

Processo de Produção de Software - PS22S

2019 - Sem 2

Prof. Rafael Oliveira (UTFPR - campus DV)



Tópicos

- 1 O que é um Processo de Software?
- 2 Definição de Processos
- 3 Modelos de Ciclo de Vida ou Modelos de Processo
 - Modelos de Processo Sequenciais
 - Modelos de Processo Incrementais
 - Modelos de Processo Evolutivo
 - O Processo Unificado

O que é um Processo de Software?

- Conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software.

O que é um Processo de Software?

- Conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software.

O que é um processo eficaz?

- Considera relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.

O que é um Processo de Software?

- Conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software.

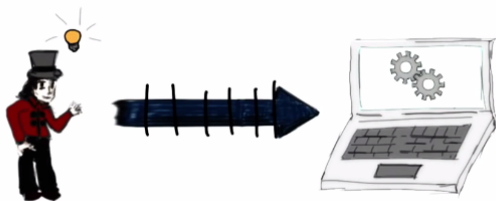
O que é um processo eficaz?

- Considera relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.

Ler capítulo 1.4 do Pressman.

O que é um Processo de Software?

SOFTWARE DEVELOPMENT



SOFTWARE PROCESS

Figura: (UDACITY, UD805)

Metodologia de Processo Genérica

Atividades estruturais aplicáveis a todos os projetos de software:

Metodologia de Processo Genérica

Atividades estruturais aplicáveis a todos os projetos de software:

Comunicação

Compreender os objetivos das partes interessadas (*stakeholders*).

Planejamento*

O projeto de software é uma jornada complicada. É preciso de um mapa: um *plano de projeto de software*, que inclui tarefas, riscos prováveis, recursos necessários, produtos resultantes e cronograma.

Modelagem

Esboço da coisa. Compreender melhor o problema. Desenho (design) de modelos.

Construção

Geração de código e testes.

Emprego

Software entregue e avaliado pelo cliente (*feedback*)

Elementos do Processo de Software

Processo de Software

Processos

Atividades

Pré-atividades

Subatividades

Artefatos

Insumos

Produtos

Recursos

Recursos Humanos

Ferramentas de Software

Hardware

Procedimentos

Métodos

Técnicas

Roteiros

Decomposição do Processo de Software

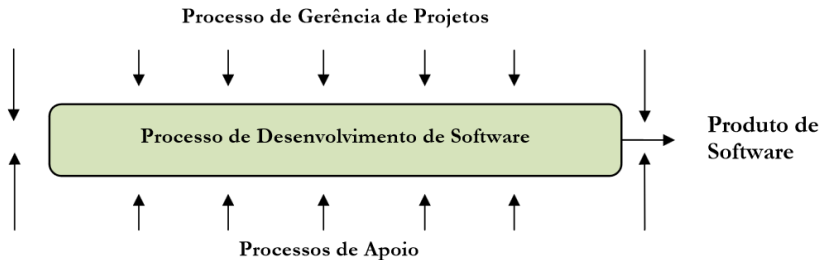


Figura: (FALBO, 2014)

O que são as atividades de apoio?

Atividades de apoio típicas:

- Controle e acompanhamento do projeto;
- Administração de riscos;
- Garantia da qualidade de software;
- Revisões técnicas;
- Medição;
- Gerenciamento de configuração (gestão de mudanças);
- Gerenciamento de reusabilidade;
- Preparo e produção de artefatos de software.

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal
- Devem ser definidos caso a caso
 - Deve ser adequado às especificidades do projeto

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal
- Devem ser definidos caso a caso
 - Deve ser adequado às especificidades do projeto

O que deve ser considerado?

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal
- Devem ser definidos caso a caso
 - Deve ser adequado às especificidades do projeto

O que deve ser considerado?

- Características da aplicação (domínio do problema, tamanho, complexidade, etc.)

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal
- Devem ser definidos caso a caso
 - Deve ser adequado às especificidades do projeto

O que deve ser considerado?

- Características da aplicação (domínio do problema, tamanho, complexidade, etc.)
- Tecnologia (paradigma de desenvolvimento, linguagem de programação, mecanismo de persistência, etc.)

