

# Aplicando um Modelo de Maturidade no Projeto CIn

Luckerson Cruz<sup>1</sup>, Alexandre Vasconcelos<sup>1</sup>

<sup>1</sup>Centro de Informática – Universidade Federal de Pernambuco  
Caixa Postal 7851 – 50732-970 Recife, PE

lcc2@cin.ufpe.br, amlv@cin.ufpe.br

**Abstract.** *The Capability Maturity Model (CMM) has a long and successful history of being used by software organizations for assessing and improving their software process. However, one of the known limitations of the CMM is that it does not sufficiently address the software test process. In this way, this article presents a way for providing enough visibility into the test process maturity of an organisation using the Testing Maturity Model (TMM) for complementing both CMM and CMM Integration (CMMI) by specifically addressing important issues to test managers, test specialists and software quality assurance staff.*

**Resumo.** *O Capability Maturity Model (CMM) tem uma longa e bem-sucedida história de uso por organizações de software na avaliação e melhoria de seus processos de software. Contudo, uma de suas limitações conhecidas é a de não lidar suficientemente com o processo de teste de software. Dentro desse contexto, o presente artigo tratará de uma forma de visibilidade da maturidade do processo de teste de uma organização utilizando o Testing Maturity Model (TMM) tendo o intuito de complementar o CMM e o CMM Integration (CMMI) destinando questões importantes a gerentes de teste, especialistas em testes e à equipe de garantia de qualidade de software.*

## 1. Introdução

O *Capability Maturity Model* (CMM) desenvolvido pelo *Software Engineering Institute* (SEI) tem uma longa e bem-sucedida história de uso por organizações de software na avaliação e melhoria de seus processos de software. Outra forte tendência - o desenvolvimento de software terceirizado - também contribuiu para o elevado uso do CMM. Várias empresas usam esse *framework* com o intuito de selecionar seus parceiros terceirizados. Da mesma forma, companhias de desenvolvimento de software terceirizadas usam a certificação CMM como uma ferramenta de marketing para promover seus serviços e concorrer a contratos.

Uma das limitações conhecidas do CMM é que ele não lida suficientemente com o processo de teste de software. As poucas práticas relacionadas a testes definidas pela *key process area* (KPA) Engenharia de Produto de Software no Nível 3 do CMM não fornece visibilidade suficiente da capacidade do processo de teste, nem podem ser usadas como um *framework* para melhoria do processo de teste.

Para preencher essa lacuna, uma série de modelos de maturidade de teste emergiram nos anos 90. Alguns deles foram projetados para ser usados em conjunção com o CMM [Gelperin and Hayashi, 1996] [Burnstein, 2003]. Contudo, nenhum desses modelos conseguiu muita aceitação, o que motiva a continuidade de pesquisas nessa área.

## 2. Visão Geral dos Modelos de Processo de Teste Existentes

Uma interessante análise e comparação entre alguns dos atuais modelos de processo de teste pode ser encontrada em [Weatheill, 2001]. Os três modelos seguintes possuem uma descrição mais detalhada e serão o escopo de uma análise inicial:

**Test Process Improvement (TPI)** Tal modelo é bem documentado [Koomen and Pol., 1999] e apresenta uma série de áreas chave e suas práticas que são organizadas por níveis de processo de maturidade de teste. Contudo, o TPI não tem a intenção de ser um modelo compatível com CMM. Sua estrutura é diferente do CMM, o que o torna difícil de ser aplicado por organizações que já adotaram CMM como um *framework* padrão para melhoria de processo.

**Software Evaluation and Test Key Process Area** Foi definido e proposto por Richard Bender como uma área de processo adicional ao CMM Nível 2. Esse modelo é completamente compatível com a estrutura CMM e tem detalhes suficientes para sua implementação. É uma ferramenta bastante útil que pode auxiliar organizações a avaliarem seu atual processo e estabelecer práticas básicas de teste. Contudo, tal modelo tem duas limitações. Primeiramente, faltam práticas relacionadas a projeto de *testware*, tais como planos de teste, projetos de teste e especificações de casos de teste. Em segundo lugar, ele não pode ser, e não foi projetado para ser, usado acima do Nível 2 do CMM.

**Testing Maturity Model (TMM)** É um modelo bastante abrangente e bem documentado [Paulk et al., 1995] desenvolvido por um grupo do Instituto de tecnologia de Illinois. O Modelo de Maturidade de Testes tem o intuito de complementar o CMM e o CMMI exclusivamente endereçando questões importantes a gerentes de teste, especialistas em testes e à equipe de garantia de qualidade de software. O TMM contém uma série de níveis de maturidade através dos quais uma organização pode progredir em direção à maturidade do processo de testes, uma série de práticas recomendadas em cada nível de maturidade e um modelo de avaliação que permite às organizações avaliarem e aperfeiçoarem seu processo de testes.

Assim, devido à sua facilidade de compreensão e uso, permitindo as organizações executarem auto-avaliações, à capacidade de fornecer um ponto de partida do atual estado de teste e um guia para melhorias e à capacidade de ser utilizado em conjunto com o CMM, trataremos exclusivamente da aplicação do TMM a partir desse ponto.

## 3. Plano para Desenvolvimento do Modelo

O TMM será utilizado:

- por um time de avaliação interna para identificar o atual estado do processo de testes;
- pela alta gerência para iniciar um programa de aprimoramento do processo de testes;
- por times de desenvolvimento na melhoria do processo de testes;
- por usuários e clientes para definir seus papéis no processo de testes.

