# FUNDAMENTAÇÃO TEÓRICA

As áreas tratadas nesta fundamentação teórica são: modelos de maturidade de software, mais especificamente o CMMI, Gerenciamento de Requisitos (Requirements Management - REQM) e desenvolvimento ágil de software, uma vez que são os assuntos centrais desta pesquisa. Os trabalhos relacionados também são descritos neste capítulo.

**2.1 Modelo Integrado de Maturidade e Capacidade (CMMI)**

Segundo CMMI Institute (2015), O CMM surgiu na década de 1980 com foco em avaliação de riscos para o DoD, U.S Departament of Defence, ou Departamento de Defesa do Estados Unidos da América que desejava encontrar uma forma de avaliar os processos de desenvolvimento utilizados por seus fornecedores que concorriam em licitações sendo julgados pela avaliação da qualidade, custos e prazos nos projetos que saíam vencedores. Para viabilizar o desenvolvimento desse processo, o DoD constituiu o SEI (Software Engineering Institute) junto a universidade de Carnegie-Mellon. Atualmente o SEI é um dos maiores pesquisadores na área de engenharia de software.

Para o Software Engineering Institute (SEI, 2010), modelos de maturidade atuam na melhoria de processos em uma organização. Eles contém os elementos essenciais de processos efetivos para uma ou mais disciplinas e descrevem um caminho através de boas práticas reconhecidas pelo mercado alcançando processos disciplinados, buscando adquirir eficácia e melhoria na qualidade dos produtos de software.

A adoção dos níveis de maturidade do CMMI pelas empresas é um sucesso, cada vez mais empresas optam pelo CMMI e se submetem ao processo de avaliação e em seguida ao SCAMPI (Standard CMMI Appraisal Method for Process Improvement ou Método Padrão de Avaliação CMMI para Melhoria de Processo) CMMI. O CMMI Institute (2015) anunciou um aumento de 12% entre 2012 e 2014 de avaliações concluídas, foi um número recorde de 1626 avaliações só em 2014. Significou o terceiro ano consecutivo de recorde de crescimento ao redor do mundo.

Segundo o SEI (2010), o CMMI é um modelo de maturidade de melhoria de processo para o desenvolvimento de produtos e serviços. Ele utiliza das melhores práticas de mercado para endereçar atividades de manutenção e desenvolvimento. Estas cobrem o ciclo de vida, da concepção a entrega e manutenção do produto. Por esse estudo focar no CMMI para desenvolvimento (CMMI-DEV), daremos maior enfoque nessa questão. A proposta do SEI (2010) para o CMMI-DEV é fornecer um guia para aplicar as melhores práticas em uma organização de desenvolvimento de software. Elas focam nas atividades para desenvolver produtos e serviços de qualidade, a fim de atender às necessidades dos clientes e usuários finais, promovendo estrutura para gerar rastreabilidade e melhoria contínua de processo.

O CMMI propõe níveis de maturidade como diferencial competitivo para as empresas que atuam no segmento. As empresas que alcançam níveis superiores no modelo de maturidade conquistam maior credibilidade e, mais importante, agregam valor comprovado aos seus produtos. Niazi et al. (2010) afirma que a avaliação CMMI aumenta de forma significativa as chances de uma empresa conquistar novos contratos de desenvolvimento.

O SEI (2010) destaca que as 22 áreas de processo do CMMI representam um conjunto de práticas que, preferencialmente, devem ser implementadas de forma coletiva, pois dessa forma, irá atingir um conjunto considerado de importantes objetivos que por consequência terá impacto positivo. A aderência ao modelo se da pela representação contínua ou por estágio. A representação contínua trabalha com níveis de capacidade para uma área de processo específica. Essa abordagem permite maior flexibilidade para a empresa que poderá escolher o foco de seus esforços de melhoria de processo, pois poderá escolher a área mais crítica de acordo com sua necessidade e estratégia. Já na representação por estágio, as áreas e processo são organizadas em cinco níveis, conforme ilustrado no Quadro 1 (2).

