



Telecom & IT Solutions

Modelo de Maturidade de Capabilidade de Software (CMM)



Versão 1.2

11/10/2001

Tradução

José Marcos Gonçalves

André Villas Boas

Índice

Nota dos Tradutores	i
Das Responsabilidades	ii
DISCLAIMER	iv
Ao leitor	I
Qual o propósito deste documento?	II
Quem deveria ler este documento?	II
Como este documento está organizado?	III
Quais são os outros Produtos CMM?	IV
Como você pode receber mais informações?	IV
1 A ESTRUTURA DA MATURIDADE DO PROCESSO	1
1.1 ORGANIZAÇÕES IMATURAS VERSUS ORGANIZAÇÕES MADURAS	1
1.2 CONCEITOS FUNDAMENTAIS ASSOCIADOS À MATURIDADE DE PROCESSO	2
1.3 VISÃO GERAL DO MODELO DE MATURIDADE DA CAPABILIDADE	4
2 OS CINCO NÍVEIS DA MATURIDADE DE PROCESSO DE SOFTWARE	6
2.1 CARACTERIZAÇÃO COMPORTAMENTAL DOS NÍVEIS DE MATURIDADE	7
2.1.1 <i>Nível 1 – O Nível Inicial</i>	8
2.1.2 <i>Nível 2 – O Nível Repetível</i>	8
2.1.3 <i>Nível 3 – O Nível Definido</i>	9
2.1.4 <i>Nível 4 – O Nível Gerenciado</i>	10
2.1.5 <i>Nível 5 – O Nível Em Otimização</i>	10
2.2 ENTENDENDO OS NÍVEIS DE MATURIDADE	11
2.2.1 <i>Entendendo o Nível Inicial</i>	11
2.2.2 <i>Entendendo dos Níveis Repetível e Definido</i>	12
2.2.3 <i>Entendendo os Níveis Gerenciado e Em Otimização</i>	12
2.3 VISIBILIDADE INTERNA DO PROJETO DE SOFTWARE	14
2.4 CAPABILIDADE DO PROJETO E A PREVISÃO DE DESEMPENHO	16
2.5 PULANDO DE NÍVEIS DE MATURIDADE	18
3 DEFINIÇÃO OPERACIONAL DO MODELO DE MATURIDADE DE CAPABILIDADE	21
3.1 ESTRUTURA INTERNA DOS NÍVEIS DE MATURIDADE	21
3.2 NÍVEIS DE MATURIDADE	22
3.3 ÁREAS-CHAVE DE PROCESSO	22
3.4 CARACTERÍSTICAS COMUNS	28
3.5 PRÁTICAS-CHAVE	29
4 USANDO O CMM	32
4.1 MÉTODOS DE AVALIAÇÃO DO PROCESSO DE SOFTWARE E AVALIAÇÃO DA CAPABILIDADE DO SOFTWARE	33
4.2 DIFERENÇAS ENTRE AVALIAÇÃO DE PROCESSO DE SOFTWARE E AVALIAÇÃO DA CAPABILIDADE DE SOFTWARE ..	35
4.3 OUTROS USOS DO CMM EM MELHORIA DE PROCESSOS	36
5 FUTURAS DIREÇÕES DO CMM	38
5.1 O QUE O CMM NÃO COBRE	38
5.2 ATIVIDADES DE CURTO PRAZO	38
5.3 ATIVIDADES DE LONGO PRAZO	39
5.4 CONCLUSÃO	39
6 REFERÊNCIAS	41
7 APÊNDICE A: METAS PARA CADA ÁREA-CHAVE DE PROCESSO	44

Nota dos Tradutores

Este trabalho tem como objetivo único facilitar a divulgação de um modelo de melhoria de processos que vem sendo muito utilizado e tem trazido resultados positivos àqueles que o estão usando.

O interesse no assunto é fruto de um trabalho que vem sendo realizado na Fundação CPqD, na área da qualidade, com a avaliação de vários modelos de melhoria de processos de software e posterior adoção do CMM. Após muitos estudos, várias palestras, adaptação do modelo para o CPqD e execução de mini-avaliações, além de participação em uma avaliação oficial do SEI, foi observado que um dos principais problemas de divulgação das práticas do modelo residia no fato do mesmo estar escrito em inglês. Diante disto, decidiu-se traduzi-lo e torná-lo público, esperando com isto facilitar o acesso de mais pessoas ao assunto.

Esta não é uma tradução oficial do documento do SEI e qualquer uso formal do CMM deve ser feito com base na documentação oficial do SEI, a qual pode ser obtida no endereço: <http://www.sei.cmu.edu>.

A Fundação CPqD não se responsabiliza pelo uso do material aqui contido, já que o mesmo tem caráter apenas informativo.

Todo e qualquer comentário, bem como o interesse em receber versões aprimoradas deste texto, podem ser enviados para os tradutores que, desde já, agradecem a atenção.

Um abraço e bom trabalho,

André Villas-Boas (villas@cpqd.com.br)
José Marcos Gonçalves (imarcos@cpqd.com.br)

Campinas, outubro de 2001.

Outras traduções relacionadas:

- TR 25 – Nível 2 (disponível)
- TR 25 – Interpretando o CMM (em elaboração)
- TR 25 – Nível 3 (em elaboração)

Das Responsabilidades

Uma permissão especial para reproduzir material sobre o Capability Maturity Model® (CMM®), que representa direitos autorais da Universidade Carnegie Mellon é concedida pela Universidade Carnegie Mellon.

® Capability Maturity Model é uma marca registrada da Universidade Carnegie Mellon.

® CMM está registrado no Escritório de Marcas e Patentes dos Estados Unidos da América.

DA GARANTIA. O MATERIAL DA UNIVERSIDADE CARNEGIE MELLON E DO INSTITUTO DE ENGENHARIA DE SOFTWARE É FORNECIDO EM BASES “AS IS”. A UNIVERSIDADE CARNEGIE MELLON NÃO FORNECE QUALQUER GARANTIA SOB NENHUMA ESPÉCIE, SE EXPRESSA OU IMPLÍCITA, A QUALQUER TEMA INCLUSO, MAS NÃO LIMITADA A, GARANTIA DE ADEQUAÇÃO PARA O PROPÓSITO DESTE OU MERCADOLÓGICO, EXCLUSIVIDADE OU RESULTADOS OBTIDOS DO USO DO MATERIAL AQUI CONTIDO. A UNIVERSIDADE CARNEGIE MELLON NÃO OFERECE QUALQUER GARANTIA, SOB QUALQUER ESPÉCIE, COM RESPEITO A ESTAR AS PATENTES, MARCAS OU DIREITOS AUTORAIS LIVRES DE QUALQUER INFRAÇÃO.

Da Isenção de Responsabilidade. A Fundação CPqD assegura à Universidade Carnegie Mellon a isenção de todas as responsabilidades demandadas, danos, despesas e perdas ocorridas de uso negligente ou uso indevido deste, ou conduta indevida pela Fundação CPqD considerando a Propriedade Intelectual da Universidade Carnegie Mellon. Essa cláusula deve permanecer após o término desse Acordo.

Do Foro. Esse Acordo deve ser governado pelas leis da Comunidade da Pennsylvania. Qualquer disputa ou reclamação ocorrida disso ou em relação a esse Acordo terá arbitragem estabelecida em Pittsburgh, Pennsylvania (EUA), de acordo com as regras da Associação Americana de Arbitragem e com julgamento prestado por árbitro(s) em qualquer corte de sua(s) jurisdição.

Se o material da Universidade Carnegie Mellon, protegido pelos direitos autorais, for incluído em materiais de treinamento ou num produto, o qual será vendido ou licenciado, a Fundação CPqD será responsável a impor todas as cláusulas necessárias para o cumprimento das obrigações referentes a esse Acordo, especificamente incluindo, mas não limitado a, à cláusula “Da Garantia” e a outras condições estabelecidas neste Acordo.

A permissão é concedida em termos de não exclusividade para propósitos comerciais.

Das Responsabilidades

Permissão adicional precisa ser obtida por escrito do Instituto de Engenharia de Software (SEI) para qualquer outro uso da Propriedade Intelectual do Instituto de Engenharia de Software da Universidade Carnegie Mellon.

A exatidão e interpretação dessa tradução são de responsabilidade da Fundação CPqD. O Instituto de Engenharia de Software e a Universidade Carnegie Mellon não têm qualquer participação nessa tradução.

DISCLAIMER

“Special permission to reproduce material on the Capability Maturity Model® (CMM®) copyrighted by Carnegie Mellon University is granted by Carnegie Mellon University.

® Capability Maturity Model is a registered mark of Carnegie Mellon University.

® CMM is registered in the U.S. Patent and Trademark Office.

“NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

“Hold Harmless. Fundação CPqD holds CMU harmless for all liabilities, demands, damages, expenses and losses arising out of the negligent use or willful misuse of or negligent conduct or willful misconduct by Fundação CPqD regarding CMU Intellectual Property. This provision shall survive the termination of this agreement.

“Disputes. This Agreement shall be governed by the laws of the Commonwealth of Pennsylvania. Any dispute or claim arising out of or relating to this Agreement will be settled by arbitration in Pittsburgh, Pennsylvania in accordance with the rules of the American Arbitration Association and judgment upon award rendered by the arbitrator(s) may be entered in any court having jurisdiction.

“If the CMU copyrighted material is included in training matter or a product which is sold or sublicensed, Fundação CPqD is responsible to impose all provisions necessary to fulfill its obligations under this agreement, including, but not limited to the “No Warranty” clause and the other conditions stated in this agreement.

“This permission is granted on a non-exclusive basis for commercial purposes.

“Additional permission must be obtained in writing from the SEI for any other use of CMU/SEI intellectual property”.

Disclaimer

“Accuracy and interpretation of this translation are the responsibility of Fundação CPqD. The Software Engineering Institute and Carnegie Mellon University has not participated in this translation.

Ao leitor

A descrição do Modelo de Maturidade de Capabilidade de Software foi produzida inicialmente por um grupo dedicado de pessoas que passaram muito tempo discutindo o modelo e suas características e tentaram documentá-lo em “CMM v1.0”. Esse grupo era constituído por: Mark Paulk, Bill Curtis, Mary Beth Chrissis, Edward Averill, Judy Bamberger, Tim Kasse, Mark Konrad, Jeff Perdue, Charlie Weber e Jim Withey.

Este documento é baseado na visão de Watts Humphrey, primeiro diretor do “Programa de Processo de Software” do SEI. Foram elaboradas várias versões até que o documento chegasse ao produto final. Jim Withey, Mark Paulk e Gynthia Wise produziram a primeira versão em 1990. Watt Humphrey elaborou a segunda versão do documento e Mark Paulk então concluiu o documento e ficou responsável pela produção do livro até o final. Mary Beth Chrissis e Bill Curtis ajudaram Mark a produzir a revisão do “CMM v1.0” em Agosto de 1991. Mark Paulk produziu a revisão do “CMM v1.1, que é este relatório técnico.

Em vários estágios, várias pessoas contribuíram com os conceitos expressos neste documento. São elas: Joe Besselman, Marilyn Bush, Anita Carleton, Marty Carlson, Betty Deimel, Suzie Garcia, Richard Kauffold, Steve Masters, Mary Merrill, Jim Over, George Pandelios, Jane Siegel e Charlie Weber.

Prezamos a ajuda administrativa de Todd Bowman, Dorothy Josephson, Debbie Punjack, Carolin Tady, Marcia Theoret, Andy Tsounos e David White e a assistência editorial de Mary Beth Chrissis, Suzanne Couturiaux e Bill Pollack. Renne Dutkouski do Instituto Americano para Pesquisa forneceu sugestões para o *design* do documento.

Em Novembro de 1986, o Instituto de Engenharia de Software (SEI), com a assistência da *Mitre Corporation*, começou a desenvolver a estrutura de maturidade do processo que ajudaria as organizações a melhorar seus processos de software. Esse esforço foi iniciado em resposta a uma solicitação de fornecer ao governo federal um método para avaliar a capacidade de software de seus fornecedores de software. Em Setembro de 1987, o SEI liberou uma breve descrição da estrutura de maturidade de software [Humphrey 87a] e um questionário da maturidade [Humphrey 87b] o SEI pretendeu o questionário da maturidade para fornecer uma ferramenta simples para identificar as áreas onde um processo de software da organização precisava de melhoria. Infelizmente, o questionário da maturidade era muito freqüentemente visto como “o modelo” ao invés do veículo para explorar as questões de maturidade do processo.

Após quatro anos de experiência com a estrutura de maturidade do processo de software e com a versão preliminar de questionário da maturidade, o SEI evoluiu a estrutura da maturidade do processo de software para o Modelo de Maturidade da

Ao Leitor

Capabilidade para Software (CMM) [Paulk91, Weber91]. O CMM é baseado no conhecimento adquirido a partir de verificações do processo de software e da realimentação extensiva da indústria e do governo. Através da elaboração da estrutura de maturidade, o modelo tem mostrado que fornece às organizações um guia mais efetivo para o estabelecimento de programas de melhoria de processo.

A versão inicial do CMM, Versão 1.0, foi revisada e utilizada pela comunidade de software durante 1991 e 1992. Um *workshop* foi realizado em Abril de 1992, em CMM Versão 1.0, e reuniu mais de 200 profissionais de software. Esta versão do CMM, Versão 1.1, é o resultado da realimentação desse *workshop* e da contínua realimentação da comunidade de software.

O CMM é o fundamento para se construir sistematicamente um conjunto de ferramentas, inclusive o questionário da maturidade, que são úteis na melhoria do processo de software. O ponto essencial a ser lembrado é que o modelo, não o questionário, é a base para a melhoria do processo de software. Este documento pretende introduzir o leitor ao CMM v1.1.

Qual o propósito deste documento?

Este documento fornece uma visão técnica do Modelo de Maturidade da Capabilidade para Software e reflete a Versão 1.1. Especificamente, este documento descreve a estrutura da maturidade do processo dos cinco níveis de maturidade, os componentes estruturais que englobam o CMM, como o CMM é utilizado na prática e as futuras direções do CMM. Este documento serve como uma das melhores fontes para entender o CMM e deveria esclarecer algumas das concepções errôneas associadas aos conceitos de maturidade do processo de software como advogado pelo SEI.

O SEI tem trabalhado com a indústria e com o governo para refinar e expandir o modelo, sendo que as organizações de software são incentivadas a focarem-se no CMM e não no questionário da maturidade. O SEI desenvolveu, e está desenvolvendo, um conjunto de produtos de processo para encorajar esse foco. Este documento [Paulk93a], em combinação com as “Práticas-chave do Modelo de Maturidade da Capabilidade, Versão 1.1” descreve as práticas-chave de cada nível de maturidade. Este documento descreve os princípios que suportam a maturidade do processo de software e pretende ajudar as organizações de software a utilizar o CMM V1.0 como guia para melhorar a maturidade dos seus processos de software.

Quem deveria ler este documento?

Este documento apresenta uma introdução ao CMM e seus produtos associados. Portanto, qualquer pessoa que está interessada em aprender sobre o CMM deveria ler este documento. Entretanto, este documento assume que o leitor tenha algum

Ao Leitor

conhecimento, ou experiência, em desenvolvimento e/ou manutenção de software tanto quanto um entendimento dos problemas que a comunidade de software enfrenta hoje.

Este documento pode ser usado de várias formas:

- por qualquer um que queira entender que as práticas-chave são parte do efetivo processo para desenvolvimento ou manutenção de software,
- por qualquer um que queira identificar as práticas-chave que são necessárias para alcançar o próximo nível de maturidade no CMM,
- pelas organizações que queiram entender e melhorar sua capacidade de desenvolver software efetivamente,
- pelas organizações que estão adquirindo ou contratantes que queiram conhecer os riscos de ter uma organização de software específica realizando o trabalho de um contrato,
- pelo SEI como base para desenvolver questões para o questionário da maturidade, e
- por instrutores que preparam equipes para realizarem avaliações de processo de software ou avaliações da capacidade de software.

Como este documento está organizado?

Este documento possui 5 capítulos:

<i>Capítulo 1</i>	Define os conceitos necessários para entender o CMM e a motivação e o objetivo por de trás dele.
<i>Capítulo 2</i>	Descreve os 5 níveis do CMM e os princípios que os embasam.
<i>Capítulo 3</i>	Descreve como o CMM está estruturado em áreas-chave de processo, organizado por características comuns, e descrito em termos de práticas-chave.
<i>Capítulo 4</i>	Fornecer uma visão de alto nível de como o CMM provê um guia para as avaliações do processo de software, avaliação da capacidade de software e programas de melhoria de processos.
<i>Capítulo 5</i>	Conclui fornecendo uma descrição das futuras direções do CMM e seus produtos relacionados.

