Qualidade, Métricas e Swebok

A T2Ti

Apresentação

A T2Ti nasce do sonho de três colegas que trabalhavam no maior banco da américa latina.

Tudo começa em 2007 com o lançamento do curso Java Starter. Logo depois veio o Siscom Java Desktop seguido de outros treinamentos.

Desde então a Equipe T2Ti se esforça para produzir material de qualidade que possa formar profissionais para o mercado, ensinando como desenvolver sistemas de pequeno, médio e grande porte.

Um dos maiores sucessos da Equipe T2Ti foi o Projeto T2Ti ERP que reuniu milhares de profissionais num treinamento dinâmico onde o participante aprendia na prática como desenvolver um ERP desde o levantamento de requisitos. Foi através desse treinamento que centenas de desenvolvedores iniciaram seu negócio próprio e/ou entraram no mercado de trabalho.

Em 2010 a T2Ti lança sua primeira aplicação para produção, o Controle Financeiro Pessoal. O sucesso foi tanto que saiu até em matéria no site Exame, ficando entre os 10 aplicativos mais baixados da semana.

Começa então a era de desenvolvimento de sistemas para alguns clientes exclusivos, pois o foco ainda era em desenvolvimento de treinamentos. A T2Ti desenvolve sistemas para o mercado nacional e internacional.

Atualmente a T2Ti se concentra nas duas vertentes: desenvolver sistemas e produzir treinamentos.

_____

Sumário

Qualidade, Métricas e Swebok

Qualidade de Software
Introdução.

CMMI
Histórico; Organização e Componentes do Modelo | Representação por Estágio | Representação Contínua | Contínua versus Estágio; Componentes; Metas e Práticas Genéricas | MG1 a MG5; Considerações.

MPS.Br
Introdução; Motivação; Visão Geral.

SWEBOK
Introdução; Áreas de Conhecimento.

Métricas de Software
Introdução; Conceitos; Método GQM; Normas ISO; Métricas de Produto de Software; Métricas de Processo de Software.

Qualidade de Software

Introdução

A qualidade de software é uma área de conhecimento da engenharia de software que objetiva garantir a qualidade do software através da definição e normatização de processos de desenvolvimento.

Apesar dos modelos aplicados na garantia da qualidade de software atuarem principalmente no processo, o principal objetivo é garantir um produto final que satisfaça às expectativas do cliente, dentro daquilo que foi acordado inicialmente.

Segundo a norma ISO 9000 (versão 2000), a qualidade é o grau em que um conjunto de características inerentes a um produto, processo ou sistema cumpre os requisitos inicialmente estipulados para estes.

No desenvolvimento de software, a qualidade do produto está diretamente relacionada à qualidade do processo de desenvolvimento.

Rodney Brooks, diretor do Laboratório de Inteligência Artificial e Ciência da Computação do MIT, define qualidade como a conformidade aos requisitos.

Essa definição exige determinar dois pontos: I) o que se entende por conformidade; e II) como são especificados - e por quem - os requisitos.

A comunidade de software tem se dedicado em encontrar maneiras de aumentar a qualidade do produto: Swebok, CMM, MPS.BR e outras iniciativas tem sido criadas e utilizadas nesse sentido.

Estudaremos os conceitos de tais iniciativas para compreendermos os modelos.

Histórico

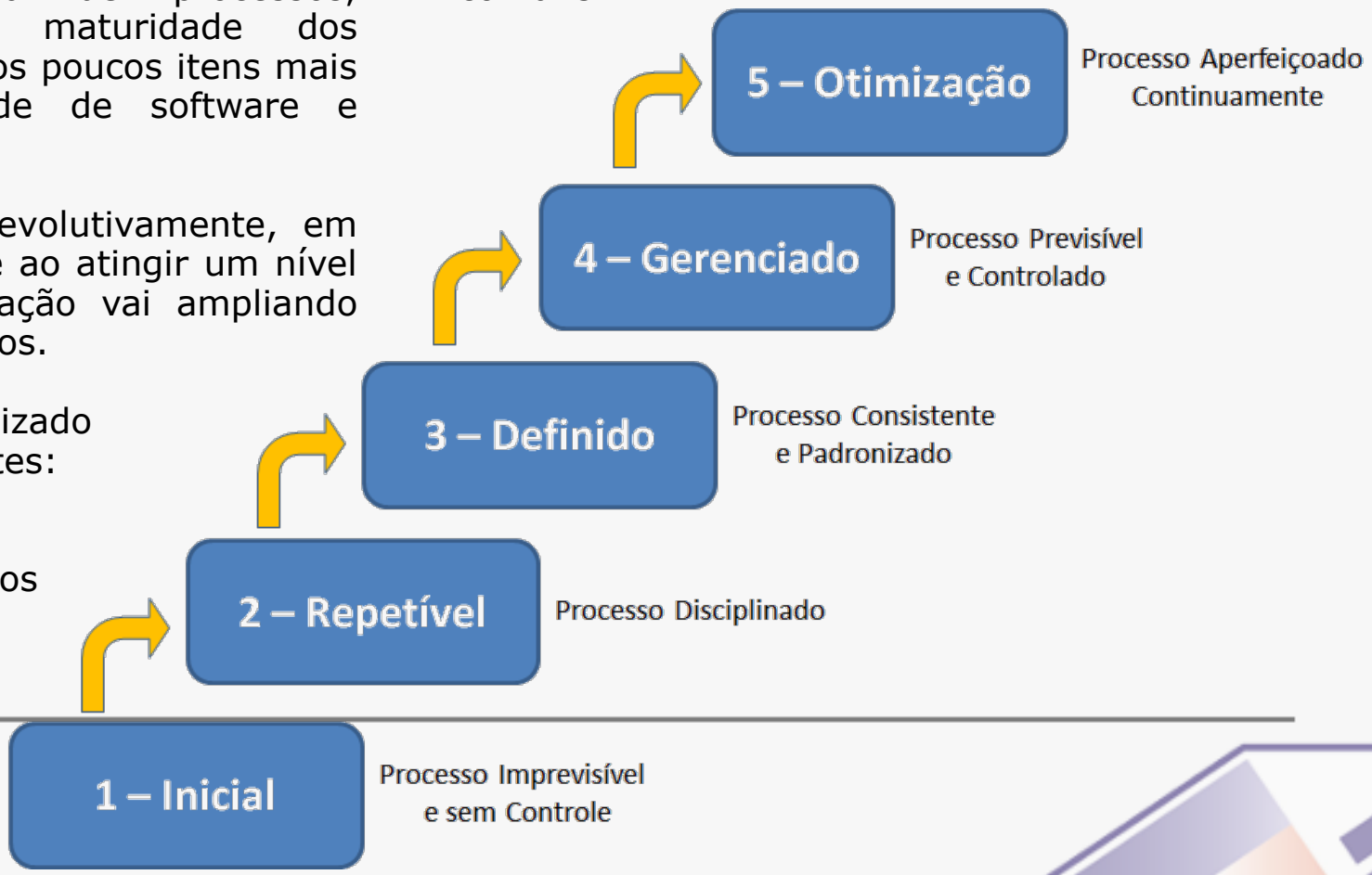
Segundo Paulk et al (1993), o modelo de capacidade de maturidade para software (CMM) "foi desenvolvido para guiar as organizações de software a selecionar estratégias de melhoria de processos, determinando a atual maturidade dos processos e identificando os poucos itens mais críticos para a qualidade de software e melhoria de processos".

Este modelo é disposto evolutivamente, em cinco níveis, da forma que ao atingir um nível de maturidade a organização vai ampliando sua capacidade de processos.

Internamente este é organizado em 4 principais componentes:

- níveis de maturidade
- áreas chaves de processos
- características comuns
- práticas chaves

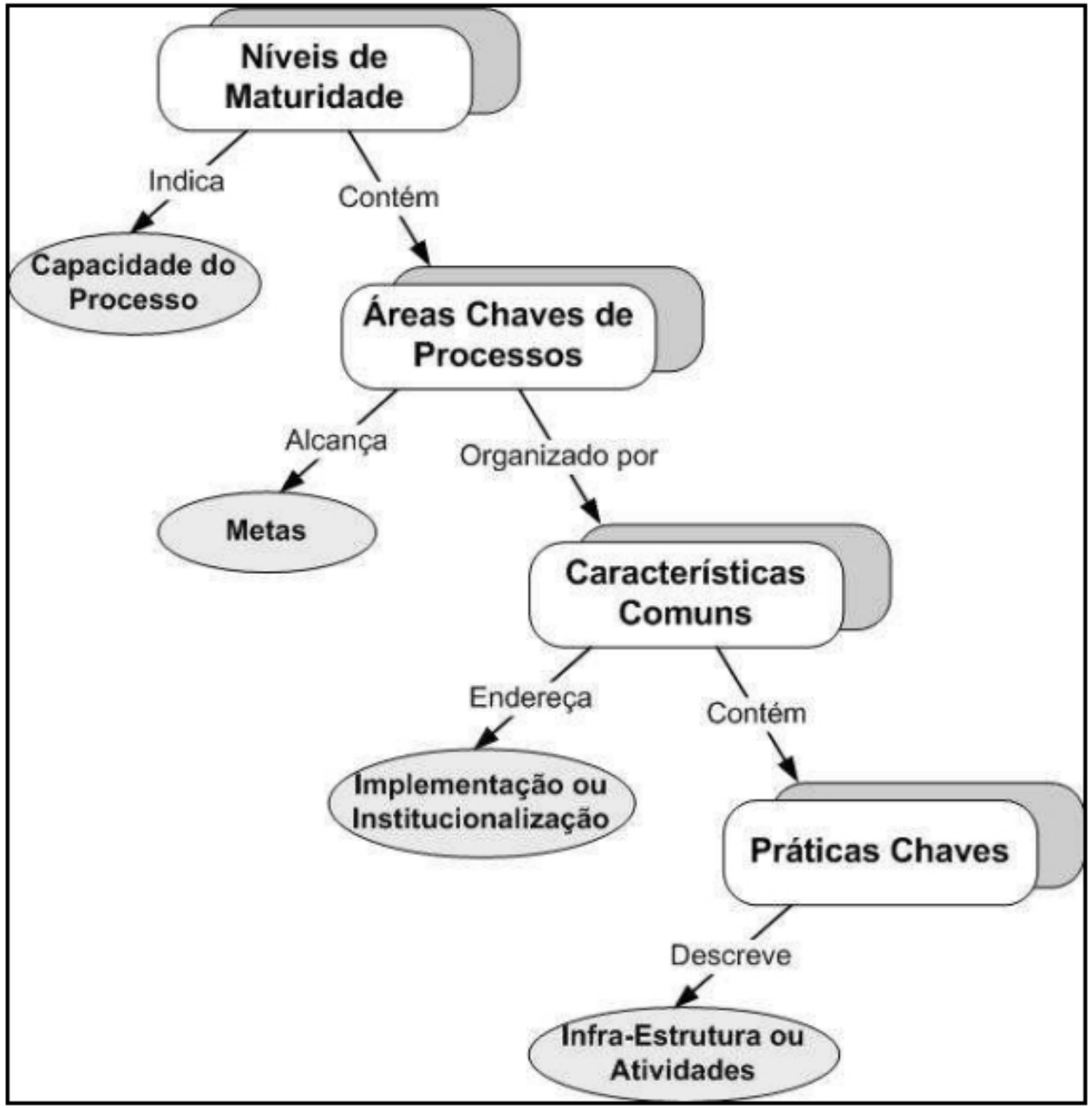
Cada nível de maturidade, com exceção do inicial, contém áreas chaves de processos (KPA – Key Process Areas) que são dirigidas por metas e organizadas por características comuns.



