AULA 05 – Projeto e Gerenciamento de Projeto de Software

Introdução:

Em nossa quinta aula, falaremos sobre projeto e gerenciamento de projeto de software. Exploraremos dentro do contexto gerenciamento de projeto, as principais fases de um projeto: planejamento e início do projeto, execução, controle e finalização. Como em qualquer tipo de projeto, o projeto de software segue uma metodologia para garantir sucesso tanto quanto o produto final quanto ao processo.

Contextualizando:

Gerenciar projetos é uma tarefa árdua, e gerenciar projetos de desenvolvimento de software é algo desafiador. Um projeto de software tem por finalidade traduzir necessidades dos *stakeholders* em aplicações, sistemas para web, sistemas convencionais, entre outros tipos de software. É um processo de tradução de necessidades humanas para algo tecnológico que será utilizado novamente por pessoas.

PROJETO DE SOFTWARE

Projeto de software é um empreendimento com objetivo bem definido, que consome recursos e ocorre geralmente com prazos, custos e qualidade bem definidos. Projetos tornam-se cada vez maiores e mais complexos composto por atividades multifuncionais. O perfil de um gerente de projetos deve ser mais integrador que especialista técnico. Um projeto prevê algumas etapas importantes: planejamento, programação e controle de tarefas integradas.

PROJETO DE SOFTWARE

O gerenciamento de projetos possui tarefas com objetivos para benefício de todos os participantes. Uma boa gestão de projetos exige um bom planejamento e um excelente controle. A gestão de projetos é dita horizontal e traz consigo mais produtividade, eficiência e eficácia. Para que o gerenciamento seja eficiente e eficaz, ele deve abordar de forma clara três itens importantes: pessoas, produtos e projeto.

PESSOAS

Segundo a People-CMM (People Capability and Maturity Model), toda organização necessita aprimorar continuamente suas habilidades para atrair, desenvolver, motivar, organizar e reter a força de trabalho necessária para se atingir objetivos estratégicos em um negócio.

PRODUTO

O desenvolvimento de qualquer produto necessita a sequência de algumas tarefas importantes: escopo, soluções, restrições técnicas e restrições de gerenciamento. Todo produto é desenvolvido para pessoas. No caso de gerência de projetos a engenharia de requisitos pode ser um grande trunfo.

PROCESSO

Pode ser considerado como uma metodologia que será utilizada para o desenvolvimento de um produto ou software. Processos envolvem atividades-tafefas, pontos de controle, artefatos de software e pontos de garantia de qualidade.

Por mais que tenhamos evoluídos nos últimos 30 anos, projetos de desenvolvimento de software continuam apresentando vários tipos de problemas. Entre 1998 e 2004 constatou-se entre 250 grandes projetos que apenas 25% dos projetos são realizados com sucesso, enquanto que 50% não cumpriram cronograma, custos e objetivos de qualidade. Enquanto isto, 35% não obtiveram problemas tão sérios, porém, tiveram muitas manutenções até seu pleno funcionamento. Uma das grandes questões no gerenciamento de projetos de software encontra-se no fato de que produzimos capital intelectual, e para isto precisamos de bons especialistas, uma seleção bem elaborada para constituição da equipe e um bom ambiente de trabalho. Outra forma de melhorar a garantia da qualidade de um projeto encontra-se no comprometimento de gerentes, líderes técnicos, programadores, clientes e usuários finais. Líderes devem trabalhar questões de motivação, organização, ideias e inovação e uma excelente comunicação.

ESTRUTURA DE PROJETO DE SOFTWARE

A estrutura de um projeto de software pode ser compreendida como um conjunto de etapas ou tarefas a serem executadas, através de uma metodologia utilizando-se engenharia de requisitos, projetos de arquitetura, componentes, interfaces e padrões. Dentro as metodologias mais comuns encontram-se a Clássica, Incremental, Evolucionária, de componentes, o processo unificado e métodos ágeis.

