



Engenharia de Software

Aula 5

Profa. Maristela Weinfurter Teixeira

Organização da Aula

▪ Projeto de software

- Espectro de gerenciamento
- Estrutura de projeto
 - ✓ Metodologias e requisitos
 - ✓ Arquitetura
 - ✓ Interfaces
 - ✓ Padrões
 - ✓ Estimativas e riscos

Projeto de Software

▪ Espectro de Gerenciamento

O Gerenciamento do Projeto

▪ O que é um projeto?

- Um empreendimento com objetivo bem definido, que consome recursos e opera sob pressões de prazos, custos e qualidade

- Projetos tornam-se cada vez maiores e mais complexos

- Um projeto também pode ser definido como uma atividade multifuncional



- Gerentes de projetos são cada vez mais integradores do que especialistas técnicos

- Ou seja, projeto pode ser definido como:
 - o planejamento
 - a programação
 - o controle de uma série de tarefas integradas

- Tarefas que atinjam seus objetivos com êxito, para benefício dos participantes

- Gestão de projetos bem-sucedida exige:
 - planejamento e controle

- Tal gestão é dita horizontal, e traz consigo mais produtividade, eficiência e eficácia

- Até aqui vimos conceitos sobre gerenciamento para qualquer tipo de projeto



- E para gerenciar desenvolvimento de software, é diferente?

- Não, mas há particularidades...

- Precisamos entender:

- ✓ pessoas
- ✓ produtos
- ✓ processos
- ✓ projeto

- Pessoas

People-CMM (People Capability and Maturity Model para RH)

- ❖ Define que toda organização precisa aprimorar continuamente sua habilidade para atrair, desenvolver, motivar e organizar e reter a força de trabalho necessária para atingir os objetivos estratégicos de seus negócios

- Produtos

Antes de traçar qualquer projeto é importante estabelecermos:



escopo

soluções

restrições técnicas

restrições de gerenciamento

❖ Lembre-se:

Produto é desenvolvido para
uma ou um grupo de pessoas

Engenharia de Requisitos pode
ajudar muito

➤ Processos

A metodologia que iremos
utilizar para o desenvolvimento
de software

Atividades-tarefas

Pontos de controle

Artefatos de software

Pontos de garantia de qualidade

➤ Projeto

Há mais de 30 anos aplicam-se
projetos com planejamento e
controle, porém, em uma
pesquisa entre 1998 a 2004,
entre 250 grandes projetos,
constatou-se:



25% de projetos com sucesso

50% não cumpriram cronograma, custos e objetivos de qualidade

35% não obtiveram problemas sérios

Apesar dos números, evoluímos muito nos projetos, porém, ainda precisamos melhorar muito

- Pessoas
 - O sucesso de um projeto de software são pessoas

- Produzimos capital intelectual, então precisamos de:
 - especialistas
 - seleção da equipe
 - bom ambiente

**▪ Comprometidos:**

- gerentes
- líderes técnicos
- programadores
- clientes
- usuários finais

▪ Líderes:

- motivação
- organização
- ideias e inovação
- comunicação

Projeto de Software**▪ Estrutura, Metodologias e Requisitos****Projeto de Software**

- Estrutura
 - Metodologia
 - Engenharia de Requisitos
 - Projeto e Arquitetura
 - Projeto de Componentes
 - Projeto de Interfaces
 - Projeto Baseado em Padrões
 - Projeto WebApps

PS – Metodologia

- Metodologia
 - Clássico
 - Incremental
 - Evolucionário
 - Componentes
 - Processo Unificado
 - Desenvolvimento Ágil

PS – Engenharia de Requisitos

- Engenharia de Requisitos
 - Criar software pode ser desafiador, criativo e divertido



- Mas a tarefa de entendimento dos requisitos é ampla

Fases da Engenharia de Requisitos

Concepção

Levantamento

Elaboração

Negociação

Especificação

- Concepção
 - Pessoas + Viabilidade + Informações + Solução desejada + COLABORAÇÃO

- Levantamento
 - Problemas de escopo
 - Problemas de entendimento
 - Problemas de volatilidade (mudam)

- Elaboração
 - Criação e refinamento de cenários de usuários
 - Cada cenário pode ser analisado para extração de classes e serviços
 - ✓ Pode-se utilizar diagramas de casos de uso e classes, por exemplo

- Negociação
 - Usuários podem propor necessidades conflitantes
 - Conciliação de conflitos
 - Processo de negociação
 - Priorizações



- Especificação
 - Documento escrito
 - Gráficos
 - Modelos matemáticos formais
 - Cenários de uso
 - Protótipo
 - Combinação de todos

- SRS é um tipo de especificação de requisitos criado para estabelecer um modelo-guia de especificação de requisitos

✓ SRS:

1. Introdução
2. Descrição geral
3. Características do sistema
4. Requisitos de interfaces externas
5. Outros requisitos funcionais
6. Outros requisitos
7. Apêndices

