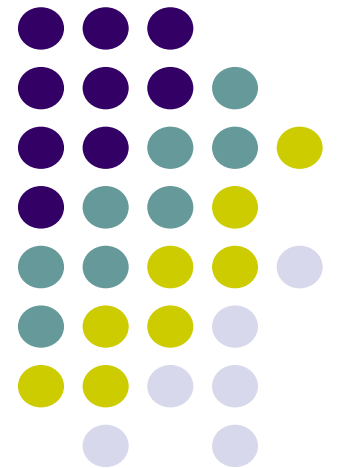
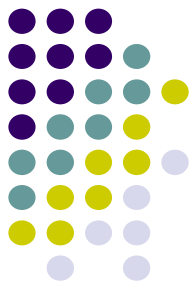


Gerência de Projeto: Planejamento e Estimativas

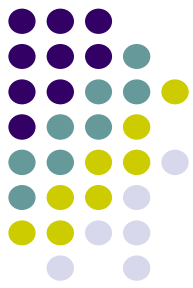


O Processo de Gerência de Projetos



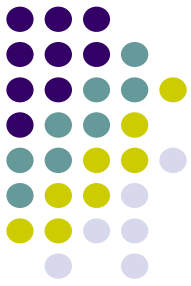
- Abrange todo o processo de desenvolvimento de software (do começo ao fim).
- Objetivo:
 - Fornecer uma compreensão do escopo do trabalho que será desenvolvido, dos riscos, dos recursos necessários (custo), das tarefas que serão executadas, dos prazos, do esforço despendido e da programação a ser seguida (planejamento).

Problemas da Gerência de Projetos de Software



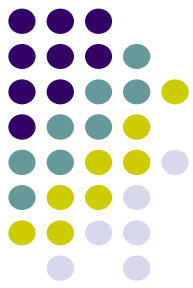
- Não existe um processo padrão de desenvolvimento.
 - Ainda não sabemos exatamente que processo de desenvolvimento utilizar para desenvolver cada tipo de software.

Problemas da Gerência de Projetos de Software



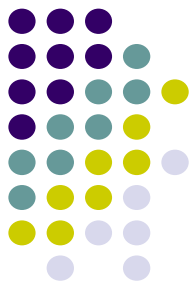
- O software é um produto intangível.
 - Atrasos na programação de seu desenvolvimento não são vistos claramente.

Problemas da Gerência de Projetos de Software



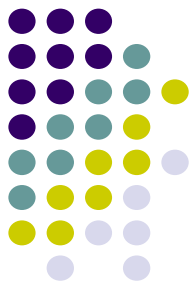
- É difícil usar experiências anteriores no desenvolvimento de novos produtos.
 - Os projetos normalmente têm poucas características em comum.
 - Rápidas mudanças tecnológicas tornam as experiências anteriores ultrapassadas.

O Papel do Gerente de Projetos



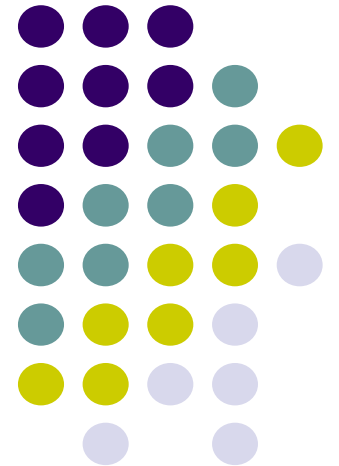
- Varia dependendo da organização e do produto de software a ser desenvolvido.
- Planejar o projeto:
 - Definir o escopo do software.
 - Estimar seu tamanho.
 - Estimar o esforço necessário para seu desenvolvimento.
 - Distribuir o esforço entre as atividades.
 - Determinar os recursos necessários para a execução do projeto.
 - Definir cronogramas.
 - Avaliar os riscos do projeto.

