

Engenharia de Software II

Aula 2

<http://www.ic.uff.br/~bianca/engsoft2/>

Ementa

- **Processos de desenvolvimento de software**
- Estratégias e técnicas de teste de software
- Métricas para software
- Gestão de projetos de software: conceitos, métricas, estimativas, cronogramação, gestão de risco, gestão de qualidade e gestão de modificações
- Reengenharia e engenharia reversa

Revisão: Processo de software

- É um roteiro que determina quais são as **tarefas necessárias** e em que **ordem** elas devem ser executadas para construir softwares de **alta qualidade**.
- Ele **organiza** uma atividade que pode, sem controle, tornar-se caótica.
- O processo adotado deve ser **adaptado** ao **tipo de software** que se está construindo.
 - Software para aeronave vs. Software para internet.

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



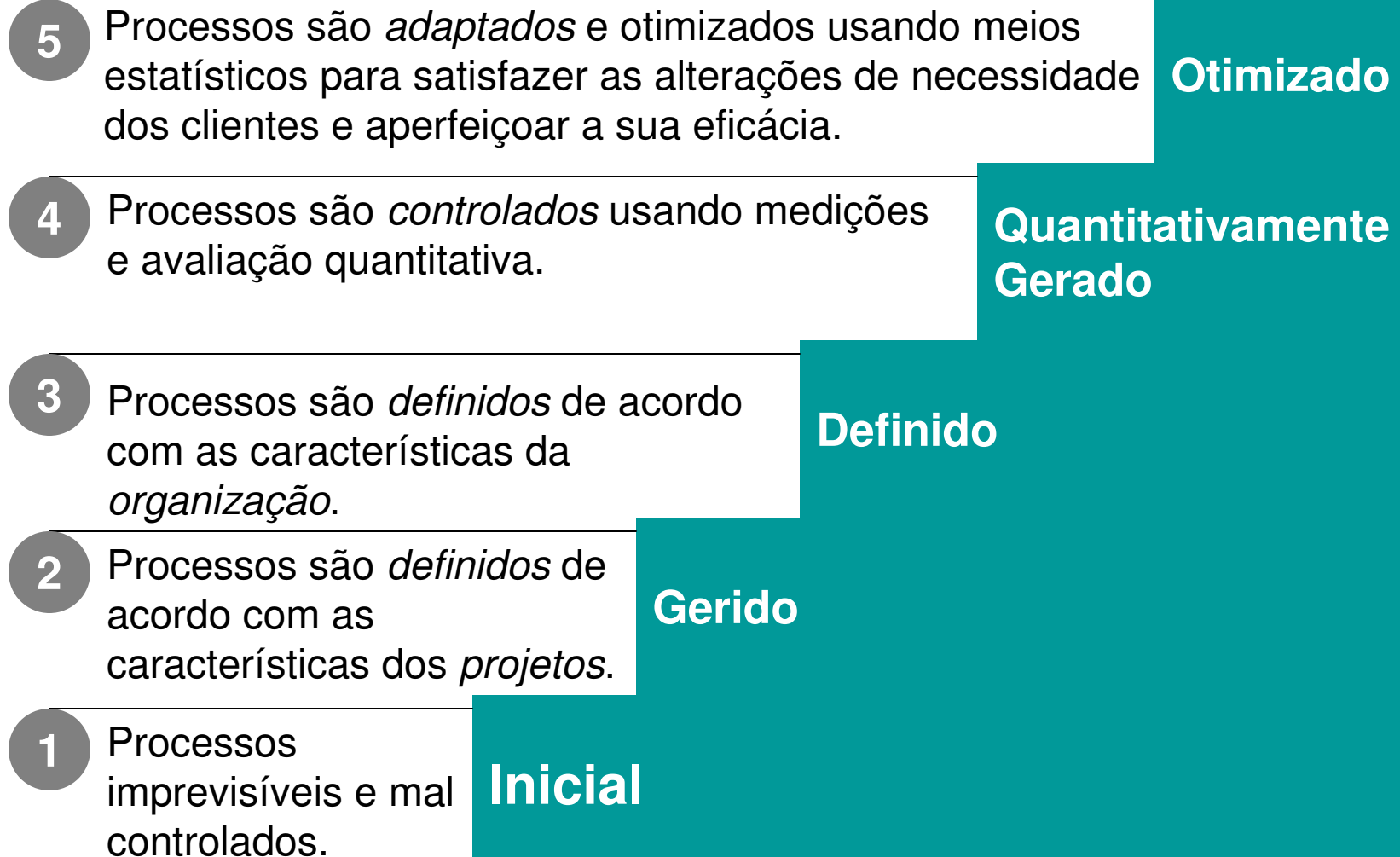
Modelos de Maturidade

- Descrevem as características de processos efetivos.
- Definem quais ações devem ser priorizadas e quais objetivos devem ser atingidos.
- Reúnem conhecimento e experiências diversas num único modelo.
- Podem ser usados para comparar organizações diferentes quanto a eficácia dos seus processos.

