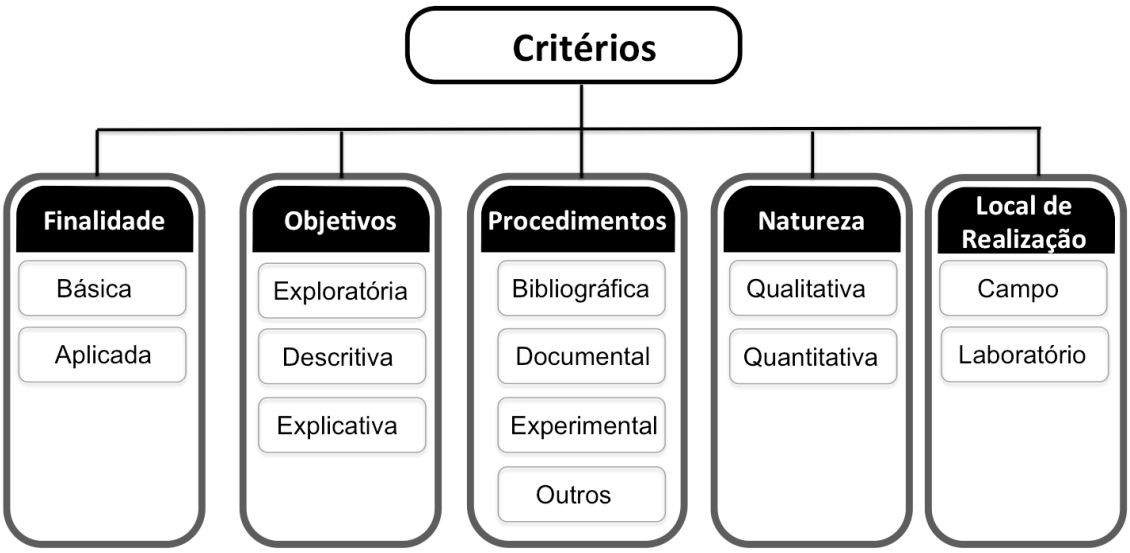
5. METODOLOGIA

Segundo Kourganoff (1990), “Pesquisa é o conjunto de investigações, operações e trabalhos intelectuais ou práticos que tenham como objetivo a descoberta de novos conhecimentos, a invenção de novas técnicas e a exploração ou a criação de novas realidades”.

A pesquisa empregada nesta dissertação tem o propósito de identificar ações que as empresas desprenderam para integrar o modelo CMMI ao desenvolvimento ágil; sistematizar a realidade empírica de empresas que adotaram essas duas abordagens e responder a questionamentos de como essas empresas conseguiram integrar atividades conflitantes presentes na abordagem tradicional e ágil.

Segundo Santos (2002), os critérios utilizados para escolha do tipo de pesquisa podem ser sintetizados na Figura 21.

Figura 21 – Critérios para tipo de pesquisa



Fonte: Modelo de Santos (2002)

Ao adicionar a explicação de Gil (1999) quanto à forma de classificar pesquisas, esta dissertação tem as seguintes características:

Finalidade: Aplicada, por entender que a aplicação prática dirigida a problemas específicos envolve a verdade de um determinado contexto;

Objetivos: Pesquisa exploratória, por procurar conhecer os fatos e fenômenos relacionados ao tema de modo a proporcionar maior familiaridade com o problema

tendo o intuito de torná-lo explícito. Envolve levantamento bibliográfico; entrevista com pessoas que puderam vivenciar o problema e podem compartilhar as experiências.

Procedimentos: Bibliográfico, com o objetivo de recolher e analisar as principais contribuições de publicações de artigos periódicos (descritos na Seção 3, *Análise Bibliométrica*, e 4, *Pesquisa Bibliográfica*); estudos de casos, com objetivo de explorar situações reais em sua profundidade e descrever seu contexto;

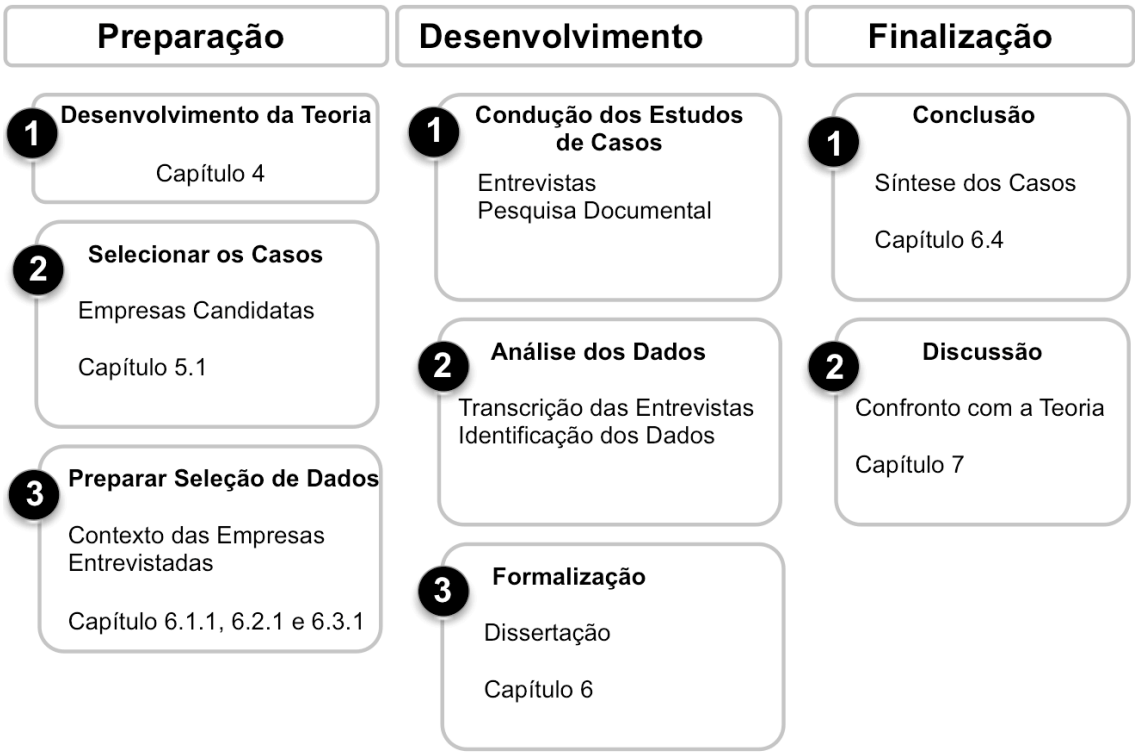
Natureza: Qualitativo, ao considerar que a dissertação conta com estudos de casos para interpretação de fenômenos; a análise dos dados é efetuada de forma indutiva;

Local de realização: Campo, a dissertação conta com coleta de dados e observação do fato efetuado no local do acontecimento do fenômeno, sendo executada na forma de estudos de casos.

A estratégia de pesquisa adotada foi o estudo de caso que, segundo Yin (2010), é um plano qualitativo utilizado para se conhecer o “como?” e o “porquê?“ quando o pesquisador detém escasso controle dos acontecimentos reais ou mesmo quando este é inexistente e o campo de investigação se concentra num fenômeno natural dentro de um contexto de vida real. Como apontado por Yin (2010), deve-se ter rigor para evidenciar a validade e confiabilidade do estudo de caso, quanto ao plano de investigação. Foi elaborado um plano de etapas para condução da pesquisa, conforme Figura 22 constituído por três etapas:

- Preparação: constitui a fundamentação da pesquisa, composta principalmente pelo Capítulo 4 (*Pesquisa Bibliográfica*) concebida a partir da análise descrita no Capítulo 3 (*Análise Bibliométrica*). Esta é a base para definição das questões de pesquisa e característica das empresas candidatas ao estudo de caso;
- Desenvolvimento: o objetivo é definir a melhor forma de conduzir os estudos de casos com intuito de realizar entrevistas e pesquisa documental que possa promover a análise dos dados. O resultado desta etapa está registado no Capítulo 6.
- Finalização: o objetivo é refletir sobre os casos estudados de forma a expor a síntese e concluir as questões de pesquisa promovendo a discussão e confronto com a teoria.

Figura 22 – Etapas da pesquisa



Fonte: Produzida pela autora

5.1. PREPARAÇÃO

Dado que o objetivo da pesquisa é identificar como as empresas que decidiram adotar o CMMI e o desenvolvimento ágil agiram para solucionar a adaptação das atividades que possuem tratamento divergente pelas abordagens tradicional e ágil, a primeira etapa foi definir o perfil das empresas candidatas ao estudo de caso, sendo:

- Devem ser empresas cuja atividade fim seja de tecnologia da informação (TI) no intuito de produzir *software*s sob o modelo fábrica de *software*:

> "Como o nome já diz, a Fábrica de *Software*, para ser considerada desta forma, deve possuir alguns atributos oriundos de uma Fábrica Industrial. O objetivo da Fábrica de *Software* deve ser a geração de produtos requeridos pelos usuários ou clientes, com o mínimo de defeitos possível e a um preço (ou custo) competitivo e compatível que forneça a margem necessária para os investimentos em manutenção e melhoria da Fábrica." (FERNANDES; TEIXEIRA, 2004, p. 116)

- As empresas candidatas devem possuir a certificação CMMI e adotar as práticas ágeis no ambiente de desenvolvimento de *software*.

A próxima etapa foi contatar as empresas candidatas de forma a explicar a proposta do estudo, convidá-las a participar da pesquisa e a dedicar-se a entrevistas formais com possibilidade de pesquisa documental e oportunidade para observação direta. O resultado foi a resposta positiva de três empresas. A última etapa da fase de preparação foi averiguar o contexto das empresas candidatas com intuito de buscar os contatos e agendar reuniões com os responsáveis.

5.2. DESENVOLVIMENTO

Para esta etapa foram levados em consideração os critérios lógicos propostos por Yin (2010) para qualidade da pesquisa empírica:

Figura 23 – Critérios de Qualidade dos projetos de pesquisa

Desenvolvimento		
Coleta de Dados	**Validade dos Constructos**	**Confiabilidade**
Análise dos Dados	**Validade Interna**	**Validade Externa**

Fonte: Yin (2010)

Os quatro critérios são:

- Validade dos Constructos: estabelecer medidas operacionais corretas para os conceitos que estão sob estudo. Com base no quadro de Integração CMMI *versus* Desenvolvimento Ágil (Seção 4.4) foi construído o questionário de entrevista, ao qual todas as empresas estudadas foram submetidas, conforme item 5.2.1. Com intuito de averiguar as hipóteses da pesquisa, todas as empresas forneceram um breve histórico e explanação de seus processos.
- Confiabilidade: demonstrar que as operações de um estudo, como os procedimentos de coleta de dados, podem ser repetidas, apresentando os mesmos resultados. Para atender este requisito, todos os questionários foram respondidos pelos responsáveis (perante o departamento de recursos

humanos da empresa) da fábrica de *software*, ora denominados gerentes ou diretores da fábrica, respeitando a nomenclatura individual de cada empresa;

- Validade Interna: estabelecer uma estratégia de investigação de triangulação. Para atender este ponto, foi acordado com os respondentes dos questionários que as áreas mencionadas no processo também seriam entrevistadas com intuito de enriquecer a seção descrita como *Processo* (Capítulo 6). Na prática, foram entrevistados os responsáveis pela fábrica de *software*, escritório de projetos e área de processos;
- Validade Externa: estabelecer o domínio no qual as descobertas de um estudo podem ser generalizadas. Este item foi contemplado pelo fato de as empresas possuírem certificação CMMI e por adotarem práticas ágeis no âmbito de desenvolvimento de *software*.

5.2.1. Desenvolvimento do Questionário

Tendo em vista a necessidade de desenvolver um roteiro de entrevista que pudesse ser submetido às empresas estudadas, foi elaborado um questionário de pesquisa com intuito de orientar a entrevista. Para elaboração das perguntas foi utilizado o quadro de Integração CMMI *versus* Desenvolvimento Ágil (Seção 4.4), conforme abaixo:

1. Gerenciamento de Projeto
   1.1. Como é efetuada a gestão do projeto quanto a planejamento, controle e monitoração?
      - O intuito é entender como a empresa faz a gestão de projetos, atendendo os itens “Planejamento do Projeto (PP)”, “Controle e Monitoração do Projeto (PMC)”, “Gestão de Projeto Integrado (IPM)”, classificados como “Suportados”, não devendo demandar, assim, um esforço de adaptação.
   1.2. De que forma é efetuada a gestão de contrato de fornecedores?
      - O objetivo é atender o item “Gestão de Contrato do Fornecedor (SAM)” classificado como “Não Suportado” no quadro de integração.
   1.3. De que forma é aplicada a gestão de riscos?
      - O objetivo é atender o item “Gestão de Risco (RSKM)” classificado como “Parcialmente Suportado” no quadro de integração.

1.4. De que forma é executada a gestão quantitativa do projeto? Há métricas de desempenho comparáveis entre projetos?

- O objetivo é atender o item “Gestão Quantitativa de Projeto (QPM)” classificado como “Não Suportado” no quadro de integração.

2. Gerenciamento de Processo

2.1. Há atividades para identificar, avaliar, implementar e monitorar pontos de melhoria no processo organizacional?

- O objetivo é atender os itens “Foco no Processo Organizacional (OPF)” e “Definição do Processo Organizacional (OPD)” classificados, respectivamente, como “Não Suportado” e “Parcialmente Suportado” no quadro de integração.

2.2. Há formas de avaliar o desempenho do processo organizacional?

- O objetivo é atender o item “Desempenho do Processo Organizacional (OPP)” classificado como “Parcialmente Suportado” no quadro de integração.

2.3. De que forma é identificado e estabelecido um plano de treinamento organizacional?

- O objetivo é atender o item “Treinamento Organizacional (OT)” classificado como “Parcialmente Suportado” no quadro de integração.

2.4. Há formas de apurar o desempenho organizacional, quanto ao negócio?

- O objetivo é atender o item “Gestão do Desempenho Organizacional (OPM)” classificado como “Parcialmente Suportado” no quadro de integração.

3. Engenharia

As perguntas abaixo são para entender como a empresa trata os processos referentes à engenharia, um ponto que não deve demandar esforço de adaptação.

3.1. Como é feito o gerenciamento de requisitos?

- O intuito é atender os itens “Gestão de Requisitos (REQM)” e “Desenvolvimento de Requisitos (RD)” classificados como “Suportado”.

3.2. Como é estabelecida a solução técnica?

- O intuito é atender os itens “Solução Técnica (TS)” e “Integração de Produto (PI)” classificados como “Suportado”.

3.3. Existem critérios para verificação e validação do produto?

- O intuito é atender os itens “Verificação (VER)” e “Validação (VAL)” classificados como “Suportado”.

4. Suporte

4.1. Como é efetuada a gestão de configuração?

- O intuito é atender o item “Gestão de Configuração (CM)” classificado como “Suportado”.

4.2. De que forma é efetuada a garantia da qualidade do produto e processo?

- O intuito é atender o item “Garantia da Qualidade do Produto e Processo (PPQA)” classificado como “Parcialmente Suportado”.

4.3. Possui atividades de análise e métricas dos resultados dos projetos?

- O intuito é atender o item “Análise e Métricas (MA)” classificado como “Parcialmente Suportado”.

4.4. De que forma é efetuada a análise de decisão e resolução?

- O intuito é atender o item “Resolução e Análise de Decisão (DAR)” classificado como “Parcialmente Suportado”.

4.5. Há atividades de análise causal?

- O intuito é atender o item “Resolução e Análise Causal (CAR)” classificado como “Parcialmente Suportado”.

5.3. FINALIZAÇÃO

A etapa final dos estudos de casos procura sintetizar todos os aspectos apresentados durante a condução da pesquisa, de forma a destacar pontos em comum ou mesmo distintos presentes nos cenários das empresas estudadas. O Capítulo 7 desta dissertação se constitui de forma a concluir as questões de pesquisa, confrontar a teoria e indicar proposta para trabalhos futuros.