Histórico

Estas por sua vez procuram institucionalizar os processos e são formadas pelas práticas chaves, que estão no final da cadeia descrevendo as atividades.

Desta forma, ao todo, o modelo é composto de 5 níveis de maturidade, 18 áreas chaves de processo, 20 características comuns (5 para cada KPA) e 316 práticas chaves.



CMMI

Histórico

Com o grande sucesso do modelo CMM e o surgimento de novos modelos de melhoria como os de aquisição de software (SA-CMM – Software Acquisition Capability Maturity Model), gerenciamento de força de trabalho (P-CMM – People Capability Maturity Model) e desenvolvimento de produto e processo de software integrados (IPD-CMM - Integrated Product Development Capability Maturity Model), surgiu a necessidade de se agrupar alguns destes para facilitar a aplicação nas organizações que utilizavam mais de um modelo.

Por conta disso, o CMMI (Capability Maturity Model Integration), foi definido como um conjunto de boas práticas, que pretende prover uma visão integrada da maturidade de processos de desenvolvimento.

Inicialmente, o CMMI teria como objetivo integrar em um único modelo três outros: o CMM-SW 2.0, o EIA/IS 731 e o IPD-CMM 0.98. No entanto, durante o seu desenvolvimento, foi também previsto que o modelo fosse consistente e compatível com a norma ISO/IEC 15504.

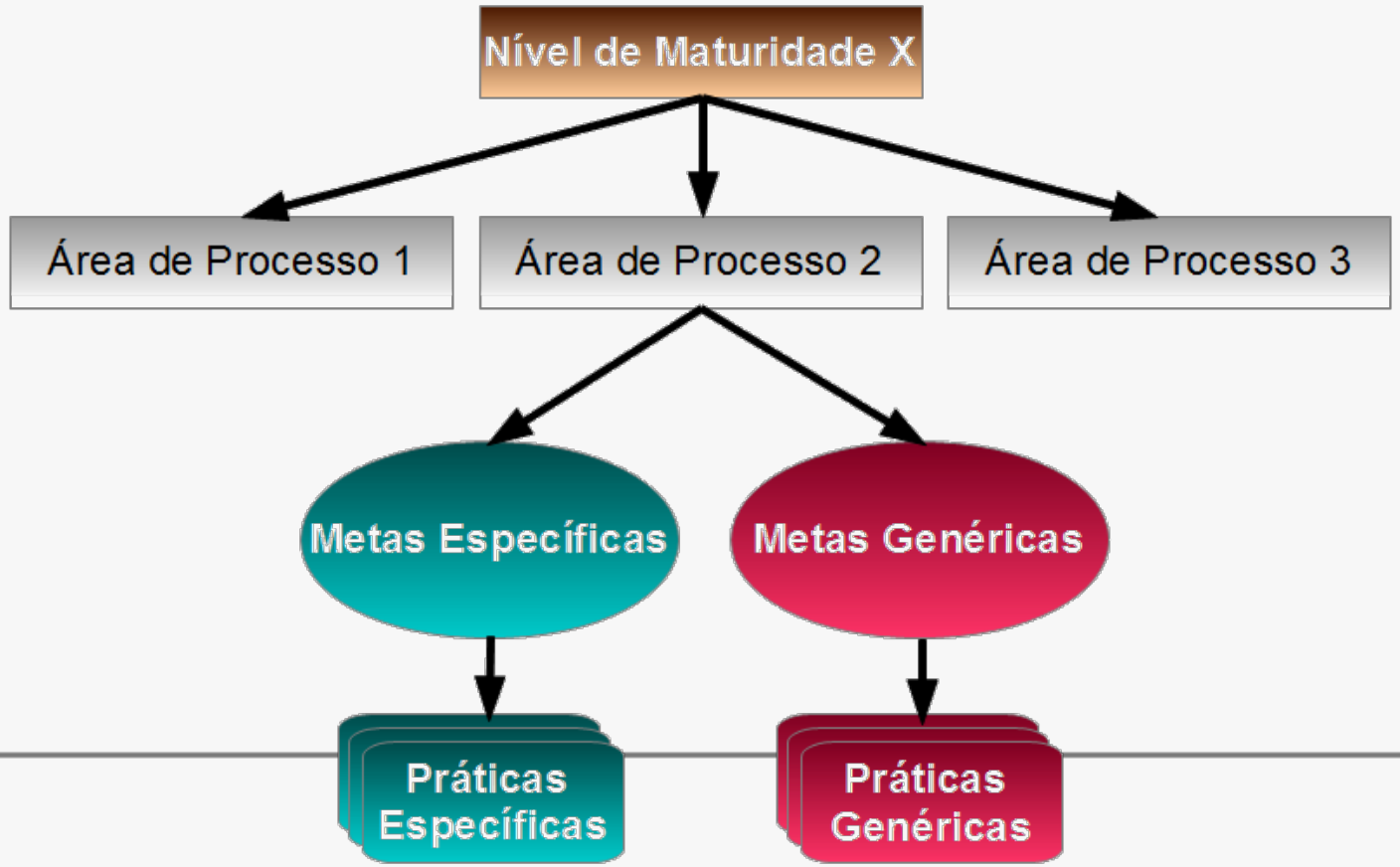
Organização e Componentes do Modelo

Estruturalmente, o CMMI é composto de níveis de maturidade ou de capacidade, áreas de processo, metas genéricas e específicas, práticas genéricas e específicas. Esses componentes são organizados em duas representações:

- Estágio
- Contínua

Organização e Componentes do Modelo | Representação por Estágio

A representação por estágio é uma abordagem que usa um conjunto predefinido de áreas de processo para especificar um guia de melhoria para a organização. Na representação por estágio as áreas estão organizadas em níveis de maturidade. Os níveis estão classificados em cinco valores de 01 (um) a 05 (cinco).



Organização e Componentes do Modelo | Representação por Estágio

Nível 1 – Inicial: Os processos são usualmente "ad hoc" e caóticos. Os sucessos geralmente dependem da competência e "heroísmos" das pessoas. Frequentemente cronograma e orçamento não são cumpridos.

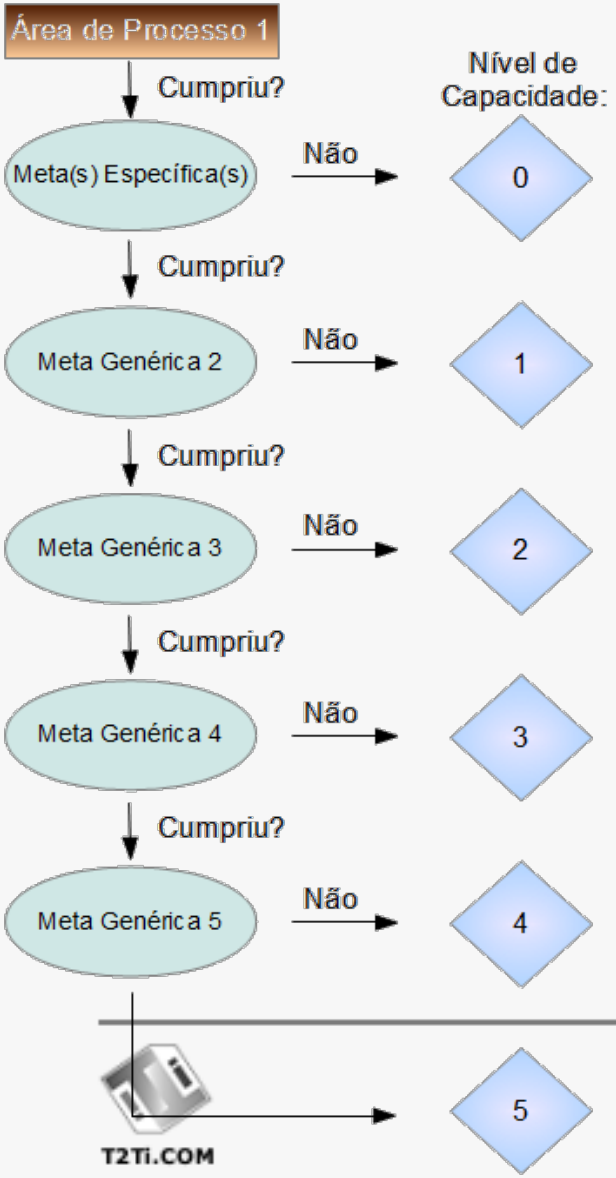
Nível 2 – Gerenciado: Os processos são executados de acordo com uma política; os projetos utilizam pessoas com a capacidade necessária para produzir saídas controladas; os stakeholders necessários são envolvidos; os processos são monitorados, controlados e revisados e sua aderência verificada em relação a descrição existente.

Nível 3 – Definido: Os processos estão bem caracterizados e definidos, e estão descritos em padrões, procedimentos, ferramentas e métodos. O conjunto de processos padrões da organização estão definidos e são evoluídos durante o tempo.

Nível 4 – Quantitativamente Gerenciado: A organização e os projetos estabelecem objetivos quantitativos para a qualidade e desempenho dos processos, e utiliza estes objetivos como critério de gerenciamento.

Nível 5 – Otimizado: Melhorias contínuas são realizadas nos processos da organização, motivadas pelas análises quantitativas, evoluções tecnológicas e objetivos de negócio.

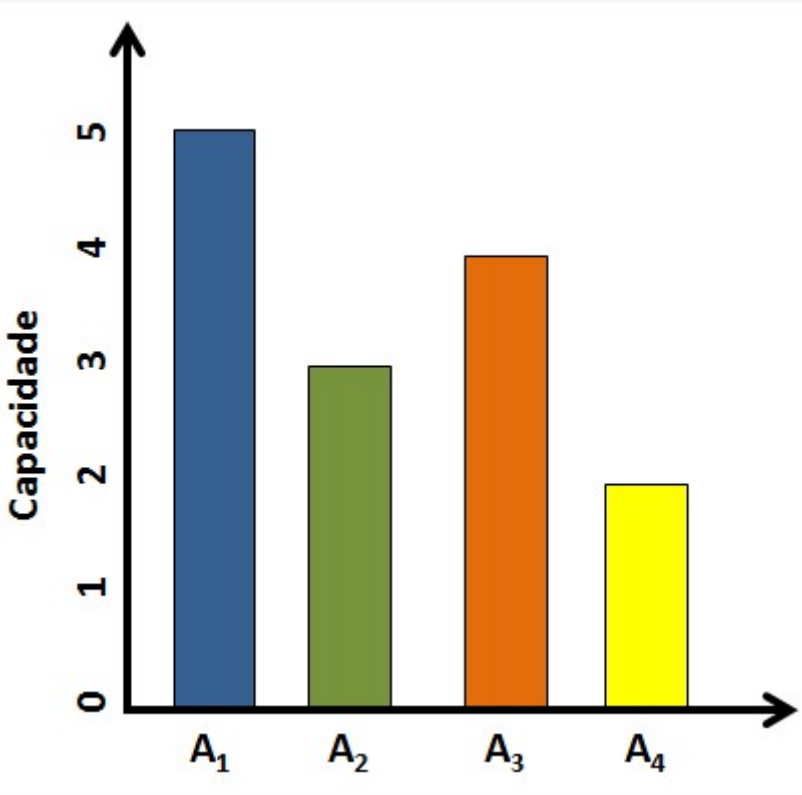
Organização e Componentes do Modelo | Representação Contínua



A representação contínua é indicada para organizações que desejam realizar melhorias relativas a uma área de processo específica. Nesta representação os níveis são denominados como níveis de capacidade e estão relacionados às práticas específicas e genéricas que, por sua vez, estão associadas a uma área de processo. Nesta forma, cada Área de Processo (A1, A2, A3, etc.) é avaliada individualmente de acordo com uma determinada estrutura de Metas Genéricas - MG e Metas Específicas – ME.