Quais são os outros Produtos CMM?

Embora este documento possa ser lido isoladamente, ele está designado para ser o ponto de lançamento de outros produtos. Este documento e os produtos associados ajudam o leitor entender e usar o CMM. Todos os produtos baseados em CMM são, ou serão, sistematicamente derivados do modelo. Quando este documento foi escrito, muitos desses produtos não estavam disponíveis em sua forma final, embora as versões preliminares estivessem em vários estágios de teste e versões piloto.

O conjunto de produtos baseados no CMM inclui várias ferramentas de diagnóstico, que são usadas pelas equipes de verificação do processo de software¹ e avaliação da capacidade de software² para identificar resistências, fraquezas e riscos do processo de software de uma organização. Provavelmente a mais conhecida dessas ferramentas é o questionário da maturidade. A verificação do processo de software e os métodos e treinamentos de avaliação da capacidade de software também contam com o CMM.

Os usuários desses produtos formam a comunidade dedicada a melhorar a maturidade dos seus processos de software. O SEI continuará trabalhando com a comunidade de software para elevar a categoria do modelo e seus produtos associados.

Como você pode receber mais informações?

Para mais informações a respeito do CMM e seus produtos associados, inclusive treinamento em CMM e como realizar verificações de processo de software e avaliações da capacidade de software, entrar em contato com:

SEI Customer Relations
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
(412) 268-5800
Internet: customer-relations@sei.cmu.edu

Os relatórios técnicos do SEI, tais como este documento e “Key Process of the Capability Maturity Model, Version 1.1”, são obtidos diretamente de: Defense Techni-

¹ A verificação do processo de software é uma apreciação realizada por uma equipe treinada de profissionais de software para determinar a situação do processo de software atual de uma organização, para determinar as questões relacionadas ao processo de software mais prioritárias da organização e obter o suporte organizacional para a melhoria do processo de software.

² A avaliação da capacidade de software é uma apreciação realizada por uma equipe treinada de profissionais para identificar subcontratados que são qualificados para realizar o trabalho de software ou monitorar a situação do processo de software utilizado em um esforço de software já existente.

Ao Leitor

cal Information Center (DTIC), National Technical Information Service (NTIS), e Research Access Inc. (RAI). Estes documentos podem ser obtidos contatando:

RAI: Research Access Inc.
3400 Forbes Avenue
Suite 302
Pittsburg, PA 15213
Phone: (800) 685-6510
FAX: (412) 682-6530

NTIS: National Technical Information Service
U.S. Department of Commerce
Springfield, VA 22161-2103
Phone: (703) 487-4600

DTIC: Defense Technical Information Center
ATTN: FDRA Cameron Station
Alexandria, VA 22304-6145
Phone: (703) 274-7633

Os relatórios técnicos do SEI também estão disponíveis via Internet. Para usar ftp anônimo de um sistema Unix Internet:

```
ftp ftp.sei.cmu.edu3  
login: anonymous  
password: <seu user-id ou qualquer string>  
cd pub/cmm  
get READ.ME  
get <files>  
quit
```

O arquivo READ.ME contém informações sobre quais arquivos estão disponíveis. Outras publicações do SEI estão disponíveis de forma similar.

³ O endereço da máquina para ftp do SEI é 128.237.2.179.

1 A Estrutura da Maturidade do Processo

Após duas décadas de promessas apenas parcialmente cumpridas sobre ganhos de produtividade e qualidade a partir da aplicação de novas metodologias e tecnologias de software, a indústria e as organizações governamentais estão compreendendo que seu problema fundamental reside na inabilidade para gerenciar o processo de software [DoD87]. Os benefícios de melhores métodos e ferramentas não podem ser calculados na efervescência de um projeto carente de disciplina, caótico. Em muitas organizações, os projetos freqüentemente atrasam excessivamente, gastando o dobro do orçamento planejado [Siegel90]. Em tais casos, a organização freqüentemente não está provendo a infra-estrutura e o suporte necessários para apoiar os projetos de modo a evitarem esses problemas.

Mesmo em organizações carentes de disciplina, entretanto, alguns projetos isolados produzem excelentes resultados. Quando tais projetos são bem sucedidos, é geralmente graças a esforços heróicos de uma equipe dedicada, e não através da repetição de métodos provados de uma organização com um processo de software maduro. Na ausência de um processo de software abrangente na organização, a repetição dos resultados depende inteiramente de se ter as mesmas pessoas disponíveis para o próximo projeto. O sucesso, que depende da disponibilidade de pessoas específicas, não fornece condições para a melhoria da produtividade e da qualidade na organização por um longo período. Melhoramentos contínuos só podem ocorrer através esforços focados e sustentados na direção da construção de uma infra-estrutura de processo de desenvolvimento de software e práticas de gestão efetivas

1.1 Organizações Imaturas Versus Organizações Maduras

O estabelecimento de metas sensatas para o melhoramento do processo exige um entendimento das diferenças entre as organizações de software imaturas e as organizações de software maduras. Em uma organização imatura, os processos de software geralmente são improvisados por pessoas experientes, em conjunto com seus gerentes, durante o curso do projeto. Mesmo que tenha sido especificado, o processo de software não é rigorosamente seguido ou não é obrigatório. A organização de software imatura é reacionária e os gerentes geralmente estão focados na solução de problemas imediatos (ação mais conhecida como “apagar incêndios”). Os cronogramas e os orçamentos são rotineiramente estourados porque não estão baseados em estimativas realistas. Quando são impostos prazos que não podem ser ultrapassados, a funcionalidade e a qualidade do produto são freqüentemente comprometidas para que o cronograma seja cumprido.

Em uma organização imatura, não há bases objetivas para a avaliação da qualidade do produto e nem para a resolução de problemas associados a ele ou ao processo utilizado. Sendo assim, é difícil antever a qualidade do produto. As atividades

A Estrutura da Maturidade do Processo

que objetivam aumentar a qualidade, tais como revisões e testes, são freqüentemente reduzidas ou eliminadas em função dos atrasos ocorridos no andamento do projeto.

Por outro lado, uma organização de software madura possui habilidade para gerenciar o desenvolvimento de software e os processos de manutenção em toda a organização. O processo de software é cuidadosamente comunicado à equipe já existente e aos novos funcionários, sendo que as atividades são realizadas de acordo com processos planejados. Os processos obrigatórios estão voltados para o uso [Humphrey91b], sendo consistentes com a forma natural de se fazer as coisas. Esses processos definidos são atualizados sempre que necessário e as melhorias são implementadas através de testes-piloto e/ou análise de custo-benefício. As regras e as responsabilidades no processo definido são claras em toda a parte do projeto e da organização.

Em uma organização madura, os gerentes monitoram a qualidade dos produtos de software e a satisfação do cliente. Há uma referência objetiva e quantitativa para avaliar a qualidade do produto e para analisar os problemas relacionados a ele e ao processo. Os cronogramas e os orçamentos estão baseados em desempenho histórico e são realistas; os resultados esperados para custo, cronograma, funcionalidade e qualidade do produto são quase sempre alcançados. Em geral, um processo disciplinado é seguido de forma consistente porque todos os participantes compreendem a importância disso. Além disso, existe infra-estrutura necessária para dar suporte ao processo.

Para tirar proveito dessas observações sobre as organizações de software maduras e imaturas, é necessário que se construa de uma estrutura de maturidade de processo de software. Essa estrutura descreve um caminho evolutivo, desde os processos caóticos, *ad hoc*, até os processos maduros, disciplinados. Sem essa estrutura, os programas de melhoria podem se mostrar ineficientes porque não são estabelecidos os fundamentos necessários para dar suporte às sucessivas melhorias. A estrutura de maturidade de processo de software emerge da integração dos conceitos de processo, capacidade, desempenho e maturidade de processo de software, todos definidos nos parágrafos seguintes.

1.2 Conceitos Fundamentais Associados à Maturidade de Processo

De acordo com o dicionário Webster's, um **processo**⁴ é “um sistema de operações que produz alguma coisa... uma série de ações, mudanças ou funções que atinge um fim ou resultado.” O IEEE define um processo como: “uma seqüência de passos realizados com um dado propósito” [IEEE-STD-610]. Um *processo de software* pode ser definido como um conjunto de atividades, métodos, práticas e transforma-

⁴ De acordo com o dicionário Aurélio, **processo** é maneira pela qual se realiza uma operação, segundo determinadas normas.

A Estrutura da Maturidade do Processo

ções que as pessoas utilizam para desenvolver e manter um software e os seus produtos associados (por exemplo: planos e documentos de projeto, código, casos de teste e manuais de usuário). À medida que uma organização vai se tornando madura, o processo de software vai ficando mais definido, possibilitando que o mesmo seja implementado de modo mais consistente em toda a organização.

A **capabilidade do processo de software** descreve a gama de resultados esperados que podem ser alcançados com a aplicação do processo de software. A capacidade do processo de software de uma organização fornece um meio de se prever os resultados mais prováveis a serem esperados no próximo projeto a ser empreendido pela organização. constrain

O **desempenho do processo de software** representa os resultados reais alcançados seguindo-se o processo de software. Assim, o desempenho do processo de software foca nos resultados alcançados, enquanto que a capacidade do mesmo foca nos resultados esperados. Com base nos atributos de um projeto específico e no contexto no qual ele é conduzido, o desempenho real de um projeto pode não refletir a capacidade do processo total da organização; isto é, a capacidade do projeto pode ser restringida pelo seu ambiente. Por exemplo, alterações radicais na aplicação ou na tecnologia empregada podem colocar o pessoal do projeto na situação de aprendizes, o que faz com que a capacidade e o desempenho do seu projeto diminuam com relação à capacidade de processo total da organização.

A **maturidade do processo de software** é a extensão para a qual um processo específico é explicitamente definido, gerenciado, medido, controlado e efetivado. A maturidade representa o potencial de crescimento de capacidade e indica a riqueza do processo de software da organização e a consistência com que o mesmo é aplicado em todos os seus projetos. Em uma organização madura, o processo de software é bem compreendido - o que geralmente é feito por meio de documentação e treinamento - e está sendo continuamente monitorado e melhorado pelos seus usuários. A capacidade de um processo de software maduro é conhecida. A maturidade de processo de software implica que a produtividade e a qualidade resultantes do processo possam ser continuamente melhoradas através de ganhos consistentes na disciplina alcançada com a sua utilização.

Quando uma organização obtém ganhos na maturidade de um processo de software, ela o institucionaliza por meio de políticas, padrões e estruturas organizacionais. A **institucionalização** exige a construção de uma infra-estrutura e de uma cultura corporativa que possa dar suporte aos métodos, práticas e procedimentos de negócio que perdurem após possíveis afastamentos daqueles que originalmente os definiram.

1.3 Visão geral do Modelo de Maturidade da Capabilidade

Mesmo tendo pleno conhecimento de seus problemas, engenheiros e gerentes de software podem discordar quanto a importância das melhorias. Sem uma estratégia organizada, é difícil a gerência e a equipe de funcionários chegarem a um consenso sobre a prioridade dessas atividades. Para se conseguir resultados duradouros a partir de esforços em melhoria de processos, é necessário projetar um caminho evolutivo que incremente, em estágios, a maturidade do processo de software da organização. A estrutura de maturidade [Humphrey 87a] ordena esses estágios pré-estabelecidos, onde os resultados positivos alcançados em cada estágio são utilizados como embasamento para o próximo, objetivando melhorias no processo em sua totalidade. Dessa forma, uma estratégia de melhoria projetada a partir de uma estrutura de maturidade de processo de software orienta quanto ao caminho (como um mapa) a ser seguido para a contínua melhoria do mesmo. Orienta também os avanços e identifica deficiências na organização, não objetivando fornecer soluções rápidas para projetos em dificuldades.

O Modelo de Maturidade da Capabilidade de Software fornece às organizações de software um guia de como obter controle em seus processos para desenvolver e manter software e como evoluir em direção a uma cultura de engenharia de software e excelência de gestão. O CMM foi projetado para guiar as organizações de software no processo de seleção das estratégias de melhoria, determinando a maturidade atual do processo e identificando as questões mais críticas para a qualidade e melhoria do processo de software. Focando em um conjunto limitado de atividades e trabalhando agressivamente para concluí-las com êxito, a organização pode melhorar o processo em toda a sua estrutura, possibilitando ganhos contínuos e duradouros na capacidade do processo de software.

A estrutura em estágios do CMM é baseada em princípios de qualidade de produto dos últimos sessenta anos. Nos anos 30, Walter Shewhart promulgou os princípios de controle estatístico da qualidade. Seus princípios foram desenvolvidos e demonstrados com sucesso no trabalho de W. Edwards Deming [Deming86] e Joseph Juran [Juran88, Juran89]. Esses princípios foram adaptados pelo SEI dentro da estrutura de maturidade que estabelece a gestão de projeto e os fundamentos de engenharia para o controle quantitativo do processo de software, que é a base para a contínua melhoria do processo.

A estrutura de maturidade da qual esses princípios de qualidade foram adaptados foi primeiramente inspirada por Philip Crosby em seu livro *Quality is Free* [Crosby70]. O quadro de maturidade de gestão da qualidade de Crosby descreve cinco etapas para a adoção das práticas de qualidade. Essa estrutura de maturidade foi adaptada para o processo de software por Ron Radice e seus colegas, trabalhando sob a direção de Watts Humphrey da IBM [Radice85]. Humphrey trouxe essa estrutura de maturidade para o Instituto de Engenharia de Software em 1986, acrescentou o conceito de níveis de maturidade e desenvolveu o fundamento para o seu uso atual através da indústria de software.

A Estrutura da Maturidade do Processo

As primeiras versões da estrutura de maturidade de Humphrey foram descritas em relatórios técnicos do SEI [Humphrey87a, Humphrey87b], artigos [Humphrey88] e em seu livro *Managing the Software Process* [Humphrey89]. Um questionário de maturidade preliminar [Humphrey87b] foi liberado em 1987 como uma ferramenta para fornecer às organizações uma maneira de caracterizar a maturidade de seus processos de software. Dois métodos, avaliação de processo de software e avaliação de capacidade de software, foram desenvolvidos para estimar a maturidade do processo de software em 1987. Desde 1990, o SEI, com a ajuda de muitas pessoas do governo e da indústria, tem expandido e refinado o modelo baseado em vários anos de experiência na aplicação do mesmo na melhoria do processo de software.

2 Os Cinco Níveis da Maturidade de Processo de Software

A melhoria contínua de processo é baseada em muitas etapas evolutivas pequenas ao invés de fundamentar-se em inovações revolucionárias [Imai86]. O CMM fornece uma estrutura para organizar essas etapas evolutivas em cinco níveis de maturidade que estabelecem fundamentos sucessivos para a contínua melhoria do processo. Esses cinco níveis de maturidade definem uma escala ordinal para medir a maturidade de um processo de software da organização e para avaliar a sua capacidade de processo de software. Os níveis também ajudam uma organização a priorizar seus esforços de melhorias.

Nível de maturidade é um estágio evolutivo bem definido em busca de um processo de software maduro. Cada nível de maturidade fornece uma gama de fundamentos para a melhoria contínua do processo. Cada nível compreende um conjunto de objetivos de processos que, quando satisfeitos, estabilizam um componente importante do processo de software. Alcançando cada nível da estrutura de maturidade, estabelecem-se diferentes componentes no processo de software, resultando em um crescimento na capacidade de processo da organização.

