

INFO1003 – Eng Sw II

Engenharia de Software

Prof. Marcelo Soares Pimenta
mpimenta@inf.ufrgs.br

slides – arq 5

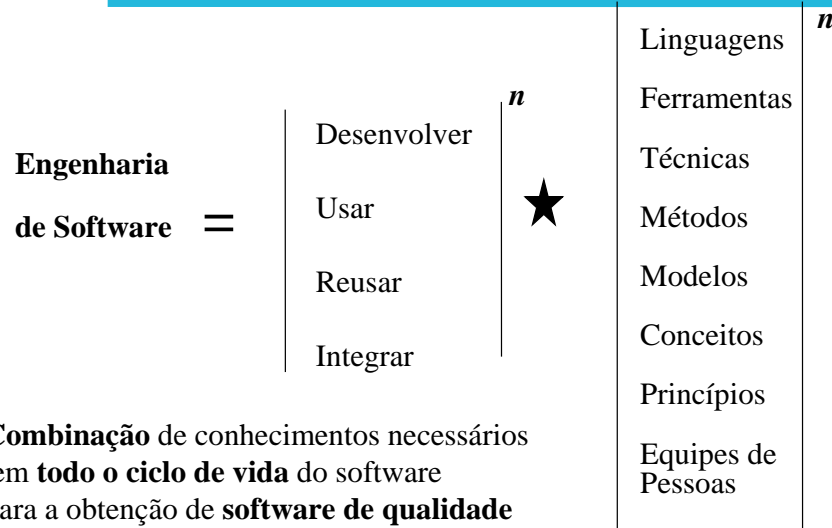
©Pimenta 2010

Programa

- 1 - Revisão de Fundamentos de Engenharia de Software:
- 2-Análise e Projeto Orientados a Objetos: Aprofundando a prática de modelagem com UML
- 3-Teste de Software
4. Gerenciamento de Versões e Configurações e Gerência de Requisitos
5. Qualidade de Software e Modelos de Maturidade: CMMI, SPICE, MPS-BR
7. Reuso de Software: teoria e prática, abordagens, patterns, frameworks, componentes
8. Métodos Ágeis : Motivação, Características, Exemplos
- 9.Novas Tendências em Engenharia de Software

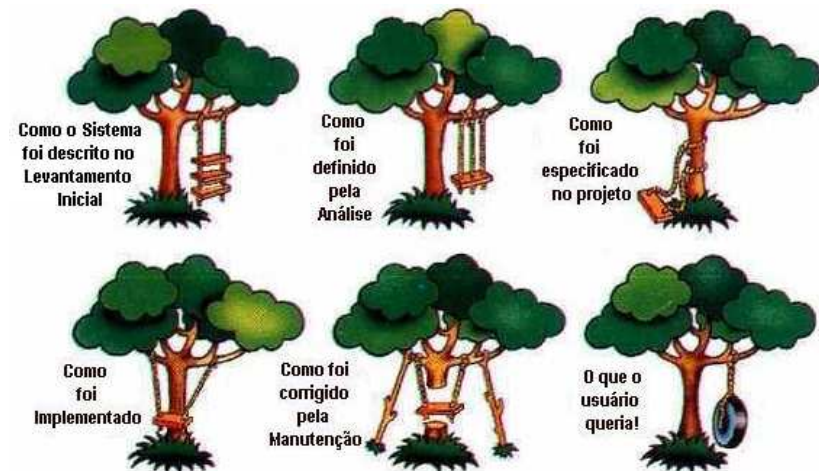
©Pimenta 2010

Engenharia de Software

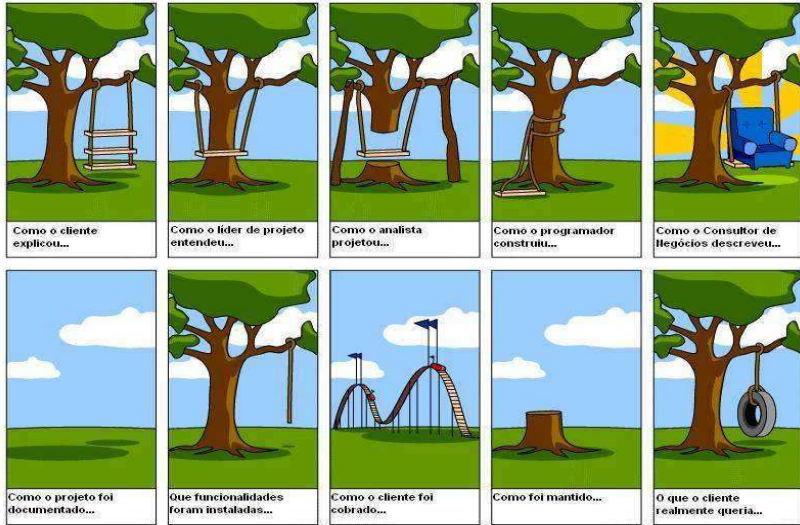


©Pimenta 2010

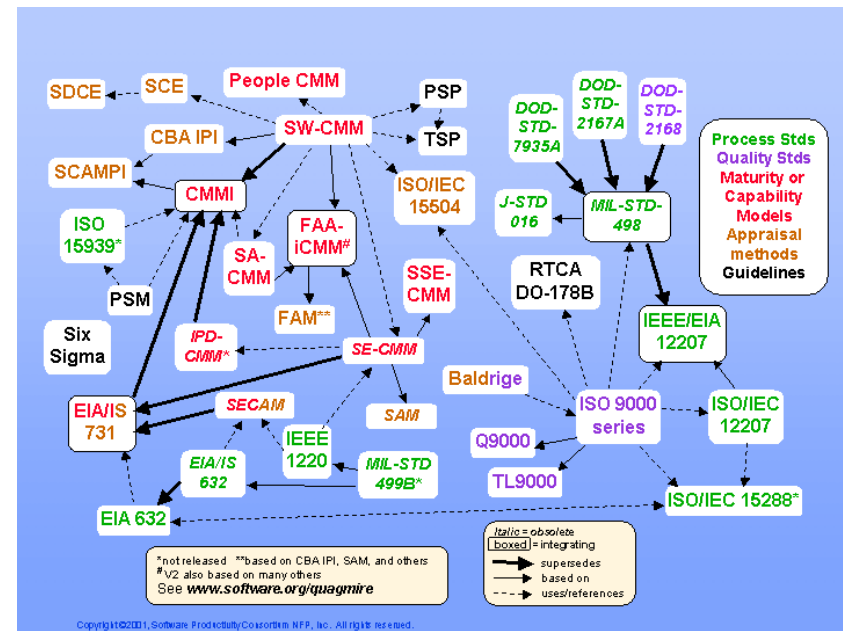
Qual o problema do software?



Qual o problema do Software?

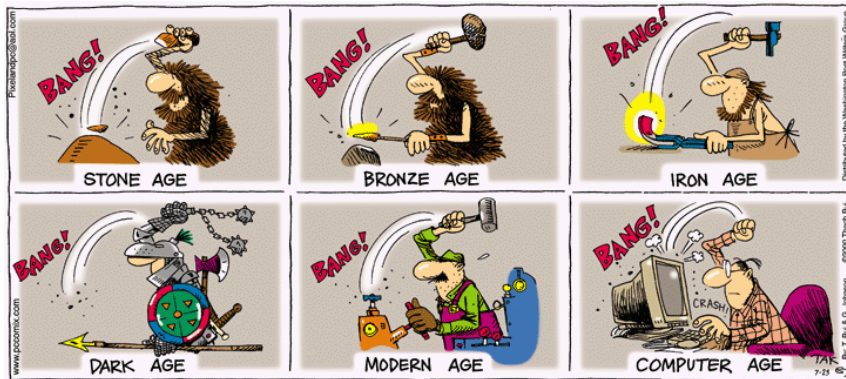


©Pimenta 2010



©Pimenta 2010

Qual o problema das organizações?



©Pimenta 2010

O que é ter qualidade?

- ✓ *Satisfazer as necessidades e expectativas dos clientes?*
- ✓ *Estar conforme com as especificações de produto?*
- ✓ *Ter a melhor relação custo x benefício?*
- ✓ *Fornecer produtos adequados ao uso?*
- ✓ *Fazer mais, melhor e mais barato?*
- ✓ *Ser o mais atrativo do mercado?*
- ✓ *Etc ...*

- Subjetiva demais ...
- Centrada no Produto ...
- Centrada no Valor ...
- Centrada no Processo ...
- Centrada no Cliente ...

PROCESSO: Um sistema de atividades que usa recursos para transformar entradas em saídas.

PRODUTO: O resultado de um processo.

SISTEMA: Conjunto de elementos inter-relacionados ou interconexos.

CARACTERÍSTICA: Propriedade (coisa) distinguível.

REQUISITO: Necessidade ou expectativa que é declarada, usualmente implícita ou obrigatória.

QUALIDADE: Habilidade de um conjunto de características inerentes de um produto, sistema ou processo para atender plenamente os requisitos dos clientes ou outras partes interessadas.

Características do software

CARACTERÍSTICA	SUB-CARACTERÍSTICA	PERGUNTA CHAVE
Funcionalidade Satisfaz as necessidades?	Adequação	Propõe-se a fazer o que é apropriado?
	Acurácia	Faz o que foi proposto de forma correta?
	Interoperabilidade	Interage com os sistemas especificados?
	Conformidade	Está de acordo com as normas, leis, etc.?
Confiabilidade É imune a falhas?	Segurança de acesso	Evita acesso não autorizado aos dados?
	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas, como ele reage?
	Recuperabilidade	É capaz de recuperar dados em caso de falha?
Usabilidade É fácil de usar?	Inteligibilidade	É fácil entender o conceito e a aplicação?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar?

* Características do Produto de Software (ISO/IEC 9126)

Características do software

CARACTERÍSTICA	SUB-CARACTERÍSTICA	PERGUNTA CHAVE
Eficiência É rápido e "enxuto"?	Tempo	Qual é o tempo de resposta, a velocidade de execução?
	Recursos	Quanto recurso usa? Durante quanto tempo?
Manutenibilidade É fácil de modificar?	Analisabilidade	É fácil de encontrar uma falha, quando ocorre?
	Modificabilidade	É fácil modificar e adaptar?
	Estabilidade	Há grande risco quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade É fácil de usar em outro ambiente?	Adaptabilidade	É fácil adaptar a outros ambientes?
	Cap. para ser instalado	É fácil instalar em outros ambientes?
	Conformidade	Está de acordo com padrões de portabilidade?
	Cap. para substituir	É fácil usar para substituir outro?