**Quadro 1 (2)-**Níveis da representação por estágio do CMMI

|  |  |
| --- | --- |
| Nível 1 – Inicial | Geralmente processos ad hoc e caóticos. |
| Nível 2 – Gerenciado | Projetos da organização têm a garantia de que os processos são planejados e executados de acordo com uma política. |
| Nível 3 – Definido | Processos são bem caracterizados e entendidos, e são descritos em normas, procedimentos, ferramentas e métodos |
| Nível 4 - Quantitativamente Gerenciado | A organização e os projetos estabelecem objetivos quantitativos para a qualidade e desempenho de processos, utilizando-os como critérios na gestão de projetos. |
| Nível 5 – Em Otimização | A organização melhora continuamente seus processos com base em um entendimento quantitativo dos seus objetivos de negócios e necessidades de desempenho. |

**Fonte:**  Extraído de SEI (2010)

Segundo o SEI (2010) o CMMI adota a abordagem de objetivos específicos e genéricos para cada uma de suas 22 áreas de processos. Logo, um objetivo específico responde as especificidades de determinada área de processo descrevendo suas características únicas e o que é preciso atentar para cumprir com os requisitos da área, pois ele serve como um componente obrigatório nas avaliações para ajudar a determinar se uma área de processo é satisfeita. Diferente dos objetivos específicos que atendem uma área específica, os objetivos genéricos se aplicam a múltiplas áreas de processos. Eles descrevem as características que devem satisfazer os processos que implementam uma determinada área. Como com os objetivos específicos, os genéricos são obrigatórios no contexto de uma avaliação para determinar se uma dada área esta satisfeita.

Seguindo o objetivo deste estudo, abordaremos a área de processo Gerenciamento de Requisitos (Requirements Management – REQM). Esta representa uma das áreas chave entre as áreas de processo do CMMI, pois representa importância fundamental no ciclo de vida do produto.

**2.2 Gerenciamento de Requisitos (Requirements Management – REQM)**

O destaque para a importância dos requisitos é abordado por Summerville (1997), onde ele destaca que existem três fatores devem ser mantidos em mente ao desenvolver um documento de requisitos. Em primeiro lugar, o escritor deve investir bastante tempo e esforço em escrever requisitos porque os requisitos serão lidos várias vezes. Em segundo lugar, o escritor não deve presumir que leitor vai possuir o mesmo conhecimento de quem escreveu. Em terceiro lugar, escrever requisitos de forma clara e concisa, que não é uma tarefa fácil. Isto requer muito mais tempo e concentração para que se possa escrever um requisito com uma qualidade significativa. Segundo Summerville (1997), engenharia de requisitos é um processo que cobre todas as atividades de descoberta, documentação e manutenção de um grupo de requisitos para um sistema baseado em computador. Dean Leffingwell et al. Apresenta uma definição similar, no entanto não utiliza o termo engenharia de requisitos, e sim como processo de gerenciamento de requisitos, e independente das definições semânticas, destaca que o principal objetivo da engenharia de requisitos é desenvolver requisitos de sistema inequívocos e desejados.

O gerenciamento de requisitos no CMMI é apresentado no nível 2, conforme a representação por estágio. Nesta área de processo, segundo o SEI (2010), são destacadas ações necessárias a serem realizadas pelas organizações que desejarem adquirir maturidade exclusivamente nessa área considerando o modelo por estágio. O objetivo desta área em questão é gerenciar os requisitos dos produtos e componentes de produto do projeto, bem como, identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto. Ainda de acordo com o SEI (2010) o gerenciamento de requisitos do CMMI deverá atingir 1 objetivo específico: SG 1 Gerenciar Requisitos. Todos os objetivos específicos são desdobrados em práticas específicas, neste caso o SG 1 possui 5 práticas específicas, sendo elas as seguintes:

SP 1.1 - Obter um Entendimento dos Requisitos

SP 1.2 – Obter Comprometimento com os Requisitos

SP 1.3 – Gerenciar Mudanças de Requisitos;

SP 1.4 - Manter Rastreabilidade Bidirecional dos Requisitos;

SP 1.5 - Identificar Inconsistências entre Trabalho de Projeto e Requisitos.