6. ESTUDOS DE CASOS

Stake (2005) coloca que a investigação num estudo de caso não é baseada em amostragem. Não se estuda um caso para compreender outros casos, mas para compreender “o caso”. Este capítulo foi estruturado de forma a descrever os estudos de casos, organizados nos seguintes tópicos:

- *Sobre a Empresa*: caracteriza a empresa submetida ao estudo, de maneira a apresentar um breve histórico de seu contexto, valores e missão;
- *O Processo da Empresa*: apresenta a forma de organização de trabalho da empresa no ambiente de desenvolvimento de *software*;
- *Respostas ao Questionários*: corresponde à interpretação das respostas do questionário submetido aos respondentes;
- *Considerações do Caso*: expõe a percepção do pesquisador quanto ao ambiente estudado e relata a opinião de funcionários da empresa pesquisada.

6.1. EMPRESA A

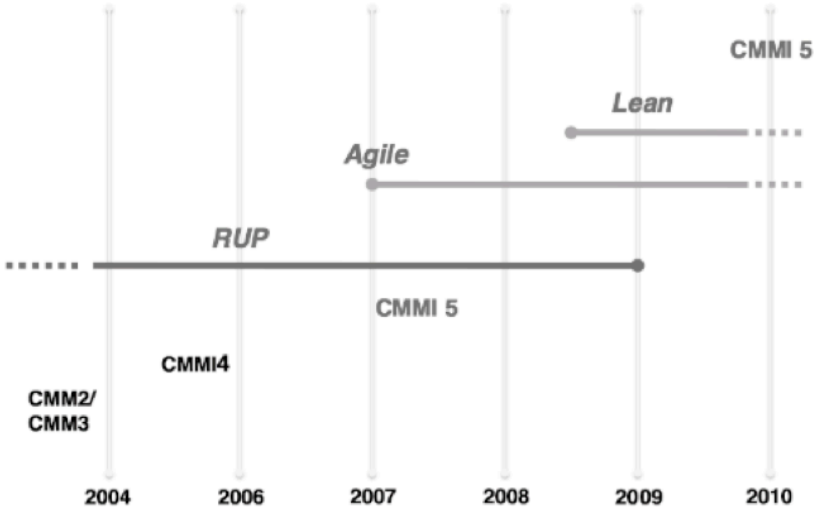
6.1.1. Sobre a Empresa

A empresa A é uma multinacional brasileira especializada em serviços de TI. Atualmente conta com mais de 1.500 colaboradores, possui cerca de 35% de sua receita proveniente de projetos internacionais e, de 2000 a 2009, cresceu aproximadamente 40%. Foi constituída em 1995 e, em 2007, após o investimento de 1 milhão de dólares, obteve o nível 5 do CMMI. O objetivo de obter esse titulo foi para ingressar no mercado internacional com exportação de serviços de *software*. Os recursos investidos foram destinados a treinamento, preparação e adequação de processos, ao longo de quatro anos.

No mesmo ano de 2007, após a obtenção do CMMI nível 5, a diretoria da empresa percebeu que a organização estava muito burocrática; grandes projetos mostravam-se morosos e desmotivavam a equipe. Então a organização decidiu implantar algumas células de desenvolvimento ágil. Montaram uma equipe dedicada a aplicar

a metodologia de desenvolvimento e implantaram o método *scrum* de trabalho. Essa metodologia progrediu para o pensamento *Lean*. Em 2010, a organização foi submetida à recertificação do CMMI, que ocorre a cada 3 anos, e a empresa garantiu o nível 5 de maturidade, agora, sob a ótica do desenvolvimento ágil. Na Figura 24 é possível visualizar a evolução da metodologia de desenvolvimento.

Figura 24 – Evolução da metodologia de desenvolvimento da Empresa A



Fonte: Empresa A

A empresa prega que seu principal objetivo é formar times de alta performance e, por isso, investe na habilidade de atrair e reter talentos, em construir ambientes motivadores com baixo índice de conflito, total transparência no relacionamento e com interações diárias. Promove entre os colaboradores a cultura *Lean*, na qual a mente é voltada a: solução de problemas; eliminação de desperdícios; priorização baseada em engenharia de valor e melhoria contínua. Incentiva a pró-atividade, colaboração e ciclos curtos de *feedback* de forma a produzir rápido aprendizado.

O faturamento da empresa em 2011 foi de 135 milhões de reais e a diretoria se orgulha da taxa de rotatividade (considerando quem saiu e quem entrou) ter ficado em torno de 5%, sendo que no setor é de 20% (conforme informações da empresa). Acreditam que a fidelidade dos funcionários se deve principalmente a uma filosofia de trabalho que reúne meritocracia e flexibilidade. A empresa possui uma política de construir um ambiente agradável, promovendo políticas para “o bolso” (como participação nos lucros e plano de carreira), iniciativas para “o corpo” (como planos de qualidade de vida e exames médicos pagos pela empresa) e para “a mente”

(como um programa feito para o funcionário desabafar sobre qualquer tipo de preocupação).

6.1.2. O Processo da Empresa

A fábrica de *software* é dividida em 7 segmentos, cada qual com um gerente responsável: telecomunicações, finanças, construção, indústria, comércio, governo e educação. O intuito dessa separação foi facilitar a comunicação do gerente da fábrica com o cliente. Esses 7 gerentes de fábrica respondem a um gerente geral da fábrica, responsável, principalmente, pelo controle orçamentário do departamento como um todo. Cada gerente de fábrica é responsável por gerenciar uma ou mais células de projeto, constituída basicamente por 1 *ScrumMaster* (ou líder técnico), 1 analista sênior, 1 analista pleno e 3 analistas júnior, sendo que o número dessa formação pode aumentar de acordo com o tamanho do projeto. Toda célula de projeto conta com o suporte da área de arquitetura e escritório de projetos.

A área de arquitetura é constituída por um gerente de arquitetura, responsável pela administração do departamento e por vários arquitetos. Cada arquiteto atende basicamente 2 projetos ao mesmo tempo.

A área de escritório de projetos é constituída por um gerente sênior, responsável pela administração do departamento e por duas hierarquias: gerente de projetos e gerente de negócios. A hierarquia de gerente de projetos é constituída por 1 gerente e vários analistas de projetos e cada um atende basicamente 2 projetos ao mesmo tempo. A hierarquia de gerente de negócio é constituída por 1 gerente e vários analistas de negócios dedicados ao projeto.

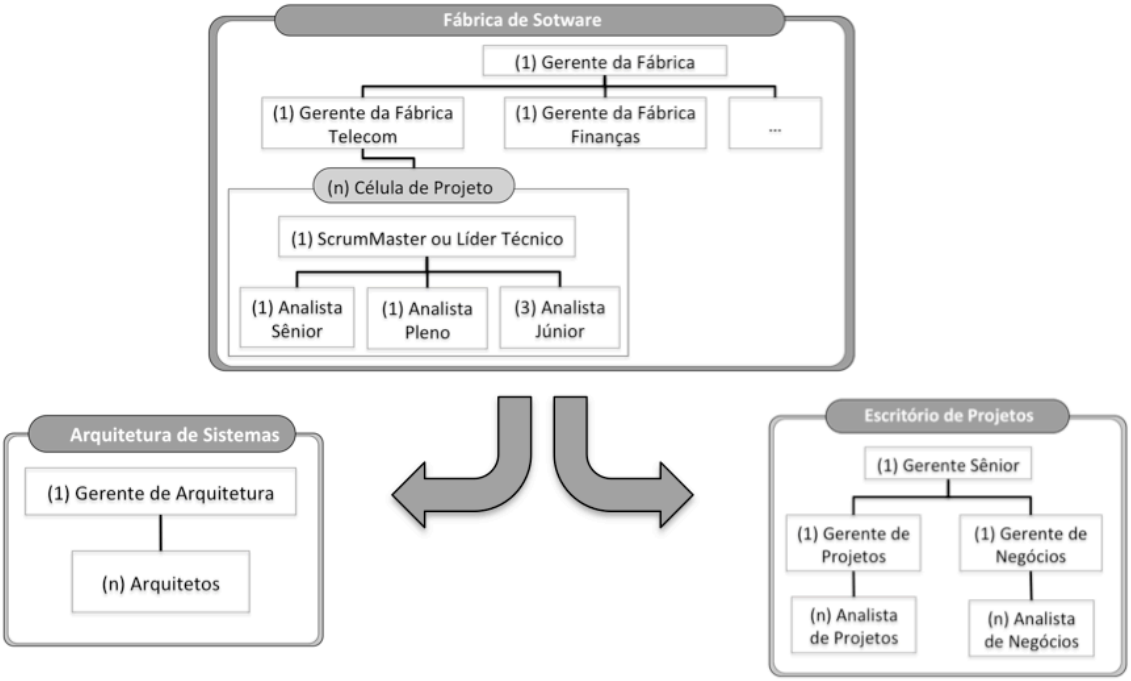
Cada célula de projeto tem, então, um arquiteto de sistemas, um analista de negócio e um analista de projeto:

- Arquiteto: responsável por direcionar a solução técnica;
- Analista de projeto: responsável pelo cronograma, cumprimento das etapas do projeto e níveis de qualidade;
- Analista de negócio: responsável pela captação do cliente; participa da especificação funcional;

- *Scrum master* (ou líder técnico): responsável pelo projeto; deve direcionar as atividades e decidir quais funcionalidades serão entregues (constituindo as etapas do projeto).

Uma visão dessa forma de trabalho pode ser vista na Figura 25.

Figura 25 – Divisão de tarefas da Empresa A



Fonte: Produzida pela autora

O gerente e os analistas de negócio são responsáveis por captar novos clientes e oportunidades enquanto o gerente e os analistas de projetos são responsáveis por captar novas demandas no cliente, de forma a manter um fluxo de trabalho contínuo. Quando uma nova demanda ou oportunidade é capturada, o gerente da fábrica deste segmento é acionado e indica um *scrum master*. Este, juntamente com a área de projetos, elabora uma estimativa que inclui até o treinamento do cliente na metodologia de desenvolvimento da Empresa A. O cliente é solicitado, formalmente, a indicar o usuário da organização que será responsável pelo projeto, sendo o ponto focal com a equipe de desenvolvimento.

Ao elaborar a estimativa, a equipe da Empresa A faz um documento de intenções que relaciona todos os recursos necessários ao projeto (técnicos, gestão, cursos, testes entre outros). Esse documento é feito na forma de *checklist* e o cliente, juntamente com o analista de projetos, seleciona os itens que deseja e, no final, é gerado um valor do projeto (custo) associado à qualidade esperada. A qualidade do

projeto é decidida em conjunto, de acordo com as intenções de recurso selecionadas. Essa foi uma forma encontrada pela Empresa A para, desde o início do projeto, deixar clara a participação do cliente no sucesso do mesmo. As atividades assumidas pelo cliente farão parte do projeto e a Empresa A se compromete a cobrar os resultados e não a controlar as atividades.

O *layout* físico da Empresa A obedece a segregação de departamentos, porém não há divisórias. No departamento de fábrica de *software,* os membros da célula de projeto (*scrum master*, analista sênior, pleno e júnior) sentam-se próximos, isto quer dizer que estas pessoas estão habituadas a mudar de lugar no início de cada projeto, de forma a promover maior integração e facilitar a comunicação. Por toda área física há monitores que mostram o andamento de todos os projetos.

A fábrica de *software* trabalha sob a abordagem ágil, o projeto é dividido em ciclos de 4 semanas (cada ciclo é denominado *Sprint*, termo usado no método *Scrum* de trabalho) e no término de cada ciclo deve haver um entregável ao cliente.

Para gerir o ciclo de vida do projeto foi desenvolvida uma ferramenta de gestão que associa alguns sistemas de terceiros de forma a integrar vários departamentos da Empresa A no processo de desenvolvimento de *software*.

6.1.3. Respostas do Questionário

O questionário foi respondido pelo gerente de fábrica do segmento de finanças.

1. Gerenciamento de Projeto
   1.1. Como é efetuada a gestão do projeto quanto a planejamento, controle e monitoração?

A empresa utiliza a metodologia *Scrum* baseada no conceito de *sprint*, um esforço concentrado no desenvolvimento em ciclos de quatro semanas. Dessa maneira, os clientes recebem resultados mais rapidamente e isso permite flexibilidade de acordo com suas necessidades. A empresa possui uma ferramenta de gestão que permite registrar o projeto em *sprints*. Para cada *sprint* é registrado um conjunto de funcionalidades que devem ser entregues e permite atribuir o profissional que será responsável pelo seu desenvolvimento. Com esta ferramenta é possível extrair informações de cobertura de código (corresponde ao percentual de código acessado

por testes automatizados) fornecendo uma base para dimensionar o progresso de projeto, permitindo seu controle e monitoração.

1.2. De que forma é efetuada a gestão de contrato de fornecedores?

No processo de desenvolvimento de *software* não há contratação de terceiros, ou seja, todo o trabalho é desenvolvido internamente. Há contratação de treinamentos, recursos de infraestrutura, etc.

1.3. De que forma é aplicada a gestão de riscos?

No início de cada *sprint,* o líder de projeto relaciona os possíveis riscos do projeto e no final do *sprint* deve informar se os riscos foram mitigados ou se devem ser relacionados para o próximo *sprint*. Esses dados são obtidos pela ferramenta de gestão.

1.4. De que forma é executada a gestão quantitativa do projeto? Há métricas de desempenho comparáveis entre projetos.

Todos os projetos são submetidos à mesma metodologia do *Scrum* e, com uso da ferramenta de gestão, é possível extrair dados comparativos entre os projetos. Para projetos em andamento é possível saber quantos *sprints* são esperados e, até o momento, quanto foi consumido de recurso. Para *sprints* futuros há pouca informação (como os riscos não mitigados até o *sprint* atual, e quanto de recurso ainda falta para conclusão do projeto), mas, assim que encerrado o projeto, cada *sprint* fornece os seguintes dados: quantos riscos foram identificados e mitigados; qual a taxa de falha (erros encontrados no cliente) e qualidade (testes internos); quanto recurso a mais foi necessário para manter a data de entrega do *sprint*; qual o nível de satisfação da equipe; qual a taxa de *turnover* (rotatividade da equipe).

2. Gerenciamento de Processo

2.1. Há atividades para identificar, avaliar, implementar e monitorar pontos de melhoria no processo organizacional?

Na ferramenta de gestão, assim que o projeto é iniciado, a equipe alocada deve informar quais são suas expectativas para o projeto e, neste momento, são questionados alguns pontos da organização, como possíveis pontos de melhoria ou atenção. Essas respostas ficam sempre disponíveis ao líder do projeto, RH e gestores da empresa. Para as perguntas organizacionais, o RH é encarregado de

direcionar cada ponto levantado e, até no final do projeto, deve registrar qual foi sua ação na questão. No final do projeto, a ferramenta apresenta a cada membro qual o *feedback* para cada ponto registrado no início do projeto.

2.2. Há formas de avaliar o desempenho do processo organizacional?

Pela ferramenta de gestão é possível identificar se há questões sem respostas. As questões são direcionadas ao projeto, à empresa, ao plano de carreira e expectativa pessoal de cada membro do projeto.

2.3. De que forma é identificado e estabelecido um plano de treinamento organizacional?

Pela ferramenta de gestão, o RH recebe todas as perguntas organizacionais coletadas por projetos e tem a responsabilidade de analisar e encaminhar um direcionamento que muitas vezes resulta em treinamento interno, palestras ou autorização para cursos externos, realizados ao longo de um ano.

2.4. Há formas de apurar o desempenho organizacional, quanto ao negócio?

Sim, medimos pela quantidade de clientes novos que conseguimos no ano, taxa de evasão de clientes e lucratividade dos projetos.