* Características do Produto de Software (ISO/IEC 9126)

O que é gestão da qualidade?

ORGANIZAÇÃO: Grupo de pessoas e instalações com um arranjo ordenado de responsabilidades, autoridades e relações.

GESTÃO: Atividades coordenadas para dirigir e controlar uma organização.

GESTÃO DA QUALIDADE: Atividades coordenadas para dirigir e controlar uma organização no que diz respeito à qualidade.

OBJETIVOS DA QUALIDADE: Alguma coisa pensada, ou alvos, relacionados à qualidade.

POLÍTICA DA QUALIDADE: Totalidade das intenções e direção de uma organização relativas à qualidade, formalmente expressas pela alta direção.

SISTEMA DE GESTÃO DA QUALIDADE: Sistema para estabelecer uma política e objetivos da qualidade da qualidade, bem como os métodos para alcançar esses objetivos.

No fundo, é sermos efetivos!

EFICÁCIA: Medida da extensão na qual atividades planejadas são realizadas e os resultados planejados são alcançados.
Isto é, "o quanto fazemos a coisa certa".

EFICIÊNCIA: Relação entre o resultado obtido e os recursos usados.
Isto é, "com quanto fazemos a coisa certa".

EFETIVIDADE = EFICÁCIA + EFICIÊNCIA



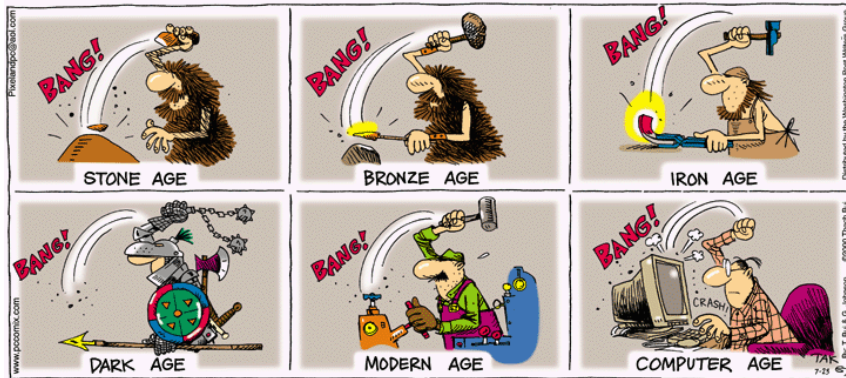
Como ser efetivo?



"Será muito difícil aprender a nadar somente com a leitura do manual do bom nadador!"

Pratique continuamente o método e aprimore a técnica com base em orientações e resultados!

Qual o problema da organização?



Visões da Qualidade

- Visão da operação:
 - Alvo: atingir as expectativas dos consumidores;
- Visão do consumidor:
 - Objetivo: atingir a percepção do produto ou serviço prestado.

Controle da Qualidade

- Etapas:
 1. Definir as características de qualidade do produto ou serviço.
 2. Definir como medir cada característica.
 3. Estabelecer padrões de qualidade.
 4. Controlar a qualidade pelos padrões.
 5. Encontrar e corrigir causas de má qualidade.
 6. Continuar a fazer melhoramentos.

Problemas do Software

- Software não possui existência física.
- “Clientes não sabem o que querem”.
- “Clientes mudam de idéia”, durante o desenvolvimento de sistemas.
- Mudanças de hardware e software de apoio durante o desenvolvimento.
- Clientes possuem altas expectativas.

Definições de Qualidade de Software

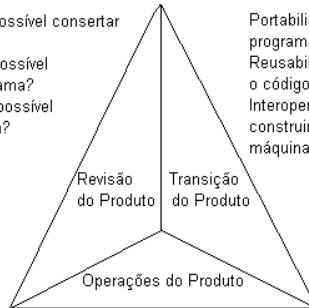
- Qualidade relacionada à conformidade do software com os requisitos:
 - Problema: raramente os requisitos estão completos.
- Qualidade relacionada à satisfação do usuário:
 - Problema: usuários diferentes.

Modelo de McCall - 1977

- Identifica três áreas de trabalho:
 - operação;
 - revisão;
 - transição.
- Identificação de critérios em cada área de trabalho.

Modelo de McCall

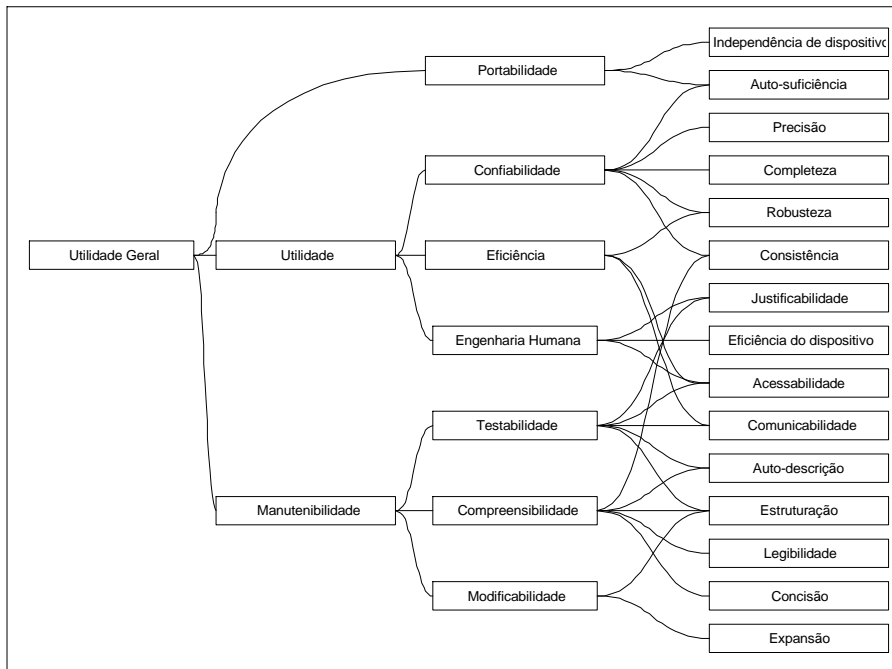
Manutenção: É possível consertar o programa?
Flexibilidade: É possível modificar o programa?
Testabilidade: É possível testar o programa?



Portabilidade: É possível usar o programa em outra máquina?
Reusabilidade: É possível reutilizar o código do programa?
Interoperabilidade: É possível construir a interface com outra máquina?

Correção: O programa faz o desejado?
Confiabilidade: O programa faz o desejado de maneira precisa?
Eficiência: O programa executará tão bem quanto possível?
Integridade: O programa é seguro?
Usabilidade: O programa pode ser executado?

Modelo de Boehm - 1978



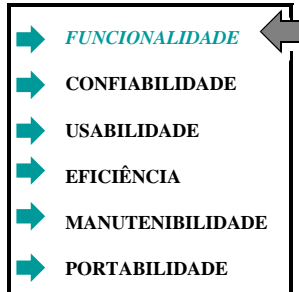
CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE NBR ISO / IEC 9126

- ➡ FUNCIONALIDADE
- ➡ CONFIABILIDADE
- ➡ USABILIDADE
- ➡ EFICIÊNCIA
- ➡ MANUTENIBILIDADE
- ➡ PORTABILIDADE

CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE

NBR ISO / IEC 9126

1



FUNCIONALIDADE

Adequação

PARA AS TAREFAS ESPECIFICADAS

Acurácia

RESULTADOS / EFEITOS CORRETOS

Interoperabilidade

INTERAGIR COM SISTEMAS ESPECIFICADOS

Conformidade

A NORMAS, CONVENÇÕES, LEIS, DESCRIÇÕES

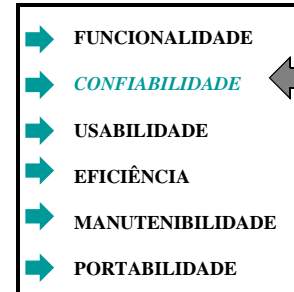
Segurança de acesso

EVITAR ACESSO ACIDENTAL OU DELIBERADO

CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE

NBR ISO / IEC 9126

2



CONFIABILIDADE

Maturidade

FREQUÊNCIA DE FALHAS POR DEFEITOS

Tolerância a falhas

CAPACIDADE EM MANTER DESEMPENHO

- FALHAS NO SOFTWARE
- VIOLAÇÃO NAS INTERFACES

Recuperabilidade

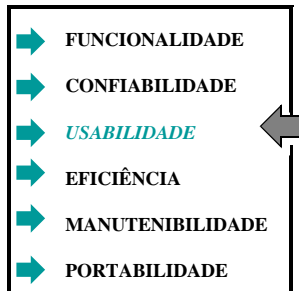
DE DADOS E DE DESEMPENHO

TEMPO E ESFORÇO NECESSÁRIOS

CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE

NBR ISO / IEC 9126

3



USABILIDADE

Inteligibilidade

ESFORÇO PARA ENTENDER, IDENTIFICAR

Apreensibilidade

ESFORÇO PARA APRENDER, APLICAR

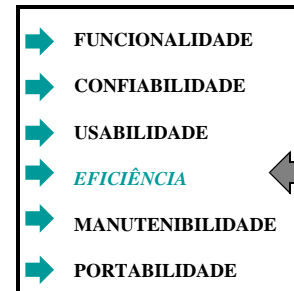
Operacionalidade

ESFORÇO PARA OPERAR, CONTROLAR

CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE

NBR ISO / IEC 9126

4



EFICIÊNCIA

Em relação ao tempo

TEMPO DE RESPOSTA, DE PROCESSAMENTO
VELOCIDADE DE EXECUÇÃO DAS FUNÇÕES

Em relação aos recursos

QUANTIDADE UTILIZADA E
DURAÇÃO DO SEU USO

CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE NBR ISO / IEC 9126

5

MANUTENIBILIDADE

Analísabilidade

ESFORÇO PARA DIAGNÓSTICO, IDENTIFICAÇÃO DE FALHAS

Modificabilidade

ESFORÇO PARA MODIFICAÇÃO, ADAPTAÇÃO, REMOÇÃO DE DEFEITOS

Estabilidade

RISCO DE EFEITOS INESPERADOS OCACIONADOS POR MODIFICAÇÕES

Testabilidade

ESFORÇO PARA VALIDAÇÃO DAS MODIFICAÇÕES

→ FUNCIONALIDADE
→ CONFIABILIDADE
→ USABILIDADE
→ EFICIÊNCIA
→ **MANUTENIBILIDADE**
→ PORTABILIDADE



CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE NBR ISO / IEC 9126

6

PORTABILIDADE

Adaptabilidade

A OUTROS AMBIENTES POR MEIOS E AÇÕES PRÓPRIAS

Instalabilidade

ESFORÇO PARA A INSTALAÇÃO

Conformidade

ADERÊNCIA A CONVENÇÕES E PADRÕES FORMAIS DE PORTABILIDADE

Capacidade para substituir

ESFORÇO E CAPACIDADE PARA SUBSTITUIR OUTRO SOFTWARE

→ FUNCIONALIDADE
→ CONFIABILIDADE
→ USABILIDADE
→ EFICIÊNCIA
→ MANUTENIBILIDADE
→ **PORTABILIDADE**



Quality Management

Sommerville, I. Software Engineering, 7a ed

Objectives

- To introduce the quality management process and key quality management activities
- To explain the role of standards in quality management
- To explain the concept of a software metric, predictor metrics and control metrics
- To explain how measurement may be used in assessing software quality and the limitations of software measurement