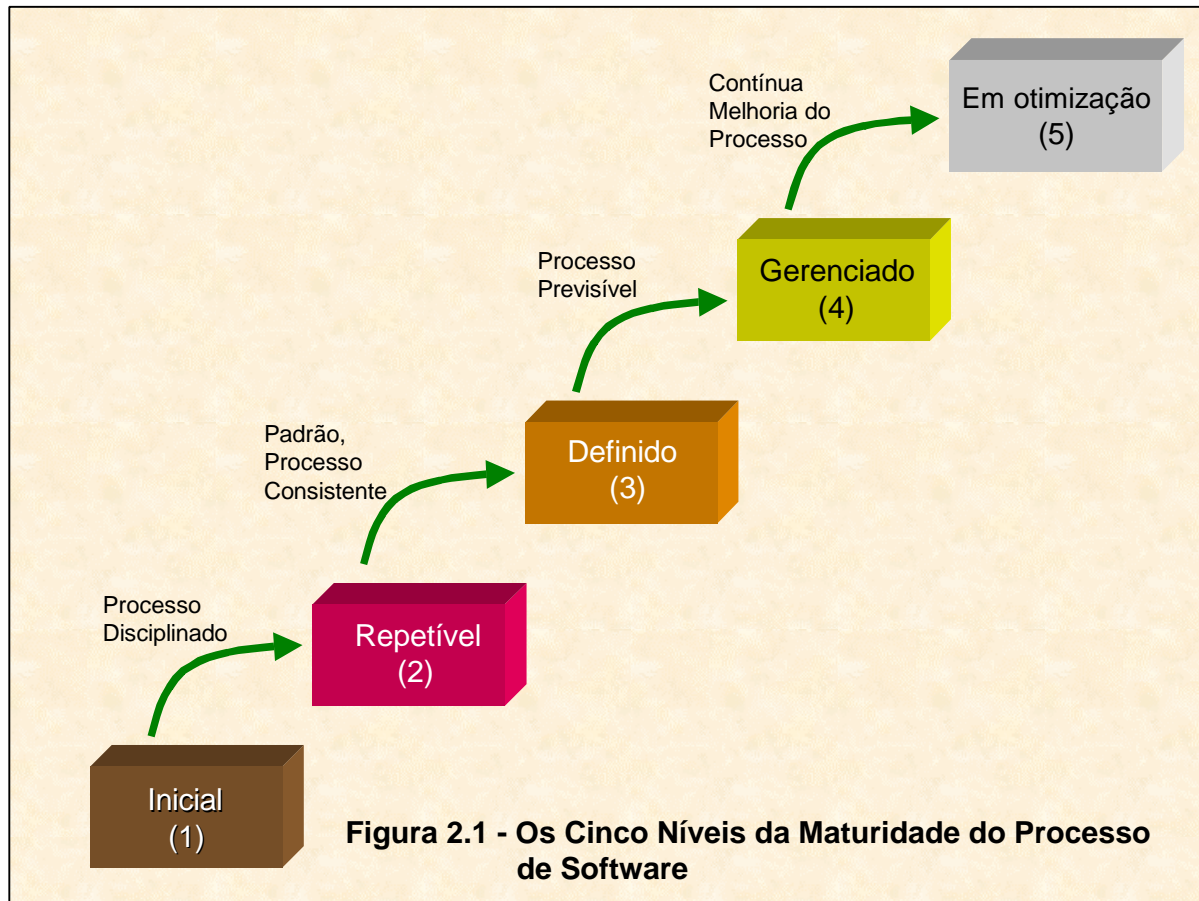
Organizando o CMM nos cinco níveis mostrados na Figura 2.1, prioriza-se ações de melhoria para o crescimento da maturidade do processo de software. As setas rotuladas na Figura 2.1 indicam o tipo de capacidade de processo que está sendo institucionalizado pela organização a cada etapa da estrutura de maturidade.

As caracterizações dos cinco níveis de maturidade, descritos a seguir, destacam as mudanças, realizadas em cada nível, no processo principal.

- 1) Inicial** O processo de software é caracterizado como “*ad hoc*” e até mesmo ocasionalmente caótico. Poucos processos são definidos e o sucesso depende de esforço individual.
- 2) Repetível** Os processos básicos de gestão de projeto são estabelecidos para acompanhar custo, cronograma e funcionalidade. A necessária disciplina do processo existe para repetir sucessos anteriores em projetos com aplicações similares.
- 3) Definido** O processo de software para as atividades de gestão e engenharia é documentado, padronizado e integrado em um processo de software padrão para a organização. Todos os projetos utilizam uma versão aprovada do processo de software padrão para desenvolver e manter software.
- 4) Gerenciado** Medidas detalhadas do processo de software e da qualidade do produto são realizadas. O processo e os produtos de software são quantitativamente compreendidos e controlados.

Os Cinco Níveis da Maturidade do Processo de Software

5) **Em Otimização** A melhoria contínua do processo é propiciada pelo *feedback* quantitativo do processo e pelas idéias e tecnologias inovadoras.



2.1 Caracterização Comportamental dos Níveis de Maturidade

Os níveis de maturidade de 2 a 5 podem ser caracterizados pelas atividades executadas pela organização para estabelecer ou melhorar o processo de software, pelas atividades realizadas em cada projeto e pela capacidade de processo resultante nos projetos. A caracterização comportamental do Nível 1 também é descrita para estabelecer uma base de comparação com as melhorias de processos alcançadas nos níveis mais elevados de maturidade.

Os Cinco Níveis da Maturidade do Processo de Software

2.1.1 Nível 1 – O Nível Inicial

No Nível Inicial, a organização, de um modo geral, não fornece um ambiente estável para desenvolvimento e manutenção de software. Quando uma organização não dispõe de práticas de gestão bem estabelecidas, os benefícios das boas práticas de desenvolvimento de software são minados por um planejamento ineficiente e por um contexto onde os compromissos são sempre reativos. Entende-se por compromissos reativos aqueles que são firmados como reação a algum acontecimento não previsto no planejamento.

Em meio a uma crise, os projetos tipicamente abandonam os procedimentos que foram planejados e partem para a codificação e testes. O sucesso depende inteiramente de ser ter um gerente excepcional e uma equipe de software madura e eficiente. Ocasionalmente, gerentes de software eficientes, que dispõem de poder, podem resistir às pressões de recorrer a atalhos no processo de software; contudo, quando deixam o projeto, a influência estabilizadora se vai com eles. Mesmo um forte processo de desenvolvimento não pode superar a instabilidade criada pela ausência de práticas sólidas de gestão.

A capacidade de processo de software em organizações de Nível 1 é imprevisível porque o processo de software é constantemente alterado à medida que o trabalho progride (ou seja, o processo é “*ad hoc*”). Cronogramas, orçamentos, funcionalidades e qualidade do produto são geralmente imprevisíveis. O desempenho depende da capacidade dos indivíduos e varia com as suas habilidades, conhecimentos e motivações inatas. Raros são os processos de software estáveis em evidência e o desempenho só pode ser previsto através de habilidades individuais e não por meio da capacidade da organização.

2.1.2 Nível 2 – O Nível Repetível

No Nível Repetível, as políticas de gestão de projeto de software e os procedimentos para implementá-las são estáveis. O planejamento e a gestão de novos projetos são baseados na experiência adquirida em projetos similares. Um dos objetivos alcançados no nível 2 é a institucionalização dos processos para os projetos de software, possibilitando que as organizações repitam as práticas bem sucedidas de envolvidas em projetos anteriores, apesar dos processos específicos implementados pelos projetos serem diferentes. Um processo efetivo pode ser caracterizado como hábil, documentado, robusto, treinado, medido e com capacidade de poder ser melhorado.

Os projetos nas organizações de nível 2 possuem controles básicos de gestão de software. Os compromissos realistas do projeto são baseados em resultados observados em projetos anteriores e nos requisitos do projeto atual. Os gerentes de software do projeto acompanham custos, cronogramas e funcionalidades do software; os problemas com compromissos são identificados quando surgem. Os requisitos e os produtos de trabalho de software desenvolvidos para satisfazê-los são armazenados

Os Cinco Níveis da Maturidade do Processo de Software

de forma criteriosa e a integridade dos mesmos é controlada. Os padrões do projeto de software são definidos e a organização garante que eles sejam seguidos fielmente. Existindo subcontratados, o projeto de software trabalha em conjunto com os mesmos, visando fortalecer a relação cliente-fornecedor.

A capacidade de processo de software das organizações de nível 2 pode ser resumida como sendo disciplinada, uma vez que o planejamento e o acompanhamento de projeto de software são estáveis e os sucessos mais recentes podem ser repetidos. Os processos estão sob um controle efetivo de um sistema de gestão de projeto, seguindo planos realistas baseados no desempenho de projetos anteriores.

2.1.3 Nível 3 – O Nível Definido

No Nível Definido, o processo padrão de desenvolvimento e manutenção de software global da organização é documentado, inclusive o desenvolvimento de software e a gestão de processos, estando estes últimos integrados em um todo coerente. Esse processo padrão é referido em todo o modelo CMM como “processo de software padrão da organização”. Os processos no nível 3 são utilizados (e alterados, quando apropriado) para apoiar os gerentes de software e o pessoal técnico, contribuindo para que possam atuar de forma mais efetiva. Ao padronizar seus processos de software, a organização exercita práticas de engenharia de software efetivas. Existe uma equipe responsável pelas atividades de processo de software da organização, por exemplo, a equipe de processo de engenharia de software, ou SEPG (Software Engineering Process Group) [Fowler90]. Um programa de treinamento é implementado em toda a organização para garantir que os gerentes e funcionários tenham os conhecimentos e as habilidades necessárias ao cumprimento de suas funções.

Os projetos adaptam o processo de software padrão da organização para desenvolver seus próprios processos de software definidos, que consideram as características únicas de cada projeto. Esse processo concebido é referido no modelo CMM como “processo de software definido do projeto”. Um processo de software definido contém uma integração coerente do desenvolvimento de software com os processos de gestão bem definidos. Um processo bem definido pode ser caracterizado como possuidor de critérios de disponibilidade, entradas, padrões e procedimentos para realização do trabalho, mecanismos de verificação (tais como revisões por pares), saídas e critérios de finalização. Como o processo de software é bem definido, a gerência tem uma boa percepção do progresso técnico em todos os projetos.

A capacidade de processo de software das organizações de nível 3 pode ser resumida como sendo padronizada e consistente porque tanto as atividades de gestão como a engenharia de software são estáveis e repetíveis. Nas linhas de produtos estabelecidas, a qualidade de software é acompanhada, além do custo, cronograma e funcionalidades estarem sob controle. Essa capacidade de projeto é baseada em um

Os Cinco Níveis da Maturidade do Processo de Software

entendimento global da organização com relação as atividades, regras e responsabilidades decorrentes do processo de software definido.

2.1.4 Nível 4 – O Nível Gerenciado

No Nível Gerenciado, a organização estabelece metas quantitativas de qualidade para os produtos e processos de software. A produtividade e a qualidade são medidas nas atividades importantes de processo de software em todos os projetos, como parte de um programa organizacional de medições. Um banco de dados de processo de software, que abrange a organização toda, é utilizado para coletar e analisar os dados disponíveis dos processos de software definidos dos projetos. No nível 4, os processos de software são instrumentalizados com medições consistentes e bem definidas. Essas medições estabelecem as bases para avaliar os processos e os produtos de software do projeto.

Os projetos possuem controle sobre seus produtos e processos, reduzindo a variação no desempenho desses últimos, para atingir limites quantitativos aceitáveis. As variações significativas no desempenho dos processos podem ser distinguidas das variações aleatórias (ruídos), particularmente dentro de linhas de produtos estabelecidas. Os riscos decorrentes da introdução de um novo domínio de aplicação são conhecidos e cuidadosamente gerenciados.

A capacidade de processo de software das organizações de Nível 4 pode ser resumida como sendo previsível porque o processo é medido e opera dentro de limites mensuráveis. A capacidade de processo desse nível permite que a organização preveja as tendências na qualidade dos produtos e dos processos dentro das fronteiras quantitativas desses limites. Quando esses limites são excedidos, são tomadas providências para corrigir a situação. Como era de se esperar, os produtos de software são de alta qualidade.

2.1.5 Nível 5 – O Nível Em Otimização

No Nível Em Otimização, toda a organização está voltada para a melhoria contínua de processo. A organização tem meios de identificar as oportunidades de melhoria e fortalecer o processo de maneira pró-ativa, com o objetivo de prevenir a ocorrência de falhas. Os dados sobre a efetividade de processo de software são utilizados para realizar análises de custo benefício das novas tecnologias e das mudanças propostas no processo de software da organização. As inovações decorrentes das melhores práticas de engenharia de software são identificadas e transferidas para a organização toda.

As equipes de projeto de software das organizações de Nível 5 analisam as falhas para determinar suas causas. Os processos de software são revistos para prevenir tipo de falhas já conhecidos que normalmente se repetem e as lições aprendidas são disseminadas para outros projetos.

Os Cinco Níveis da Maturidade do Processo de Software

A capacidade de processo de software das organizações de Nível 5 pode ser resumida como sendo de melhoria contínua porque estas estão continuamente se esforçando para melhorar a abrangência de sua capacidade de processo, melhorando dessa forma o desempenho dos processos de seus projetos. As melhorias ocorrem através de avanços incrementais nos processos já existentes e através de inovações que utilizam novos métodos e tecnologias.

2.2 Entendendo os Níveis de Maturidade

O CMM é um modelo descritivo, no sentido de descrever atributos essenciais (ou chave) que seriam esperados para caracterizar uma organização em um nível particular de maturidade. É um modelo normativo, no sentido de que as práticas detalhadas caracterizam os tipos normais de comportamento que seriam esperados em uma organização que desenvolve projetos em larga escala num contexto de contratação governamental. A intenção é que o CMM tenha um nível suficiente de abstração que não restrinja desnecessariamente a maneira que o processo de software é implementado pela organização; ele simplesmente descreve o que normalmente seria esperado dos atributos essenciais do processo de software.

Em qualquer contexto em que o CMM for aplicado, deveria ser utilizada uma interpretação razoável das práticas. O CMM deve ser interpretado apropriadamente, utilizando-se o conhecimento de peritos quando o ambiente comercial da organização difere significativamente de uma grande organização fornecedora.

O CMM não é prescritivo; ele não diz à organização como melhorar. O CMM descreve a organização em cada nível de maturidade sem prescrever os meios específicos para conseguí-lo. Pode levar vários anos para se passar do Nível 1 para o Nível 2; a movimentação entre os outros níveis geralmente levará cerca de dois anos.

A melhoria de processo de software ocorre dentro do contexto dos planos e estratégicos e dos objetivos de negócio da organização, da sua estrutura organizacional, das tecnologias em uso, da sua cultura social e sistema de gestão. O CMM está voltado para os aspectos de processo da Gestão da Qualidade Total; a melhoria de processo bem sucedida implica que os aspectos fora do escopo de processo de software também sejam encaminhados (por exemplo: as questões pessoais envolvidas nas mudanças da cultura organizacional que possibilitem a implementação e a institucionalização das melhorias de processo).

2.2.1 Entendendo o Nível Inicial

Embora as organizações de Nível 1 sejam frequentemente caracterizadas como tendo processos “*ad hoc*”, caóticos mesmo, elas desenvolvem produtos que funcionam, apesar de poderem estar com o orçamento e com o cronograma estoura-

Os Cinco Níveis da Maturidade do Processo de Software

dos. O sucesso das organizações de Nível 1 depende da competência e heroísmo das pessoas que atuam na organização. A seleção, contratação, desenvolvimento e a retenção de pessoas competentes são questões significativas para as organizações de todos os níveis de maturidade, mas estão fora do escopo do CMM.

2.2.2 Entendendo dos Níveis Repetível e Definido

À medida que os projetos crescem em tamanho e complexidade, a atenção se desloca das questões técnicas para as questões organizacionais e administrativas – o foco da maturidade do processo [Siegel90, DoD87, GAO-92-48]. O processo possibilita que as pessoas trabalhem mais efetivamente na organização, no sentido de incorporar aos processos formais (documentados) as lições aprendidas pelo melhor grupo de trabalho. Dessa forma, as habilidades necessárias para executar eficientemente os processos são construídas (geralmente através de treinamento) e melhoradas continuamente, aprendendo-se com as pessoas que estão realizando o trabalho.

Para alcançar o Nível 2, a gerência deve se concentrar em seus próprios processos a fim de conseguir um processo de software disciplinado. O Nível 2 fornece as bases necessárias para o Nível 3 porque o objetivo é atuar na melhoria dos seus processos, antes de cuidar das questões técnicas e organizacionais do Nível 3. A gerência estabelece uma posição de liderança ao alcançar o Nível 2 através da documentação e do acompanhamento dos processos de gestão do projeto.

Os processos podem diferir entre os projetos das organizações de Nível 2; o requisito organizacional para alcançar o Nível 2 é que existam políticas para orientar os projetos no estabelecimento de processos de gestão apropriados. Os procedimentos formais (documentados) fornecem embasamento para processos consistentes que podem ser institucionalizados na organização, através de treinamento e de garantia da qualidade de software.

Sobre essa base de gestão de projeto, o Nível 3 se estabelece, definindo, integrando e documentando o processo de software completo. Integração, neste caso, significa que a saída de uma tarefa flui suavemente para a entrada da próxima. Não havendo casamento entre as tarefas, essas são identificadas e tratadas ainda nas etapas de planejamento do processo de software, ao invés de só serem detectadas na execução do processo. Um dos desafios do Nível 3 é a elaboração de processos que autorizem as pessoas a realizar o trabalho sem serem excessivamente rígidos [Humphrey 91b].

