



Processo de Desenvolvimento de Software

Profº - Dr. Thales Levi Azevedo Valente
thales.l.a.valente@ufma.com.br

Sejam Bem-vindos !



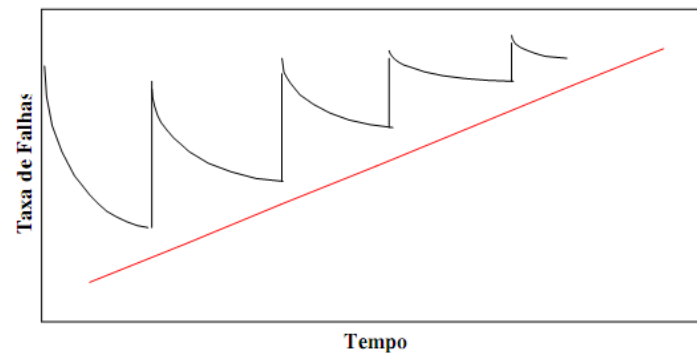
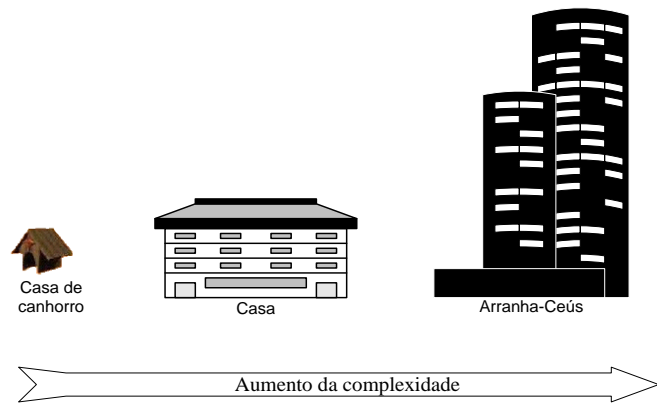
**Os celulares devem
ficar no silencioso
ou desligados**

Pode ser utilizado
apenas em caso
de emergência



**Boa tarde/noite, por
favor e com licença
DEVEM ser usados**

Educação é
essencial

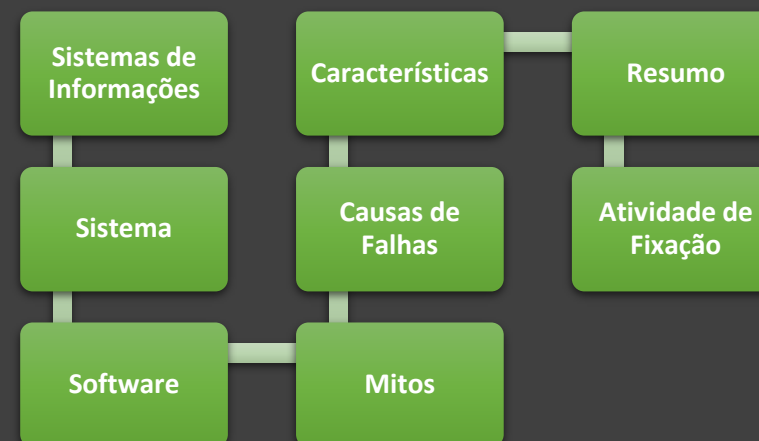
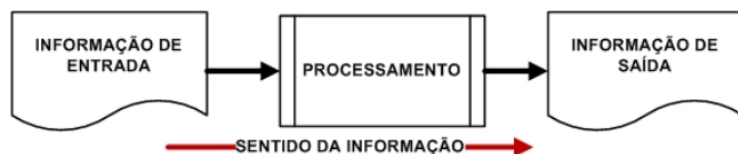


Mitos do Software

- Administrativos
- Cliente
- Profissional



Na aula anterior...



Objetivos de hoje



Oferecer uma visão geral do processo de desenvolvimento de software;



Ao final da aula, os alunos serão capazes de entender a importância da modelagem e de seguir as etapas do processo de desenvolvimento de software.



Roteiro: Processo de Software



Conceitos Básicos – Métodos

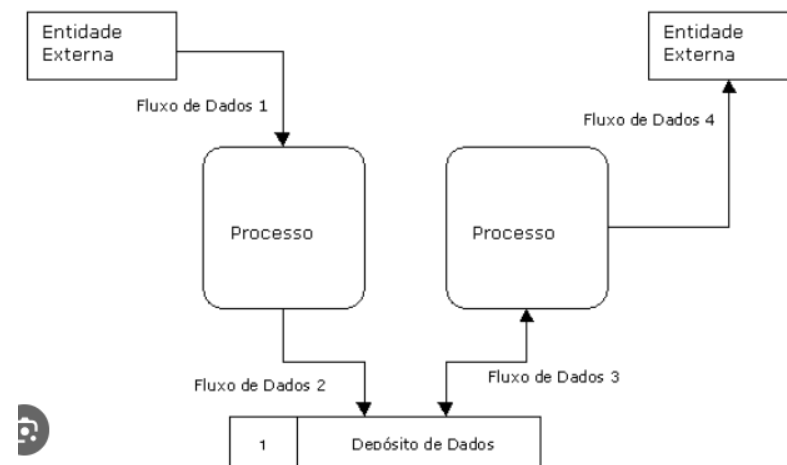
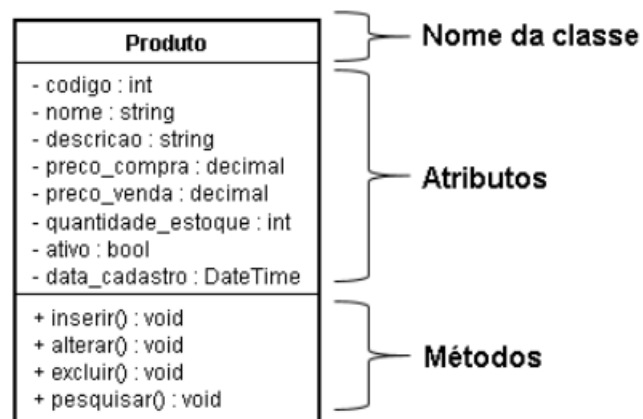
- Métodos: proporcionam os detalhes de “como fazer” para construir o software.
- Envolvem um amplo conjunto de tarefas
- Um método de ES é uma aproximação estruturada para o desenvolvimento de software
- Objetivo: Produção de software de alta qualidade e com bom custo-benefício.