Só é possível passar de nível, caso todas metas sejam atendidas ou cumpridas.

Existem seis níveis de capacidade indo do 0 (zero) ao 05 (cinco).



Organização e Componentes do Modelo | Representação por Estágio

Nível 0 – Incompleto: O processo não é executado ou é só parcialmente executado. As metas específicas não são alcançadas e não existem metas genéricas neste nível. Por estes motivos, não existe razão para institucionalizar o processo.

Nível 1 – Executado: O processo é executado completamente e já satisfaz as metas específicas da área de processo. Contudo o processo ainda não está institucionalizado na organização.

Nível 2 – Gerenciado: Neste nível o processo é executado conforme políticas, os recursos possuem a habilidade necessária para produzir "saídas" controladas, este é monitorado, controlado, revisado, e avaliado em relação a sua aderência ao processo descrito. Agora o processo já é mantido durante o tempo na organização.

Nível 3 – Definido: O processo é adaptado a partir de um conjunto padrão de processos da organização e definido conforme guia de adaptação da organização, e contribui com produtos de trabalho, medições e outras informações de melhoria de processo para os recursos da organização.

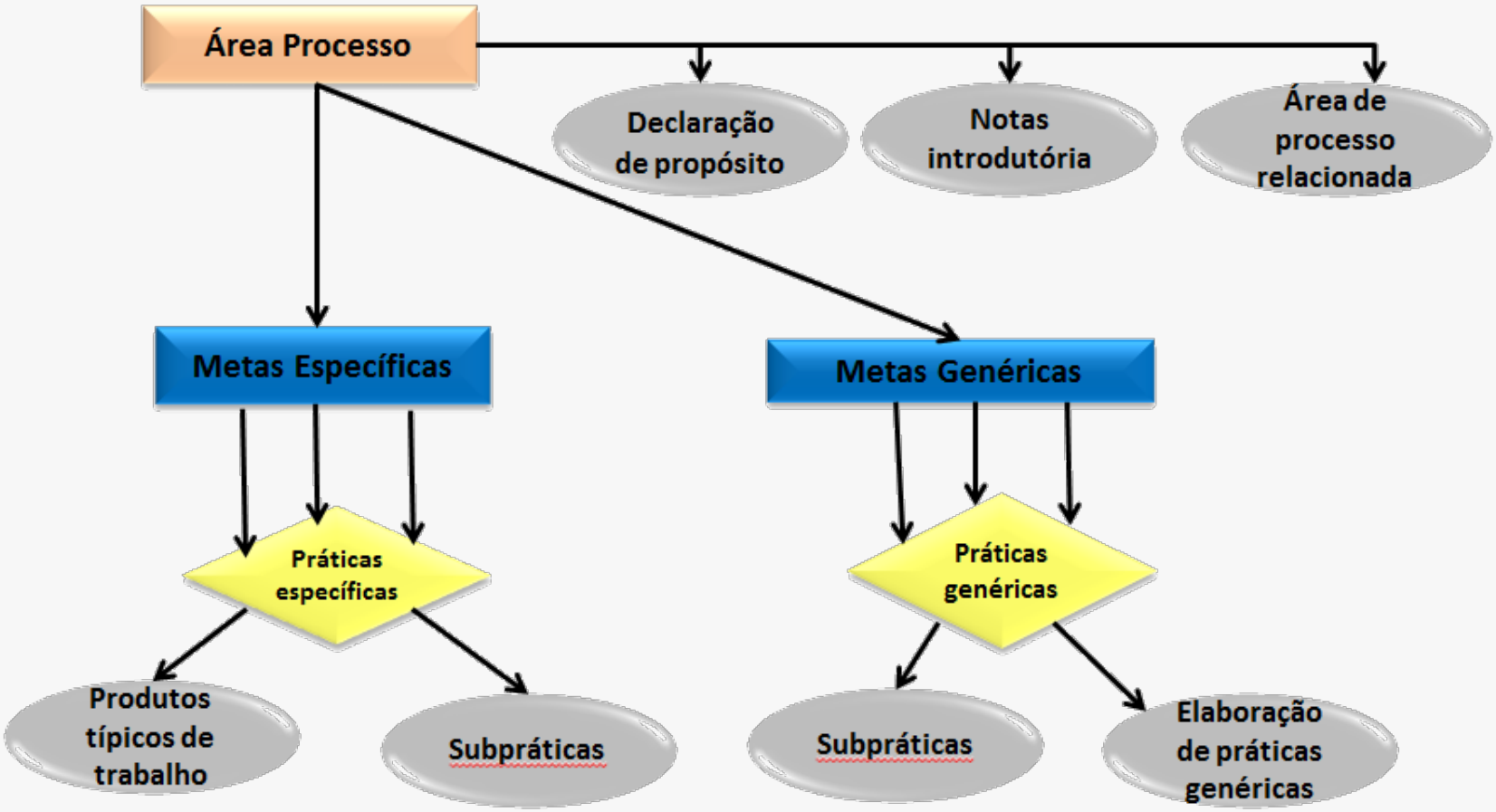
Nível 4 – Quantitativamente Gerenciado: O processo é controlado utilizando estatísticas e outras técnicas quantitativas. Objetivos quantitativos para qualidade e desempenho dos processos são estabelecidos e utilizados como critério no gerenciamento.

Nível 5 – Otimizado: O processo é continuamente evoluído através de melhorias incrementais e inovações.

Organização e Componentes do Modelo | Contínua versus Estágio

Representação Estágio	Representação Contínua
Disponibiliza a organização um caminho pré-definido (comprovado) de melhoria	Liberdade de escolher a ordem das melhorias que melhor se adequam aos objetivos de negócio e mitigam as áreas de risco
Foca em um conjunto de processos que fornece a organização uma capacidade específica, caracterizada por um nível de maturidade	Permite um aumento da visibilidade da capacidade alcançada em cada área de processo
Sumariza a melhoria de processo de uma forma simples – Nível de maturidade	Permite melhorias de diferentes processos em diferentes níveis
Baseado em uma relativa longa história de uso, que incluem casos de uso e dados que demonstram o retorno do investimento	Reflete uma nova abordagem que ainda não possui dados para demonstrar o retorno do investimento

Componentes



Componentes

O CMMI possui como componentes de seu modelo diversos itens que formam uma área de processo. Conforme pode ser observado na imagem da página anterior, estes componentes podem ser obrigatórios, esperados ou apenas de caráter informativo.

Área de Processo

Uma área de processo e um conjunto de práticas relacionadas a um determinado assunto, que quando executadas em conjunto, satisfazem as metas consideradas importantes para aquela área.

Ex.: Gerenciamento de Requisitos, Gerenciamento de Configuração, et.

Metas Específicas e Genéricas

As metas no CMMI são objetivos definidos para o desenvolvimento e avaliação das práticas específicas e genéricas de uma área de processo. Todas as práticas obrigatoriamente são executadas com o objetivo de se atingir uma meta.

Ex.: Meta Genérica 2 – O processo está gerenciado e está institucionalizado, Meta Específica 1 (Gerenciamento de Configuração) – Estabelecer Baselines.

Componentes

Práticas Específicas e Genéricas

Uma prática é a descrição de uma atividade considerada importante em um processo para se atingir determinada meta específica ou genérica. Conforme já exposto anteriormente, o modelo CMMI não detalha como estas atividades devem ser executadas, ficando a cargo da organização escolher a forma que lhe parecer melhor.

Ex.: Prática Genérica 2.1 – Estabelecer uma política organizacional, Prática Específica 1.1 (Gerenciamento de Configuração) – Identificar os itens de configuração.

Subpráticas

Componente informativo do modelo que auxilia a implementação das práticas específicas e genéricas. Ou seja, é um nível de detalhe a mais em algumas práticas para que o quê está sendo solicitado fique mais claro.

Ex.: Subprática Genérica 2.2.1 – Definir e documentar o plano para executar o processo; (Prática: Planejar o Processo), Subprática Específica 1 (Gerenciamento de Configuração) – Selecionar os itens de configuração e produtos de trabalho baseado em um critério documentado. (Prática: Identificar itens de configuração).

Metas e Práticas Genéricas

Para fornecer uma visão de como as coisas funcionam, vamos abordar as metas e práticas genéricas. Elas são aplicadas a todas as áreas de processo e têm por objetivo garantir a institucionalização do processo na organização. Isto é, procura garantir que há comprometimento e consistência na execução dos processos. As metas genéricas definem graus de institucionalização do processo, conforme apresentado na tabela abaixo.

Meta Genérica	Progresso dos Processos
MG 1 – Alcançar Metas Específicas	Processo executado
MG 2 – Institucionalizar processo gerenciado	Processo gerenciado
MG 3 – Institucionalizar o processo definido	Processo gerenciado
MG 4 – Institucionalizar processo quantitativamente gerenciado	Processo quantitativamente gerenciado
MG 5 – Institucionalizar processo otimizado	Processo otimizado

Metas e Práticas Genéricas | MG1

Esta meta prevê que o processo permita que sejam alcançadas metas específicas de uma área de processo, pela transformação de produtos de trabalho identificados de entrada em produtos de trabalho identificados de saída. Contém uma prática.

PG 1.1 – Executar Práticas Específicas

Executar a pratica específica de uma área de processo para desenvolver produtos de trabalho e prover serviços que atendam as metas específicas da área de processo.

Metas e Práticas Genéricas | MG2

Esta meta prevê que o processo está institucionalizado como um processo gerenciado dentro da organização. É a meta com o maior número de práticas (10), pois prevê o maior número de controles.

PG 2.1 - Estabelecer uma política organizacional

Estabelecer e manter uma política organizacional para o planejamento e execução do processo, isto é, definir expectativas organizacionais dos processos e fazer com que estas expectativas sejam visíveis a quem é afetado dentro da organização.

PG 2.2 - Planejar o processo

Estabelecer e manter um plano detalhado para a execução do processo, determinando o que é necessário para a sua execução e alcance dos objetivos estabelecidos. Este plano deve ser aprovado por todos os stakeholders relevantes ao processo.

PG 2.3 - Prover recursos

Prover recursos adequados para a execução do processo, desenvolvimento dos produtos de trabalho, e provimento de serviços do processo. Os recursos descritos aqui podem ser de diversos tipos como: financeiros, instalações físicas, pessoal qualificado, ferramentas, etc.

Metas e Práticas Genéricas | MG2

PG 2.4 - Atribuir responsabilidades

Atribuir responsabilidades e autoridades para a execução do processo, desenvolvimento de produtos de trabalho, e provimento de serviços do processo.

PG 2.5 - Treinar as pessoas

Treinar as pessoas para executar e suportar as necessidades do processo. Existem vários métodos de treinamento que podem servir para melhorar as habilidades para execução de um processo como estudo próprio, mentoring (acompanhamento) ou treinamento em sala de aula.

PG 2.6 - Gerenciar a configuração

Colocar os produtos de trabalho identificados do processo sob os níveis de controle apropriados. Os níveis de controle podem variar conforme a criticidade do produto, por exemplo, em alguns casos apenas um controle de versão pode ser suficiente e em outros pode ser necessário a criação de baselines, aprovações, etc.

Metas e Práticas Genéricas | MG2

PG 2.7 - Identificar e envolver os stakeholders relevantes

Identificar e envolver os stakeholders relevantes ao processo conforme planejado. Para as diversas etapas de um projeto determinadas pessoas devem ser envolvidas conforme a ação necessária, por exemplo, quem será responsável pelo planejamento, da coordenação, das avaliações, da resolução de problemas, entre outros.

