

Darcio Joel Ferreira dos Santos

**Uma Taxonomia para Métodos de Melhoria de
Processos de Software**

Monografia apresentada à
Escola Politécnica da
Universidade de São Paulo
para a conclusão do curso de
MBA em Engenharia de
Software

**São Paulo
2005**

1825991

Darcio Joel Ferreira dos Santos

**Uma Taxonomia para Métodos de Melhoria de
Processos de Software**

Monografia apresentada à
Escola Politécnica da
Universidade de São Paulo
para a conclusão do curso de
MBA em Engenharia de
Software

Área de concentração:
Qualidade de Software

Orientador:
Prof. Dr. Kechi Hirama

**São Paulo
2005**

A minha esposa Sheila que
pacientemente me apoiou durante esta
longa jornada e sabiamente me incentivou
e aconselhou.

AGRADECIMENTOS

A Prof.^a Selma Shin Shimizu Melnikoff por sua compreensão, sensibilidade e por seu belíssimo trabalho na coordenação do curso.

Ao Prof.^o Dr. Kechi Hiramã por ter aceito o desafio e me orientado de forma segura e sabia no desenvolvimento desta monografia.

Aos amigos e parentes que compreenderam a minha ausência em momentos importantes de nossas vidas.

RESUMO

Este trabalho propõe uma taxonomia para métodos de processo de software. Uma série de características, tidas como relevantes, foi relacionada e cada método descrito em termos destas características.

O levantamento dos atributos de cada método foi feito através do estudo aprofundado das referências bibliográficas apresentadas no final deste trabalho.

Para contextualizar a taxonomia, o trabalho apresenta duas das mais importantes referências para projetos de melhoria de processos de software: o modelo de capacidade e maturidade CMMI e a norma ISO/IEC TR 15504.

Também são apresentados histórico, descrição, estratégia e processos dos seguintes métodos de melhoria: PDCA, QIP e IDEALSM.

Ao final, apresenta-se a comparação dos métodos de melhoria de processos de forma tabular, facilitando a leitura e permitindo consultas rápidas.

ABSTRACT

This paper proposes a taxonomy for software process methods. A set of important characteristics have been listed and based upon these characteristics some methods were described.

The elicitation of each method's attribute is based on a deep research of the bibliographic reference that is listed on the end of this work.

To set the context, two of the most important references for software process improvement projects are presented: the maturity and capability model CMMI and the ISO/IEC TR 15504 standard.

The history, description and the approach of the PDCA, QIP, IDEALSM improvement methods are also presented.

Finally, to facilitate the reading of the comparison of the improvement methods they are shown in a tabular format.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	ii
LISTA DE ABREVIATURAS E SIGLAS.....	iii
1. INTRODUÇÃO.....	1
1.1 Objetivo.....	2
1.2 Motivação.....	2
1.3 Justificativa	3
1.4 Estrutura do trabalho	4
2. PROCESSOS DE SOFTWARE	6
2.1 Modelo de Capacidade e Maturidade – CMMI.....	6
2.1.1 Introdução [9].....	6
2.1.2 Representações: Contínua e por Estágios [16].....	7
2.1.3 Estrutura da representação por Estágios [9].....	10
2.1.4 Estrutura representação Contínua [17].....	10
2.1.5 Componentes Obrigatórios, Esperados e Informativos[9]	11
2.1.6 Componentes do Modelo[9].....	12
2.2 Norma ISO/IEC TR 15504	13
2.2.1 Introdução[18].....	13
2.2.2 O modelo de referência bidimensional – ISO/IEC TR 15504-2 [19]	16
A) A dimensão do processo [19].....	16

B)	A dimensão de capacidade do processo [19]	17
2.3	Necessidade de estratégia de implantação de processos	19
3.	MÉTODOS DE MELHORIA DE PROCESSOS.....	20
3.1	PDCA	20
3.1.1	Histórico	20
3.1.2	Descrição.....	20
3.1.3	Estratégia e Processo.....	21
3.2	QIP	24
3.2.1	Histórico.....	24
3.2.2	Descrição.....	24
3.2.3	Estratégia e Processo.....	25
A)	O Ciclo Organizacional.....	26
B)	O Ciclo do Projeto.....	28
3.3	IDEAL.....	29
3.3.1	Histórico.....	29
3.3.2	Descrição.....	29
3.3.3	Estratégia e Processo.....	30
A)	A etapa Inicial	32
B)	A etapa de diagnóstico	34
C)	A etapa de estabelecimento.....	35
D)	A etapa de Ação	36

E) A etapa de Aprendizado	37
4. TAXONOMIA DOS MÉTODOS DE MELHORIA DE PROCESSOS DE SOFTWARE.....	38
4.1 Categoria Geral	39
4.2 Categoria Processo	39
4.3 Categoria Organizacional.....	40
4.4 Categoria Resultados.....	41
5. CONCLUSÃO	43
5.1 Contribuições do Trabalho	43
5.1.1 Trabalhos Futuros	43
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	44
7. BIBLIOGRAFIA COMPLEMENTAR	47

LISTA DE FIGURAS

Figura 1: Componentes Modelo CMMI representação por Estágios [9]	10
Figura 2: Componentes Modelo CMMI representação Contínua [17]	11
Figura 3: Ciclo PDCA [21]	21
Figura 4: Rampa de melhorias PDCA [13]	22
Figura 5: Quadro de ferramentas de Gestão da Qualidade e ciclo PDCA [13]	23
Figura 6: Ciclo QIP [20]	25
Figura 7: O modelo IDEAL [15]	30
Figura 8: Visão bidimensional de uma atividade de melhoria de processo [15]	31

LISTA DE TABELAS

Tabela 1: Modelos CMMI e sua aplicação [16].....	7
Tabela 2: Diferenças entre as representações Contínua e Estágio [16]	8
Tabela 3: Níveis de Maturidade e Capacidade do CMMI e CMM [16]	9
Tabela 4: Matriz relacionando os níveis de maturidade às PAC [16].....	9
Tabela 5: Grupos e categorias de processos.....	17
Tabela 6: Atividades do modelo IDEAL SM [15].....	31
Tabela 7: Categorias adaptado de [27].....	38
Tabela 8: Comparação dos métodos de melhoria PDCA, QIP e IDEAL SM adaptado de [27]	42

LISTA DE ABREVIATURAS E SIGLAS

CL	Nível de Capacidade do inglês Capability Level
CMMI	Capability Maturity Model® Integration
FDD	Flight Dynamic Division at Goddard Space Flight Center
GQM	Goal Question Metric
IDEAL	Initiating, Diagnosing, Establishing, Acting & Learning
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ML	Nível de Maturidade do inglês Maturity Level
MPS	Melhoria de Processo de Software
PA	Área de Processo do inglês Process Area
PDCA	Plan, Do, Check, Act
SEI	Software Engineering Institute
SEL	Software Engineering Laboratory
SEPG	Software Engineering Process Group
SPICE	Software Process Improvement and Capability Determination
UM	University of Maryland

1. INTRODUÇÃO

A revolução da informação, cuja trajetória é marcada pela superação de dificuldades abordadas a seguir, avança a passos largos e a importância do software e seu impacto na sociedade continuam a crescer. Segundo Nielsen, o software afeta a todos nós e traz o século XXI à vida. Por isto, necessitamos de software que seja construído com qualidade, dentro do orçamento e prazo definidos e, que seja utilizável, escalável e manutenível. [1]

A Tecnologia da Informação tornou-se uma ferramenta fundamental para os setores privados e públicos e o software um diferencial competitivo, seja possibilitando a oferta de novos serviços ou, simplesmente, tornando as organizações mais eficientes, ágeis e competitivas.

A revolução digital teve início nos anos 50, se solidificou na década de 60 e somente no final da década de 70 as dificuldades inerentes à produção de software começaram a tomar vulto, levando a comunidade de software a acreditar que a crise do software estava prestes a se concretizar. [1] e [2]

Todavia, após décadas de existência e apesar do acúmulo de inúmeros insucessos, a indústria de software não enfrentou a tão alardeada crise. Para Pressman, os sucessos conquistados pela engenharia de software são superiores aos fracassos e a crise prevista por mais de 30 anos, nunca se materializou. [3]

Essa constatação e o fato de que antigos problemas, como baixa produtividade, baixa qualidade do produto e custo além do estimado, ainda afligem a indústria de software, levaram seus observadores e estudiosos a substituir o termo crise de software pelo termo “aflição crônica”. A expressão “aflição crônica” é usada para definir uma situação de insegurança, onde algo provoca um desconforto contínuo ou recorrente, sem previsão de solução e com possibilidades de persistir por muito tempo. [3]

A busca de soluções para essa aflição estimulou a aplicação de conceitos de outras áreas do conhecimento à engenharia de software. Na década de oitenta os esforços voltaram-se para a modelagem de processos capazes de garantir a qualidade do

produto. Segundo Chrissis, a premissa da gestão de processos de software de que a qualidade do sistema de software é altamente influenciada pela qualidade do processo utilizado para seu desenvolvimento e sua manutenção, e a convicção dos movimentos de qualidade em sua eficácia, levou a indústria de software a adotar amplamente modelos de melhoria contínua de processos. [4]

Esses modelos aplicam conceitos e princípios de melhoria contínua estabelecidos por Walter Shewhart [5] e aprimorados por W. Edwards Deming [6] e Joseph Juran [7] ao processo desenvolvimento de software, a partir da idéia de que processos sob controle produzem resultados semelhantes na repetição. Desse modo, melhores processos implicam em melhores resultados. Segundo Halvorsen, pode-se claramente representar essa dependência através da seguinte relação causal:

$$\text{Qualidade (Processo)} \Rightarrow \text{Qualidade (Produto)} [8]$$

Para apoiar a necessidade intrínseca desses modelos de melhorar continuamente o processo, devemos utilizar algum método de melhoria de processos que nos auxilie a introduzir alterações de forma planejada, coerente, verificável, segura e com uma margem de erro aceitável. Desta forma, minimizamos os riscos, os impactos e reações adversas permitindo o estabelecimento e permanência da melhoria contínua.

1.1 Objetivo

O objetivo deste trabalho é apresentar uma taxonomia dos métodos de melhoria de processos comumente utilizados na implantação de modelos de maturidade e capacidade.

1.2 Motivação

A comunidade do software há muito tempo reconhece a importância do estabelecimento de processos eficientes e eficazes que viabilizem a produção de software de qualidade, de forma previsível e lucrativa.

Modelos e normas como CMMI [9], ISO 9001 [10], ISO/IEC 15504 [11] entre outras facilitam o trabalho das organizações na concretização deste desejo ao estabelecerem prioridades e ciclos de melhoria incrementais, desenvolvendo nas organizações, deste modo, uma cultura de aperfeiçoamento contínuo.

Os métodos de melhoria de processo são ferramentas que alavancam as transformações pelas quais as organizações deverão passar em sua jornada até os níveis de maturidade mais elevados.

No entanto, devido à diversidade das propostas disponíveis de métodos, detecta-se uma dificuldade na escolha do método apropriado às necessidades das organizações, aumentando os riscos na definição do foco e estratégia do processo de melhoria.

Além disto, é muito comum às organizações iniciantes em projetos de melhoria de processos a ausência de maturidade suficiente para a condução deste tipo de desafio. É neste ponto, que os métodos de melhoria de processos podem, e muito, auxiliá-las pois prevêem estruturas para gerenciar as atividades de melhoria e manter o apoio da alta direção das organizações.

1.3 Justificativa

Segundo Hefner, os erros mais comuns cometidos pelas organizações no planejamento e na execução de programas de melhoria de processos estão relacionados à definição de objetivos implausíveis e alheios às metas do negócio estabelecidas pela alta gerência. [12]

Os métodos de melhoria de processos podem evitar estes erros já que preconizam a definição de objetivos e de mecanismos para verificar se os mesmos foram alcançados.

Este trabalho apresenta os métodos de melhoria de processo que auxiliam as organizações a conduzirem seus esforços de melhoria de forma planejada, ordenada e organizada, permitindo que suas metas sejam alcançadas.