3. Engenharia

3.1. Como é feito o gerenciamento de requisitos?

Com uso da ferramenta de gestão, cada requisito é informado por *sprint* e todos devem ser atendidos no prazo estabelecido. Dessa forma, os líderes de projeto são orientados quanto à data fixa de entrega da cada *sprint*, podendo, quando atrasado, adicionar recursos para cumprimento do prazo. Mas quando isso ocorre, a métrica de custo do projeto tem um desvio que tende a distanciar o projeto do ponto de excelência.

3.2. Como é estabelecida a solução técnica?

No início do projeto a equipe discute a solução técnica e o arquiteto da solução é responsável por assegurar que esta seja cumprida ao longo do projeto.

3.3. Existem critérios para verificação e validação do produto?

Para cada *sprint* é estabelecido um conjunto de características que devem ser atendidas e, assim que entregue, o cliente deve informar se corresponde às

expectativas funcionais e não funcionais. Essas informações são registradas na ferramenta de gestão pelo líder do projeto de forma a compor as métricas de satisfação do cliente quanto ao produto. Outro ponto levado em consideração é o retorno da equipe de qualidade quanto à detecção de falhas.

4. Suporte

4.1. Como é efetuada a gestão de configuração?

Através da ferramenta de gestão, cada *sprint* é estabelecido e recebe um conjunto de requisitos. Para cada requisito é aberto um *ticket* que possui um responsável pela atividade que deve registrar o início do trabalho, comentar as alterações efetuadas e, uma vez terminada, deve submeter o código à ferramenta para montagem de versão, a qual deverá ser submetida à avaliação da equipe de qualidade.

4.2. De que forma é efetuada a garantia da qualidade do produto e processo?

É assegurado que todos os projetos estão submetidos à mesma ferramenta de gestão e por isso todos têm o questionário inicial do projeto que inclui perguntas organizacionais e de processo. Por política interna é estabelecido que todas as perguntas de projeto devem ter um direcionamento e no final de cada *sprint* coletamos a satisfação do cliente quanto a entrega, taxa de falhas (detectada pelo cliente) e de qualidade. Com base nesses dados é estabelecida uma métrica interna com foco em 100% de respostas dos questionários de projetos, 75% de satisfação da equipe de projeto, 90% de satisfação do cliente e 92% de qualidade da entrega.

4.3. Possui atividades de análise e métricas dos resultados dos projetos?

Com uso da ferramenta de gestão, a que todos os projetos são submetidos, dados de cada *sprint* são coletados, como por exemplo: desvio de prazo e custo; satisfação interna (equipe do projeto); satisfação do cliente; qualidade da entrega (taxa de falhas). No encerramento do projeto é acrescida a informação de admissões e demissões. Essas informações são analisadas pelo escritório de projetos, responsável por gerar relatórios de acompanhamento.

4.4. De que forma é efetuada a análise de decisão e resolução?

Com uso da ferramenta de gestão, o escritório de projetos é responsável pelo acompanhamento e por sinalizar os projetos que não estão cumprindo as métricas estabelecidas internamente. Cada líder de projeto é informado quanto a possíveis

desvios do projeto e estes devem atuar para solucionar a questão, esperando-se uma melhora no próximo *sprint* do projeto. São registradas na ferramenta de gestão quantas intervenções do escritório de projeto foram necessárias, pois é responsabilidade do líder do projeto que todas as métricas internas sejam cumpridas.

4.5. Há atividades de análise causal?

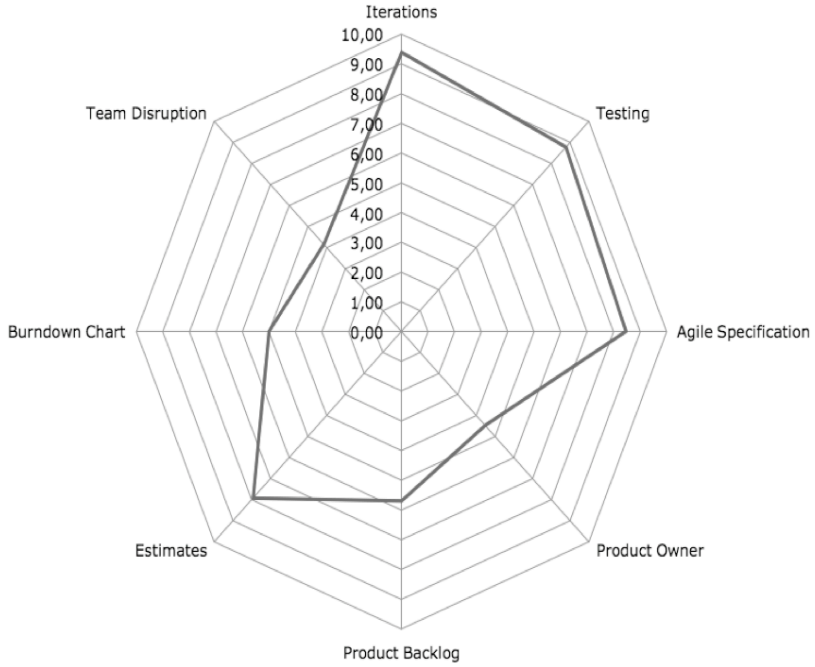
Sim. Para cada *sprint* é obtida a taxa de qualidade (classes de testes executadas com sucesso) que deve ser no mínimo de 92%. Se o *sprint* não alcançou essa qualidade, deve ser corrigido e submetido a uma nova avaliação. A partir da terceira reavaliação de qualidade, o líder do projeto é questionado a informar qual o motivo da baixa qualidade. Este é registrado na ferramenta de gestão, o escritório de projetos é informado e espera-se uma análise do problema. No final de cada projeto é apresentado um conjunto de problemas, obtido através da ferramenta de gestão, que inclui: baixa taxa de qualidade; desvios de prazo e custo; baixa satisfação interna e de cliente. O líder do projeto deve registrar como cada problema foi endereçado e solucionado e, na sequência, o escritório de projetos também informa se o problema foi resolvido internamente no projeto ou se foi necessária a intervenção ou mesmo o ajuste organizacional ou de processo.

6.1.4. Considerações do Caso

A Empresa A expressa em números a satisfação em trabalhar com desenvolvimento ágil associado ao CMMI. Em 2010, ano da recertificação CMMI nível 5, a empresa tinha 80% das células de projeto sob abordagem ágil//*lean* e outros 20% na abordagem tradicional.

Um ponto que a Empresa A considera primordial para ter sucesso na implantação dos princípios ágeis no ambiente de desenvolvimento de *software* é a constante monitoração da empresa para averiguar se a organização está aderente aos princípios ágeis. A Figura 26 expressa a monitoração que a empresa fez em seus projetos, submetidos à abordagem ágil, no início de 2008 com intuito de averiguar o quanto estava aderente a metodologia *Scrum*.

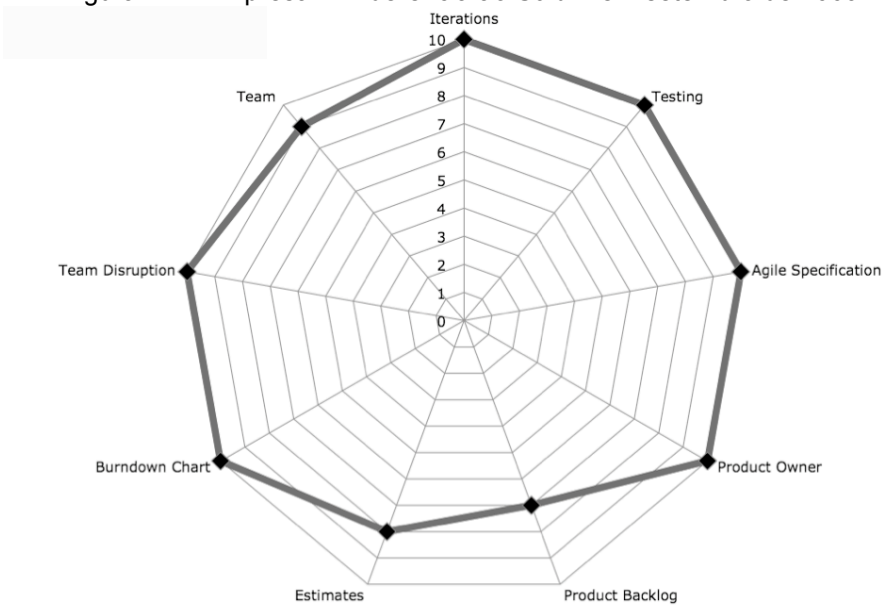
Figura 26 – Empresa A: Aderência ao *Scrum* no início de 2008



Fonte: Empresa A

De posse desses números, a empresa constatou que havia pontos a serem melhorados; culminando com a implantação dos princípios *lean*, em 2009, trabalharam para permear pela organização os princípios ágeis/*lean* e, em setembro de 2009, essa aderência expressava números melhores, conforme Figura 27.

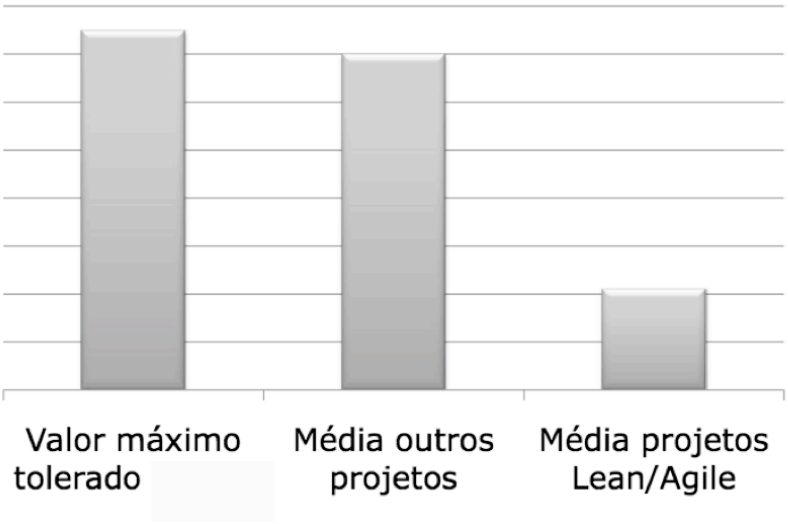
Figura 27 – Empresa A: Aderência ao *Scrum* em setembro de 2009



Fonte: Empresa A

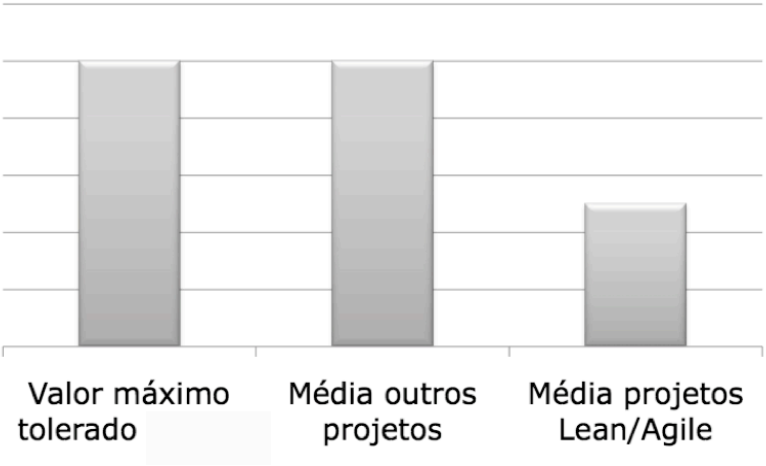
Quanto à produtividade de equipe de desenvolvimento, a empresa forneceu os dados da Figura 28 e Figura 29 que expressam, respectivamente, a quantidade de defeitos em homologação e produção por Kloc (*Kilo Lines of Code*, métrica em sistemas para medir o tamanho do programa). É possível notar que há o valor máximo tolerado pela empresa e os projetos submetidos à abordagem ágil têm números melhores que os projetos submetidos à abordagem tradicional.

Figura 28 – Empresa A: Defeitos de Homologação / *Kloc*



Fonte: Empresa A

Figura 29 – Empresa A: Defeitos de Produção / Kloc



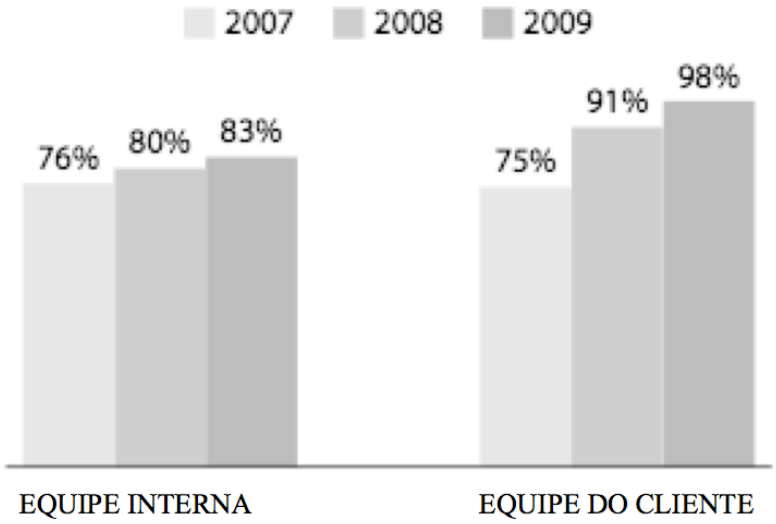
Fonte: Empresa A

Um ponto colocado como de extrema importância pela empresa foi quanto à melhora na satisfação da equipe, a empresa busca um nível de 75 % de satisfação e, de acordo com as métricas de 2009, os projetos submetidos ao tradicional

resultou em 76,46% e, dos projetos na abordagem ágil, o nível foi de 83,08%, conforme Figura 30. Para a empresa, equipe satisfeita contribui para se obter uma equipe motivada que tende a obter um produto de melhor qualidade, resultando em cliente mais satisfeito. Nas palavras do gerente geral da fábrica:

> "Temos sempre em mente dois pontos primordiais para o nosso sucesso: a velocidade e a complexidade. Uma força representa a velocidade com que as mudanças vêm aumentado ao longo do tempo e isto exige velocidade de adaptação e o desenvolvimento ágil nos permite rápidas respostas às mudanças. A outra força representa a complexidade que vem aumentando ao longo do tempo, exigindo processos maduros e extrema disciplina. O CMMI 5 nos garante a maturidade e a disciplina necessárias".

Figura 30 – Empresa A: Nível de satisfação



Fonte: Empresa A

6.2. EMPRESA B

6.2.1. Sobre a Empresa

É uma multinacional brasileira prestadora de serviços, soluções para informática e consultoria. Fundada em 1987, adquiriu o CMMI nível 3 em 2004 e logo após, em 2005, conquistou o nível 5, após um investimento de 3 milhões de reais. Atualmente conta com mais de 20 fábricas de *software* localizadas no Brasil e exterior, totalizando aproximadamente 7.800 colaboradores. A empresa se orgulha de assegurar que a qualidade dos serviços prestados por ela é garantida pelas melhores práticas e pelo controle de qualidade. Nas palavras do diretor:

> "Estamos continuamente investindo em qualidade, nossos clientes buscam isso e com qualidade criamos relacionamentos duradouros. O *framework* das fábricas de *software* é focado nas melhores práticas de mercado e pode ser facilmente adaptado às necessidades de nossos clientes. Como reconhecimento, diversos clientes nos pedem para apoiá-los a melhorar suas áreas, tanto de gestão quanto de desenvolvimento. Assim, entregamos mais valor, mais qualidade e mais confiança no relacionamento".

