

INSTITUTO FEDERAL
SANTA CATARINA

Processo de Desenvolvimento de Software

Felipe dos Passos Cardoso

Projeto Integrador II
Engenharia de Telecomunicações
Março 2020

Agenda

- **Motivação**
- **Atividades Típicas**
- **Participantes do processo**
- **Mecanismos Gerais**

Motivação

- **Atividade complexa**
- **Projetos que não são finalizados**
- **Causas do Insucesso (Standish Group)**
 - Reinicializações.
 - Excedente de custos.
 - Excedente de prazo.
 - Falta de características e funcionalidades.

Motivação

- **Causas do Sucesso (Standish Group)**
 - Envolvimento do usuário.
 - Suporte da gerência executiva.
 - Declaração clara dos requisitos.

Motivação

- Pesquisa (Standish Group)

TRADITIONAL RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

Processo de Desenvolvimento

- **Processo de Desenvolvimento de software**

- Definir, desenvolver, testar e manter.

- **Objetivos:**

- quais atividades a serem executadas
- como, quando e por quem
- pontos de controle
- padronizar a forma de desenvolvimento

Atividades Típicas

- **Levantamento de requisitos**
- **Análise de requisitos**
 - Prototipagem
- **Projeto**
- **Implementação**
- **Testes**
- **Implantação**

Levantamento de requisitos

- Compreensão do problema
- Objetivo: usuários e desenvolvedores tenham mesma visão do problema
- Definir as necessidades (Requisitos)
- Requisitos definidos a partir do **domínio do negócio**
- Estudo Exploratório: Leitura, observação, entrevistas com os usuários e **especialistas do domínio.**

Levantamento de requisitos

- O produto gerado é o **documento de requisitos**
- Seções do documento:
 - ***Requisitos Funcionais*** - funcionalidades do sistema
 - a. “... Permitir que cada professor realize o lançamento de notas das turmas ...”
 - b. “... Permitir que um aluno realize a sua matrícula nas disciplinas ...”

Levantamento de requisitos

- **Requisitos Não-Funcionais** - características de qualidade que o sistema deve possuir.
 - a. *Confiabilidade*: medidas quantitativas, como tempo médio entre falhas, recuperação, quantidade de erros.
 - b. *Desempenho*: tempo de respostas esperados.
 - c. *Portabilidade*: restrições sobre as plataformas de software e hardware. Grau de facilidade para mudar o sistema de plataforma.

Levantamento de requisitos

- d. *Segurança*: limitações sobre a segurança do sistema (acessos não-autorizados).
- e. *Usabilidade*: facilidade de uso, necessidade de treinamento dos usuários
- **Restrições** - restrições impostas sobre o desenvolvimento.
 - a. adequação a custos e prazos.
 - b. aspectos legais (licenciamento).
 - c. limitação sobre a interface com usuário.

Levantamento de requisitos

- Utilidade x requisitos
- Escrito para leitores técnicos e não técnicos.
- “O que o usuário necessita do novo sistema?”.
- Termo de consenso.
- Escopo do sistema.
- Características de volatilidade.
- Ordenação dos requisitos.

Análise de requisitos

- Estudo detalhado dos requisitos.
- Construção de modelos.
- Especificação de requisito.
- Estratégia de solução.
- “paralisa de análise”
- Validação e Verificação.

Análise de requisitos

- Modelos:
 - Modelo de casos de uso.
 - Modelo de classes de domínio.
 - Modelo de interações.
 - Modelo de estados.
 - Modelo de atividades.

Prototipagem

- Complemento à análise de requisitos.
- Esboço de alguma parte do sistema.
- Usado na validação.
- Dificuldades no entendimentos dos requisitos.
- Não substitui os modelos.

Projeto

- “Como” o sistema funcionará.
- Aspectos considerados:
 - Arquitetura do sistema.
 - Padrão da interface gráfica.
 - Linguagem de programação.
 - Banco de dados.
- Descrição computacional.

Projeto

- Projeto da arquitetura
 - Distribuir classes em subsistemas e seus componentes.
 - Distribuir componentes fisicamente.
 - Diagramas de implementação.
 - Arquiteto de software.

Projeto

- Projeto detalhado
 - Colaborações entre os objetos de cada módulo.
 - Projeto da interface.
 - Projeto de banco de dados.
 - Diagrama de classes.
 - Diagrama de casos de uso.