A taxonomia de métodos de melhoria de processos de software, proporcionará subsídios para escolher, dentre os métodos abordados, o mais adequado às especificidades do seu projeto de melhoria de processos.

A caracterização dos métodos de melhoria de processos de software, para a realização da taxonomia, fornecerá uma visão geral dos métodos de melhoria e poderá ser utilizada para a realização de outras comparações. Entretanto, a descrição das

características será de alto nível e maiores detalhes deverão ser obtidos em outros trabalhos.

Como serão abordados apenas três métodos há a possibilidade da extensão deste trabalho através da inclusão de mais métodos na estrutura utilizada para a elaboração da comparação. Este trabalho também pode ser estendido através da utilização de outras técnicas de comparação tais como: mapeamento dos principais conceitos dos métodos, mapeamento de necessidades dos usuários com as propriedades dos métodos, etc.

Este trabalho será desenvolvido com as seguintes itens principais:

- Levantar os benefícios da utilização de métodos de melhoria em projetos de melhoria de software
- Apresentar uma taxonomia de métodos de melhoria de processos de software
- Auxiliar as pessoas envolvidas no planejamento e execução de projetos de melhoria de software a escolherem um método de melhoria de processos adequado as suas necessidades.

1.4 Estrutura do trabalho

Este trabalho está dividido em 7 capítulos.

O primeiro capítulo engloba a introdução, o objetivo, motivação, a justificativa e a própria estrutura do trabalho.

O segundo capítulo apresenta os modelos de processo de software CMMI e ISO/IEC 15504 respectivamente [9] e [11]. Esses modelos são baseados nos conceitos desenvolvidos por Crosby, Deming, Juran, e Humphrey [4].

O terceiro capítulo apresenta os métodos de melhoria de processos existentes e sua importância. Este trabalho será focado nos seguintes métodos de melhoria de processos: PDCA, QIP e IDEAL.[13], [14] e [15]

O quarto capítulo apresenta a análise comparativa dos métodos de melhoria de processos apresentados no terceiro capítulo.

O quinto capítulo traz a conclusão deste estudo, bem como sugere possíveis trabalhos a serem realizados.

As referências bibliográficas utilizadas no presente trabalho encontram-se apontadas no sexto capítulo.

O sétimo e último capítulo apresenta a bibliografia complementar que apesar de não ser diretamente referenciada auxiliou na formação do conhecimento necessário para realizar este trabalho.

2. PROCESSOS DE SOFTWARE

2.1 Modelo de Capacidade e Maturidade – CMMI

2.1.1 Introdução [9]

Com a publicação do modelo de capacidade e maturidade integrado (CMMI) o Software Engineering Institute (SEI), da Universidade de Carnegie Mellon (CMU), iniciou uma nova etapa da jornada iniciada nos anos 90. Após a publicação do CMM v1.1 (Software CMM) em 1993, os esforços do SEI para o lançamento da versão 2.0 foram canalizados para a criação de um modelo CMM abrangente e que integrasse diversas disciplinas.

Desde 1991, inúmeros modelos CMMs foram desenvolvidos para uma grande variedade de disciplinas. Alguns dos mais notáveis incluem modelos para Engenharia de Sistemas, Engenharia de Software, Aquisição de Software, Gerência e Desenvolvimento de Pessoas, e Produto Integrado ao Processo de Desenvolvimento.

Apesar destes modelos terem provado sua importância para muitas organizações, a utilização de mais de um modelo em conjunto, uma necessidade de muitas organizações, mostrou-se problemática. Significativas diferenças de arquitetura, conteúdo e abordagem dos modelos inviabilizaram sua adoção, aumentando o custo de treinamentos, avaliações e da condução do processo de melhoria. A manutenção dos modelos pelo SEI também se tornou difícil pois os modelos apresentavam inúmeras redundâncias, inconsistências e vocabulários distintos.

O modelo CMMI foi idealizado para solucionar os problemas da adoção de múltiplos CMMs e é a combinação em uma única estrutura básica de três modelos:

1. Capability Maturity Model for Software (SW-CMM) v2.0 draft C,
2. Electronic Industries Alliance Interim Standard (EIA/IS) 731
3. Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98

O modelo resultante é constituído de uma estrutura central que engloba diversas disciplinas mantendo um grau de flexibilidade suficiente para suportar duas representações distintas (contínua e por estágios), a expansão através da integração futura com modelos CMMI de disciplinas específicas e a compatibilidade com a norma ISO/IEC 15504 Technical Report for Software Process Assessment.

O modelo central é o CMMI-SE/SW (System Engineering and Software Engineering). Os modelos estendidos são os CMMI-SE/SW/IPPD (Integrated Product and Process Development), CMMI-SE/SW/IPPD/SS (Supplier Sourcing ou formalmente aquisição). A tabela 1 apresenta a aplicação de cada modelo:

Tabela 1: Modelos CMMI e sua aplicação [16]

Modelo CMMI	Tipo de Organização Alvo
SW	Engenharia de Software
SE/SW	Engenharia de Sistemas/Software
SE/SW/IPPD	Engenharia de Sistemas/Software com foco no envolvimento precoce e contínuo dos clientes
SE/SW/IPPD/SS	Engenharia de Sistemas/Software com foco no envolvimento precoce e contínuo dos clientes e fornecedores

2.1.2 Representações: Contínua e por Estágios [16]

O desenvolvimento do CMMI levou em conta desde do seu início dois objetivos:

1. Os níveis de maturidade do modelo por estágios não permitiam ao CMM ser flexível o bastante para atender as organizações que precisavam ajustar seus processos de melhoria aos objetivos de negócio e não o contrário.
2. A transição das organizações que já utilizavam o CMM v1.1 para o CMM deveria ser o mais fácil possível de forma a preservar ao máximo seus investimentos.

A resposta a estes desafios estava na possibilidade de representar o mesmo modelo de duas formas: estágio e contínua.

A representação contínua é mais flexível pois permite a adoção de uma estratégia de melhoria alinhada aos objetivos de toda a organização. Também devemos salienta

que a representação contínua é mais compatível com os tipos de avaliações definidas na ISO 15504.

Em contraste, o modelo por estágios é o mais adequado às organizações que desejam migrar do CMM v.1.1 para o CMMI com o mínimo de retrabalho.

A tabela 2 apresenta as principais diferenças entre as duas representações:

Tabela 2: Diferenças entre as representações Contínua e Estágio [16]

Contínua	Estágio
Melhorar o desempenho em um processo único	Enfoque de melhoria do processo de forma sistêmica e estruturada
Melhorar desempenho em várias áreas alinhadas aos objetivos de negócio da organização	Atingir cada um dos estágios, garante a base fundamentada necessária para o próximo estágio
Níveis de capacidade utilizados para medir as melhorias	Áreas de Processo (PA) organizadas em níveis de maturidade
Melhorar diferentes processos com diferentes classificações	Permite a organização ter um caminho evolutivo pré-definido para melhoria
Necessário conhecimento das dependências e interações entre áreas de processo (PA)	Provê a migração mais fácil do SW-CMM para o CMMI
Necessário conhecimento das dependências e interações entre áreas de processo	Recomendado para quem está atuando com SW-CMM
Apropriado para quem sabe que processo deve ser melhorado	Apropriado para quem não sabe como iniciar um processo de melhoria ou qual processo deve ser prioridade
Alinhado com a ISO/IEC 15504 devido a organização idêntica das PA's	Possui grande número de casos de estudo e dados históricos de práticas bem sucedidas

As duas representações possuem praticamente o mesmo conteúdo. Ambas possuem as mesmas 22 Áreas de Processo (PA), os objetivos genéricos e específicos são muito parecidos, e é possível estabelecer uma relação unidirecional dos resultados de avaliações de estágios equivalentes da representação contínua para a por estágios.

Uma grande diferença reside no nível mais alto das representações. Na representação por estágio ainda temos, da mesma forma como foram definidos no CMM v1.1, os Níveis de Maturidade (Maturity Levels). As Áreas de Processo são distribuídas nos quatros níveis superiores dos cinco níveis de Maturidade: Gerenciado, Definido, Gerenciado Quantitativamente e Otimização.

Entretanto, na representação contínua os Níveis de Maturidade foram substituídos por Níveis de Capacidade (Capability Levels) como uma medida atribuída individualmente a cada Área de Processo. Outra grande diferença é o agrupamento

das Áreas de Processo em quatro categorias: Gerência de Processo, Gerência de Projeto, Engenharia e Suporte.

A tabela 3 relaciona os níveis de maturidade ou capacidade dos modelos CMMI e CMM:

Tabela 3: Níveis de Maturidade e Capacidade do CMMI e CMM [16]

Representação Nível	CMMI Estágio Maturidade	CMMI Contínuo Capacidade	CMM v.1.1 (1993) Maturidade
5	Otimização	Otimização	Otimização
4	Gerenciado Quantitativamente	Gerenciado Quantitativamente	Previsível
3	Definido	Definido	Estabelecido
2	Gerenciado	Gerenciado	Repetível
1	Inicial	Executado	Inicial
0	-	Incompleto	-

A tabela 4 relaciona os níveis de Maturidade da representação por estágio às Categorias de Áreas de Processos da representação contínua:

Tabela 4: Matriz relacionando os níveis de maturidade às PAC [16]

ESTÁGIO

CONTÍNUA				
PA categorias Maturidade	Gerência de Processo	Gerência de Projeto	Engenharia	Suporte
5	OID CL 0-5			CAR CL 0-5
4	OPP CL 0-5	QPMCL CL 0-5		
3	OPD CL 0-5 OT CL 0-5 OPF CL 0-5	RSKM CL 0-5 IPM CL 0-5	PI CL 0-5 RD CL 0-5 TS CL 0-5 VER CL 0-5 VAL CL 0-5	DAR CL 0-5
2		PP CL 0-5 PMC CL 0-5 SAM CL 0-5	REQM CL 0-5	PPQA CL 0-5 CM CL 0-5 MA CL 0-5
1				

Áreas de Processo:

CAR - Análise Causal e Resolução

CM Gerência de Configuração

DAR - Análise de Decisão e Resolução

IPM - Gerência Integrada de Projeto

MA - Medição e Análise

OID - Inovação e Melhoria Organizacional

OPD - Definição de Processo da Organização

OPF - Foco no Processo da Organização

OPP - Desempenho do Processo Organizacional

OT - Treinamento Organizacional

PI - Integração de Produto

PMC - Acompanhamento e controle de Projeto

PP - Planejamento de Projeto

PPQA - Garantia da Qualidade de Processo e Produto

QPM - Gerência Quantitativa de Projeto

RD - Desenvolvimento de Requisitos

REQM - Gerência de Requisitos

RSKM - Gerência de Risco

SAM - Gerência de Acordos com Fornecedores

TS - Solução Técnica

VAL - Validação

VER - Verificação

2.1.3 Estrutura da representação por Estágios [9]

A figura 1 apresenta a representação por estágios do modelo CMMI:

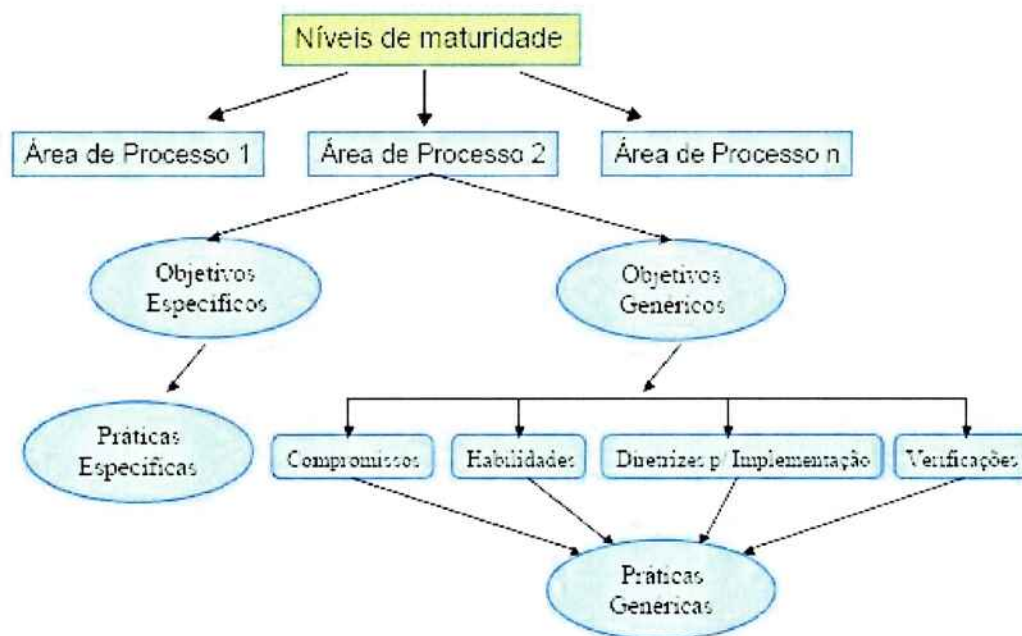


Figura 1: Componentes Modelo CMMI representação por Estágios [9]

A representação por estágios organiza as Áreas de Processo em cinco Níveis de Maturidade para auxiliar e guiar a melhoria de processo. As áreas de processo são agrupadas em níveis de maturidade, indicando as PAs que devem ser implementadas para se alcançar um determinado nível de maturidade. Os Níveis de Maturidade representam um caminho para a melhoria de processos que ilustra para toda a organização a evolução de sua capacidade.