No fim de 2007, a empresa decidiu aumentar sua exposição no exterior e adquiriu várias empresas ao longo dos anos. Atualmente 40% de suas operações são no exterior e o objetivo é alcançar 50% ou um pouco mais. A empresa coloca que a internacionalização abre os horizontes e dá acesso às melhores práticas do mundo, abre mercados e a perspectiva de crescimento no exterior é maior do que no Brasil. Como alicerce para este crescimento internacional destaca a importância das certificações de qualidade, por aumentar a capacidade de planejamento e de organização.

A empresa possui uma política de inovação constante, seja em gestão, técnica, de processo ou de produto. Uma célula de inovação em São Paulo decidiu implantar o modelo de desenvolvimento ágil nas áreas de manutenção e sustentação de sistemas. A estratégia de iniciar nesta área é porque não há projetos grandes, pois por definição seria considerado um novo desenvolvimento e não manutenção (correção) ou sustentação (melhorias) de sistemas existentes. No final de 2010, ocorreu o primeiro *workshop* para explicar à equipe o método *scrum* e desde então foram sucessivas palestras para disseminar o conceito. Mas, como o próprio diretor coloca: "*Mais difícil que mudar as práticas é mudar a cultura para agile*".

O desenvolvimento ágil disseminou-se pela empresa a ponto de, em meados de 2011, ter seu primeiro grande projeto inserido nessa abordagem, desde seu início, a pedido de um cliente internacional.

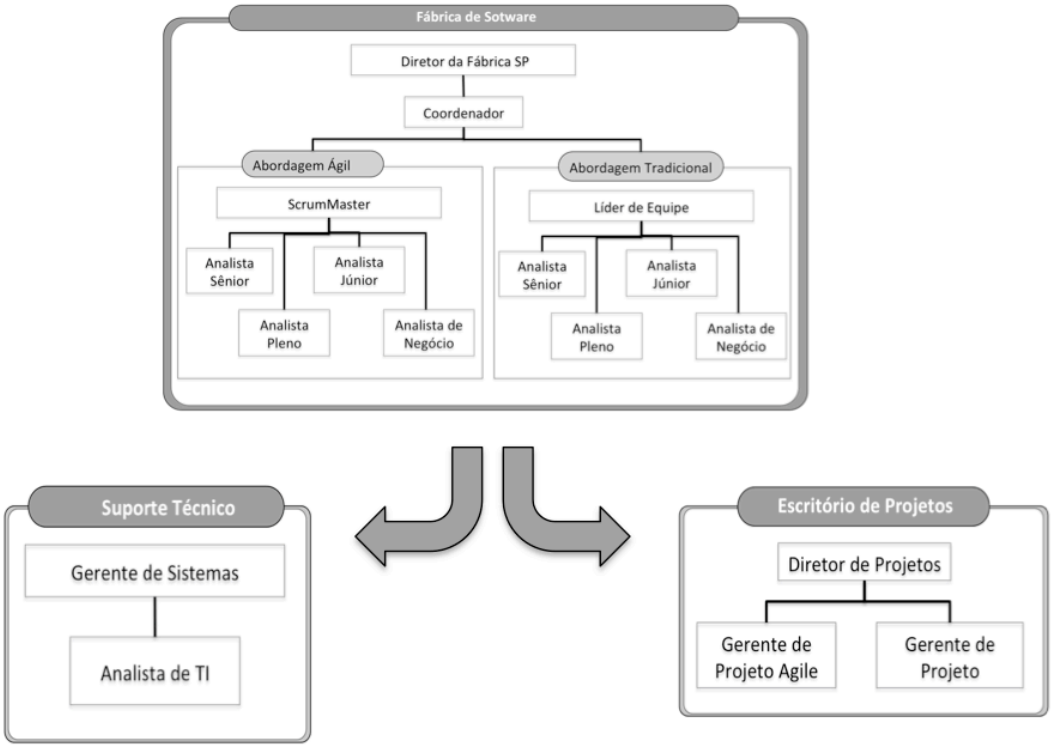
Como a empresa seguia a abordagem tradicional, para adaptar-se ao desenvolvimento ágil foi necessário contratar profissionais que já tinham trabalhado sob esses princípios. Foram criados departamentos ditos "*agiles*", compostos por profissionais que atuariam somente na abordagem ágil.

6.2.2. O Processo da Empresa

A fábrica de *software* de São Paulo foi dividida em projetos sob abordagem tradicional e ágil. Na estratégia para adotar o desenvolvimento ágil foi decidido utilizar o método *Scrum* associado a prática de *Kanban* do Desenvolvimento *Lean*.

A estrutura da fábrica de São Paulo é constituída por 1 diretor e vários coordenadores. Um coordenador pode trabalhar com projetos sob abordagem tradicional ou ágil, ainda não há coordenadores dedicados por abordagem. Na abordagem ágil há a figura do *ScrumMaster* e a equipe é formada por analista sênior, pleno, júnior e analista de negócio. O tamanho do projeto definirá a quantidade de profissionais na equipe. A equipe de desenvolvimento ágil terá o suporte do gerente de projeto *agile*, pois foi treinado no método, que responde hierarquicamente ao diretor de projetos constituindo o escritório de projetos. Todos os projetos da fábrica também contam com um suporte técnico, sob demanda, que auxiliam nas tarefas de desenho técnico, uso de novas tecnologias ou resolução de problemas técnicos. A estrutura organizacional pode ser visualizada na Figura 31.

Figura 31 – Empresa B: Estrutura organizacional



Fonte: Produzida pela autora

Na equipe sob abordagem ágil optou-se em manter o analista de negócio e este auxilia o *ScrumMaster* na especificação funcional e comunicação com o cliente. O gerente de projeto *agile,* que dá suporte à equipe, tem o ônus de manter um trabalho duplo, ao controlar e monitorar o projeto sob aspectos ágeis e reportar no modelo tradicional, pois o escritório de projetos mantém os relatórios no modelo tradicional.

Com intuito de não perder os controles e métricas do processo atual da fábrica, certificado no modelo CMMI nível 5, a equipe *agile* deve seguir a diretriz organizacional. Dessa forma, a equipe aplica a gestão e desenvolvimento ágil com o intuito de garantir a qualidade do produto e segue os processos organizacionais necessários para continuidade do modelo CMMI nível 5.

6.2.3. Respostas do Questionário

O questionário foi respondido pelo coordenador da fábrica de *software*.

1. Gerenciamento de Projeto
   1.1. Como é efetuada a gestão do projeto quanto a planejamento, controle e monitoração?

O processo organizacional da empresa B exige que todo projeto tenha um cronograma definido pelas fases de conceituação, especificação, desenvolvimento, homologação e acompanhamento da produção, devendo ter no mínimo uma tarefa do tipo *milestone* (marcos de projeto) por fase. Este cronograma resumido será acompanhado pela diretoria da empresa B como forma de monitorar o progresso dos projetos, o qual possui uma ferramenta de monitoramento e controle que permite comparar o *status* dos projetos, devendo o reporte ser mensal. No contexto do processo de gestão de projetos, há um escritório de projetos, com profissionais treinados na metodologia ágil, que auxilia os líderes dos projetos (*ScrumMaster*) a gerir o projeto. Cada projeto é dividido em n *sprints* (ciclos de desenvolvimento da metodologia *Scrum*) de 5 semanas. Todos os projetos submetidos ao desenvolvimento ágil estão registrados na ferramenta de gestão e controle de projetos, que sofreu alterações para permitir coleta de dados por *sprint* e não somente por fases (como é feito atualmente para projetos submetidos à abordagem tradicional).

1.2. De que forma é efetuada a gestão de contrato de fornecedores?

Na equipe de desenvolvimento de *software* não há possibilidade de contratar fornecedores para repassar o desenvolvimento de algumas funcionalidades. Se há necessidade de contratar fornecedor então será para o projeto todo e não por fases ou *sprints*. A contratação de fornecedores é mais presente em treinamentos e para serviços de infraestrutura.

1.3. De que forma é aplicada a gestão de riscos?

A ferramenta de gestão e controle de projetos exige que sejam elencados os riscos do projeto no momento de sua criação e, ao longo do tempo, gera relatórios mostrando com que frequência o gestor do projeto atualiza a matriz de risco. No encerramento do projeto a ferramenta exige que todos os riscos tenham comentários e definição da ação tomada.

1.4. De que forma é executada a gestão quantitativa do projeto? Há métricas de desempenho comparáveis entre projetos.

Todos os projetos estão submetidos à mesma ferramenta de gestão e controle de projetos e, no momento da criação, deve-se definir qual será a forma de condução, tradicional ou ágil. Uma vez estando sob abordagem ágil, a ferramenta passa a exigir informações por *sprint*, que são comparáveis entre os projetos.

2. Gerenciamento de Processo

2.1. Há atividades para identificar, avaliar, implementar e monitorar pontos de melhoria no processo organizacional?

Os projetos sob abordagem ágil são controlados pela ferramenta de gestão e controle de projetos, porém para os processos organizacionais o gestor deste projeto deve alimentar a ferramenta organizacional. Esta ferramenta tem a característica de coletar informações sob a ótica de aspectos gerenciais com intuito de melhorar o processo organizacional. Todo gestor de projeto deve alimentar esta ferramenta de acordo com as fases do projeto (conceituação, especificação, desenvolvimento, homologação e acompanhamento da produção). O questionário é direcionado para o departamento de processos da organização, que analisa quais processos devem ser melhorados ou mesmo alterados para facilitar a condução dos projetos por fases.

2.2. Há formas de avaliar o desempenho do processo organizacional?

Como a ferramenta organizacional exige informações pelas fases do projeto, o departamento de processos da organização solicita informações ao escritório de projetos para então relacioná-los, com intuito de averiguar se projetos que relatam maiores dificuldades organizacionais refletem em maiores atrasos ou custos, de acordo com a fase. Por exemplo, a fase de conceituação, que é muito dependente do cliente, tem um peso menor do que homologação, por ser uma área de serviço da própria organização.

2.3. De que forma é identificado e estabelecido um plano de treinamento organizacional?

O departamento de processos da organização que consolida as informações de aspectos organizacionais *versus* de projetos é responsável por identificar e estabelecer treinamentos, como por exemplo, de gestão de projetos, solicitados ao escritório de projetos, ou de homologação solicitados à área de qualidade.

2.4. Há formas de apurar o desempenho organizacional, quanto ao negócio?

Sim, a medição é efetuada considerando a quantidade em horas de projetos *versus* número de colaboradores, com intuito de averiguar o aumento da produtividade.

3. Engenharia

3.1. Como é feito o gerenciamento de requisitos?

O processo organizacional exige que os projetos sejam classificados nas fases de conceituação, especificação, desenvolvimento, homologação e acompanhamento da produção, e nos casos dos projetos submetidos à abordagem ágil a fase de especificação é muito rápida, pois representa a aceitação do projeto pela equipe técnica, e na fase de desenvolvimento quando os *sprints* forem estabelecidos é que de fato ocorre o levantamento de requisitos. Dessa forma, todo o gerenciamento de requisitos é estabelecido por *sprint*.

3.2. Como é estabelecida a solução técnica?

Na fase de especificação, correspondente à aceitação do projeto pela equipe, é estabelecida a solução técnica com auxílio do *pool* de suporte técnico de TI (vide Figura 31).

3.3. Existem critérios para verificação e validação do produto?

No final de cada *sprint* o produto a ser entregue é submetido a testes pela própria equipe de desenvolvimento e entregue ao cliente. A fase de homologação, formalmente exigida pelo processo da organização, é executada pela área de qualidade a cada quatro *sprints*, sendo que o último *sprint* é obrigatório.

4. Suporte

4.1. Como é efetuada a gestão de configuração?

A empresa B possui ferramenta de controle de versão de código que, por determinação da organização, deve ser utilizada diariamente, e o *ScrumMaster* é o responsável por gerar a versão no final do *sprint*.

4.2. De que forma é efetuada a garantia da qualidade do produto e processo?

O departamento de processos da organização é responsável por gerir, controlar e manter o processo organizacional enquanto os processos referentes ao projeto são de responsabilidade do escritório de projetos, o qual deve prezar pela qualidade do produto. A cada 4 *sprints*, que corresponde à homologação do *software* pela equipe de qualidade, o escritório de projetos coleta as informações dos testes internos e submete ao cliente um questionário de satisfação do produto.

4.3. Possui atividades de análise e métricas dos resultados dos projetos?

O escritório de projetos possui uma ferramenta de controle de projetos, alimentada por todos os gerentes de projetos, até mesmo pelos projetos submetidos à abordagem ágil. A análise é efetuada comparativamente entre os projetos submetidos à abordagem tradicional, ágil e tradicional x ágil.

4.4. De que forma é efetuada a análise de decisão e resolução?

O líder do projeto (*ScrumMaster*) é responsável por gerir o *sprint*, sendo responsável por garantir o seu progresso. Quando necessário, pode solicitar auxílio da equipe de suporte TI e gerente de projeto, para determinação de solução ou suporte na tomada de decisão.

4.5. Há atividades de análise causal?

Sim. Para análises no âmbito organizacional e seu impacto no processo dos projetos deve ser efetuada pelo departamento de processos. A análise individual do projeto e

entre projetos sob mesma abordagem deve ser efetuada pelo escritório de projetos. Essas atividades compreendem identificar, analisar e executar melhorias para garantia do processo e produto.

6.2.4. Considerações do Caso

A comparação entre projetos submetidos à abordagem ágil *versus* tradicional, efetuada pelo escritório de projetos, resulta em melhores resultados para projetos ágeis, como melhora significativa na satisfação do cliente quanto ao produto entregue, menor desvio de prazo e custo. Concluindo, em média, os projetos ágeis foram 17% superiores aos projetos tradicionais, nos critérios de sucesso de projeto quanto a prazo e custo. A satisfação do cliente foi 13,8% superior. Um importante ponto destacado foi o aumento da satisfação da equipe, que até então não era mensurado pela empresa.

Tanto o departamento de processos quanto o escritório de projetos afirmam que há ganhos no processo e na qualidade do produto quando o projeto é submetido à abordagem ágil. Mas, ao serem questionados por que a organização não assume a abordagem ágil, para toda a fábrica a resposta foi a dificuldade pela aprovação de orçamentos. Como na empresa há um alto número de intermediários e cargos, a política de aprovação de orçamentos emperra a adoção do *agile* na organização. Concluem que no momento há necessidade de quebra de paradigma da empresa, mostrando que há vantagens em investir tempo para economizar um valor que já esteja aprovado para gastar.

Outro ponto importante colocado pelos gestores da empresa B é quanto à necessidade de métricas e indicadores, uma herança muito útil do CMMI, mas que para o desenvolvimento ágil devem ser repensadas, no intuito de torná-las mais diretas e simples.

> "Métricas e indicadores são muito importantes, principalmente para empresas de serviços de *software* ou fábricas de *software*. Os clientes exigem esses números e a alta cúpula de executivos precisa disso para tomar suas decisões. Agora, não dá para falar de cultura ágil e continuar medindo as pessoas da mesma forma que fazemos na indústria convencional."

6.3. EMPRESA C

6.3.1. Sobre a Empresa

A empresa iniciou suas atividades em 1987 e atualmente possui 537 colaboradores, considera que sua marca seja o resultado constante de investimento em métodos e processos, bem como a preocupação com a capacitação de seus profissionais. Alinhada com os movimentos de globalização da economia, a empresa também explora oportunidades no exterior e concentra esforços na exportação de seus serviços de desenvolvimento *offshore*.

A empresa certificou-se, em 2005, no nível 2 de maturidade, envolvendo cerca de 100 profissionais e consumiu cerca de R$800 mil de investimento. Além dos resultados positivos obtidos com o uso dos novos processos na gerência dos projetos, a empresa espera ainda perceber outro grande benefício dessa conquista. O presidente da empresa afirma que:

> "Já exportamos para os Estados Unidos, mercado com um alto nível de exigência e, por isso, o CMMI vem apoiar bastante o desenvolvimento dessa linha de negócios. O CMMI é reconhecido no mercado americano como uma demonstração de competência naquilo que fazemos".

