

CI 221

DISCIPLINA: Engenharia de Software
AULA NÚMERO: 5

DATA: ____/____/____
PROFESSOR: Andrey

APRESENTAÇÃO

Nesta aula serão apresentados e discutidos os conceitos de Gestão de projetos de software, riscos de software, planejamento de um projeto de software e elaboração de um cronograma.

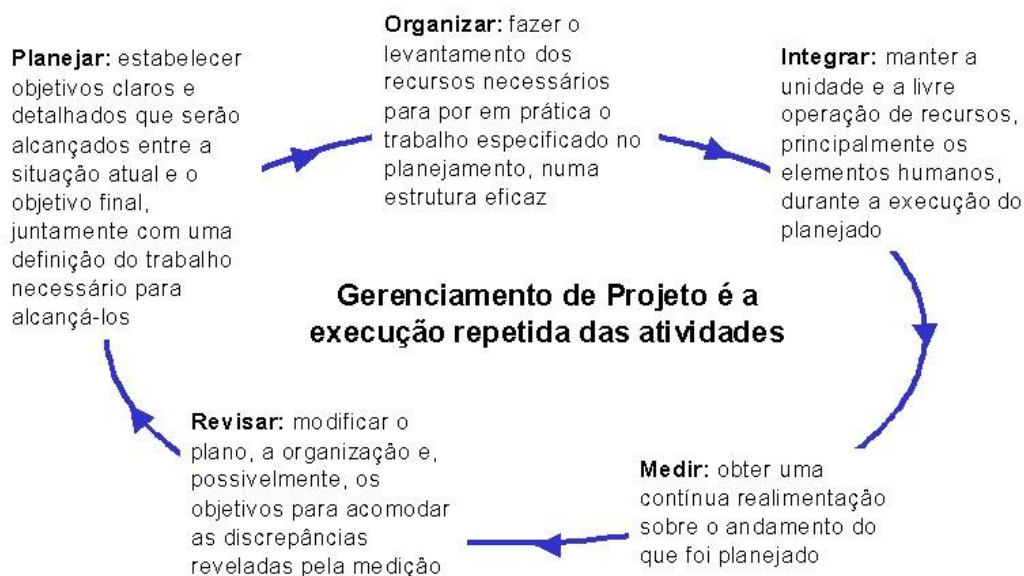
DESENVOLVIMENTO

Introdução

“Visitei dezenas de instalações comerciais, tanto boas como ruins, e observei um grande número de gerentes de processamento de dados, tanto bons quanto ruins. Frequentemente, observei como esses gerentes lutavam inutilmente com projetos que constituíam um verdadeiro pesadelo, espremidos por prazos de entrega inexecutáveis, ou entregavam sistemas que irritavam seus usuários e prosseguiram devorando enormes parcelas de tempo de manutenção.” (Page-Jones, 85)

Gestão de Projeto

A gestão de um projeto envolve algumas atividades a serem realizadas repetidamente até que o projeto tenha sido completado:



A gestão efetiva de projetos de software focaliza 4 Ps:

- Pessoal
- Produto
- Processo
- Projeto

Pessoal

Interessados

- **Gerentes seniores.** Definem os aspectos do negócio
- **Gerentes de projeto.** Devem planejar, motivar, organizar e controlar os profissionais que fazem o trabalho de software.
- **Profissionais** (desenvolvedores). Fornecem aptidões técnicas para fazer a engenharia de um produto ou aplicação.
- **Clientes.** Especificam os requisitos para o software submetido à engenharia.
- **Usuários finais.** Interagem com o software depois que ele é liberado para uso.

Líderes de equipe. Modelo de liderança:

- **Motivação.** Habilidade de encorajar o pessoal técnico a produzir no melhor da sua capacidade.
- **Organização.** Habilidade de moldar processos existentes permitindo que o conceito inicial seja traduzido em um produto final.
- **Idéias ou inovação.** Habilidade de encorajar o pessoal a criar e se sentir criativo.
- **Solução de problemas.** Diagnosticar os aspectos técnicos e organizacionais relevantes, estruturar sistematicamente uma solução ou motivar outros profissionais a desenvolvê-la e quando for necessário mudar o rumo.
- **Identidade gerencial.** Um bom gerente deve assumir o encargo do projeto.
- **Realização.** Deve premiar a iniciativa e a realização.
- **Influência e construção de espírito de equipe.** “Ver” as pessoas, perceber sinais verbais ou não e reagir às suas necessidades.