Conceitos Básicos - Métodos

- Métodos devem incluir os seguintes componentes
 - Descrição gráficas
 - Regras
 - Recomendações
 - Diretrizes de processo
 - Ferramentas
 - Procedimentos

Conceitos Básicos - Métodos

- Descrição gráficas.
 - Descrições dos modelos do sistema que deverão ser desenvolvidos e da notação usada para os definir.
 - Ex. Modelos de objetos, fluxos de dados etc.



Conceitos Básicos - Métodos

- Regras
 - Restrições que se aplicam a modelos de sistema
 - Ex. cada entidade (pessoa) deve ter um único CPF

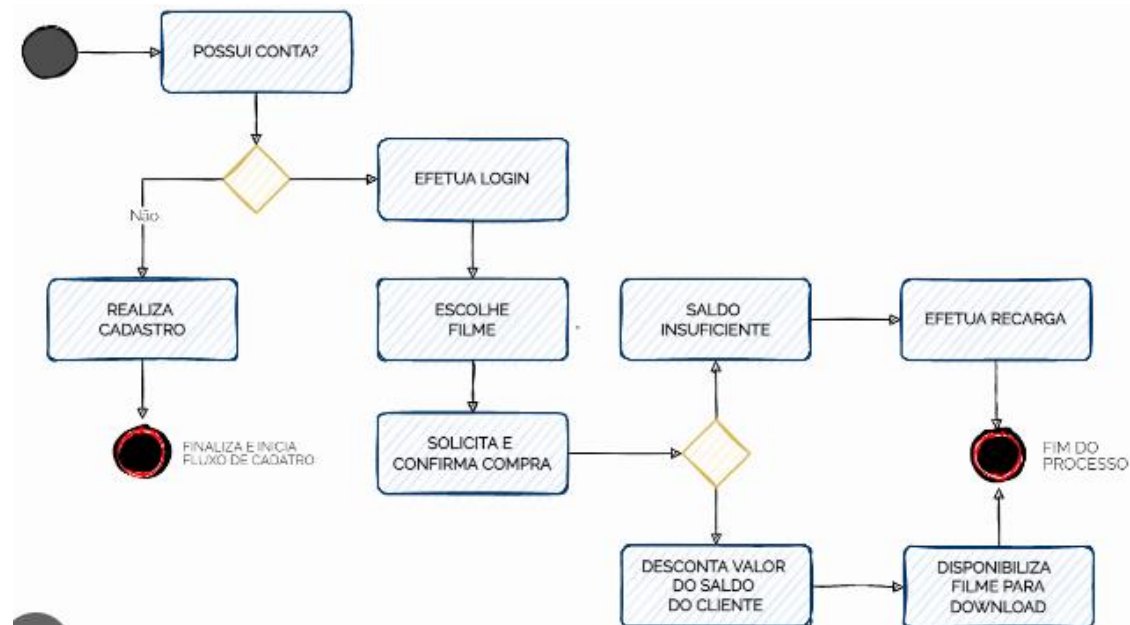


Conceitos Básicos - Métodos

- Recomendações
 - Conselho em prática de projeto
 - Ex. Nenhum objeto (pessoa) deve ter mais que 2 dependentes (sub-objetos)