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal
- Devem ser definidos caso a caso
 - Deve ser adequado às especificidades do projeto

O que deve ser considerado?

- Características da aplicação (domínio do problema, tamanho, complexidade, etc.)
- Tecnologia (paradigma de desenvolvimento, linguagem de programação, mecanismo de persistência, etc.)
- A organização (políticas da empresa, padrões, etc.)

Definição do Processo de Desenvolvimento

Objetivo

Favorecer a produção de sistemas de alta qualidade, dentro de um cronograma e um orçamento previsíveis.

- Não existe um processo universal
- Devem ser definidos caso a caso
 - Deve ser adequado às especificidades do projeto

O que deve ser considerado?

- Características da aplicação (domínio do problema, tamanho, complexidade, etc.)
- Tecnologia (paradigma de desenvolvimento, linguagem de programação, mecanismo de persistência, etc.)
- A organização (políticas da empresa, padrões, etc.)
- A equipe de desenvolvimento

Definição de Processos

Atividades-chave (a base para todo processo):

- Análise e especificação de requisitos
- Projeto (*design*)
- Implementação e testes

Definição de Processos

Atividades-chave (a base para todo processo):

- Análise e especificação de requisitos
- Projeto (*design*)
- Implementação e testes

O que mais precisa ser definido?

- Modelo de Ciclo de Vida (Modelo de Processo)
- Detalhamento (decomposição) das macro-atividades
- Escolha de métodos, técnicas e roteiros (procedimentos)
- Definição de recursos e artefatos necessários e produzidos

Definição de Processos

Atividades-chave (a base para todo processo):

- Análise e especificação de requisitos
- Projeto (*design*)
- Implementação e testes

O que mais precisa ser definido?

- **Modelo de Ciclo de Vida (Modelo de Processo)**
- Detalhamento (decomposição) das macro-atividades
- Escolha de métodos, técnicas e roteiros (procedimentos)
- Definição de recursos e artefatos necessários e produzidos

Modelos de Ciclo de Vida ou Modelos de Processo

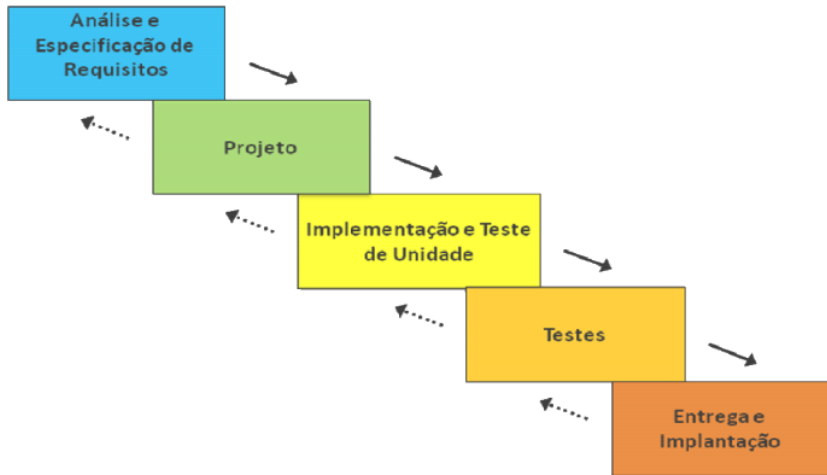
- **É um esqueleto de processo**
- Tipicamente inclui:
 - Algumas atividades principais
 - Ordem de precedência das atividades
 - Às vezes: artefatos requeridos e produzidos
- Estruturação das atividades em **fases**

Modelos de Processo - Fases

Principais Fases:

- Planejamento
- Análise e Especificação de Requisitos
- Projeto (design)
- Implementação
- Testes
- Entrega e Implantação
- Operação
- Manutenção

Modelo Cascata



Modelo Cascata

- Modelo sequencial
- Guiado por documentos

Modelo Cascata

- Modelo sequencial
- Guiado por documentos
- (*Infelizmente*) Muito Utilizado

Modelo Cascata

- Modelo sequencial
- Guiado por documentos
- (*Infelizmente*) Muito Utilizado
- Infelizmente por quê? Quais são os problemas?