Topics covered

- Process and product quality
- Quality assurance and standards
- Quality planning
- Quality control

Software quality management

- Concerned with ensuring that the required level of quality is achieved in a software product.
- Involves defining appropriate quality standards and procedures and ensuring that these are followed.
- Should aim to develop a 'quality culture' where quality is seen as everyone's responsibility.

What is quality?

- Quality, simplistically, means that a product should meet its specification.
- This is problematical for software systems
 - There is a tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.);
 - Some quality requirements are difficult to specify in an unambiguous way;
 - Software specifications are usually incomplete and often inconsistent.

The quality compromise

- We cannot wait for specifications to improve before paying attention to quality management.
- We must put quality management procedures into place to improve quality in spite of imperfect specification.

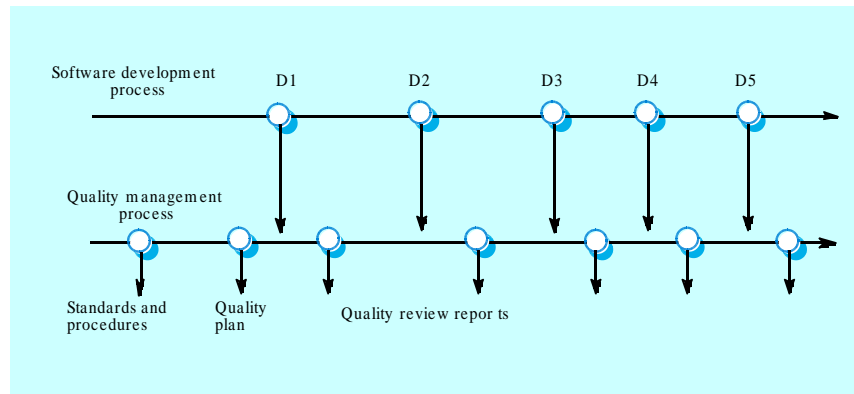
Scope of quality management

- Quality management is particularly important for large, complex systems. The quality documentation is a record of progress and supports continuity of development as the development team changes.
- For smaller systems, quality management needs less documentation and should focus on establishing a quality culture.

Quality management activities

- Quality assurance
 - Establish organisational procedures and standards for quality.
- Quality planning
 - Select applicable procedures and standards for a particular project and modify these as required.
- Quality control
 - Ensure that procedures and standards are followed by the software development team.
- Quality management should be separate from project management to ensure independence.

Quality management and software development



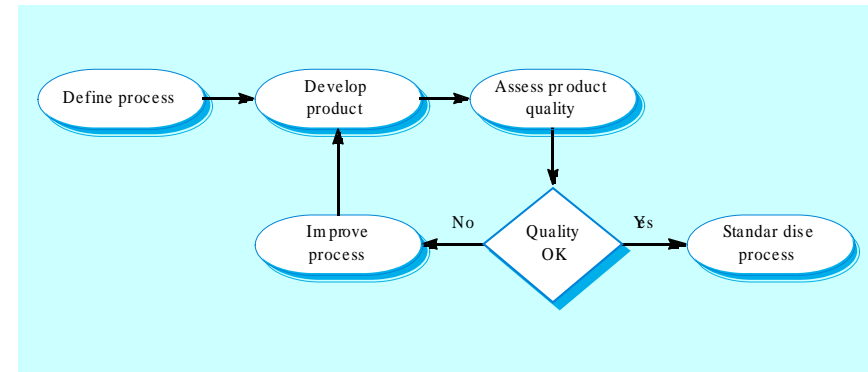
Process and product quality

- The quality of a developed product is influenced by the quality of the production process.
- This is important in software development as some product quality attributes are hard to assess.
- However, there is a very complex and poorly understood relationship between software processes and product quality.

Process-based quality

- There is a straightforward link between process and product in manufactured goods.
- More complex for software because:
 - The application of individual skills and experience is particularly important in software development;
 - External factors such as the novelty of an application or the need for an accelerated development schedule may impair product quality.
- Care must be taken not to impose inappropriate process standards - these could reduce rather than improve the product quality.

Process-based quality



Practical process quality

- Define process standards such as how reviews should be conducted, configuration management, etc.
- Monitor the development process to ensure that standards are being followed.
- Report on the process to project management and software procurer.
- Don't use inappropriate practices simply because standards have been established.

Quality assurance and standards

- Standards are the key to effective quality management.
- They may be international, national, organizational or project standards.
- **Product standards** define characteristics that all components should exhibit e.g. a common programming style.
- **Process standards** define how the software process should be enacted.

Importance of standards

- Encapsulation of best practice- avoids repetition of past mistakes.
- They are a framework for quality assurance processes - they involve checking compliance to standards.
- They provide continuity - new staff can understand the organisation by understanding the standards that are used.

Product and process standards

Product standards

Design review form

Requirements document structure

Method header format

Java programming style

Project plan format

Change request form

Process standards

Design review conduct

Submission of documents to CM

Version release process

Project plan approval process

Change control process

Test recording process

Problems with standards

- They may not be seen as relevant and up-to-date by software engineers.
- They often involve too much bureaucratic form filling.
- If they are unsupported by software tools, tedious manual work is often involved to maintain the documentation associated with the standards.

Standards development

- Involve practitioners in development. Engineers should understand the rationale underlying a standard.
- Review standards and their usage regularly. Standards can quickly become outdated and this reduces their credibility amongst practitioners.
- Detailed standards should have associated tool support. Excessive clerical work is the most significant complaint against standards.

ISO 9000

- An international set of standards for quality management.
- Applicable to a range of organisations from manufacturing to service industries.
- ISO 9001 applicable to organisations which design, develop and maintain products.
- ISO 9001 is a generic model of the quality process that must be instantiated for each organisation using the standard.

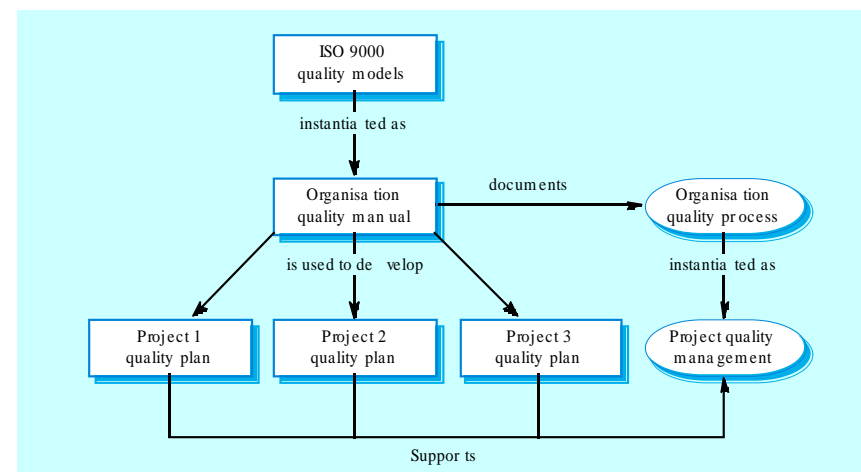
ISO 9001

Management responsibility	Quality system
Control of non-conforming products	Design control
Handling, storage, packaging and delivery	Purchasing
Purchaser-supplied products	Product identification and traceability
Process control	Inspection and testing
Inspection and test equipment	Inspection and test status
Contract review	Corrective action
Document control	Quality records
Internal quality audits	Training
Servicing	Statistical techniques

ISO 9000 certification

- Quality standards and procedures should be documented in an organisational quality manual.
- An external body may certify that an organisation's quality manual conforms to ISO 9000 standards.
- Some customers require suppliers to be ISO 9000 certified although the need for flexibility here is increasingly recognised.

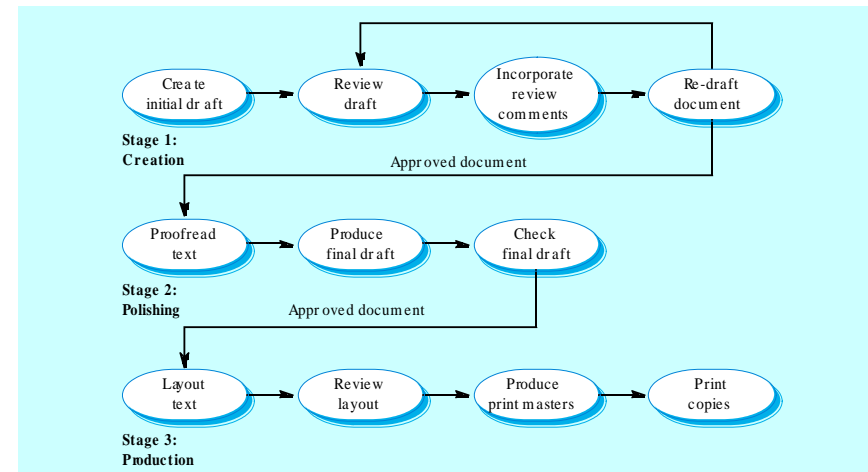
ISO 9000 and quality management



Documentation standards

- Particularly important - documents are the tangible manifestation of the software.
- Documentation process standards
 - Concerned with how documents should be developed, validated and maintained.
- Document standards
 - Concerned with document contents, structure, and appearance.
- Document interchange standards
 - Concerned with the compatibility of electronic documents.