Conceitos Básicos - Métodos

- Diretrizes de processo
 - Descrição das atividades que podem ser seguidas

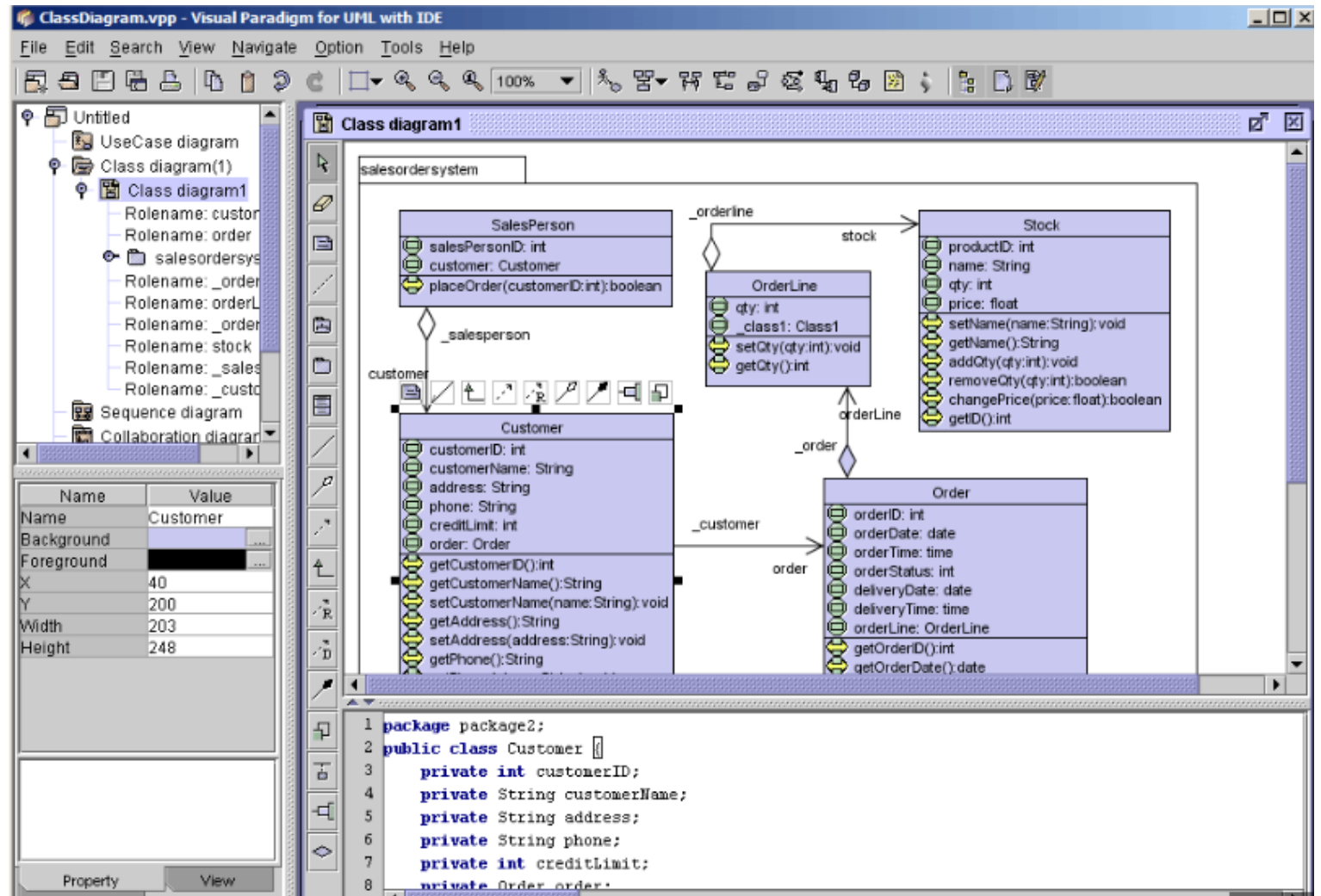


Conceitos Básicos - Métodos

- Ferramentas: fornecem suporte automatizado ou semi aos métodos
 - Existem atualmente ferramentas para sustentar cada um dos métodos
 - Quando as ferramentas são integradas é estabelecido um sistema de suporte ao desenvolvimento de software chamado CASE - Computer Aided Software Engineering