Em 2010, a empresa teve crescimento de cerca de 28% e teve aumento de 25% na carteira de clientes. Em 2011, a empresa aprovou um projeto de investimento de 8,1 milhões junto ao BNDES e, para o final de 2013, está prevista a certificação para nível 3 do CMMI.

> “Apesar dos riscos de enrijecer a empresa ao seguir um processo, a vantagem do CMMI é ter processos bem definidos, geração de documentação de projetos e garantia da qualidade de processo, e ao associarmos com o desenvolvimento ágil podemos garantir a qualidade do produto”.

Estas são as palavras do gerente de operações da empresa ao mencionar a vantagem da implantação do modelo de desenvolvimento ágil, ocorrida em 2011. Acrescenta sua opinião sobre a implantação da metodologia:

> “Não creio haver diferenças entre qualquer tipo de organização por mais ou menos dinâmica que ela seja. O problema é que a Metodologia é vista, geralmente, como um fardo que os analistas e gerentes têm que carregar. A

> empresa tem procurado chegar a um nível de excelência em que a Metodologia seja realmente uma ferramenta de Gestão de Projetos e, mais do que isso, uma ferramenta de autogestão das áreas de desenvolvimento de sistemas".

6.3.2. O Processo da Empresa

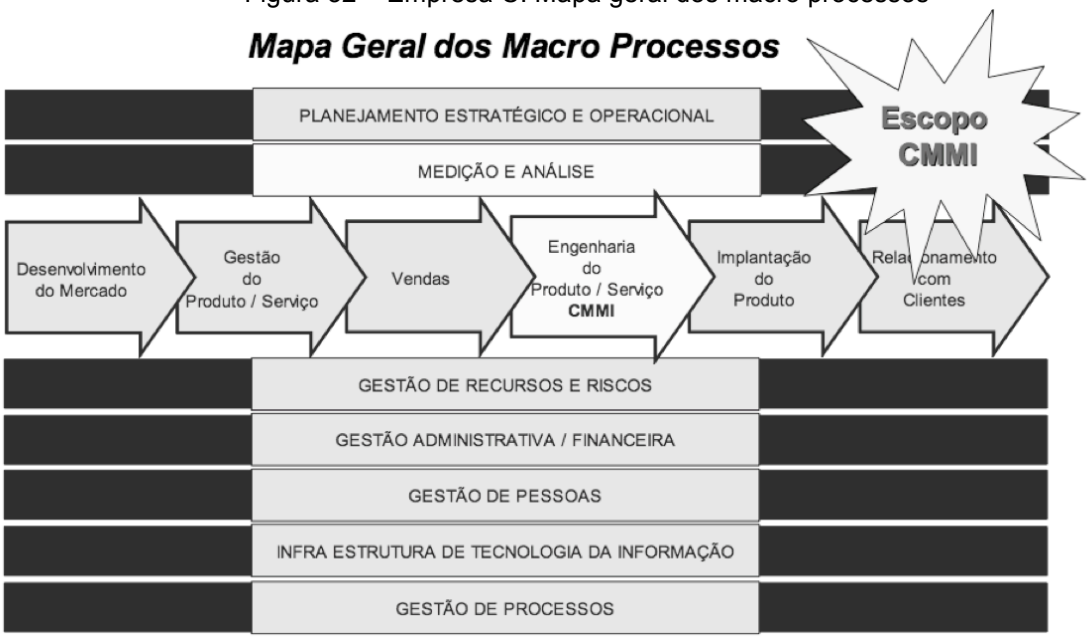
A empresa possui a fábrica de *software* em Campinas e define que seu processo é uma combinação de desenvolvimento e operações, significando uma nova abordagem para agilizar a geração de novos produtos e serviços. Pode ser visto como mais um campo para aplicação de técnicas ágeis.

A ideia básica é integrar e otimizar todas as atividades feita em conjunto pelos times de desenvolvimento e operações que trabalham como se fossem uma única equipe. Na abordagem tradicional, a função de desenvolvimento é transformar requisitos em programas e a função do time de operações é instalar e manter esses programas rodando na infraestrutura existente. Na abordagem da empresa, tudo é um único processo e todos devem trabalhar como equipe tendo um único objetivo final: entregar uma nova versão de forma mais rápida e eficiente possível.

A motivação para a integração dos times e para adoção das técnicas ágeis é a necessidade de reagir rapidamente às mudanças do negócio. Quanto mais mudanças, mais alterações são necessárias, mais produtos e serviços novos devem ser disponibilizados para os clientes.

Se cada nova implantação é um processo difícil e arriscado, a mudança é vista como um problema e não como necessidade ou oportunidade. Tendo em vista essa forma de trabalho, a empresa assumiu o macro processo definido na Figura 32.

Figura 32 – Empresa C: Mapa geral dos macro processos



Fonte: Empresa C

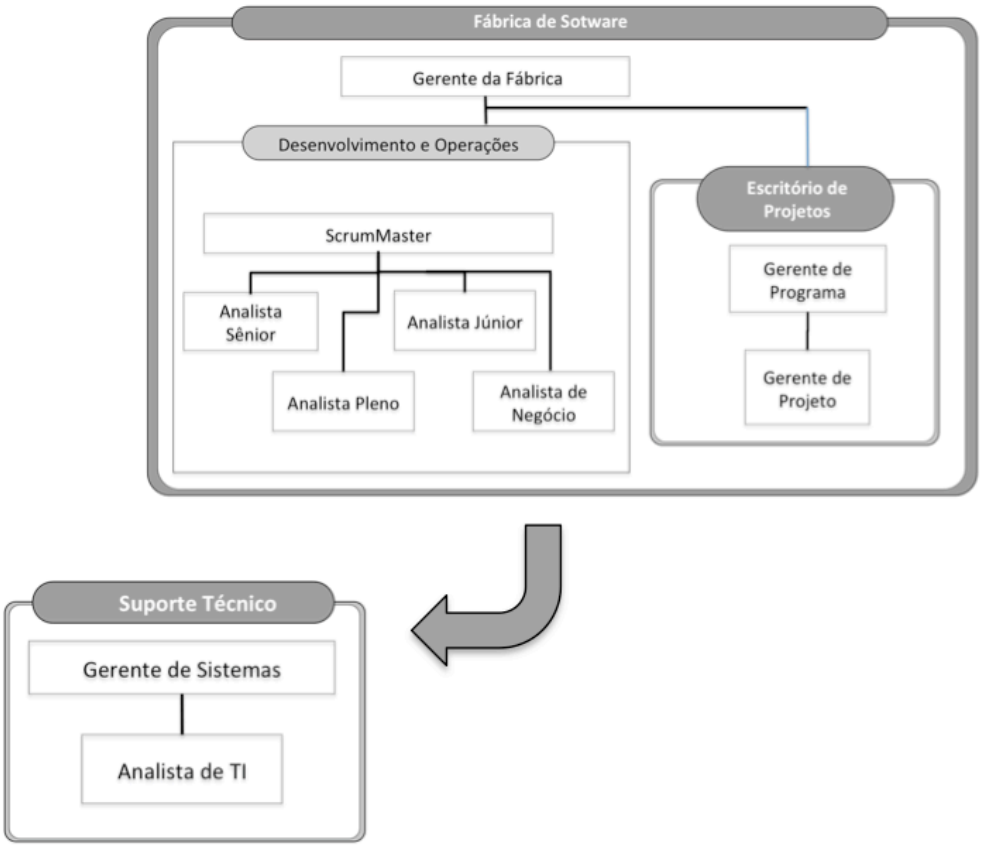
O desenvolvimento de *software* corresponde à área de Engenharia do Produto/Serviço CMMI responsável pelo processo de medição e análise. O escritório de projetos corresponde à área de Gestão do Produto/Serviço, sendo responsável pelo processo de gestão de recursos e riscos.

O gerente de projetos suporta a equipe de vendas definindo o macro escopo (fronteiras do projeto) e prévia do esforço, orçando a quantidade necessária de recursos, respeitando o perfil necessário para atender o projeto. Quando o projeto é assinado a equipe de desenvolvimento é autorizada a iniciar o trabalho. A forma de trabalho baseia-se no *Scrum*, na qual os ciclos de entregas são definidos.

A equipe do projeto é composta por desenvolvedores: analistas de testes e arquitetos, denominados como analista de TI, compõem a área de infraestrutura sendo consumidos na forma de serviço, por demanda. O líder do projeto, o desenvolvedor mais experiente da equipe, assume o papel de *ScrumMaste*r, e é responsável por alimentar a ferramenta de gestão com os dados de métricas e análises do progresso do projeto.

O escritório de projetos responde ao gerente de fábrica, responsável pelos projetos. Uma característica importante da estrutura de desenvolvimento é que não possuem a figura do coordenador, conforme mostra a Figura 33.

Figura 33 – Empresa C: Processo



Fonte: Produzida pela autora

6.3.3. Respostas do Questionário

O questionário foi respondido pelo gerente da fábrica

1. Gerenciamento de Projeto
   1.1. Como é efetuada a gestão do projeto quanto a planejamento, controle e monitoração?

A empresa C desenvolveu uma ferramenta própria de gestão de projetos, sendo que o objetivo e o cronograma são de responsabilidade do gerente de projetos, e o *ScrumMaster* do projeto é responsável por alimentar com métricas e análise de progresso.

   1.2. De que forma é efetuada a gestão de contrato de fornecedores?

A área de desenvolvimento e operações não comporta contratação de terceiros.

1.3. De que forma é aplicada a gestão de riscos?

O gerente de projetos é responsável por elencar e direcionar a mitigação dos riscos. O *ScrumMaster* deve acomodar as atividades elencadas de risco nos ciclos de desenvolvimento estabelecidos durante o projeto.

1.4. De que forma é executada a gestão quantitativa do projeto? Há métricas de desempenho comparáveis entre projetos?

A ferramenta de gestão de projetos foi implantada na empresa em 2006 e desde então tem sofrido alterações de métrica de projetos, com intuito de amadurecer o processo. Assim, ainda não há base constituída para comparar dados de projetos.

2. Gerenciamento de Processo

2.1. Há atividades para identificar, avaliar, implementar e monitorar pontos de melhoria no processo organizacional?

A empresa criou uma área de gestão de processos, com a responsabilidade de mapear os processos internos com intuito de melhorar sua eficiência. Esta área deve se preocupar primeiramente com os processos organizacionais para depois se preocupar com os processos de projetos.

2.2. Há formas de avaliar o desempenho do processo organizacional?

A área de gestão de processos tem criado métricas para medir o desempenho, tais como:

- Questionário de pesquisa de clima, com periodicidade semestral: o intuito é capturar o nível de satisfação da equipe diante dos recursos e benefícios da empresa.
- *Turnover* da equipe técnica: mede a rotatividade da equipe, considerando a quantidade de demissões e admissões. Há uma separação das saídas por pedido de demissão e demitidos. A periodicidade também é semestral.
- Taxa de sucesso dos projetos: esta é uma métrica em implementação, apesar de a ferramenta de gestão ter sido implantada há alguns anos, só recentemente as mesmas métricas estão sendo submetidas a todos os projetos da empresa. Para projetos iniciados há dois anos, é possível capturar os níveis de satisfação do cliente, desvio do cumprimento de prazo e custo.

2.3. De que forma é identificado e estabelecido um plano de treinamento organizacional?

A área de gestão de processos é responsável por direcionar os treinamentos organizacionais e a área de recursos humanos é responsável por direcionar os treinamentos individuais. Se houver necessidade de treinamento específico por um projeto então o escritório de projetos deve alocar este custo no orçamento do projeto.

2.4. Há formas de apurar o desempenho organizacional, quanto ao negócio?

Esta é uma métrica ainda em construção pela empresa, mas avaliam a possibilidade de capturar esta informação pelo percentual de alocação da fábrica de *software*, cuja meta é de 94% de alocação contínua, ou seja, só admitem ter 6% de ociosidade de força de trabalho na fábrica. Atualmente a fábrica possui picos de trabalho e momentos de ociosidade, sendo que a média de alocação mensal é torno de 87%.

3. Engenharia

3.1. Como é feito o gerenciamento de requisitos?

A ferramenta de gestão permite o registro dos requisitos, contemplando ferramentas de rastreabilidade que permitem registar alterações de requisitos e analisar seu impacto no projeto. Esta ferramenta foi desenvolvida com intuito de suportar as exigências de gestão do nível 2 do CMMI, sendo assim, os requisitos são registrados como funcionalidades e mapeados na forma tradicional de desenvolvimento. Com a adoção das técnicas ágeis de desenvolvimento, a equipe técnica registra as estórias (forma de trabalho do *Scrum*) em outra ferramenta. Este é um ponto importante, pois o *ScrumMaster* deve garantir que não haja requisitos não mapeados em qualquer uma das ferramentas.

A empresa está com um projeto interno de atualizar esta ferramenta de gestão, permitindo que seu registro possa respeitar as técnicas ágeis de desenvolvimento. A conclusão deste desenvolvimento estava prevista para o segundo semestre de 2013.

3.2. Como é estabelecida a solução técnica?

A solução técnica é de responsabilidade da equipe de desenvolvimento, podendo acionar alguns recursos da área de infraestrutura como, por exemplo, analistas de TI para auxiliarem na solução técnica. Esta atividade é a primeira a ser executada.

3.3. Existem critérios para verificação e validação do produto?

No final de cada *sprint,* a versão é entregue à área de infraestrutura, responsável pelas áreas de qualidade e garantia do produto. A equipe técnica do projeto designa um recurso para acompanhar os testes, auxiliando com respostas imediatas e no manuseio do sistema.

4. Suporte

4.1. Como é efetuada a gestão de configuração?

A empresa utiliza uma ferramenta de gestão de configuração acoplada à ferramenta de gestão de projetos, então é possível rastrear qual requisito está presente em certa versão. A responsabilidade de gerir este processo é do *ScrumMaster*.

4.2. De que forma é efetuada a garantia da qualidade do produto e processo?

A área de gestão de processos emprega o uso de questionários de pesquisa de clima com intuito de averiguar se os colaboradores estão motivados, esta é uma das métricas organizacionais para medição de processo. Para medição do produto há duas etapas: para projetos em andamento, é considerado o desvio de prazo; para projetos encerrados, é considerado o número de implementações pós-implantação.

O argumento da diretoria para escolha destas métricas é que pessoas motivadas tendem a seguir o processo estabelecido, quando o projeto está com dificuldades, as pessoas tendem a consumir mais horas para contornar ou mesmo solucionar um problema encontrado e, por último, um projeto que resulta num produto de baixa qualidade consome muitas horas de pós-implantação para ajustes.

4.3. Possui atividades de análise e métricas dos resultados dos projetos?

O escritório de projetos é responsável por formalizar o início e o encerramento de cada projeto. Durante todo o ciclo de vida o gerente de projetos é responsável por gerar relatórios que mostrem o desvio de prazo e custo. Como este relatório é obtido pela ferramenta de gestão, as informações possuem o formato da abordagem tradicional, ou seja, requisitos vistos na forma de funcionalidades e não mapeados

como estórias de *Scrum*. Este foi um forte argumento que motivou a empresa a adaptar esta ferramenta para o uso de técnicas ágeis.

4.4. De que forma é efetuada a análise de decisão e resolução?

As decisões de progresso do projeto são de responsabilidade do *ScrumMaster*, ou seja, a empresa delega a autoridade para que este possa utilizar os recursos da empresa para diminuir ou mesmo mitigar problemas durante o desenvolvimento. O escritório de projetos é responsável por averiguar se as ações estão direcionando para o resultado esperado.

4.5. Há atividades de análise causal?

Esta é uma atividade em estudo, ainda não estabelecida pela empresa.