Há vários modelos para avaliação de processos, dentre eles CMM [Paulk et al., 1995], ISO 9001 [Coallier, 1994] e SPICE [Paulk and Konrad, 1994]. Contudo, nenhum desses modelos tem como foco primário o processo de testes. O CMM, que possui extensivo uso atualmente, não foca adequadamente conteúdos relacionados a testes. Por exemplo, no CMM:

- o conceito de maturidade de testes não é abordado;

- não há inclusão adequada de práticas de testes como um mecanismo de melhoria do processo;
- itens relacionados a testes não são adequadamente abordados nas áreas-chaves do processo;
- itens relacionados a qualidade tais como testabilidade, critério de adequação de testes e planejamento de testes não são satisfatoriamente abordados.

Devido à importância do papel dos testes no processo de software e na qualidade do produto e às limitações dos modelos de avaliação de processos existentes, um grupo do Instituto de tecnologia de Illinois focou sua pesquisa no TMM. Foram propostos os seguintes componentes para dar suporte aos objetivos de desenvolvimento do TMM:

- Uma série de níveis que define uma hierarquia de maturidade de testes. Cada nível representa um estágio na evolução para um processo de testes maduro. Um passo em direção a um nível superior implica que práticas dos níveis inferiores continuam em execução;
- Uma série de objetivos de maturidade para cada nível (exceto Nível 1), e as atividades, tarefas e responsabilidades necessárias para apoiá-los. As organizações alcançarão a maturidade de testes focando nos objetivos definidos para cada nível;
- Um modelo de avaliação composto por três componentes: uma série de questões relacionadas a objetivos de maturidade projetadas para avaliar a maturidade do processo de testes, um programa de treinamento projetado para selecionar e instruir o time de avaliação que deve conduzir a avaliação da maturidade e um método de avaliação que permite que uma organização faça uma auto-avaliação baseada em respostas de questionários e dados de entrevistas.

Os requisitos gerais para o desenvolvimento do modelo são os seguintes: o modelo deve ser aceito pela comunidade de desenvolvimento de software e ser baseado em práticas e princípios de engenharia de software pré-estabelecidos. Ele deve permitir que o desenvolvimento da maturidade do processo de testes seja gradativamente estruturado em fases que sigam o processo de evolução natural. Também deve haver um mecanismo de suporte à avaliação e melhoria do processo de teste. Atualmente quatro fontes servem como entradas principais ao desenvolvimento do TMM: o Capability Maturity Model [Paulk et al., 1995], o Modelo de Testes Evolucionário [Gelperin and Hetzel, 1988] as Fases Progressivas do Modelo Mental de Testadores [Beizer, 1990] e atuais práticas de teste na indústria.

### 3.1. Capability Maturity Model

O CMM é um abrangente modelo de avaliação e aperfeiçoamento de processos desenvolvido pelo *Software Engineering Institute* (SEI) que foi amplamente aceito e aplicado à indústria de software [Paulk et al., 1995]. Pesquisas no SEI e em seus parceiros resultaram no refinamento dos conceitos fundamentais do processo de maturidade. O apoio ao desenvolvimento do processo de maturidade é obtido através da hierarquia de níveis no CMM. Cada nível é composto de uma série de objetivos de processo, e cada objetivo estabiliza um importante componente do processo de software. Cada nível fornece a base para avançar ao próximo nível. Todos os níveis no CMM tem uma estrutura interna (exceto o Nível 1) que consiste em áreas de processos-chave organizadas por características comuns. As áreas de processos-chave indicam onde a organização deve concentrar esforços para alcançar maturidade em um dado nível. As características em comum especificam as principais práticas que quando alcançadas dão apoio às metas exigidas das áreas de processos-chave.

Várias características do CMM foram aplicadas ao desenvolvimento do TMM. Assim como o CMM, o TMM usa o conceito de níveis de maturidade como um roteiro para

a avaliação e melhoria do processo. Os níveis de TMM tem um suporte estrutural similar ao dos níveis em CMM. Foi adicionado um componente chamado "visões críticas" de forma que os principais participantes do processo de teste sejam incluídos no crescimento da maturidade do processo. Ambos os modelos exigem que todas as habilidades em cada nível inferior sejam incluídas nos níveis seguintes. Para apoiar o processo de auto-avaliação, o TMM também usa a técnica de avaliação de questionários e entrevistas do CMM. O TMM não é só estruturalmente similar ao CMM, deve ser visto e utilizado como um complemento a ele, desde que processos de teste maduros dependem de processos de maturidade gerais. Foram estabelecidas relações entre os dois modelos que refletem essa dependência. As seções subsequentes mostram como os níveis e objetivos de ambos os modelos se sobrepõem.

### **3.2. Estágios Evolucionários de Testes**

Um Modelo de Maturidade de Testes deveria refletir o padrão evolucionário do desenvolvimento da maturidade do processo de teste documentado nas últimas décadas. Foi utilizado o modelo histórico fornecido em um *paper* fundamental de Gelperin e Hetzel como a base para a diferenciação de níveis no TMM [Gelperin and Hetzel, 1988]. Tal modelo descreve fases e objetivos de teste desde a década de 50. O período inicial do modelo é descrito como "Orientado a Remoção de Defeitos". Durante aquele período a maioria das organizações de desenvolvimento de software não tinha claramente diferenciado testes e remoção de defeitos. Teste não era vista como uma atividade que ajudava a remoção de defeitos. No período "Orientado a Propaganda", a meta primária dos testes era demonstrar que o software satisfazia o que era anunciado pela especificação. Teste e remoção de erros ainda eram ligados aos esforços de detectar, localizar e corrigir falhas. O período "Orientado a Destruição" focava o teste como uma atividade de detectar falhas de implementação. Remoção de erros era uma série de atividades separadas necessárias para localizar e corrigir defeitos. No período "Orientado a Avaliação", o teste se tornou uma atividade que foi integrada ao ciclo de vida do software. O valor de atividades de revisão foi reconhecido. A visão dos testes foi ampliada e seus objetivos eram detectar erros em requisitos, projeto e implementação. O modelo histórico de Gelperin-Hetzel culminou pelo que eles denominam de período "Orientado a Prevenção", que reflete o Nível 5 de otimização do CMM e do TMM. O escopo dos testes é amplamente definido e inclui revisão de artefatos. A meta primária dos testes é prevenir falhas de requisitos, projeto e implementação. Atividades de revisão agora apóiam planejamento de testes, projeto de testes e avaliação do produto.