Documentation process



Document standards

- Document identification standards
 - How documents are uniquely identified.
- Document structure standards
 - Standard structure for project documents.
- Document presentation standards
 - Define fonts and styles, use of logos, etc.
- Document update standards
 - Define how changes from previous versions are reflected in a document.

Document interchange standards

- Interchange standards allow electronic documents to be exchanged, mailed, etc.
- Documents are produced using different systems and on different computers. Even when standard tools are used, standards are needed to define conventions for their use e.g. use of style sheets and macros.
- Need for archiving. The lifetime of word processing systems may be much less than the lifetime of the software being documented. An archiving standard may be defined to ensure that the document can be accessed in future.

Quality planning

- A quality plan sets out the desired product qualities and how these are assessed and defines the most significant quality attributes.
- The quality plan should define the quality assessment process.
- It should set out which organisational standards should be applied and, where necessary, define new standards to be used.

Quality plans

- Quality plan structure
 - Product introduction;
 - Product plans;
 - Process descriptions;
 - Quality goals;
 - Risks and risk management.
- Quality plans should be short, succinct documents
 - If they are too long, no-one will read them.

Software quality attributes

Safety	Understandability	Portability
Security	Testability	Usability
Reliability	Adaptability	Reusability
Resilience	Modularity	Efficiency
Robustness	Complexity	Learnability

Quality control

- This involves checking the software development process to ensure that procedures and standards are being followed.
- There are two approaches to quality control
 - Quality reviews;
 - Automated software assessment and software measurement.

Quality reviews

- This is the principal method of validating the quality of a process or of a product.
- A group examines part or all of a process or system and its documentation to find potential problems.
- There are different types of review with different objectives
 - Inspections for defect removal (product);
 - Reviews for progress assessment (product and process);
 - Quality reviews (product and standards).

Types of review

Review type	Principal purpose
Design or program inspections	To detect detailed errors in the requirements, design or code. A checklist of possible errors should drive the review.
Progress reviews	To provide information for management about the overall progress of the project. This is both a process and a product review and is concerned with costs, plans and schedules.
Quality reviews	To carry out a technical analysis of product components or documentation to find mismatches between the specification and the component design, code or documentation and to ensure that defined quality standards have been followed.

Quality reviews

- A group of people carefully examine part or all of a software system and its associated documentation.
- Code, designs, specifications, test plans, standards, etc. can all be reviewed.
- Software or documents may be 'signed off' at a review which signifies that progress to the next development stage has been approved by management.

Review functions

- Quality function - they are part of the general quality management process.
- Project management function - they provide information for project managers.
- Training and communication function - product knowledge is passed between development team members.

Quality reviews

- The objective is the discovery of system defects and inconsistencies.
- Any documents produced in the process may be reviewed.
- Review teams should be relatively small and reviews should be fairly short.
- Records should always be maintained of quality reviews.

Review results

- Comments made during the review should be classified
 - No action. No change to the software or documentation is required;
 - Refer for repair. Designer or programmer should correct an identified fault;
 - Reconsider overall design. The problem identified in the review impacts other parts of the design. Some overall judgement must be made about the most cost-effective way of solving the problem;
- Requirements and specification errors may have to be referred to the client.

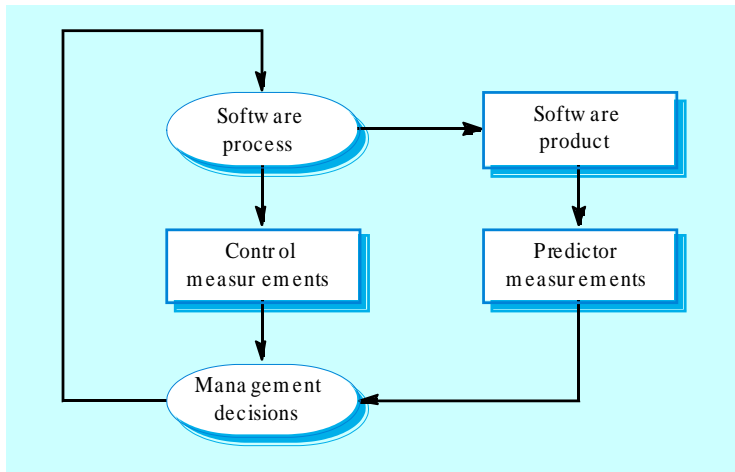
Software measurement and metrics

- Software measurement is concerned with deriving a numeric value for an attribute of a software product or process.
- This allows for objective comparisons between techniques and processes.
- Although some companies have introduced measurement programmes, most organisations still don't make systematic use of software measurement.
- There are few established standards in this area.

Software metric

- Any type of measurement which relates to a software system, process or related documentation
 - Lines of code in a program, the Fog index, number of person-days required to develop a component.
- Allow the software and the software process to be quantified.
- May be used to predict product attributes or to control the software process.
- Product metrics can be used for general predictions or to identify anomalous components.

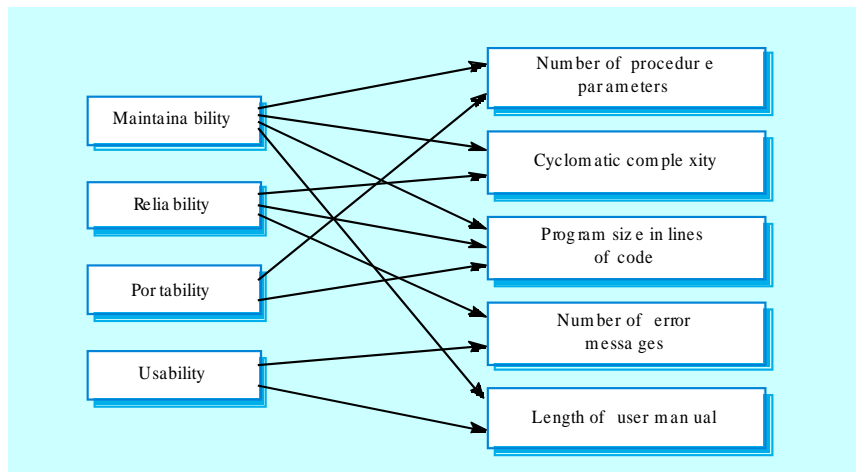
Predictor and control metrics



Metrics assumptions

- A software property can be measured.
- The relationship exists between what we can measure and what we want to know. We can only measure internal attributes but are often more interested in external software attributes.
- This relationship has been formalised and validated.
- It may be difficult to relate what can be measured to desirable external quality attributes.

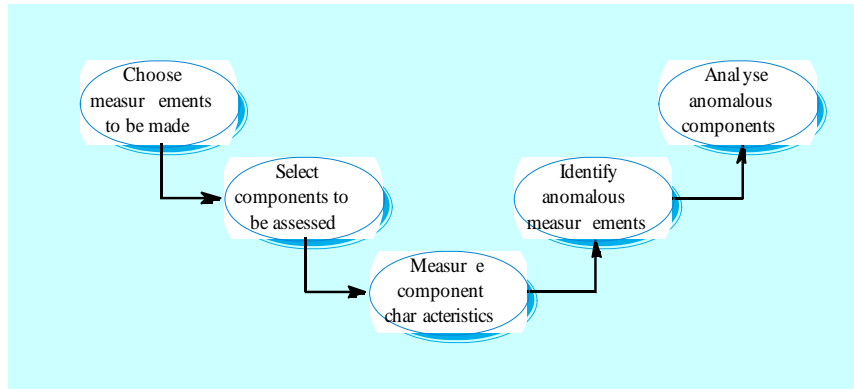
Internal and external attributes



The measurement process

- A software measurement process may be part of a quality control process.
- Data collected during this process should be maintained as an organisational resource.
- Once a measurement database has been established, comparisons across projects become possible.

Product measurement process



Data collection

- A metrics programme should be based on a set of product and process data.
- Data should be collected immediately (not in retrospect) and, if possible, automatically.
- Three types of automatic data collection
 - Static product analysis;
 - Dynamic product analysis;
 - Process data collation.

Data accuracy

- Don't collect unnecessary data
 - The questions to be answered should be decided in advance and the required data identified.
- Tell people why the data is being collected.
 - It should not be part of personnel evaluation.
- Don't rely on memory
 - Collect data when it is generated not after a project has finished.

Product metrics

- A quality metric should be a predictor of product quality.
- Classes of product metric
 - Dynamic metrics which are collected by measurements made of a program in execution;
 - Static metrics which are collected by measurements made of the system representations;
 - Dynamic metrics help assess efficiency and reliability; static metrics help assess complexity, understandability and maintainability.

Dynamic and static metrics

- Dynamic metrics are closely related to software quality attributes
 - It is relatively easy to measure the response time of a system (performance attribute) or the number of failures (reliability attribute).
- Static metrics have an indirect relationship with quality attributes
 - You need to try and derive a relationship between these metrics and properties such as complexity, understandability and maintainability.

Software product metrics

Software metric	Description
Fan in/Fan-out	Fan-in is a measure of the number of functions or methods that call some other function or method (say X). Fan-out is the number of functions that are called by function X. A high value for fan-in means that X is tightly coupled to the rest of the design and changes to X will have extensive knock-on effects. A high value for fan-out suggests that the overall complexity of X may be high because of the complexity of the control logic needed to coordinate the called components.
Length of code	This is a measure of the size of a program. Generally, the larger the size of the code of a component, the more complex and error-prone that component is likely to be. Length of code has been shown to be one of the most reliable metrics for predicting error-proneness in components.
Cyclomatic complexity	This is a measure of the control complexity of a program. This control complexity may be related to program understandability. I discuss how to compute cyclomatic complexity in Chapter 22.
Length of identifiers	This is a measure of the average length of distinct identifiers in a program. The longer the identifiers, the more likely they are to be meaningful and hence the more understandable the program.
Depth of conditional nesting	This is a measure of the depth of nesting of if-statements in a program. Deeply nested if statements are hard to understand and are potentially error-prone.
Fog index	This is a measure of the average length of words and sentences in documents. The higher the value for the Fog index, the more difficult the document is to understand.