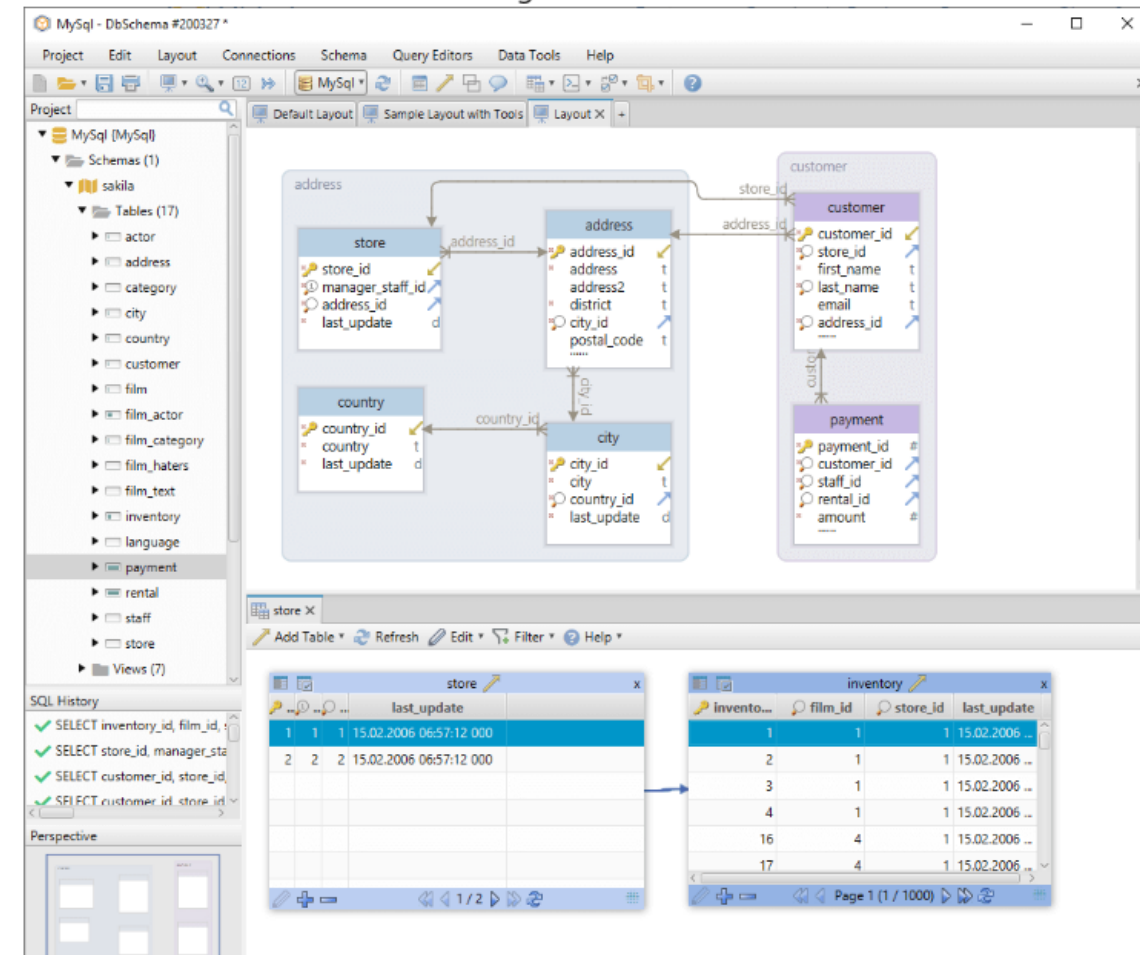
Conceitos Básicos - Métodos

CASE - Computer Aided Software Engineering



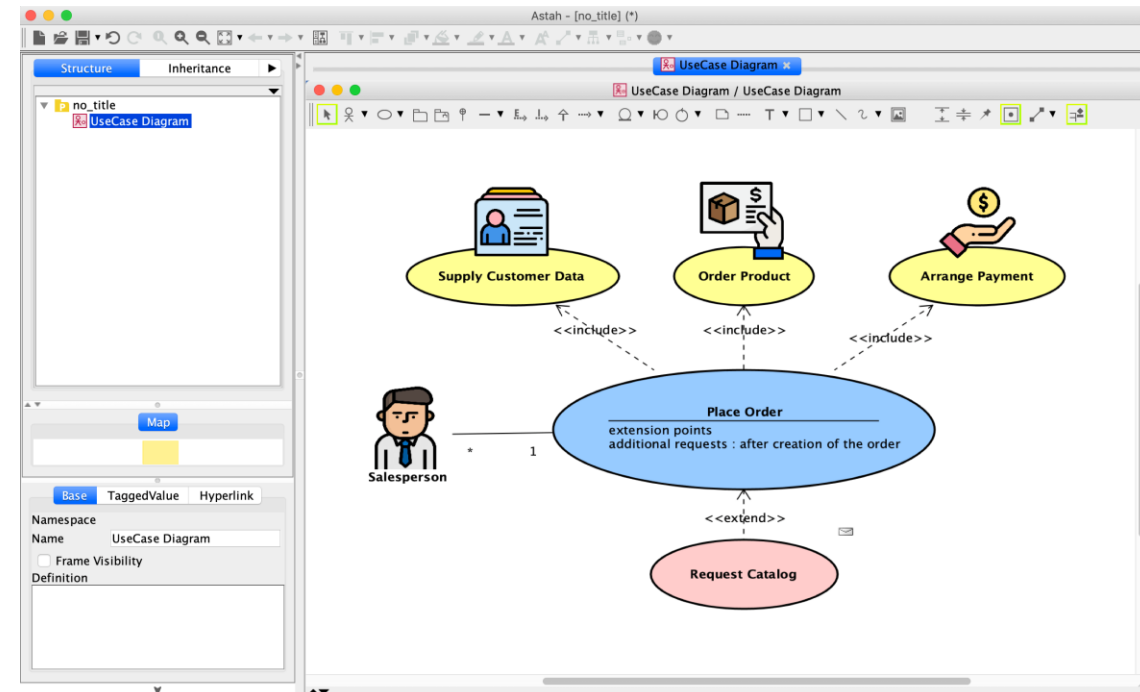
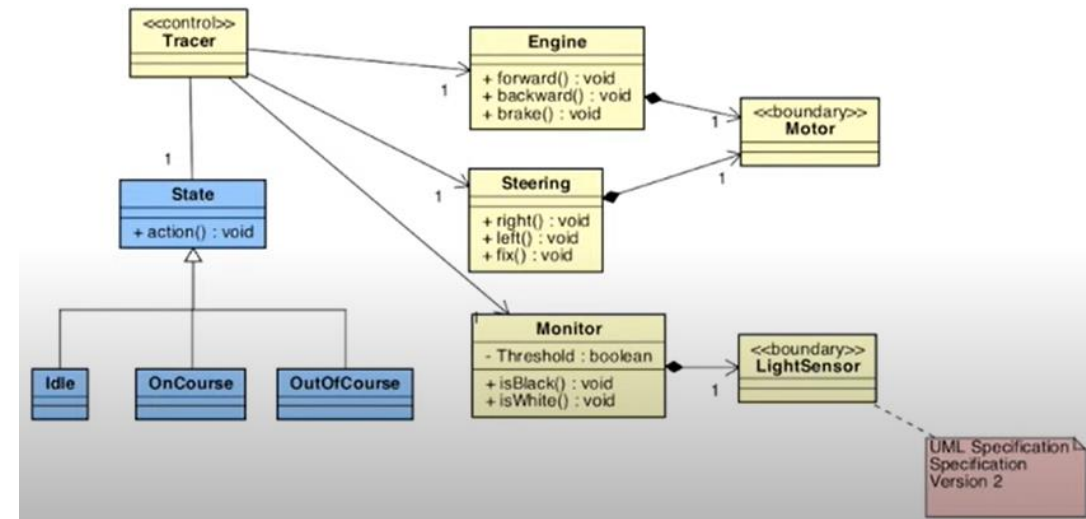
Conceitos Básicos – Ferramentas CASE

- DBSchema
- Designer e gerenciador de banco de dados visual que oferece suporte a uma ampla variedade de tipos de banco de dados, incluindo SQL, NoSQL e bancos de dados em nuvem
- Gerador de dados aleatórios integrado



Conceitos Básicos – Ferramentas CASE

- Astah
- Ferramenta de modelagem UML (Unified Modeling Language) e de diagramação de software
- Suporte a diversos tipos de diagramas, como diagramas de classes, diagramas de sequência, diagramas de casos de uso, diagramas de atividade, entre outros, que são amplamente usados na engenharia de software



Conceitos Básicos - Métodos

- Procedimentos: constituem o elo que mantém juntos os métodos e as ferramentas para desenvolvimento do software.
 - Seqüência em que os métodos serão aplicados
 - Produtos (deliverables) que se exige que sejam entregues
 - Controles que ajudam assegurar a qualidade e coordenar as alterações
 - Marcos de referência que possibilitam administrar o progresso do software