Modelo Cascata - Problemas

- Dificuldade de manter a serialização proposta

Modelo Cascata - Problemas

- Dificuldade de manter a serialização proposta
- Modificações constantes nos requisitos

Modelo Cascata - Problemas

- Dificuldade de manter a serialização proposta
- Modificações constantes nos requisitos
- Primeira versão do software... lá longe...

Modelo Cascata - Problemas

- Dificuldade de manter a serialização proposta
- Modificações constantes nos requisitos
- Primeira versão do software... lá longe...

No entanto, em que situações seria possível / adequado utilizar?

Modelo em V

Uma variação do Modelo Cascata, o Modelo em V.

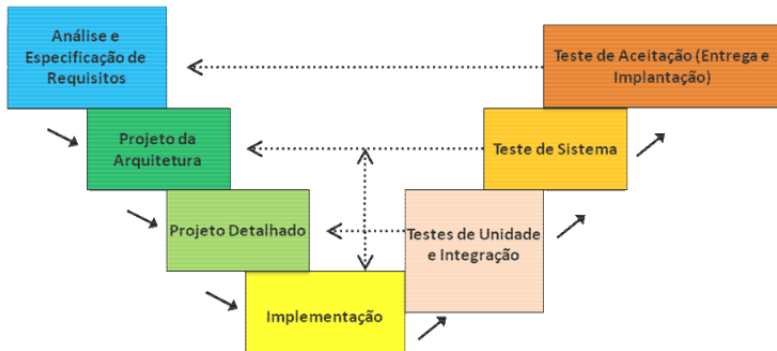


Figura: Modelo em V (FALBO, 2014)

Modelo em V

- Foco na relação das atividades de teste com as demais fases do processo
- Uma fase pode ser repetida caso os testes correspondentes falhem
- Ainda apresenta os problemas do Modelo Cascata
- Clientes não estão dispostos a esperar um tempo longo para o primeiro *release*

Modelos de Processo Incrementais

Princípio

Entregar ao cliente, a cada ciclo ou **iteração**, uma versão operacional do sistema para uso e avaliação do cliente.

Modelos de Processo Incrementais

Princípio

Entregar ao cliente, a cada ciclo ou **iteração**, uma versão operacional do sistema para uso e avaliação do cliente.

- Requisitos devem ser levantados com muito cuidado

Modelos de Processo Incrementais

Princípio

Entregar ao cliente, a cada ciclo ou **iteração**, uma versão operacional do sistema para uso e avaliação do cliente.

- Requisitos devem ser levantados com muito cuidado
- Planejamento do desenvolvimento em incrementos: Sistema Modular

Modelos de Processo Incrementais

Princípio

Entregar ao cliente, a cada ciclo ou **iteração**, uma versão operacional do sistema para uso e avaliação do cliente.

- Requisitos devem ser levantados com muito cuidado
- Planejamento do desenvolvimento em incrementos: Sistema Modular
- 1º incremento: Funcionalidades centrais (requisitos básicos)

Modelos de Processo Incrementais

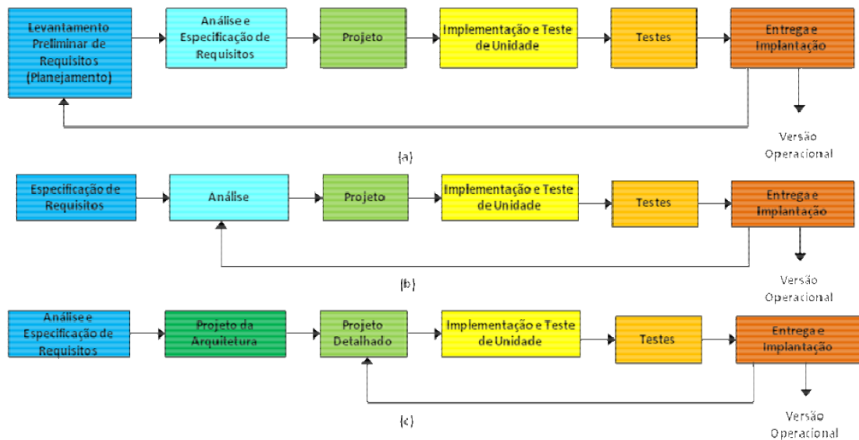


Figura: Variações do Modelo Incremental (FALBO, 2014)