Na representação por estágios, os Níveis de Maturidade sugerem uma ordem para implementar os estágios de melhoria de processo. Como ilustrado na figura 1, os níveis de maturidade organizam as PAs. Dentro de cada Área de Processo existem objetivos genéricos e específicos assim como práticas genéricas e específicas e Aspectos Comuns.

2.1.4 Estrutura representação Contínua [17]

A figura 2 apresenta a representação Contínua do modelo CMMI:

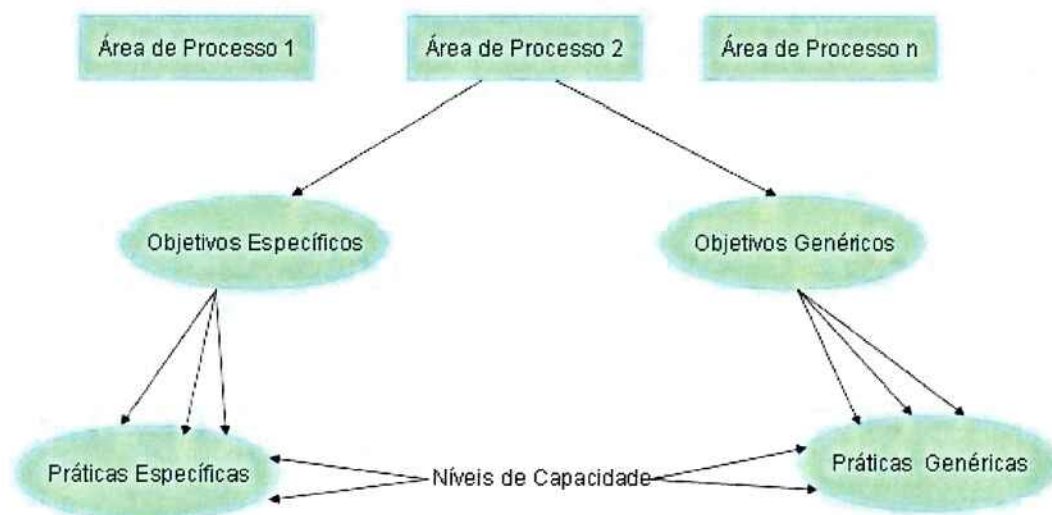


Figura 2: Componentes Modelo CMMI representação Contínua [17]

Como ilustra a figura 2, os objetivos específicos organizam as práticas específicas e os objetivos genéricos organizam as práticas genéricas. Cada prática genérica e específica correspondem a um nível de capacidade. Objetivos e práticas específicas são aplicados às Áreas de Processo.

Objetivos genéricos e práticas genéricas são aplicados a diversas Áreas de Processo. Os objetivos genéricos e as práticas genéricas definem uma sequência de níveis de capacidade que representam melhorias na implementação e eficiência de todos os processos a que são aplicados

Na representação contínua, os Níveis de Capacidade fornecem um ordem recomendada para a melhoria de processos dentro de cada Área de Processo. Ao mesmo tempo, permite uma flexibilidade em relação à ordem em que as Pas são atendidas.

2.1.5 Componentes Obrigatórios, Esperados e Informativos[9]

Os componentes de um modelo CMMI são agrupados em três categorias:

- **Obrigatórios:** Objetivos genéricos e específicos são componentes obrigatórios do modelo. Estes componentes devem ser cumpridos pelos processos planejados ou em execução da organização. Os componentes obrigatórios são essenciais para a avaliação das Áreas de Processo.

- **Esperado:** Práticas específicas e genéricas são componentes esperados pelo modelo. Estes componentes descrevem o que as organizações tipicamente implementam para satisfazer os componentes obrigatórios.
- **Informativos:** Subpráticas, Produtos de trabalhos típicos, amplificação de disciplinas, elaboração de práticas genéricas, títulos de objetivos e práticas, notas de práticas e objetivos, referências e componentes informativos do modelo que auxiliam os usuários a entender os objetivos e práticas específicas e o modo pelo qual podem ser satisfeitos.

2.1.6 Componentes do Modelo[9]

Áreas de Processo (PA): Uma área de processo é um cluster de práticas relacionadas de uma área que quando realizadas coletivamente satisfazem um conjunto de objetivos considerados importantes para a realização de melhorias significantes neste contexto.

Objetivos Específicos: Os objetivos específicos são aplicados a uma área de processo e atendem as características únicas que descrevem o que deve ser implementado para satisfazer esta área de processo. Objetivos Específicos são componentes obrigatórios do modelo e são utilizados nas avaliações para auxiliar na determinação se uma área é satisfeita.

Práticas Específicas: Uma prática específica é uma atividade considerada importante para o alcance de um Objetivo Específico associado. As práticas específicas descrevem os resultados esperados das atividades ao atingirem os objetivos da Área de Processo. As práticas específicas são componentes esperados no modelo

Aspectos Comuns: Quatro Aspectos Comuns organizam as Práticas Genéricas de cada área de processo e são componentes do modelo que não são avaliados mas apresentam as práticas genéricas:

- Compromissos (CO)
- Habilidades (AB)
- Diretrizes para implementação (DI)
- Verificações (VE)

Produtos de trabalhos típicos: Os produtos de trabalho típicos são um componente informativo do modelo que fornecem exemplos de saídas de Práticas Específicas ou Genéricas.

Sub Práticas: As Sub Práticas são descrições detalhadas que servem de guia para a interpretação de Práticas Específicas e Genéricas. Fornecem idéias que podem ser úteis para o processo de melhorias

Amplificação de disciplinas: São componentes informativos do modelo que contém informações relevantes a uma disciplina em particular sendo associadas a uma Prática Específica.

Objetivos Genéricos: Objetivos genéricos são assim chamados por abrangerem múltiplas áreas de processo. O atendimento de um Objetivo Genérico em uma Área de Processo significa uma melhora controlada no planejamento e implementação do processo associado a esta PA e indica se este processo é efetivo, repetível e duradouro. Objetivos Genéricos são componentes obrigatórios e são utilizados nas avaliações para determinar se uma área de processo é satisfeita.

Práticas Genéricas: As Práticas Genéricas institucionalizam os processos relacionados a uma Área de Processo garantindo que sejam efetivos, repetíveis e duradouros. Práticas Genéricas são categorizadas em Objetivos Genéricos e Aspectos Comuns e são componentes desejados no modelo CMMI.

Elaboraões de Práticas Genéricas: São componentes informativos do modelo que aparecem em cada Área de Processo e servem de guia para demonstrar como as Práticas Genéricas devem ser aplicadas exclusivamente a cada PA.

2.2 Norma ISO/IEC TR 15504

2.2.1 Introdução[18]

Inúmeros modelos foram definidos com o intuito de definir as melhores maneiras de melhorar e disponibilizar meios para certificar a capacidade dos processos de software. Exemplos destes esforços são a Norma ISO 9001[10], CMMI[9] e outros. De modo geral, estes modelos definem as características ideais de um processo

permitindo que seja possível atribuir um determinado nível de capacidade definido pelo modelo.

Segundo Salviano, [18], nos últimos anos, a ISO em conjunto com a comunidade internacional através do projeto SPICE (Software Process Improvement and Capability dEtermination) vem desenvolvendo um modelo único, mais abrangente que os modelos existentes, como o CMMI [9], e mais específico que a Norma ISO 9001 [10].

A futura norma ISO/IEC 15504, foi publicada pela primeira vez em 1998 como um Relatório Técnico do Tipo 2: ISO/IEC 15504. Um relatório técnico deve ser revisto num período de três anos após sua publicação já que o seu assunto ainda está em desenvolvimento técnico ou não foi possível alcançar um entendimento necessário para uma norma Mundial.

Por isto, os trabalhos de revisão ISO/IEC TR 15504 começaram imediatamente após sua publicação. O lançamento da norma ISO/IEC 15504 está programado para ocorrer entre 2003 e 2005 sendo que algumas partes já foram publicadas. Para este trabalho iremos nos concentrar no Relatório Técnico.

O ISO/IEC TR 15504 sob o título genérico Tecnologias de Informação – Avaliação de Processos de Software, é composta por nove partes:

Parte 1 (informativa) - Conceitos e guia introdutório: é uma introdução à ISO/IEC TR 15504. Descreve as interações das partes da norma e orienta a sua seleção e utilização. Ainda descreve os requisitos contidos na ISO/IEC TR 15504 e a sua aplicabilidade na condução de uma avaliação.

Parte 2 (normativa) - Um modelo de referência para processos e para capacidade de processos: define um modelo de referência bidimensional que descreve os processos e os seus níveis de capacidade usado nas avaliações. O modelo de referência define um conjunto de processos em termos de objetivos e resultados, e uma plataforma para a apreciação das capacidades do processo através da avaliação dos seus atributos estruturados em níveis de capacidade. São definidos requisitos para o estabelecimento de compatibilidade entre diferentes modelos de avaliação e o

modelo de referência. Um adendo à Parte 2 foi publicado em 2003 para incluir nas definições processos relacionados à aquisição.

Parte 3 (normativa) - Execução de uma avaliação: define os requisitos para a execução da avaliação permitindo a obtenção de resultados confiáveis, consistentes e verificáveis.

Parte 4 (informativa) - Guia para a execução de avaliações: fornece linhas de orientação para a execução de avaliações de processos de software e a interpretação de requisitos do ISO/IEC TR 15504-2 e ISO TR 15504-3 para avaliações em contextos diferentes. Este conjunto de orientações cobre: a seleção e utilização de um processo para a avaliação; de um modelo de avaliação compatível; e de um instrumento ou ferramenta de suporte à avaliação. Estas linhas de orientação são suficientemente genéricas para serem aplicadas a todas as organizações, para a execução de avaliações utilizando uma variedade de métodos e técnicas, e para serem suportadas por uma série de ferramentas.

Parte 5 (informativa) - Um modelo de avaliação e linhas de orientação para indicadores: fornece um exemplo de um modelo para a execução de avaliações de processos compatível com o modelo de referência do ISO/IEC TR 15504-2. Os(s) modelos(s) de avaliação estendem o modelo de referência introduzindo um conjunto de indicadores que facilitam a compreensão do desempenho e capacidade dos processos.

Parte 6 (informativa) - Guia para competência de avaliadores: descreve o perfil, competência, formação, treino e experiência, necessários para os condutores da avaliação de processos. Esta parte descreve os mecanismos que podem ser usados para demonstrar a competência e para validar a formação, treino e experiência dos avaliadores.