Object-oriented metrics

Object-oriented metric	Description
Depth of inheritance tree	This represents the number of discrete levels in the inheritance tree where sub-classes inherit attributes and operations (methods) from super-classes. The deeper the inheritance tree, the more complex the design. Many different object classes may have to be understood to understand the object classes at the leaves of the tree.
Method fan-in/fan-out	This is directly related to fan-in and fan-out as described above and means essentially the same thing. However, it may be appropriate to make a distinction between calls from other methods within the object and calls from external methods.
Weighted methods per class	This is the number of methods that are included in a class weighted by the complexity of each method. Therefore, a simple method may have a complexity of 1 and a large and complex method a much higher value. The larger the value for this metric, the more complex the object class. Complex objects are more likely to be more difficult to understand. They may not be logically cohesive so cannot be reused effectively as super-classes in an inheritance tree.
Number of overriding operations	This is the number of operations in a super-class that are over-ridden in a sub-class. A high value for this metric indicates that the super-class used may not be an appropriate parent for the sub-class.

Measurement analysis

- It is not always obvious what data means
 - Analysing collected data is very difficult.
- Professional statisticians should be consulted if available.
- Data analysis must take local circumstances into account.

Measurement surprises

- Reducing the number of faults in a program leads to an increased number of help desk calls
 - The program is now thought of as more reliable and so has a wider more diverse market. The percentage of users who call the help desk may have decreased but the total may increase;
 - A more reliable system is used in a different way from a system where users work around the faults. This leads to more help desk calls.

Key points

- Software quality management is concerned with ensuring that software meets its required standards.
- Quality assurance procedures should be documented in an organisational quality manual.
- Software standards are an encapsulation of best practice.
- Reviews are the most widely used approach for assessing software quality.

Key points

- Software measurement gathers information about both the software process and the software product.
- Product quality metrics should be used to identify potentially problematical components.
- There are no standardised and universally applicable software metrics.

Normalização - Brasil

- *Programa Brasileiro de Qualidade e Produtividade (PBQP).*
 - Subprograma Setorial de Qualidade e Produtividade em Software (SSQP/SW).
- *ABNT - Subcomitê de Software (SC21:10/CB21/ABNT).*
 - CT.01 - engenharia de software.
 - CT.02 - linguagens de programação e sistemas operacionais.

Normalização - Internacional

- ISO 9000-3: qualidade do processo.
- Modelo CMM/SEI: qualidade no processo, através de níveis de maturação.
- ISO 9126: qualidade do produto final.

Validação e Verificação de Sistemas

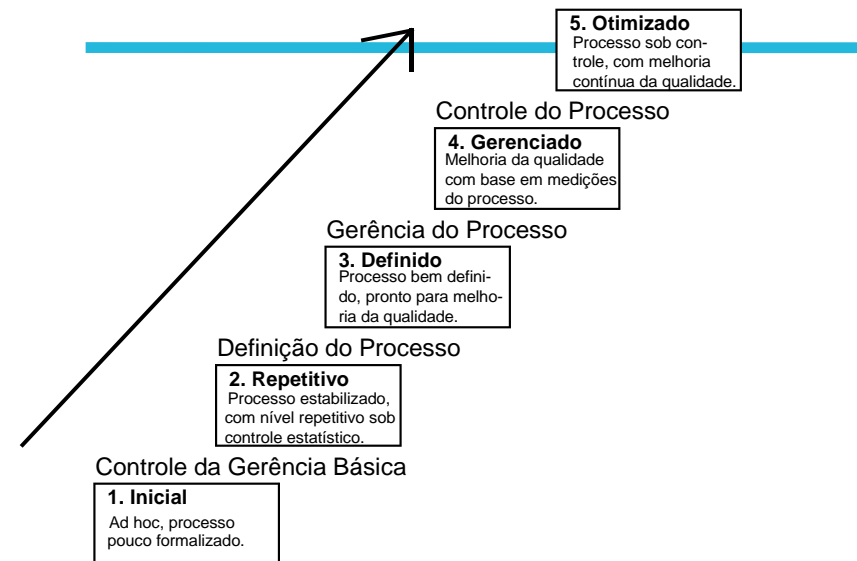
Norma ISO 9000 - 3

- Diretrizes para a aplicação da ISO 9001 em empresas de software.
- Organizações que desenvolvem, fornecem e mantêm software.
- Facilita (e padroniza) a demonstração da capacidade de um desenvolvedor.

Validação e Verificação de Sistemas

Modelo CMM/SEI

Capability Maturity Model
Software Engineering Institute



Validação e Verificação de Sistemas

Representação dos Níveis do Modelo

- Fases históricas de um processo evolucionário de aprimoramento.
- Fornecem passos alcançáveis de aprimoramento de processos.
- Sugerem objetivos e métricas para identificar o alcance dos objetivos.
- Oferecem prioridades de maturidade dos recursos de uma organização.

O que é CMM (Capability Maturity Model)

- “O CMM for Software é uma estrutura (framework) que descreve os principais elementos de um processo de software efetivo.”
- Descreve um caminho de melhoria evolutiva a partir de um processo ad hoc para um processo maduro e altamente disciplinado.

O que é CMM (Capability Maturity Model)

- Cinco níveis de maturidade: inicial, repetitivo, definido, gerenciado e em otimização
- O CMM não é um método é um modelo
- O CMM diz *o que* e não *como*
- Premissa: “a qualidade de um produto é altamente influenciada pela qualidade do processo utilizado no desenvolvimento e manutenção”

Sobre o SEI/CMU

- Responsável pelo desenvolvimento do CMM
- É um centro de pesquisa e desenvolvimento financiado pelo DoD sediado no CMU
- Tem por missão aprimorar a prática de Engenharia de Software
- Emprega cerca de 400 profissionais (empresas, governo e universidades)

Outros modelos CMM

- SW-CMM (Capability Maturity Model for Software)
- SE-CMM (Systems Engineering Capability Maturity Model)
- SA-CMM (Software Acquisition Capability Maturity Model)
- P-CMM (People Capability Maturity Model)
- IPD-CMM (Integrated Product Development Capability Maturity Model)

CMMI

- CMMI
 - O que é, evolução, situação atual
 - Modelos CMM
 - Corpos de conhecimento
 - Áreas de processo
 - Formas de Representação
 - Componentes
 - Níveis
 - Vantagens
- Processo de avaliação no Brasil
- Bibliografia

O que é CMMI (Capability Maturity Model Integration)

- “O CMMI consiste das melhores práticas que envolvem o desenvolvimento e manutenção de produtos e serviços abrangendo o ciclo de vida do produto desde sua concepção até a entrega e manutenção”
- Exemplos: aeronave, câmera digital, pacote de software, suporte técnico para produto de software, serviços de processamento de dados, etc.

Evolução do CMMI

- Desde 1991 os CMMs tem desenvolvido uma grande quantidade de disciplinas;
- Modelos: utilidade X custo;
- Solução: modelo integrador (CMMI)
- O framework CMMI foi desenvolvido pelo projeto CMMI - SEI;

Evolução do CMMI

- Missão do time de produto CMMI:
Combinar 3 modelos
 - SW-CMM (CMM for software);
 - SECM (System engineering capability model);
 - IPD-CMM (Integrated product development CMM).
- O framework CMMI foi projetado:
 - para suportar futuras integrações;
 - para ser consistente/compatível com a ISO/IEC 15504;
- Publicada a versão 0.2, 1.0 e 1.1;

Situação atual

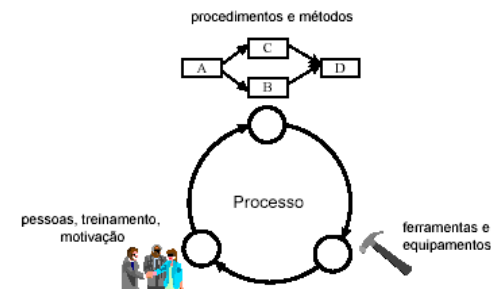
- Organizações esperam entregar produtos melhores, mais rápido e mais barato
- Terceirização
- Complexidade dos produtos
- Diferencial: software
- Diferentes modelos e padrões:
 - SW-CMM (CMM for software);
 - SECM (Systems Engineering CM), etc.

Sobre os modelos de maturidade de capacitação

- Década de 30, Walter Shewhart iniciou trabalhos sobre melhoria de processos “princípios de controle de qualidade estatístico”
- Edwards Deming e Joseph Juran refinam os princípios
- Watts Humphrey, Ron Radice estendem os princípios e aplicam em trabalhos na IBM, SEI

Sobre os modelos de maturidade de capacitação

- Focos de uma Organização



Sobre os modelos de maturidade de capacitação

- Premissa do SEI: “a qualidade de um produto é altamente influenciada pela qualidade do processo utilizado no desenvolvimento e manutenção”
- Os grandes movimentos de qualidade (ISO/IEC) também partem dessa premissa
- Os CMMs focam na melhoria de processos de uma Organização

Corpos de conhecimentos

- Atualmente são 4 os corpos de conhecimento:
 - Engenharia de sistemas;
 - Engenharia de software;
 - Desenvolvimento integrado de produto e processo;
 - Aquisição de fornecedores;
- Para o CMMI, estes *corpos de conhecimento* são referenciados como *disciplinas*.