PG 2.8 - Monitorar e controlar o processo

Monitorar e controlar o processo em relação ao plano e tomar as ações apropriadas de correção, caso necessário. Monitorar e controlar o processo envolve em medir os atributos corretos do processo ou produtos do processo.

PG 2.9 - Avaliar objetivamente a aderência ao processo

Avaliar objetivamente a aderência do processo em relação a sua descrição, padrões, e procedimentos, e endereçar suas não conformidades. Esta prática pode ser implementada em parte avaliando os produtos de trabalho de processo.

PG 2.10 - Revisar o status com a alta gerência

Revisar as atividades status, e resultados do processo com a alta gerência e resolver os problemas.

Metas e Práticas Genéricas | MG3

Esta meta prevê que a organização deve possuir processos padrões que são adaptados conforme a necessidade, a partir do guia de adaptação da organização. Os processos adaptados obrigatoriamente são gerenciados conforme as atividades previstas na MG 2.

A principal diferença entre um processo gerenciado (MG 2) e um processo definido (MG 3) é o escopo da aplicação das descrições dos processos, padrões e procedimentos. Para o processo gerenciado estes itens são aplicados a um projeto em particular, enquanto no definido eles valem para toda a organização. O nível de detalhamento e rigor na implementação de um processo definido também tende a ser maior, visto que a visibilidade é para a organização como um todo e não somente no escopo de um projeto. Essa meta possui duas práticas.

PG 3.1 – Estabelecer e manter a descrição de um processo definido.

Estabelecer e manter a descrição dos processos que são adaptados do conjunto de processos padrões da organização, que atendem a necessidade de um uso específico.

PG 3.2 – Coletar informações de melhoria.

Coletar produtos de trabalho, medições, resultados de medições, e informações de melhorias originárias do planejamento e execução do processo.

Metas e Práticas Genéricas | MG4

Esta meta prevê que um processo após estar definido, deve ser controlado utilizando técnicas e estatísticas ou quantitativas. A qualidade do produto, do serviço, e os atributos de desempenho do processo são medidos e controlados pelo projeto. Possui duas práticas.

PG 4.1 – Estabelecer objetivos quantitativos para o processo.

Estabelecer objetivos quantitativos para o processo que tenham como meta o desempenho e qualidade, baseado nas necessidades do cliente e objetivos de negócio.

PG 4.2 – Estabilizar o desempenho dos subprocessos

Estabilizar o desempenho de um ou mais processos para determinar a habilidade do processo pai em alcançar os objetivos de qualidade e desempenho definidos.

Metas e Práticas Genéricas | MG5

Esta meta prevê a mudança e adaptação de um processo quantitativamente gerenciado para atender os objetivos correntes de negócio. Um processo otimizado foca em melhorias contínuas incrementais ou inovações tecnológicas. Possui duas práticas.

PG 5.1 – Garantir a melhoria contínua do processo.

Selecionar e implantar sistematicamente melhorias de processo e tecnológicas que contribuem para atender os requisitos de qualidade e desempenho.

PG 5.2 – Corrigir a causa raiz dos problemas.

Analisar defeitos e outros problemas que são encontrados em um processo quantitativamente gerenciado, para corrigir a causa raiz destes, evitando assim, que ocorram novamente no futuro.

Considerações

Tivemos até agora uma visão geral do CMMI. É interessante que o leitor tire um tempo para se aprofundar no modelo, estudando o gerenciamento de processos, o gerenciamento de projetos, a engenharia e o suporte. Provavelmente você só verá o CMMI em grandes organizações.

Existe, porém, uma iniciativa brasileira para pequenas e médias empresas conhecida como MPS.Br.

Vejamos do que se trata.

MPS.Br

Introdução

O MPS.Br ou Melhoria de Processos do Software Brasileiro é simultaneamente um movimento para a melhoria da qualidade (Programa MPS.BR) e um modelo de qualidade de processo (Modelo MPS). Voltado para a realidade do mercado de pequenas e médias empresas de desenvolvimento de software no Brasil, ele é baseado nas normas ISO/IEC 12207 e ISO/IEC 15504 e compatível com o CMMI.

O projeto tem apoio do Ministério da Ciência e Tecnologia, da FINEP e do Banco Interamericano de Desenvolvimento. No Brasil o projeto é desenvolvido pela Softex, interagindo com as universidades e com o Governo Federal. Uma das principais vantagens do modelo é seu custo reduzido de certificação em relação as normas estrangeiras, sendo ideal para micro, pequenas e médias empresas que são a grande maioria no Brasil.

Um dos objetivos do projeto é replicar o modelo na América Latina, incluindo o Chile, Argentina, Costa Rica, Peru e Uruguai.

MPS.BR

Motivação

O Brasil é um país cujo desenvolvimento de produtos de software está entre os maiores do mundo. A cada dia aumenta o nível de exigência por parte dos clientes no que diz respeito à qualidade e complexidade dos produtos. A partir deste ponto, podemos observar que as empresas estão buscando cada vez mais a maturidade nos seus processos de software para atingir padronizações de qualidade e produtividade internacionais, que são essenciais para a sobrevivência no mercado de TI.

Porém, o custo de uma certificação para uma empresa pode ser de até US$ 400 mil, o que se torna inviável para empresas de micro, pequeno e médio porte. Então, em uma parceria entre a Softex, Governo e Universidades, surgiu o projeto MPS.Br (melhoria de processo de software brasileiro), que é a solução brasileira compatível com o modelo CMMI, está em conformidade com as normas ISO/IEC 12207 e 15504, além de ser adequado à realidade brasileira.

MPS.BR

Visão Geral

O MPS.Br é dividido em 3 partes: MR-MPS, MA-MPS, MN-MPS.

MR-MPS - Modelo de referência para melhoria do processo de software

O MPS.Br apresenta 7 níveis de maturidade (o que é um diferencial em relação aos outros padrões de processo) que são:

A - Em Otimização;
B - Gerenciado quantitativamente;
C - Definido;
D - Largamente Definido;
E - Parcialmente Definido;
F - Gerenciado;
G - Parcialmente Gerenciado.

Visão Geral

Cada nível de maturidade possui suas áreas de processo, onde são analisados:

Processos Fundamentais

- aquisição;
- gerência de requisitos;
- desenvolvimento de requisitos;
- solução técnica;
- integração do produto;
- instalação do produto;
- liberação do produto.

Visão Geral

Cada nível de maturidade possui suas áreas de processo, onde são analisados:

Processos Organizacionais

- gerência de projeto;
- adaptação do processo para gerência de projeto;
- análise de decisão e resolução;
- gerência de riscos;
- avaliação e melhoria do processo organizacional;
- definição do processo organizacional;
- desempenho do processo organizacional;
- gerência quantitativa do projeto;
- análise e resolução de causas;
- inovação e implantação na organização.

Visão Geral

Cada nível de maturidade possui suas áreas de processo, onde são analisados:

Processos de Apoio

- garantia de qualidade;
- gerência de configuração;
- validação;
- medição;
- verificação;
- treinamento.

Visão Geral

Em seguida vem a Capacidade, onde são obtidos os resultados dos processos analisados, onde cada nível de maturação possui um número definido de capacidades a serem vistos.

AP 1.1 - O processo é executado;
AP 2.1 - O processo é gerenciado;
AP 2.2 - Os produtos de trabalho do processo são gerenciados;
AP 3.1 - O processo é definido;
AP 3.2 - O processo está implementado;
AP 4.1 - O processo é medido;
AP 4.2 - O processo é controlado;
AP 5.1 - O processo é objeto de inovações;
AP 5.2 - O processo é otimizado continuamente.

MPS.BR

Visão Geral

MA-MPS - Método de avaliação para melhoria do processo de software

Tem como objetivo orientar a realização de avaliações, em conformidade com a norma ISO/IEC 15504, em empresas e organizações que implementaram o MR-MPS. Avaliação MA-MPS:

Equipe de avaliação: 3 a 8 pessoas, sendo:

- 1 avaliador líder
- no mínimo 1 avaliador adjunto
- no mínimo 1 técnico da empresa

Duração: 2 a 4 dias;
Validade: 3 anos;

MPS.BR

Visão Geral

MA-MPS - Método de avaliação para melhoria do processo de software

Estruturação da Avaliação:

- Planejar e preparar avaliação - Plano de Avaliação / Descrição dos indicadores de processo;
- Conduzir Avaliação - Resultado da avaliação;
- Relatar resultados - Relatório da avaliação;
- Registrar e publicar resultados - Banco de dados Softex.

MPS.BR

Visão Geral

MN-MPS - Modelo de negócio para melhoria do processo de software

Instituições que se propõem a implantar os processos MPS.Br (Instituições Implementadoras) podem se credenciar através de um documento onde é apresentada a instituição proponente, contendo seus dados com ênfase na experiência em processos de software, estratégia de implementação do modelo, estratégia para seleção e treinamento de consultores para implementação do MR.MPS, estratégia para seleção e treinamento de avaliadores, lista de consultores de implementação treinados no modelo e aprovados em prova específica, lista de avaliadores treinados no modelo e aprovados em prova específica. E como o leitor deve proceder para aprender tudo isso? A Softex realiza cursos para formação de consultores, compradores e avaliadores MPS.BR. São ao todo 4 cursos:

- Curso de Introdução - C1
- Curso de Implementação - C2
- Curso de Avaliação - C3
- Curso de Aquisição - C4

MPS.BR

Visão Geral

Periodicamente, são realizadas provas em nível nacional para certificar profissionais em cada um dos cursos descritos acima. Tanto os cursos e as provas são realizadas nos Agentes SOFTEX em cada Estado.

Para mais detalhes, acesse o site da Softex clicando na imagem abaixo.

Softex

TECNOLOGIA DA INFORMAÇÃO BRASILEIRA

SWEBOK

Introdução

O SWEBOK (Software Engineering Body of Knowledge) é um guia que pretende prover uma caracterização validada e consensual dos limites da disciplina da engenharia de software, fornecendo acesso ao corpo de conhecimento desta disciplina.

