

CI 221

DISCIPLINA: Engenharia de Software  
AULA NÚMERO: 5B

DATA: \_\_\_\_/\_\_\_\_/\_\_\_\_  
PROFESSOR: Andrey

## APRESENTAÇÃO

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

## DESENVOLVIMENTO

### 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

Antes de começar um projeto o gerente deve estimar o trabalho a ser feito, os recursos necessários e o tempo que irá decorrer do início ao fim do projeto.

A estimativa de custos, cronograma e recursos exige experiência e uma boa base de informações históricas.

O risco da estimativa é o grau de incerteza nas estimativas qualitativas estabelecidas.

A variabilidade dos requisitos de software implica em uma instabilidade no custo e no cronograma.

O planejamento provê um arcabouço para o projetista fazer estimativas razoáveis de recursos, custos e cronograma.

### Tarefas para o planejamento:

- Estabeleça o escopo do projeto;
- Determine a viabilidade;
- Analise riscos;
- Determine os recursos necessários
  - Humanos
  - Softwares reusáveis
  - Ambientais (ambiente de desenvolvimento)
- Estime custo e esforço
  - Decomponha o problema
  - Desenvolva duas ou mais estimativas usando tamanho FP ou PCU
- Desenvolva um cronograma

- Conjunto de tarefas
- Rede de tarefas
- Use uma ferramenta para fazer um cronograma
- Rastreamento

## **Recursos**

### Recursos Humanos

- Aptidões necessárias (posição na organização e especialidade)
- Esforço de desenvolvimento (pessoa - mês)

### Recursos de Software

- Componentes de prateleira
- Componentes de experiência plena
- Componentes de experiência parcial
- Componentes novos

### Recursos de Ambiente

- Ambiente de engenharia de software
- Softwares de desenvolvimento
- Ferramentas de planejamento
- Ferramentas de gerenciamento
- Ferramentas de apoio
- Ferramentas de análise e projeto (CASE)
- Ferramentas de programação
- Ferramentas de testes
- Ferramentas de manutenção
- Frameworks

### Dicas para se fazer uma estimativa

1. adie a estimativa o mais que puder
2. use técnicas de decomposição simples
3. baseie as estimativas em projetos semelhantes
4. use um modelo empírico para estimativa

## **Análise de Riscos**

Um risco é um problema em potencial, que pode ou não ocorrer.

Riscos em Engenharia de Software :

- No futuro do ambiente interno e externo ... mercado, tecnologia, economia, governo

**UNIVERSIDADE FEDERAL DO PARANÁ - UFPR**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

- nas mudanças do ambiente, nos requisitos dos clientes, em ambientes tecnológicos, equipamentos, política econômica e tecnológica, leis e programas de apoio a P&D, orçamentos.
- nas decisões: O que produzir? Com que métodos e ferramentas? Com que equipe? Com que nível de qualidade? Com qual orçamento?

Na análise de riscos são feitas as seguintes tarefas:

- Identificar o risco,
- avaliar a probabilidade que ele ocorra,
- estimar seu impacto,
- priorizar os riscos mais importantes e
- estabelecer um plano de contingência para resolver, administrar ou mitigar os riscos mais importantes.

Tipos de riscos:

- Riscos de projeto (riscos de que o desenvolvimento não ande de acordo com as expectativas) : identificam problemas orçamentários, de cronograma, de pessoal, de recursos, de clientes e de requisitos.
- Riscos técnicos (riscos de não conseguir construir uma solução que atenda as necessidades): identificam problemas de projeto, implementação, interface, manutenção, ambigüidade nas especificações, obsolescência, e tecnologia de “ponta”.
- Riscos de negócio (riscos de que a solução perca o interesse econômico)
- Riscos conhecidos: aqueles que podem ser descobertos com uma avaliação cuidadosa do plano de projeto, ambiente, etc.
- Riscos previsíveis: extrapolados de experiência em projetos anteriores. (rotatividade de pessoal, má comunicação com o cliente, etc.)
- Riscos imprevisíveis: extremamente difíceis de identificar antecipadamente.

Checklist de riscos conhecidos e previsíveis:

- tamanho do produto;
- impacto do negócio;
- características do cliente;
- definição do processo;
- ambiente de desenvolvimento;
- tecnologia para a construção;
- tamanho e experiência da equipe.

Componentes e fatores de risco:

- Risco de desempenho. Risco do produto não atender aos requisitos;
- Risco de custo. Risco do orçamento ser mantido;

- Risco de suporte. Risco do software não ser fácil de corrigir, adaptar e aperfeiçoar
- Risco de cronograma: risco de não conseguir cumprir os prazos

Impacto dos riscos

1. negligível;
2. marginal,
3. crítico e
4. catastrófico.