Modelos de Processo Incrementais

Vantagens

- ↓ tempo e ↓ custo para entregar 1ª versão
- ↓ riscos associados ao desenvolvimento de um incremento
- Mudanças de requisitos durante o desenvolvimento podem diminuir

Modelos de Processo Incrementais

Vantagens

- ↓ tempo e ↓ custo para entregar 1ª versão
- ↓ riscos associados ao desenvolvimento de um incremento
- Mudanças de requisitos durante o desenvolvimento podem diminuir

Desvantagens

- Requisitos podem não ser tão completos ou estáveis quanto se esperava ⇒ Mudanças nos incrementos
- Gerência do projeto mais complexa se a divisão em subsistemas não for boa

Modelo RAD

- Modelo incremental para projetos curtos (max 90 dias)
- Incrementos desenvolvidos em paralelo por diferentes equipes
- Entrega única

Modelo RAD

- Modelo incremental para projetos curtos (max 90 dias)
- Incrementos desenvolvidos em paralelo por diferentes equipes
- Entrega única

Quando utilizar?

- Requisitos devem ser bem definidos
- Escopo do projeto restrito
- Sistema modular

Modelo RAD

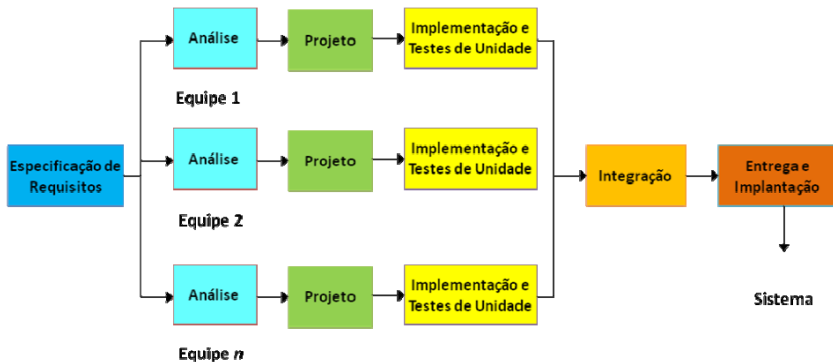


Figura: Modelo RAD (FALBO, 2014)

Modelos de Processo Evolutivo

- Sistemas de software são complexos e mudam com frequência;
- Necessário lidar com a incerteza e mudanças contínuas;
- Objetivo: produzir software de **alta qualidade** de modo **iterativo e incremental** frente aos prazos apertados do mercado e volatilidade de requisitos

Modelos de Processo Evolutivo

- Sistemas de software são complexos e mudam com frequência;
- Necessário lidar com a incerteza e mudanças continuas;
- Objetivo: produzir software de **alta qualidade** de modo **iterativo e incremental** frente aos prazos apertados do mercado e volatilidade de requisitos

Incrementais

- Pressupõem requisitos bem definidos
- Entrega de versões operacionais desde o primeiro ciclo

Evolutivos

- Acomodam requisitos voláteis
- Entrega de protótipos e modelos no primeiro ciclo, que vão evoluindo para modelos operacionais