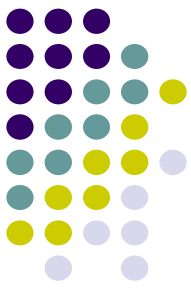
O Papel do Gerente de Projetos



- Controle do Projeto:
 - Fazer com que o planejamento seja executado.
 - Monitorar o progresso do projeto.
- Selecionar profissionais.
 - Profissionais pouco experientes podem precisar ser usados por restrições no orçamento.
 - Profissionais experientes podem não estar disponíveis.

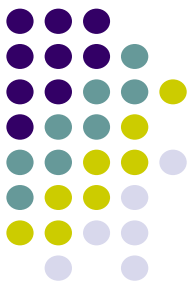
Planejamento do Projeto





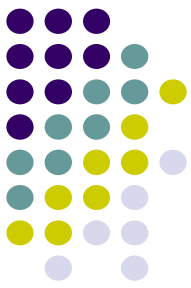
Planejamento do Projeto

- Planejamento do projeto é um processo iterativo que só se completa quando o software é entregue.



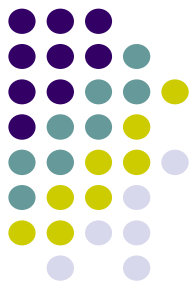
Planejamento do Projeto

- O planejamento se inicia com uma estimativa das restrições que afetam o projeto.
 - Data de entrega desejada, profissionais disponíveis, orçamento disponível, etc.
- Estimativas de tamanho do projeto e distribuição das funções são então realizadas.



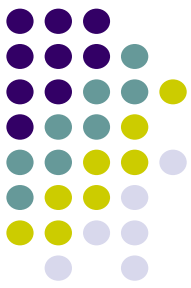
Planejamento do Projeto

- São definidos os milestones e as entregas previstas em cada milestone.
- Milestone é um ponto final de alguma atividade do processo de desenvolvimento. Representam, normalmente, o final de cada etapa do projeto.
- Entregas são resultados do projeto que são entregues ao cliente.



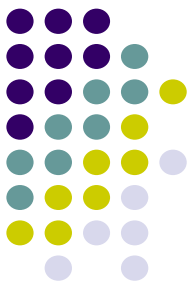
Planejamento do Projeto

- Até que se conclua o projeto, deve-se:
 - Elaborar programação para o projeto.
 - Iniciar atividades programadas.
 - Após algum tempo (2 ou 3 semanas), analisar o progresso do projeto e registrar as discrepâncias.
 - Revisar as restrições do projeto.
 - Atualizar a programação.
 - Renegociar restrições e entregas.
 - Se problemas surgirem, encontrar alternativas técnicas para desenvolver o produto dentro das restrições e da programação.



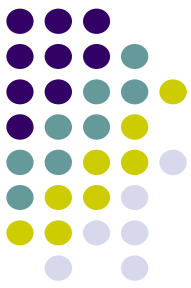
O Plano do Projeto

- Documento que resume os recursos disponíveis para o projeto, o trabalho a ser desenvolvido e a programação para executar o trabalho.



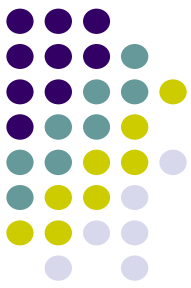
O Plano do Projeto

- Estrutura do plano:
 - Introdução: Descreve os objetivos do projeto e as restrições (orçamento, prazo, etc.) que afetam o projeto.
 - Organização do Projeto: Descreve como a equipe de desenvolvimento está organizada, quem são as pessoas envolvidas e qual é o papel de cada uma delas.
 - Análise de Riscos: Descreve os possíveis riscos do projeto, a probabilidade deles ocorrerem e as estratégias de redução de riscos propostas.



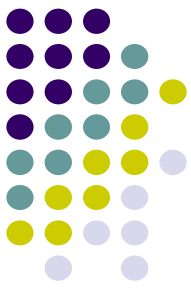
O Plano do Projeto

- Estrutura do plano:
 - Requisitos de Recursos de Hardware e Software: Descreve o hardware e o software requeridos para desenvolver o projeto.
 - Organização do Trabalho: Organiza o projeto em atividades e identifica os milestones e as entregas associadas a cada atividade.



