

Engenharia de Software – ESOF

Processos de software

Prof. Leonardo Vieira Barcelos

O que é?

- Quando você elabora um produto ou sistema é importante percorrer uma série de passos previsíveis – um roteiro que o ajuda a criar a tempo um resultado de alta qualidade. O roteiro que você segue é chamado de processo de software.

Quem faz?

- Os engenheiros de software adaptam o processo a suas necessidades e depois o seguem.
- Além disso, o pessoal que solicitou tem um papel a desempenhar no processo de defini-lo, construí-lo e testa-lo.

Por que é importante?

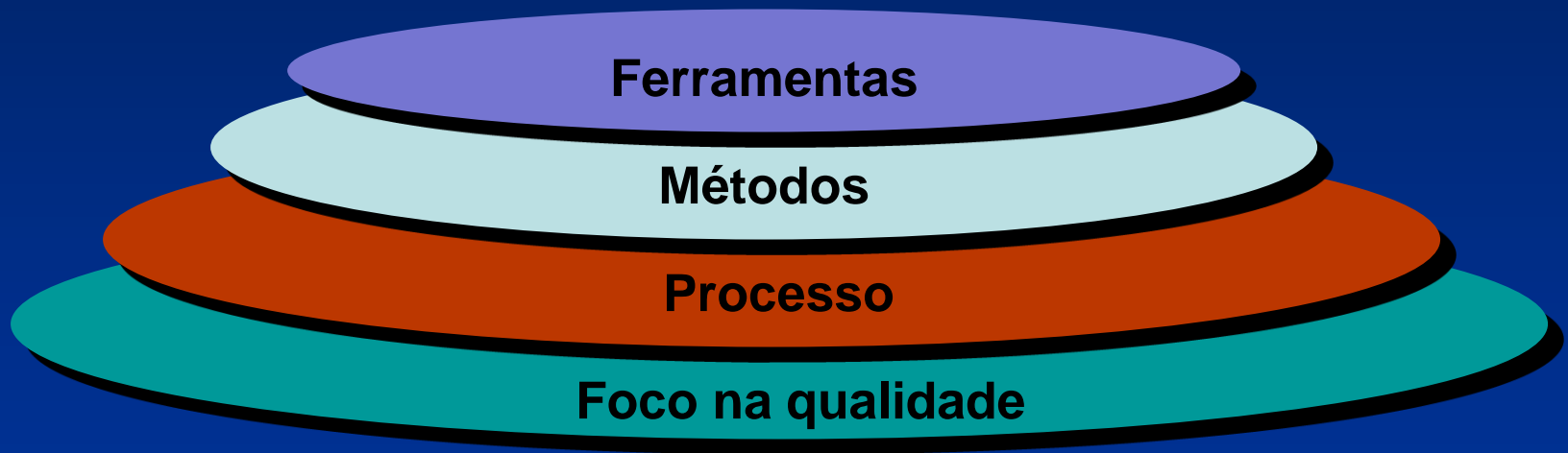
- Porque fornece estabilidade, controle e organização para uma atividade que pode, se deixada sem controle, tornar-se caótica. No entanto, uma abordagem moderna de engenharia de software precisa ser “ágil”.

“Um processo define quem está fazendo o quê, quando e como para alcançar um certo objetivo”.

Ivar Jacobson, Grad Booch e James Rumbaugh

O Processo de desenvolvimento de Software

Abrange um conjunto de quatro elementos fundamentais:
Ferramentas, Métodos, Processo e Foco na qualidade para projetar, construir e manter grandes sistemas de software de forma profissional



O Processo de desenvolvimento de Software

- **MÉTODOS**: fornecem a técnica de “como fazer” para construir softwares. Abrange um amplo conjunto de tarefas que incluem:
 - Comunicação
 - Análise de requisitos
 - Modelagem de projeto
 - Construção de programas
 - Testes
 - Manutenção

O Processo de desenvolvimento de Software

- **FERRAMENTAS**: fornece apoio automatizado para os métodos.
 - Existem atualmente ferramentas para sustentar os métodos, denominadas ferramentas *CASE* (*Computer Aided Software Engineering* - engenharia de software apoiada por computador).