6.3.4. Considerações do Caso

A diretoria da empresa atribui alguns fatores de sucesso para conquista do nível 2 do CMMI:

- Organização possui forte cultura de processo;
- Diretoria e gerentes comprometidos com a metodologia;
- Evolução da ferramenta de gestão para cumprimento das exigência do CMMI;
- Constante capacitação das pessoas quanto à utilização e à manutenção da metodologia de trabalho;
- Constante monitoração dos projetos.

Com o investimento do BNDES, obtido em 2011, espera adaptar a ferramenta de gestão de projetos com as técnicas ágeis de desenvolvimento, para então submeter-se à certificação nível 3. Dessa maneira, a diretoria percebe que, ao utilizar o desenvolvimento ágil, a empresa possui os seguintes ganhos:

- Definição mais detalhada de requisitos;
- Aumento dos pontos de validação;
- Aumento da reutilização do código;
- Maior controle dos prazos de entrega.

Nas palavras do gerente de operações da fábrica:

> "A sinergia proveniente do uso destas duas abordagens pode trazer muitos benefícios. O CMMI pode fornecer todo o gerenciamento de processos que ajudarão a manter e melhorar continuamente a implantação de uma abordagem ágil em uma organização".

Ele ainda expõe o seu receio pelo cenário atual da empresa: em busca da certificação para nível 3: "As práticas, ao se atingir um determinado nível de maturidade, podem trazer valores imediatos. No entanto, a longo prazo, essa maturidade pode ser perdida devido à sobrecarga de processos rumo a um novo nível". A diretoria da empresa é clara ao expor sua satisfação com os resultados da integração do CMMI e desenvolvimento ágil quanto a: maior satisfação dos clientes, maior satisfação interna e melhora no produto de *software* gerado. Deixa claro seu anseio por conquistar o nível 3 de maturidade no final de 2013, mas não demonstra interesse na continuidade da certificação. A empresa acredita que o nível 3 de maturidade lhe garante a confiabilidade necessária quanto à maturidade de seus processos, atendendo requisitos de clientes internacionais e não sobrecarrega a aplicação do ambiente de desenvolvimento ágil.

6.4. SÍNTESE DOS ESTUDOS DE CASOS

O cenário das três empresas estudadas é interessante; pode-se observar que todas tiveram a percepção de que a integração CMMI e o desenvolvimento ágil foram benéficos quanto à maior satisfação do cliente e da equipe interna e à melhora do produto e processo. Mas a forma de estabelecer essa integração foi distinta, uma vez que, no caso da empresa A, a diretriz são os processos de gestão de projetos, cujos processos organizacionais são suportados e a própria empresa se autodesigna como ágil. Já na empresa B, a diretriz são os processos organizacionais e a agilidade está na gestão dos projetos. Por sua vez, a empresa C está na fase de decisão de estabelecer se sua diretriz será de projetos ou organizacional e, na fase atual, está se aproximando mais do cenário da empresa B, ou seja, sua diretriz são os processos organizacionais.

A observação da aplicabilidade da abordagem ágil das três empresas estudadas evidencia o observado por Lindvall et al. (2004), que colocam que o grande desafio é

difundir as práticas ágeis pela organização e evitar o duplo trabalho causado pelo conflito das abordagens tradicionais e ágeis. Confirmando o que foi visto no trabalho de Garcia, Pacheco e Calvo-Manzano (2010), ter uma ferramenta de gestão de projetos é de suma importância para viabilizar a aplicabilidade das práticas ágeis, de forma a manter controles e métricas.

Um ponto inesperado observado na empresa B foi constatar que sua diretoria é direcionada aos custos já orçados, ou seja, o objetivo é não estourar o orçado, mas não há preocupação, pelo menos até o momento, em reduzir o orçamento. A maior glória dos projetos na abordagem ágil tem sido quanto a finalizar o projeto dentro do orçado, no entanto, pouco se reconhece de projetos que ficaram abaixo do orçado.

Um ponto interessante observado nas três empresas estudadas é o fato de terem iniciado seus processos visando os níveis de maturidade CMMI, todas tinham clara visão quanto ao objetivo e missão organizacional, a importância de processos bem estabelecidos e conhecidos por todos e principalmente quanto ao beneficio da disciplina. Mas eram carentes de melhoria no produto e procuravam formas de aumentar a satisfação de seus clientes e de sua equipe interna, que claramente pediam maior autonomia (autogestão) e permissão para criatividade e inovação. Através das práticas ágeis de desenvolvimento conseguiram aumentar a satisfação interna e, de forma disciplinada, promoveram autogestão, maior criatividade e inovação. A consequência foi melhor qualidade no produto de *software*, promovendo melhora na percepção do cliente quanto ao produto final.

Uma relevante constatação observada nas três empresas estudadas é que todas relataram a importância da certificação CMMI para o mercado internacional. A motivação para obter a certificação não foi quanto à necessidade interna de melhoria contínua, mas sim para ingressar no mercado internacional.

7. CONSIDERAÇÕES FINAIS

O *Technical Note* (2008), relatório publicado pelo SEI em novembro de 2008, expõe que, ao combinar o CMMI com as práticas ágeis, é possível obter uma melhora do desempenho da organização. Acrescenta que, quando o CMMI é interpretado como um modelo de processo ao invés de padrão, as organizações têm liberdade para implementar suplementos que estejam de acordo com suas necessidades de melhoria de processo. O CMMI é uma abordagem para melhoria de processo e, ao ser apoiado pela organização, pode perpetuar para além de sua implementação com intuito de não onerar a equipe de desenvolvimento com resíduos ou retrabalho.

Ao observar a forma de trabalho das empresas estudadas, é possível notar que, se não houver uma sinergia entre diretriz organizacional e forma de gestão de projetos, a consequência é o retrabalho, seja na forma de geração de relatórios para atender requisitos gerenciais da alta administração (confeccionados sob a ótica da abordagem tradicional) e relatórios para atender a gestão da equipe técnica (confeccionados sob os princípios ágeis); ou ainda, na coleta de dados de projetos sob abordagem ágil, para atender métricas organizacionais, sob a ótica da abordagem tradicional, de forma a constituir base de comparação com demais projetos da empresa, que não estão aplicando os princípios ágeis. Quando a alta administração e a equipe técnica estão alinhadas quanto aos princípios ágeis, como observado na empresa A, é possível eliminar resíduos ou mesmo retrabalho durante o processo de desenvolvimento de *software,* contribuindo para tornar a organização mais eficiente.

Um ponto relevante do *Technical Note* (2008) é a constatação que, ao interpretar o CMMI como um modelo, é possível observar que as áreas de processo não são os processos propriamente ditos, mas que podem ser operacionalizados onde, quando e em qualquer sequência necessária para realizar o trabalho da empresa. O fato de alcançar os objetivos implica que as áreas de atividades em que os processos são aplicados estão, de fato, sendo melhoradas em algum grau. Os objetivos são componentes requeridos pelo CMMI, as práticas contidas no objetivo são esperadas, mas não obrigatórias. Essa afirmação significa que, para satisfazer um objetivo,

algum tipo de atividade deve ocorrer, de forma a gerar uma saída ou condição persistente que possa fazer entender como a meta foi atingida.

Ao interpretar o CMMI como um modelo e não como padrão, as empresas estudadas estabeleceram seus processos calcados nos princípios ágeis, de modo a produzir saídas que pudessem ser validadas pelo processo de auditoria do CMMI. Estas comprovavam, de alguma forma, que o processo estava estabelecido, podendo ser submetido à avaliação e à análise.

Um ponto que merece atenção, até mesmo como proposta para futuros trabalhos, é quanto ao estabelecimento de quais métricas considerar no ambiente de desenvolvimento ágil. Ao relembrar a análise bibliométrica de métodos ágeis (Seção 3.3), é possível notar que técnicas de gestão de projetos têm sido uma fonte de estudos acadêmicos.

Se técnicas de gestão de projetos, sob a ótica da abordagem ágil, ainda merecem estudos e amadurecimento quanto a sua interpretação e análise, é curioso notar como a Empresa A conseguiu difundir os princípios ágeis para além de seus projetos, de maneira a permear pela organização, a ponto de se autodesignar como Empresa Ágil. A questão não seria como tornar os projetos ágeis, mas sim como tornar a organização ágil.

No cenário da Empresa B, a gestão de orçamento rege a diretriz organizacional, a alta gestão parabeniza projetos concluídos dentro do orçamento previsto, não se preocupando com quanto desse orçamento foi reduzido. Não conseguem notar o tamanho do benefício que os projetos submetidos à abordagem ágil resultaram para a empresa, e um desafio é mostrar que há vantagens em investir tempo para economizar um valor já aprovado para gastar.

O estudo de Mafakheriab et al. (2008) propõe indicadores de agilidade (compostos por: dinamismo, tamanho da equipe, comunicação, testes, perfil e conhecimento dos desenvolvedores e cultura) e o estudo proposto por Qumer e Henderson-Sellers (2008) define o método ágil como uma função envolvendo agilidade, abstração, pessoas, processo, produto, ferramentas, conhecimento e governança. Fica a proposta, para estudos futuros, de mais um componente: a gestão de orçamento. A agilidade atribuída à organização é um conceito importante a ser aprofundado, ainda mais se já há estudos propondo indicadores de agilidade.

Ser ágil e flexível às mudanças é importante desde que tenha um processo estabelecido, de forma a capturar bons resultados de forma contínua propiciando à empresa ser eficaz de maneira sustentável. O modelo CMMI é capaz de prover o conhecimento para amadurecimento dos processos organizacionais e, o desenvolvimento ágil, de acomodar as mudanças e fornecer respostas rápidas neste cenário de incertezas que permeia o desenvolvimento de projetos de *software*s.

Hipótese Básica

Relembrando, a hipótese básica desta dissertação é que os processos elencados na Tabela 1 de fato constituiriam o maior esforço de adaptação, sendo organizados de acordo com a categoria do CMMI: gerenciamento de projeto, gerenciamento de processo, engenharia e suporte. Abaixo há uma síntese sobre os principais pontos a serem considerados por categoria:

Gerenciamento de projeto: muitos métodos ágeis são designados a mitigar certos tipos de riscos, particularmente aqueles que afetam a mudança de requisitos e planejamento, porém é menos óbvio como endereçam os riscos a longo prazo. Já para a gestão quantitativa de projeto, a dificuldade estaria em processar os dados estatísticos.

Gerenciamento de processo: a dificuldade estaria em atender todos os objetivos desta categoria, justamente pela abordagem ágil não ter foco no processo organizacional, não compreendendo sua definição e condução; falta de definição de treinamento organizacional; omissão quanto à definição do processo de desempenho organizacional, seja como forma de melhoria ou inovação.

Engenharia: não deve gerar dificuldade por se tratar de questões de interesse do desenvolvimento ágil, que compreendem o gerenciamento de requisitos, solução técnica, integração de produto, verificação e validação.

Suporte: as dificuldades seriam manter o processo de garantia de qualidade de produto para todos os projetos e definir métricas e forma de análise destes dados.

Com intuito de averiguar se esses seriam os pontos de maior esforço de adaptação para integração do CMMI e o desenvolvimento ágil, foi elaborado o questionário de pesquisa (Seção 5.2.1), submetido às empresas estudadas (Capítulo 6). Pela interpretação das respostas foi possível confirmar pontos de maior esforço de adaptação, conforme Tabela 16.

Tabela 16 - Quadro de Integração conclusão dos esforços de adaptação

Gerenciamento de Projeto		Desenvolvimento Ágil	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Gestão de Contrato do Fornecedor (SAM)	2	Não Suportado	No desenvolvimento ágil não há nenhuma abordagem específica para fornecedores, já que as iterações são planejadas para serem realizadas pela equipe e não por terceiros. Pelo fato de as iterações serem curtas, a entrega de um componente de produto pelo fornecedor poderia ser considerada crítica e dificultar as atividades de integração.
As empresas estudadas possuem processo para contratação de fornecedores, porém por diretriz organizacional aplica-se somente a questões de infraestrutura ou treinamentos.			
Gestão de Risco (RSKM)	3	Parcialmente Suportado	Não há uma forma ou recomendação explícita de gestão de riscos para abordagem ágil de como identificar, analisar, priorizar ou mesmo mitigar. Geralmente essa atividade fica sob os critérios do líder de projeto. Sendo assim, há necessidade de maior formalismo quanto à forma. Um ponto positivo é que as iterações curtas permitem que identificação, análise e mitigação sejam aplicadas rapidamente.
As empresas estudadas possuem processo para gestão de riscos, aplicado por iteração (*sprint*) de forma que todos os pontos de risco levantados devem ser respondidos até a conclusão do projeto. Apesar do processo estabelecido, as empresas diferem na forma de condução da gestão de riscos, na designação do responsável e na forma de formalização.			
Gestão Quantitativa de Projeto (QPM)	4	Não Suportado	Os objetivos e gestão de desempenho de projeto são estabelecidos, porém para conseguirmos um efeito estatístico é necessário certa homogeneidade nos dados, o que geralmente não ocorre nos projetos ágeis devido ao fato de o foco estar nas pessoas e não no processo. Cada projeto tende a ser único na sua forma de condução.
As empresas estudadas possuem processo para gestão quantitativa de projeto, sendo que as empresas A e B já possuem dados históricos. As empresas determinaram que a coleta dos dados deve ser por *sprint,* a escolha de quais métricas seguir foi individual, orientada de acordo com a necessidade organizacional. Este ponto merece um grande esforço de adaptação, pois a empresa deve possuir um sistema capaz de coletar informações sobre o progresso dos projetos e definir métricas que possam ser aplicados a todos os projetos.			

Gerenciamento de Processo		Desenvolvimento Ágil	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Foco no Processo Organizacional (OPF)	3	Não Suportado	O desenvolvimento ágil tem maior foco no produto e não no processo. Os princípios e conceitos ágeis são geralmente aplicados aos projetos e não a organização. E há um estimulo quanto ao desenvolvimento do conhecimento tácito o que pode ser aproveitado entre projetos, pelo aproveitamento dos recursos humanos, mas dificulta a disseminação deste conhecimento para organização.
As empresas estudadas possuem processos organizacionais definidos: na empresa A estão submetidos aos processos de gestão de projetos; nas empresas B e C os processos organizacionais são mandatórios e no caso, a gestão de projetos fica submetida a essas diretrizes. Neste caso, há um retrabalho, visto que o processo organizacional está montado sob a ótica tradicional e os processos de gestão de projetos estão sob a abordagem ágil. Este ponto merece um grande esforço de adaptação visto que o processo organizacional deve ser repensado.			
Definição do Processo Organizacional (OPD)	3	Parcialmente Suportado	No desenvolvimento ágil há regras e diretrizes para equipe, definição do ciclo de vida e critérios para forma de trabalho. A grande questão é que esses conceitos geralmente ficam na fronteira dos projetos e deve-se propiciar uma forma de esses princípios permearem pela organização.
As empresas estudadas possuem processos organizacionais definidos: na empresa A estão submetidos aos processos de gestão de projetos; nas empresas B e C os processos organizacionais são mandatórios e, no caso, a gestão de projetos fica submetida a essas diretrizes.			
Treinamento Organizacional (OT)	3	Parcialmente Suportado	O plano e a capacidade de treinamento limitam-se às fronteiras do projeto o que muitas vezes acaba sendo executado de modo informal, não havendo explicitamente métricas para avaliar efetivamente, deste treinamento, senão a própria evolução das iterações. Mais uma vez, há necessidade de ampliar essa fronteira do projeto para organização.
As empresas estudadas possuem processos para treinamento organizacional, sendo que a administração deste item fica fora do contexto da equipe de desenvolvimento, estando sob responsabilidade da área de recursos humanos ou escritório de projetos, dependendo de cada caso. Mas em todas as empresas estudadas, a necessidade é oriunda de projetos, sendo inclusive o responsável financeiro.			