Parte 7 (informativa) - Guia para melhoria de processos: descreve o procedimento para se definir as entradas da avaliação e como utilizar os resultados da mesma para a melhoria de processos. A parte 7, apresenta ainda exemplos da aplicação do processo de melhoria em variadas situações.

Parte 8 (informativa) - Guia para determinação da capacidade dos processos do fornecedor: descreve o procedimento para se definir as entradas da avaliação e como analisar os resultados da mesma com o objetivo de determinar a capacidade dos processos. A determinação da capacidade pode ser utilizada pelas organizações para uma auto-avaliação ou para avaliar um fornecedor em potencial.

Parte 9 (normativa) - Vocabulário: apresenta o vocabulário consolidado de todos os termos específicos definidos no âmbito do ISO/IEC TR 15504.

2.2.2 O modelo de referência bidimensional – ISO/IEC TR 15504-2 [19]

A parte 2 da norma ISO/IEC TR 15504, define um modelo de referência para processos de software e para capacidade de processos que definem a base da avaliação de processos. O modelo de referência define em alto nível os objetivos fundamentais que formam a essência da boa prática da engenharia de software entretanto estes objetivos descrevem o que deve ser alcançado e não as melhores práticas.

Este modelo de referência pode ser aplicado às organizações de software que desejam estabelecer e aprimorar sua capacidade na aquisição, fornecimento, desenvolvimento, operação e manutenção de software.

A arquitetura do modelo de referência é composta por duas dimensões:

- a dimensão de **processo**, define o propósito dos processos, ou seja, seus objetivos essenciais e mensuráveis.
- a dimensão de **capacidade do processo**, uma série de atributos de processo, aplicáveis a qualquer processo e que representam características mensuráveis fundamentais para a gerenciá-lo e melhorar sua capacidade de execução.

A) A dimensão do processo [19]

A dimensão do processo agrupa os processos em três grupos : processos fundamentais, processos de Suporte ou Apoio e processos Organizacionais conforme a tabela 5.

Tabela 5: Grupos e categorias de processos

Grupo de Processos	Categoria de Processos
Primários	Aquisição/Fornecimento
	Engenharia
Apoio	Suporte
Organizacionais	Gerência
	Organização

Os processos primários compreendem as categorias de processo de Aquisição/Fornecimento e Engenharia. Os processos da categoria Aquisição/Fornecimento envolvem diretamente o cliente, apoiando o desenvolvimento e a transição do software ao ambiente do cliente e a correta operação e utilização de qualquer produto ou serviço. Já os processos de Engenharia são os que diretamente especificam, implementam ou mantêm o produto de software.

Os processos de Apoio são processos que apóiam outros processos ao longo do ciclo de vida do software.

Os processos Organizacionais compreendem as categorias Gerência e Organização. Os processos de Gerência contem as práticas universais de gestão que podem ser aplicadas a qualquer tipo de projeto ou processo dentro de um ciclo de vida de software. A categoria de processos Organizacionais engloba os processos que auxiliam as organizações a alcançarem seus objetivos de negócio.

B) A dimensão de capacidade do processo [19]

A dimensão de capacidade do processo é dividida em cinco níveis. Os atributos ou características de um processo são avaliados em uma escala de completeza fornecendo uma medida de sua capacidade

Cada atributo ou característica descreve um aspecto da capacidade de gerenciamento e melhoria de eficiência de um processo no cumprimento de seus objetivos e na sua contribuição para a satisfação dos objetivos do negócio.

Um nível de capacidade é caracterizado por um conjunto de atributos que permitem a melhora da capacidade de execução de um processo. Os níveis são uma maneira racional de se caminhar de níveis inferiores rumo a melhores níveis de capacidade de qualquer processo.

O modelo de referencia estabelece seis níveis de capacidade:

Nível 0 - Incompleto: De um modo geral o objetivo do processo não é alcançado. Os produtos de trabalho ou as saídas do processo dificilmente podem ser identificados.

Nível 1 - Executado: O objetivo do processo geralmente é alcançado. A realização do objetivo nem sempre é planejado e rastreado. Os indivíduos da organização reconhecem as ações que devem ser executadas e há um consenso que estas devem ser executadas assim que solicitadas. Há produtos de trabalho identificáveis para o processo que atestam a realização de seu objetivo.

Nível 2 - Gerenciado: O processo produz produtos de trabalho em consonância com os procedimentos especificados, é planejado e monitorado. Os produtos de trabalho estão em conformidade com os requisitos e padrões especificados. A primeira distinção entre o nível Executado e o Gerenciado é que o processo gera artefatos com o nível de qualidade especificado dentro dos prazos e recursos conhecidos.

Nível 3 - Estabelecido: As atividades são executadas e gerenciadas utilizando um processo definido a partir dos bons princípios de engenharia de software. Implementações individuais do processo utilizam versões customizadas e aprovadas para obter os resultados do processo. Os recursos necessários para o estabelecimento da definição do processo estão disponíveis. A primeira distinção do nível Gerenciado é que o processo do nível Estabelecido utiliza um processo definido, capaz de alcançar os resultados desejados.

Nível 4 - Previsível: O processo definido é executado consistentemente no dia a dias dentro dos limites de tolerância definidos para o cumprimento dos objetivos do processo. Medidas de performance detalhadas são coletadas e analisadas permitindo a avaliação quantitativa da capacidade do processo. A capacidade de se prever e gerir a performance são substancialmente ampliados. A performance é gerenciada quantitativamente pois a qualidade dos produtos de trabalho é quantitativamente conhecida. A principal diferença entre este nível e o Estabelecido é que o processo definido é executado consistentemente dentro dos limites estabelecidos e os resultados desejados são alcançados

Nível 5 - Otimizado: A performance do processo é otimizada de acordo com as necessidades atuais e futuras do negócio, sendo que os objetivos de negócio definidos para o processo são alcançados repetitivamente. A partir dos objetivos de negócio da organização são estabelecidas metas quantitativas de performance e eficiência para o processo. O monitoramento contínuo destes objetivos é realizado a partir dos dados quantitativos. A melhoria do processo é obtida através da análise destes dados, e da adequação dos processos ineficientes aos objetivos do negócio. A diferença fundamental do nível Previsível é a de que o processo padrão definido pode ser dinamicamente alterado e adaptado para que os objetivos atuais e futuros do negócio sejam eficientemente cumpridos.

2.3 Necessidade de estratégia de implantação de processos

O desenvolvimento de software apresenta aspectos únicos que não são reproduzidos em outros segmentos tecnológicos. Uma das descobertas do estudo da qualidade é a de que a qualidade do produto é determinada pela qualidade do processo que o produz.

Aplicando este conceito à produção de software pode-se verificar a qualidade do produto final avaliando o modo como foi desenvolvido. Por este motivo, os esforços mais recentes para a garantia da qualidade são na sua maioria voltados para o melhoramento do processo de desenvolvimento de software.

Neste cenário, onde a estratégia para a implantação de processos torna-se extremamente necessária já que podem, se bem sucedidas, aumentar a produtividade das equipes técnicas, a previsibilidade e rentabilidade dos projetos e ao mesmo tempo apoiando o desenvolvimento de software de melhor qualidade e de acordo com os requisitos dos clientes.

É importante lembrarmos que o oposto também pode ocorrer caso a implantação não seja adequada. Daí a importância de se estabelecer uma estratégia abrangente, baseada em metodologias já conhecidas e que tem sua eficácia comprovada.

3. MÉTODOS DE MELHORIA DE PROCESSOS

3.1 PDCA

3.1.1 Histórico

O ciclo de melhorias PDCA, sigla de Planejar – Executar – Verificar – Atuar em inglês (Plan-Do-Check-Act), é o método de melhorias mais antigo, tendo sido originalmente desenvolvido na década de 30, nos laboratórios da Bell Laboratories – EUA, pelo estatístico americano *Walter A. Shewhart*. [20] Segundo Andrade, o método consiste em um ciclo de controle estatístico do processo, que pode ser repetido indefinidamente sobre processos ou problemas. [13]

Na década de 50, logo após a Segunda Guerra Mundial, o especialista em qualidade *W. E. Deming* torna o método mundialmente conhecido ao aplicá-lo aos conceitos de qualidade em trabalhos realizados na indústria Japonesa. [13]

Além de ser responsável pela popularização do método, Deming o refinou, desenvolvendo o que por ele foi chamado de Shewart PDCA Cycle. [21]

Segundo Kinnula, existem muitas variações deste método, sendo que a maioria se diferencia do ciclo PDCA, graças a pequenas melhorias, como por exemplo, a adição de etapas no início do ciclo ou entre suas etapas intermediárias. [20]

3.1.2 Descrição

Para Basili, o objetivo do método é otimizar e melhorar um único modelo de processo, ou linha de produção. Para tanto, utiliza técnicas, como o controle estatístico da qualidade e ciclos iterativos em experimentos de métodos de melhoria, e, na construção de modelos de previsão do produto. [14]

O modelo PDCA é constituído de quatro etapas básicas representadas pelas iniciais das palavras, em seu idioma de origem, PLAN, DO, CHECK, ACT, o que significa PLANEJAR; EXECUTAR; VERIFICAR, ATUAR. (conforme a figura 3).

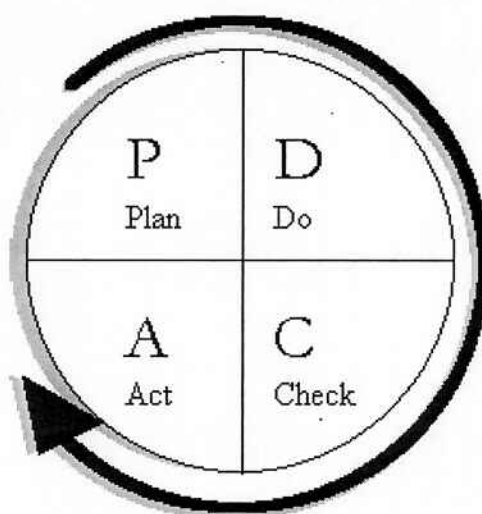


Figura 3: Ciclo PDCA [21]

Segundo Andrade, o ciclo PDCA deve ser utilizado como um modelo dinâmico. O término de uma volta no ciclo implica no início de um novo ciclo, e assim por diante [13]. Seguindo o espírito da melhoria de qualidade contínua, o processo sempre pode ser revisto resultando em um novo processo de mudança. [21]

Para Basili, apesar do modelo ter sido desenvolvido para a indústria manufatureira, ele vem sendo aplicado por muitas empresas na melhoria de seus processos de desenvolvimento de software. [14]

3.1.3 Estratégia e Processo

Segundo Basili, a idéia básica do método é a de que se um Processo (P) produz um Produto (X) então ao aplicarmos a abordagem PDCA, introduziremos uma série de modificações do processo P (cada modificação é considerada uma evolução de uma versão prévia do produto propriamente dito, ou da eficiência do processo), resultando em uma família de processos e uma série de versões do produto X:

$$P_0, P_1, P_2, \dots, P_n \longrightarrow X_0, X_1, X_2, \dots, X_n$$

onde P_i , representa uma melhoria de P_{i-1} e/ou X_i possui uma melhor qualidade do que X_{i-1} . [14]

Esse ciclo ininterrupto de mudanças e melhorias incrementais pode ser representado pela rampa de melhoria (conforme a figura 4), onde o aprendizado obtido em uma

aplicação do ciclo PDCA, pode iniciar outro ciclo, em uma tentativa mais complexa, e assim sucessivamente. [13]