Projeto de Software

- **Estrutura, Projeto de Arquitetura e Componentes**

PS – Projeto de Arquitetura

- Esse projeto reflete a estrutura de dados e componentes de programa para construção do sistema

- A arquitetura não é o software operacional, mas a representação para análise, alternativas e redução de riscos para construção de software



- Qual a importância?
 - Facilitar a comunicação entre as partes
 - Trabalhar a engenharia do software
 - Modelo compreensível da estrutura do software

- Estilos de arquitetura
 - Centrada em dados
 - Centrada em fluxo de dados
 - Centrada em chamadas e retornos
 - Orientada a objetos
 - Em camadas
 - Em padrões

PS – Projeto de Componentes

- Projeto de Componentes é um conjunto completo de componentes (código)

- Visão pode ser orientada a objetos (serviços)

- Visão pode ser tradicional (elementos funcionais e estruturas de dados)

- Visão pode ser orientada a processos



Projeto de Software

▪ Interfaces

PS – Interfaces

- Projeto de Interfaces
 - Nas primeiras décadas da era computacional, usabilidade não era preocupação dominante, hoje é essencial

- Cria um meio de comunicação efetivo entre o ser humano e o computador

- Regras importantes
 - Deixe o usuário no comando
 - Reduza carga de memória do usuário
 - Interface consistente

- Usabilidade
 - É uma medida do quanto um sistema facilita o aprendizado, ajuda os aprendizes a se lembrarem do que aprenderam, reduz a probabilidade de erros, permite que sistemas se tornem eficientes e criam satisfação de uso do sistema

- Há processos, técnicas e ferramentas para apoiar o projeto de interfaces



- Hoje há uma variedade de interfaces: mobile, WebApps, entre outras

- O projeto de interface inicia com uma série de tarefas. Definição de perfis de usuários, aplicação de casos de uso, elaboração de tarefas, objetos e análise de fluxos de trabalho

Projeto de Software

▪ Padrões e WebApps

PS – Padrões e WebApps

- Padrão é uma regra de três partes que expressa a relação entre contexto, problema e solução

- Tipos
 - Criacionais
 - Estruturais
 - Comportamentais

- ✓ Criacionais
 - Fábrica abstrata
 - Métodos de fábrica
 - Construtor
 - Protótipo
 - Único



✓ Estruturais

- Adaptador
- Agregação
- Ponte
- Composição (composite)
- Container
- Proxy
- Tubos e filtros

✓ Comportamentais

- Cadeia de responsabilidades
- Comandos
- Escutador de eventos
- Interpretador

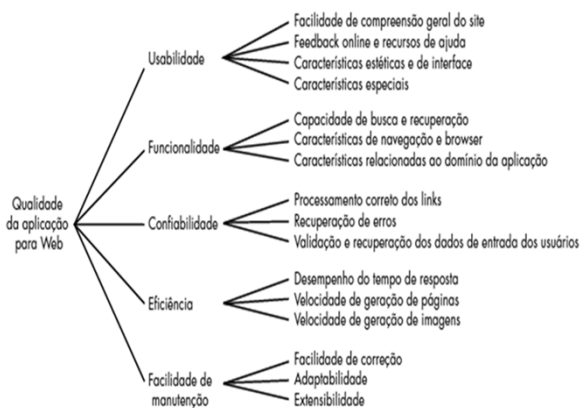
- Iterador
- Mediador
- Visitante
- Visitante atendimento único
- Visitante hierárquico

- Tarefas no projeto
 - Examinar modelo de requisitos
 - Desenvolver hierarquia de problemas
 - Determinar padrões cfe linguagem e domínio do problema
 - Critérios de qualidade

- Não esqueça de documentar os padrões, o que pode ser feito através de descritivo de tabelas

- Padrões WebApps
 - Arquitetura de informações
 - Navegação
 - Interação
 - Apresentação
 - Funcionais

- Segurança
- Disponibilidade
- Escalabilidade
- Tempo para colocação no mercado



- Simplicidade
- Consistência
- Identidade
- Robustez
- Navegabilidade



- Apelo visual
- Compatibilidade
- Projeto de conteúdo



- Não se pode esquecer da fase de gestão da qualidade, com testes de avaliação e verificação

Projeto de Software

▪ Estimativas e Riscos

PS – Estimativas e Riscos

- Estimativas de custo e esforço de software não são parte de uma ciência exata

- Fatores humanos
 - Técnicas
 - Ambientais
 - Políticos
- ✓ São difíceis de se medir em conjunto

- Estimativas confiáveis podem ser baseadas em:
 - projetos similares completos
 - técnicas de decomposição para gerar estimativas de custo e esforço
 - modelos empíricos para estimativas de custo e esforço

- Técnicas de decomposição
 - Dimensionamento Lógica Fuzzy
 - Dimensionamento de pontos de função
 - Dimensionamento de componentes-padrão
 - Dimensionamento de alteração



- Não importa o quão sofisticada seja a técnica, deve passar por uma verificação cruzada com outra abordagem
- Bom senso e experiência prevalecem