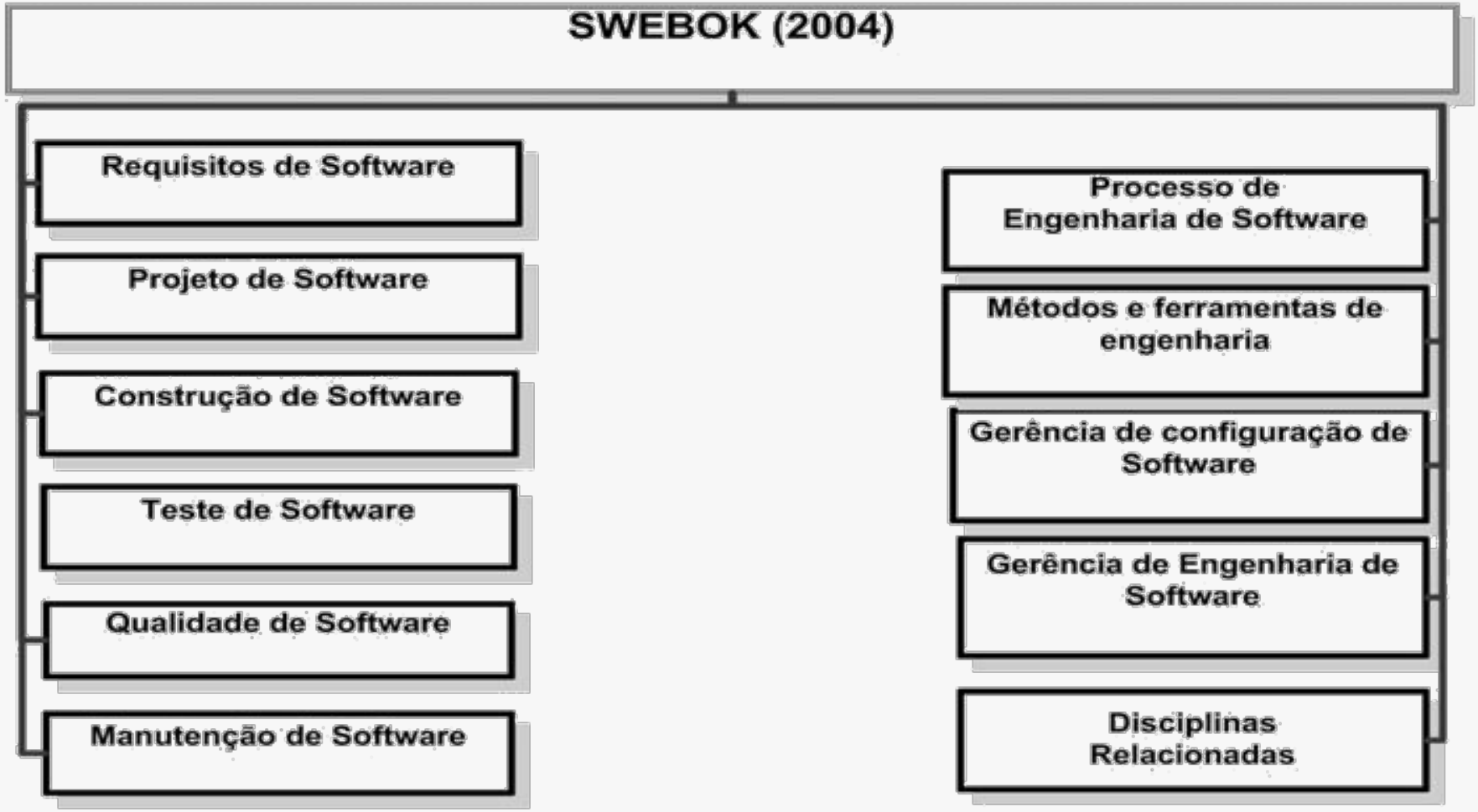
O SWEBOK possui cinco objetivos:

1. Promover uma visão consistente da engenharia de software mundialmente;
2. Clarear a localização e determinar os limites da engenharia de software com outras disciplinas como: ciência da computação, gerenciamento de projeto, engenharia da computação e matemática;
3. Caracterizar o conteúdo de uma disciplina de engenharia de software;
4. Prover um acesso no formato de lista do corpo de conhecimento em engenharia de software;
5. Prover os fundamentos para o desenvolvimento de um curriculum, ou certificações individuais ou material de licenciamento.

Este corpo de conhecimento está organizado no guia em áreas de conhecimento, sendo 10 relativas à engenharia de software e 1 de disciplinas correlatas.

SWEBOK

Introdução



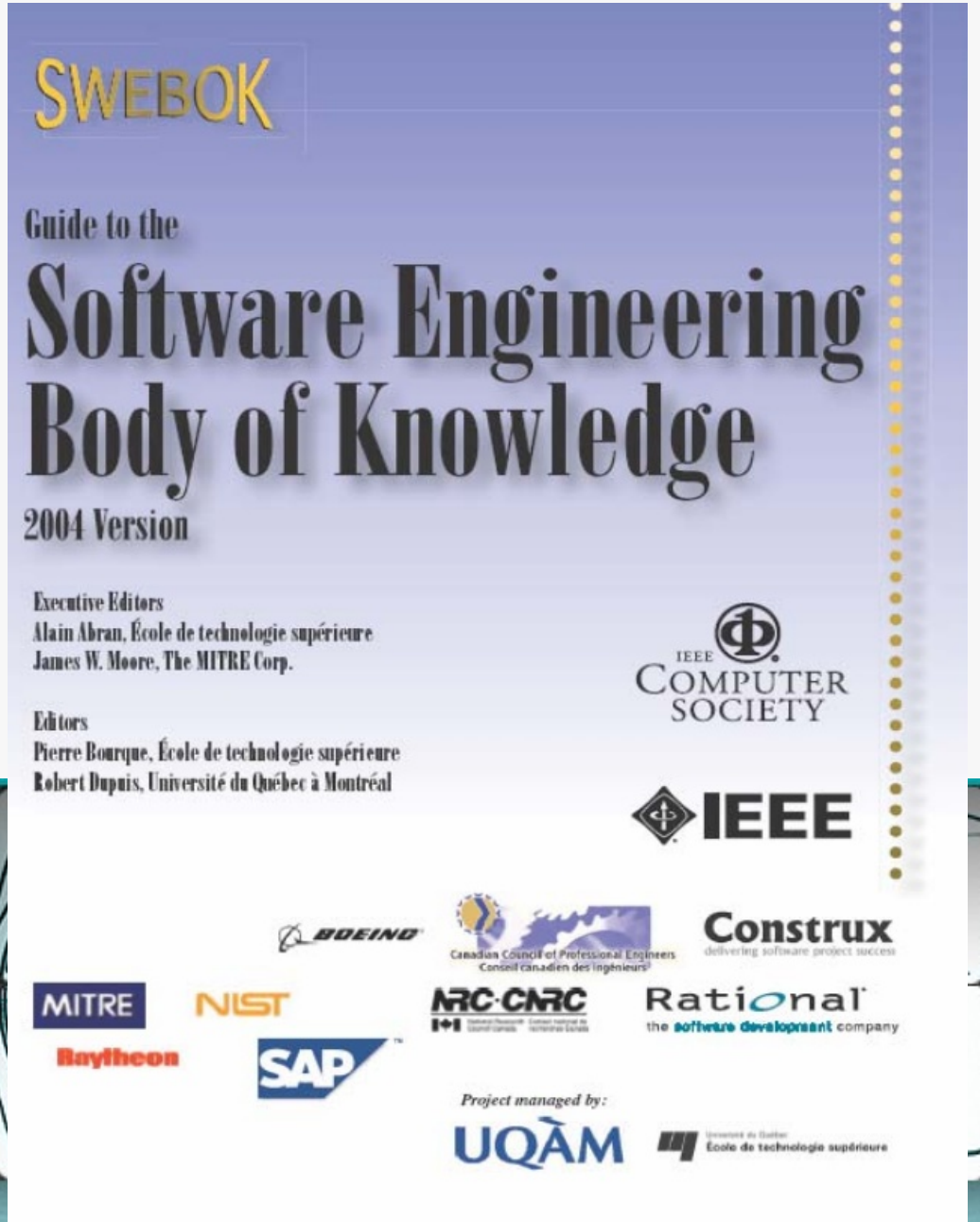
SWEBOK

Introdução

Por ser um guia genérico sobre as atividades e processos envolvidos na engenharia de software, o SWEBOK não traz nenhum conhecimento relacionado à:

- Linguagens de programação;
- Banco de dados;
- Rede de computadores.

Devido a estas limitações e a sua forte orientação acadêmica a utilização do guia é um pouco restrita, mas pode ser um bom começo na hora de procurar referências para implementar processos dentro de uma organização.



SWEBOK

Áreas de Conhecimento

Requisitos de Software

Segundo o SWEBOK (2004, p. 33), "Um requisito de software é uma propriedade que precisa ser exibida com o objetivo de resolver um problema do mundo real".

Assim, a disciplina que lida com este tipo de requisito tem por objetivo tratar a elicitação, analise, especificação e validação destes.

Para não fazer confusão com o próprio termo engenharia de software, os autores preferiram não utilizar no guia o termo "Engenharia de Requisitos".

A disciplina de requisitos de software está subdividida no SWEBOK em 7 tópicos.

SWEBOK

Áreas de Conhecimento

1 - Fundamentos

Inclui a definição dos requisitos de software e também seus principais tipos: produtos vs processos, funcional vs não funcional, propriedades emergentes. Também descreve a importância de se quantificar os requisitos e separá-los em requisitos de sistema e requisitos de software.

2 - Processo de Requisitos

Introduz o processo de requisitos, servindo de orientação para as demais subáreas e descreve como a engenharia de requisitos se relaciona com as demais áreas da engenharia de software. Descreve os modelos do processo, atores, o suporte e gerenciamento, e a qualidade e melhoria.

3 - Elicitação de Requisitos

Esta subárea se preocupa em identificar de onde os requisitos são coletados, e como estes podem ser coletados. Inclui fontes de requisitos e técnicas de elicitação.

Áreas de Conhecimento

4 - Análise de Requisitos

Está focada no processo de análise de requisitos para:

- Detectar e resolver os conflitos entre os requisitos.
- Descobrir os limites do software e como este deve interagir com o ambiente.
- Elaborar requisitos de sistema para os requisitos de software.

A análise de requisitos inclui a classificação, modelagem conceitual, projeto arquitetural, alocação e negociação.

5 - Especificação de Requisitos

Trata da produção de um documento (ou versão eletrônica equivalente), que possa ser sistematicamente revisado, avaliado e aprovado.

Áreas de Conhecimento

6 - Validação de Requisitos

Trata do processo de examinar os documentos de requisitos para garantir que eles estão definindo o sistema correto. Está subdividida em descrições dos requisitos, revisões, prototipação, e modelo de validação e teste de aceitação.

7 - Considerações Práticas

Apresenta tópicos que ajudam a entender o processo de requisitos como: a natureza interativa, o gerenciamento de mudança, manutenção dos requisitos, medições, entre outros.

SWEBOK

Áreas de Conhecimento

Projeto de Software

Segundo o SWEBOK (2004, p. 26), o projeto de software é "o processo de definição de arquitetura, componentes, interfaces e outras características de um sistema ou componente". Esta área de conhecimento foi organizada em 6 subáreas.

1 - Fundamento do projeto de software

Apresenta a base para o entendimento do papel e escopo do projeto de software. São conceitos gerais de software do projeto de software, processos e técnicas.

2 - Pontos chaves do projeto de software

Inclui concorrência controle e tratamento de eventos, distribuição de componentes, tratamento de erro e exceção, tolerância a falhas, interação e apresentação e persistência de dados.

SWEBOK

Áreas de Conhecimento

3 - Arquitetura e Estrutura de Software
Descreve quais são as estruturas arquiteturais, estilos, padrões de projeto, e famílias de programas e frameworks.

4 - Análise de Qualidade e Avaliação do Projeto de Software
Apresenta aspectos de qualidade (atributos, análise, avaliação e técnicas), no contexto do projeto de software.

5 - Notação de projeto de software
Descreve algumas notações e linguagem que ajudam a representar os artefatos de projeto de software. (Ex.: Diagramas de atividade, diagrama entidade relacionamento, diagramas de colaboração, etc.)

6 - Estratégia e métodos do projeto de software
Descreve estratégias gerais seguidas de projeto orientado a função, projeto orientado a objeto, projeto centrado em estrutura de dados; projeto baseado em componentes e outros.

SWEBOK

Áreas de Conhecimento

Construção de Software

A construção de software se refere a criação detalhada de um software funcional através da combinação de codificação, verificação, teste unitário, teste de integração e debugging (SWEBOK, 2004). Esta área de conhecimento foi dividida em três subáreas.

1 - Fundamentos da construção de software
Discute alguns dos princípios básicos da construção: minimizar complexidade, antecipar mudança e construção para verificação, além de alguns padrões para construção.

2 - Gerenciar construção
Descreve algumas formas de modelos, planejamento e medições da construção de software.

3 - Considerações práticas
Detalha alguns pontos relacionados à linguagens de programação, codificação, teste, reuso e integração.