As práticas específicas (Specific Practices – SP) precisam ser cumpridas, considerando suas especidades e subpráticas associadas, pois somente com isso é possível atingir os objetivos específicos das áreas de processo. No contexto deste trabalho, fizemos o mapeamento desses objetivos e práticas por meio das metodologias ágeis de desenvolvimento.

**Quadro 2 (2)-**Práticas específicas dos objetivos específicos de REQM

|  |  |
| --- | --- |
| SG1 | SP 1.1 – Obter Entendimento dos Requisitos;  SP 1.2 – Obter Comprometimento com os Requisitos;  SP 1.3 – Gerenciar Mudanças de Requisitos..  SP 1.4 – Manter Rastreabilidade Bidirecional dos Requisitos.  SP 1.5 – Identificar Inconsistências entre Trabalho de Projeto e Requisitos. |

**Fonte:** Extraído de SEI (2010)

Segundo SEI (2010), um objetivo específico descreve as características únicas que precisam ser apresentadas para satisfazer a área de processo. Ele é um componente obrigatório do modelo e é utilizado em avaliações para ajudar a determinar se uma área do processo é satisfeita. A prática específica é a descrição de uma atividade que é considerada importante para alcançar o objetivo específico associado. As práticas específicas descrevem as atividades que deverão resultar em realização dos objetivos específicos de uma área de processo, sendo um componente esperado do modelo.

**2.3 Desenvolvimento ágil de software**

Como evento formal de disseminação de uma cultura ágil, o manifesto ágil surgiu a partir de um encontro de profissionais experientes em um resort em Utah, EUA. De acordo com o Manifesto Ágil (2001), dezessete pessoas se reuniram e desse encontro gerou a Agile Alliance, emergindo, assim, o Manifesto para Desenvolvimento Ágil de Software, que propõe alternativas para a gestão tradicional de projetos. Essa abordagem está descobrindo maneiras melhores de desenvolver softwares e enfatiza quatro valores, conforme mostra o Quadro 3 (2).

**Quadro 3 (2)-** Valores do manifesto ágil

|  |
| --- |
| 1. indivíduos e interações mais que processos e ferramentas; 2. software em funcionamento mais que documentação abrangente; 3. colaboração com o cliente mais que negociação de contratos; 4. responder a mudanças mais que seguir um plano.   Ou seja, mesmo havendo valor nos itens à direita, valoriza-se mais os da esquerda. |

**Fonte:** Manifesto Ágil (2001)

O Manifesto Ágil (2001) foi definido com princípios fundamentais, que deram o direcionamento para o movimento ágil desde então. Estes estão listados no Quadro 4 (2). Ele cita os doze princípios que foram estabelecidos pelos autores do manifesto e que embasam e fundamentam o desenvolvimento ágil de software.

**Quadro 4 (2)-** Princípios do manifesto ágil

|  |
| --- |
| Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor. |
| Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adéquam a mudanças, para que o cliente possa tirar vantagens competitivas. |
| Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos. |
| Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto. |
| Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho. |
| O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara. |
| Software funcional é a medida primária de progresso. |
| Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes. |
| Contínua atenção à excelência técnica e bom design, aumenta a agilidade. |
| Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito. |
| As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis. |
| Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo. |

**Fonte:** Manifesto Ágil (2001)

A pesquisa realizada por Williams et al. (2012), que envolveu 326 profissionais agilistas e que tinha objetivo de descobrir a visão deles sobre os princípios ágeis, através de afirmações que trouxesse o interesse dos entrevistados em manifestar concordância ou não. O Quadro 5 (2) demonstrou o resultado, evidenciando a importância dos princípios diante da comunidade de profissionais experientes em desenvolvimento ágil. O conteúdo de outros, respondido por 20.7% dos entrevistados, é melhor representado pela seguinte afirmação: o propósito de qualquer princípio é fornecer uma fonte clara e simples de orientação e inspiração. Os princípios ágeis são importantes porque eles destilam os valores de agilidade de forma inteligentemente resumida. Ao analisá-los, considerando especificidades de implementação, é possível garantir que os processos diários estão servindo aos propósitos acordados.