Conceitos Básicos - Métodos

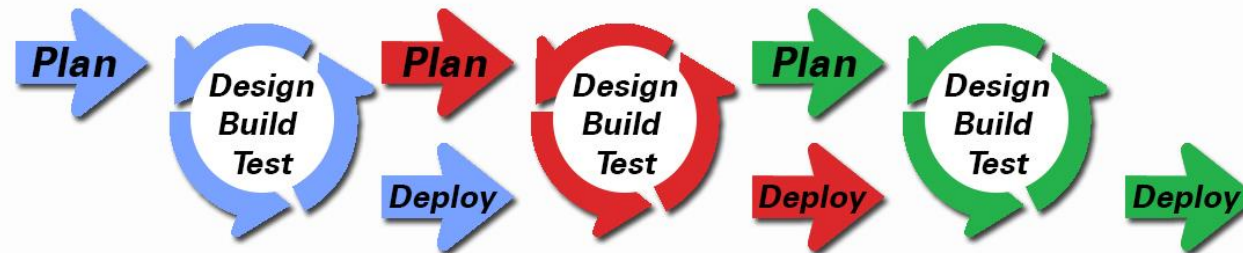
- Sequência -> A ordem ou sequência de aplicação dos métodos pode variar de acordo com a metodologia ou abordagem utilizada (por exemplo, desenvolvimento ágil versus cascata). Por exemplo:
 - Em um modelo cascata, as atividades ocorrem em uma sequência rígida, começando com a coleta de requisitos e avançando para o design, implementação, teste e manutenção.
 - Já em abordagens ágeis, como o Scrum, as atividades são realizadas em ciclos curtos e iterativos (sprints), onde requisitos, design, implementação e testes podem acontecer quase simultaneamente dentro de um sprint.

Métodos -> Procedimento -> Sequência

Waterfall
(Plan Driven)



Agile
(Value Driven)



Project Timeline

Conceitos Básicos - Métodos

- Produtos (deliverables) que se exige que sejam entregues: Cada fase ou atividade do processo de desenvolvimento geralmente resulta em um ou mais artefatos ou "deliverables". Estes podem incluir:
 - documentos de requisitos
 - designs arquitetônicos
 - código-fonte
 - casos de teste, entre outros.
- É importante especificar claramente quais produtos são esperados em cada fase, bem como seus critérios de aceitação.

Conceitos Básicos - Métodos

- Controles que ajudam a assegurar a qualidade e coordenar as alterações: Estes controles podem incluir:
 - revisões de código
 - testes automatizados
 - integração contínua
 - testes de regressão, etc.
- Controles para gerenciar mudanças nos requisitos ou no código, como um processo formal de solicitação de mudança (Change Request).

Conceitos Básicos - Métodos

- Marcos de referência que possibilitam administrar o progresso do software.
- Os marcos são pontos específicos no tempo durante o processo de desenvolvimento onde determinados objetivos devem ser alcançados.
- Por exemplo, a conclusão de uma fase de design pode ser um marco.
- Estes marcos ajudam a equipe a medir o progresso, verificar se estão no caminho certo e fazer ajustes conforme necessário.

Processo de Software

- Um **processo de software** é um método para desenvolver ou produzir software.
- A pesquisa em **processo de software** lida com métodos e tecnologias estimativas, suporte e melhoria das atividades de desenvolvimento de software.
- Define quem faz **o que, quando e como.**

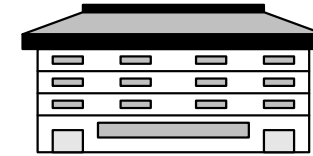
Processo de Software

- Modelagem: é uma técnica de engenharia aprovada e bem aceita
 - modelos de arquitetura de casas e de grandes prédios
 - modelos matemáticos a fim de analisar os efeitos de ventos e tremores de terra --> causas
- O que é um MODELO?
 - Simplificação da realidade
 - Detalhes podem ser estruturais (organização) e comportamentais (dinâmica e funcionamento)

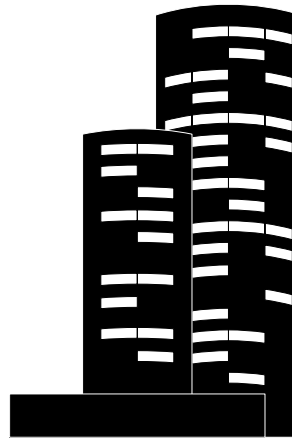
Modelo de Software



Casa de canhorro

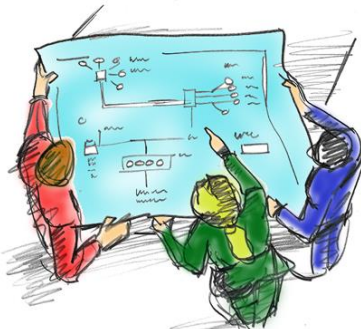


Casa



Arranha-Ceús

- Assim como em qualquer construção, a construção dos softwares precisa ser planejada
- Modelo: representação idealizada de um sistema a ser construído
- Maquetes de edifícios e de aviões e plantas de circuitos eletrônicos são apenas alguns exemplos de modelos.



Importância da modelagem

- Modelar atrasa o desenvolvimento?
- Modelar é tempo desperdiçado?
- Modelar é enrolação?
- Como você explicaria ao cliente que o custo do desenvolvimento do software pedido é o custo de modelagem (sem desenvolvimento) mais o custo de desenvolvimento?