O Processo de desenvolvimento de Software

- **PROCESSOS:** constituem o elo de ligação entre os métodos e ferramentas.
 - É a base para o controle gerencial de projetos de software
 - Estabelece o contexto em que os métodos serão aplicados
 - Os produtos de trabalho (modelos, documentos, dados, relatórios etc.) são produzidos
 - Controles que ajudam assegurar a qualidade e coordenar as alterações
 - Marcos de referência que possibilitam administrar o progresso do software.

O Processo de desenvolvimento de Software

- **Foco na qualidade:** qualquer abordagem de engenharia deve se apoiar num compromisso organizacional com a qualidade.
 - *A Qualidade do Processo de Desenvolvimento de Software* está relacionada à extensão na qual um processo de desenvolvimento de software específico é *eficiente* e é explicitamente *definido, gerenciado, medido e controlado*.

Estrutura comum de processo

Atividades guarda-chuva

Atividades de estrutura 1

Ação de engenharia de software 1.1

Conjunto de
tarefas

tarefas de trabalho
produtos de trabalho
pontos de garantia de qualidade
marcos de projeto

Ação de engenharia de software 1.2

Conjunto de
tarefas

tarefas de trabalho
produtos de trabalho
pontos de garantia de qualidade
marcos de projeto

Atividades de estrutura 2

Ação de engenharia de software 2.1

Conjunto de
tarefas

tarefas de trabalho
produtos de trabalho
pontos de garantia de qualidade
marcos de projeto

Ação de engenharia de software 2.2

Conjunto de
tarefas


tarefas de trabalho
produtos de trabalho
pontos de garantia de qualidade
marcos de projeto

O processo de software

- Uma **estrutura comum de processo** estabelece o alicerce para um processo de software completo pela identificação de um pequeno número de **atividades de estrutura** aplicáveis a todos os projetos de software, independente de seu tamanho ou complexidade.
- Cada atividade de estrutura é preenchida por um conjunto de *ações de engenharia de software* – uma coleção de tarefas relacionadas que produz um produto importante do trabalho de engenharia de software.
 - Por exemplo, levantamento de requisitos é uma ação de engenharia de software.
- Cada ação é preenchida por tarefas de trabalho individuais que realizam alguma parte do trabalho determinado pela ação.

Atividades de estrutura

Estas atividades de processo é aplicável à grande maioria dos projetos de software.



- Comunicação
- Planejamento
- Modelagem
 - Análise de requisitos
 - Modelos
- Construção
 - Geração de códigos
 - Testes
- Implantação

Atividades de estrutura

- **Comunicação:** Essa atividade de estrutura envolve alta comunicação e colaboração com o cliente e abrange o levantamento de requisitos e outras atividades relacionada.
- **Planejamento:** estabelece um plano de trabalho de engenharia de software que se segue. Descreve as tarefas técnicas a ser conduzidas, os riscos prováveis, os recursos que serão necessários, os produtos de trabalho a ser produzidos e um cronograma de trabalho.
- **Modelagem:** Essa atividade inclui a criação de modelos que permitam ao desenvolvedor e ao cliente, entender melhor os requisitos do software e o projeto que vai satisfazer a esses requisitos.
- **Construção:** geração de código (quer manual ou automática) e os testes necessários para revelar erros no código.
- **Implementação:** o software é entregue ao cliente, que avalia o produto entregue e fornece feedback com base na avaliação.