CMMI (Capability Maturity Model Integration)

- É um modelo que especifica um conjunto de capacidades de engenharia de software que uma empresa deve ter estabelecido para atingir diferentes níveis de maturidade.
- 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.

Níveis de Maturidade



Metas e Práticas Específicas

- Definidas pelo CMMI para cada área de processo:
 - Planejamento de projeto
 - Gestão de requisitos
 - Gestão de configuração
 - Gestão de qualidade e projeto de produto
 - Medição de análise

Métodos e Práticas para Planejamento de Projeto

- **ME 1 – Estabelecer estimativas**
 - PE 1.1-1 – Estime o escopo do projeto.
 - PE 1.2-1 – Estabeleça estimativas de produto do trabalho
 - PE 1.3-1 – Defina o ciclo de vida do projeto.
 - PE 1.4-1 – Determine estimativas de esforço e custo.
- **ME 2 – Desenvolver um plano de projeto**
 - PE 2.1-1 – Estabeleça o orçamento e o cronograma.
 - PE 2.2-1 – Identifique os riscos do projeto.
 - PE 2.3-1 – Planeje a gestão de dados.
 - PE 2.4-1 – Planeje os recursos de projeto.
 - PE 2.5-1 – Planeje as habilidades e conhecimentos necessários.
 - PE 2.6-1 – Planeje o envolvimento dos interessados.
 - PE 2.7-1 – Estabeleça o plano de projeto.
- **ME 2 – Obter comprometimento com o plano**
 - PE 3.1-1 – Revise planos que afetam o projeto.
 - PE 3.2-1 – Reconcilie os níveis de trabalho e os recursos.
 - PE 3.3-1 – Obtenha o comprometimento com o plano.

Padrões de Processo

- O processo de software pode ser definido como uma coleção de **padrões** que definem um conjunto de:
 - Atividades
 - Ações
 - Tarefas de trabalho
 - Produtos de trabalho
 - Comportamentos de trabalhonecessários ao desenvolvimento de software.
- Pela combinação de padrões, uma equipe pode construir um processo que melhor satisfaça às características de um projeto.

Descrição de Padrão de Processo

- **Nome do Padrão**: descreve sua função dentro do processo.
- **Intenção**: descreve o objetivo do padrão.
- **Tipo**:
 - Padrão de Tarefa: define uma ação ou tarefa de trabalho.
 - Padrão de Estágio: define uma atividade de arcabouço.
 - Padrão de Fase: define uma seqüência de atividades de arcabouço.
- **Contexto Inicial**: descreve condições sob as quais o padrão se aplica.
- **Problema**: descreve o problema a ser resolvido pelo padrão.
- **Solução**: descreve como o estado inicial do processo é modificado em consequência da aplicação do padrão.
- **Contexto Resultante**: condições que resultarão quando o processo for aplicado.
- **Padrões Relacionados**: lista de todos os padrões relacionados a este.
- **Usos Conhecidos/Exemplos**: instâncias específicas nas quais o padrão é aplicável.

Exemplo

- **Nome do Padrão:** Prototipação
- **Intenção:** Construir um protótipo que será avaliado iterativamente pelos interessados em solidificar requisitos de software.
- **Tipo:** Padrão de fase
- **Contexto Inicial:**
 - Interessados identificados
 - Modo de comunicação estabelecido
 - Problema identificado
 - Entendimento inicial do escopo, requisitos e restrições.
- **Problema:** interessados estão inseguros do que desejam.
- **Solução:** descrição do processo de prototipação.
- **Contexto Resultante:** um protótipo de software que identifique os requisitos básicos é aprovado pelos interessados.
- **Padrões Relacionados:** comunicação-com-o-cliente, projeto-iterativo, desenvolvimento-iterativo, avaliação-pelo-cliente, levantamento-de-requisitos.
- **Usos Conhecidos/Exemplos:** a prototipação é recomendada quando os requisitos são incertos.