### **Tabela de Riscos**

<i>Riscos</i>	<i>Categoria</i>	<i>Probabilidade</i>	<i>Impacto</i>	<i>RMMM</i>
A estimativa de tamanho pode ser significativamente baixa	Tamanho do produto	60 %	2	
Usuários finais resistem ao sistema	Impacto do negócio	40 %	3	
...				

Atenuação Monitoração e gestão de Risco

Uma estratégia para lidar com riscos deve considerar 3 pontos:

- evitar o risco
- monitorar o risco
- gerenciar risco e planejar para contingência.

### **Elaboração de um Cronograma**

Segundo (Carvalho; Chiossi, 2001), a programação de projeto tem como objetivo organizar as atividades de desenvolvimento em uma seqüência coerente. O objetivo dessa organização é equilibrar recursos de pessoal, hardware e software, de maneira que sejam usados da melhor forma possível.

Depois de estimados os recursos necessários para o projeto de software, o cronograma do projeto especifica como e quando esses recursos devem estar disponíveis. Além disso, deve-se especificar, também, quem será responsável pelas atividades do ciclo de vida do sistema. Segundo De-Marco [DEM 82], a determinação de um cronograma para o projeto de software pode ser vista a partir de duas perspectivas:

- 1) Uma data final para a entrega do sistema já foi estabelecida de forma irrevogável. Nesse caso, o esforço deverá ser distribuído dentro desse espaço de tempo.

**UNIVERSIDADE FEDERAL DO PARANÁ - UFPR**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

2) Limites cronológicos aproximados são discutidos, mas a data final para a entrega é estabelecida pela equipe de engenharia de software. O esforço é distribuído para se tirar o melhor proveito dos recursos, e uma data final é definida após cuidadosa análise.

Segundo (Carvalho; Chiossi, 2001), a fixação de prazos para projetos tenta responder a uma série de perguntas. Por exemplo: como se relaciona o tempo cronológico com o esforço humano? Quais tarefas e paralelismos devem ser esperados? Quais marcos de referência podem ser usados para mostrar o progresso? Como o esforço é distribuído ao longo do processo de engenharia de software?

Para elaborar um cronograma, é necessário que o esforço de desenvolvimento seja decomposto em um conjunto de tarefas distintas. Para cada uma destas tarefas é feita uma estimativa de duração da tarefa. Além disso são determinadas as dependências entre as tarefas. Algumas destas atividades podem ser executadas em paralelo, enquanto que algumas podem depender do término de outras tarefas.

A figura abaixo representa um conjunto de atividades determinadas para o desenvolvimento de um sistema de vendas. Nesta figura estão representadas em cada uma das linhas, as tarefas, sua descrição, duração estimada e dependências entre as tarefas.

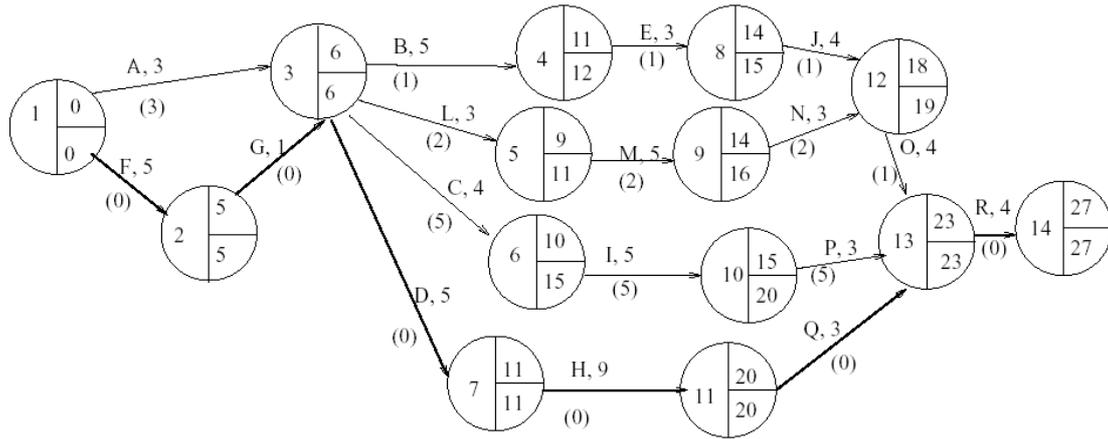
Tarefa	Descrição	Semanas	É precedida por
A	1-3 Criar telas	3	nenhuma
B	3-4 Implementar cadastrar cliente	5	A e G
C	3-6 Implementar estoque matéria-prima	4	A e G
D	3-7 Implementar estoque produto	5	A e G
E	4-8 Implementar verificar crédito	3	B
F	1-2 Criar banco de dados	5	nenhuma
G	2-3 Inicializar base de dados teste	1	F
H	7-11 Implementar guia de busca	9	D
I	6-10 Implementar comprar matéria-prima	5	C
J	8-12 Implementar controlar pedidos	4	E
L	3-5 Implementar controlar contas a receber	3	A e G
M	5-9 Implementar compor custos de produto	5	L
N	9-12 Implementar emitir fatura	3	M
O	12-13 Integrar subsistema vendas	4	J e N
P	10-13 Integrar subsistema estoque	3	I
Q	11-13 Integrar subsistema produção	3	H
R	13-14 Realizar o teste do sistema	4	O, P e Q