Exemplo – Conjunto de tarefas

- Um conjunto de tarefas define o trabalho real a ser feito para atingir os objetivos de uma ação de engenharia de software. Por exemplo, “levantamento de requisitos” é uma importante ação de engenharia de software que ocorre durante a atividade de **comunicação**. O objetivo do levantamento de requisitos é entender o que vários interessados desejam do software que está sendo construído.
- Para um projeto pequeno e relativamente simples, o conjunto de tarefas para o levantamento de requisitos pode ter o seguinte aspecto:

1. Comunicação

1.1 Levantamento de requisitos

1. Faça uma lista de interessados no processo.
2. Convide todos os interessados para uma reunião informal.
3. Peça a cada interessado para fazer uma lista das características e funções desejadas.
4. Discuta os requisitos e construa uma lista final.
5. Priorize os requisitos.
6. Observe áreas de incerteza.

Exemplo – Conjunto de tarefas

- Projeto de software maior e mais complexo, um conjunto de tarefas diferente seria necessário. Ele pode incluir as seguintes tarefas de trabalho

1. Comunicação

1.1 Levantamento de requisitos

1. Faça uma lista de interessados no processo.
2. Entreviste cada interessado separadamente para determinar seus desejos e necessidades gerais.
3. Construa uma lista preliminar de funções e características com base nas informações fornecidas pelos interessados.
4. Programe uma série de reuniões facilitadas de levantamento de requisitos.
5. Realize as reuniões.
6. Produza cenários informais de usuários como parte de cada reunião.
7. Refina os cenários de usuários com base no feedback dos interessados.
8. Construa uma lista revisada de requisitos dos interessados.
9. Use técnicas de implantação de função de qualidade para priorizar os requisitos.
10. Empacote os requisitos de modo que eles possam ser entregues incrementalmente.
11. Anotes as restrições e limitações que serão colocadas no sistema.
12. Discuta métodos de validação do sistema.

Atividades guarda-chuva

- A estrutura de processo engloba um conjunto de **atividades guarda-chuva** que são aplicáveis durante todo o processo de software.
- **Acompanhamento e controle de projeto de software:**
 - permite a equipe de software avaliar o progresso com base no plano de projeto e tomar a ação necessária para manter o cronograma.
- **Gestão de risco:**
 - avaliar os riscos que podem afetar o resultado do projeto ou a qualidade do produto.
- **Garantia de qualidade de software:**
 - define e conduz as atividades necessárias para garantir a qualidade de software.
- **Revisões técnicas formais:**
 - avaliam os produtos de trabalho de engenharia de software num esforço para descobrir e remover erros antes que eles seja propagados para a próxima ação ou atividade.
- **Medição:**
 - define e reúne medidas de processo, projeto e produto que ajudam a equipe a entregar um software que satisfaça às necessidades do usuário.
- **Gestão de configuração de software:**
 - gerencia os efeitos das modificações ao longo de todo o processo de software.
- **Gestão de reusabilidade:**
 - define critérios para a reutilização dos produtos de trabalho (inclusive componentes de software).
- **Preparação e produção do produto do trabalho:**
 - abrange as atividades necessárias para criar produtos do trabalho como modelos, documentos, registros, formulários e listas.

Sintomas de Falha de Processo

- Atrasos na entrega
- Custos crescentes
- Problemas de qualidade
 - Funções não funcionam corretamente
 - Reclamações de clientes
 - O trabalho tem que ser refeito
- Funcionários frustrados