Criação de um software é algo desafiador, criativo e divertido, mas a tarefa de entendimento dos requisitos além de ampla é crucial para o sucesso ou fracasso do projeto. A engenharia de requisitos deveria ser tratada como uma etapa extremamente importante e subdividida em assuntos relacionados à concepção, ao levantamento, à elaboração, à negociação e à especificação dos requisitos. Na concepção estuda-se a viabilidade da solução através da coleta de informações e necessidades dos stakeholders (pessoas que irão utilizar direta ou indiretamente o software). É no levantamento dos requisitos que são identificados problemas de escopo, entendimento do problema e volatilidade dos requisitos (requisitos que mudam!!!). Ao passar-se para fase de elaboração, utiliza-se toda coleta de requisitos para então efetuar-se a criação e refinamento de cenários de usuários, que podem ser modelados como casos de uso da UML. Uma vez a especificação elaborada, passa-se para fase de negociação com os usuários, os quais proporão outras necessidades ou avaliarão como corretos os requisitos. Neste momento há necessidade de se trabalhar conciliação de conflitos e priorizações das necessidades. Próxima fase inicia-se a especificação definitiva dos requisitos através de documentos escritos, gráficos, modelos matemáticos, cenários, protótipos ou a combinação de vários ou todos. Para a especificação pode-se utilizar uma SRS (especificação de requisitos criada para estabelecer um modelo-guia de especificações de requisitos). Uma SRS é composta por vários níveis:

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

PROJETO DE ARQUITETURA

O projeto da arquitetura reflete a estrutura de dados e componentes do programa para construção de um sistema. A arquitetura não é o software operacional, mas a representação para análise, alternativas e redução de riscos para construção de software. Ela facilita a comunicação entre as partes envolvidas e cria um modelo compreensível da estrutura do software. Há vários estilos de arquitetura, entre eles:

1- Centrada em dados
2- Centrada em fluxo de dados
3- Centrada em chamadas e retornos
4- Orientadas a objetos
5- Em camadas
6- Em padrões.

PROJETO DE COMPONENTES

O projeto de componentes é um conjunto completo de códigos (de programas) dentro de uma visão orientada a objetos (serviços), visão tradicional (elementos funcionais e estruturas de dados) ou visão orientada a processos.

PROJETO DE INTERFACES

O projeto de interfaces, tão importante quanto código de programa bem implementado, deve seguir algumas premissas básicas:

1- Deixar o usuário no comando do software

2- Reduzir a carga de memória do usuário
3- Criar interfaces consistentes.

Outros dois aspectos importantes na criação das interfaces são usabilidade e acessibilidade. A 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. Assim como em qualquer fase do projeto, o projeto de interfaces congrega processos, técnicas e ferramentas como apoio. Hoje há uma variedade de interfaces: mobile, WebApps, entre tantos outros. Nesta etapa é necessário a definição de dos perfis de usuários, aplicação de casos de uso, elaboração de tarefas, objetos e análise de fluxos de trabalho para suporte à uma construção eficiente de interfaces.

PROJETO DE PADRÕES E WEBAPPS

Padrão é uma regra de três partes que expressa a relação entre contexto, problema e solução. São três tipos principais:

1- Criacionais
    a. Fábrica abstrata
    b. Métodos de fábrica
    c. Construtor
    d. Protótipo
    e. único
2- Estruturais e
    a. Adaptador
    b. Agregação
    c. Ponte
    d. Composição
    e. Container
    f. Proxy
    g. Tubos e filtros
3- Comportamentais

a. Cadeia de responsabilidades
b. Comandos
c. Escutador de eventos
d. Interpretador
e. Iterador
f. Mediador
g. Visitante
h. Visitante atendimento único
i. Visitante hierárquico.

As tarefas no projeto de padrões são:

1- Exame do modelo de requisitos
2- Desenvolvimento da hierarquia de problemas
3- Determinação de padrões conforme linguagem e domínio do problema
4- Adoção de critérios de qualidade.

Dentre as tarefas desta etapa não é importante salientar a necessidade da documentação dos padrões, que pode ser através de forma descritiva.

Em padrões para WebApps, é importante desenvolver uma arquitetura de informações, de navegação, interação, de apresentação e de funcionalidades. Tal tipo de aplicação requer ainda a preocupação com segurança, disponibilidade das interfaces e tempo, escalabilidade e tempo para colocação no mercado. Projetos para Web devem ser também simples, consistentes, possuir identidade, robustez e excelente navegabilidade.