### **3.3. Práticas da Indústria e o Modelo Beizer**

Uma pesquisa das atuais práticas industriais também foi uma das entradas na definição dos níveis TMM. Ela ilustra os melhores e piores ambientes de testes na indústria do software e também permitiu a extração de um critério realista na avaliação e melhoria das práticas de teste. Foram também incorporados ao TMM conceitos associados ao modelo evolucionário de Beizer do processo de pensamento dos testadores, que em muitas formas compara-se ao padrão de desenvolvimento da maturidade de testes descrita pelo modelo Gelperin-Hetzel [Gelperin and Hetzel, 1988] [Beizer, 1990]. Sua influência no desenvolvimento do TMM é baseada na premissa que uma organização de testes madura é construída com a habilidade, dom, talento, aptidão e atitudes dos indivíduos que nela trabalham.

## **4. Definindo o Processo de Maturidade de Testes**

Paulk, Weber, et al. compararam e contrastaram as características comportamentais de organizações de software maduras e imaturas e descreveram uma série de conceitos fun-

damentais da maturidade de processos de software [Paulk et al., 1995]. Foi usado essa série básica de conceitos para definir a maturidade do processo de testes. Extrapolando Paulk, um processo de testes maduro é gerenciado, medido, monitorado e efetivo. Na descrição da maturidade do processo de software o termo "gerenciado" é interpretado no mais amplo sentido incluindo planejamento, recrutamento de pessoal, supervisionar, controlar e organizar componentes [Thayer, 1990]. Além disso, um processo de testes maduro reflete-se a toda a instituição, é apoiada pela administração e é parte da cultura organizacional. Finalmente, o processo de testes maduro é bem compreendido e tem a capacidade de desenvolvimento contínuo e aperfeiçoamento. Os atributos de um processo de testes maduro são descritos abaixo, cada um apoiando um ou mais critérios de maturidade do processo descrito no CMM. Acredita-se que um processo de testes maduro tem as seguintes características:

- **Uma série de políticas de testes definidas.** Há uma série de políticas de testes documentadas e bem definidas que são aplicadas por toda a organização. As políticas de teste são apoiadas pela alta gerência, institucionalizadas e integradas na cultura organizacional;
- **Um processo de planejamento de testes.** Há um processo de planejamento de testes documentado e bem definido utilizado por toda a organização que permita a especificação de objetivos de teste e metas, alocação de recursos de teste, projetos de teste, casos de teste, agenda de testes, custo de testes e tarefas de teste. Planos de teste refletem os riscos de falhas e alocação de tempo e recursos;
- **Um ciclo de vida do teste.** O processo de teste inclui atividades de testes baseada em execução. Há um ciclo de vida de testes bem definido com uma série de fases e atividades integradas ao ciclo de vida do software. O ciclo de vida dos testes abrange todas as atividades de teste, por exemplo: planejamento de teste, revisões, projeto de teste, implementação de software relacionado a testes e manutenção de artefatos relacionados a teste. É aplicado a todos os projetos;
- **Um grupo de testes.** Há um grupo de testes independente. A posição do testador é definida e apoiada pela alta gerência. Instruções e oportunidades de treinamento devem existir para educar e motivar a equipe de teste;
- **Um grupo de aprimoramento do processo de teste.** Há um grupo dedicado à melhoria do processo. Eles podem ser parte de um grupo de aprimoramento geral do processo, um grupo de garantia de qualidade do software ou um componente do grupo de testes. Levando-se em conta que o processo de teste é bem definido e mensurado, o grupo de aprimoramento de testes pode manifestar liderança para refinar o processo, aplicar técnicas de melhoria incrementais e avaliar seus impactos;
- **Uma série de métricas relacionadas a testes.** A organização deve ter um programa de métricas. Uma série de métricas relacionadas a testes deve ser definida, dados devem ser coletados e analisados com suporte automatizado. As métricas são utilizadas para dar suporte a ações necessárias à melhoria do processo;
- **Ferramentas e equipamento.** Ferramentas apropriadas devem estar disponíveis para auxiliar o grupo de teste com as tarefas de teste e para coletar e analisar dados relacionados a testes. O grupo de melhoria do processo de testes deve determinar a avaliação de ferramentas potenciais, fiscalizar questões relacionadas a transferência de tecnologia associadas à integração de ferramentas no ambiente organizacional;
- **Controle e acompanhamento.** O processo de teste é monitorado e controlado pelos gerentes de teste para acompanhar o progresso, agir quando problemas ocorrerem e avaliar performance e capacidade. Técnicas quantitativas são usadas para analisar o processo de teste e determinar efetividade e capacidade do processo de

teste;

- **Controle de qualidade do produto.** Métodos estatísticos devem ser usados para seguir padrões de qualidade. Critérios para "Parada de Teste" são quantitativos. A qualidade do produto deve ser monitorada, defeitos devem ser rastreados e uma análise de causas deve ser aplicada para prevenção de defeitos.