Atividade

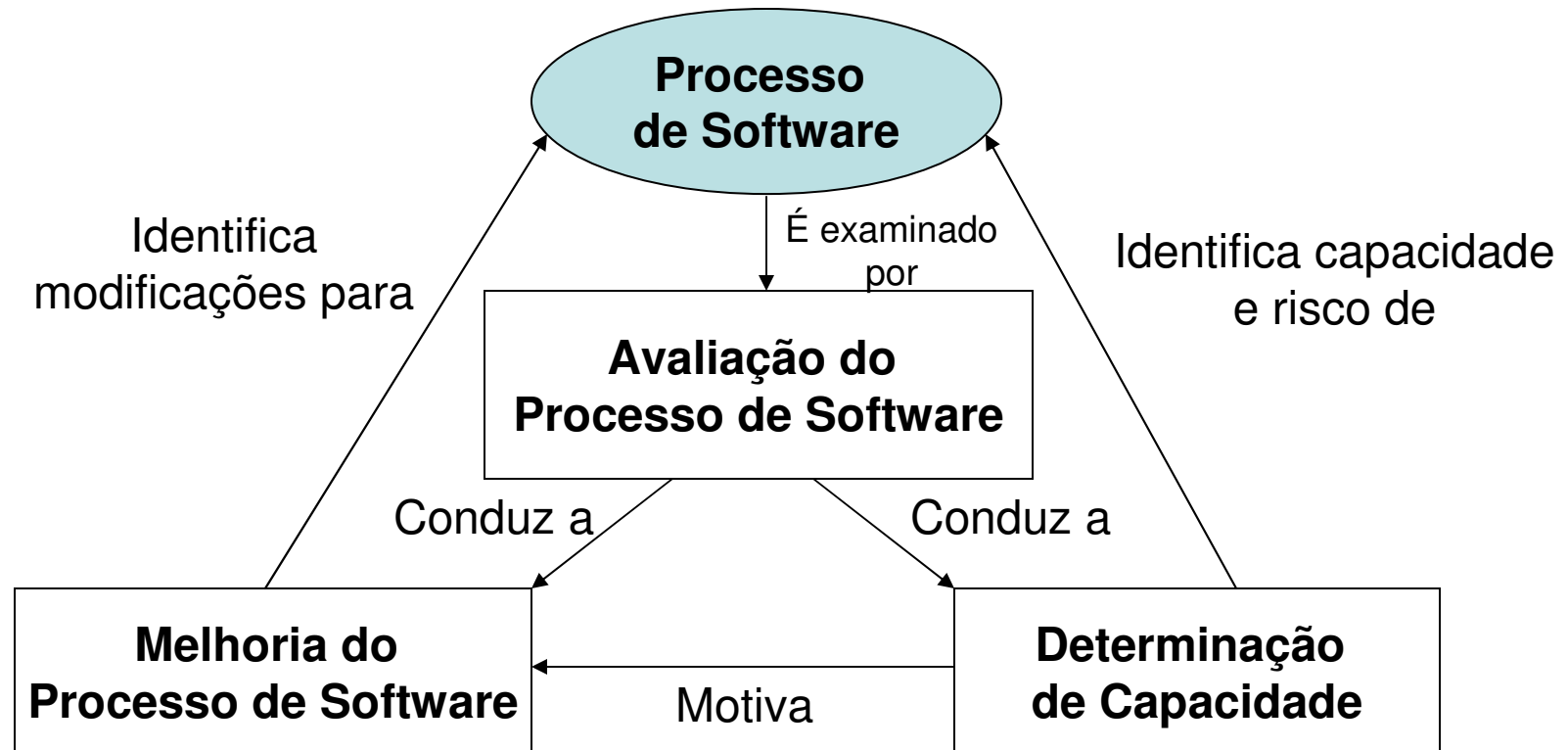
- Forme uma dupla e escreva uma descrição para o padrão de processo:

Comunicação-inicial-com-o-cliente

incluindo nome, intenção, tipo (tarefa, estágio ou fase), contexto inicial, problema, solução, contexto resultante, padrões relacionandos e usos conhecidos.

Avaliação de Processo

- O processo deve ser avaliado para garantir que ele satisfaça a um conjunto de critérios básicos.



Abordagens de Avaliação

- **SCAMPI = Standard CMMI Assessment Method for Process Improvement**
 - Usa o CMMI como base para a avaliação.
- **CBA IPI = CMM-Based Appraisal for Internal Process Improvement**
 - Avalia a maturidade relativa de uma organização de software usando o modelo CMM (precursor do CMMI).
- **Norma SPICE (ISO/IEC 15504)**
 - Define um conjunto de requisitos para avaliação de processos de software.
- **Norma ISO 9001:2000 para Software**
 - Define os requisitos de um sistema de gestão de qualidade, enfatizando o ciclo “planejar-fazer-verificar-agir”.

PSP – Processo Pessoal de Software

- É um modelo de processo que deve ser seguido individualmente por cada desenvolvedor de software.
- Define cinco atividades de arcabouço:
 1. Planejamento = requisitos e estimativas
 2. Projeto de alto nível = especificações e protótipo
 3. Revisão do projeto de alto nível = verificação formal
 4. Desenvolvimento = geração e teste de código
 5. Pós-conclusão = medição e aperfeiçoamento
- O PSP não tem sido adotado amplamente na indústria porque exige um treinamento demorado e com custos altos. Além disso, a medição exigida é culturalmente difícil.

TSP – Processo de Equipe de Software