ESTIMATIVAS E RISCOS

Estimativas de custo e esforço de software não fazem parte de uma ciência exata. Envolvem fatores humanos, técnicas, questões ambientais e políticas das organizações. São baseadas em:

1- Projetos similares completos

2- Técnicas de decomposição para geração de estimativas de custos e esforço
    a. Dimensionamento lógica fuzzy
    b. Dimensionamento de pontos por função
    c. Dimensionamento de componentes-padrão
    d. Dimensionamento de alteração
3- Modelos empíricos para estimativa de custo e esforço.

O importante não é a sofisticação da técnica, mas a verificação e cruzamento com outras abordagens. Bom senso e experiência sempre prevalecem.

Trabalharemos alguns conceitos das estimativas LOC (linhas de código com medida-chave) e FP (Orientação à função ou Function Point).

LOC – linhas de código com medida-chave:

- Não acomoda linguagens não-procedurais
- Nível de detalhe difícil
- Penaliza programas bem projetados.

A figura 1 explicita um exemplo de uso de 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*

Figura 1: LOC – Linhas de Código com Medida-Chave

FP – Function Point:

- 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.

A figura 2 demonstra o uso da FP.

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:

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

Figura 2 – Function Point

Além das técnicas LOC e FP citadas anteriormente, temos também:

1- Baseada em processos
2- Casos de uso
3- Modelos empíricos
4- COCOMO II
5- Orientada a Objetos

6- Métodos Ágeis.

Até agora falamos muito sobre estimativas, técnicas, ferramentas e formas de melhorarmos nossas previsões e planejamento. Agora trabalharemos com o tema RISCO.

Riscos existem para quaisquer tipos de projetos, sendo assim, com a engenharia de software não seria diferente. Gestão de riscos auxilia nas ações que suportam as equipes de software no entendimento de incertezas. O risco independentemente de ocorrer ou não, deve ser previsto em qualquer tipo de projeto e em especial em projeto de software.

Riscos ameaçam o planejamento do projeto, tanto a nível de orçamento, quanto de cronograma, recursos humanos e materiais, clientes e requisitos. Riscos técnicos ameaçam a qualidade e adata 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 e a ser criado, bem como o projeto ou o produto.

Pesquisa

Desenvolva uma pesquisa sobre Function Point, sua aplicação e certificação como apoio ao gerenciamento de projetos de desenvolvimento de software.

Trocando Ideias

O projeto de software é uma fase que ocorre imediatamente antes da implementação dos códigos (programação). Compartilhe com seus colegas o que cada um compreendeu e se alguém possui experiência em algum tópico que compartilhamos neste encontro. Senão, busque mais informações sobre padrões, projetos de interface, estimativa e riscos no projeto de software.

Síntese

Nessa aula tivemos uma visão geral sobre algumas fases do projeto de software bem como características sobre o gerenciamento de projeto de software. Observamos que após a fase de elaboração e identificação do domínio e problema em questão,

podemos tomar diversos caminhos até chegarmos no código (programas) de fato. O importante é adotarmos metodologias, técnicas e utilizarmos ferramentas que nos apoiem na construção de um projeto de software mais maduro, documentado e planejado. Isto evitará no futuro construirmos um software que não será utilizado por estar fora do escopo, das necessidades reais de nossos usuários ou então por conter falhas funcionais e de interface.

Compartilhando

Busque informações sobre projetos de software que obtiveram sucesso e também que obtiveram problemas. Tente buscar informações sobre características de cada projeto. Entreviste amigos, familiares ou conhecidos que já atuem no desenvolvimento de software ou acompanhe notícias relacionadas a problemas com software. Há vários casos de software desenvolvidos por órgãos públicos por exemplo, e que de alguma forma afetam nossa vida.

Autoavaliação

- Comente sobre Pessoas, Processos e Produto dentro do contexto de projeto de software.
- Fale sobre a estrutura de projeto de software.
- Como uma SRS auxilia no projeto de software?
- Quais os benefícios de se adotar padrões e projeto de arquitetura de software?
- Como o projeto de interface pode afetar um projeto de software?
- Quais peculiaridades num projeto para Web?
- Quais as diferenças entre LOC e FP?