## **5. Componentes do TMM**

O TMM tem dois componentes principais. Cada um deles tem vários subcomponentes, que são descritos a seguir.

### **5.1. A Série de Níveis**

O TMM tem uma série de níveis bem-definidos. Cada um deles dita uma posição na hierarquia de maturidade de teste. As características de cada nível são descritas em termos de capacidade de teste e objetivos da organização. Cada nível, excetuando-se o Nível 1, tem uma estrutura. Uma estrutura consiste em:

- Uma série de objetivos de maturidade. Os objetivos de maturidade identificam melhorias que devem ser cumpridas para alcançar maturidade naquele nível;
- Sub-objetivos de apoio. Definem o escopo, fronteiras e características necessárias para um nível particular;
- Atividades, tarefas e responsabilidades. São necessárias para alcançar os objetivos associados a cada nível.

### **5.2. O Modelo de Avaliação**

O modelo de avaliação é composto pelos seguintes itens:

#### **5.2.1. O Questionário**

Contém perguntas que são planejadas para determinar um nível da maturidade de teste. As questões são relacionadas aos objetivos de maturidade e fluxo de processo descritos em cada nível. Eles ajudam a determinar até qual extensão a organização tem mecanismos para alcançar tais objetivos e resolver problemas de maturidade.

#### **5.2.2. O Procedimento de Avaliação**

Avaliar um processo de testes para determinar seu nível de maturidade depende de entrevistas a membros do corpo de funcionários e de respostas do questionário. O procedimento de avaliação fornece guias ao time de avaliadores sobre quem entrevistar e sobre como coletar, organizar, analisar e interpretar os dados coletados dos questionários e das entrevistas pessoais. Um procedimento para determinar níveis de maturidade a partir de resultados analisados faz parte do procedimento de avaliação. Um mecanismo de relatório dá suporte à disseminação de resultados e recomendações para melhorias no processo de teste, com itens de alta prioridade identificados.

#### **5.2.3. O Procedimento de Treinamento e Seleção do Time**

O procedimento de avaliação é realizado por um time de avaliação treinado, interno à organização sendo avaliada. Um procedimento de treinamento instrui o corpo de funcionários envolvido na avaliação do processo de teste usando TMM. O programa de

treinamento inclui instruções sobre conceitos de maturidade, modelos de maturidade, os princípios sobre os quais os níveis do TMM foram idealizados, a estrutura interna dos níveis, técnicas de entrevista e um profundo entendimento do procedimento de avaliação.

### **5.3. O Modelo Estrutural: Uma Estrutura para os Níveis**

O TMM é caracterizado por cinco níveis de maturidade composto por um *framework* de objetivos, sub-objetivos, atividades, tarefas e responsabilidades. Cada nível indica uma maturidade de teste específica. Excetuando-se o Nível 1, vários objetivos de maturidade, que identificam as áreas-chave do processo, são apontados em cada nível. Os objetivos de maturidade idetificam metas de melhorias de teste que devem ser satisfeitas para alcançar maturidade naquele nível. Para se posicionar em determinado nível, uma organização deve satisfazer os objetivos de maturidade naquele nível. Esse requisito é refletido no questionário para avaliação de processo. Cada objetivo de maturidade é auxiliado por um ou mais sub-objetivos de maturidade. Os sub-objetivos de maturidade especificam objetivos menos abstratos, e eles definem o escopo, fronteiras e realizações necessárias para um nível em particular. Os dubobjetivos de maturidade são alcançados através de um grupo de atividades e tarefas com responsabilidades. As atividades e tarefas com responsabilidades direcionam questões de implementação e adaptação organizacional em um dado nível. Atividades e tarefas são definidas em termos de ações que devem ser executadas em determinado nível para melhorar a capacidade de teste; elas são ligadas a compromissos organizacionais. Responsabilidades são designadas para essas atividades e tarefas a três grupos que acredita-se representarem os principais participantes no processo de teste: gerentes, desenvolvedores/testadores e usuários/clientes. No modelo eles são tratados como "as três visões críticas". A definição desses papéis é essencialno desenvolvimento de uma estrutura de maturidade. A visão do gerente envolve compromisso e habilidade de executar atividades e tarefas relacionadas a melhoria da capacidade de teste. A visão do desenvolvedor e testador abrange atividades e tarefas técnicas que, quando aplicadas, constituem práticas de teste maduras. A visão dos usuários e clientes é definida como uma visão de suporte ou cooperação. Os desenvolvedores e testadores trabalham com o cliente e grupos de usuários em atividades relacionadas a qualidade e tarefas que relacionam-se a necessidades orientadas ao usuário. O foco é em solicitar suporte do cliente/usuário, concenso e participação em atividades como análise de requisitos, testes de usabilidade e planejamento de testes de aceitação.

### **5.4. A Estrutura Operacional do TMM**

Os cinco níveis que formam a estrutura do TMM são os seguintes:

- Nível 1: Inicial;
- Nível 2: Definição de Fase;
- Nível 3: Integração;
- Nível 4: Gerenciamento e Métricas;
- Nível 5: Otimização, Prevenção de Defeitos e Controle de Qualidade.