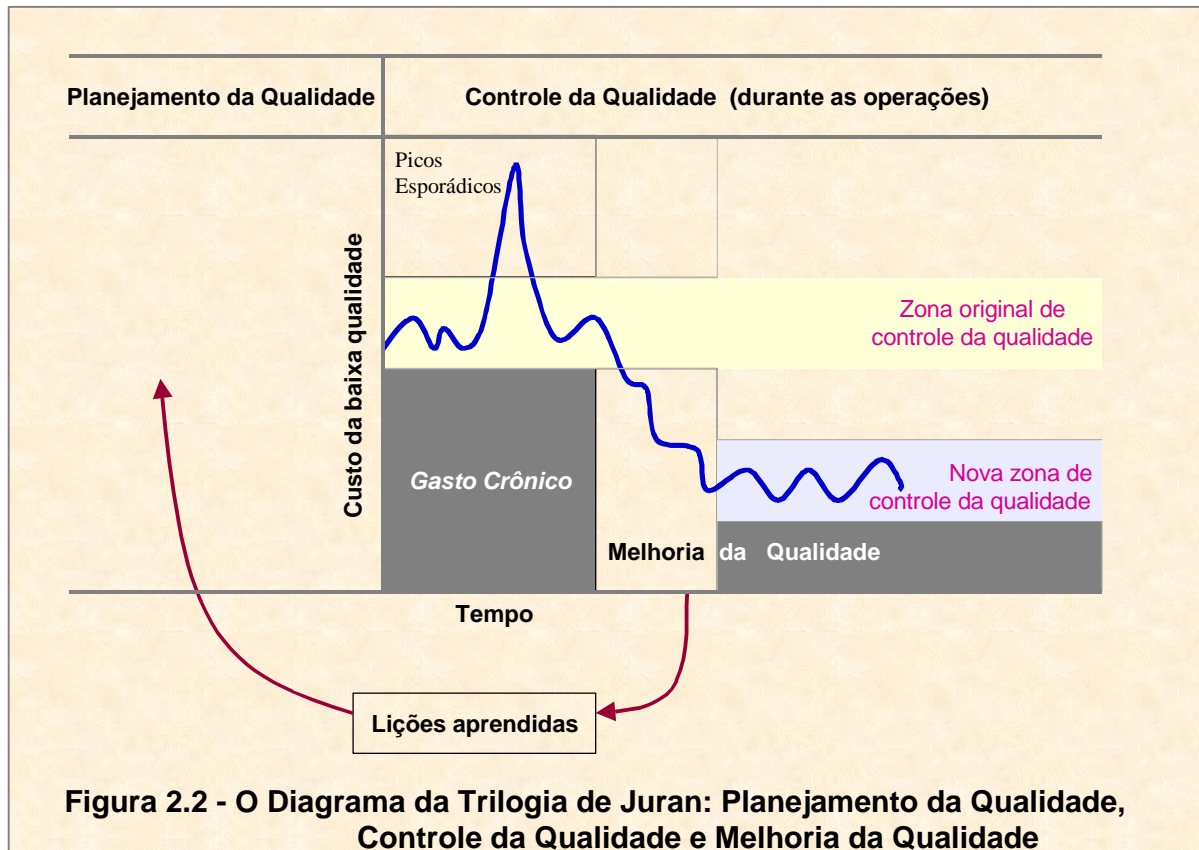
2.2.3 Entendendo os Níveis Gerenciado e Em Otimização

Os Níveis de maturidade 4 e 5 são territórios relativamente desconhecidos para a indústria de software. Poucos são os exemplos de projetos e organizações de software de Nível 4 e 5; existindo, portanto, poucos exemplos para se extrair conclusões gerais sobre as características dessas organizações. As características desses

Os Cinco Níveis da Maturidade do Processo de Software

níveis têm sido definidas por analogia a outros tipo de indústrias e aos poucos exemplos na indústria de software que possuem esse nível de capacidade de processo.

Muitas características dos Níveis 4 e 5 são baseadas em conceitos de controle de processo estatístico, como exemplificado na Figura 2.2. O Diagrama da Trilogia Juran [Juran88] ilustra os objetivos básicos da gestão de processos.



Juran divide a gestão da qualidade em três processos gerenciais básicos [Juran88]. O propósito do planejamento da qualidade é fornecer às forças operacionais, isto é, aos produtores de software, condições de desenvolver produtos que vão de encontro às necessidades do cliente. As forças operacionais desenvolvem o produto mas existe a necessidade de algum re-trabalho devido às deficiências de qualidade. Este custo é crônico porque o processo foi planejado dessa forma; o controle de qualidade é realizado para prevenir e evitar que as coisas fiquem ainda pior. Os picos e esporádicos dentro do processo, como mostrado na Figura 2.2, representam atividades de conhecidas como “apagar incêndios”. O custo crônico fornece a oportunidade de melhoria; aproveitar essa oportunidade é entendido como melhoria da qualidade.

A primeira responsabilidade, e o foco do Nível 4, é o controle de processo. O processo de software é gerenciado, operando de forma estável dentro da zona de

Os Cinco Níveis da Maturidade do Processo de Software

controle de qualidade. Há inevitavelmente algum custo crônico e pode haver picos nos resultados medidos que precisam ser controlados, mas o sistema como um todo geralmente é estável. Esse é o ponto onde o conceito de controle de causas especiais de desvios se aplica. Uma vez que o processo é estável e medido, quando ocorre alguma circunstância excepcional, a “causa especial” do desvio pode ser identificada e tratada.

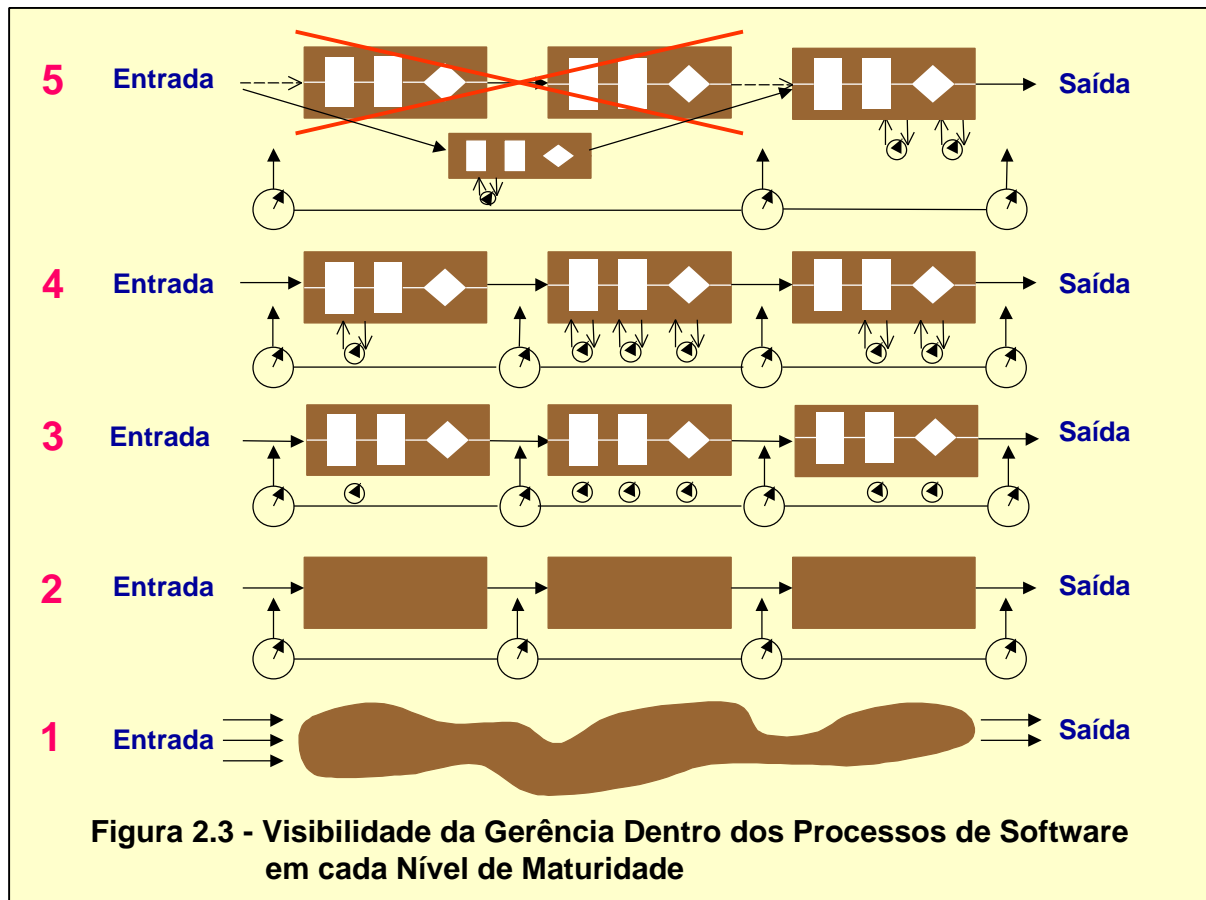
A segunda responsabilidade, e foco do Nível 5, é a melhoria contínua do processo. O processo de software é alterado para melhorar a qualidade e, em consequência, a zona de controle de qualidade se move. Um novo padrão de referência para desempenho é estabelecido, reduzindo o custo crônico. Conhecimentos adquiridos com a melhoria de processos são aplicados no planejamento de processos futuros. Este é o ponto onde o conceito de tratamento de causas comuns de desvios se apresenta. Existe custo crônico em qualquer sistema, na forma de re-trabalho, simplesmente devido aos desvios aleatórios. Custo adicional é inaceitável; os esforços organizados para eliminá-los resultam na alteração do sistema, isto é, na melhoria do processo através de alteração das “causas comuns” de ineficiência, objetivando prevenir a ocorrência de custos imprevistos.

É previsto que as organizações que alcançam os níveis de maturidade mais elevados do CMM possuam um processo capaz de produzir software extremamente confiável dentro de limites de custo e de cronograma previsíveis. À medida que cresce o entendimento dos níveis de maturidade mais elevados, as áreas-chave de processo existentes vão sendo redefinidas e outras ainda podem ser adicionadas ao modelo. O CMM é derivado de idéias sobre processos inspirados na manufatura. Contudo, os processos de software não são caracterizados pela fabricação de réplicas como os processos da manufatura. O processo de software é caracterizado pela geração de projetos, sendo uma atividade de conhecimento intensivo [Curtis88].

2.3 Visibilidade Interna do Projeto de Software

Os engenheiros de software possuem uma visão detalhada da situação interna do projeto porque possuem as informações, em primeira mão, sobre a situação e o desempenho do mesmo. Entretanto, em projetos grandes, suas percepções geralmente são extraídas apenas de experiências pessoais em suas áreas de atuação. Aqueles que não participam do projeto e que não têm contato como os gerentes superiores não possuem a visibilidade interna dos processos e contam com as revisões periódicas para obter as informações de que necessitam para monitorar o progresso. A Figura 2.3 ilustra o nível de visibilidade interna da situação e do desempenho do projeto alcançado em cada nível de maturidade do processo. Cada nível de maturidade sucessivo fornece melhor visibilidade interna do processo de software.

Os Cinco Níveis da Maturidade do Processo de Software



No Nível 1, o processo de software é uma entidade amorfa – uma caixa preta – e a visibilidade interna dos processos do projeto é limitada. Uma vez que a preparação das atividades é definida de forma precária, os gerentes passam por fases extremamente difíceis para estabelecer a situação do progresso e das atividades do projeto⁵.

Os requisitos fluem no interior do processo de uma forma descontrolada e surge o produto. O desenvolvimento de software é frequentemente visualizado como uma espécie de “magia negra”, especialmente por gerentes não familiarizados com software.

No Nível 2, os requisitos do cliente e os produtos de trabalho são controlados, sendo que as práticas básicas de gestão de projeto estão estabelecidas. Esses controles da gestão possibilitam a visibilidade interna do projeto em momentos definidos. O processo de construção de software pode ser visualizado como uma sucessão de caixas pretas, permitindo a visibilidade da gestão nos pontos de transição como fluxos

⁵ Isso conduz à Regra Noventa-Noventa: 90% do projeto é realizado em 90% do tempo.

Os Cinco Níveis da Maturidade do Processo de Software

de atividades entre as caixas (marcos do projeto). Mesmo que a gerência não conheça detalhes do que está acontecendo dentro da caixa, os produtos e os pontos de verificação dos processos são identificados e conhecidos, através dos quais pode-se confirmar que o processo está funcionando. A gerência reage aos problemas quando os mesmos ocorrem.

No Nível 3, a estrutura interna das caixas, isto é, as tarefas dentro do processo de software definido, é visível. A estrutura interna representa a maneira que o processo de software padrão é aplicado aos projetos específicos. Tanto os gerentes como os engenheiros compreendem seus papéis e responsabilidades dentro do processo e entendem como suas atividades se interagem, em um nível apropriado de detalhes. A gerência se prepara de maneira pró-ativa para os riscos que possam surgir. As pessoas que não participam diretamente do projeto podem obter uma atualização rápida e precisa sobre a situação do mesmo porque os processos definidos permitem grande visibilidade dentro das atividades do projeto.

No Nível 4, os processos de software definidos são instrumentalizados e controlados quantitativamente. Os gerentes são capazes de medir os progressos e os problemas. Eles possuem bases objetivas e quantitativas para tomada de decisões. Suas habilidades de prever resultados crescem constantemente, tornando-se mais precisas à medida que a variação no processo diminui.

No Nível 5, maneiras novas e aprimoradas de construir software são continuamente experimentadas, de uma forma controlada, para melhorar a produtividade e a qualidade. Mudança disciplinada é o comportamento usual quando atividades ineficientes ou que tendem a apresentar defeitos são identificadas. Elas são substituídas ou revistas. A percepção se estende além dos processos existentes, chegando aos efeitos das potenciais mudanças nos processos. Os gerentes são capazes de estimar e acompanhar quantitativamente o impacto e a eficiência da mudança.

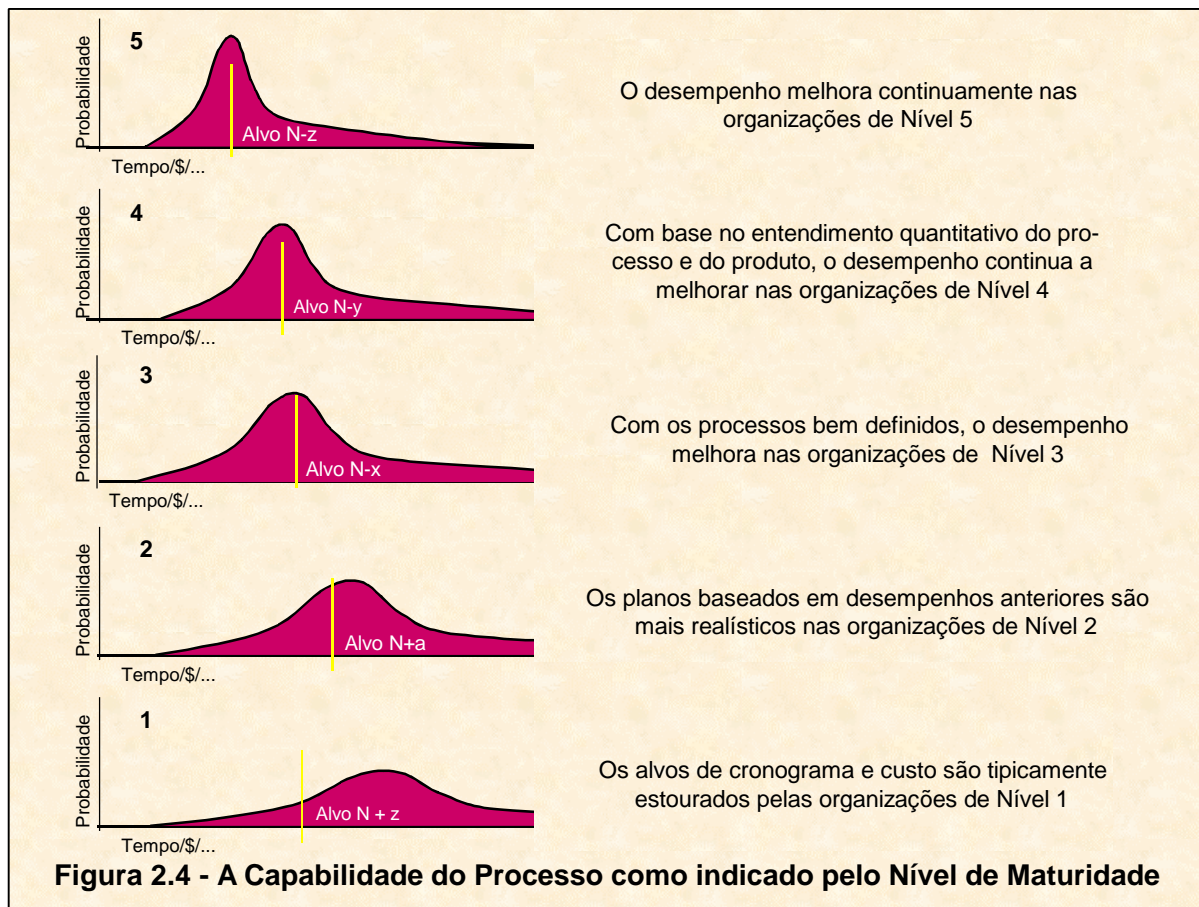
2.4 Capabilidade do Projeto e a Previsão de Desempenho

A maturidade do processo de software de uma organização ajuda a se prever a habilidade do projeto em atingir as suas metas. Os projetos nas organizações de Nível 1 apresentam grandes desvios em relação às metas de custo, cronograma, funcionalidade e qualidade. Como pode ser visto na Figura 2.4, três melhorias relacionadas ao alcance de metas são observadas à medida que o processo de software da organização vai se tornando mais maduro.