Engenharia de sistemas

- Cobre o desenvolvimento total do sistema
- Foca na transformação das necessidades, expectativas e restrições dos clientes em produtos e suporte a esses produtos durante as suas vidas

Engenharia de software

- Cobre o desenvolvimento de sistemas de software
- Foca na aplicação sistemática e disciplinada de métodos de desenvolvimento, operação e manutenção de software

Desenvolvimento integrado de produto e processo (IPPD)

- É um método sistemático que alcança uma colaboração adequada dos relevantes *stakeholders* durante a vida do produto
- É integrado com outros processos da Organização

Aquisição de fornecedores

- Permite que os Projetos utilizem fornecedores
- Permite o monitoramento do fornecedor

Selecionando disciplinas

- As disciplinas são endereçadas através de :
 - áreas de processo associadas;
 - modelos de componentes (amplificações de disciplinas).
- Área de processo: é um agrupamento de melhores práticas
- As melhores práticas quando implementadas coletivamente, satisfazem um conjunto de objetivos

Áreas de processo para engenharia de sistemas

1. Análise causal e resolução (CAR – causal analysis and resolution)
2. Gerenciamento de configuração (CM – configuration management)
3. Análise de decisão e resolução (DAR – decision analysis and resolution)
4. Gerenciamento de projeto integrado (os primeiros dois objetivos específicos – IPM – integrated project management)
5. Medição e análise (MA – measurement and analysis)
6. Preparação e inovação Organizacional (OID – organizational innovation and deployment)
7. Definição do processo Organizacional (OPD – organizational process definition)
8. Foco no processo Organizacional (OPF – organizational process focus)
9. Desempenho de processo Organizacional (OPP – organizational process performance)
10. Treinamento Organizacional (OT – organizational training)

Áreas de processo para engenharia de software

1. Integração de produto (PI – product integration)
2. Controle e monitoria de projeto (PMC – project monitoring and control)
3. Planejamento do projeto (PP – project planning)
4. Garantia da qualidade do produto e processo (PPQA – process and product quality assurance)
5. Gerenciamento quantitativo do projeto (QPM – quantitative project management)
6. Desenvolvimento de requisitos (RD – requirements development)
7. Gerenciamento de requisitos (REQM – requirements management)
8. Gerenciamento de riscos (RSKM – risk management)
9. Gerenciamento de contrato de fornecedores (SAM – Supplier agreement management)
10. Solução técnica (TS – technical solution)
11. Validação (VAL - validation)
12. Verificação (VER - verification)

Áreas de processo para engenharia de software

- São as mesmas listadas para a engenharia de sistemas
- Possuem amplificações

Áreas de processo para desenvolvimento integrado de produto e processo

- São as mesmas listadas para a engenharia de sistemas, com duas áreas de processo adicionais:
 - Times integrados (IT – integrated teaming);
 - Ambiente Organizacional para integração (OEI – organizational environment for integration).
- Possuem amplificações

Áreas de processo para aquisição de fornecedores

- São as mesmas listadas para a engenharia de sistemas, com uma área de processo adicional:
 - Gerenciamento de fornecedor integrado (ISM integrated supplier management).
- Possuem amplificações.

Diferentes métodos de CMMI

- No CMMI, existem dois tipos de métodos, chamados de representações: por estágios (*staged*) e contínua (*continuous*)
- Uma representação reflete a Organização, o uso e apresentação de componentes em um modelo

Representação por estágios

- Utilizado no software CMM
- Utiliza um conjunto pré-definido de áreas de processo para definir um caminho de melhoria para uma Organização
- Esse caminho de melhoria é descrito por um modelo de componente chamado *nível de maturidade*

Representação contínua

- Utilizado no SECM e IPD-CMM
- Permite que uma Organização selecione uma área de processo específica e realize melhorias com relação a essa área
- Utiliza *níveis de capacidade* para caracterizar melhorias relativas a uma área de processo individual

Componentes da área de processo

- São agrupados em três categorias:
 - Componentes obrigatórios (required components);
 - Componentes esperados (expected components);
 - Componentes informativos (informative components).

Componentes obrigatórios

- Descrevem o que uma Organização deve alcançar para satisfazer a área de processo
- Correspondem aos objetivos específicos e genéricos (specific and generic goals)
- O atendimento a um objetivo é utilizado em avaliações

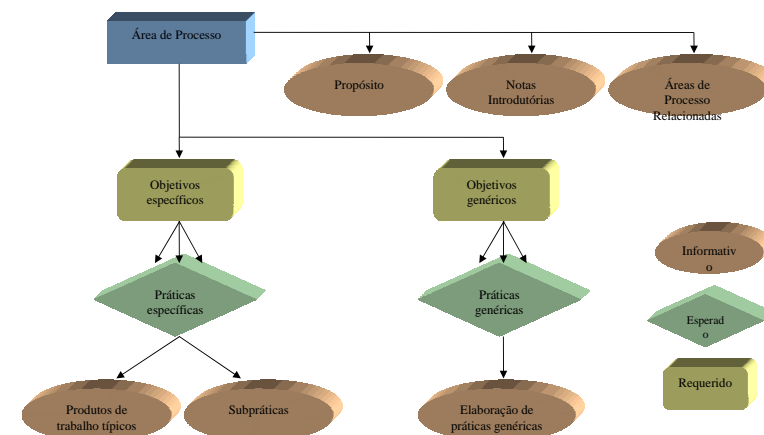
Componentes esperados

- Descrevem o que uma Organização irá implementar para atender um componente obrigatório
- Correspondem às práticas específicas e genéricas (specific and generic practices)
- Guiam aqueles que implementam melhorias ou executam avaliações

Componentes informativos

- Provêm detalhes que ajudam a utilização dos componentes esperados e obrigatórios
- Correspondem às Subpráticas, produtos de trabalho, amplificações de disciplinas, elaborações de práticas genéricas, títulos de objetivos e práticas, notas de objetivos e práticas e referências

Componentes da área de processo



Entendimento dos níveis

“Os níveis são utilizados no CMMI para descrever o caminho evolucionário recomendado para a Organização que espera melhorar os processos e seu uso para o desenvolvimento e manutenção de produtos e serviços”

Entendimento dos níveis

- Para a representação contínua, utiliza-se o termo *nível de capacidade*
- Para a representação por estágios, utiliza-se o termo *nível de maturidade*
- Ambas as representações providenciam maneiras de implementar melhorias de processo para atingir os objetivos de negócio

Níveis de capacidade X maturidade

Nível	CMM	Represent. contínua Níveis de capacidade	Represent. por estágios Níveis de maturidade
0	-	Incompleto	
1	Inicial	Executado	Inicial
2	Repetitivo	Gerenciado	Gerenciado
3	Definido	Definido	Definido
4	Gerenciado	Quantitativamente gerenciado	Quantitativamente gerenciado
5	Em Otimização	Otimizado	Otimizado

Nível de capacidade 0: Incompleto

- Um processo incompleto é um processo que não é executado ou é parcialmente executado
- Um ou mais dos objetivos específicos da área de processo não foram satisfeitos

Nível de capacidade 1: Executado

- Um processo executado é um processo que satisfaz os objetivos específicos para a área de processo

Nível de capacidade 2: Gerenciado

- Um processo gerenciado é um processo executado que tem a infra-estrutura básica para suportar o processo
- É executado e planejado de acordo com as políticas
- Emprega pessoas especialistas que possuem recursos adequados para produzir saídas controladas

Nível de capacidade 2: Gerenciado

- Envolvem os relevantes stakeholders
- É monitorado, controlado e revisado
- É avaliado pela aderência com a descrição do processo

Nível de capacidade 3: Definido

- Um processo definido é um processo gerenciado que é adaptado a partir de um conjunto de processos padrão da Organização
- Para a adaptação segue as diretrizes de adaptação da Organização

Nível de capacidade 4: Quantitativamente gerenciado

- Um processo gerenciado quantitativamente é um processo definido que é controlado utilizando técnicas estatísticas ou quantitativas
- A qualidade e desempenho do processo são entendidos em termos estatísticos e gerenciados através da vida do processo

Nível de capacidade 5: Otimizado

- Um processo otimizado é um processo gerenciado quantitativamente que é melhorado baseado em um entendimento de causas comuns de variação pertencentes ao processo
- O foco de um processo otimizado é a melhoria contínua do desempenho do processo através de melhorias incrementais e inovadoras

Nível de maturidade 1: Inicial

- Um processo de nível 1 é geralmente ad hoc e caótico
- A Organização normalmente não provê um ambiente estável que suporte os processos
- O sucesso nessas Organizações depende da competência e heroísmo das pessoas e não de processos experimentados

Nível de maturidade 1: Inicial

- As Organizações freqüentemente produzem produtos que funcionam, entretanto, eles freqüentemente excedem os prazos e orçamentos
- Esse nível ainda se caracteriza pela tendência de excesso de comprometimentos, abandono do processo durante as crises e uma incapacidade de repetir os sucessos

Nível de maturidade 2: Gerenciado

- Os projetos da Organização tem assegurado que as requisições são gerenciadas e os processos são planejados, executados, medidos e controlados
- O *status* dos produtos de trabalho e a entrega de serviços são visíveis para o gerenciamento em pontos definidos

Nível de maturidade 2: Gerenciado

- Comprometimentos são estabelecidos entre os relevantes *stakeholders* e são revisados quando necessário
- Os produtos de trabalho e serviços satisfazem as suas especificações de processos, padrões e procedimentos

Nível de maturidade 3: Definido

- Os processos são bem caracterizados e entendidos e são descritos em padrões, procedimentos, ferramentas e métodos
- O conjunto de processos padrão da Organização é estabelecido e melhorado periodicamente
- Esses processos padrão são utilizados para estabelecer uma consistência através da Organização

Nível de maturidade 4: Quantitativamente gerenciado

- A Organização e projetos estabelecem objetivos quantitativos para a qualidade e desempenho do processo e as utiliza como critério para gerenciamento de processos
- Objetivos quantitativos são baseados nas necessidades dos clientes, usuários finais, Organização e implementadores do processo

Nível de maturidade 4: Quantitativamente gerenciado

- A qualidade e desempenho do processo são entendidos em termos estatísticos e são gerenciados durante a vida dos processos
- No nível 3, os processos são somente previsíveis qualitativamente
- No nível 4, o desempenho do processo é previsível quantitativamente

Nível de maturidade 5: Otimizado

- A Organização melhora continuamente seus processos baseado em um entendimento quantitativo de causas comuns de variação pertencentes ao processo
- Os objetivos da melhoria de processo quantitativa para a Organização são estabelecidas e continuamente revisadas para refletir as mudanças dos objetivos de negócio

Vantagens do CMMI

- É a próxima geração de melhoria de processos
- Provê uma cobertura mais detalhada do ciclo de vida dos produtos
- Os produtos CMMI incorporaram mais lições aprendidas

Vantagens do CMMI

- Integra a engenharia de software e engenharia de sistemas na engenharia de produtos;
- Cada modelo CMMI suporta ambas as representações (por estágios e contínua)
- Foco inicial em produtos e serviços de engenharia mas flexível para outras disciplinas