Esses níveis refletem a evolução do processo de teste partindo de um que é caótico e indefinido para um que é medido e otimizável. Isso é compatível com toda a abrangência de hierarquia de desenvolvimento de maturidade do CMM. Os níveis também refletem os cinco períodos de evolução do modelo de teste proposto por [Gelperin and Hetzel, 1988], anteriormente discutido na Seção 3.2. A correspondência pode ser brevemente descrita da seguinte forma: Nível 1 do TMM é relacionado ao período "Orientado a Remoção de Defeitos"do modelo Gelperin/Hetzel; O Nível 2, Definição de Fase, tem características tanto do período "Orientado a Propaganda"quanto do período "Orientado a Destruição"; O Nível 3, Integração, é relacionado ao período "Orientado a Avaliação"de Gelperin e

Hatzel; Os níveis 4 e 5 refletem tendências e práticas mostradas no período "Orientado a Prevenção". Além disso, os níveis de TMM comparam-se a modelo de 5 fases de Beizer do aperfeiçoamento de maturidade do testador. Beizer descreve uma progressão relacional que começa com o modelo Fase 0 no qual o testador é incapaz de distinguir entre remoção de defeitos e teste. A progressão culmina na Fase 4, na qual os testadores se vêem como profissionais disciplinados cuja tarefa é dar suporte ao desenvolvimento de software altamente testável e de baixo risco.

#### **5.4.1. Nível 1: Inicial**

Teste é um processo caótico. Testes são desenvolvidos *ad hoc* após feita a codificação. Testes e remoção de defeitos são entrelaçados com o intuito de retirar *bugs* do software. O objetivo do teste é mostrar que o software funciona. Produtos de software são lançados sem garantia de qualidade. Há falta de recursos, ferramentas e equipe treinada. Esse tipo de organização estaria no nível 1 do CMM. Não há metas de maturidade nesse nível.

#### **5.4.2. Nível 2: Definição de Fase**

As atividades de teste são separadas da remoção de defeitos. Teste é a fase posterior à codificação. É uma atividade planejada, contudo, o planejamento de testes no nível 2 pode ocorrer após a codificação por razões relacionadas à imaturidade do processo de teste. Por exemplo, no nível 2 há a percepção que todo o teste é baseado em execução e dependente do código, e conseqüentemente só deve ser planejado quando o código estiver completo. O objetivo primário dos testes nesse nível de maturidade é mostrar que o software está de acordo com as especificações [Gelperin and Hetzel, 1988]. Métodos e técnicas básicas de testes são utilizados. Muitos problemas de qualidade ocorrem nesse nível já que o planejamento dos testes só ocorrerá numa próxima fase do ciclo de desenvolvimento de software. Além disso, os defeitos se propagam no código vindo das fases de requisitos e projeto, por não haver programas de revisão que cuidem desses problemas. Testes executados em execução pós-código ainda são considerados a atividade primária de testes.

#### **5.4.3. Nível 3: Integração**

Teste não é mais a fase que segue a codificação; ela é integrada em todo o ciclo de vida do software. Diferentemente do Nível 2, o planejamento dos testes começa na fase de requisitos e continua por todo o ciclo. Objetivos de teste são estabelecidos de acordo com os requisitos baseados em necessidades dos clientes e usuários e são usados no projeto de casos de teste e critérios de sucesso. Há uma organização de teste, e o teste é reconhecido como uma atividade profissional. Há uma organização de treinamentos técnicos com foco em teste. Ferramentas básicas dão suporte a atividades chave de teste. Apesar de as organizações nesse nível começarem a entender o importante papel das revisões no controle da qualidade, não há um programa de revisão formal, e as revisões ainda não ocorrem em todo o ciclo de vida. Um programa de métricas de testes ainda não foi estabelecido para qualificar atributos do processo e do produto.

#### **5.4.4. Nível 4: Gerenciamento e Métricas**

Teste é um processo medido e quantificado. Revisões em todas as fases do processo de desenvolvimento são agora reconhecidas como atividades de teste e controle da quali-

dade. Produtos de software são testados em atributos de qualidade tais como confiabilidade, usabilidade e manutenibilidade. Casos de teste de todos os projetos são coletados e gravados num banco de dados para reuso de casos de teste e testes de regressão. Defeitos são gravados e associados a um nível de severidade. Deficiências no processo de teste agora são frequentemente devido à falta de uma filosofia de prevenção de defeitos e a deficiências no suporte automatizado de coleta, análise e disseminação de métricas relacionadas a teste.

#### **5.4.5. Nível 5: Otimização, Prevenção de Defeitos e Controle de Qualidade**

Devido à estrutura fornecida pelo alcance dos objetivos de qualidade dos Níveis 1 ao 4 do TMM, o processo de teste é agora dito ser definido, com custos gerenciados e efetividade monitorada. No nível 5 há mecanismos que regulam e continuamente melhoram os testes. A prevenção de defeitos e o controle de qualidade são praticados. O processo de testes é conduzido por amostragem estatística, medidas dos níveis de confiança e segurança. Há um procedimento estabelecido para selecionar e avaliar ferramentas de teste. Ferramentas automatizadas dão suporte total à execução e re-execução dos casos de teste, fornecendo suporte a projetos de caso de teste, manutenção de itens relacionados a teste, análise e coleta de falhas e à aplicação, análise e coleta de métricas relacionadas a teste.

### **5.5. Objetivos de Maturidade nos Níveis do TMM**

A estrutura operacional do TMM fornece uma sequência de níveis hierárquicos que contém os objetivos de maturidade, sub-objetivos, atividades, tarefas, e responsabilidades que definem a capacidade de testes de uma organização em um nível particular. Eles identificam as áreas nas quais uma organização deve focar para melhorar seu processo de testes. Abaixo serão descritos os objetivos de maturidade para todos os níveis exceto o Nível 1, que não tem objetivos de maturidade.