Primeiro, à medida que a maturidade cresce, a diferença entre os resultados esperados e os resultados reais diminui nos projetos. Por exemplo, se dez projetos de mesmo tamanho fossem programados para serem entregues em 1^o de maio, então a data média de cada entrega estaria mais próxima de 1^o de maio, tanto quanto mais madura fosse a organização. As organizações de Nível 1 frequentemente atrasam

Os Cinco Níveis da Maturidade do Processo de Software

muito suas datas de entrega originalmente previstas, enquanto que as organizações de Nível 5 seriam capazes de prever essas datas com considerável precisão. Isso acontece porque as organizações de Nível 5 utilizam um processo operacional de software cuidadosamente elaborado dentro de parâmetros conhecidos, sendo a escolha da data de entrega baseada no grande volume de dados que possuem sobre seus processos e em suas habilidades de tirar proveito do mesmo. (Isso é ilustrado na Figura 2.4 através da parte da área sob a curva que se encontra à direita da data de entrega.)



Segundo, à medida que a maturidade cresce, a variação dos resultados reais em torno dos resultados esperados diminui. Por exemplo, as datas de entrega nas organizações de Nível 1, em projetos de tamanhos similares, são imprevisíveis e muito variáveis. Projetos similares em organizações de Nível 5, entretanto, serão entregues dentro de um intervalo de tempo muito menor. Isso ocorre nos níveis de maturidade mais elevados porque, na prática, todos os projetos são executados dentro de parâmetros controlados, aproximando-se da capacidade do processo para custo, cronograma, funcionalidade e qualidade. (Isso é ilustrado na Figura 2.4 através da parte da área sob a curva concentrada próximo à linha alvo).

Os Cinco Níveis da Maturidade do Processo de Software

Terceiro, os resultados esperados melhoram à medida que a maturidade da organização aumenta. Isto é, à medida que a organização de software amadurece, os custos diminuem, o tempo de desenvolvimento se torna menor, e a produtividade e a qualidade aumentam. Numa organização de Nível 1, o tempo de desenvolvimento pode ser muito longo devido ao volume de re-trabalho necessário para correção de erros. Contrastando com isso, as organizações de Nível 5 utilizam melhoria contínua de processo e técnicas de prevenção de defeitos para aumentar a eficiência do processo e eliminar re-trabalho oneroso, permitindo que o tempo de desenvolvimento seja reduzido. (Isso é ilustrado na Figura 2.4 através do deslocamento horizontal da linha alvo em relação à origem).

Os aprimoramentos na previsão dos resultados do projeto representados na Figura 2.4 assumem que os resultados do projeto de software se tornam mais previsíveis à medida que a interferência, na forma de re-trabalho, é eliminada. Sistemas sem precedentes complicam o quadro, uma vez que as novas tecnologias e as novas aplicações reduzem a capacidade do processo através do aumento da variabilidade. Mesmo no caso de sistemas sem precedentes, as características das práticas de gestão e de engenharia de muitas organizações maduras ajudam a identificar e encaminhar os problemas no ciclo de desenvolvimento antes mesmo de serem detectados em organizações menos maduras. A detecção antecipada de defeitos contribui para a estabilidade e desempenho do projeto, eliminando o re-trabalho durante as fases posteriores. A gestão de riscos é uma parte integrante da gestão de projeto num processo maduro. Em alguns casos, um processo maduro pode significar que projetos “fadados ao fracasso” sejam identificados mais cedo no ciclo de vida de software e o investimento em causa perdida seja minimizado.

2.5 Pulando de Níveis de Maturidade

Os níveis de maturidade no CMM descrevem as características de uma organização em um determinado nível de maturidade. Cada nível constrói a base para os níveis seguintes alavancarem a implementação de processos de forma efetiva e eficiente. As organizações podem, entretanto, usar proveitosamente os processos descritos em um nível de maturidade mais elevado que aquele em que se encontram. A engenharia de processos, tal como a análise de requisitos, projeto, codificação e teste, não é discutida até o nível 3, ainda que até mesmo as organizações de Nível 1 devam realizar essas atividades. As organizações de Nível 1 ou Nível 2 podem ser capazes de realizar revisões por pares (Nível 3), fazer análise de Pareto (Nível 4) ou experimentar novas tecnologias (Nível 5) proveitosamente. Ao prescrever os passos que uma organização deveria trilhar para passar do Nível 1 para o Nível 2, freqüentemente uma das recomendações é estabelecer um conjunto de processos de desenvolvimento de software; isso é que é um atributo das organizações de Nível 3. Embora seja o foco do Nível 4, “medição” também é parte integrante dos níveis de maturidade mais baixos.

Os Cinco Níveis da Maturidade do Processo de Software

Entretanto, esses processos não podem alcançar seus potenciais completos antes que uma base adequada seja estabelecida. As revisões por pares não podem ser totalmente efetivas, por exemplo, a menos que sejam implementadas de forma consistente, mesmo quando os “incêndios” colocam o projeto em risco. Os níveis de maturidade descrevem os problemas que predominam em cada nível. Os problemas predominantes nas organizações de Nível 1 são gerenciais; os outros problemas tendem a ser encobertos pelas dificuldades de planejamento e gestão dos projetos de software.

Pular níveis de maturidade é contraproducente porque cada nível constitui a base necessária a partir da qual se alcança o próximo nível. O CMM identifica os níveis através dos quais uma organização deve se desenvolver para estabelecer uma cultura de excelência em desenvolvimento de software. Os processos sem as bases adequadas falham no ponto extremo onde estão mais carentes – sob *stress* – e não fornecem bases para melhorias futuras.

Uma organização de Nível 1 que está tentando implementar o *processo definido* (Nível 3), antes de ter estabelecido o *processo repetível* (Nível 2), geralmente é mal sucedida porque os gerentes de projeto estão subjulgados com as pressões de custo e cronograma. Esta é a razão fundamental para se focar em gestão de projeto antes de se implementar engenharia de processo. Pode parecer que a definição e a implementação da engenharia de processo deva acontecer antes que a gestão de processo (especialmente aos olhos do pessoal técnico), mas, sem a disciplina gerencial, a engenharia de processo fica prejudicada pelas pressões de cronograma e de custo [Humphrey88].

Uma organização que está tentando implementar o *processo gerenciado* (Nível 4), sem os fundamentos do *processo definido* (Nível 3), geralmente é mal sucedida porque não há bases comuns para a interpretação das medidas sem os processos definidos. Se, por um lado, os dados podem ser coletados individualmente pelos projetos, por outro, poucas medidas são significativas no âmbito de todos os projetos, não tendo condições de fazer crescer substancialmente o entendimento do processo de software.

Uma organização que está tentando implementar um processo otimizado (Nível 5), sem as bases do *processo gerenciado* (Nível 4), provavelmente está fadada ao fracasso devido a falta de entendimento do impacto das mudanças no processo. Sem controlar o processo dentro de fronteiras estatisticamente estreitas (pequenas variações nas medidas dos processos), existe muita interferência nos dados para que se possa determinar objetivamente se uma dada melhoria de processo tem algum efeito. As decisões podem se degenerar em conflitos subjetivos porque existe pouco fundamento quantitativo para se tomar decisões racionais.

O esforço na melhoria de processo deveria se concentrar nas necessidades da organização com relação ao contexto do seu ambiente de negócio. A habilidade de

Os Cinco Níveis da Maturidade do Processo de Software

implementar processos de altos níveis de maturidade não implica que os níveis de maturidade inferiores possam ser pulados.

3 Definição Operacional do Modelo de Maturidade de Capacabilidade

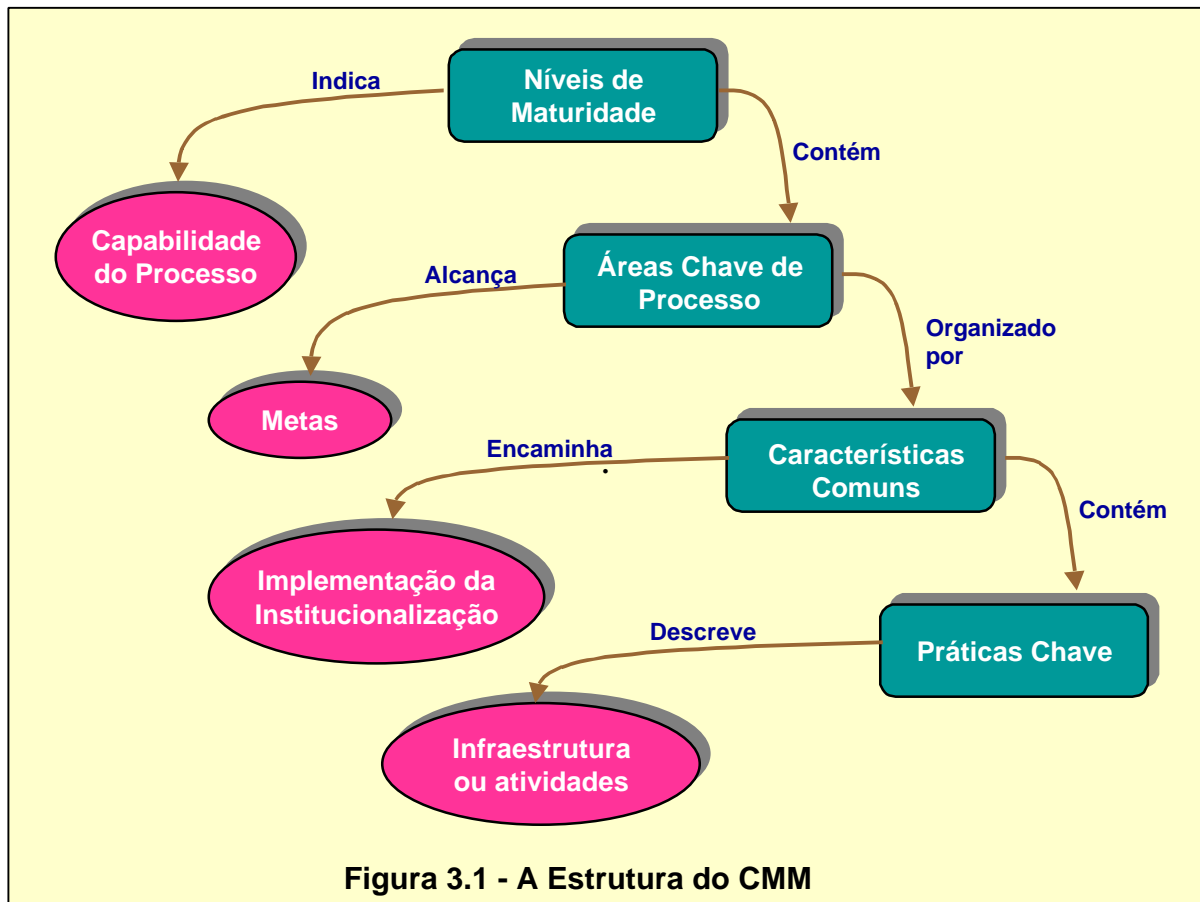
O CMM é uma estrutura que representa um caminho para melhoria de processos, recomendada para organizações de software que desejam melhorar seus processos de software. Esta estrutura operacional do CMM é projetada para dar suporte às suas várias formas de utilização. Há, pelo menos, quatro modalidades de aplicação previstas para o CMM:

- As equipes de avaliação poderão utilizar o CMM para identificar pontos fortes e oportunidades de melhoria na organização.
- As equipes de avaliação poderão utilizar o CMM para identificar riscos na seleção de diferentes prestadores de serviço, tanto para a concessão de negócios, quanto para o monitoramento de contratos.
- Gerentes e técnicos poderão utilizar o CMM para entender as atividades necessárias ao planejamento e implementação de um programa de melhorias no processo de software de suas organizações.
- Os grupos de melhoria de processo, tais como o Grupo de Processos (SEPG), poderão utilizar o CMM como um guia para ajudá-los a definir e melhorar o processo de software de suas organizações.

Devido a essas diversas modalidades de uso, o CMM deve ser decomposto em detalhes suficientes para que recomendações reais sobre o processo possam ser derivadas a partir da estrutura dos níveis de maturidade. Essa decomposição também indica os processos e a estrutura dos mesmos que caracteriza a sua maturidade e capacidade.

3.1 Estrutura Interna dos Níveis de Maturidade

Cada nível de maturidade foi decomposto em partes menores. Com exceção do Nível 1, a decomposição de cada nível de maturidade estende-se desde um resumo de cada nível até sua definição operacional nas práticas-chave, como mostrado na figura 3.1. Cada nível de maturidade é composto de várias áreas-chave de processo. Cada área-chave de processo é organizada em cinco seções denominadas *características comuns*. As características comuns especificam as práticas-chave que, quando tratadas coletivamente, atingem as metas previstas para a área-chave de processo.



3.2 Níveis de Maturidade

Um nível de maturidade representa um estágio evolutivo bem definido com relação à obtenção de um processo de software maduro. Cada nível de maturidade indica um nível de capacidade de processo, com ilustrado na figura 2.1. Por exemplo, no Nível 2 a capacidade de processo de uma organização foi elevada de uma situação “ad-hoc” para uma situação disciplinada, por meio do estabelecimento de sólidos controles de gestão de projeto.

3.3 Áreas-chave de Processo

Com exceção do Nível 1, cada nível de maturidade é decomposto em várias áreas-chave de processo, que indicam as áreas nas quais uma organização deveria focar seus esforços para a melhoria de seu processo de software. As áreas-chave de processo identificam os assuntos que devem ser tratados para se obter um determinado nível de maturidade.

Definição Operacional do Modelo de Maturidade de Capabilidade

Cada área-chave de processo identifica um grupo de atividades relacionadas, as quais, quando executadas coletivamente, atingem um conjunto de metas consideradas importantes para aumentar a capacidade do processo. As áreas-chave de processo foram definidas para residir em um único nível de maturidade, como mostrado na figura 3.2. A forma de atingir as metas de uma área-chave de processo pode diferir entre projetos, de acordo com as diferenças de domínios de aplicação ou de ambientes. Todavia, todas as metas de uma área-chave de processo devem ser atingidas para que uma organização satisfaça essa área-chave. Quando as metas de uma área-chave são cumpridas de maneira continuada em um projeto, a organização pode ser considerada como tendo institucionalizado a capacidade do processo caracterizada por essa área-chave.