Áreas de Conhecimento

Teste de Software

Consiste de verificação dinâmica do comportamento de um programa em um conjunto finito de casos de teste, selecionados de um conjunto infinito de execuções (SWEBOK, 2004). Esta área de conhecimento está organizada em 5 subáreas.

1 - Fundamentos do teste de software
Terminologia relacionada ao teste de software e relacionamento do teste com outras atividades.

2 - Níveis de teste
Descreve os diversos níveis de teste como teste unitário, de integração, de sistema, etc.

3 - Técnicas de teste
Descreve as várias técnicas de teste agrupadas em categorias, como teste baseado na intuição e experiência, baseadas em especificação e em códigos.

4 - Medições relacionadas aos testes
Apresenta comentários sobre a medição do teste de software, relacionado ao programa sob teste. As medições relativas ao processo são tratadas no tópico seguinte.

5 - Processo de Teste
Apresenta considerações práticas sobre o processo de teste, como as medições do processo, reuso dos testes, geração de caso de teste.

Áreas de Conhecimento

Manutenção de Software

Uma vez em operação anomalias são descobertas, alterações no ambiente operacional são feitas, e novos requisitos de usuários surgem. A fase de manutenção do ciclo de vida se inicia após a entrega, contudo as atividades de manutenção ocorrem bem antes (SWEBOK, 2004). A área de conhecimento de manutenção no Swebok é dividida em 4 subáreas.

1 - Fundamentos da manutenção de software
Introduz os conceitos e terminologia que formam o escopo da manutenção de software. Os tópicos provêm a definição e onde eles são empregados no processo de manutenção.

2 - Pontos chave da manutenção de software
Apresenta alguns tópicos importantes relacionados à aspectos técnicos e gerenciamento da manutenção de software.

3 - Processo de manutenção
Apresenta as referências e padrões utilizados para implementar o processo de manutenção.

4 - Técnicas de manutenção
Apresenta algumas técnicas aceitas no contexto da manutenção de software, como reegenharia, engenharia reversa, etc.

SWEBOK

Áreas de Conhecimento

Gerência de Configuração

A gerência de configuração é a disciplina que identifica a configuração do software em um ponto distinto no tempo, com o objetivo de sistematicamente controlar as mudanças, integridade e rastreabilidade (SWEBOK, 2004). Esta área de conhecimento está dividia em 6 seis subáreas.

1 - Gerenciamento do processo SCM
Cobre os tópicos do contexto organizacional, restrições e guias, planejamento, o plano e a vigilância do SCM.

2 - Identificação da configuração de software
Esta subárea identifica os itens a serem controlados, o estabelecimento de esquemas para os itens e suas versões, estabelecimento de técnicas e ferramentas, utilizados para adquirir e controlar os itens de software.

SWEBOK

Áreas de Conhecimento

3 - Controle da configuração de software
Apresenta o gerenciamento das mudanças durante o ciclo de vida de software.

4 - Status da contabilidade da configuração de software
Trata da gravação e relatórios da informação necessária para o efetivo gerenciamento da configuração de software.

5 - Auditoria da configuração de software
Esta subárea trata das auditorias nos itens de configuração de software, podem ser física ou funcionais.

6 - Gerenciamento da entrega de software
Trata da entrega da distribuição do software, tanto em distribuições internas quanto para os clientes.

SWEBOK

Áreas de Conhecimento

Gerência de Engenharia de Software

Segundo o Swebok (2004, p. 119), a gerência de configuração trata da aplicação das atividades de gerenciamento – planejamento, coordenação, medição, monitoramento, controle e publicação – para garantir o desenvolvimento e manutenção do software de maneira sistemática, disciplinada e quantificada. No guia esta disciplina está organizada em 6 subáreas.

1 - Inicialização de definição de escopo
O foco desta área é na determinação dos requisitos de software, através dos vários métodos de levantamento.

2 - Planejamento do projeto de software
Esta subárea prevê o planejamento de processo, determinação de entregas, esforço cronograma e estimativa de custo, alocação de recurso, gerenciamento de risco, gerenciamento de qualidade e planejamento do gerenciamento.

SWEBOK

Áreas de Conhecimento

3 - Execução do Projeto de Software
Trata da implementação dos planos, gerenciamento dos contratos com fornecedores, implementação do processo de medição, monitoração de processos, controle e relatórios.

4 - Revisar e avaliar
Esta área trata da determinação da satisfação de requisitos, e revisão e avaliação do desempenho.

5 - Encerramento
Apresenta as atividades para tratar o encerramento de projeto.

6 - Medição da Engenharia de Software
Trata dos programas de medição para os produtos e processos de software.

Áreas de Conhecimento

Processo de Engenharia de Software

A área de conhecimento do processo de engenharia de software atua em dois níveis: A definição e implementação do ciclo de vida do software e na definição, melhoria e avaliação de processos (SWEBOK 2004). Esta foi organizada em 4 subáreas.

1 - Implementação e Mudança de Processo
Esta subárea foca na mudança organizacional. Descreve a infraestrutura, atividades, modelos e considerações práticas de implementação do processo de mudança.

2 - Definição de processo
A definição de processo pode ser um procedimento, uma política ou um padrão. Pode contemplar processos como: suporte, gerenciamento de processos, processo automatizado.

3 - Avaliação de processo
Trata tanto do modelo como os métodos de avaliação de processos.

4 - Medição de Processo e Produto
Trata dos itens relevantes da medição de processo e produto de software.

SWEBOK

Áreas de Conhecimento

Métodos e Ferramentas de Engenharia de Software

Esta área de conhecimento trata de dois tópicos: os métodos e as ferramentas.

Os métodos trazem formalismo (estrutura) às atividades de engenharia de software com objetivo de executar estas atividades de forma sistemática e com sucesso.

Já as ferramentas baseadas em computador tem por objetivo auxiliar o ciclo de vida de desenvolvimento de software.

SWEBOK

Áreas de Conhecimento

Qualidade de Software

Procura endereçar os itens que são relacionados ao conceito de qualidade para o software, ou seja, atender os requisitos do software e alcançar a satisfação do cliente. Esta área cobre as técnicas estáticas, sem previsão de execução do software, e no guia está organizada em três subáreas.

1 - Fundamentos da qualidade de software
Trata de conceitos fundamentais relacionados a qualidade como cultura e ética, custos, etc.

2 - Processo de gerenciamento da qualidade de software
Define os processos, responsáveis, requisitos, medições e suas saídas e os canais de feedback.

3 - Considerações práticas
Trata de aspectos práticos da qualidade de software como requisitos e técnicas de gerenciamento da qualidade.

Métricas de Software

Introdução

As métricas de software são um importante recurso utilizado na engenharia de software para avaliar a qualidade e adequação de um produto de software, ou monitorar e controlar a execução de um processo.

No contexto da engenharia de software possuímos dois tipos métricas: as métricas de produto e as métricas do processo. As métricas de produto que têm por objetivo avaliar a adequação dos produtos de software a fatores de qualidade. Estas métricas podem ser aplicadas em itens como o código fonte, modelos (Estrutura dos componentes), o produto executável e a documentação.

Já as métricas do processo de software têm por objetivo acompanhar a execução de um processo com o objetivo de encontrar desvios. Usualmente fatores como produtividade, efetividade e acurácia são avaliados com este tipo de métrica.

Métricas de Software

Conceitos

Quando estamos tratando de métricas de software é importante ter bem identificado alguns conceitos sobre medições que podem gerar confusões de entendimento.

Ao falarmos em medir um produto ou processo de software podemos estar tratando de três itens: As medidas, as métricas e os indicadores. Segundo Pressman (2006) estes três termos são categorizados da seguinte forma:

- Medida: Fornece uma indicação quantitativa da extensão, quantidade, dimensão capacidade ou tamanho de algum atributo de um produto ou processo. Ex.: Número de defeitos e uma função do sistema;
- Métrica: Uma medida quantitativa do grau em que um sistema, componente ou processo possui um determinado atributo. Ex.: Número médio de erros encontrados por revisão, Número médio de erros encontrados no teste de unidade;
- Indicador: É uma métrica ou combinação de métricas que fornece profundidade na visão do processo de software, projeto de software ou produto em si. Ex.: Número médio esperado de detecção de defeitos nos testes unitários.

Métricas de Software

Conceitos

As métricas podem utilizar alguns recursos matemáticos para transformar em números (com significado) informações colhidas dos produtos ou processos de software.

Com certa frequência identificamos a razão, a proporção, o percentual e a taxa como as técnicas matemáticas mais empregadas.

Exemplos:

- Razão: (N° de desenvolvedores/ N° de Testadores) X 100%
- Proporção: N° de clientes satisfeitos/ N° total de clientes de um produto de software
- Percentual: % de defeitos relacionados a uma etapa do desenvolvimento
- Taxa: Taxa de Defeito= (N° de defeitos/Oportunidades de Falhas) x K*1000 linhas de código

Métricas de Software

Método GQM

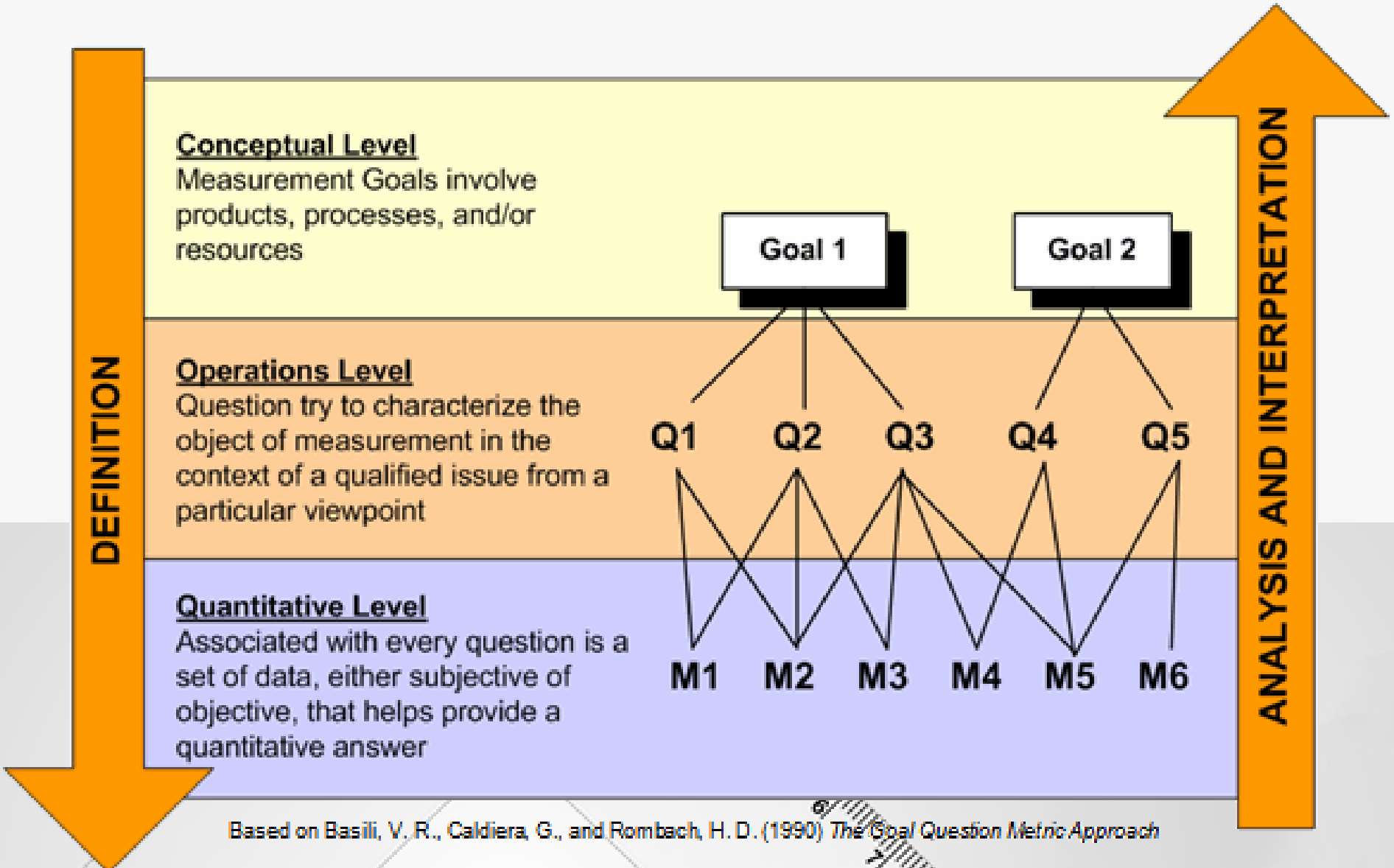
Apesar do formalismo aplicado às métricas e a necessidade da utilização de recursos matemáticos, esta não é a parte mais complicada do processo. Antes de se pensar nas métricas são necessários objetivos e metas claros que precisam ser endereçados por estas medições.