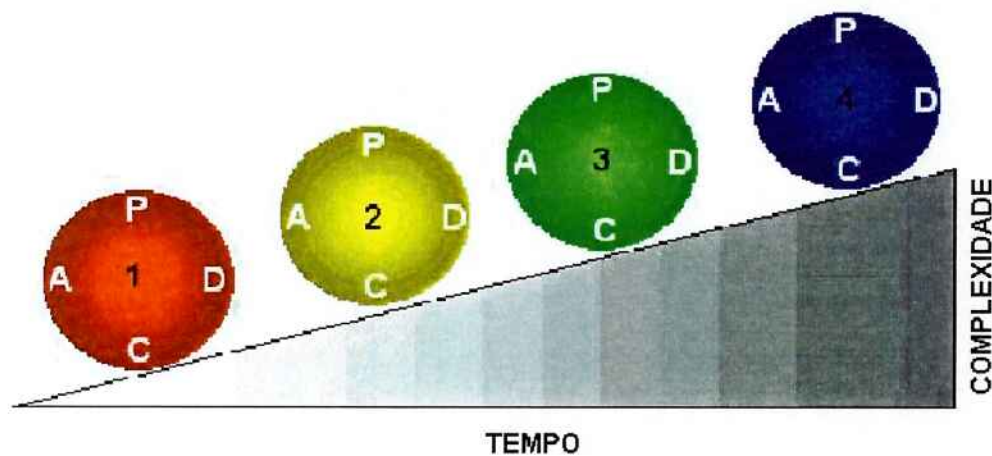


Figura 4: Rampa de melhorias PDCA [13]

O ciclo PDCA pode também ser encarado como uma lista de verificação de quatro etapas obrigatórias, para se ir de um problema detectado, para um problema resolvido.[21]

Na etapa PLANEJAR, deve-se planejar a melhoria de suas operações a partir da identificação dos problemas, ou oportunidades de melhoria, e da proposição de idéias para solucioná-los. [21]

Durante a etapa EXECUTAR, as idéias propostas no plano de ação elaborado na etapa anterior, serão colocadas em prática. Em um primeiro momento, as modificações devem ser executadas de modo experimental e em pequena escala. Desta forma, o desconforto gerado pela quebra de rotina é minimizado até a verificação do funcionamento das mudanças conforme idealizadas. [21]

Após a execução experimental e em pequena escala das mudanças é hora de VERIFICAR se os resultados desejados estão sendo alcançados. É fundamental, o monitoramento continuado das atividades chave para se garantir o conhecimento do nível de qualidade do produto gerado e para detectar o surgimento de novos problemas. [21]

Por fim, durante a etapa ATUAR, se o experimento for um sucesso, as mudanças serão implementadas em uma escala maior, tornando-as atividades parte da rotina da organização. Para facilitar a transição, é fundamental obter o envolvimento das pessoas, departamentos, clientes ou fornecedores afetados pelas mudanças e aqueles cuja cooperação seja necessária para implementá-las em larga escala, ou ainda aqueles que podem se beneficiar com o aprendizado obtido. [21]

Desta forma, o ciclo se completa e o problema é resolvido. Neste caso deve-se retornar a etapa PLANEJAR para a identificação do próximo problema a ser solucionado, dando início a uma nova iteração no ciclo PDCA. Nos casos em que o experimento falhar, deve-se pular a etapa ATUAR e retornar ao módulo PLANEJAR; buscar novas idéias para solucionar o problema e iniciar o ciclo novamente. [21]

Para auxiliar na execução dos estágios das atividades de melhorias deve-se utilizar técnicas e ferramentas específicas de gestão da qualidade, como as listadas na figura 5:

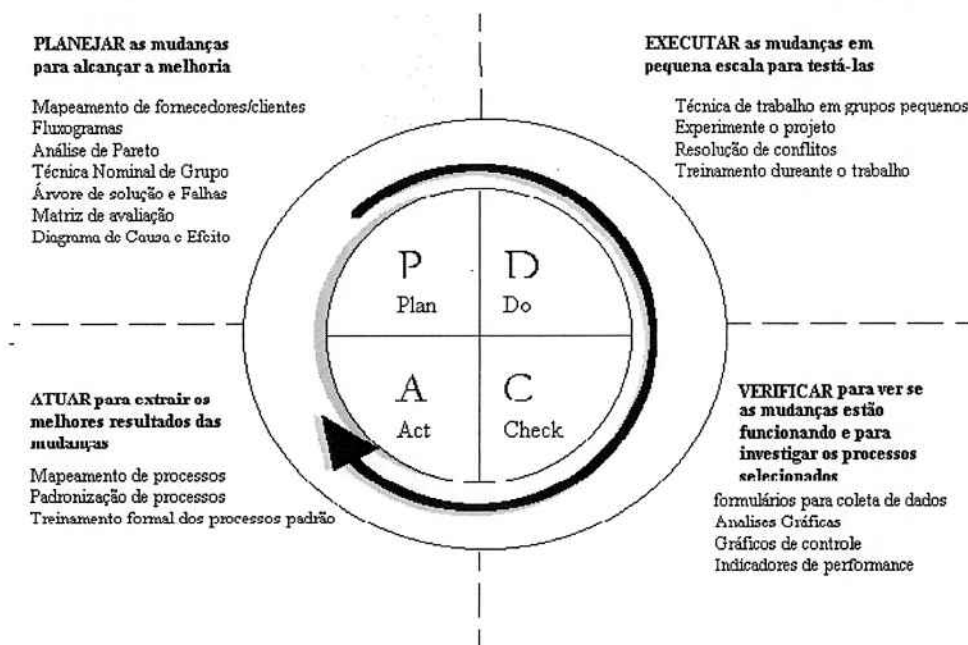


Figura 5: Quadro de ferramentas de Gestão da Qualidade e ciclo PDCA [13]

A classificação das ferramentas em seções do ciclo PDCA apresentada na figura 5 não restringe seu uso a essas etapas mas auxilia a escolha de ferramentas para serem utilizadas em cada estágio do ciclo. [13]

3.2 QIP

3.2.1 Histórico

O modelo de melhorias QIP, sigla de Quality Improvement Paradigm, é o resultado do esforço de cooperação entre a NASA Goddard Space Flight Center / Flight Dynamics Division e o departamento de Ciência da Computação da Universidade de Maryland (UM) dos Estados Unidos (EUA). O convênio, denominado Software Engineering Laboratory (SEL), foi firmado em 1976 com o objetivo de reduzir a taxa de defeitos, o custo e duração do ciclo de desenvolvimento de software.[22] .

Fruto das experiências do SEL, o modelo QIP, desenvolvido para apoiar o processo de melhoria e adoção tecnológica, foi publicado em 1984 por Victor Basili. [23] Desde então, o modelo tem sido aprimorado e refinado sendo este trabalho realizado atualmente pelo Instituto Fraunhofer. [24]

3.2.2 Descrição

O propósito do modelo QIP é o de apoiar o processo de melhoria contínua, a engenharia dos processos de desenvolvimento e para auxiliar na adoção de novas tecnologias. Uma maneira de interpretar o método é vê-lo como um modelo para apoiar o aprendizado, onde a organização através da experimentação estabelece maneiras de desenvolver práticas, capturá-las e empacotá-las de forma a ser possível reutilizá-las em outras experiências , considerando algumas restrições. [20]

Conforme apontando por Basili, o modelo é baseado no princípio de que a disciplina de software é evolutiva e experimental. O trabalho de desenvolver software, distintamente do trabalho fabril, é essencialmente humano e de projeto, portanto de pouca repetição. Esta premissa torna o uso do controle estatístico, como utilizado na engenharia de produção, extremamente difícil e duvidoso. [14]

Segundo Kinnula, a partir dessas constatações, Victor Basili, desenvolveu uma abordagem diferente das adotadas nos modelos baseados no controle estatístico dos

processos, como por exemplo o CMMI [20]. O QIP parte da premissa de que todos os ambientes de projetos e produtos são diferentes, portando certos pré-requisitos para a reutilização devem ser respeitados. Esses pré-requisitos incluem a identificação e o empacotamento das experiências, explicações do tipo de projeto onde foram aplicadas com sucesso ou não, e como adaptá-las em produtos e ambientes distintos. [14]

O desenvolvimento de experiências reutilizáveis é análogo ao desenvolvimento de código reutilizável. No modelo QIP, isto é realizado através das experimentações repetidas e na existência de recursos dedicados ao empacotamento. [14]

A filosofia do método é a realização de estudos científicos através da experimentação. É uma abordagem de baixo para cima (bottom-up), com ciclos definidos e processo iterativo. [20]

3.2.3 Estratégia e Processo

De acordo com Basili, o QIP é o resultado da aplicação do método científico ao problema da melhoria de qualidade de software, sendo inspirado no ciclo de Shewart-Deming PDCA. O QIP é uma abordagem de qualidade com ênfase na melhoria contínua apoiada no acúmulo de conhecimento obtido tanto na realização de experimentos nos processos da organização como na aplicação de medições para verificar a validade dos resultados.[14]

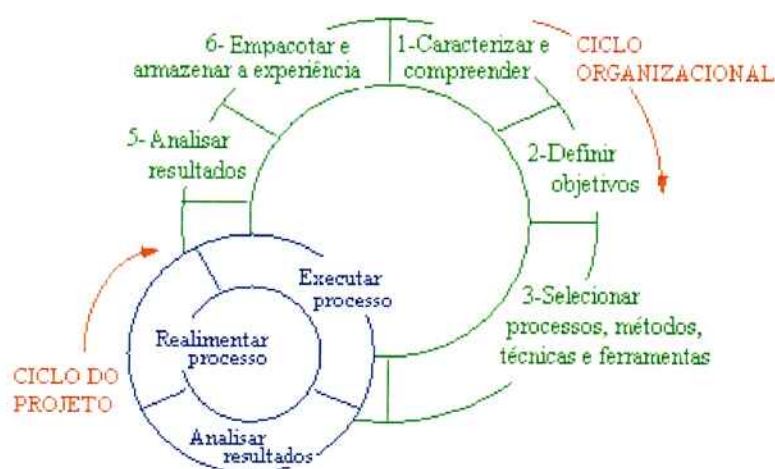


Figura 6: Ciclo QIP [20]

Segundo Kinnula, o ciclo QIP é formado por dois ciclos bem definidos: um grande ciclo denominado de *ciclo Organizacional* e um ciclo menor denominado de *ciclo do Projeto*, conforme ilustra a figura 6.

A função do *ciclo do Projeto* é prover informações durante a execução do projeto para prevenir e solucionar os problemas, monitorar e acompanhar o andamento do projeto e ajustar o processo selecionado aos objetivos definidos. Já o *ciclo Organizacional* tem o propósito de analisar a conformidade dos dados coletados em relação aos modelos e ao histórico dos experimentos anteriores, permitindo um melhor entendimento do experimento concluído e o acúmulo de conhecimento na forma de experiências reutilizáveis. [20]

A) O Ciclo Organizacional

O *ciclo Organizacional* representa o modo como a organização aprende sendo dividido em seis fases:

1. Caracterizar e Compreender
2. Definir Objetivos
3. Selecionar processos, métodos, técnicas e ferramentas
4. Executar os processos (executar o ciclo de projeto)
5. Analisar os resultados
6. Empacotar e armazenar a experiência

O ciclo QIP pode ser utilizado tanto como uma ferramenta para aprender alguma experiência já empacotada, ou para criar uma totalmente nova. Quando o objetivo for produzir uma nova experiência e empacotá-la para reuso futuro, a quarta fase (Executar os processos) deve ser executada algumas vezes. A razão disto, é que a experiência não deve ser empacotada enquanto não houver conhecimento suficiente sobre sua aplicabilidade e pré-requisitos.[20]

A etapa inicial do ciclo é a *Caracterizar e Compreender*. O foco é descrever e compreender o projeto em questão e seu ambiente, considerando os processos,

produtos, modelos de qualidade, dados e experiências disponíveis. Durante esta etapa, com base nas experiências anteriores, objetivos verificáveis e sua criticidade são estabelecidos. A caracterização modela diversos fatores e estuda suas interações, aumentando o entendimento do contexto onde a melhoria deverá ser implantada. Os fatores a serem considerados incluem pessoas, problemas pessoas, produtos, recursos, e assim por diante. [14]

Durante a segunda fase, *Definir objetivos*, ocorre à definição de objetivos que caracterizem um projeto de sucesso (abrangendo tanto o processo como o produto) e os resultados da melhoria. A idéia é obter dados confiáveis e mensuráveis da melhoria e, por este motivo, os objetivos devem ser quantificáveis e definidos segundo uma variedade de perspectivas, incluindo os pontos de vista do cliente, do projeto e da organização. Para serem aplicáveis ao esforço em questão, os objetivos devem ser baseados na caracterização inicial e nos objetivos estratégicos da melhoria. A medição exerce diversos papéis na engenharia de processos. Quando as medições são registradas, elas criam uma memória organizacional. Ao longo de projetos, elas provêm idéias sobre os pontos fortes e fracos do processo corrente e seus produtos. Elas são ferramentas de tomada de decisão, pois podem ser utilizadas para decidir quais técnicas devem ser adotadas e refinadas. Elas também fornecem evidências objetivas do impacto das mudanças e, em geral, podem ser utilizadas para verificar a qualidade do processo e do produto. [20]

O objetivo da terceira fase conforme o próprio nome diz é o de *Selecionar processos, métodos, técnicas e ferramentas* que sejam apropriados para o projeto. A decisão é baseada na caracterização do ambiente e do produto e nos objetivos anteriormente definidos. A seleção deve ser consistente com os objetivos definidos para os produtos e processos, do contrário não haverá sentido em realizar as medições que serão derivadas dos objetivos estabelecidos. Os processos, métodos de apoio, técnicas e ferramentas selecionados precisam ser ajustados para juntos formarem um conjunto integrado, que seja aplicável ao projeto e ao seu contexto. [22]

Durante a quarta fase do ciclo organizacional o projeto selecionado é executado conforme o processo definido. Do ponto de vista da organização, esta fase é onde o ciclo do projeto acontece. Esse ciclo será descrito no próximo item.