Equipe de Software

Modelos de equipes (paradigmas):

- **Fechado.** Estrutura hierárquica rígida
- **Aleatório.** Estrutura fraca e dependente da iniciativa individual. É vantajosa em projetos de inovação ou desenvolvimento tecnológico.
- **Aberto.** Combina aspectos dos paradigmas fechado e aleatório.
- **Síncrono.** Apoiado na compartimentalização natural do problema. Organiza as equipes para trabalhar em partes do problema. Pouca comunicação entre as equipes.
- **Equipes ágeis.**

Características para um bom desempenho:

- Os membros da equipe devem confiar uns nos outros
- A distribuição de aptidões deve ser adequada ao problema
- Estrelas podem ter que ser excluídas da equipe, se a coesão tiver que ser mantida

Produto

Escopo do software

O escopo do software é definido pelas seguintes questões:

- **Contexto.** Como o software a ser construído se encaixa no contexto de um sistema maior, do produto ou do negócio?
- **Objetivos da informação.** Que objetos de dados visíveis para o cliente são produzidos como saída e quais são necessários para a entrada?
- **Função e desempenho.** Que funções o software desempenha para transformar a entrada na saída? Existem características de desempenho que devem ser tratadas?

O escopo do projeto de software não deve ser ambíguo e deve ser inteligível para os níveis gerenciais e técnicos.

Decomposição do software

A decomposição do problema (particionamento ou elaboração do problema) é uma atividade que se situa no núcleo da análise dos requisitos. São descompostos:

- a funcionalidade que precisa ser entregue e
- o processo que será usado para entregá-la.

Processo

O gerente de projeto deve decidir qual o modelo de processo mais adequado para:

- o cliente e o pessoal que executará o trabalho
- as características do produto
- o ambiente da equipe de projeto

Deve ser criado um plano de completo que reflita as tarefas de trabalho necessárias para preencherem as atividades de arcabouço.

Plano de Projeto

O planejamento de projeto de software abrange 5 atividades:

- estimativa de recursos, tempo e custos.
- cronogramação
- análise de riscos
- planejamento de gestão da qualidade
- planejamento de gestão de modificação

Modelo de Documento de Visão do Sistema

Este modelo de documento de visão do Sistema tem o objetivo de servir como roteiro para a elaboração do 1o. Trabalho prático da disciplina.

1. Introdução

O propósito deste documento é coletar, analisar e definir necessidades e funcionalidades em alto nível do <<Nome do Sistema>>. Seu foco está nas capacidades necessárias para os stakeholders, e para os usuários-alvo, e no porquê dessas necessidades existirem. Os detalhes de como o <<Nome do Sistema>> preenche essas necessidades estão nas especificações de casos de uso.

1.1 Escopo

[Uma breve descrição do escopo do Sistema a ser desenvolvido; que projetos estão associados a ele e listagem de tudo que é afetado ou influenciado por este documento].

Exemplo: Este documento vem fundamentar as especificações do sistema <<Nome do Sistema>>.

1.2 Definições, Acrônimos e Abreviações.

Exemplo: Para facilitar o entendimento de termos técnicos, há um documento específico, o Glossário. *[Link para o Glossário do sistema]*

1.3 Referências

[Esta subseção deve prover uma lista completa de todos os documentos referenciados em qualquer lugar da Visão do Sistema. Especifique as origens de onde as referências podem ser obtidas e coloque links nelas].

1.4 Visão Geral do Documento

[Esta subseção deve descrever o que o restante da Visão do Sistema contém e explicar como o documento está organizado, listando as seções que se encontram a seguir].

O <<Nome do Projeto>> é um produto escalável que...