#### **5.5.1. Nível 2 - Definição de Fase**

No Nível 2 do TMM, uma organização definiu uma fase de testes posterior à codificação no ciclo de vida do software. A fase é planejada e pode ser repetida em todos os projetos de software. É separada da remoção de defeitos, que é uma atividade não planejada. Seguem abaixo os objetivos de maturidade do nível 2.

- 1. Desenvolver Objetivos de Teste e Remoção de Defeitos.** Isso significa que uma organização deve claramente distinguir os processos de teste e remoção de defeitos. Os objetivos, tarefas, atividades e ferramentas para cada uma deve ser identificada. responsabilidades para cada uma delas devem ser determinadas. A gerência deve desenvolver planos e políticas para acomodar e institucionalizar ambos os processos. A separação desses dois processos é essencial para o crescimento da maturidade de teste, já que elas têm diferentes objetivos, métodos e psicologia. Testes nesse nível agora são uma atividade planejada, dessa forma pode ser gerenciada enquanto que a remoção de defeitos não pode. Sub-objetivos de maturidade que dão suporte a esse objetivo incluem:

- A organização deve formar um comitê de teste e remoção de defeitos com suporte e capital;
- O comitê deve desenvolver e listar objetivos de teste;
- O comitê deve desenvolver e listar objetivos de remoção de defeitos;
- Os objetivos de teste e remoção de defeitos documentados devem ser distribuídos a todos os gerentes de projeto e desenvolvedores;

- Objetivos do teste devem ser refletidos nos planos de teste.
2. **Iniciar um Processo de Planejamento de Teste.** Planejar é essencial para um processo que deve ser repetível, definido e gerenciado. Planejamento de testes envolve expressar objetivos, analisar riscos, traçar estratégias e desenvolver especificações de projeto de teste. Além disso, o plano de teste deve definir a alocação de recursos e as responsabilidades dos testes nos níveis unitário, de integração, de sistema e de aceitação. Subobjetivos de maturidade que dão suporte a esse objetivo incluem:
    - Um comitê de planejamento de testes, com recursos, deve ser estabelecido por toda a organização;
    - Uma estrutura para política de planejamento de testes para toda a organização deve ser estabelecida e apoiada pela gerência;
    - Um modelo de plano de testes deve ser desenvolvido, registrado e distribuído para gerentes de projeto e desenvolvedores;
    - Ferramentas básicas de planejamento devem ser avaliadas, recomendadas e adquiridas. O uso deve ser apoiado pela gerência.
  3. **Institucionalizar Métodos e Técnicas Básicas de Teste.** Para aprimorar a capacidade de teste do processo, técnicas e métodos de teste básicos devem ser aplicados na organização. Como e quando essas técnicas e métodos devem ser aplicados e qualquer suporte básico às ferramentas devem ser claramente especificados. Alguns exemplos de métodos e técnicas básicas são estratégias de teste caixa-preta e caixa-branca, uso de matriz de validação de requisitos e divisão de testes baseados em execução em sub-fases tais como testes unitários, de integração, de sistema e de aceitação. Subobjetivos de maturidade que dão suporte a esse objetivo incluem:
    - Um grupo de tecnologia de teste deve ser formado para estudar, avaliar e recomendar uma série de técnicas e métodos básicos de teste, e recomendar uma série de ferramentas simples para suportá-las;
    - A gerência deve instituir uma série de políticas que assegurem que as técnicas e métodos recomendados são consistentemente aplicados por toda a organização.

### 5.5.2. Nível 3 - Integração

esse nível de maturidade crítico é essencial à construção de produtos de software com qualidade. Nesse nível, a fase de testes não é mais somente a fase de testes que segue a codificação. Em vez disso ela é expandida numa série de atividades bem definidas que são integradas ao ciclo de vida do software. Todas as fases do ciclo de vida têm atividades de testes associadas. Nesse nível a gerência apoia a formação e o treinamento de grupos especialistas de testes de software responsáveis pelos testes. Abaixo seguem os objetivos do Nível 3 de maturidade:

1. **Estabelecer uma Organização de Testes de Software.** Uma organização de testes de software deve ser criada para identificar um grupo de pessoas que são responsáveis por testar. Levando-se em conta que os testes, em seu mais completo sentido, têm uma grande influência na qualidade do produto e que os testes consistem em atividades complexas geralmente executadas com cronogramas apertados e alta pressão, a gerência percebe que é necessário ter um grupo de especialistas bem treinado e dedicado responsável por esse processo. O grupo de testes é responsável por planejamento de testes, execução e registro de testes, padrões associados a testes, métricas de teste, o banco de dados dos testes, reuso de testes e avaliação. Sub-objetivos de maturidade que dão suporte a esse objetivo incluem:

- Um grupo de testes deve ser estabelecido com liderança, suporte e recursos da alta gerência;
  - Papéis e responsabilidades devem ser definidas para o grupo de testes.
2. **Estabelecer um Programa de Treinamento Técnico.** Um programa de treinamento técnico assegura que uma equipe experiente está disponível para o grupo de testes. Os testadores devem ser propriamente treinados de forma a executar suas funções eficiente e efetivamente. No nível 3 do TMM a equipe é treinada em planejamento de testes, métodos de teste, padrões, técnicas e ferramentas. Sub-objetivos de maturidade que dão suporte a esse objetivo incluem:
    - A gerência deve estabelecer um programa de treinamento organizacional, fornecendo recursos e suporte;
    - Objetivos e planos de treinamento devem ser desenvolvidos;
    - Um grupo de treinamento interno deve ser estabelecido com ferramentas, meios e materiais disponíveis.
  3. **Integrar Testes ao Ciclo de Vida do Software.** Gerência e equipe técnica agora reconhecem que a maturidade do processo de testes e a qualidade do produto de software exigem que as atividades de teste sejam conduzidas paralelamente a todas as fases do ciclo de vida. Sub-objetivos de maturidade que apóiam esse objetivo incluem:
    - A fase de testes deve ser particionada em subfases que podem ser integradas ao ciclo de vida do software;
    - Definições-padrão devem ser desenvolvidas para produtos de trabalho relacionados a testes.
  4. **Controlar e Monitorar o Processo de testes.** Quando as atividades realizadas desviarem das planejadas, a gerência pode tomar ações efetivas para corrigir os desvios e retornarem aos objetivos do plano de teste. O progresso do teste é determinado pela comparação dos atuais produtos de teste, esforço, custo e cronograma com os do plano. Sub-objetivos de maturidade que apóiam esse objetivo incluem:
    - A organização deve desenvolver mecanismos e políticas para controlar e monitorar o processo de teste;
    - Uma série de medidas básicas relacionadas a teste deve ser definida, registrada e distribuída;
    - Uma série de ações corretivas e planos de contingência devem ser desenvolvidos, gravados e documentados para uso quando os testes desviarem significativamente do que foi planejado.

### 5.5.3. Nível 4 - Gerenciamento e Métricas

O principal foco nesse nível é ampliar o conceito de "atividade de teste" e precisamente mensurar o processo de teste. Controlar e monitorar funções pode ser agora completamente auxiliado por um programa de métricas de teste estabelecido. A definição de atividade de teste agora inclui revisões e inspeções por todas as fases do ciclo de teste. Todos os planos de teste, projeto de testes e procedimentos de teste são revisados. Seguem abaixo os objetivos de maturidade no Nível 4.

1. **Estabelecer uma Programa de Revisão em Toda a Organização.** No Nível 3 do TMM uma organização integra atividades de teste no ciclo de vida do software. No nível 4 essa integração é aumentada pelo estabelecimento de um programa de revisão. Revisões são conduzidas em todas as fases do ciclo de vida para identificar, catalogar e remover defeitos. Sub-objetivos de maturidade que apóiam esse objetivo incluem:

- Alta gerência deve desenvolver políticas de revisão, apoiar o processo de revisão e ter a responsabilidade de integrá-los à cultura organizacional;
  - O grupo de testes e o grupo de garantia de qualidade devem desenvolver e documentar objetivos, planos e mecanismos de registro para revisões em todo o ciclo de vida do software;
  - Itens de revisão devem ser especificados pelos grupos supracitados.
2. **Estabelecer um Programa de Métricas.** Uma programa de métricas de teste é essencial na avaliação da qualidade do processo de testes, para avaliar a produtividade dos funcionários e a efetividade do processo de teste, e para a melhoria do processo de teste. Métricas de teste são fundamentais no monitoramento e controle do processo de teste. Métricas de cada fase do ciclo de vida devem ser especificadas. Sub-objetivos de maturidade que apóiam esse objetivo incluem:
- Políticas de métricas de teste e objetivos devem ser definidos;
  - Um plano de métricas de teste deve ser desenvolvido com mecanismos para coleta, análise e aplicação;
  - Planos de ação que utilizam resultados de métricas a malhorias no processo de teste devem ser desenvolvidas e documentadas.

#### 5.5.4. Nível 5 - Otimização, Prevenção de Defeitos e Controle da Qualidade

Organizações nesse nível testam para assegurar que o software satisfaz a especificação, é confiável e que a organização acredita em sua segurança. Há testes para detectar e prevenir falhas. Prevenção se aplica a falhas de requisitos, projeto e implementação. Como o processo é repetível, definido, gerenciado e mensurado, ele pode ser refinado e continuamente melhorado. Abaixo seguem os objetivos de maturidade do nível 5:

1. **Aplicação de Dados do Processo para Prevenção de Defeitos.** Organizações maduras são capazes de aprender a partir de seu histórico. Seguindo essa filosofia, organizações no mais alto nível de maturidade registram defeitos, analisam padrões de defeitos e identificam as causas dos erros. Planos de ação são desenvolvidos, ações são tomadas para prevenir recorrência de defeitos e há um mecanismo de acompanhamento do do progresso. Sub-objetivos de maturidade que apóiam esse objetivo incluem:
- Um time de prevenção de defeitos deve ser estabelecido com o apoio de gerência;
  - Defeitos injetados ou removidos devem ser identificados e gravados durante cada fase do ciclo de vida;
  - Planos de ação devem ser desenvolvidos através de interações de gerentes, desenvolvedores e testadores para prevenir recorrência de defeitos identificados. Tais planos devem ser acompanhados.
2. **Controle de Qualidade.** No Nível 5 as organizações usam amostragens estatísticas, medidas de níveis de confiança e objetivos de segurança para lidar com o processo de testes. Ferramentas automatizadas dão suporte à execução e re-execução de casos de teste e coleta e análise de defeitos. Sub-objetivos que dão suporte ao controle de qualidade estatístico incluem:
- O grupo de teste de software e o grupo de garantia de qualidade devem estabelecer objetivos de qualidade para produtos de software tais como taxa de defeitos e níveis de confiança;
  - Gerentes de teste devem incorporar esses objetivos de qualidade nos planos de teste;
  - O grupo de teste deve ser treinado em métodos estatísticos.