O Processo de Avaliação no Brasil

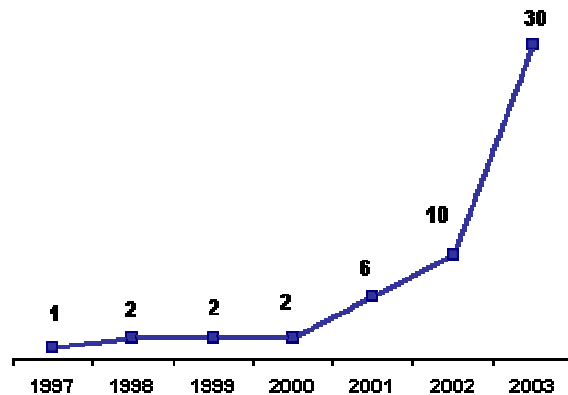
Relatório do SEI/CMU (set/2003) com dados até junho (2003)

- O Brasil permanece em 13º lugar (países com maior número de avaliações)
- Integrated System Diagnostics Brasil (ISD-Brasil): credenciais do SEI para efetuar avaliações CMM
- JDFurlan & Associados (representa a Procesix): credenciais do SEI para efetuar avaliações CMM e CMMI

O Processo de Avaliação no Brasil

Desde	N2	N3	N4	No ano	Até o ano
1997		1		1	2
1998	1			1	2
1999					2
2000					2
2001	1	3		4	6
2002	4			4	10
2003	18	1	1	20	30

O Processo de Avaliação no Brasil



Bibliografia

- SEI - www.sei.cmu.edu
- [http://www.mct.gov.br/temas/info/Dsi/qualidad/CM M.htm](http://www.mct.gov.br/temas/info/Dsi/qualidad/CM%20M.htm)
- Fiorini, Soeli T. Staa, Arndt Von. Baptista, Renan Martins. **Engenharia de Software com CMM**. Brasport, 2002.
- **Capability Maturity Model Integration (CMMI)**. Version 1.1. Carnegie Mellon, Software Engineering Institute, march 2002.



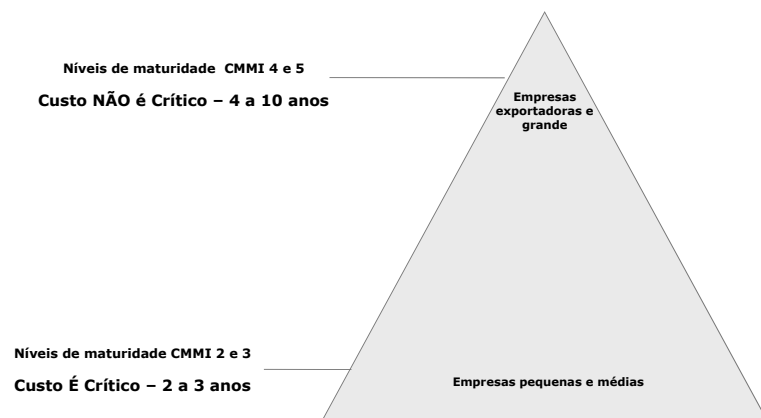
Melhoria de Processo do Software Brasileiro

Roteiro

- Motivação
- Organização do Projeto
- Estrutura do Modelo

©Pimenta 2010

Motivação



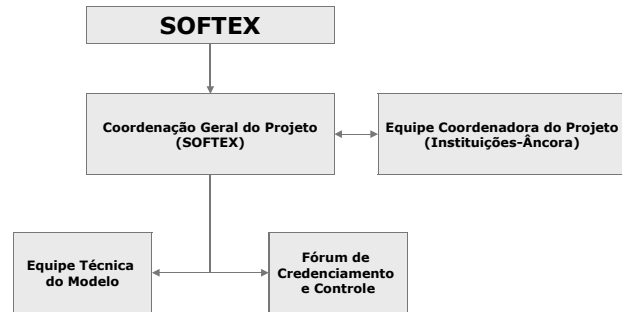
©Pimenta 2010

Motivação

*“Melhoria de **processos** de software nas **micro, pequenas e médias** empresas, a um custo acessível, em diversos locais do país.”*

©Pimenta 2010

Organização do Projeto



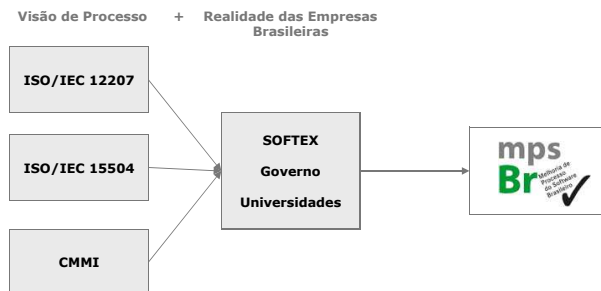
©Pimenta 2010

Organização do Projeto

- SOTEX
- Equipe Coordenadora do Projeto
 - Sociedade SOTEX
 - Instituições de Ensino, Pesquisa e Centros Tecnológicos (COPPE, CenPRA, CESAR, ...)
 - Sociedade de Economia Mista (CELEPAR)
 - Organizações da Sociedade Civil de Interesse Público (RIOSOFIT, Agente SOTEX Campinas)
- Equipe Técnica do Modelo (Ana Regina Rocha, Ana Cristina Rouiller, ...)
- Fórum de Credenciamento e Controle
 - Representante da Sociedade SOTEX
 - Representante das Instituições de Ensino, Pesquisa e Centros Tecnológicos
 - Representante do Governo

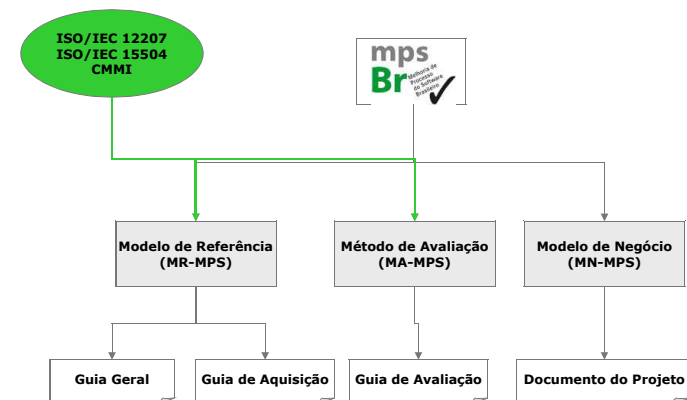
©Pimenta 2010

Estrutura do Modelo



©Pimenta 2010

Estrutura do Modelo



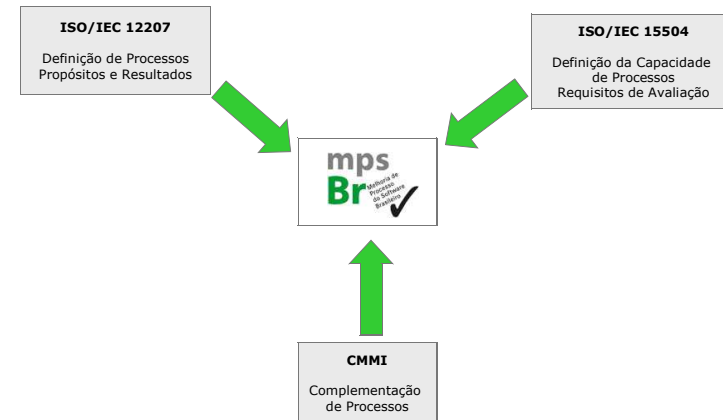
©Pimenta 2010

Estrutura do Modelo

- ISO/IEC 12207 (Processos de Ciclo de Vida de Software)
- ISO/IEC 15504 (Framework para Avaliação (e Melhoria) de Processo)
- CMMI (Modelo para Melhoria de Processos de Software)

©Pimenta 2010

Estrutura do Modelo Visão de Processo



©Pimenta 2010

Guia Geral

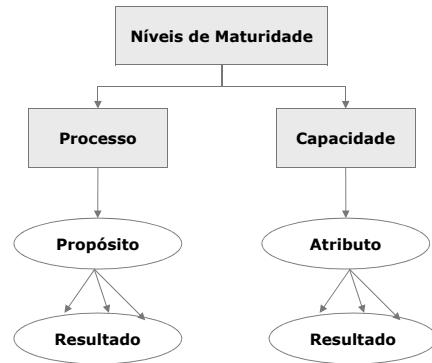
- Estrutura do MR-MPS
- Definições
- Níveis de Maturidade
- Níveis de Capacidade
- Processos

Estrutura do Modelo Guia Geral

- Objetivo
 - Descreve o Modelo de Referência para Melhoria do Processo de Software (MR-MPS) e fornece uma visão geral sobre os demais guias que apóiam os processos de avaliação e de aquisição.
- Público alvo
 - Instituições interessadas em aplicar o MR-MPS para melhoria de seus processos de software.
 - Instituições implementadoras e avaliadoras segundo o MR-MPS.
- Referências
 - **Básicas** -> ISO/IEC 12207:1995/Amd 2:2002 e ISO/IEC 15504
 - **Complementar** -> CMMI

©Pimenta 2010

Estrutura do MR-MPS



©Pimenta 2010

Definições

- **Nível de Maturidade**
 - “Grau de melhoria de processo para um pré-determinado conjunto de processos no qual todos os objetivos dentro do conjunto são atendidos”
- **Capacidade do Processo**
 - “Uma caracterização da habilidade do processo atingir os objetivos de negócio atuais ou futuros”
- **Processo**
 - “Um conjunto de atividade inter-relacionadas que transforma entradas em saídas”
- **Atributo de Processo**
 - “Uma característica mensurável da capacidade do processo aplicável a qualquer processo”
- **Propósito do Processo**
 - “O principal objetivo da execução do processo e os prováveis resultados obtidos com a efetiva implementação do mesmo. Convém que a implementação do processo forneça benefícios tangíveis aos envolvidos”
- **Resultado Esperado do Processo**
 - “Um resultado observável do sucesso do alcance do propósito do processo”

©Pimenta 2010

Níveis de Maturidade

	Nível	Processo	Capacidade
Em Otimização	A (mais alto)	Inovação e Implantação na Organização	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
		Análise e Resolução de Causas	
Gerenciado Quantitativamente	B	Desempenho do Processo Organizacional	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
		Gerência Quantitativa do Projeto	
Definido	C	Análise de Decisão e Resolução	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
		Gerência de Riscos	
Largamente Definido	D	Desenvolvimento de Requisitos	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
		Solução Técnica	
		Integração do Produto	
		Instalação do Produto	
		Liberação do Produto	
		Verificação	
		Validação	
Parcialmente Definido	E	Treinamento	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
		Avaliação e Melhoria do Processo Organizacional	
		Definição do Processo Organizacional	
		Adaptação do Processo para Gerência de Projeto	
Gerenciado	F	Medição	AP 1.1, AP 2.1 e AP 2.2
		Gerência de Configuração	
		Aquisição	
		Garantia da Qualidade	
Parcialmente Gerenciado	G	Gerência de Requisitos	AP 1.1 e AP 2.1
		Gerência de Projeto	