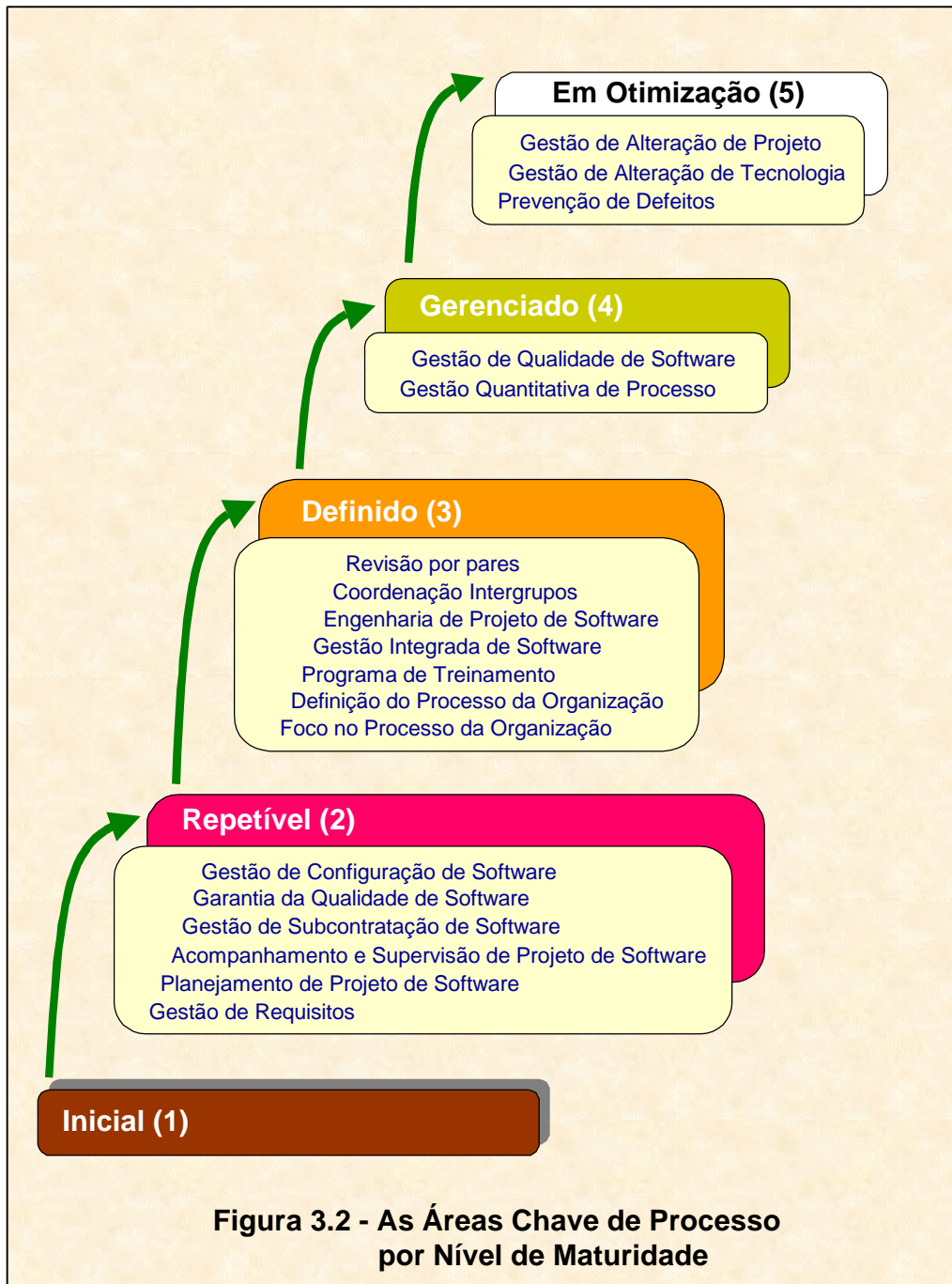
O adjetivo “chave” indica que existem áreas de processo (e processos) que não são imprescindíveis para a obtenção de um determinado nível de maturidade. O CMM não descreve em detalhe todas as áreas de processo que estão relacionadas com desenvolvimento e manutenção de software. Algumas áreas de processo foram identificadas como determinantes da capacidade do processo, sendo essas descritas no CMM.

Embora outros assuntos afetem o desempenho de um processo, as áreas-chave de processo foram identificadas em função de sua eficácia e eficiência na melhoria da capacidade do processo de uma organização. Elas podem ser consideradas como sendo os requisitos para a obtenção de um nível de maturidade. A figura 3.2 mostra as áreas-chave de processo para cada nível de maturidade. Para se obter um certo nível de maturidade, as áreas-chave de processo daquele nível devem ser satisfeitas. Para satisfazer uma área-chave de processo, cada uma das metas dessa área-chave deve ser satisfeita. As metas resumem as práticas-chave de uma área-chave de processo e podem ser utilizadas para determinar se uma organização, ou projeto, efetivamente implementou essa área-chave. As metas traduzem o escopo (abrangência), os limites e a intenção de cada área-chave de processo⁶.

As práticas específicas a serem executadas em cada área-chave de processo evoluirão de acordo com a obtenção, por parte da organização, de níveis mais elevados de maturidade. Por exemplo, a maioria das capacidades de estimativas de projeto descritas na área-chave de processo *Planejamento de Projeto de Software*, do Nível 2, deve evoluir para tratar os dados adicionais de projeto disponíveis nos Níveis 3, 4 e 5. A *Gestão Integrada de Projeto*, no Nível 3, é uma evolução do *Planejamento de Projeto de Software* e do *Acompanhamento e Supervisão de Projeto*, do Nível 2, à medida que o projeto é gerido utilizando um processo de software definido.

⁶ Para uma lista das metas da cada área-chave de processo, veja Apêndice A.

Definição Operacional do Modelo de Maturidade de Capacidade



As áreas-chave de processo do CMM representam um meio de descrever o grau de maturidade de uma organização. Estas áreas-chave de processo foram definidas com base em anos de experiência em gerência e desenvolvimento de software e em cinco anos de experiência com avaliação de processo e de capacidade de software.

Definição Operacional do Modelo de Maturidade de Capacidade

As áreas-chave de processo do Nível 2 são focadas em assuntos de projeto de software relacionados com o estabelecimento de controles básicos de gerência de projeto. As descrições de cada área-chave do Nível 2 são apresentadas a seguir:

- O propósito da **Gestão de Requisitos** é estabelecer um entendimento comum entre o cliente e o projeto de software sobre os requisitos do cliente que serão tratados pelo projeto. Este acordo com o cliente é a base para o planejamento (como descrito em *Planejamento de Projeto de Software*) e para a gerência do projeto de software (como descrito em *Acompanhamento de Projeto de Software*). A coordenação do relacionamento com o cliente depende da observância de um efetivo processo de controle de alteração (como descrito em *Gestão de Configuração de Software*).
- O propósito do **Planejamento de Projeto de Software** é estabelecer planos razoáveis para a execução das atividades de gerência e desenvolvimento de projeto de software. Estes planos constituem a base necessária para se gerenciar o projeto (como descrito em *Acompanhamento e Supervisão de Projeto de Software*). Sem planos realistas, uma gestão efetiva de projeto não pode ser implementada.
- O propósito do **Acompanhamento e Supervisão de Projeto de Software** é estabelecer a adequada visibilidade do progresso real, de tal maneira que a gerência possa tomar ações efetivas quando o desempenho do projeto sofre um desvio significativo com relação ao planejado.
- O propósito da **Gestão de Subcontratação de Software** é selecionar subcontratados qualificados e gerenciá-los de modo efetivo. Esta área-chave combina os assuntos relativos à *Gestão de Requisitos*, *Planejamento de Projeto de Software* e *Acompanhamento e Supervisão de Projeto de Software*, para obter um controle básico de gestão, juntamente com a necessária coordenação da *Garantia de Qualidade de Software* e da *Gestão de Configuração de Software*, que são aplicadas apropriadamente aos subcontratados.
- O propósito da **Garantia de Qualidade de Software** é fornecer uma gestão com visibilidade apropriada sobre os processos utilizados e produtos desenvolvidos pelo projeto de software. A *Garantia de Qualidade de Software* é parte integrante da maioria dos processos de gerência e de desenvolvimento de software.
- O propósito da **Gestão de Configuração de Software** é estabelecer e manter a integridade dos produtos do projeto ao longo de todo o ciclo de vida. A *Gestão de Configuração de Software* é parte integrante da maioria dos processos de gerência e de desenvolvimento de software.

Definição Operacional do Modelo de Maturidade de Capabilidade

As áreas-chave de processo do Nível 3 abordam questões de projeto e da organização, já que a organização estabelece uma infra-estrutura que institucionaliza processos efetivos de gerência e de engenharia de software para todos os projetos. As descrições de cada área-chave do Nível 3 são apresentadas a seguir:

- O propósito do **Foco nos Processos da Organização** é estabelecer a responsabilidade organizacional pelas atividades de processo de software que melhorem a capacidade geral dos processos da organização. O primeiro resultado das atividades do Foco sobre os Processos da Organização é um conjunto de recursos de processo de software, descritos em *Definição do Processo Organizacional*. Estes recursos são utilizados pelo projeto de software como descrito em *Gestão Integrada de Software*.
- O propósito da **Definição do Processo da Organização** é desenvolver e manter um conjunto utilizável de recursos de processo de software que melhorem o desempenho do processo nos projetos e forneça uma base para benefícios cumulativos e de longo prazo para a organização. Esses recursos fornecem uma base estável que pode ser institucionalizada por meio de mecanismos tais como treinamento, descrito em *Programa de Treinamento*.
- O propósito do **Programa de Treinamento** é desenvolver as habilidades e conhecimentos dos indivíduos, de tal modo que eles possam executar suas tarefas de modo eficaz e eficiente. Treinamento é uma responsabilidade organizacional, mas o projeto deveria identificar suas necessidades com relação às habilidades e prover o treinamento adequado, quando as necessidades do projeto são específicas.
- O propósito da **Gestão Integrada de Software** é integrar as atividades de gerência e de desenvolvimento em um processo de software coerente e definido, desenvolvido sob medida, ou seja, adaptado a partir do processo de software padrão da organização e dos recursos de processo relacionados, descritos na *Definição do Processo da Organização*. Este desenvolvimento sob medida é baseado em necessidades técnicas e no ambiente de negócio do projeto, como descrito na *Engenharia de Produto de Software*. A *Gestão Integrada de Software* evolui do *Planejamento de Projeto de Software* e do *Acompanhamento e Supervisão de Projeto de Software* do nível 2.
- O propósito da **Engenharia de Produto de Software** é executar de maneira consistente um processo de engenharia bem definido que integre todas as atividades de desenvolvimento de software para se obter produtos de software corretos e consistentes de modo eficiente e eficaz. A *Engenharia de Produto de Software* descreve as atividades técnicas do projeto, tais como, análise de requisitos, projeto, codificação e teste.
- O propósito da **Coordenação Intergrupos** é estabelecer um meio para que o grupo de desenvolvimento de software participe ativamente com outros grupos de modo que o projeto esteja mais capacitado a satisfazer as necessidades do

Definição Operacional do Modelo de Maturidade de Capabilidade

cliente, de maneira eficiente e eficaz. A *Coordenação Intergrupos* é o aspecto interdisciplinar da *Gestão Integrada de Software* que se estende para além do desenvolvimento de software; não apenas o processo de software deveria ser integrado, como também as interações do grupo de desenvolvimento com outras equipes deveriam ser coordenadas e controladas.

- O propósito da *Revisão por Pares* é remover possíveis defeitos dos produtos de software eficientemente e o mais cedo possível. Um importante efeito colatório é desenvolver um melhor entendimento sobre os produtos de software e sobre os defeitos que podem ser evitados. A *Revisão por Pares* é um método de engenharia importante e eficaz citado na *Engenharia de Produto de Software*, podendo ser implementado por meio de inspeções no estilo Fagan [Fagan86], *walkthrough* estruturado ou outros tantos métodos de revisão [Freedman90].

As áreas-chave de processo do Nível 4 focalizam o estabelecimento de um entendimento quantitativo sobre o processo e os produtos de software que estão sendo construídos. As duas áreas-chave de processo neste nível, *Gestão Quantitativa do Processo* e *Gestão da Qualidade de Software*, são altamente interdependentes, como descrito a seguir:

- O propósito da *Gestão Quantitativa do Processo* é controlar quantitativamente o desempenho do processo do projeto de software. O desempenho de um processo representa os resultados reais obtidos seguindo-se um processo de software. O foco é identificar causas especiais de variação dentro de um processo estável que possa ser medido e corrigir, quando necessário, as circunstâncias que fazem com que a variação transiente ocorra. A *Gestão Quantitativa do Processo* adiciona um programa de medição abrangente às práticas de: *Definição do Processo da Organização*, *Gestão Integrada de Software*, *Coordenação Intergrupos* e *Revisão por Pares*.
- O propósito da *Gestão da Qualidade de Software* é desenvolver um entendimento quantitativo da qualidade dos produtos de software do projeto e obter metas de qualidade específicas. A *Gestão da Qualidade de Software* aplica um programa de medição abrangente aos produtos de software descritos na *Engenharia de Produto de Software*.

As áreas-chave de processo do Nível 5 cobrem as questões que tanto a organização quanto o projeto devem tratar para implementar a melhoria de processo de software de forma contínua e mensurável. Cada uma das áreas-chave de processo para o Nível 5 são descritas a seguir:

- O propósito da *Prevenção de Defeitos* é identificar as causas de defeitos e se prevenir contra a sua recorrência. O projeto de software analisa defeitos, identifica suas causas e altera seu processo de software definido, como

Definição Operacional do Modelo de Maturidade de Capabilidade

descrito na *Gestão Integrada de Software*. As alterações de processo de valor geral são estendidas para outros projetos de software, como descrito na *Gestão de Alteração de Processo*.

- O propósito da *Gestão de Alteração de Tecnologia* é identificar novas tecnologias vantajosas (isto é, ferramentas, métodos e processos) e transferi-las para a organização de maneira ordenada, como descrito na *Gestão de Alteração de Processo*. O foco da *Gestão de Alteração de Tecnologia* é inovar de modo eficiente num contexto em constante alteração.
- O propósito da *Gestão da Alteração de Processo* é melhorar continuamente o processo de software utilizado na organização com a intenção melhorar a qualidade do software, aumentar a produtividade e reduzir o tempo de desenvolvimento do produto. A *Gestão da Alteração de Processo* utiliza as melhorias incrementais da *Prevenção de Defeitos* e as melhorias inovadoras da *Gestão de Alteração de Tecnologia* e as torna disponíveis para toda a organização.

3.4 Características Comuns

Por conveniência, as áreas-chave de processo são organizadas em características comuns. As características comuns são atributos que indicam se a implementação e a institucionalização de uma área-chave de processo são eficazes, repetíveis e duradouras. As cinco características comuns são listadas a seguir:

Compromissos Os compromissos descrevem as ações que a organização deve tomar para garantir que o processo seja estabelecido e que será duradouro. Tipicamente envolvem o estabelecimento das políticas organizacionais e o patrocínio da alta gerência.

Habilidades As habilidades descrevem as pré-condições que devem existir no projeto ou na organização para se implementar de maneira competente o processo de software. Tipicamente envolvem recursos, estruturas organizacionais e treinamento.

Atividades As atividades descrevem os papéis e os procedimentos necessários para a implementação de uma área-chave de processo. Tipicamente envolvem o estabelecimento de planos e procedimentos, a execução do trabalho, o acompanhamento do mesmo e a tomada de ações corretivas quando necessário.

Medições e As medições e análises descrevem a necessidade de se medir o

Definição Operacional do Modelo de Maturidade de Capacidade

Análises	processo e analisar as medições. Tipicamente incluem exemplos de medições que poderiam ser feitas para determinar a situação e a eficácia das atividades.
Verificação da implementação	A verificação da implementação descreve os passos para se garantir que as atividades sejam executadas em conformidade com o processo estabelecido. Tipicamente engloba revisões e auditorias pela gerência e pela garantia da qualidade de software.