Gerenciamento de Processo		Desenvolvimento Ágil	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Desempenho do Processo Organizacional (OPP)	4	Parcialmente Suportado	Modelos e linhas de controle de desempenho são estabelecidas ao projeto, mas não externalizadas à organização.
Um ponto em comum das 3 empresas estudadas é que estas tentam, através de métricas como índice de satisfação interna e de cliente, externalizar o desempenho do processo organizacional. Pode ser uma maneira de medir como a forma de gestão do projeto influencia na organização.			
Gestão do Desempenho Organizacional (OPM)	5	Parcialmente Suportado	A gestão do desempenho é efetuada no contexto do projeto e não da organização.
Assim como visto no item “Desempenho do Processo Organizacional”, a medição do índice de satisfação interna e de cliente pode ser uma forma de gestão.			

Suporte		Desenvolvimento Ágil	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Garantia da Qualidade do Produto e Processo (PPQA)	2	Parcialmente Suportado	No desenvolvimento ágil a preocupação maior é quanto ao produto e menos ao processo. Há uma necessidade quanto à melhora da percepção, ao processo e, consequentemente, à coleta de dados.
As empresas estudadas possuem método de coleta de dados definido por *sprint*, onde ao longo do projeto é possível analisar as métricas de qualidade do produto e processo. A questão é que estas métricas foram construídas individualmente por cada empresa.			
Análise e Métricas (MA)	2	Parcialmente Suportado	Durante as iterações é possível coletar vários dados e estabelecer métricas, mas deve haver disciplina pois os ciclos de desenvolvimento mais curto propiciam maior informalidade. Outro ponto importante a considerar é quanto à medição do progresso que tende a ser por ciclo e não do projeto.
As empresas estudadas fundaram suas análises e métricas por *sprint*, sendo primordial o uso de um sistema de gestão para coleta dos dados.			

Suporte		Desenvolvimento Ágil	
Áreas de Processo	**Nível**	**Critério**	**Explicação**
Resolução e Análise de Decisão (DAR)	3	Parcialmente Suportado	O processo de avaliação de alternativas existe, porém não é executado de modo formal com critérios e métodos de avaliação.
As empresas estudadas possuem um sistema de gestão para coleta de dados do progresso de projetos, e possuem processos que suportam a tomada de decisão, mas devem ser ágeis na análise da informação, uma vez que são estabelecidos por sprint.			
Resolução e Análise Causal (CAR)	5	Parcialmente Suportado	A identificação das causas e as ações necessárias para correção são endereçadas, porém de modo informal e implementadas rapidamente.
As empresas A e B conseguiram atender este item ao designar ao *scrumaster* a responsabilidade e autonomia para gerir as causas e ações necessárias para resolução dos problemas.			

Fonte: Produzida pela autora

A conclusão é que apesar de as empresas estudadas terem como base a definição de trabalho por *sprint* (iterações) fica clara a distinção na forma de condução, definição das métricas e análise dos dados. Todas optaram por usar um sistema de gestão de projetos como apoio do processo de desenvolvimento de *software*, principalmente como forma de coleta de dados, mas os sistemas são distintos e customizados. Uma sugestão para estudo futuro seria estabelecer um modelo de gestão de projetos em ambientes ágeis sob a ótica do CMMI.

Constatou-se que a integração CMMI e o desenvolvimento ágil são possíveis, inclusive essa sinergia traz benefícios à empresa. O importante é interpretar o CMMI como um modelo, sendo que para o nível de projeto, o foco deve ser de alto nível de abstração em “o que” deve ser feito e não em qual metodologia de desenvolvimento deve ser aplicada; a abordagem ágil deve ser vista como a forma de trabalho a ser estabelecida e focar em “como” desenvolver projetos de produtos. Desta forma, CMMI e ambiente de desenvolvimento ágil podem coexistir, propiciando que a empresa possa, com disciplina, amadurecer com seus processos e ser flexível para atender mudanças de requisitos em seus produtos.

Hipóteses Complementares

Para atender as hipóteses complementares foi considerada a pesquisa de campo efetuada nas empresas estudadas, de forma a entender o processo organizacional expresso nos itens *O Processo* e *Considerações Finais*, presentes no Capítulo 6. Temos as hipóteses complementares:

- As empresas, presentes neste estudo, foram motivadas a adotar a abordagem ágil por acreditar que haveria melhoria de qualidade no produto.
    - Hipótese confirmada.
- A percepção das empresas que integraram o CMMI e o desenvolvimento ágil é que o processo melhorou.
    - Hipótese confirmada.

A curiosidade destes dois itens é que todas as empresas estudadas afirmaram ter optado inicialmente pelo CMMI como forma de conquistar o mercado, por acreditar que a certificação creditaria maior confiabilidade de possíveis clientes ao processo maduro da empresa. Mas ter somente o CMMI não refletiu em melhoria de qualidade do produto e, por isso, optaram pela adoção do ambiente de desenvolvimento ágil e esta sinergia refletiu na melhoria do produto.

- Houve evolução nos projetos tidos como de sucesso, respeitando os critérios individuais das empresas quanto a qualidade, prazo ou custo.
    - Hipótese confirmada.

A sinergia do CMMI e ambiente de desenvolvimento ágil propiciou melhoria de processo e produto e em todas as empresas estudadas foi notório o aumento da satisfação da equipe interna que se mostrou muito positivo para melhoria dos índices de sucesso dos projetos.

Uma limitação deste estudo foi não ter encontrado disponibilidade de empresas de *software* com certificação CMMI de nível 3 e 4 que tivessem adotado ambiente de desenvolvimento ágil, no intuito de engrossar e endossar as descobertas desta dissertação. Outro ponto limitante foi descobrir que as empresas estudadas, apesar de possuírem certificação CMMI e adotarem o ambiente de desenvolvimento ágil, distinguiram na forma de implantação e condução, mostrando que a sinergia do CMMI e ambiente ágil é possível e benéfica, mas não há um modelo a ser seguido.

As observações quanto ao esforço de adaptação das atividades são singulares ao contexto de cada empresa.

Apesar deste último item ser um ponto limitante é também uma grande oportunidade de estudo tanto com intuito de estabelecer um modelo de implantação da integração CMMI e desenvolvimento ágil quanto na definição de dados e métricas que sistemas de gestão de projetos podem considerar com intuito de propiciar gestão de riscos, quantitativa e qualitativa dos projetos, de forma a medir a agilidade entre as empresas e não somente entre os projetos da mesma empresa.

REFERÊNCIAS BIBLIOGRÁFICAS

ABRAHAMSSON P.; SALO O.; RONKAINEN J. Agile Software development methods, **VTT Technical Research Centre of Finland**, 2002.

AGILE ALLIANCE. **Manifesto for agile software development**. Disponível em <http://www.agilemanifesto.org/>. Acessado em 15/08/2011.

AHERN, Dennis M.; CLOUSE, Aaron; TURNER, Richard. **CMMI Distilled:** A Practical Introduction to Integrated Process Improvement. – 3nd edition. Addison Wesley, 2008.

BARTIÉ, Alexandre. **Garantia da Qualidade de Software:** adquirindo maturidade organizacional. Rio de Janeiro, Elservier, 2002.

BASKERVILLE, R. et al. How Internet software companies negotiate quality. **Computer**, v. 34, n. 5, p. 51-+, May 2001.

BOEHM B.; TURNER R.; **Balancing Agility and Discipline:** a guide for the perplexed. Addison-Wesley Professional, 2003.

BOEHM, B.; TURNER, R. Management challenges to implementing agile process in traditional development organizations. **IEEE Software**, 2005.

COCKBURN, A**. Agile Software Development**. Boston: Addison- Wesley, 2001.

DECARLO, D. **Extreme Project Management:** using leadership, principles, and tools to deliver value in the face of volatility. Sao Francisco, 2004.

DOS SANTOS, R. P.; DE OLIVEIRA, K. M.; DA SILVA, W. P. Evaluating the service quality of software providers appraised in CMM/CMMI. **Software Quality Journal**, v. 17, n. 3, p. 283-301, 2009.

EBERT, C.; DE MAN, J. Effectively utilizing project, product and process knowledge. **Information and Software Technology**, v. 50, n. 6, p. 579-594, 2008.

FERNANDES, Aguinaldo Aragon; TEIXEIRA, Descartes de Souza. **Fábrica de Software**: Implantação e gestão de operações. São Paulo: Editora Atlas, 2004.

FOGELSTROM, N. D. et al. The impact of agile principles on market-driven software product development. **Journal of Software Maintenance and Evolution-Research and Practice**, v. 22, n. 1, p. 53-80, 2010.

GARCIA, I.; PACHECO, C. A Web-based Tool for Automatizing the Software Process Improvement Initiatives in Small Software Enterprises. **Ieee Latin America Transactions**, v. 8, n. 6, p. 685-694, 2010.

GARCIA, I.; PACHECO, C.; CALVO-MANZANO, J. Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting**. Iet Software**, v. 4, n. 4, p. 237-251, 2010.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 1991.

GOULD, P. What is agility? **IEE Manufacturing Enginner**, vol. 76, n. 1, p. 28-31, 1997.

GREER, D.; HAMON, Y. Agile Software Development. **Software-Practice & Experience**, v. 41, n. 9, p. 943-944, Aug 2011.

HANSSON, C.; DITTRICH, Y; GUSTAFSSON B.; ZARNAK, S. How agile are industrial software development pratices? **Journal of Systems and Software**, vol. 79, n. 9, p. 1295-1311, 2006.

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

HSUEH, N.-L. et al. Applying UML and software simulation for process definition, verification, and validation. **Information and Software Technology**, v. 50, n. 9-10, p. 897-911, 2008.

HUANG, S. J.; HAN, W. M. Selection priority of process areas based on CMMI continuous representation. **Information & Management**, v. 43, n. 3, p. 297-307, 2006.

KAHKONEN, T.; ABRAHAMSSON, P. Achieving CMMI level 2 with enhanced extreme programming approach. In: BOMARIUS, F. I. H. (Ed.). **Product Focused Software Process Improvement**, v.3009, 2004. p.378-392. (Lecture Notes in Computer Science).

KOURGANOFF, W. **Face oculta da universidade**, São Paulo: Unesp, 1990.

KULPA, M. K.; JOHNSON, K. A. **Interpreting the CMMI®:** A Process Improvement Approach. 2nd. ed. Florida: Auerbach, 2008.

LARSSON, S.; EKDAHL, F. Selecting CMMI appraisal classes based on maturity and openness. In: BOMARIUS, F. I. H. (Ed.). **Product Focused Software Process Improvement**, v.3009, 2004. p.457-470. (Lecture Notes in Computer Science).

LAZZARI, M. L. **Implantação de Melhoria de Processos de Desenvolvimento de Software Utilizando o CMMI**: Um Estudo de Caso em uma Empresa de Varejo. 2006. 70p. Monografia (Sistemas de Informação) – Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2006.

LEE S.; YONG H. Distributed agile: project management in a global environment.Empir **Software Eng**, v.15, p. 204-217, 2010.

LESTER, N. G. et al. Investigating the role of CMMI (R) with expanding company size for small- to medium-sized enterprises. **Journal of Software Maintenance and Evolution-Research and Practice**, v. 22, n. 1, p. 17-31, 2010.

LIKER, J. K. **The Toyota way**: 14 management principles from the world's greatest manufacturer. New York: McGraw-Hill, 2004.

LIPPERT, M. et al. Developing complex projects using XP with extensions. **Computer**, v. 36, n. 6, p. 67-+, Jun 2003.

LIPPERT, M. Towards a proper integration of large refactorings in agile software development. In: ECKSTEIN, J. B. H. (Ed.). Extreme Programming and Agile Processes in Software Engineering, **Proceedings**, v.3092, 2004. p.113-122. (Lecture Notes in Computer Science).

LINDVALL, M. et al. Agile software development in large organizations. **Computer**, v. 37, n. 12, p. 26-+, Dec 2004.

LIOU, J.-C. On Improving CMMI in an Immature World of Software Development. **Journal of Information Science and Engineering**, v. 27, n. 1, p. 213-226, 2011.

LYCETT, M. et al. Migrating agile methods to standardized development practice. **Computer**, v. 36, n. 6, p. 79-+, Jun 2003.

MAFAKHERIAB, F; NASIRIB, F; MOUSAVI M. Project agility assessment: an integrated decision analysis approach. **Production Planning & Control**. Vol. 19, No. 6, p. 567–576, 2008.

MCMAHON, P. E. Extending Agile Methods: A Distributed Project and Organizational Improvement Perspective, CrossTalk. **The Journal of Defense Software Engineering**, vol. 18, issue 5, pp. 1619, 2005.

MEGLIORINI, E.; GUERREIRO, R. A percepção dos gestores sobre quanto a fatores competitivos nas empresas produtoras de bens de capital sob encomenda: um estudo exploratório. **Base Revista de Administração e Contabilidade da Unisinos**, São Leopoldo - RS, v. 1, n. 2, p. 5-14, 2004.

MIDDLETON, P. Lean software development: Two case studies. **Software Quality Journal**, v. 9, n. 4, p. 241-252, 2001.

MIDDLETON, P.; FLAXEL, A.; COOKSON, A. Lean software management case study: Timberline Inc. In: BAUMEISTER, H. M. M. H. M. (Ed.). **Extreme Programming and Agile Processes in Software Engineering, Proceedings**, v.3556, 2005. p.1-9. (Lecture Notes in Computer Science).

MIKULENAS, G.; BUTLERIS, R.; NEMURAITE, L. An approach for the metamodel of the framework for a partial agile method adaptation. **Information Technology and Control**, v. 40, n. 1, p. 71-82, 2011 2011.

MISRA, S.; KUMAR, V.; KUMAR, U. Identifying some important success factors in adopting agile software development practices. **The Journal of Systems and Software**, v.82, p.1869–1890, 2009.

NIELSEN, J.; MCMUNN, D. The agile journey - Adopting XP in a large financial services organization. In: BAUMEISTER, H. M. M. H. M. (Ed.). Extreme Programming and Agile Processes in Software Engineering, **Proceedings**, v.3556, 2005. p.28-37. (Lecture Notes in Computer Science).

PETERSEN, K. Measuring the flow in lean software development. **Software-Practice & Experience**, v. 41, n. 9, p. 975-996, Aug 2011.

PETERSEN, K.; WOHLIN, C. Software process improvement through the Lean Measurement (SPI-LEAM) method. **Journal of Systems and Software**, v. 83, n. 7, p. 1275-1287, 2010.

PINO, F. J.; GARCIA, F.; PIATTINI, M. Software process improvement in small and medium software enterprises: a systematic review. **Software Quality Journal**, v. 16, n. 2, p. 237-261, Jun 2008.

POPPENDIECK M., POPPENDIECK T. **Lean software development an agile toolkit**. Boston: Addison-Wesley, 2003.

PORT, D.; BUI, T. Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications. **European Journal of Information Systems**, v. 18, n. 4, p. 317-331, Aug 2009.

PORTER, MICHAEL E. **Vantagem competitiva das nações**. Rio de Janeiro: Ed. Campus, 1993.

QUMER, A.; HENDERSON-SELLERS, B. An evaluation of the degree of agility in six agile methods and its applicability for method engineering. **Information and Software Technology**, v. 50, n. 4, p. 280-295, Mar 2008.

SANTOS, A. R. **Metodologia científica: a construção do conhecimento**, 5. ed. Rio de Janeiro: 2002.

SANTOS, P. SCRUM meets CMMi - Agility and discipline combined. **Dr Dobbs Journal**, v. 32, n. 9, p. 28-33, 2007.

SEI CMMI-DEV v1.3 (2010). Disponível em <http://www.sei.cmu.edu/reports/10tr033.pdf>. Acessado em 15/08/2011.

SENHAR, A. Strategic project leadership: toward a strategic approach to project management. **R&D Management**, v. 34, n.5, p..394-414, 2001.