Pensando nisso foi que Basili, Caldiera e Rombach desenvolveram um método chamado GQM (Goal-Question-Metric), que organiza o paradigma de medição de software em três níveis (GQM, 2009):

- Nível conceitual: Definição de metas de medição envolvendo produtos, processos e/ou recursos (Goal).
- Nível operacional: Definição de questões que tentem caracterizar o objetivo da medição no contexto de um problema de qualidade de um ponto vista específico (Question).
- Nível quantitativo: Associar a cada questão um conjunto de dados, tanto subjetivos como objetivos, que ajudam a prover uma resposta quantitativa à questão (Metric).

Métricas de Software

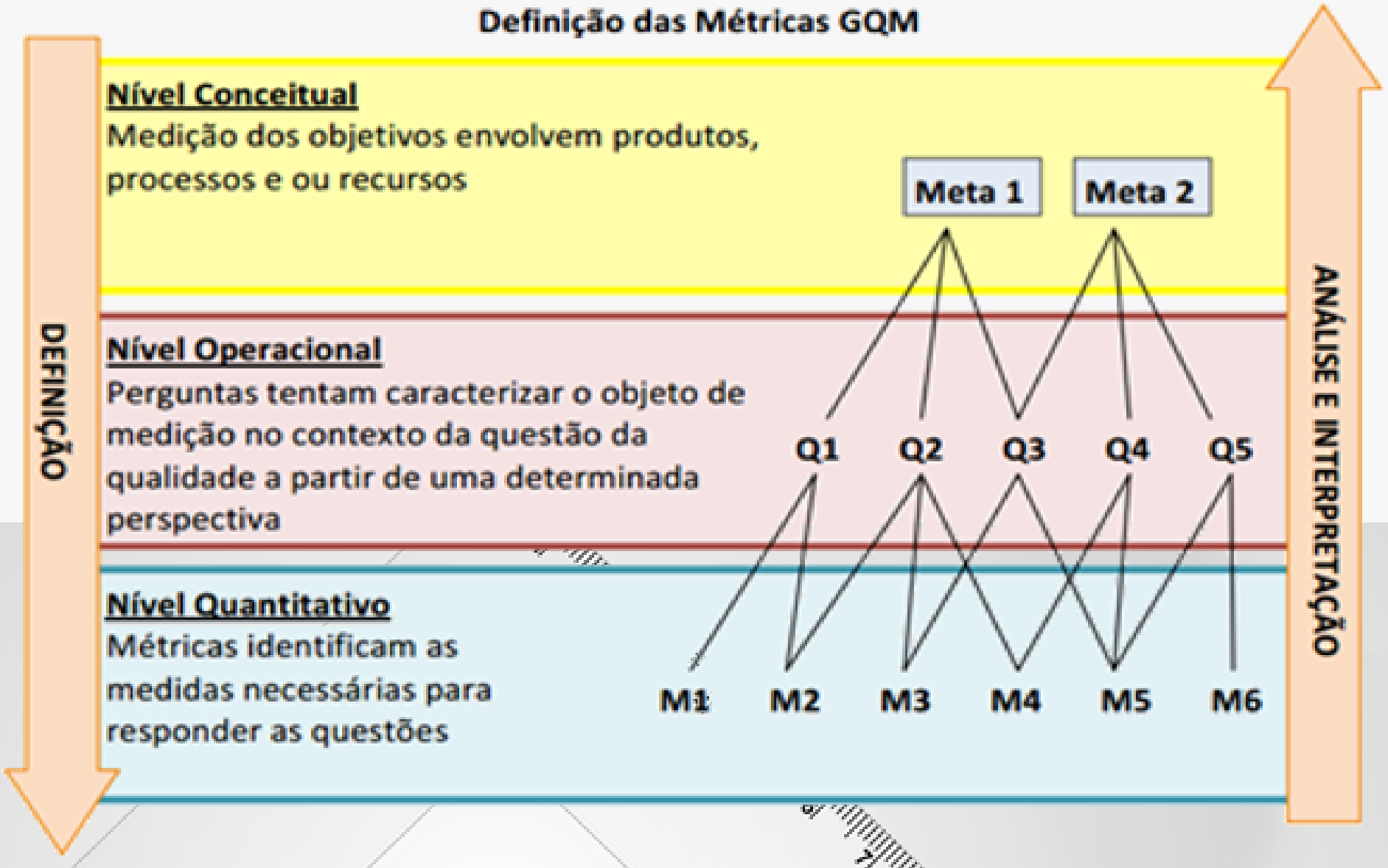
Método GQM



Based on Basili, V. R., Caldiera, G., and Rombach, H. D. (1990) *The Goal Question Metric Approach*

Métricas de Software

Método GQM



Método GQM

Segundo os autores, a utilização do método GQM pode ser melhor compreendido se tratado em 6 passos fundamentais (GQM, 2009):

1. Desenvolver um conjunto de metas de negocio, corporativas e de projetos, para produtividade e qualidade;
2. Gerar questões (baseada em modelos) que definam estas metas da maneira mais completa possível e quantificável;
3. Especificar as necessidades de medições a serem coletadas para responder estas questões e rastrear a conformidade do produto e processo com as metas;
4. Desenvolver mecanismos para a coleta de dados;
5. Coletar, validar e analisar os dados em tempo real provendo um feedback para os projetos para a tomada de ações corretivas;
6. Analisar os dados de uma maneira postmortem para avaliar a conformidade com as metas e fazer recomendações de melhorias futuras.

Métricas de Software

Normas ISO

Dentro do contexto de medição de software existem algumas normas da ISO (International Organization for Standardization) que tratam do assunto tanto nos seus aspectos técnicos como de gerenciamento e processuais. A norma ISO/IEC-9126 (Qualidade do Produto de Software) trata das características e sub características de qualidade bem como métricas que podem ser utilizadas.

Esta norma trata de três tipos de avaliação:

- Qualidade Interna: Aplicada na etapa de desenvolvimento e nos os itens não executáveis;
- Qualidade Externa: Aplicada na etapa de testes e nos itens executáveis do software;
- Qualidade em Uso: Procura avaliar o quanto o produto atende aos requisitos do usuário em seu ambiente previsto de uso.

Métricas de Produto de Software

Veremos agora algumas métricas relacionadas às medições do produto de software. Estas métricas em sua grande parte foram retiradas da norma ISO/IEC 9126 (2 e 3) e estão relacionadas a uma das 6 características de qualidade prevista nesta norma: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

Funcionalidade

A métrica a seguir trata da verificação da adequação da implementação do software em relação às funções previstas na especificação de requisitos.

Métricas de Software

Métricas de Produto de Software

Funcionalidade

Quesito	Descrição
Pergunta	Quão completa está a implementação em relação à especificação de requisitos?
Método	Executar os testes funcionais, com base na ER, e contar o número de funções que não estão implementadas
Formula	X = 1 – A/B A – Número de funções faltantes B – Número de funções da ER
Análise do resultado	0<=X<=1 (Quanto mais próx. de 1 melhor)

Métricas de Produto de Software

Confiabilidade

Uma das métricas mais comuns no quesito confiabilidade é a que trata da maturidade do software, o MTBF (Mean Time Between Failure) ou Taxa Média Entre Falhas. Esta métrica procura avaliar com que frequência o software falha quando está em operação.

Quesito	Descrição
Pergunta	Com que frequência o software falha em operação?
Método	Contabilizar as falhas durante um período de operação e computar a média dos intervalos entre as falhas
Formula	X = T1/A Y = T2/A A – Número total de defeitos identificados (no período observado) T1 – Tempo de operação
	T2 – Soma do intervalo de tempo entre a ocorrência de falhas consecutivas
Análise do resultado	0<X,Y (Quanto maior melhor, maior tempo entre falhas)

Métricas de Software

Métricas de Produto de Software

Usabilidade

A usabilidade também é um quesito que poder ser medido e avaliado para melhoria do produto de software. Um exemplo é a avaliação da operação do software, com a verificação da implementação de validação de dados de entrada no produto.

Quesito	Descrição
Pergunta	Qual a proporção de itens de entrada que possuem validação de dados?
Método	Contar o número de itens de entrada em que é possível validar os dados, e contar os itens em que não é possível validar a entrada
Formula	X = A/B A – Número de itens de entrada que são validados B – Número de itens de entrada que não são validados
Análise do resultado	0<=X<=1(Quanto mais próximo de 1 melhor)

Métricas de Software

Métricas de Produto de Software

Eficiência

Um das formas de se analisar a eficiência de um software ou produto de software é através da avaliação do *Throughput*. Esta métrica permite identificar quantas tarefas foram executadas em um período de tempo de execução do software. Esta métrica é muito utilizada em testes de carga e desempenho com o objetivo de validar se os requisitos não funcionais do software estão sendo cumpridos.

Quesito	**Descrição**
Pergunta	Quantas tarefas podem ser executadas com sucesso em um período de tempo?
Método	Iniciar vários Jobs de tarefas e verificar o tempo que leva para completar a operação
Formula	X = A/T A – Número de tarefas completadas com sucesso T – Período de tempo observado
Análise do resultado	0<X (Quanto maior melhor)

Métricas de Produto de Software

Manutenibilidade

Até aspectos relacionados à manutenção de software podem ser avaliados. Um dos pontos mais interessantes é saber qual o impacto de uma mudança no software. Dependendo do resultado, esta métrica pode gerar várias atividades de melhoria de processo ou até mesmo uma reavaliação da arquitetura, documentação e códigos fontes do software.

Quesito	Descrição
Pergunta	Com que frequência ocorrem impactos adversos após uma modificação?
Método	Contabilizar o número de impactos adversos depois das modificações e compará-lo com o número total de modificações
Formula	X = 1-A/B A – Número de impactos adversos detectados após modificações B – Número de modificações feitas
Análise do resultado	0<=X<=1 (Quanto mais perto de 1 melhor)

Métricas de Software

Métricas de Produto de Software

Portabilidade

Os quesitos relacionados a portabilidade de um software podem ser avaliados, como por exemplo, o quão fácil é adaptar um software a um novo ambiente ou plataforma.