**Tarefas para implementação, integração e teste do sistema.**

O diagrama Pert/CPM, pode ser utilizado para representar o cronograma referente aos dados da Tabela acima. A figura abaixo representa todas as informações da tabela, além das folgas, tempo mais cedo e tempo mais tarde de cada atividade. Pelo diagrama é possível saber quais são as atividades do caminho crítico (F, G, D, H, Q, R).

Essas atividades têm folga igual a zero; assim, elas não podem ser atrasadas para que o projeto seja realizado em 27 semanas.

**UNIVERSIDADE FEDERAL DO PARANÁ - UFPR**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**



**Pert/CPM para o sistema de venda de sementes**

Usando-se a tabela das atividades e o diagrama PERT/CPM pode-se gerar um diagrama de barras ou gráfico de Grant para expor de maneira diferente as relações entre recursos e tarefas. Para cada atividade, esse diagrama indica a data prevista de início e de término e a pessoa responsável pela atividade.

O diagrama de barras pode servir para mostrar aspectos como:

- especificar a duração de cada tarefa (atividade),
- o que será feito e o executor de cada tarefa.
- controlar o projeto marcando-se o tempo estimado e o tempo gasto em cada tarefa.
- registrar o acompanhamento do projeto, acrescentando-se ao diagrama um outro tipo de barra, que representa as datas de início e término reais da atividade.

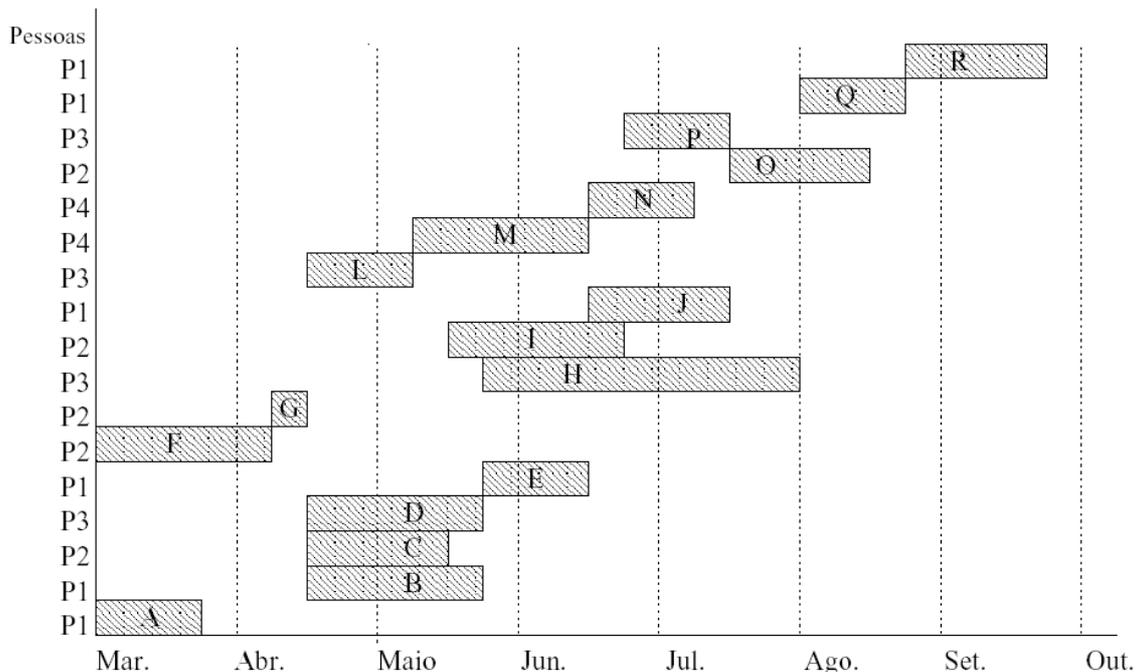


Diagrama de barras para o sistema de venda de sementes.

## ATIVIDADE

1. Quais são as tarefas para o planejamento
2. Quais os tipos de recursos que devem ser estimados
3. Identifique os riscos para o seu projeto da disciplina.
4. Crie uma tabela de riscos para estes riscos.
5. Quais são os componentes de risco?
6. Descreva os tipos de risco.

## 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.