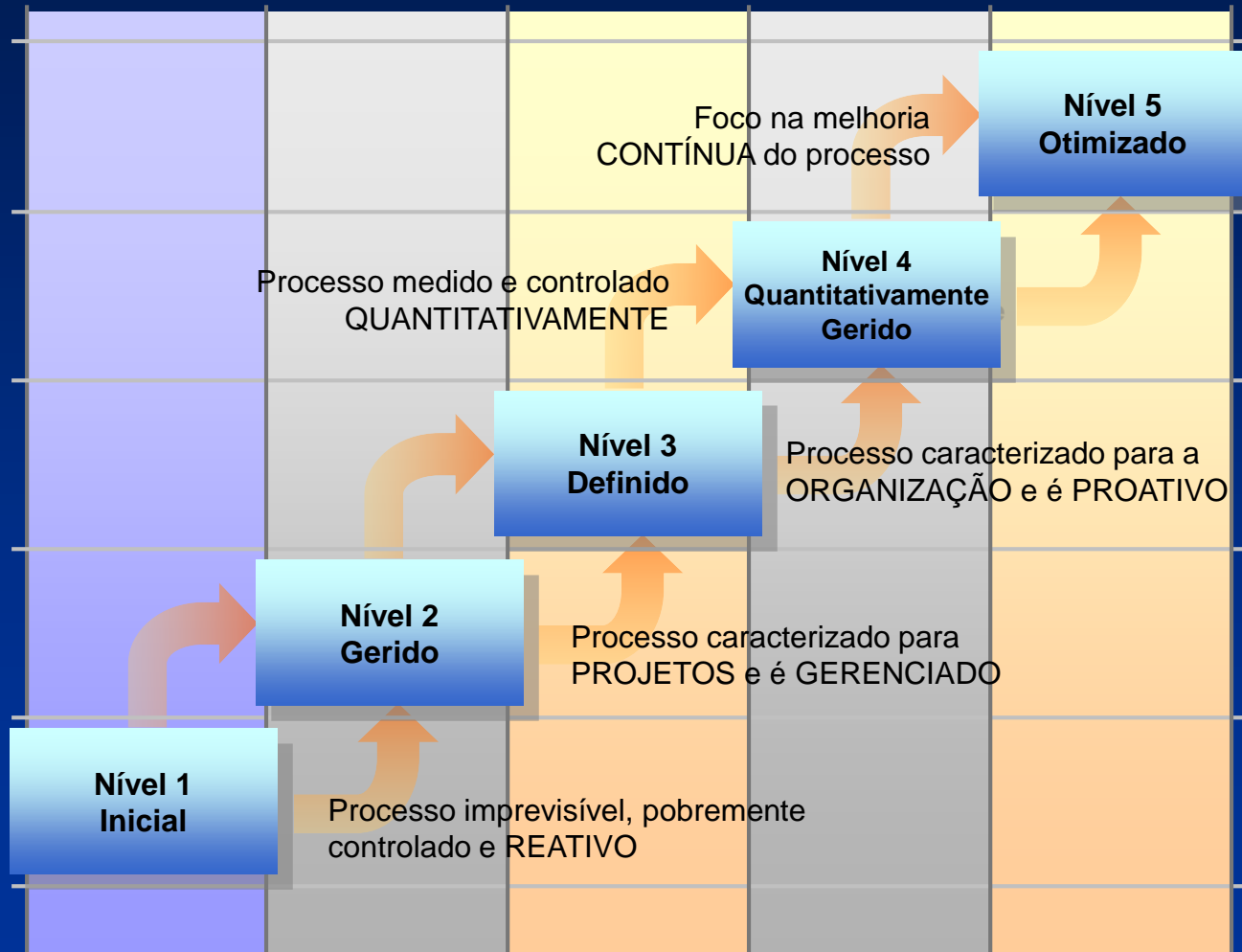
CMMI (Capability Maturity Model Integration)

- O CMMI é um metamodelo de processo baseado em um conjunto de capacidades de engenharia de software que devem estar presentes à medida que as empresas alcançam diferentes níveis de capacidade e maturidade de processo.
- Ele define (em mais de 700 páginas) as características de processo que devem existir quando uma organização deseja estabelecer um processo que seja completo.
- Através de um processo formal de avaliação, uma organização é classificada em um “nível de maturidade” que varia de um a cinco.