2 Posicionamento

2.1 Oportunidade de Negócios

[Brevemente descrever a oportunidade de negócios que é meta deste projeto]

2.2 Descrição do Problema

[Frase resumida sobre o problema a ser resolvido por este projeto. O seguinte formato pode ser usado:].

O problema é	(descrever o problema)
E afeta	(os stakeholders afetados pelo problema).
O impacto disto é	(qual é o impacto do problema).
Uma solução de sucesso seria	(listar alguns benefícios chave da solução de sucesso).

2.3 Descrição da Posição do Sistema

[Frase resumida sobre a posição que o produto se destina a alcançar no mercado. O seguinte formato pode ser usado:].

Para	(cliente alvo)
Quem	(frase da necessidade ou oportunidade)
O (nome do produto)	É um (categoria do produto)
Que	(frase de benefícios chave – o que é – razão convincente para comprar.).
Diferente de	(alternativa de competição primária)
Nosso produto	(frase de diferenciação primária)

3 Descrição dos Usuários e Stakeholders

[Para fornecer produtos e serviços que reúnem as necessidades reais de seus stakeholders e usuários, é

necessário identificar e envolver todos eles como parte do processo de Modelagem de Requisitos. Você deve também identificar os usuários do sistema e certificar-se de que todos os stakeholders estão representados adequadamente.].

3.1 Resumo de Stakeholders

[Listagem resumida de todos os stakeholders identificados:]

Nome	Representa	Papel
Nome do tipo de stakeholder	Breve descrição de que ele representa com respeito ao desenvolvimento. Qual sua área de atuação dentro da organização.	Ex. Garantir características financeiras para o projeto; estimular a inovação tecnológica. Ex. Garantir a integridade da informação.

3.2 Resumo de Usuários

[Resumo de todos os tipos de usuários identificados:]

Nome	Descrição
Nome do tipo de usuário	Breve descrição do que eles representam em relação ao sistema

3.3 Ambiente do Usuário

- *Quais plataformas de sistema estão em uso hoje? E as plataformas futuras?*
- *Que outras aplicações estão em uso? Sua aplicação precisa se integrar com ela?*
- *Quais as versões dos aplicativos que estarão instalados na mesma máquina?*

3.4 Competição e Alternativas

[Identificação de produtos alternativos que o stakeholder percebe como disponíveis. Essas podem incluir comprar um produto concorrente, construir uma solução do mesmo nível ou simplesmente manter o status-quo. Listar as opções conhecidas dos concorrentes que existem, ou talvez tornar disponível. Incluir as maiores forças e fraquezas de cada concorrente como percebíveis pelo stakeholder.].

Exemplo: Dois softwares recentemente apareceram de concorrentes fabricantes que são similares ao <<Nome do Sistema>>.

<Produto Concorrente Um>

<Outro Produto Concorrente>

4 Visão Geral do Produto

[Esta seção fornece uma visão em alto nível das capacidades do produto, interface para outras aplicações e configurações de sistema.].

4.1 Perspectiva do Produto

[Esta subseção deve por o produto na perspectiva de outros produtos relacionados e no ambiente do usuário. Se o produto é independente e totalmente independente. Se o produto é um componente de um sistema maior, então esta subseção deve relacionar como estes subsistemas interagem e devem identificar as interfaces relevantes entre os sistemas. Uma maneira fácil de mostrar os maiores

componentes do maior sistema, interconexões e interfaces externas por via de um diagrama de blocos].

Exemplo: Inicialmente, <<Nome do Sistema>> será apresentado como uma aplicação Windows que suporta vários ambientes...

4.2 Objetivos do Projeto

[Listagem dos principais objetivos que o projeto deverá cumprir].

Exemplo: aumentar a visibilidade da existência dos suprimentos para o cliente atual.

Exemplo: aumentar a satisfação de usuários novos e atuais.

Exemplo: aumentar as vendas de suprimentos...

4.3 Suposições e Dependências

[Listar cada um dos fatores que afetam as funcionalidades listadas neste documento (próxima seção). Por exemplo, uma suposição pode expressar que um sistema operacional específico estará disponível para o hardware designado para o produto software. Se o sistema operacional não está disponível, o documento de visão precisará mudar.]