SHENHAR, A.; DVIR, D. Project Management Research - the challenge and opportunity. **Project Management Journal**, v. 38, n.2, p.93-99, 2007.

SPINAK, E. **Diccionario enciclopédico de bibliometría, cienciometría y informetría**. Montevideo, 1996.

STAKE, Robert E. **The art of case estudy research**. Thousand Oaks, CA: Sage Publications, 2005.

STAPLES, M; NIAZI, M. Systematic review of organizational motivations for adopting CMM-based SPI. **Information and Software Technology**, v. 50, n.7, p.605-620, 2008.

STAPLES, M. et al. An exploratory study of why organizations do not adopt CMMI. **Journal of Systems and Software**, v. 80, n. 6, p. 883-895, 2007.

SUIKKI, R; TROMSTEDT, R.; HAAPASALO, H. Project management competence development framework in turbulent business environment. **Technovation**, v. 26, n.5, p.723-738, 2006.

SWABER, K. **Agile Project Management with SCRUM**. Microsoft Press: Washington, 2004.

TALBY, D. et al. Agile software testing in a large-scale project. **Ieee Software**, v. 23, n. 4, p. 30-+, Jul-Aug 2006.

TECHNICAL NOTE CMU/SEI-2008-TN-003 **Software Engineering Process Management Unlimited distribution subject to the copyright**. Page 2. www.sei.cmu.edu/reports/08tn003.pdf-2008-11-12.

TURNER, R. Agile development: Good process or bad attitude? In: OIVO, M. K. S. (Ed.). Product Focused Software Process Improvement, **Proceedings**, v.2559, 2002. p.134-144. (Lecture Notes in Computer Science).

TURNER, R.; JAIN, A. **Agile Meets CMMI**: Culture Clash or Common Cause, In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/Agile Universe, pp. 153165, 2002.

TONINI, A. C.; CARVALHO, M. M.; SPINOLA, M. M. Contribuição dos modelos de qualidade e maturidade na melhoria dos processos de software. **Produção**, v. 18, n. 2, p. 275-286, 2008.

VELDMAN, J.; KLINGENBERG, W. Applicability of the capability maturity model for engineer-to-order firms. **International Journal of Technology Management**, v. 48, n. 2, p. 219-239, 2009.

VENUGOPAL, C. Single coal set: A new paradigm for IT megaproject success. **Ieee Software**, v. 22, n. 5, p. 48-+, Sep-Oct 2005.

VON WANGENHEIM, C. G.; ROSSA HAUCK, J. C.; VON WANGENHEIM, A. Enhancing Open Source Software in Alignment with CMMI-DEV. **Ieee Software**, v. 26, n. 2, p. 59-67, 2009.

WANG, X.; GOBBO, F.; LANE, M. **Turning Time from Enemy into an Ally Using the Pomodoro Technique**. 2010.

WEBER, B.; WILD, W. Application of lean and agile principles to workflow management. In: ECKSTEIN, J. B. H. (Ed.). Extreme Programming and Agile Processes in Software Engineering, **Proceedings**, v.3092, 2004. p.258-261. (Lecture Notes in Computer Science).

YIN R. **Estudo de caso planejamento e método**. Porto Alegre: Bookman, 2010.

APÊNDICE 1 - CMMI-DEV V1.3

Para a construção deste quadro foram considerados os conceitos de objetivos genéricos e específicos por área de processo CMMI, extraídos do SEI através do CMMI-DEV versão 1.3 de 2010, conforme ilustrado na Tabela 17.

Tabela 17 - Objetivos genéricos e especifícos do CMMI

Gerenciamento de Projeto			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Planejamento do Projeto (PP)	2	SG 1: Estabelecer estimativas SG 2: Desenvolver um plano de projeto SG 3: Obter compromissos de planejamento	SP 1.1: Estimar o escopo do projeto SP 1.2: Estabelecer estimativas de produto de trabalho e atributos de tarefas SP 1.3: Definir as fases do ciclo de vida do projeto SP 1.4: Estimar esforço e custo SP 2.1: Estabelecer prazo e orçamento SP 2.2: Identificar os riscos do projeto SP 2.3: Plano de gerenciamento de dados SP 2.4: Plano de recursos de projeto SP 2.5: Plano de conhecimento e perfil requerido SP 2.6: Plano de envolvimento de *Stakeholders* SP 2.7: Estabelecer o plano de projeto SP 3.1: Revisar planos que afetam o projeto SP 3.2: Reconciliar nível de trabalho e recurso SP 3.3: Obter compromisso de planejar
Controle e Monitoração do Projeto (PMC)	2	SG 1: Monitorar o projeto *versus* o planejado SG 2: Gerenciar ações corretivas para encerramento	SP 1.1: Monitorar parâmetros de planejamento de projeto SP 1.2: Monitorar compromissos SP 1.3: Monitorar riscos de projetos SP 1.4: Monitorar gerenciamento de dados SP 1.5: Monitorar envolvimento de *Stakeholders* SP 1.6: Conduzir revisões de progresso SP 1.7: Conduzir revisões de marco de datas SP 2.1: Analisar questões SP 2.2: Tomar ações corretivas SP 2.3: Gerenciar ações corretivas
Gestão de Contrato do Fornecedor (SAM)	2	SG 1: Estabelecer contrato de fornecedor SG 2: Satisfazer contrato de fornecedor	SP 1.1: Determinar o tipo de aquisição SP 1.2: Selecionar fornecedores SP 1.3: Estabelecer o contrato de fornecedor SP 2.1: Executar o contrato de fornecedor SP 2.2: Aceitar o produto requerido SP 2.3: Assegurar transição de produtos

Gerenciamento de Projeto			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Gestão de Projeto Integrado (IPM)	3	SG 1: Usar processo definido de projeto SG 2: Coordenar e colaborar com *Stakeholders* relevantes	SP 1.1: Estabelecer um processo definido de projeto SP 1.2: Usar processos organizacionais para planejamento das atividades de projeto SP 1.3: Estabelecer o ambiente de trabalho do projeto SP 1.4: Integrar planos SP 1.5: Gerenciar o projeto usando planos integrados SP 1.6: Estabelecer a equipe SP 1.7: Contribuir ao processo organizacional SP 2.1: Gerenciar o envolvimento de *Stakeholders* SP 2.2: Gerenciar dependências SP 2.3: Resolver questões de coordenação
Gestão de Risco (RSKM)	3	SG 1: Preparar para gerenciamento de riscos SG 2: Identificar e analisar riscos SG 3: Mitigar riscos	SP 1.1: Determinar origem dos riscos e categorias SP 1.2: Determinar os parâmetros de riscos SP 1.3: Estabelecer uma estratégia de gerenciamento de riscos SP 2.1: Identificar riscos SP 2.2: Avaliar, categorizar e priorizar riscos SP 3.1: Desenvolver plano de mitigação de riscos SP 3.2: Implementar plano de mitigação de riscos
Gestão Quantitativa de Projeto (QPM)	4	SG 1: Preparar para o gerenciamento quantitativo SG 2: Gerenciar quantitativamente o projeto	SP 1.1: Estabelecer os objetivos do projeto SP 1.2: Compor o processo definido SP 1.3: Selecionar subprocessos e atributos SP 1.4: Selecionar métricas e técnicas de análise SP 2.1: Monitorar o desempenho de subprocessos selecionados SP 2.2: Gerenciar o desempenho de projetos SP 2.3: Analisar a causa raiz do desempenho

Gerenciamento de Processo			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Foco no Processo Organizacional (OPF)	3	SG 1: Determinar as oportunidades de melhoria no processo SG 2: Planejar e implementar ações no processo SG 3: Implantar processos organizacionais e incorporar experiências	SP 1.1: Estabelecer as necessidades do processo organizacional SP 1.2: Avaliar o processo organizacional SP 1.3: Identificar as melhorias no processo organizacional SP 2.1: Estabelecer planos de ação do processo SP 2.2: Implementar planos de ação do processo SP 3.1: Implantar processo organizacional SP 3.2: Implantar processos padrões SP 3.3: Monitorar a implementação SP 3.4: Incorporar experiências no processo organizacional
Definição do Processo Organizacional (OPD)	3	SG 1: Estabelecer processo organizacional	SP 1.1: Estabelecer processo padrão SP 1.2: Estabelecer a descrição do modelo de ciclo de vida SP 1.3: Estabelecer critérios e diretriz para forma de trabalho SP 1.4: Estabelecer o repositório de métricas da organização SP 1.5: Estabelecer a biblioteca de processos da organização SP 1.6: Estabelecer padrões de ambiente de trabalho SP 1.7: Estabelecer regras e diretrizes para equipes
Treinamento Organizacional (OT)	3	SG 1: Estabelecer a capacidade de treinamento organizacional SG 2: Prover treinamento	SP 1.1: Estabelecer necessidade de treinamento estratégico SP 1.2: Determinar quais necessidades de treinamento são de responsabilidade da organização SP 1.3: Estabelecer um plano de treinamento tático organizacional SP 1.4: Estabelecer uma capacidade de treinamento SP 2.1: Entregar o treinamento SP 2.2: Estabelecer os registros de treinamento SP 2.3: Avaliar a efetividade do treinamento
Desempenho do Processo Organizacional (OPP)	4	SG 1: Estabelecer modelos e linhas de controle de desempenho	SP 1.1: Estabelecer objetivos de desempenho de processo e qualidade SP 1.2: Selecionar processos SP 1.3: Estabelecer métricas de desempenho de processo SP 1.4: Analisar desempenho de processo e estabelecer linhas de controle de desempenho de processo SP 1.5: Estabelecer modelos de desempenho de processo

Gerenciamento de Processo			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Gestão do Desempenho Organizacional (OPM)	5	SG 1: Gerenciar o desempenho do negócio SG 2: Selecionar melhorias SG 3: Implantar melhorias	SP 1.1: Manter objetivos do negócio SP 1.2: Analisar dados de desempenho do processo SP 1.3: Identificar potenciais áreas para melhoria SP 2.1: Extrair melhorias sugeridas SP 2.2: Analisar melhorias sugeridas SP 2.3: Validar melhorias SP 2.4: Selecionar e implementar melhorias para desenvolvimento SP 3.1: Planejar a implantação SP 3.2: Gerenciar a implantação SP 3.3: Avaliar os efeitos da melhoria

Engenharia			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Gestão de Requisitos (REQM)	2	SG 1: Gerenciar requisitos	SP 1.1: Entender os requisitos SP 1.2: Obter compromisso de requisitos SP 1.3: Gerenciar mudanças de requisitos SP 1.4: Gerenciar rastreabilidade bidirecional de requisitos SP 1.5: Assegurar alinhamento entre trabalho de projeto e requisitos
Desenvolvimento de Requisitos (RD)	3	SG 1: Desenvolver requisitos do cliente SG 2: Desenvolver requisitos do produto SG 3: Analisar e validar requisitos	SP 1.1: Extrair necessidades SP 1.2: Transformar necessidades de *Stakeholders* em requisitos do cliente SP 2.1: Estabelecer os requisitos de produto e componentes de produto SP 2.2: Alocar requisitos de componentes de produtos SP 2.3: Identificar requisitos de interfaces SP 3.1: Estabelecer cenários e conceitos operacionais SP 3.2: Estabelecer uma definição de funcionalidade requerida e atributos de qualidade SP 3.3: Analisar requisitos SP 3.4: Analisar requisitos para alcançar o equilíbrio SP 3.5: Validar requisitos
Solução Técnica (TS)	3	SG 1: Selecionar soluções de componente de produtos SG 2: Desenvolver o projeto SG 3: Implementar o projeto de produto	SP 1.1: Desenvolver soluções alternativas e critérios de seleção SP 1.2: Selecionar soluções de componente de produtos SP 2.1: Desenhar o produto ou componente de produto SP 2.2: Estabelecer um pacote de dados técnicos SP 2.3: Desenhar interfaces usando critérios SP 2.4: Desempenhar a análise de reuso, compra ou construção SP 3.1: Implementar o projeto SP 3.2: Desenvolver a documentação de suporte do produto

Engenharia			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Integração de Produto (PI)	3	SG 1: Preparar para integração de produto SG 2: Assegurar compatibilidade de interface SG 3: Montar componentes de produto e entregar produto	SP 1.1: Estabelecer uma estratégia de integração SP 1.2: Estabelecer o ambiente de integração do produto SP 1.3: Estabelecer os procedimentos e critérios de integração do produto SP 2.1: Rever a plenitude das descrições de interfaces SP 2.2: Gerenciar interfaces SP 3.1: Confirmar a prontidão de componentes de produto para integração SP 3.2: Montar componentes de produto SP 3.3: Avaliar a montagem de componentes de produto SP 3.4: Empacotar e entregar o produto ou componente de produto
Verificação (VER)	3	SG 1: Preparar para verificação SG 2: Desempenhar revisões por pares SG 3: Verificar produto de trabalho selecionado	SP 1.1: Selecionar produto de trabalho para verificação SP 1.2: Estabelecer o ambiente de verificação SP 1.3: Estabelecer critérios e procedimentos de verificação SP 2.1: Preparar para revisões em pares SP 2.2: Conduzir a revisão em pares SP 2.3: Analisar os dados da revisão em pares SP 3.1: Desempenhar a verificação SP 3.2: Analisar os resultados da verificação
Validação (VAL)	3	SG 1: Preparar para validação SG 2: Validar produto ou componente de produto	SP 1.1: Selecionar produtos para validação SP 1.2: Estabelecer o ambiente de validação SP 1.3: Estabelecer critérios e procedimentos de validação SP 2.1: Desempenhar a validação SP 2.2: Analisar os resultados da validação

Suporte			
Áreas de Processo	**Nível**	**Objetivos Genéricos**	**Objetivos Específicos**
Gestão de Configuração (CM)	2	SG 1: Estabelecer linha de controle SG 2: Mapear e controlar mudanças SG 3: Estabelecer integridade	SP 1.1: Identificar itens de configuração SP 1.2: Estabelecer um sistema de gerenciamento de configuração SP 1.3: Criar ou estabelecer linhas de controle SP 2.1: Solicitar o mapeamento das mudanças SP 2.2: Controlar itens de configuração SP 3.1: Estabelecer registros de gestão de configuração SP 3.2: Realizar auditoria de configuração
Garantia da Qualidade do Produto e Processo (PPQA)	2	SG 1: Avaliar objetivamente o processo e produto de trabalho SG 2: Prover percepções objetivas	SP 1.1: Avaliar objetivamente o processo SP 1.2: Avaliar objetivamente o produto do trabalho SP 2.1: Comunicar e resolver questões não conformes SP 2.2: Estabelecer registros
Análise e Métricas (MA)	2	SG 1: Alinhar atividades de análise e métricas SG 2: Prover resultados mensuráveis	SP 1.1: Estabelecer os objetivos das métricas SP 1.2: Especificar as métricas SP 1.3: Especificar a coleta de dados e procedimentos de armazenagem SP 1.4: Especificar procedimentos de análise SP 2.1: Obter dados mensuráveis SP 2.2: Analisar os dados mensurados SP 2.3: Armazenar dados e resultados SP 2.4: Comunicar resultados
Resolução e Análise de Decisão (DAR)	3	SG 1: Avaliar alternativas	SP 1.1: Estabelecer diretrizes para análise de decisão SP 1.2: Estabelecer critérios de avaliação SP 1.3: Identificar soluções alternativas SP 1.4: Selecionar métodos de avaliação SP 1.5: Avaliar soluções alternativas SP 1.6: Selecionar soluções
Resolução e Análise Causal (CAR)	5	SG 1: Determinar causas de resultados selecionados SG 2: Endereçar causas de resultados selecionados	SP 1.1: Selecionar resultados para análise SP 1.2: Analisar causas SP 2.1: Implementar propostas de ações SP 2.2 Avaliar os efeitos das ações implementadas SP 2.3 Registrar dados da análise causal

Fonte: CMMI-DEV, V1.3