A Quinta fase é a Analisar os Resultados para avaliar as práticas, determinar os problemas, registrar as descobertas e fazer recomendações para os próximos projetos de melhoria. Os dados são analisados em relação aos objetivos, são utilizados para obter uma melhor caracterização e compreensão do contexto, verificar e analisar o experimento (melhorias), determinar os problemas, adquirir mais informação para ser utilizada em melhores previsões e controle, e para motivar melhorias futuras. [20]

A Sexta e última fase no ciclo organizacional é o Empacotamento da Experiência e seu armazenamento na base de experimentos para uso futuro. Nesta fase, a experiência obtida é consolidada na forma de novos modelos, referências ou em outras formas de conhecimento estruturado. Os tipos de experiências empacotados cobrem modelos de previsão (como esforço ou defeitos), modelos de produtos, definições de processos, avaliação de métodos e técnicas, modelos de qualidade, produtos propriamente ditos, lições aprendidas e guias de adaptação. [20]

B) O Ciclo do Projeto

O ciclo do projeto, que representa como o projeto aprende e se guia, é dividido em três atividades:

1. Execução do projeto
2. Análise dos resultados
3. Realimentar o processo.

O projeto executa o processo para construir os produtos. Ao mesmo tempo dados dos recursos, processos, e produtos são coletados, validados e os resultados são analisados para medir e verificar o quanto dos objetivos foram alcançados. Esta informação é então novamente introduzida ao projeto para a tomada de ações corretivas. A coleta de dados precisa ser integrada ao processo para torná-lo factível. Os dados precisam ser validados pois são sujeitos a erro, sendo necessário treinar os envolvidos em sua coleta e interpretação garantindo o entendimento do modelo por todos. [22]

3.3 IDEAL

3.3.1 Histórico

Para atender a demanda de apoio das organizações que desejavam iniciar ou manter um projeto de melhoria de processos de software, o SEI formou uma equipe multidisciplinar para sintetizar conhecimentos e experiências e criar um guia de implementação contendo recomendações e passos a serem seguidos. [25]

Em 1994, foi publicado o trabalho *Software Process Improvement Roadmap* e dois anos depois a primeira versão do guia IDEALSM: *A User's Guide for Software Process Improvement* [15].

O modelo IDEALSM foi concebido originalmente como um modelo de ciclo de vida para melhoria de processos de software baseado no CMM. Posteriormente, o SEI percebeu que o modelo poderia se aplicado em outras áreas e decidiu revisá-lo e adaptá-lo. [26]

Em 1997, foi lançada a versão 1.1 do modelo IDEALSM reforçando a intenção de tornar o IDEALSM uma abordagem sistêmica para gerir o ciclo de vida de adoção de novas ferramentas de engenharia de software, processos e métodos. [26]

3.3.2 Descrição

O modelo IDEALSM é um modelo de melhoria organizacional que abrange todas as etapas de uma iniciativa de melhoria desde sua iniciação até sua implementação. O nome do modelo é formado pela letra inicial em inglês, das cinco etapas que o compõe: iniciando (I-Initiating), diagnosticando (D-Diagnosing), estabelecendo (E-Establishing), atuando (A-Acting) e aprendendo (L-Learning). Cada etapa, ou fase, é composta por uma série de atividades sendo que o ciclo completo é formado por 14 atividades, conforme figura 7.



Figura 7: O modelo IDEAL [15]

Segundo McFeeley, a duração de um ciclo, ou seja, o tempo necessário para se completar as cinco fases do modelo, irá variar de uma organização para outra. Em algumas organizações, dependendo dos recursos comprometidos no projeto de melhoria, é possível realizar algumas atividades em paralelo e em alguns casos, grupos de trabalho podem ao mesmo tempo realizar, atividades de fases distintas do ciclo pois o limite entre as fases não é tão rígido quanto demonstra a figura 7: O modelo IDEAL. [15]

3.3.3 Estratégia e Processo

Segundo McFeeley, um programa de melhoria de processos é constituído por dois elementos: um componente estratégico e um componente tático. As atividades do componente estratégico, de responsabilidade da gerência sênior¹, são baseadas nas diretrizes e necessidades da organização e tem por finalidade direccionar e priorizar as

¹O gerente mais alto da hierarquia e seus subordinados diretos em grandes e medias empresas. Seu foco e na manutenção do negócio ao longo prazo.

atividades do nível tático. Já as atividades do componente tático criam, modificam e executam os processos e são realizadas pelos gerentes operacionais e os usuários destes processos. [15]

McFeeley ilustra os componentes da atividade de melhoria na forma de um fluxo contínuo, representado na figura 8. À medida que a atividade de melhoria é completada, as atividades de ambos os níveis alimentam a etapa de aprendizado, onde o comprometimento da gerencia é reafirmado, novos planejamentos são realizados e a estratégia é revista.

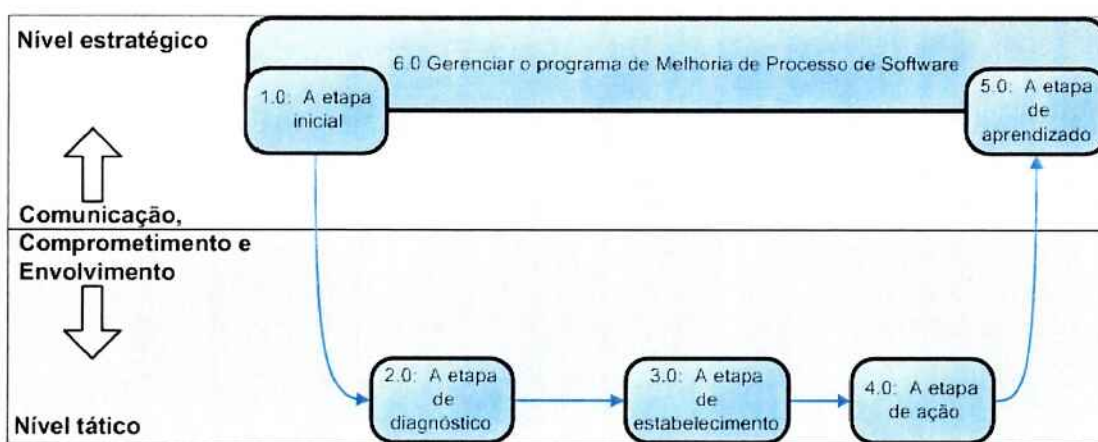


Figura 8: Visão bidimensional de uma atividade de melhoria de processo [15]

Como se pode notar McFeeley adiciona ao modelo a atividade 6.0 Gerenciar o programa de Melhoria de Processo de Software para prover uma visão gerencial do projeto. [15]

A tabela 6 apresenta um breve resumo dos objetivos das atividades do modelo:

Tabela 6: Atividades do modelo IDEALSM [15]

Atividade	Objetivo
1.0: A etapa inicial	Aprender sobre o processo de melhoria, obter comprometimento dos recursos e construir a infra-estrutura para iniciar o projeto
2.0: A etapa de diagnóstico	Identificar o nível de maturidade atual, descrever processos, métricas, etc. Iniciar o desenvolvimento do plano de ação.
3.0: A etapa de estabelecimento	Definir objetivos e prioridades, completar o plano de ação
4.0: A etapa de ação	Pesquisar e desenvolver soluções para os problemas de processo. Disseminar as melhorias bem sucedidas para o restante da organização.
5.0: A etapa de aprendizado	Preparar o próximo ciclo do modelo IDEAL SM . Aplicar o aprendizado para refinar o processo de melhoria
6.0 Gerenciar o programa de Melhoria de Processo de Software (MPS)	Prover uma visão geral do projeto e resolução de problemas

A) A etapa Inicial

Durante esta etapa, as razões que justificam o esforço de melhoria são claramente definidas, sendo fundamental a identificação de sua contribuição para os objetivos e metas do negócio bem como seu relacionamento com outras atividades da organização.

Para orientar os esforços na fase Inicial e nas demais um plano de melhoria de processo de software (MPS) deve ser gerado e aprovado pela alta gerência obtendo-se desta forma o comprometimento da organização para o fornecimento dos recursos necessários para a realização das próximas etapas.[15] .

Por fim, a infra-estrutura para gerir os detalhes da implementação é formada sendo tipicamente criados dois grupos: o MSG e o SEPG. O MSG, Grupo de Gerenciamento Estratégico tem a responsabilidade de alinhar o programa de melhoria de processo de software à visão e missão da organização, demonstrando apoio, alocando recursos, monitorando o progresso do projeto, direcionando e moderando as ações. Já o SEPG, Grupo de Engenharia de Processo de Software, é criado pela diretoria da organização para criar e reforçar o apoio ao Plano de Melhoria de Software. O SEPG, cultua e sustenta as atividades de melhoria assegurando a coordenação do Plano de Melhoria de Software através da organização

O estímulo para a mudança

Segundo Gembra e Myers, é importante justificar a mudança de práticas organizacionais a partir das necessidades do negócio pois a mudança pela mudança raramente resulta em ganhos significativos.[25]

Muitas vezes a motivação de um esforço de melhoria é fruto de uma determinação da alta gerência, outras vezes é fruto da comparação de dados obtidos graças a um esforço prévio de melhoria. De qualquer maneira, é importante que o estímulo para a mudança tenha apelo e alcance abrangente, facilitando o comprometimento de todos os envolvidos e ampliando as chances de sucesso da iniciativa. [25]

Definir Contexto

Segundo Gembra e Myers, após a clara identificação das razões para se iniciar a mudança, a gerência da organização deve definir o contexto do trabalho a ser realizado. É necessário analisar onde e como o esforço se encaixa na estratégia de negócios da organização indicando as metas e objetivos específicos que serão realizados pela mudança, qual será o impacto em outras iniciativas e trabalhos em andamento e por fim quais serão os benefícios conquistados (por exemplo, ROI, aumento de capacidade)

Ainda segundo Gembra e Myers, é importante ser o mais claro possível a respeito do contexto do projeto e suas implicações mesmo considerando que as mesmas se tornarão mais evidentes com o avanço do projeto. [25]

Obter patrocínio

Apoio efetivo é um dos mais importantes fatores para esforços de melhoria sendo necessário manter o nível de apoio durante todo o projeto. Segundo Gembra e Myers, logo no início do processo é especialmente importante obter suporte de gerências estratégicas para que seja possível transpor o período de incertezas e inseguranças pelo qual as organizações tipicamente se deparam. [25]

O comprometimento de recursos essenciais é um importante elemento do patrocínio, mas patrocinadores efetivos geralmente fazem muito mais do que isto, ao darem apoio e atenção pessoal ao projeto nos momentos mais difíceis.