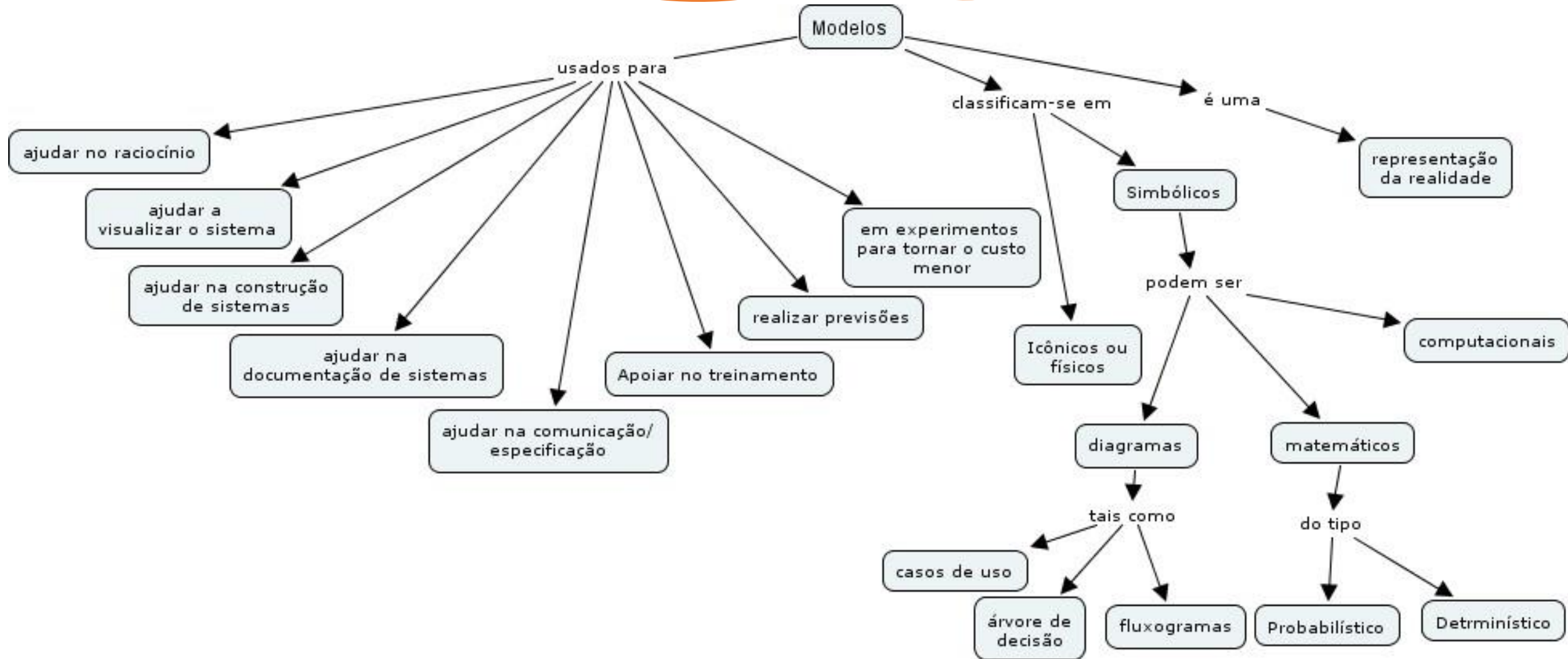
Importância da modelagem

- Gerenciamento da complexidade inerente ao desenvolvimento de software.
- Comunicação entre as pessoas envolvidas.
- Redução dos custos no desenvolvimento.
- Predição do comportamento futuro do sistema.

Importância da modelagem

- Modelos são construídos para permitir um melhor entendimento sobre o sistema que está sendo construído
 - especificar a estrutura e comportamento
 - guia para construção do sistema
 - documentam as decisões tomadas
- Nenhum modelo único é suficiente
 - conjunto de modelos independentes

Importância da Modelagem



Importância da modelagem



Importância da modelagem



Importância da modelagem



Importância da modelagem



Importância da modelagem



Objetivos da modelagem

- Auxiliar no processo de produção → produtos de alta qualidade, produzidos mais rapidamente e a um custo cada vez menor.
- Atributos: abstração, visibilidade, especificação, construção, confiabilidade, manutenibilidade, segurança e documentação.

Objetivos da modelagem

- Abstração
 - Melhor entendimento e maior compreensão
- Visualização
 - Visualização antecipada antes da implementação
 - Visões complementares do software
- Especificação
 - descrição precisa do que deve ser feito

Objetivos da modelagem

- Construção
 - geração automática com ferramentas baseadas em modelos
- Documentação
 - comunicação entre equipes na diferentes fases do ciclo de vida

Modelo x Processo

- Um modelo é algo teórico, um conjunto de possíveis ações.
- O processo deve determinar ações práticas a serem realizadas pela equipe como prazos definidos e métricas para se avaliar como elas estão sendo realizadas

Modelo + Planejamento = Processo

Modelo do processo de software

- Um conjunto de atividades fundamentais exigida para desenvolver um sistema de software
 - Especificação.
 - Projeto e implementação.
 - Validação.
 - Evolução.