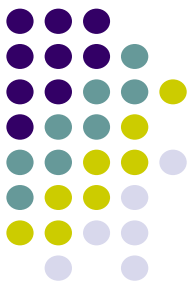
O Plano do Projeto

- Estrutura do plano:
 - Programação: Descreve as dependências entre as atividades, o tempo necessário para alcançar cada milestone e a alocação de pessoas para cada atividade.
 - Mecanismos de Monitoração: Descreve os relatórios de gerenciamento que devem ser produzidos e quando eles devem ser produzidos.



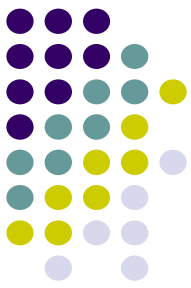
Organizando Atividades

- Quebre o processo de desenvolvimento em atividades.
 - Estudo de viabilidade, Análise de Requisitos, Construção do Protótipo.
- Estabeleça milestones onde for necessário.
 - Em cada milestone, deve ser produzido um relatório de progresso.
- Defina o que deve ser entregue ao cliente ao final de cada atividade.



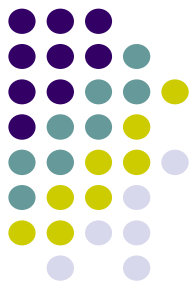
Elaborando a Programação

- Separar o trabalho total envolvido no projeto em atividades separadas.
- Julgar o tempo necessário para completar cada atividade.
 - Algumas atividades podem ser executadas em paralelo.
 - O uso da força de trabalho deve ser otimizado.
 - Evitar situações onde o projeto inteiro fique aguardando a conclusão de uma atividade.
 - Considerar um tempo para problemas não previstos.



Elaborando a Programação

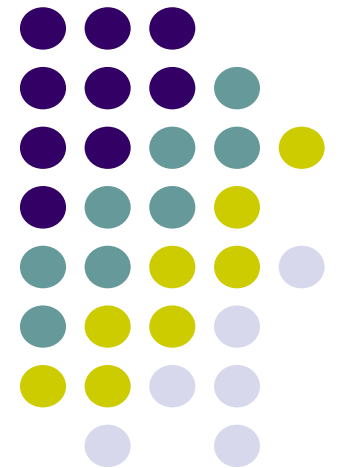
- Estimar os recursos necessários para completar cada tarefa.
 - Recursos humanos, recursos de hardware e software.

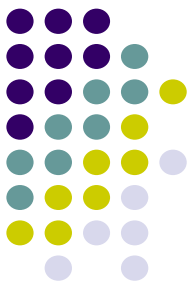


Elaborando a Programação

- A programação é normalmente representada através de gráficos que mostram as atividades, suas dependências e os recursos associados.
 - Redes de atividades, Gráficos de PERT e Gráficos de Gantt.
 - Use ferramentas para construir estes gráficos.

Estimativas

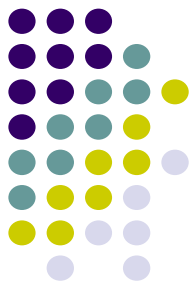




Tipos de Métricas

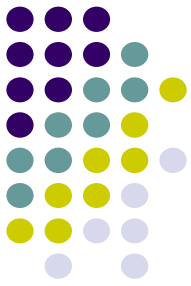
- Existem dois tipos de métricas que podem ser usadas para fazer estimativas:
 - Métricas orientadas ao tamanho.
 - Métricas orientadas à função.

Métricas Orientadas ao Tamanho

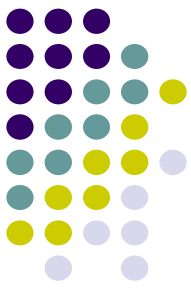


- Baseiam-se no tamanho da saída de uma atividade.
 - Número de linhas de código (KLOC).
 - A forma de calcular o número de linhas de código pode variar de uma organização para outra.
 - O número de linhas de código varia com a linguagem de programação usada.
 - Programas escritos em linguagens de baixo nível são maiores do que os programas escritos em linguagens de alto nível.
 - Número de erros encontrados.
 - Número de páginas documentadas.