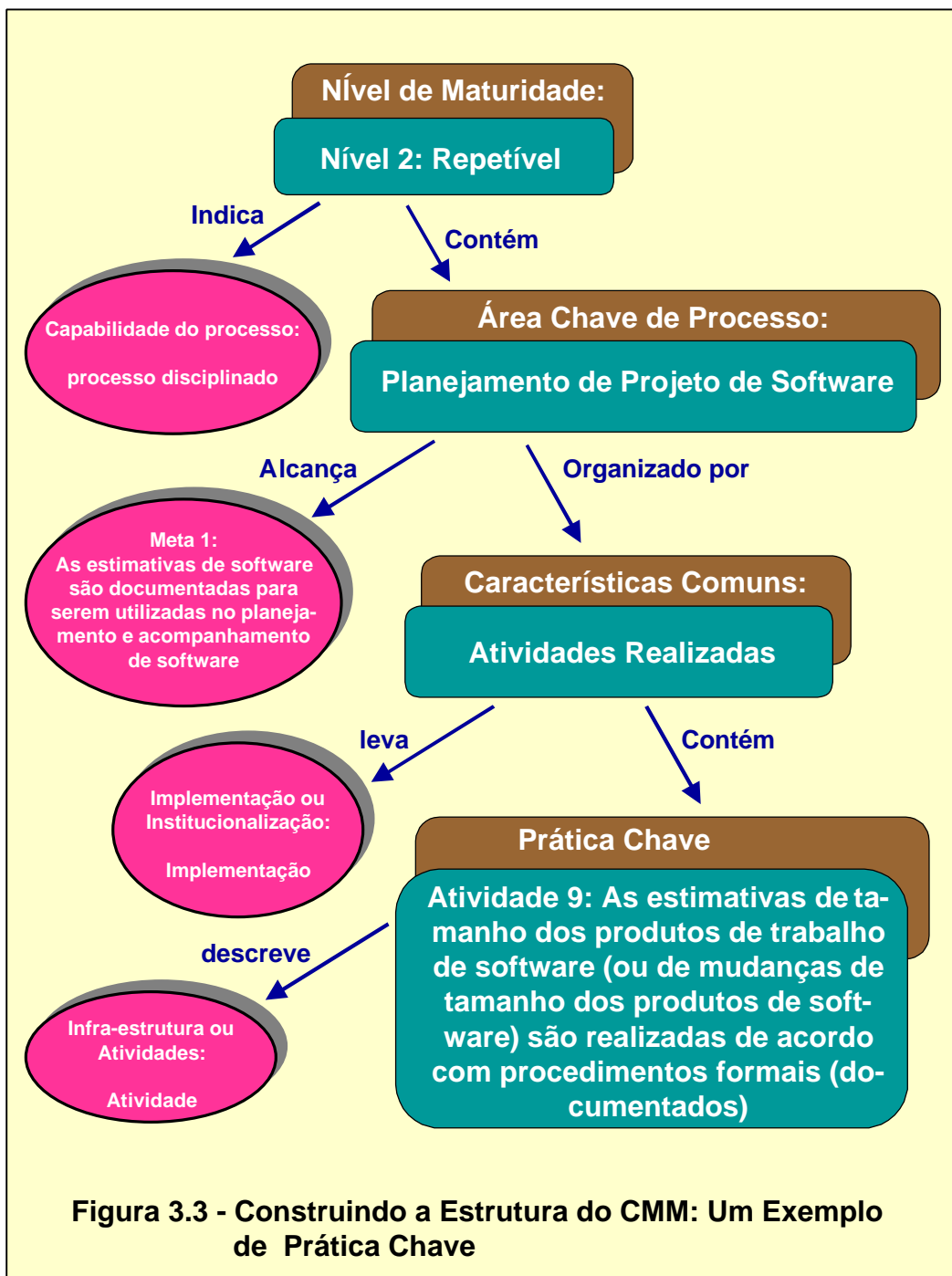
Nas características comuns, as práticas *Atividades* descrevem o que deve ser implementado para se estabelecer a capacidade do processo. As outras práticas, como um todo, formam a base na qual a organização pode institucionalizar as práticas descritas na característica comum *Atividades*.

3.5 Práticas-chave

Cada área-chave de processo é descrita em termos de práticas-chave que contribuem para satisfazer suas metas. As práticas-chave descrevem a infra-estrutura e atividades que mais contribuem para a implementação e institucionalização eficazes da área-chave de processo.

Cada prática-chave consiste de uma sentença simples, freqüentemente seguida por uma descrição mais detalhada, que pode incluir exemplos e explicações mais detalhadas. Estas práticas-chave, também referidas como as práticas-chave de alto nível, estabelecem as políticas, procedimentos e atividades fundamentais para a área-chave de processo. Os componentes da descrição detalhada são freqüentemente referidos como subpráticas. A figura 3.3 fornece um exemplo da estrutura básica de uma prática-chave para a área-chave de processo *Planejamento de Projeto*.

De acordo com a Figura 3.3, para garantir uma realização consistente da meta de documentar planos para o planejamento e acompanhamento do projeto, a organização deve estabelecer um procedimento documentado para produzir estimativas de tamanho de software. Se não forem desenvolvidas a partir de um procedimento documentado, essas estimativas podem variar sem controle, já que diferenças nas suposições de tamanho nunca são explicitadas. A descrição detalhada do que seria esperado num procedimento como esse inclui o uso de dados históricos, suposições documentadas e revisão de estimativas. Esses critérios guiam o julgamento se um procedimento razoável de estimativas de tamanho é seguido.



As práticas-chave descrevem “o que” deve ser feito, mas não deveriam ser interpretadas de forma obrigatória sobre “como” as metas deveriam ser alcançadas. Práticas alternativas podem completar as metas da área-chave de processo. As práticas-chave deveriam ser interpretadas racionalmente para julgar se as metas da área-chave de processo são efetivamente alcançadas. As práticas-chave estão

Definição Operacional do Modelo de Maturidade de Capabilidade

contidas em *“Key Practices of the Capability Maturity Model, Version 1.1”* [Paulk93b], juntamente com uma orientação sobre sua implementação.

4 Usando o CMM

O CMM estabelece um conjunto de critérios públicos e disponíveis que descrevem as características de organizações de software maduras. Esses critérios podem ser utilizados por organizações para melhorar seus processos de desenvolvimento e manutenção de software, ou por organizações governamentais ou comerciais para avaliar os riscos de uma contratação para um projeto de software junto a uma determinada companhia.

Este capítulo enfoca dois métodos desenvolvidos pelo SEI para avaliação da maturidade de uma organização com relação à execução do processo de software de uma organização: avaliação do processo e avaliação da capacidade do software.

- *A Avaliação do Processo de Software* é utilizada para determinar o estágio real de um processo de software de uma organização, para determinar os assuntos de mais alta prioridade relacionados ao processo e obter o apoio organizacional para a melhoria do processo de software.
- *A Avaliação da Capacidade do Software* é utilizada com a finalidade de identificar fornecedores qualificados para executar o trabalho relacionado a software, ou para monitorar o estado de um processo de software que está sendo utilizado em um empreendimento existente.

Esta visão geral não é suficiente, por si só, para que os leitores conduzam uma das duas avaliações. Qualquer pessoa que desejasse aplicar o CMM por um desses métodos deveria requisitar mais informações sobre treinamento em avaliações.

O CMM é um fundamento comum para as avaliações do processo de software e para as avaliações da capacidade do software. Os propósitos dos métodos são completamente diferentes, havendo diferenças significativas entre eles. Ambos são baseados no modelo e nos produtos dele derivados. O CMM, em conjunto com os produtos nele baseados, habilita os métodos a serem aplicados de modo confiável e consistente.

4.1 Métodos de Avaliação do Processo de Software e Avaliação da Capacidade do Software

A avaliação do processo de software está focada na identificação das prioridades de melhoria dentro do próprio processo de uma organização. As equipes de avaliação utilizam o CMM para orientá-las na identificação e na atribuição de prioridades às observações levantadas. Estas observações, em conjunto com as orientações fornecidas pelas práticas-chave do CMM, são utilizadas (por um grupo de processo de engenharia de software, por exemplo) para planejar uma estratégia de melhoria para a organização.

A avaliação da capacidade do software está focada na identificação de riscos associados a um projeto em particular ou a um contrato para desenvolvimento de um software de alta qualidade, dentro do prazo e do orçamento. Durante o processo de aquisição, avaliações da capacidade do software podem ser realizadas sobre os participantes da licitação. As observações levantadas numa avaliação, como estruturado pelo CMM, podem ser utilizadas para identificar os riscos de se selecionar um determinado fornecedor. As avaliações podem também ser realizadas em contratos já existentes para monitorar o desempenho de seus processos, com a intenção de identificar potenciais melhorias no processo de software do fornecedor.

O CMM estabelece uma estrutura comum de referência para a execução de avaliações do processo de software e avaliações da capacidade do software. Embora difiram em seus propósitos, os dois métodos utilizam o CMM como fundamento para avaliar a maturidade do processo de software. A figura 4.1 fornece uma descrição resumida dos passos comuns às duas avaliações.

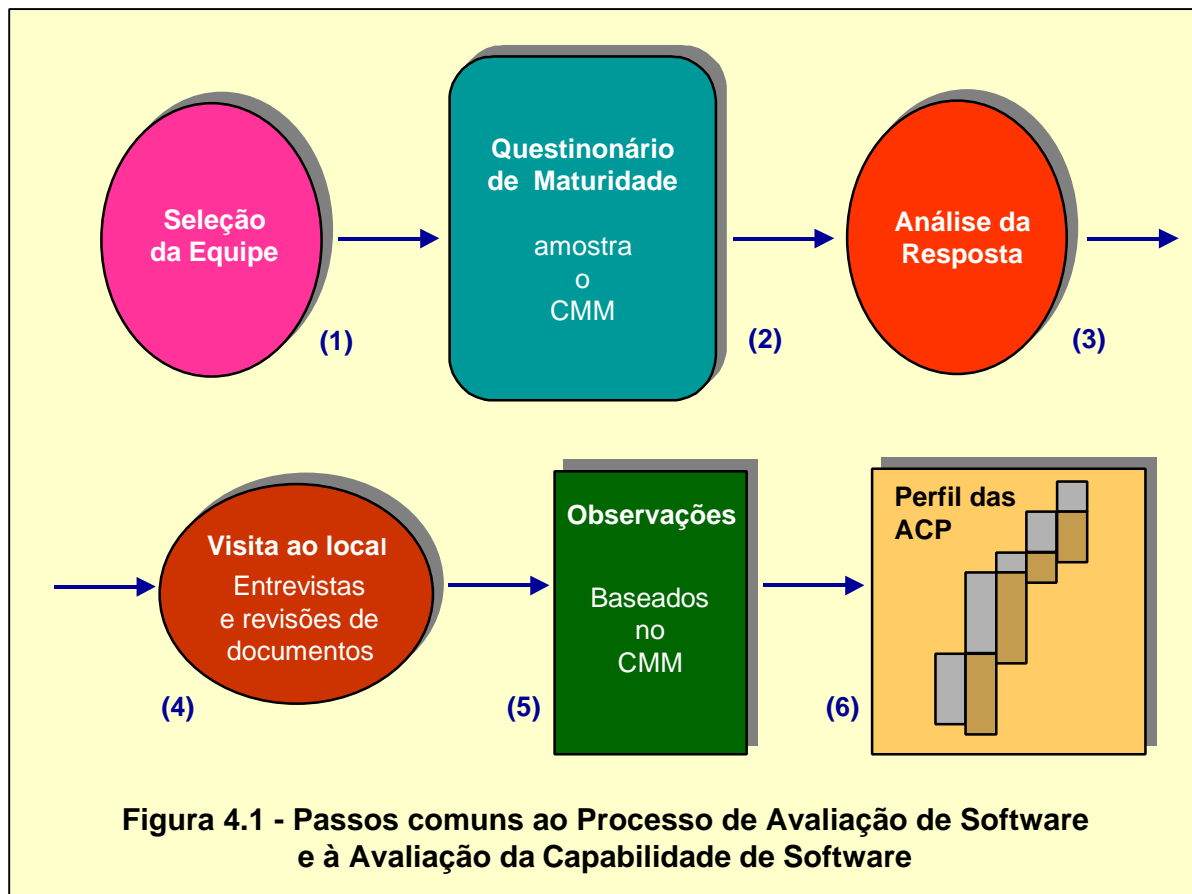
O primeiro passo é escolher a equipe de avaliação. Esta equipe deveria ser treinada nos conceitos básicos do CMM, tanto quanto no método de avaliação escolhido. Os membros da equipe deveriam ser profissionais com conhecimento em engenharia de software e gestão de software.

O segundo passo é obter dos representantes da organização a ser avaliada o questionário de maturidade respondido, ou outro instrumento de diagnóstico. Uma vez que esta atividade esteja completa, a equipe de avaliação executa uma análise das respostas (passo 3), as quais correspondem às respostas das questões e identifica as áreas onde um exame mais cuidadoso deve ser efetuado mais a diante. Essas áreas a serem investigadas correspondem às áreas-chave de processo do CMM.

A equipe está pronta agora para visitar a organização a ser avaliada (passo 4). Começando com os resultados da análise das respostas, a equipe realiza entrevistas e executa revisões na documentação para entender o processo de software que é

Usando o CMM

utilizado na organização avaliada. As áreas-chave de processo e as práticas-chave do CMM orientam os membros da equipe em como perguntar, entender, revisar e sintetizar as informações obtidas nas entrevistas e nos documentos. A equipe aplica um julgamento profissional para decidir se a implementação das áreas-chave de processo na organização satisfaz as metas das áreas-chave de processo relevantes⁷. Quando existe uma diferença clara entre as práticas-chave do CMM e as práticas locais, a equipe deve documentar suas razões para julgamento daquela Área-chave de Processo (ACP).



Ao fim do período interno à organização, a equipe elabora uma lista de observações (passo 5) que identifica os pontos fortes e as oportunidades de melhoria do processo de software da organização. Na avaliação do processo de software (*software process assessment*), as observações tornam-se a base das recomendações para melhoria do processo; já na avaliação da capacidade de software (*software capability evaluation*), as observações tornam-se parte da análise de risco efetuada pelo responsável pela aquisição.

⁷ Estes julgamentos podem ter que ocorrer sem completa informação quando questões de segurança ou de propriedade estiverem envolvidas.

Usando o CMM

Finalmente, a equipe prepara um perfil das áreas-chave de processo (passo 6) para mostrar onde a organização satisfaz, ou não, as metas das áreas-chave de processo. Uma área-chave de processo pode ser satisfeita e ainda assim ser alvo de observações que não identifiquem problemas graves que inibam a realização de quaisquer metas da área-chave de processo.

Em resumo, ambos os métodos de avaliação...

- utilizam o questionário de avaliação como um trampolim para o trabalho interno à organização,
- utilizam o CMM como um mapa que guia a investigação local,
- elaboram observações que identificam pontos fortes e oportunidades de melhoria do processo de software, com relação às áreas-chave de processo do CMM,
- elaboram um perfil baseado na análise de satisfação das metas das áreas-chave de processo, e
- apresentam seus resultados, para uma audiência apropriada, em termos de observações e de um perfil das áreas-chave de processo.

4.2 Diferenças entre Avaliação de Processo de Software e Avaliação da Capacidade de Software

A despeito dessas similaridades, os resultados dos dois métodos podem diferir, mesmo em aplicações sucessivas do mesmo método. Uma razão é que tanto o objetivo quanto a abrangência dos métodos podem variar. Primeiro, a organização que está sendo investigada deve ser determinada. Para uma grande empresa, várias definições diferentes de “organização” são possíveis. A abrangência pode ser baseada em uma gerência sênior comum, em uma localização geográfica comum, em uma designação como um centro de custo e de lucro, em um domínio de aplicação comum, ou outras considerações. Segundo, mesmo no que parece ser uma organização, a amostra de projetos selecionada pode afetar a abrangência. Uma divisão dentro de uma companhia pode ser avaliada, com a equipe chegando a observações baseadas na abrangência da divisão. Mais tarde, uma linha de produto naquela divisão pode ser avaliada, com a equipe elaborando observações sobre uma abrangência muito mais restrita. As comparações entre os resultados sem entender o contexto são problemáticas.

Os dois métodos diferem em motivação, objetivos, resultados produzidos e propriedade desses resultados. Esses fatores conduzem a diferenças substanciais na dinâmica das entrevistas, na abrangência dos questionamentos, na informação levantada e na formulação dos resultados. Os dois métodos são diferentes quando

Usando o CMM

são examinados os procedimentos detalhados empregados. O treinamento em um método não prepara a equipe para o outro método, ou seja, treinamentos em ambos os métodos se fazem necessários.

As avaliações de processo de software são executadas em um ambiente aberto e colaborativo. Seu sucesso depende de um compromisso entre a gerência e seus profissionais de melhorar a organização. O objetivo é levantar problemas e ajudar os gerentes e engenheiros de software a melhorar sua organização. Enquanto o questionário tem seu valor em focar a equipe de avaliação em questões dos níveis de maturidade, a ênfase dada está nas entrevistas, estruturadas ou não, como uma ferramenta para entender o processo de software da organização. Além da identificação de questões relacionadas com o processo de software da organização, a “compra da idéia da melhoria”, o foco da organização sobre processo e a motivação e entusiasmo em executar um plano de ação são os resultados mais importantes numa avaliação de processo de software.

As avaliações de capacidade de software, por outro lado, são executadas em um ambiente mais orientado para auditoria. O objetivo está ligado às considerações monetárias, já que as recomendações da equipe de avaliação ajudarão a selecionar um fornecedor ou estabelecer bônus. A ênfase está numa auditoria documentada que revele o processo de software realmente implementado pela organização.

Isto não significa, entretanto, que os resultados dos dois métodos de avaliação não deveriam ser comparáveis. Uma vez que os dois métodos são baseados no CMM, os pontos semelhantes e divergentes deveriam ser evidentes e explicáveis.