**Quadro 5 (2)-**Valores dos princípios ágeis

|  |  |
| --- | --- |
| **Por que princípios ágeis são valiosos?** | **Percentual das respostas** |
| Porque todo time ágil escolhe práticas de desenvolvimento de software, mas, se eles querem ser ágil, eles devem escolher as práticas que estão alinhadas com os princípios. | 64.6% |
| Porque eles orientam novos times ágeis. | 48.8% |
| Outros. | 20.7% |
| Eles não são realmente importantes; o importante é usar as práticas ágeis. | 9.8% |
| Eles não são realmente importantes; o importante é olhar para o manifesto ágil. | 0% |
| Eles não são realmente importantes; ninguém olha para eles. | 0% |

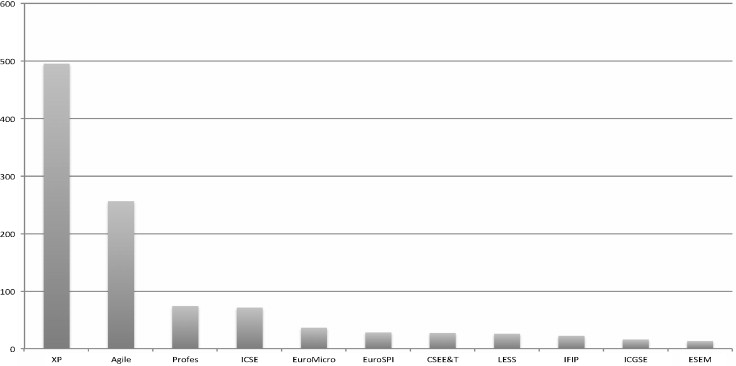
**Fonte:** Williams et al. (2012)

Segundo a VersionOne (2014), em um levantamento realizado com indivíduos de todo o mundo na indústria de desenvolvimento de software, 90% dos profissionais conhecem técnicas de desenvolvimento ágil há pelo menos um ano. Ainda nesse universo da pesquisa, 28% possuem larga experiência com 5 anos ou mais, 30% entre 3 e 4 anos e 32% entre 2 e 3 anos. Apenas 10% desses indivíduos tiveram contato com desenvolvimento ágil há menos de 6 meses. Ainda sobre as organizações, O resultado da pesquisa foi que 94% delas praticam ágil. Os principais benefícios elencados foram: A visibilidade dos resultados de mudanças de prioridades, a produtividade e a visibilidade do projeto.

Dingsøyr et al. (2012), alertou para o comportamento histórico que acontece com qualquer disciplina nova, quando nos primeiros anos de desenvolvimento ágil tivemos exuberâncias de alguns e ceticismo de muitos. Aderindo a graus diversos aos princípios do manifesto, apareceu uma série de metodologias, incluindo, mas não se restringe, o Extreme Programming (XP), Scrum, Lean Software Development (Lean), Feature-Driven Development (FDD) e a metodologia Crystal, dentre outros.

Dingsøyr et al. (2012) percebeu também que o movimento na comunidade acadêmica estava com um direcionamento significativo para o desenvolvimento ágil, quando desde sua concepção, em 2001, as quantidades de publicações científicas cresceu de forma significativa. Periódicos importantes passaram a dar espaço para esse momento científico, muitos países passaram a se envolver no tema do desenvolvimento ágil. A Figura 1 (2) mostra alguns números acerca das publicações em desenvolvimento de software ágil por conferências de 2001 até 2010, feitas na pesquisa de Dingsøyr et al. (2012).

**Figura 1 (2)-**Publicações em desenvolvimento de software por conferências



**Fonte:** Dingsøyr et al. (2012)

Como se pode ver na Figura 1 (2), que foi retirada do estudo de Dingsøyr et al. (2012), a maior parte das publicações se concentra em artigos de conferência. Nota-se que houve uma diminuição de publicações entre o período de 2009 e 2010 em artigos de conferência, enquanto nos artigos de periódico (journal) houve um leve aumento.