Modelo em Espiral

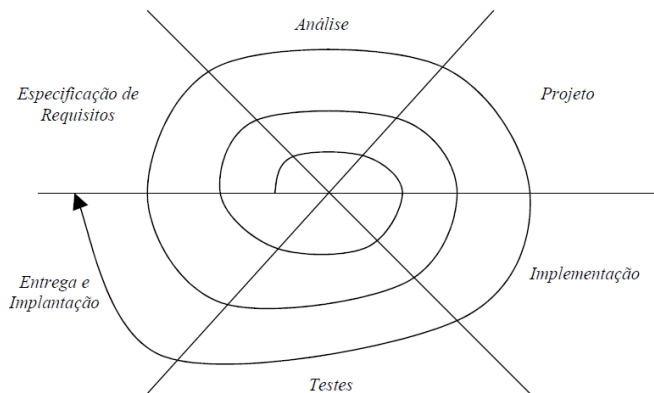


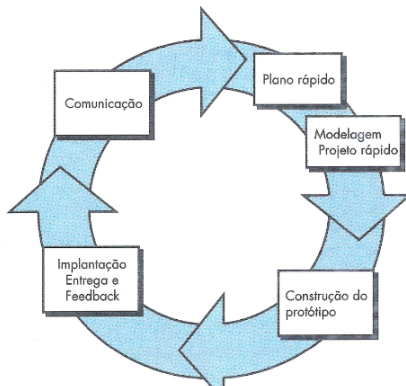
Figura: Modelo Espiral (FALBO, 2014)

Modelo em Espiral

- Primeiros ciclos nem sempre todas as atividades são realizadas
- Passadas subsequentes podem ser usadas para desenvolver protótipos

Prototipação

- Os próprios clientes podem não entender o que realmente querem/precisam.
- Modelos e protótipos podem ajudar a levantar e validar requisitos



Processos Evolutivos - Desvantagens

- Número incerto de ciclos
- Velocidade incerta dos ciclos
- Dificuldades de planejamento e gerência do projeto
- **Polêmica:** o processo deve manter o foco mais na flexibilidade, extensibilidade e velocidade de desenvolvimento do que na alta qualidade...

Modelos de Processo Especializado

Pesquisar:

- Desenvolvimento baseado em componentes
- Modelo de métodos formais
- Desenvolvimento de software orientado a aspectos

O Processo Unificado (RUP)

- Também conhecido como RUP (*Rational Unified Process*)
- Tentativa de incorporar as melhores práticas de desenvolvimento de software
 - prototipação;
 - entrega incremental;
 - princípios do desenvolvimento ágil aproveitando os melhores recursos dos modelos tradicionais

O Processo Unificado (RUP)

- Também conhecido como RUP (*Rational Unified Process*)
- Tentativa de incorporar as melhores práticas de desenvolvimento de software
 - prototipação;
 - entrega incremental;
 - princípios do desenvolvimento ágil aproveitando os melhores recursos dos modelos tradicionais
- Ele é mais que um modelo de processo: é uma abordagem completa p/ o desenvolvimento de software
 - Definição detalhada de papéis, atividades, artefatos e fluxos de trabalho

O Processo Unificado (RUP)

- Também conhecido como RUP (*Rational Unified Process*)
- Tentativa de incorporar as melhores práticas de desenvolvimento de software
 - prototipação;
 - entrega incremental;
 - princípios do desenvolvimento ágil aproveitando os melhores recursos dos modelos tradicionais
- Ele é mais que um modelo de processo: é uma abordagem completa p/ o desenvolvimento de software
 - Definição detalhada de papéis, atividades, artefatos e fluxos de trabalho
- Processo "dirigido a casos de uso, centrado na arquitetura, iterativo e incremental";
- Engenharia de Software orientada a objetos usando a UML.

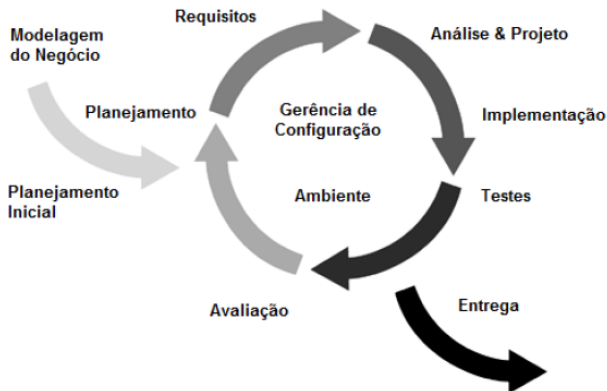


Figura: Desenvolvimento Iterativo no RUP (FALBO, 2014)

Perspectivas do RUP

- **Perspectiva dinâmica:** mostra as fases do modelo ao longo do tempo
- **Perspectiva estática:** mostra as atividades realizadas no processo
- **Perspectiva prática:** sugere boas práticas a serem usadas durante o processo