PS – Estimativas e Riscos (Método LOC)

Função	LOC estimado
Interface de usuário e recurso de controle	2.300
Análise geométrica bidimensional	5.300
Análise geométrica tridimensional	6.800
Gerenciamento de base de dados	3.350
Recursos de visualização da computação gráfica	4.950
Função de controle de periféricos	2.100
Módulos de análise do projeto	8.400
Linhas de código estimadas	33.200

- LOC – linhas de código com medida-chave
 - Não acomoda linguagens não procedurais
 - Nível de detalhe difícil
 - Penaliza programas bem projetados

PS – Estimativas e Riscos (Método FP)

Valor do domínio	Esti-Saídas	Estimativa	FB
	mativa	Consulta computada	Peso computado
Número de entradas externas	20 24 30	24	4 97
Número de saídas externas	12 15 22	16	5 78
Número de consultas externas	16 22 28	22	5 88
Número de arquivos lógico internos	4 4 5	4	10 42
Número de arquivos de interface externos	2 2 3	2	7 15
Contagem total			320

Fator	Valor
Backup e recuperação	4
Comunicações de dados	2
Processamento distribuído	0
Desempenho crítico	4
Ambiente operacional existente	3
Entrada de dados on-line	4
Transações de entrada em múltiplas telas	5
Arquivos mestres atualizados on-line	3
Complexidade dos valores dos domínios de informação	5
Complexidade do processamento interno	5
Código projetado para reutilização	4
Conversão/instalação no projeto	3
Instalações múltiplas	5
Aplicação projetada para alteração	5
Fator de ajuste de valor	1,17

Por fim, é obtido o número estimado de FP:

$$FP_{\text{estimado}} = \text{contagem total} \times [0,65 + 0,01 \times \Sigma(F_i)] = 375$$

PS – Estimativas e Riscos

- FP – Orientada à função
 - Determina tamanho e complexidade do software sob perspectiva do usuário
 - Quantifica a funcionalidade proporcionada ao usuário a partir do desenho lógico



- Oferece ferramenta para dimensionar aplicações
- Quantifica custo, esforço e tempo
- Calcula índices de produtividade e qualidade
- Normalização para comparar software

- Outras estimativas
 - Baseada em processos
 - Casos de uso
 - Modelos empíricos
 - COCOMO II
 - OO
 - Métodos ágeis

▪ **Riscos?**

- Eles existem para quaisquer tipos de projetos, para Engenharia de Software não é diferente

- Gestão de Riscos: auxilia nas ações que suportam as equipes de software no entendimento do gerenciamento de incertezas

- O risco, independentemente dele ocorrer ou não, deve ser previsto em todo projeto, em especial, em projeto de software

- Riscos de projeto ameaçam o planejamento do projeto, em potencial: orçamento, cronograma, pessoal, recursos, clientes e requisitos



- Riscos técnicos: ameaçam a qualidade e a data de entrega do software a ser produzido. Problemas em potencial de projeto, implementação, interface, verificação e manutenção

- Riscos de negócio: ameaçam a viabilidade do software a ser criado, bem como o projeto ou o produto

Componentes				
Categoria		Desempenho	Suporte	Custo
Catastrófico	1	Falha em satisfazer o requisito resultaria em falha do missão		A falha resulta em aumento de custo e atrasos no cronograma com valores previstos que excedem \$ 500 mil
	2	Degradação significativa até não cumprimento do desempenho técnico	Software que não responde com agilidade ou que é difícil de dar suporte	Dificuldades financeiras significativas, provável estouro no orçamento
Crítico	1	Falha em atender o requisito degradará o desempenho do sistema até um ponto no qual o sucesso da missão é questionável		Falha resulta em atrasos operacionais e/ou aumento de custos com valores estimados entre \$ 100 mil e \$ 500mil
	2	Alguma redução no desempenho técnico	Pequenos atrasos nas modificações de software	Alguma falta de recursos financeiros, possíveis estouros de orçamento
Marginal	1	Falha em atender o requisito resultaria na degradação de missão secundária		Custos, impactos e/ou atrasos de cronograma recuperáveis com valores estimados de \$ 1 mil a \$ 100 mil
	2	De mínima a pequena redução no desempenho técnico	Suporte responsivo de software	Recursos financeiros suficientes
Negligenciável	1	Falha em atingir o requisito criaria inconveniência ou impacto não operacional		Erro resulta em pequeno impacto no custo e/ou cronograma com valor esperado de menos de \$ 1 mil
	2	Nenhuma redução do desempenho técnico	Software facilmente suportável	Possível sobre no orçamento

Referências de Apoio

- SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.
- PRESMAN, Roger. **Engenharia de Software**. 7. ed. Porto Alegre: Bookman, 2011.

- PFLEEGER, Shari L. **Engenharia de Software: teoria e prática**. 2. ed. São Paulo: Prentice Hall, 2004.
- PAGE-JONES, Meilir. **Fundamentos do desenho orientado a objeto com UML**. São Paulo: Pearson, 2001.