Modelo do processo de software

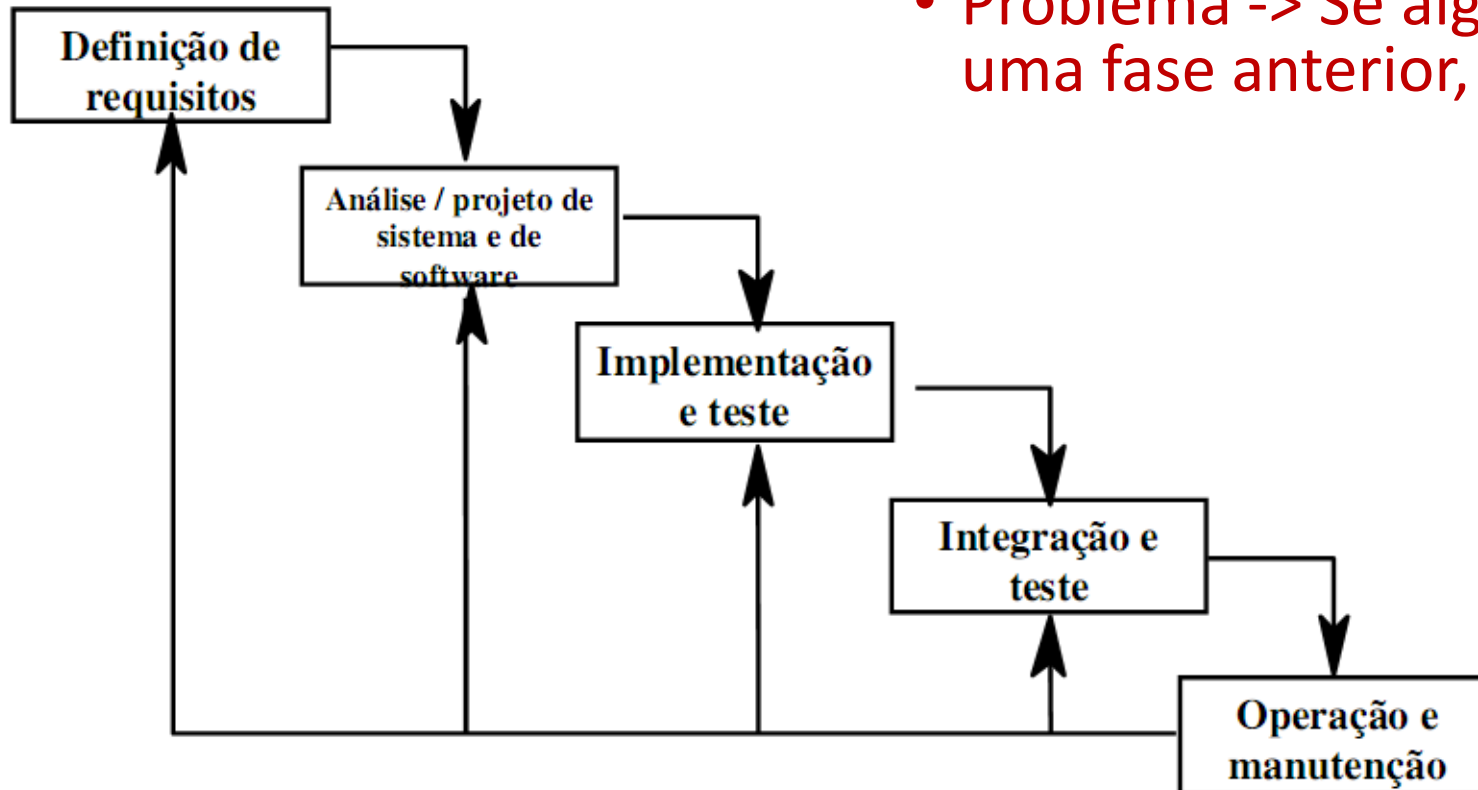
- Um modelo de processo de software é uma representação abstrata (e simplificada) de um processo de software.
- Representa uma descrição de um processo a partir de uma perspectiva particular.
- Exemplos
 - Workflow - sucessão de atividades
 - Fluxo de Dados - fluxo de informação
 - Papel/ação – representa os papéis das pessoas e as atividades pelas quais elas são responsáveis.

Modelo do processo de software

- Uma estratégia de desenvolvimento que englobe processos, métodos e ferramentas, e as fases de desenvolvimento...
- Exemplo:
 - Modelo Seqüencial (ciclo de vida clássico - cascata), Modelo de Prototipação, Modelos Evolutivos (Modelo Incremental e Espiral), Modelo de Métodos Formais, Orientado a reuso, etc.

Modelo em Cascata

- Método sistemático e sequencial
- O resultado de uma fase se constitui na entrada da outra.
- Cada fase é estruturada como um conjunto de atividades que podem ser executadas por pessoas diferentes, simultaneamente.
- Problema -> Se algo precisar ser alterado em uma fase anterior, pode ser custoso voltar.



Modelo de Prototipação

- Protótipo Descartável: o objetivo é compreender os requisitos do cliente e, a partir disso, desenvolver uma melhor definição de requisitos para o sistema.
 - Contrói-se versões simplificadas do sistema
 - o protótipo se concentra em fazer experimentos com partes dos requisitos que estejam mal entendidos
 - Ou seja, utilizado para entender e refinar requisitos
 - Pode ser descartado após seu uso ou evoluir para o sistema final

Modelo Evolutivo

- Abordagem baseada na idéia de desenvolver uma implementação inicial, expor o resultado ao comentário do usuário e fazer seu aprimoramento por meio de muitas versões.
- As atividade de desenvolvimento e validação são desempenhadas paralelamente, com um rápido feedback entre elas.

Modelo Evolutivo - Tipos

- Desenvolvimento Exploratório ou Incremental: trabalha-se junto com o cliente, a fim de explorar seus requisitos e entregar um sistema final.
 - o desenvolvimento se inicia com as partes do sistema que são mais bem compreendidas.
 - o sistema evolui com o acréscimo de novas características à medida que elas são propostas pelo cliente.
 - Entregue por partes a cada nova funcionalidade

Modelo Evolutivo - Espiral

- Desenvolvido para englobar as melhores características do ciclo de vida clássico e do paradigma evolutivo.
- São avaliados riscos explicitamente e são solucionados ao longo do processo.
- Processo é representado como uma espiral em lugar de ser representado como uma sequência de atividades

Modelo Evolutivo - Espiral



- Cada loop na espiral representa uma fase do processo de software.
- Não existem fases fixas.
- Engloba as melhores características do ciclo de vida clássico como o da Prototipação, adicionando um novo elemento: a análise de riscos

Modelo Evolutivo - Espiral

Coleta inicial dos requisitos e planejamento do projeto

Planejamento baseado nos comentários do cliente

Avaliação do cliente

Planejamento

Análise de risco

Para cada risco do projeto identificado em P é levada a cabo uma análise detalhada.