Fases do Processo Unificado

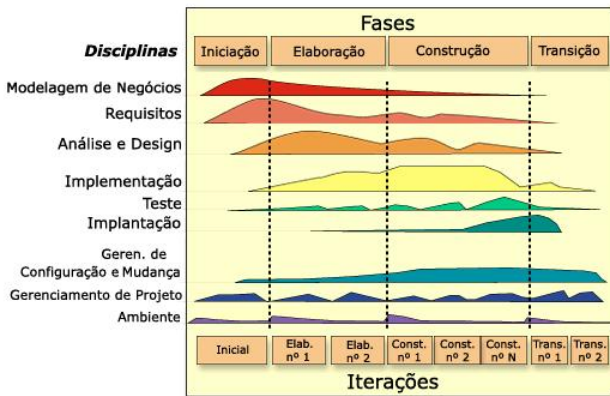


Figura: O modelo de processo RUP

Fases do Processo Unificado

O RUP considera quatro fases (**perspectiva dinâmica**):

Concepção:

- Entendimento do **escopo do projeto** e dos **requisitos de alto nível**.
- Estabelecer um *business case* para o sistema:
 - Definir interações: Identificar todas as entidades externas (usuários e sistemas)
 - Avaliar a contribuição do sistema para o negócio;
 - Pode-se concluir que a contribuição é pequena: projeto pode ser então cancelado.
- Podem ser construídos protótipos para melhorar compreensão

Fases do Processo Unificado

O RUP considera quatro fases (**perspectiva dinâmica**):

Elaboração:

■ Metas:

- Desenvolver uma compreensão do problema dominante;
- Estabelecer a arquitetura do sistema;
- Desenvolver o plano do projeto;
- Identificar os maiores riscos.

■ Artefatos gerados:

- Modelo de requisitos: conjunto de casos de uso (UML), descrição da arquitetura e/ou plano de desenvolvimento do software.

Fases do Processo Unificado

O RUP considera quatro fases (**perspectiva dinâmica**):

Construção:

- Envolve projeto, programação e testes
- As partes do sistema são desenvolvidas em paralelo e integradas;
- **Conclusão da fase:** sistema funcionando e documentação associada pronta para ser entregue.

Fases do Processo Unificado

O RUP considera quatro fases (**perspectiva dinâmica**):

Transição:

- Transferência do sistema do ambiente de desenvolvimento para ambiente do usuário;
- Muitas vezes ignorado em outros modelos de processo: mas é uma atividade cara e, às vezes, problemática.
- **Conclusão da fase:** sistema documentado e funcionando corretamente em ambiente operacional.

- Cada fase pode ser executada de forma iterativa e incremental
- Todo o conjunto de fases também pode ser executado de forma iterativa e incremental, como mostra a figura.
 - A iteração de todas as fases também é chamada de ciclo de desenvolvimento, sendo a primeira "passada" o *ciclo inicial de desenvolvimento*.

Fases no Rational Unified Process

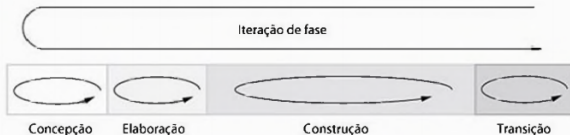


Figura: (SOMMERVILLE, 2011)

Workflows (disciplinas)

- **Perspectiva estática**
- Definem as prioridades das atividades no decorrer do processo de desenvolvimento
- Todas as disciplinas podem estar ativas em todas as fases...
- ... no entanto, varia-se os esforços desempenhados em cada disciplina dependendo da fase.
- **Exemplo:** nas fases iniciais, maior esforço nas disciplinas de modelagem de negócios e requisitos; nas fases posteriores, nas disciplinas de teste e implantação

Workflows (disciplinas)