Projeto

- Modelos:
 - Modelo de classes de especificação.
 - Modelo de interação.
 - Modelo de estados.
 - Modelo de atividades.

Implementação

- Codificação do Sistema.
- Definição das classes de objetos.
- Modelo de classes de implementação.
- Uso de componentes preexistentes.

Testes

- Verificação do sistema construído.
- Relatório de testes.
- Integração dos módulos.

Implantação

- Sistema é empacotado, distribuído e instalado.
- Produção do manuais.
- Importação dos dados.
- Treinamento dos usuários.

Participantes do Processo

- **Gerentes de projeto**
- **Analistas**
- **Projetistas**
- **Arquitetos de Software**
- **Programadores**
- **Clientes**
- **Avaliadores de Qualidade**

Gerentes de Projeto

- Coordenação das atividades.
- Orçamento do Projeto:
 - Estimar tempo necessário.
 - Definir o processo.
 - Cronograma de execução.
 - Mão-de-obra especializada.
 - Recursos de Hardware e Software.

Gerentes de Projeto

- Acompanhamento das atividades.
- Verificar gastos de recursos.
- Estudar a factibilidade do sistema.
- Escalonar equipe de desenvolvimento.

Analistas

- Conhecer o **domínio do negócio**.
- Se comunicar com **especialistas do domínio**.
- Ponte entre profissionais.
- Capacidade de comunicação.
- Ética profissional.
- Participa de todas as etapas do desenvolvimento.

Projetistas

- Avaliar as alternativas de solução.
- Gerar especificação da solução computacional.
- Alguns Tipos:
 - Projetistas de interface.
 - Projetistas de rede.
 - projetistas de banco de dados.
- Adicionam aspectos técnicos aos modelos.

Arquitetos de Software

- Elaborar arquitetura do sistema.
- Decidir sobre subsistemas que compõem o sistema e as interfaces entre eles.
- Tomar decisões técnicas detalhadas (desempenho).
- Trabalha em conjunto com o Gerente.

Programadores

- Implementação do sistema.
- Participam da Implementação e Testes.
- Preocupação apenas com aspectos tecnológicos.

Clientes

- Para quem o sistema é construído.
- Cliente usuário:
 - Utilizará o sistema.
 - Normalmente é um **especialista do domínio**.
 - Interage com o Analista.

Clientes

- Cliente contratante:
 - Solicita o desenvolvimento.
 - Custeia o desenvolvimento e manutenção.
- Software para mercado de massas.
- Importância da participação do usuário final.

Avaliadores de Qualidade

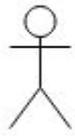
- Desempenho e qualidade.
- Assegurar a adequação do processo.
- Garantir padrões de qualidade.

Mecanismos Gerais

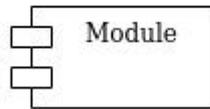
- **Estereótipos**
- **Notas explicativas**
- **Etiquetas (Tags)**

Estereótipos

- Estender o significado de um elemento.
- Devem ser documentados de forma que a semântica não tenha ambiguidade.
- Estereótipos gráficos:
 - Representado por um ícone.



Actor



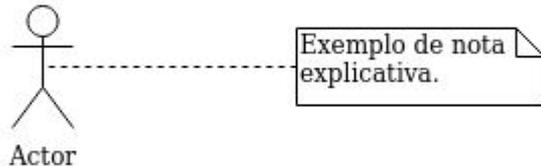
Traditional
server

Estereótipos

- Estereótipos de rótulos:
 - Representado por um nome delimitado por << >>
 - <<interface>>
 - <<controle>>
 - <<entidade>>

Notas Explicativas

- Definir informações do diagrama.
- Representadas graficamente por um retângulo com uma “orelha”



- Não modificam o elemento associado.
- Utilizar somente quando necessário.

Etiquetas (Tags)

- Definir propriedades para os elementos.
- Formas de definição:
 - { tag = valor }
 - { tag1 = valor1, tag2 = valor2 }
 - { tag }



Referências

- [1] Eduardo Bezerra. **Princípios de análise e projeto de sistemas com UML.** Campus, 2002.
- [2] Standish Group. **CHAOS Report 2015.** Disponível em:
https://wiki.sj.ifsc.edu.br/images/3/3b/CHAOSReport2015_rev.pdf