Definir infra-estrutura

Após o entendimento das razões para a mudança, seu contexto e a obtenção do comprometimento dos patrocinadores, é necessário criar o mecanismo de gerenciamento da implementação do esforço.

A infra-estrutura necessária para o projeto pode ser temporária ou permanente, sendo seu tamanho e complexidade dependentes da natureza da melhoria. Para pequenos esforços, a infra-estrutura pode ser de um funcionário em tempo parcial. Já para um processo de melhoria de software, considerado complexo e grandioso, pode-se envolver de 2-3 % dos membros da organização. Em todos os casos, é

fundamental o desenvolvimento de acordos por escrito que documentem e esclareçam todas as expectativas, responsabilidades e níveis de autoridade dos envolvidos no projeto. [25] .

Segundo McFeeley, o principal propósito desta atividade é criar os mecanismos necessários para auxiliar a organização a institucionalizar o processo de melhoria contínua, sendo por este motivo, um fator crítico para o sucesso do projeto. Uma boa e sólida infra-estrutura é capaz de sustentar o projeto até que os resultados comecem a surgir e obtenham visibilidade . [15]

B) A etapa de diagnóstico

A etapa de diagnóstico complementa a etapa inicial desenvolvendo um entendimento mais profundo do trabalho de melhoria. Durante a fase de diagnóstico, duas caracterizações da organização são desenvolvidas: o estado atual da organização e o estado futuro a ser alcançado. Esses estudos são utilizados para o desenvolvimento da estratégia de melhoria das práticas organizacionais [25] .

Caracterizar o estado corrente e o desejado

Para auxiliar esta atividade deve-se utilizar modelos de referência, como o CMMI para software. Quando um modelo não estiver disponível, um bom ponto de partida são os fatores identificados na atividade *Estímulo para a mudança* da etapa Inicial. [25]

Para Gembra e Myers, é importante lembrar que esta atividade deve ser focada em elementos críticos para as mudanças propostas separando outros aspectos do trabalho da organização. [25]

Desenvolver Recomendações

De acordo com Gembra e Myers, o desenvolvimento de recomendações freqüentemente é realizado por um time experiente e com expertise adequado para esta atividade. As recomendações geralmente têm um grande peso nas decisões tomadas pelos gerentes-chave e patrocinadores pois sugerem uma maneira de se proceder nas atividades subseqüentes e propõe medidas para adequar o estado corrente ao desejado.[25]

C) A etapa de estabelecimento

O propósito da etapa de estabelecimento é o de desenvolver um plano de trabalho detalhado. As prioridades são definidas considerando-se as recomendações feitas na fase de diagnóstico, as demais operações da organização e as restrições do seu ambiente operacional. Com as prioridades definidas, é desenvolvida a estratégia do projeto e por fim é desenvolvido o plano de ação contendo as ações específicas, marcos, entregáveis e responsabilidades. [25]

Definir prioridades

Segundo Gembra e Myers, a primeira atividade desta fase é a definição de prioridades do esforço de melhoria. Estas prioridades devem levar uma série de fatores em consideração como disponibilidade de recursos, interdependências existentes entre as atividades recomendadas, fatores externos que podem intervir, cumprimento das prioridades globais da organização. [25]

Desenvolver estratégia

Ao combinar o melhor entendimento do escopo do trabalho com um conjunto de prioridades caminha-se para o desenvolvimento de uma estratégia de realização do trabalho e identificação da disponibilidade de recursos. Fatores técnicos podem incluir os detalhes da instalação de uma nova tecnologia e novas habilidades e conhecimentos requeridos para a utilização desta tecnologia. Fatores não técnicos, incluindo a cultura da organização, como ilhas de resistência, níveis de patrocinadores, forças de mercado, também devem ser consideradas. [25]

Planejar Ações

Com a estratégia definida, um plano de implementação detalhado pode ser desenvolvido. Este plano inclui cronograma, tarefas, marcos, pontos de decisão, recursos, responsabilidades, medições, mecanismos de rastreabilidade, risco e estratégias de mitigação, e outros elementos solicitados pela organização. [25]

D) A etapa de Ação

As atividades da etapa de Ação auxiliam a organização a implementar o trabalho concebido e planejado nas três etapas anteriores. Estas atividades tipicamente consumirão mais tempo e mais recursos do que a combinação de todas as demais fases. [25]

Criar a solução

A etapa de Ação começa reunindo todos os elementos-chave para a criação da primeira versão da solução dos problemas organizacionais identificados anteriormente. Esses elementos-chave podem incluir ferramentas existentes, processos, conhecimento e habilidades, assim como novos conhecimentos, informações e ajudas externas. A solução, que pode ser um pouco complexa e multi-facetada, é geralmente criada por um grupo de trabalho técnico. [25]

Testar Solução Piloto

Segundo Gembra e Myers, a versão inicial da solução deve ser testada e exercitada pois raramente funciona como planejada. Geralmente, um teste piloto é realizado mas outros meios podem ser utilizados. [25]

Refinar a solução

Após o teste, a solução deve incorporar o conhecimento, experiência e lições adquiridas durante o teste. Inúmeras iterações do processo de teste e refinamento podem ser necessárias para se alcançar uma solução satisfatória. Entretanto, deve-se lembrar que a busca pela solução perfeita pode atrasar desnecessariamente sua implementação. [25]

Implementar a solução

Uma vez que a solução é operacional, ela pode ser implementada na organização. Inúmeras técnicas de propagação podem ser utilizadas para uma implementação, incluindo estratégias de cima para baixo (top-down, inicia-se do nível mais alto da organização para os mais baixos) e just-in-time (implementando projeto por projeto no momento apropriado do seu ciclo de vida). Nenhuma estratégia de propagação é

universalmente melhor do que outra, mas a escolhida deve considerar a essência da melhoria e as circunstâncias organizacionais. Para grandes mudanças, a implementação pode demandar muito tempo e recursos. [25] incluindo estratégias de cima para baixo (top-down, inicia-se do nível mais alto da organização para os mais baixos).

E) A etapa de Aprendizado

A etapa de Aprendizado completa o ciclo de melhoria. Um dos objetivos do modelo IDEALSM é o de melhorar continuamente a capacidade de implementação de mudanças. Na etapa de Aprendizado, a experiência IDEALSM é revisada por completo para se determinar o que foi concretizado, se o esforço alcançou os objetivos definidos, e como a organização pode implementar mudanças de uma maneira mais efetiva e eficiente no futuro. Ao longo do ciclo IDEALSM deve-se manter registros com esta etapa em mente. [25] .

Analisar e validar

Esta atividade responde a diversas questões: De que forma o esforço realizou o propósito desejado? O que funcionou bem? Lições são coletadas, analisadas, resumidas, e documentadas. As necessidades de negócios identificadas na fase de iniciação são examinadas novamente com o intuito de verificar se foram alcançadas. [25]

Propor ações futuras

Durante esta atividade, recomendações baseadas na análise e validação são desenvolvidas e documentadas. Propostas de melhorias para implementações de mudanças futuras são fornecidas para os níveis de gerência adequados para sua apreciação podendo dar início a um novo ciclo IDEALSM . [25]

4. TAXONOMIA DOS MÉTODOS DE MELHORIA DE PROCESSOS DE SOFTWARE

Para realizar a comparação dos métodos de melhoria de processos utiliza-se o método de comparação de características. Lista-se uma série características tidas como relevantes e cada método será descrito em termos destas características. Por fim apresenta-se a comparação de forma tabular facilitando a leitura e permitindo consultas rápidas da comparação.

A comparação de características fornece uma visão geral dos métodos de melhoria e pode ser utilizada para a realização de outras comparações. Entretanto, a descrição das características é de alto nível e maiores detalhes devem ser obtidos em outros trabalhos.

As características escolhidas são preferencialmente objetivas, mensuráveis e comparáveis. O trabalho de Halvorsen foi uma importante fonte para a definição das características e do método de comparação utilizado. Este trabalho define uma taxonomia para a comparação de modelos de melhoria contínua de processos algumas dimensões foram adaptadas para o foco desse trabalho e outras descartadas.[27]

As características definidas foram agrupadas em cinco categorias para facilitar a compreensão e a leitura, conforme apresenta a tabela 7:

Tabela 7: Categorias adaptado de [27]

CATEGORIAS			
Geral	Processo	Organizacional	Resultados
Origem geográfica e difusão	Iniciação da melhoria	Tamanho da organização	Objetivo
Origem científica	Foco da melhoria	Infra-estrutura de engenharia de processos de software	Produtos de trabalho
Evolução Estabilidade	Técnicas de análise	Suporte aos objetivos de negócio	Validação
Popularidade	Identificação de oportunidades		
Restrito a software	Modelo de ciclo de vida		
	Modelo de melhora contínua		

4.1 Categoria Geral

Esta categoria descreve atributos genéricos ou características dos métodos de melhorias. Estas características são específicas a cada método e revelam como o método foi construído ou projetado.

Origem geográfica e difusão: Esta característica descreve a região do mundo onde o método foi criado e onde ele é mais utilizado atualmente.

Origem científica: A origem científica de um método de melhoria revela seus fundamentos filosóficos. Alguns métodos originaram-se puramente a partir de princípios filosóficos, outros entretanto fundamentam-se em regras estatísticas ou matemáticas. Em alguns casos, os métodos são baseados em um outro método se apropriando de seus conceitos, conteúdo e foco.

Evolução/Estabilidade: Para muitos a evolução do método e sua estabilidade são obtidas ao longo de anos de utilização. O retorno dos usuários deve ser utilizado para a reformulação do método tornando o mais fácil de ser utilizado e melhorando pontos fracos. A evolução do método também é importante para sua compatibilidade com novas estruturas organizacionais e modelos de melhoria contínua de processos de software.

Popularidade: A popularidade é um fator importante pois possibilita a troca de informações entre seus praticantes. Outro aspecto a ser considerado é o fato de que métodos populares tendem a ter um melhor suporte e a permanecerem evoluindo.

Restrito a software: O fato de o método ser aplicável somente à disciplina de software revela seu escopo. Alguns métodos , são direcionados à melhoria de processos de software enquanto outros são mais genéricos.

4.2 Categoria Processo

A categoria Processo compreende as características que descrevem como o método de melhoria pode ser utilizado.

Iniciação da melhoria: Esta característica indica em que nível hierárquico o esforço de melhoria tem início, ou seja, a estratégia adotada para atingir a melhoria desejada.

As estratégias mais comuns são as de cima para baixo (top-down, inicia-se do nível mais alto da organização para os mais baixos) e a de baixo para cima (bottom-up, inicia-se do nível organizacional mais baixo para os mais altos).

Foco da melhoria: Esta característica indica as atividades que os métodos consideram como fundamentais para a melhoria. Alguns métodos focam na reutilização de experiências, outros na gestão dos processos.

Identificação de oportunidades: Alguns métodos suportam a identificação de oportunidades de melhoria outros começam a partir do planejamento do esforço de melhoria.

Modelo de ciclo de vida: Alguns métodos foram criados para suportar uma ação de melhoria, ou para descrever as principais fases da melhoria contínua de um processo em particular. Poucos modelos oferecem suporte a programas de melhoria de processos de software. A grande distinção aqui é a duração do esforço. Enquanto um ciclo para uma ação de melhoria pode durar meses, um ciclo de um programa de melhoria de software pode levar anos. [20]

Kinnula sugere a classificação dos modelos de vida dos métodos de melhoria em duas categorias: Programa ou Ação. [20]

Modelo de melhoria contínua: Alguns métodos de melhoria dão suporte a modelos de melhoria contínua de processos de software. Um exemplo disto é o modelo IDEALSM criado pelo SEI para auxiliar as organizações a implantar o modelo CMMI.

4.3 Categoria Organizacional

As características desta categoria estão relacionadas aos atributos da organização e ao ambiente no qual o método de melhoria é utilizado, as pessoas envolvidas e em que tipo de organização o método é aplicável.