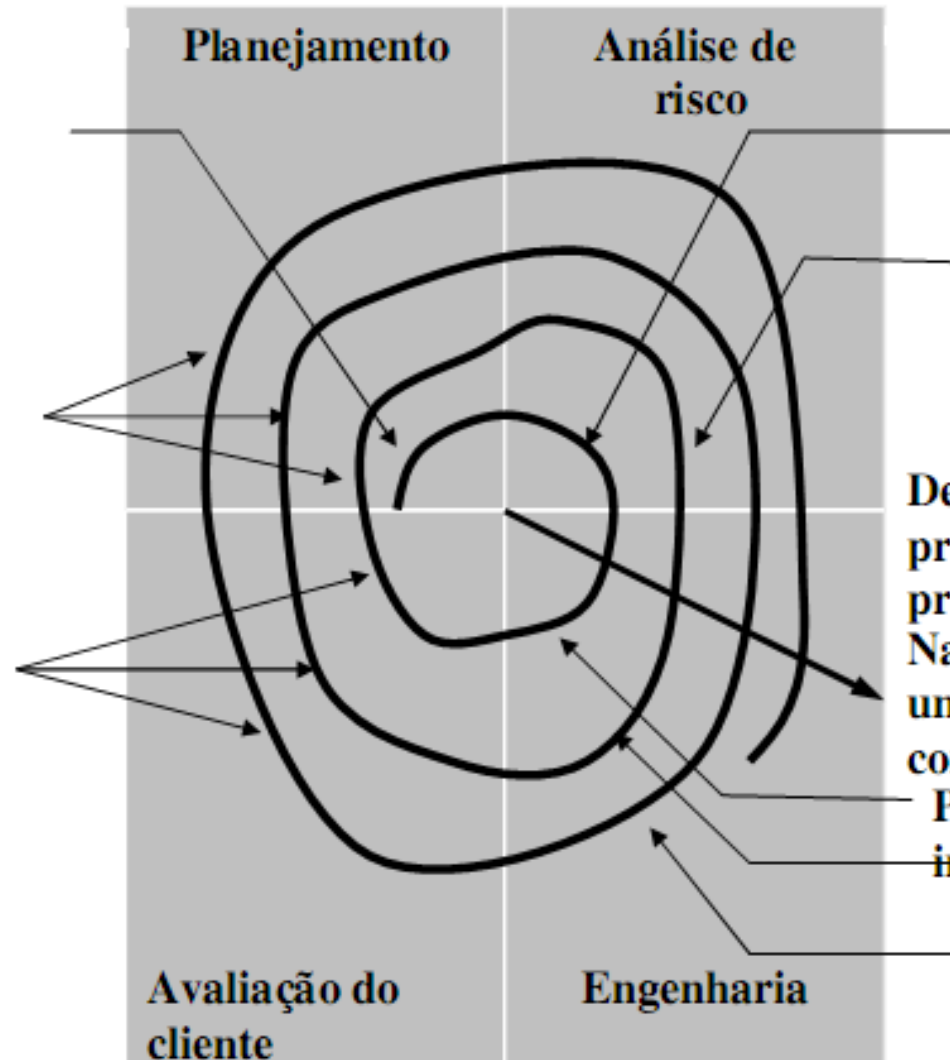
Decisão de prosseguir/não prosseguir
Na direção de um sistema concluído

Protótipo de software inicial

Avaliação do cliente

Engenharia

Sistema construído pela engenharia



Modelo de Métodos Formais

- Baseia-se na utilização de linguagens matemáticas rigorosas para especificação, desenvolvimento e verificação do software.
- É um método que busca a corretude do software desde sua especificação até sua implementação

Modelo Orientado a Reuso

- Em vez de desenvolver software do zero, busca-se reutilizar componentes ou sistemas existentes.
- Podem ser usadas abordagens como a engenharia de componentes ou a engenharia de linha de produtos.

Outros Modelos

- Desenvolvimento Ágil: Métodos como Scrum e XP se concentram em desenvolvimento iterativo, feedback constante e envolvimento direto do cliente.
- Desenvolvimento Orientado a Aspectos: Concentra-se em modularizar preocupações transversais (como log e segurança) que são difíceis de modularizar em paradigmas tradicionais.

Como escolher um Modelo de Processo de Desenvolvimento?

- Depende.....
 - do tipo de software a ser desenvolvido
 - dos requisitos
 - do tamanho da equipe
 - dos prazos
- Em muitos projetos modernos, é comum combinar características de diferentes modelos para se adequar às necessidades específicas do projeto.

Perguntas Comuns

- Qual a diferença entre processo de software e ciclo de vida?
 - Processo de software refere-se a todas as atividades, bem como relacionamentos, artefatos, ferramentas, papéis etc, necessárias para construir, entregar e manter um produto de software.
 - Já o ciclo de vida apresenta uma representação alto nível do processo de software executado (processo de software real) ou como deveria ser executado, ou seja, normalmente, ciclos de vida determinam as fases e o relacionamento entre as fases.

Perguntas Comuns

- O que são recursos?
 - Recursos relacionam-se com diversos componentes do processo de software, por exemplo, técnicas, métodos e ferramentas.

Perguntas Comuns

- O que são artefatos?
 - Artefato é um tipo de recurso produzido ou consumido em uma atividade. Nesse contexto, um artefato pode ser utilizado como uma entrada (matéria-prima) para uma determinada atividade e/ou como uma saída de uma atividade (resultado da execução de uma atividade).
 - Ex: código-fonte, código executável, manual de padrões, relatório de resultados, documento de requisitos, plano de trabalho, etc.

Perguntas Comuns

- O que são agentes?
 - Agentes ou atores são as entidades que executam atividades por intermédio de um papel.

Perguntas Comuns

- O que são papéis?
 - Papéis representam um conjunto de responsabilidades, obrigações, permissões e habilidades necessárias para executar uma atividade ou sub-atividade. Geralmente, papéis são desempenhados por agentes humanos. Um sinônimo de papel seria cargo ou função.
 - Uma atividade pode exigir diversos papéis para ser executada e um papel pode ser aplicado em diversas atividades.

Diagramas e Documentação

- Desenhos gráficos que seguem algum padrão lógico
- Coleção de elementos gráficos que possuem um significado predefinido
- Construídos sobre regras de notação bem definidas

Diagramas e Documentação

- Desenhos gráficos que seguem algum padrão lógico
- Coleção de elementos gráficos que possuem um significado predefinido
- Construídos sobre regras de notação bem definidas

Atividades de Fixação

Opa! Supresa !!!

Dúvidas?



Próxima Aula – Processo Unificado

Introdução

Características

Artefatos

Ciclo de Vida

Fluxos de
Trabalho

Atividade de
Fixação

Próxima Aula



Obrigado !





Apresentador

Thales Levi Azevedo Valente

E-mail:

thales.l.a.valente@gmail.com