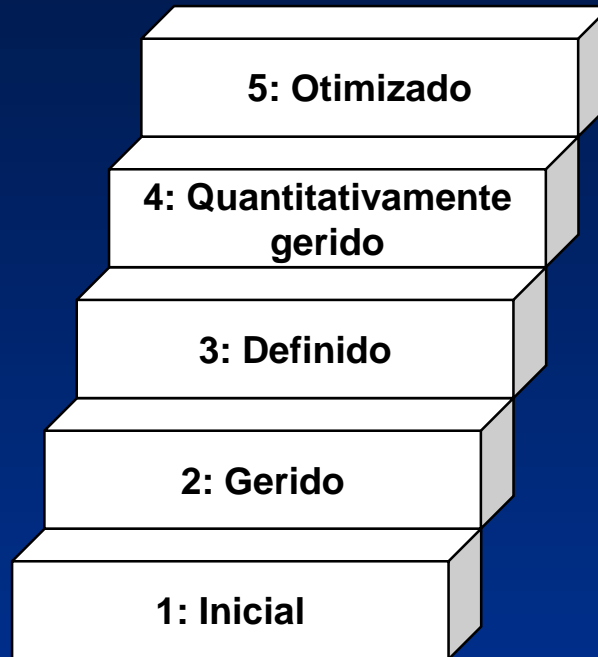
CMMI (Capability Maturity Model Integration)

- Fornece uma visão estruturada para a melhoria dos processos de uma organização.
- O CMMI é mais adequado para empresas que constroem sistemas grandes que envolvam centenas de pessoas, principalmente por causa dos custos.
- Desenvolvido pela SEI (Software Engineering Institute) – <http://www.sei.cmu.edu/cmmi/>

Capability Maturity Model Integration



CMMI: Níveis de Maturidade



“Sob nossa perspectiva, as organizações de nível 1 estão geralmente colidindo com as paredes, as de nível 2 e 3 estão aprendendo onde as paredes estão, e as de nível 4 e 5 estão construindo paredes para os outros colidirem.”

John Major,
Vice Presidente Sênior,
Motorola SW Solutions

CMMI: Níveis de Maturidade

- O SEI tem áreas-chave de processo (KPA) associadas a cada um dos níveis de maturidade. As KPA descrevem as funções de engenharia de software (por exemplo, planejamento de projetos, e gestão de requisitos), que devem estar presentes para satisfazer uma boa prática em um nível específico.

CMMI: Níveis de Maturidade

- A maioria das organizações de nível 1 realizam algumas das práticas esperadas no nível 2, mas não conseguem repetir os resultados.
- Problemas típicos:
 - Gerentes de projetos não treinados;
 - Práticas disciplinadas são sacrificadas nas crises de cronograma;
 - Desenvolvedores apóiam-se em métodos pessoais.



CMMI: Níveis de Maturidade

Nível de Maturidade	Foco	Áreas de Processo
2: Gerido	Gestão básica de projeto	<ol style="list-style-type: none">1. Gestão de requisitos2. Planejamento de projeto3. Monitoramento e controle de projeto4. Gestão de acordo com fornecedor5. Medição e análise6. Garantia de qualidade de processo e de produto.

CMMI: Níveis de Maturidade

Nível de Maturidade	Foco	Áreas de Processo
3: Definido	Padronização de processo	<ol style="list-style-type: none">1. Desenvolvimento de Requisitos2. Solução técnica3. Integração de produtos4. Verificação5. Validação6. Foco do processo organizacional7. Definição do processo organizacional8. Treinamento organizacional9. Gestão integrada de projeto10. Gestão integrada de fornecedor11. Gestão de risco12. Análise e resolução de decisão13. Ambiente organizacional para integração14. Formação integrada de equipe

CMMI: Níveis de Maturidade

Nível de Maturidade	Foco	Áreas de Processo
4: Quantitativamente gerido	Gestão quantitativa	1. Desempenho de processo organizacional 2. Gestão quantitativa de projeto

CMMI: Níveis de Maturidade

Nível de Maturidade	Foco	Áreas de Processo
5: Otimizado	Aperfeiçoamento contínuo de processo	<ol style="list-style-type: none">1. Inovação e implantação organizacional2. Análise e resolução Causal.