Segundo Dingsøyr et al. (2012), uma revisão sistemática de estudos empíricos publicados até 2005 apelou para um aumento tanto no número quanto na qualidade dos estudos. A análise também constatou que a maioria dos estudos era focada na metodologia XP e pouco em Scrum, que foi ganhando muita força na indústria. Além disso, a avaliação mostrou a necessidade urgente de mais estudos envolvendo equipes ágeis de desenvolvimento experientes, já que esses estudos foram focados em projetos que estavam apenas começando a usar métodos ágeis.

**2.4 Trabalhos relacionados**

Nesta seção será descrito os principais trabalhos que possuem relação com esta pesquisa, abordando os pontos em comuns e os pontos divergentes em comparação a presente pesquisa. Estão apresentadas três revisões sistemáticas da literatura que considerou-se muito relevante frente os estudos publicados sobre a temática, sendo elas a de DYBÅ et al. (2008), Selleri et al. (2014) e Chagas et al. (2014).

DYBÅ et al. (2008) revisou sistematicamente na literatura existente estudos empíricos que abordassem desenvolvimento ágil de software. Seu estudo é um clássico e abordou importantes conclusões em sua síntese, bem como a abordagem dada a aspectos humanos e sociais além de considerações importantes acerca de cultura organizacional no desenvolvimento ágil de software. O contraponto nesta revisão é que não há ênfase no estudo de modelo de maturidade relacionado ao desenvolvimento ágil de software. Nesta pesquisa foram selecionados 36 estudos empíricos.

Selleri et al. (2014) combinou CMMI desenvolvimento ágil em sua revisão sistemática da literatura, abordando discussões sobre benefícios e limitações do trabalho em conjunto das duas abordagens. O trabalho selecionou 81 estudos primários que abordavam CMMI e desenvolvimento ágil, destes 23 estudos abordaram gerenciamento de requisitos, estando entre as três áreas de processo mais mencionadas em sua pesquisa.

Na revisão sistemática de Chagas et al. (2014) a temática foi gerenciamento de projetos ágeis em organizações que utilizam, conjuntamente, métodos ágeis e CMMI. A revisão selecionou 34 estudos primários, tratando desde 2001 até o ano de 2013 inteiro. Concluiu que o desenvolvimento ágil conduzido de forma autônoma não são suficientes para garantir maturidade de processo e atender as exigências do modelo CMMI.

A diferença mais significativa entre este estudo e os demais referenciados anteriormente esta no destaque aos benefícios e limitações do gerenciamento de requisitos no desenvolvimento ágil no contexto do CMMI, assim como a apresentação das práticas ágeis mais recomendadas para aderência da área de processo citada com o modelo CMMI. A pesquisa de DYBÅ et al. (2008) não trata o CMMI. A pesquisa de Selleri et al. (2014) não teve como foco o gerenciamento de requisitos com CMMI e desenvolvimento ágil. Chagas et al (2014) também não deu ênfase ao gerenciamento de requisitos em sua pesquisa.

**2.5 Considerações**

Buscou-se com esse capítulo descrever os assuntos referente a temática da pesquisa. Primeiramente abordou-se os assuntos acerca do modelo de maturidade CMMI, então sobre a área de processo alvo do estudo, gerenciamento de requisitos no contexto do CMMI para, por fim, abordar os assuntos relacionados ao desenvolvimento ágil de software.

Os temas abordados foram avaliados como fundamentais para a fundamentação teórica dessa pesquisa. A adoção de modelo de maturidade é uma necessidade real para empresas que investem em melhorar seus processos e, percebe-se ainda que trata-se de um caminho obrigatório para quem pretende se manter competitivo diante do mercado. Foi apresentado o CMMI, que é parte do escopo desta pesquisa e representa um modelo consolidado no mercado. Os conceitos, benefícios e importância do gerenciamento de requisitos de forma a contextualizar a área de processo no contexto dos demais assuntos relacionados a temática também foram bastante explorados. A definição de desenvolvimento ágil, os valores e princípios destacados no manifesto ágil, a crescente produção científica enfatizando assim sua relevância para o mercado. A seção de trabalhos relacionados apresentou três revisões sistemáticas, apresentando as correlações e divergências entre as pesquisas com este estudo.