5 Funcionalidades do Produto

[Listar e brevemente descrever as funcionalidades do produto. Funcionalidades são capacidades de alto nível do sistema que são necessárias para entregar benefícios para os usuários e que atendam as necessidades dos mesmos. Cada funcionalidade é um serviço externamente desejado que tipicamente requer uma série de entradas para arquivar o resultado desejado.]

[Como este documento é revisado por várias pessoas envolvidas, o nível de detalhes deve geralmente ser suficiente para o entendimento de todos, entretanto, os detalhes disponíveis devem ser suficientes para fornecer a toda equipe do projeto informações necessárias para a elaboração do modelo de caso de uso.]

Exemplo: Os principais benefícios que esta aplicação precisa endereçar estão descritos abaixo.

5.1 <Funcionalidade Um>

5.2 Exemplo: Enviar Dados para empresa

FEAT – Enviar Dados para empresa – O cliente deve ter a opção de configurar o sistema para enviar dados históricos para a empresa. Não somente a empresa usa este histórico para analisar usos, mas para entregar serviços para o usuário baseado nos dados.

6 Precedência e Prioridade

[Define a prioridade das diferentes funcionalidades do sistema].

As prioridades devem estar definidas no atributo da funcionalidade, ao criar-se uma FEAT pelo Requisite Pro – Requirement Properties, porém, é importante fornecer uma listagem destes atributos nesta seção para facilitar sua compreensão e leitura.

Exemplo:

Funcionalidade	Prioridade	Dificuldade	Status
Funcionalidade Um	☐ Alta ☐ Média ☐ Baixa	☐ Alta ☐ Média ☐ Baixa	☐ Proposto ☐ Aprovado ☐ Incorporado

7 Requisitos de Documentação

[Esta seção descreve a documentação que deve ser desenvolvida para suportar a instalação da aplicação com sucesso.]

7.1 Manual do Usuário

[Descreve o propósito e conteúdo do Manual do Usuário, tais como estrutura, tamanho desejado, nível de detalhamento, necessidades de índice, glossário de termos, tutoriais x manual de referência de estratégias, etc. Restrições de formato e impressão também devem ser identificadas.]

7.2 Online Help

[Muitas aplicações fornecem um sistema de help on-line para ajudar o usuário. A natureza destes sistemas é única para desenvolvimento da aplicação ou como eles combinam aspectos de programação (hyperlink, etc) com aspectos de escrita técnica (organização, apresentação). Devemos considerar que muitas vezes o desenvolvimento do sistema de help on-line é um projeto dentro do projeto que traz benefícios no gerenciamento de escopo e das atividades planejadas.]

7.3 Guias de Instalação, Licenças, Configuração, Arquivo Leia-Me

[Um documento que inclui instruções de instalação e guias de configuração é importante para oferecer uma solução completa. Também, um arquivo Leia Me é tipicamente incluído como um componente padrão. O Leia Me pode incluir um “O que há de novo nesta versão”, e uma discussão de assuntos de compatibilidade com versões anteriores. A maioria dos usuários também aprecia documentação definindo bugs conhecidos e soluções para contornar o bug no arquivo Leia-Me].

7.4 Etiquetando e Empacotando

[Esta seção define as necessidades e tipos de etiquetas a serem incorporadas dentro do código. Exemplos incluem notas de copyright e patentes, logos corporativos, ícones padronizados e outros elementos gráficos, etc.]

ATIVIDADE

1. Quais são os tipos de equipes de desenvolvimento?
2. O que define o Escopo de um produto de software?

BIBLIOGRAFIA BÁSICA

CARVALHO, A. M. B. R.; CHIOSSI, T. C. S.. Introdução à Engenharia de Software. Editora da Unicamp. 2001. São Paulo.

PRESSMAN, R. S.. *Engenharia de Software*. Makron Books. 1995

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.. *UML guia do usuário*. Editora Campus. 2000.

BEZERRA, E.. *Princípios de Análise e Projeto de Sistemas com UML*. Ed. Campus. 2003.