3. **Otimização do Processo de Testes.** No mais alto nível do TMM, o processo de teste é submetido a contínua melhoria nos projetos e na organização. O processo de teste é quantificado e pode ser ajustado de forma que o crescimento da capacidade seja um processo progressivo. Otimizar o processo de testes envolve:

- Identificar práticas de teste que precisam ser melhoradas;
- Implementar as melhorias;
- Rastrear progresso das melhorias;
- Avaliação contínua de novas ferramentas relacionadas a testes e tecnologias de adaptação;
- Suporte à transferência de tecnologia;

Sub-objetivos de maturidade incluem:

- Um grupo de melhoria do processo de teste deve ser estabelecido para monitorar o processo de teste e identificar áreas de melhoria;
- Um mecanismo deve ser posto em prática para avaliar novas ferramentas e tecnologias que possam melhorar a capacidade e a maturidade do processo de teste;

## 6. Aplicando um Questionário de Avaliação TMM no Projeto CIn

Com o intuito de aplicar os conceitos aprendidos e de poder ter uma idéia do nível da maturidade de testes no projeto CIn, foi aplicado um questionário de avaliação com questões relacionadas aos objetivos de maturidade e aos fluxos de processo descritos em cada nível do TMM. Com tal avaliação busca-se também ajudar a determinar até qual extensão o projeto tem mecanismos para alcançar tais objetivos e resolver eventuais problemas de maturidade.

### 6.1. Preparação

Por critério aleatório, 6 membros do time de automação do projeto foram selecionados para participar da avaliação. Um breve relato da proposta, escopo e restrições foi preparado para guiar o desenvolvimento da avaliação. O plano de avaliação utilizado está disponível em [Olsen and Vinje, 1998] [Homyen, 1998].

### 6.2. Conduzindo a Avaliação

Foram coletados questionários, informações de entrevistas e alguns documentos relevantes relacionados ao processo de testes do projeto. O algoritmo de *ranking* utilizado na avaliação TMM foi similar ao algoritmo descrito por [S. and Bothwell, 1995]; Primeiramente ele requer uma classificação em relação aos sub-objetivos de maturidade, logo em seguida em relação aos objetivos e finalmente ao nível de maturidade.

### 6.3. Analisando as Saídas da Avaliação

A avaliação coloca a organização no nível 3. O TMM mostra alguns pontos falhos em gerenciamento, supervisionamento e métricas. Tais tópicos encontram-se basicamente no Nível 4 e no Nível 5.

O TMM também mostra que o projeto tem um excelente planejamento e organização de testes. Além disso, há uma aceitável eficiência dos testes, definida como a porcentagem dos erros encontradas nos testes. Já a taxa de erros encontrava-se na média.

Os pontos fracos também indicam que as ações efetivas para correção de desvios do plano de teste estão ocorrendo com muito atraso.

O processo de testes também possui um grande problema: Não há o uso apropriado de ferramentas de teste de alta qualidade.

Vale salientar que o TMM não pode indicar claramente problemas do testador, mas sim como a organização reage a esses problemas de forma inapropriada.

Um melhor monitoramento do processo de testes provavelmente dissolveria os principais problemas encontrados.

## 7. Conclusões

A fase de teste é um componente crítico no processo de desenvolvimento de um software maduro. Tal fase é também uma das mais desafiadoras e dispendiosas do processo, e em sua mais completa definição fornece forte apoio à produção de software com qualidade. Apesar de seu vital papel no desenvolvimento de software, os modelos de maturidade existentes não se referem adequadamente ao fluxo de testes e não possuem a natureza bem definida de um processo de testes maduro. Nesse contexto surgiu o TMM com o intuito de complementar o CMM e o CMMI endereçando questões importantes a gerentes de teste, especialistas em testes e à equipe de garantia de qualidade de software. O TMM contém uma série de níveis de maturidade através dos quais uma organização pode progredir em direção à maturidade do processo de testes, uma série de práticas recomendadas em cada nível de maturidade e um modelo de avaliação que permite às organizações avaliarem e aperfeiçoarem seu processo de testes.

## Referências

- Beizer, B. (1990). *Software System Testing Techniques*. Van Nostrand Reinhold, second edition.
- Burnstein, I. (2003). *Practical Software Testing*. Springer-Verlag.
- Coallier, F. (1994). How iso 9001 fits into the software world. *IEEE Software*, pages 98–100.
- Gelperin, D. and Hayashi, A. (1996). *How to Support Better Software Testing*. Application Development Trends.
- Gelperin, D. and Hetzel, B. (1988). The growth of software testing. *CACM*, 31(6):687–695.
- Homyen, A. (1998). An assessment model to determine test process maturity. *Diss., Illinois Institute of Technology*.
- Koomen, T. and Pol., M. (1999). *Test Process Improvement*. Addison-Wesley.
- Olsen, K. and Vinje, P. (1998). *Using the Testing Maturity Model in practical test-planning and post-evaluation*. Eurostar.
- Paulk, M. and Konrad, M. (1994). An overview of iso's spice project. *American Programmer*, pages 16–20.
- Paulk, M., Weber, C., Curtis, B., and Chrissis, M. (1995). *The Capability Maturity Model Guideline for Improving the Software Process*. Addison-Wesley.
- S. and Bothwell, C. (1995). A cmm appraisal framework, version 1.0. *CMU/SEI-95-TR-001*.
- Thayer, R. (1990). *Software Engineering Project Management*. IEEE Computer Society Press.
- Weatheill, T. (2001). In the testing maturity model maze. *Journal of Software Testing Professionals*, pages 8–13.