Quesito	Descrição
Pergunta	Pode o usuário ou o mantenedor facilmente adaptar o software a um ambiente?
Método	Observar o comportamento do usuário ou mantenedor em uma tentativa de adaptar o software ao ambiente
Formula	T = Soma do tempo gasto para completar a adaptação
Análise do resultado	0< T (Quanto menor melhor)

Métricas de Processo de Software

O processo de desenvolvimento de software, como outros processos de engenharia, podem e devem ser medidos com o objetivo de controlar e monitorar sua execução a procura de desvios ou oportunidades de melhoria. Esta monitoração e acompanhamento pode trazer benefícios como:

- Obter informações sobre custos;
- Retorno do investimento;
- Identificar pontos que precisam de melhoria;
- Avaliar o efeito ou resultado de uma melhoria;
- Avaliar a produtividade;
- Identificar o comportamento e localização dos defeitos.

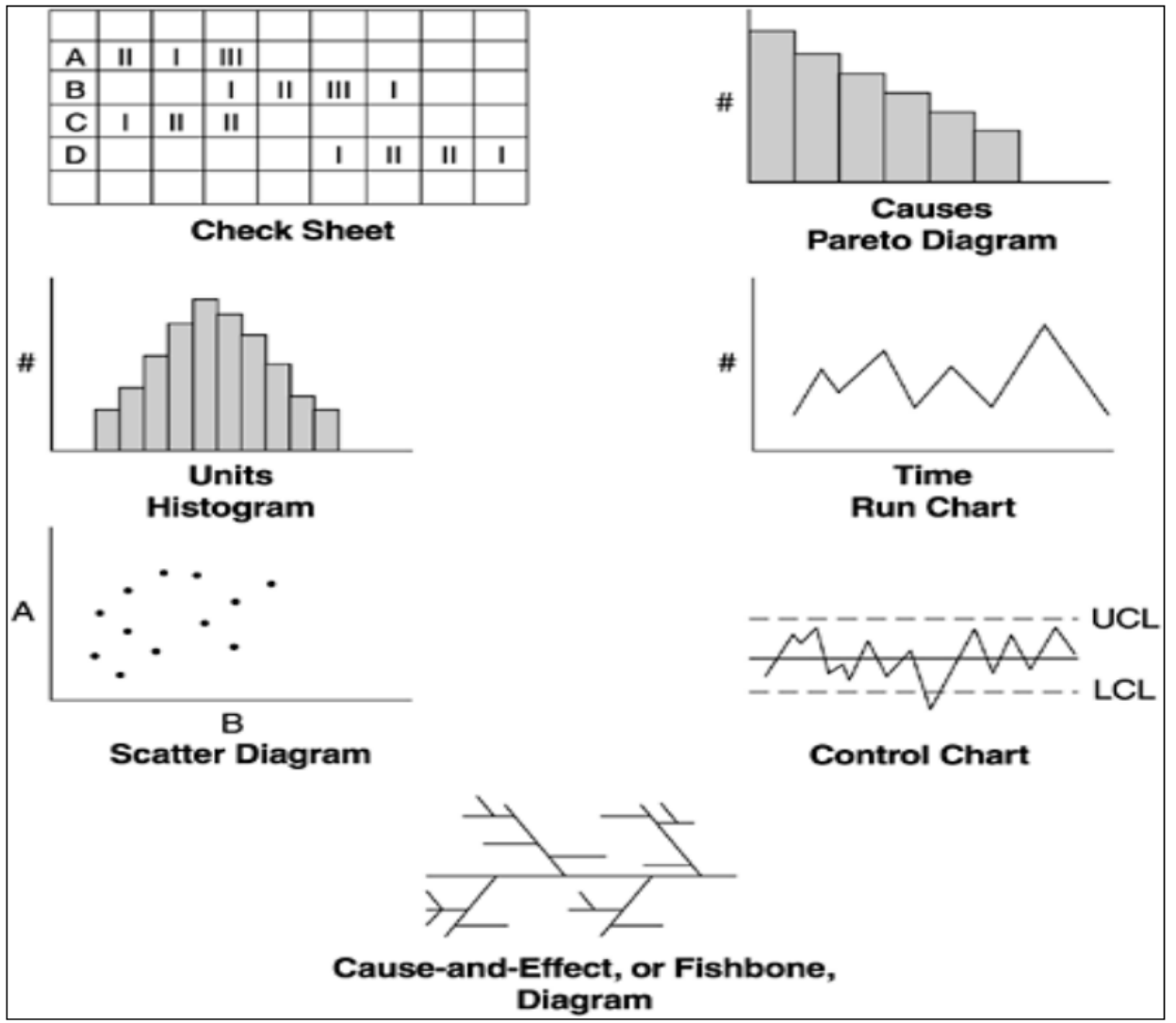
Há algumas décadas um conjunto de técnicas propostas por Ishikawa se tornaram muito comuns no âmbito da engenharia e processos industriais e também na área de desenvolvimento de software. Estas técnicas denominadas controle estatístico de processo, são utilizadas para identificar e remover variações no processo que excedam as variações esperadas de causas naturais. O propósito do controle de processo é detectar qualquer anomalia no processo.

Métricas de Software

Métricas de Processo de Software

Tais técnicas e/ou ferramentas somam sete: as planilhas de verificação (check sheet), diagramas de Pareto (causes Pareto diagram), histogramas (units histogram), gráficos de execução (time run chart), diagramas de dispersão (scatter diagram), gráficos de controle (control chart) e diagrama de causa e efeito ou espinha de peixe (cause-and-effect, or fishbone diagram).

Pesquise sobre cada uma dessas técnicas na Internet para se aprofundar no assunto.



Métricas de Processo de Software

Como exemplo, serão apresentadas algumas métricas normalmente aplicadas ao processo de desenvolvimento de software: projeto, testes e manutenção. Estas métricas também seguem o padrão do GQM.

Projeto

Uma das métricas mais aplicadas a projetos de desenvolvimento de software são as que tratam da taxa de acerto em relação a cronograma e esforço.

Quesito	Descrição
Pergunta	Qual é a taxa de acerto do cronograma/esforço do projeto?
Método	Contabilizar a o cronograma/esforço e comparar com o planejado
Formula	ACP = Duração Atual do Projeto / Duração Estimada do Projeto ACE = Esforço Atual do Projeto / Esforço Estimado do Projeto
Análise do resultado	Quanto mais próximo de 1 melhor

Métricas de Processo de Software

Testes

O processo de teste de software é um dos que permite a aplicação de vários tipos de métricas. Notadamente temos um destaque para as métricas de efetividade que procuram avaliar a capacidade dos testes de detectar defeitos.

Uma das métricas mais utilizadas é a de efetividade de remoção de defeitos, que avalia a quantidade de defeitos removidos em uma determinada etapa de teste em comparação com as etapas seguintes. Esta métrica deve ser utilizada com cuidado, pois ela pode apresentar desvios se aplicada com o projeto ainda em andamento.

Quesito	Descrição
Pergunta	Quão efetivo é determinada fase dos meus testes?
Método	Executar os testes e contabilizar a quantidade de defeitos detectados em cada fase
Formula	ERD = Defeitos Removidos Durante uma Fase / (Defeitos Removidos Durante uma Fase + Defeitos Encontrados Depois) * 100%
Análise do resultado	Quanto mais próximo de 100% melhor

Métricas de Processo de Software

Manutenção

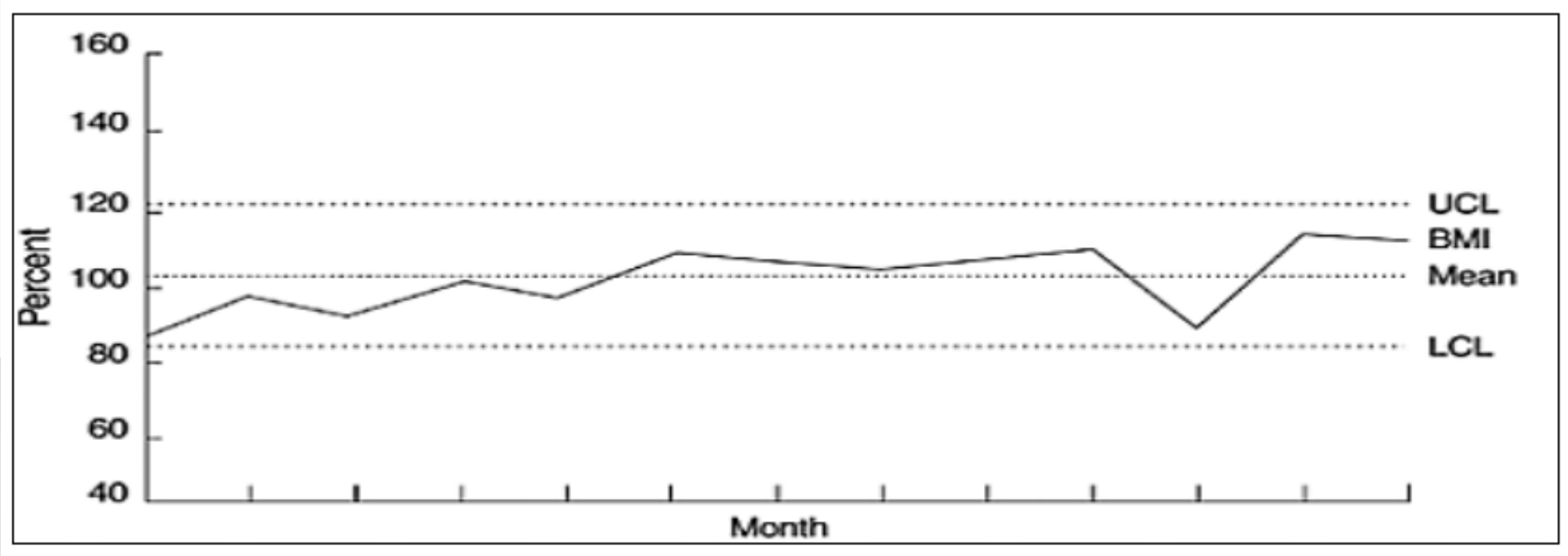
Já para o processo de manutenção uma das métricas mais comuns é a avaliação do backlog, esta métrica avalia se o ritmo de atendimento às demandas de manutenção está adequado em relação à quantidade de demandas que são recebidas.

Quesito	Descrição
Pergunta	Como está o atendimento às demandas de correção?
Método	Contabilizar o número de demandas atendidas e as que chegaram no mês
Formula	IGB = Número de Problemas Encerrados / Número de Problemas que Chegaram * 100% (Por mês)
Análise do resultado	Quanto mais próximo de 100% melhor

Métricas de Processo de Software

Manutenção

Esta é uma métrica que pode, por exemplo, ser exibida em um gráfico de controle com limites superiores e inferiores, para garantir que o esforço está adequado ao volume de demandas que são recebidas.



Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 9126-1: Engenharia de Software – Qualidade do Produto. Rio de Janeiro: ABNT, 2003.

CARNEGIE MELLON UNIVERSITY. Software Engineering Institute. CMMI for Development, Version 1.2. Pittsburgh, 2006.

IEEE Software Engeneering Coordinating Commitee. SWEBOK: Guide to the Engeneering Body of Knowledge – Version 2004. Disponível em: http://www.swebok.org.

KAN, Stephen h. Metrics and Models in Software Quality Engineering. Second Edition. Addison Wesley, 2002.

KOSCIANSKI, André; SOARES, Michel dos Santos. Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software, 2006.

PAULK, Mark C. et al. Capability Maturity ModelSM for Software. Versão 1.1. Pittsburgh: Software Engineering Institute - Carnegie Mellon University, 1993.

PRESSMAN, Roger S. Engenharia de Software. 6. ed. McGraw-Hill, 2006.

SOMMERVILLE, IAN. Engenharia de Software. 8. ed. Addison Wesley, 2007.