4.3 Outros usos do CMM em Melhoria de Processos

Para as equipes de processo de engenharia de software, ou para outros grupos que estão tentando melhorar seus processos de software, o CMM tem um valor particular nas áreas de planejamento de ações, implementação de planos de ação e definição de processos. Durante o planejamento de ações, os membros da equipe de processos, utilizando seu conhecimento sobre as questões relativas a seus processos de software e ambiente de negócios, podem comparar suas práticas correntes com as metas das áreas-chave de processos do CMM. As práticas-chave deveriam ser examinadas com relação às metas corporativas, às prioridades de gerenciamento, ao nível de desempenho da prática, ao valor de implementar cada prática para a organização e à habilidade da organização em implementar a prática à luz de sua própria cultura.

A equipe de processo deve então determinar as melhorias de processo que são necessárias, como executar a mudança, obtendo o necessário convencimento das pessoas. O CMM ajuda nessa atividade fornecendo um ponto de partida para discussões sobre melhoria de processos e também levantando suposições discrepantes so-

Usando o CMM

bre as práticas de engenharia de software comumente aceitas. Na implementação do plano de ação, o CMM e as práticas-chave podem ser utilizados pelas equipes de processo para desenvolver partes do plano de ação operacional e definir o processo de software.

5 Futuras Direções do CMM

O alcance dos níveis mais altos de maturidade do processo de software é incremental e requer um compromisso de longo prazo com a melhoria contínua de processos. As organizações de software podem levar dez anos ou mais para construir as bases para a melhoria contínua e cultura orientada nessa direção. Em bora um programa de melhoria de processo, com uma duração de uma década, seja estranho para a maioria das companhias norte-americanas, esse nível de esforço é necessário para produzir organizações de software maduras. Esse tempo é consistente com as experiências de outras indústrias, tal como a indústria automotiva norte-americana, que tem alcançado ganhos significativos no processo de maturidade [Gabor90].

5.1 O que o CMM não cobre

O CMM não é uma bala de prata [Brooks87] e não trata todos os assuntos que são importantes para o sucesso dos projetos. Por exemplo, o CMM não indica peritos em domínios específicos de aplicações, não defende tecnologias específicas de software, nem sugere como selecionar, contratar, motivar e reter pessoas competentes. Apesar desses assuntos serem cruciais para o sucesso do projeto, alguns deles têm sido analisados em outros contextos [Curtis90]. Entretanto, eles não têm sido integrados ao CMM. O CMM foi especificamente desenvolvido para prover uma estrutura ordenada e disciplinada dentro da qual possa se encaminhar assuntos de processo de gestão e desenvolvimento de software.

5.2 Atividades de Curto prazo

Os tutoriais e cursos de CMM têm sido apresentados em muitas conferências e seminários nos Estados Unidos para garantir que a indústria de software tenha uma adequada consciência do CMM e dos seus produtos associados. As ferramentas baseadas em CMM (por exemplo: o questionário de maturidade), o treinamento em avaliação de processo de software e o treinamento em avaliação da capacidade de software estão sendo desenvolvidos e/ou revisados para incorporar o CMM.

O foco de curto prazo nas atividades de desenvolvimento do CMM será orientado na direção de versões sob medida do CMM, tais como o CMM para projetos e/ou organizações pequenos. O CMM v1.1 é expresso em termos das práticas normativas de grandes organizações contratadas pelo governo, sendo que essas práticas devem ser ajustadas para as organizações que diferem desse modelo.

5.3 Atividades de longo prazo

Ao longo dos próximos anos, o CMM continuará sendo submetido a testes extensivos através da utilização das avaliações dos processos de software e das avaliações da capacidade de software. Os produtos e materiais de treinamento CMM serão desenvolvidos e revisados quando apropriado. O CMM é um documento vivo que será melhorado, mas está previsto que o CMM v1.1 permanecerá como base de referência pelo menos até 1996. Isso provê um equilíbrio apropriado e realista entre as necessidades de estabilidade e as necessidades de melhoria contínua.

Para a próxima versão do CMM, Versão 2, o SEI voltará sua atenção para a melhoria do modelo como um todo. Enquanto todos os níveis do modelo podem ser revistos, a ênfase recairá nos Níveis 4 e 5. No momento, as áreas-chave de processo dos Níveis 2 e 3 são as que estão mais completamente definidas. Uma vez que poucas organizações encontram-se nos Níveis 4 e 5 [Humphrey91a, Kitson92], pouco é conhecido sobre as características dessas organizações. As práticas para esses dois níveis serão redefinidas quando o SEI trabalhar mais intensamente com organizações que estão buscando entender e alcançar os Níveis 4 e 5. O CMM também pode ser tornar multi-dimensional para tratar questões de tecnologia e de recursos humanos.

O SEI também está trabalhando com a *International Organizations for Standardization* (ISO) em seus esforços para construir padrões internacionais para avaliação, melhoria e determinação da capacidade dos processos de software. O desenvolvimento dos padrões ISO influenciará o CMM v2, assim como o trabalho do SEI sobre processos irá influenciar as atividades da ISO.

5.4 Conclusão

A melhoria contínua se aplica ao modelo de maturidade e às práticas, assim como ao processo de software. O impacto potencial das mudanças do CMM na comunidade de software será cuidadosamente considerado, mas o CMM, o questionário de maturidade, a avaliação de processo de software e os métodos de determinação da capacidade de software continuarão a evoluir à medida que se ganha experiência no processo de software. O SEI pretende trabalhar muito próximo das indústrias, do governo e das academias, continuando essa evolução.

O CMM fornece uma estrutura conceitual para a melhoria da gestão e do desenvolvimento de produtos de software de uma forma consistente e disciplinada. Ele não garante que os produtos de software serão construídos com sucesso ou que todos os problemas de desenvolvimento de software serão resolvidos adequadamente. O CMM identifica práticas para um processo de software maduro e fornece exemplos do *estado da prática* (e, em alguns casos, do *estado da arte*), mas não tem a intenção de ser exaustivo nem ditatorial. O CMM identifica as características de um processo de software efetivo, mas a organização madura trata de todas as questões essenciais

Futuras Direções do CMM

para um projeto bem sucedido, incluindo pessoas e tecnologia, tanto quanto processos.

6 Referências

- Brooks87** F.P. Brooks, No Silver Bullet: Essence and Accidents of Software Engineering, IEEE Computer, vol.20 No.4, April 1987, pp10-19.
- Crosby79** P.B.Crosby, Quality is Free, McGraw-Hill, New York, NY, 1979.
- Curtis90** B.Curtis, Managing the Real Leverage in Software Productivity and Quality, American Programmer, Vol.3, No. July 1990, pp.4-14.
- Deming86** W. Edwards Deming, Out of the Crisis, MIT Center for Advanced Engineering Study, Cambridge, MA, 1986.
- DoD87** Report of the Defense Science Board Task Force on Military Software, Office of the Under Secretary of Defense for Acquisition, Washington, D.C. September 1987.
- Fagan86** M.E. Fagan, Advances in Software Inspections, IEEE Transaction on Software Engineering, Vol.12, No. 7, July 1986, pp744-751.
- Fowler90** P.Fowler and S.Rifkin, Software Engineering Process Group Guide, Software Engineering Institute, CMU/SEI90-TR-24, ADA235784, September, 1990.
- Freedman90** D.P Freedman and G.M. Weinberg, Handbook of Walktroughs, Inspections, and Technical Reviews, Third Edition, Dorset House, New York, NY, 1990.
- Gabor90** A. Gabor, The Man Who Discovered Quality, Random House, New York, NY, 1990.
- GAO-92-48** Embedded Computer Systems: Significant Software Problems on C-17 Must be Addresses, GAO/IMTEC-92-48 May 1992.
- Humphreys87a** W.S. Humphrey, Characterizing the Software Process: A Maturity Framework, Software Engineering Institute, CMU/SEI-87-TR-11, ADA182895, June 1987.
- Humphreys87b** W.S. Humphrey and W.L. Sweet, A method for Assessing the Software Engineering Capability of Contractors, Software Engineering Institute, CMU/SEI-87-TR-23, ADA187320. September 1987.

Referências

- Humphrey88** W.S. Humphrey, Characterizing the Software Process, IEEE Software, Vol.5, No.2, March, 1988, pp73-79.
- Humphrey89** W.S. Humphrey, Managing in the Software Process, Addison-Wesley, Reading, MA, 1989.
- Humphrey91a** W.S. Humphrey, D.H. Kitson, and J. Gale, A Comparison of U.S. and Japanese Software Process Maturity, Proceedings of the 13th International Conference on Software Engineering, Austin, TX, 13-17 May 1991, pp.38-49.
- Humphrey91b** W.S. Humphrey, Process Fitness and Fidelity, Proceedings of the Seventh International Software Process Workshop, 16-18 October 1991.
- IEEE-STD-610** ANSI/IEEE Std610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, February, 1986.
- Imai86** M.Imai, Kaizen: The Key to Japan's Competitive Success, McGraw-Hill, New York, NY, 1986.
- Juran88** J.M. Juran , Juran on Planning for Quality, Macmillan New York, NY, 1988.
- Juran89** J.M. Juran , Juran on Leadership for Quality, the Free Press, New York, NY, 1989.
- Kitson92** D.H. Kitson and S. Masters, An Analysis of SEI Software Process Assessment results: 1987-1991, Software Engineering Institute, CMU/SEI-92-tr-24, July 1992.
- Paulk91** M.C. Paulk, B. Curtis, M.B. Chrissis, et al, Capability Maturity Model for Software, Software Engineering Institute, CMU/SEI-91-TR-24, ADA240603, August 1991.
- Paulk93a** M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, Capability Maturity Model for Software, version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24 February 1993.
- Paulk93b** M.C. Paulk, C.V. Weber, S. Garcia, M.B. Chrissis and M. Bush, Key Practices of the Capability Model, Version 1.1, Software Engineering Institute, CMU/SEI-93TR-25, February 1993.
- Radice85** R. A. Radice, J.T. Harding, P.E. Munnis, and R.W. Phillips, A Programming Process Study, IBM System Journal, Vol.24, No.2, 1985.

Referências

- Siegel90** J. A. L. Siegel, et al., National Software Capacity: Near –Term Study, Software Engineering Institute, CMU/SEI-90-TR-12-ADA226694, May 1990.
- Weber91** C.V. Weber, M.C. Paulk, C.J. Wise and J.V. Withey, Key Practices of the Capabilities Maturity Model, Software Engineering Institute, CMU/SEI-91-TR-25, ADA240604, AUGUST 1991.

7 Apêndice A: Metas para cada Área-chave de Processo

As metas para cada área-chave de processo são listadas abaixo segundo o seu nível de maturidade:

A.1 As Áreas-chave de Processo do Nível 2: Repetível

Gestão de Requisitos

- Meta 1** Os requisitos de sistema alocados ao software são controlados para estabelecer uma *baseline* para a engenharia de software e para o uso pela da gerência.
- Meta 2** Os planos, produtos e atividade de software são mantidos consistentes com os requisitos de sistemas alocados ao software.

Planejamento de Projeto de Software

- Meta 1** As estimativas de software são documentadas para o uso no planejamento e no acompanhamento do projeto de software.
- Meta 2** As atividades e os compromissos do projeto de software são planejados e documentados.
- Meta 3** As equipes e os indivíduos afetados concordam com seus compromissos relacionados com o projeto de software.

Supervisão e Acompanhamento de Projeto

- Meta 1** Os resultados e o desempenho reais são comparados com os planos de software.
- Meta 2** Ações corretivas são realizadas e gerenciadas até a conclusão quando os resultados e o desempenho reais desviam significativamente dos planos de software.
- Meta 3** As alterações nos compromissos de software são realizadas em consenso pelas equipes e indivíduos afetados.

Metas para cada área de processo

Gestão de Subcontratação

- Meta 1** O contratante seleciona subcontratados qualificados em software.
- Meta 2** O contratante e o subcontratado de software concordam com seus compromissos um para com o outro.
- Meta 3** O contratante e o subcontratado de software mantêm comunicações progressivas.
- Meta 4** O contratante acompanha os resultados e o desempenho reais do subcontratado, confrontando-os com os seus compromissos.

Garantia da Qualidade de Software

- Meta 1** As atividades da garantia da qualidade de software são planejadas.
- Meta 2** A aderência dos produtos e das atividades de software aos padrões, aos procedimentos e aos requisitos aplicáveis é verificada.
- Meta 3** As equipes e indivíduos afetados são informados sobre as atividades e resultados da garantia da qualidade de software.
- Meta 4** As questões de não conformidade que não podem ser resolvidas no projeto de software são encaminhadas para um gerente superior.

Gestão de Configuração

- Meta 1** As atividades de Gestão de Configuração são planejadas.
- Meta 2** Os produtos de trabalho de software selecionados são identificados, controlados e disponibilizados.
- Meta 3** As alterações nos produtos de trabalho de software identificados são controladas.
- Meta 4** As equipes e indivíduos afetados são informados sobre a situação e o conteúdos das *baselines* de software.

Metas para cada área de processo

A.2 As Áreas-chave de Processo do Nível 3: Definido

Foco no Processo da Organização

- Meta 1** As atividades de desenvolvimento e melhoria de processo de software são coordenadas em toda a organização.
- Meta 2** Os pontos fortes e as oportunidades de melhoria dos processos de software utilizados são identificados com relação a um processo padrão.
- Meta 3** As atividades de desenvolvimento e melhoria de processo no nível de organização são planejadas.

Definição do Processo da Organização

- Meta 1** Um processo de software padrão para a organização é elaborado e mantido.
- Meta 2** As informações, relativas ao uso do processo padrão de software da organização pelos projetos de software, são coletadas, revisadas e disponibilizadas.

Programa de Treinamento

- Meta 1** As atividades de treinamento são planejadas.
- Meta 2** O treinamento para desenvolver as habilidades e conhecimentos necessários à execução de gerência de software e das funções técnicas é fornecido.
- Meta 3** Os indivíduos da equipe de desenvolvimento de software e das equipes relacionadas ao software recebem o treinamento necessário para desempenhar suas funções.

Gestão Integrada de Software

- Meta 1** O processo de software definido para o projeto é uma versão personalizada do processo padrão de software da organização.
- Meta 2** O projeto é planejado e gerenciado de acordo com o processo de software definido para o projeto.

Metas para cada área de processo

Engenharia de Produto de Software

- Meta 1** As tarefas de desenvolvimento de software são definidas, integradas e realizadas consistentemente para produzir o software.
- Meta 2** Os produtos de trabalho de software são mantidos consistentes uns com os outros.

Coordenação Intergrupos

- Meta 1** Os requisitos do cliente são consensados entre todas as equipes envolvidas.
- Meta 2** Os compromissos entre as equipes de engenharia são consensados entre as equipes afetadas.
- Meta 3** As equipes de engenharia identificam, acompanham e resolvem as questões intergrupos.

Revisões por pares

- Meta 1** As atividades de revisões por pares são planejadas.
- Meta 2** Os defeitos nos produtos de trabalho de software são identificados e removidos.

A.3 As Áreas-chave de Processo do Nível 4: Gerenciado

Gestão Quantitativa de Processo

- Meta 1** As atividades de gerência quantitativa de processo são planejadas.
- Meta 2** O desempenho do processo de software definido para o projeto é controlado quantitativamente.
- Meta 3** A capacidade do processo padrão de software da organização é conhecido em termos quantitativos.

Metas para cada área de processo

Gestão da Qualidade de Software

- Meta 1** As atividades de gestão da qualidade de software do projeto são planejadas.
- Meta 2** Metas mensuráveis para a qualidade de produto de software e suas prioridades são definidas.
- Meta 3** O progresso real em direção ao alcance das metas da qualidade para os produtos de software é quantificado e gerenciado.

A.4 As Áreas-chave de Processo do Nível 4: Em otimização

Prevenção de Defeitos

- Meta 1** As atividades de prevenção de defeito são planejadas.
- Meta 2** As causas comuns de defeito são pesquisadas e identificadas.
- Meta 3** As causas comuns de defeito são priorizadas e eliminadas sistematicamente.

Gestão de Alteração de Tecnologia

- Meta 1** A introdução de mudanças de tecnologia é planejada.
- Meta 2** As novas tecnologias são avaliadas para determinar seus efeitos sobre a qualidade e sobre a produtividade.
- Meta 3** As novas tecnologias apropriadas são transferidas para a prática normal em toda a organização.

Gestão de Mudança de Processo

- Meta 1** O processo de melhoria contínua é planejado.
- Meta 2** A participação nas atividades de melhoria de processo de software ocorre em toda a organização.
- Meta 3** O processo padrão de software da organização e o processo de software definido para a organização são continuamente melhorados.