WORKFLOW	DESCRIÇÃO
Modelagem de negócios	Os processos de negócio são modelados por meio de casos de uso de negócios.
Requisitos	Atores que interagem com o sistema são identificados e casos de uso são desenvolvidos para modelar os requisitos do sistema.
Análise e projeto	Um modelo de projeto é criado e documentado com modelos de arquitetura, modelos de componentes, modelos de objetos e modelos de sequência.
Implementação	Os componentes do sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código a partir de modelos de projeto ajuda a acelerar esse processo.
Teste	O teste é um processo iterativo que é feito em conjunto com a implementação. O teste do sistema segue a conclusão da implementação.
Implantação	Um <i>release</i> do produto é criado, distribuído aos usuários e instalado em seu local de trabalho.
Gerenciamento de configuração e mudanças	Esse <i>workflow</i> de apoio gerencia as mudanças do sistema (veja o Capítulo 25).
Gerenciamento de projeto	Esse <i>workflow</i> de apoio gerencia o desenvolvimento do sistema (veja os capítulos 22 e 23).
Meio ambiente	Esse <i>workflow</i> está relacionado com a disponibilização de ferramentas apropriadas para a equipe de desenvolvimento de software.

Figura: (SOMMERVILLE, 2011)

Artefatos e Atividades

Cada disciplina está associada a um conjunto de atividades e artefatos correlacionados.

- Um **artefato** é algum documento, relatório ou programa executável, que é produzido, manipulado ou consumido.
- Uma **atividade** descreve tarefas executadas pelos trabalhadores para criar ou modificar artefatos, possivelmente utilizando de *técnicas, diretrizes e ferramentas* para ajudar a automação de algumas tarefas

Modelos

O RUP utiliza alguns modelos baseados em UML, que, em conjunto, abrangem todas as decisões importantes para a visualização, especificação, construção e documentação de um sistema complexo de software.

1. Modelo de caso de uso de negócio	Estabelece uma abstração da empresa.
2. Modelo de análise do negócio	Estabelece o contexto do sistema.
3. Modelo de caso de uso	Estabelece os requisitos funcionais do sistema.
4. Modelo de análise (opcional)	Estabelece um projeto conceitual.
5. Modelo de projeto	Estabelece o vocabulário do problema e de sua solução.
6. Modelo de dados (opcional)	Estabelece a representação de dados para bancos de dados e outros repositórios.
7. Modelo de implantação	Estabelece a topologia do hardware em que o sistema é executado, bem como a concorrência do sistema e os mecanismos de sincronização.
8. Modelo de implementação	Estabelece as partes utilizadas para montar e liberar o sistema físico.

Outros Artefatos

1. Conjunto de requisitos
2. Conjunto de análise e projeto
3. Conjunto de teste
4. Conjunto de implementação
5. Conjunto de implantação

Descreve o que o sistema deve fazer.

Descreve como o sistema é construído.

Descreve a abordagem pela qual o sistema é validado e verificado.

Descreve a montagem dos componentes do software desenvolvido.

Fornece todos os dados para a configuração de implantação.

Figura: (BOOCH et al., 2012)

Boas práticas

Perspectiva prática

- 1 Desenvolver software iterativamente
- 2 Gerenciar os requisitos
- 3 Usar arquiteturas baseadas em componentes
- 4 Modelar o software visualmente
- 5 Verificar a qualidade do software
- 6 Controlar as mudanças do software

Sumarizando o RUP

- Processo flexível, mas bem estruturado.
- Inovações mais importantes:
 - Separação de fases e *workflows*;
 - Reconhecimento de que a implantação é parte do processo.
- As **fases** são dinâmicas e têm metas;
- Os **workflows (disciplinas)** são estáticos e são atividades técnicas que não são associadas a uma única fase, mas que podem ser utilizadas durante todo o desenvolvimento para alcançar as metas específicas das fases

Desenvolvimento Ágil

Mais detalhes em breve...

Enquanto isso, conseguem citar algumas práticas e modelos ágeis?

Escolhendo um modelo de processos

CHOOSING A SOFTWARE PROCESS MODEL



Figura: Caption

Quiz

Qual dos seguintes modelos seria o mais adequado para desenvolver um sistema de software de controle (de avião, por exemplo):

- 1 Cascata
- 2 Incremental
- 3 Espiral
- 4 Prototipação Evolutiva
- 5 RUP

Quiz

Qual dos seguintes modelos seria o mais adequado para um projeto que necessitará de correções ao longo do caminho?

- 1 Cascata
- 2 Incremental
- 3 Espiral
- 4 Prototipação Evolutiva
- 5 RUP



Dúvidas