Tamanho da organização: Esta característica indica o tamanho da organização na qual o método de melhoria é melhor aplicável. Alguns métodos são muito abrangentes e por isto requerem muitos recursos dificultando sua aplicação em pequenas empresas.

Infra-estrutura de engenharia de processos de software: Alguns métodos de melhoria estão intimamente ligados a elementos que suportam a realização das atividades de engenharia de software. Os modelos de infra-estrutura permitem a clara identificação destes elementos. Todos os métodos possuem esta dimensão, alguns com mais formalismo outros com menos. Esta categoria irá identificar as estruturais formais dos métodos.

Suporte aos objetivos de negócio: Esta categoria indica se o método possui alguma ferramenta que relacione os esforços de melhoria aos objetivos de negócio.

4.4 Categoria Resultados

Esta categoria trata das características que descrevem os resultados da aplicação de um método de melhoria e de seus mecanismos de validação.

Objetivo: O objetivo do método é uma descrição do que se pode obter ao utilizá-lo. Essa classificação difere-se do item foco da melhoria ao tratar somente os resultados e não os meios para alcançá-los.

Validação: Após identificar uma oportunidade e trabalhar para alcançá-la é fundamental verificar se a iniciativa obteve êxito e se os benefícios concretizados. Alguns métodos possuem mecanismos de medição embutidos e fases específicas de verificação.

A tabela 8 apresenta as características agrupadas por suas categorias:

Tabela 8: Taxonomia dos métodos de melhoria PDCA, QIP e IDEALSM adaptado de [27]

Categoria	Característica	PDCA	QIP	IDEALSM
Geral	Origem geográfica e difusão	USA/Mundo	USA/Mundo	USA/Mundo
	Organização patrocinadora	Não há	Universidade Maryland/SEL	SEI
	Origem científica	Controle estatístico	Parcialmente controle estatístico PDCA-	-
	Evolução/Estabilidade	Desde 1930	Desde 1984	Desde 1996
	Popularidade	Alta	Média	Alta e crescente
	Restrito a software	Não	Sim	Sim
Processo	Iniciação da melhoria	Top-down	Iterativa bottom-up	Top-down
	Foco da melhoria	Gerência de processos	Reuso de experiências	Gerência de processos
	Identificação de oportunidades	Não	Sim	Sim
	Modelo de ciclo de vida	Ação de melhoria	Ação de melhoria	Programa de melhoria
	Modelo de melhoria contínua	-	EF/QIP/GQM	CMMI
Organizacional	Tamanho da organização	Qualquer	Qualquer	Grande a Média
	Infra-estrutura de engenharia de processos de software	-	Experience Factory	EC ² ; SPIAC ³ ; MSG ⁴ ; SEPG ⁵ ; TWG ⁶
	Suporte aos objetivos de negócio	Não	Sim	Sim
Resultados	Objetivo	Apoiar melhoria contínua	Apoiar melhoria contínua; Engenharia de processo de software; Adoção de novas tecnologias	Apoiar e implantar melhoria contínua; Engenharia de processo de software; Adoção de novas tecnologias
	Validação	-	GQM	

² Executive Council: Conselho Executivo

³ Process Improvement Advisory Council: Conselho de melhoria de processos

⁴ Managment Steering Group: Grupo de gerência estratégica

⁵ Software Process Enginneering Group: Grupo de Engenharia de Processos de software

⁶ Technical Working Group: Grupo Técnico de Trabalho

5. CONCLUSÃO

5.1 Contribuições do Trabalho

Este trabalho não tem a pretensão de esgotar a discussão a respeito dos métodos de melhoria. Pretende sim, apontar os benefícios de sua utilização em projetos de melhoria de software e ser mais uma fonte para auxiliar àqueles envolvidos no planejamento e execução de tais programas.

Este trabalho ao propor uma taxonomia para métodos de melhoria de processos de software permite aos interessados analisar aspectos importantes dos métodos, como por exemplo, se estão relacionados a um modelo de melhoria contínua, seus objetivos, estratégia (ver tabela 8).

O PDCA, pode ser utilizado em qualquer ação de melhoria. Já o QIP e o IDEALSM são melhores aproveitados se utilizados dentro do modelo de melhoria a que se referem: Experience Factory e CMMI respectivamente.

Outro aspecto fundamental para a análise da aplicabilidade de um método é seu modelo de ciclo de vida. Como sabemos um projeto de melhoria de processo de software é um esforço contínuo, duradouro e portanto de longa duração. Logo, se desejamos selecionar um método para suportar a implantação de um modelo de maturidade e capacidade devemos preferencialmente escolher aquele que forneça suporte ao ciclo de vida de um programa de melhoria.

5.1.1 Trabalhos Futuros

Esta monografia pode ser completada através dos seguintes trabalhos:

- Adicionar mais métodos de melhoria à comparação
- Ampliar e aprofundar a taxonomia proposta.
- Realizar pesquisa de campo para identificar os métodos utilizados nas empresas brasileiras

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SEI Annual Report FY 2004. Pittsburgh: Software Engineering Institute, 2004- . Anual. Disponível em: < <http://www.sei.cmu.edu/annual-report/>>. Acesso em: agosto 2005.
- [2] Naur, P. Randell B. Software Engineering. Rept. NATO Sci. Comm., 1969
- [3] Pressman, Roger S.. Software Engineering: A Practitioners's Approach. 5th.ed. New York: McGraw-Hill, 2001. (McGraw-Hill series in computer science). ISBN 0-07-365578-3
- [4] Chrissis, Mary Beth; Konrad, Mike, Shrum, Sandy. CMMI®: Guidelines for Process Integration and Product Improvement. 1 ed. SEI: Addison Wesley Professional, 2003. (The SEI Series in Software Engineering). ISBN 0-32-115496-7
- [5] Shewhart, Walter A. Economic Control of Quality of Manufactured Product. New York: Van Nostrand, 1931.
- [6] Deming, W. Edwards. Out of the Crisis. Cambridge, MA: MIT Center for Advanced Engineering, 1986.
- [7] Juran, J. M. Juran on Planning for Quality. New York, New York: MacMillan, 1988.
- [8] Halvorsen, Christian Printzell; Conradi, Reidar. A Taxonomy to Compare SPI Frameworks, Lecture Notes In Computer Science; Vol. 2077, p. 217 – 235, 2001 Proceedings of the 8th European Workshop on Software Process Technology
- [9] SEI. Software Engineering Institute. CMMI for Systems Engineering/Software Engineering (CMMI-SE/SW), Staged Representation, Version 1.1, Technical report CMU/SEI-2002-TR-02. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002
- [10] ISO 9001. Quality Systems. Mode for quality assurance in: design, development, production, installation and servicing. International Standard Organization, 1994.
- [11] ISO/IEC 15504. International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 15504 Software Process Improvement and Capability dEtermination Model (SPICE), 2003.
- [12] Hefner, Rick; Tauser, Janice. Things They Never Taught You in CMM School. Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop. pp. 91-94, fev. 2001. INSPEC Accession Number 7239309
- [13] Andrade, Fabio Melo de. O método de melhorias PDCA. São Paulo, ago. 2003.

- [14] Basili, Victor R.; Caldiera, Gianluigi; Rombach, H. Dieter. Experience Factory. In John J. Marciniak, Encyclopedia of Software Engineering. , Volume 1, pp: 469-476. John Wiley & Sons, 1994
- [15] McFeeley, Bob. IDEALSM: A User's Guide for Software Process Improvement. Technical Note CMU/SEI-2005-TN-004. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, Fev. 1996
- [16] Keefer , Gerold; Lubecka, Hanna. The CMMI in 45 Minutes. Dezembro de 2001. Disponível em:
<<http://www.stickyminds.com/sitewide.asp?ObjectId=3243&Function=DETAILBROWSE&ObjectType=ART>>. Acesso em: janeiro 2006
- [17] SEI. Software Engineering Institute. CMMI for Systems Engineering/Software Engineering (CMMI-SE/SW), Continuous Representation, Version 1.1, Technical report CMU/SEI-2002-TR-02. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002
- [18] Anacleto , Alessandra; von Wangenheim, Christiane Gresse ; Salviano, Clenio F.; Savil , Rafael Savi. Uma Visão Geral da Futura Norma ISO/IEC 15504 para Avaliação de Processos de Software.
- [19] ISO/IEC TR 15504-2. International Organization for Standardization and International Electrotechnical Commission. A reference model for processes and process capability, 1998.
- [20] Kinnula, Atte. Software Process Engineering Systems: Model and Industry Cases. Oulu University Press, 2001. ISBN 951-42-6508-4. Disponível em:
<<http://herkules.oulu.fi/isbn9514265084/>>. Acesso em: outubro de 2005.
- [21] PDCA CYCLE. Disponível em
<<http://www.hci.com.au/hcsite3/toolkit/pdcacycl.htm>> Acesso em: outubro de 2005.
- [22] Basili, Victor. A Quantitative Approach to Software Management and Engineering. Institute for Advanced Computer Studies and Department of Computer Science , University of Maryland. CMSC 735, Class Notes. 1994. Disponível em:
< <http://www.cs.umd.edu/~basili/papers.html> >. Acesso em: novembro 2005
- [23] Basili, Victor; Zelkowitz, Marvin; McGarry, Frank; Page, Jerry; Waligora; Sharon; Pajerski, Rose. Special Report: SEL's Software Process-Improvement Program, IEEE Software, Volume 12, Number 6, pp 83-87, November 1995. Disponível em:
< <http://www.cs.umd.edu/~basili/papers.html> >. Acesso em: novembro 2005
- [24] Instituto Fraunhofer. Disponível em <<http://www.iese.fhg.de/>> Acesso em: novembro de 2005.
- [25] Kimbrough, Tom; Levine, Linda. The IDEALSM Transition Framework – Speeding Managed Change. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University. SEI Symposium, 1997.

[26] Gembra, Jennifer; Myers Chuck. The IDEALSM Model: A Practical Guide for Improvement. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, Bridge, issue three, 1997.

[27] Halvorsen, Christian Printzell; Conradi, Reidar. A Taxonomy to Compare SPI Frameworksnt. Proceedings of the 8th European Workshop on Software Process Technology. Source Lecture Notes In Computer Science; Vol. 2077, Pages: 217 – 235, 2001

7. BIBLIOGRAFIA COMPLEMENTAR

Humphrey, Watts S. Managing the Software Process. SEI: Addison Wesley Professional, 1989. (The SEI Series in Software Engineering). ISBN 0-201-18095-2

Rocha, Ana Regina Cavalcanti da; Maldonado, José Carlos; Weber, Kival Chaves. Qualidade de Software: Teoria e Prática. 1 ed. São Paulo: Prentice Hall, 2001. ISBN 85-87918-54-0

Vause, Jordan. Painting a Building: A Conceptual Reconstruction Based upon the IDEALSM Model. Disponível em: <<http://www.sei.cmu.edu/ideal/ideal.paint.html>>. Acesso em: agosto 2005.

Christiane Differding, Barbara Hoisl, Christopher Lott. Technology Package for the Goal Question Metric Paradigm. Internal Report Nr. 281/96, Department of Computer Science, University of Kaiserslautern, Germany, April 1996. Disponível em: <<http://www.agse.informatik.uni-kl.de/pubs/repository/differding96a/gqm-96.pdf>>. Acesso em: agosto 2005.

McFeeley, Robert S.; McKeehan, David W.; & Temple, Timothy. Software Process Improvement Roadmap. (SEI-94-SR-2)

R. Basili, "Quantitative Evaluation of Software Engineering Methodology," Proceedings of the First Pan Pacific Computer Conference, Melbourne, Australia, September 1985.

Mark C. Paulk, Analyzing the Conceptual Relationship Between ISO/IEC 15504 (Software Process Assessment) and the Capability Maturity Model for Software. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 1999.

Basili, Victor R.; Caldiera, Gianluigi; Rombach, H. Dieter. The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. , Volume 1, pp. 528-532, edited by John J. Marciniak, John Wiley & Sons, 1994, pp. 528-532. Disponível em: <>. Acesso em: agosto 2005