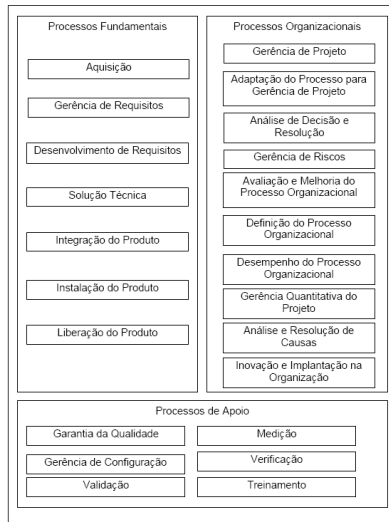
©Pimenta 2010

Níveis de Capacidade

- **Atributos de Processo**
 - **AP 1.1 O processo é executado**
 - O processo atinge seu propósito
 - **AP 2.1 O processo é gerenciado**
 - O atributo de gerência de execução é uma medida da extensão na qual a execução do processo é gerenciada
 - **AP 2.2 Os produtos de trabalho do processo são gerenciados**
 - Extensão na qual os produtos de trabalho produzidos pelo processo são gerenciados apropriadamente
 - **AP 3.1 O processo é definido**
 - Um processo-padrão é mantido para apoiar a implementação do processo definido
 - **AP 3.2 O processo está implementado**
 - O processo-padrão é efetivamente implementado como um processo definido para atingir seus resultados

©Pimenta 2010

Processos



©Pimenta 2010

Aquisição

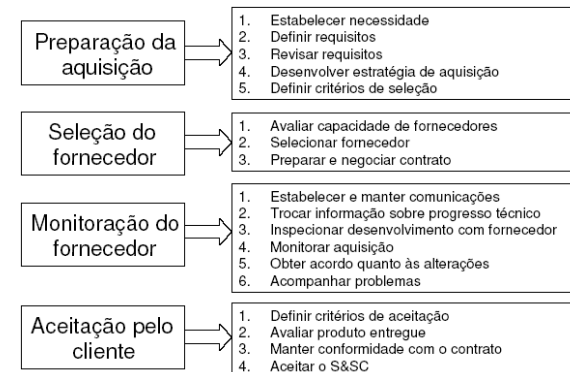
- Guia de Aquisição
- Processo de Aquisição

Estrutura do Modelo Guia de Aquisição

- **Objetivo**
 - Descrever um processo de aquisição de software.
- **Público alvo**
 - Organizações públicas e privadas que necessitem adquirir software -> **Guia**
 - Produtores de software que queiram estar preparados para este processo de aquisição -> **Orientação**
- **Referências**
 - **Básica** -> ISO/IEC 12207:1995/Amd 1:2002
 - **Complementar** -> IEEE STD 1062:1998

©Pimenta 2010

Processos de Aquisição



©Pimenta 2010

Avaliação

- Guia de Avaliação
- Equipe
- Passos da Avaliação

Estrutura do Modelo Guia de Avaliação

- Objetivo
 - Orientar a realização de avaliações, em conformidade com a norma ISO/IEC 15504, em empresas e organizações que implementaram o MR-MPS.
- Público alvo
 - Empresas e organizações que queiram ser avaliadas segundo o MA-MPS
 - Instituições Avaliadoras do Modelo (IA-MPS)
 - Instituições Implementadoras do Modelo (II-MPS)
- Referências
 - **Básica** -> ISO/IEC 15504 *Information Technology – Process Assessment*
 - **Complementar** -> SCAMPI – *Standard CMMI Appraisal Method for Process Improvement*

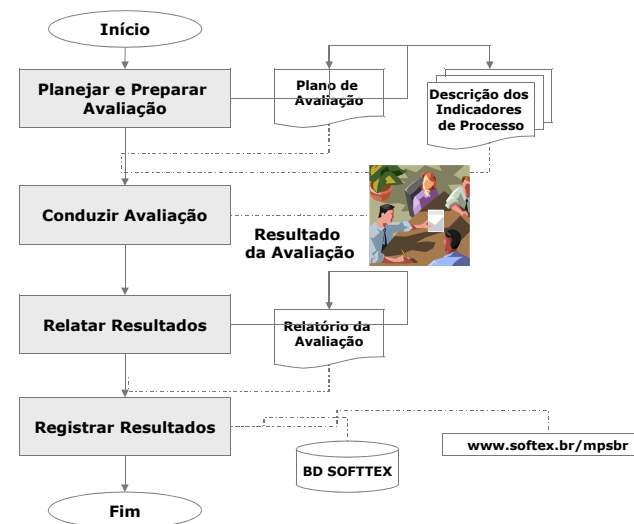
©Pimenta 2010

Equipe

- Equipe de Avaliação (mínimo de 3, dependendo do nível)
 - 1 líder da avaliação
 - 1 avaliador adjunto
 - No mínimo, 1 técnico da empresa
- Validade de 2 anos
 - Avaliação para outro nível
 - Avaliação para manter nível

©Pimenta 2010

Passos da Avaliação



©Pimenta 2010

Planejar a Avaliação

- Definir objetivos da avaliação
- Definir escopo da avaliação
- Definir patrocinador (\$)
- Definir escopo organizacional da avaliação – unidade organizacional
- Definir cronograma da avaliação
- Selecionar projetos para a avaliação
 - Pelo menos 2 projetos concluídos
 - 2 projetos em andamento
- Definir participantes da avaliação
- Definir a equipe de avaliação
- Obter acordo sobre o plano (empresa <-> avaliador)

©Pimenta 2010

Preparar a Avaliação

- Coleta de dados para a avaliação
 - Preenchimento de planilha (resultado esperado/evidências)

©Pimenta 2010

Execução da Avaliação

- Análise dos dados prévios coletados
- Entrevistas
- Verificação dos dados coletados
- Atribuição de Nível de Maturidade MR-MPS

©Pimenta 2010

Verificação dos Dados

- Avaliação feita através de indicadores
 - Diretos – produtos intermediários
 - Indiretos – documentos que indicam que a atividade foi realizada
 - Afirmação – resultantes de entrevistas

©Pimenta 2010

Decisão

Grau de Implementação	Caracterização	Grau de alcance
Totalmente Implementado	<ul style="list-style-type: none"> O indicador direto está presente e julgado adequado Existe pelo menos um indicador indireto e/ou afirmação para confirmar a implementação Não foi notada nenhuma fraqueza substancial 	> 85% a 100%
Largamente Implementado	<ul style="list-style-type: none"> O indicador direto está presente e julgado adequado Existe pelo menos um indicador indireto e/ou afirmação para confirmar a implementação Foi notada uma ou mais fraquezas 	> 50% a 85%
Parcialmente Implementado	<ul style="list-style-type: none"> O indicador direto não está presente ou é julgado inadequado Artefatos ou afirmações sugerem que alguns aspectos da prática estão implementados Fraquezas foram documentadas 	> 15% a 50%
Não Implementado	Qualquer situação diferente das acima	0 a 15%

©Pimenta 2010

Relatar e Registrar Resultado

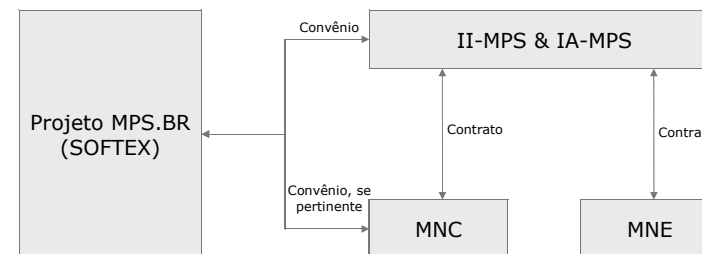
- Geração de Relatório Final da Avaliação
 - Informações sobre a empresa e a organização avaliada
 - Objetivo da avaliação
 - Projetos avaliados
 - Participantes da avaliação (equipe de avaliação e entrevistados)
 - Resultados por processo avaliado
 - Nível de Maturidade da organização
 - Acordo dos representantes da organização e equipe de avaliação sobre o conteúdo do Relatório Final
 - Entrega do Relatório Final para o patrocinador da avaliação, que é responsável por manter o documento
- Envio do Relatório à Sociedade SOFTEX
- Registro no Banco de Dados SOFTEX
- Divulgação no site www.softex.br/mpsbr

©Pimenta 2010

Modelo de Negócio

- Modelo de Negócio

Modelo de Negócio



II-MPS – Instituição Implementadora do Modelo MPS.BR
IA-MPS – Instituição Avaliadora do Modelo MPS.BR
MNE – Modelo de Negócio Específico para cada empresa (personalizado)
MNC – Modelo de Negócio em Grupo de empresas (pacote)

©Pimenta 2010

Capacitação



©Pimenta 2010

Capacitação Avaliadores

- Curso e Prova de Introdução ao MPS.BR
- Experiência Profissional
- Formação Acadêmica Sólida
- Curso e Prova de Método de Avaliação MA-MPS
- Experiência em 2 avaliações como **Adjunto** (1G/F e uma acima E)
- Observação em 2 avaliações como **Líder**

©Pimenta 2010

Os 7 Diferenciais do MR-MPS

- 7 níveis de maturidade (possibilitam uma implantação mais gradual e adequada à micro, pequena e média empresa; além disto, as avaliações considerando mais níveis permitem uma maior visibilidade dos resultados de melhoria de processo, com prazos mais curtos)
- Compatibilidade com CMMI
- ISO/IEC 15504/12207 e CMMI (2 em 1)
- Criado para a realidade da empresa brasileira (foco na micro, pequena e média empresa de software)
- Custo acessível (em R\$)
- Avaliação periódica das empresas (de 2 em 2 anos)
- Forte interação Universidade-Empresa (catalisador do desenvolvimento tecnológico e de negócios)

©Pimenta 2010

Referências

- Site oficial do MPS.BR (www.softex.br/mpsbr)
 - Guia Geral do Modelo MPS.BR
 - Guia de Aquisição
 - Guia de Avaliação
 - Guia de Implementação

©Pimenta 2010