Métricas Orientadas à Função



- Estão relacionadas à funcionalidade geral do software entregue.
- A principal proposta para medir funcionalidade são os pontos de função.



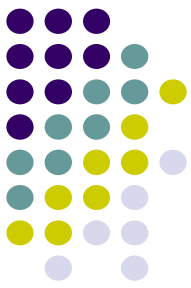
Pontos de Função

- São adequados aos sistemas onde há predominância de operações de entrada e saída.
- O número total de ponto de função de um programa é obtido medindo-se as seguintes características:
 - Entradas e saídas externas.
 - Interações com o usuário.
 - Interfaces externas.
 - Arquivos usados pelo sistema.

Pontos de Função



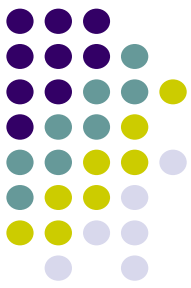
Parâmetro de medida	Contagem	Fator de Ponderação				
		Simplex	Médio	Complexo		
Número de entradas do usuário	<input type="text"/>	x 3	4	5	=	<input type="text"/>
Número de saídas do usuário	<input type="text"/>	x 4	5	7	=	<input type="text"/>
Número de consultas do usuário	<input type="text"/>	x 3	4	6	=	<input type="text"/>
Número de arquivos	<input type="text"/>	x 7	10	15	=	<input type="text"/>
Número de interfaces externas	<input type="text"/>	x 5	7	10	=	<input type="text"/>
Contagem – total	—————→					<input type="text"/>



Pontos de Função

$$FP = \text{contagem total} \times [0,65 + 0,01 \times \text{SOMA}(F_i)]$$

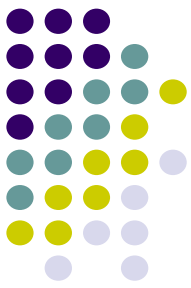
- Contagem total é obtida através da tabela.
- As constantes foram definidas empiricamente.
- F_i ($i = 1$ a 14) são valores de ajuste de complexidade.



Ajustes de Complexidade

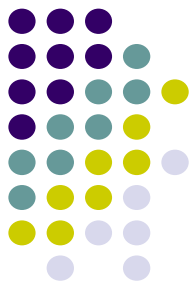
- Escala de pontuação (0 a 5):
 - 0: Sem influência.
 - 1: Incidental.
 - 2: Moderado.
 - 3: Médio.
 - 4: Significativo.
 - 5: Essencial.

Ajustes de Complexidade

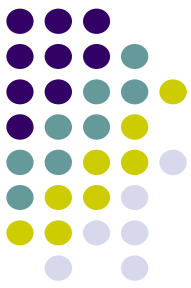


- F_i
 1. O sistema requer backup e recuperação confiável?
 2. São exigidas comunicações de dados?
 3. Há funções de processamento distribuídas?
 4. O desempenho é crítico?
 - ⋮
 11. O código foi projetado de forma a ser reusável?
 - ⋮
 14. A aplicação foi projetada de forma a facilitar mudanças e o uso pelo usuário?

Estimativa de Custo do Software

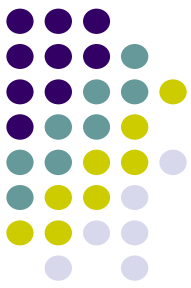


- Estimar o custo total do software é necessário para:
 - Estabelecer um orçamento para o projeto.
 - Informar o preço do software para o cliente.
- Para computar o custo total do software devemos considerar:
 - Custos de manutenção do hardware e software usados no desenvolvimento.
 - Custos com treinamento e viagens.
 - Custos com pessoal.



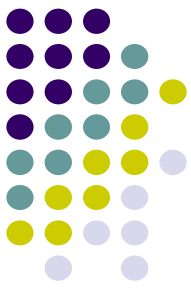
Estimativa de Produtividade

- Estimar produtividade é necessário para avaliar se o processo de desenvolvimento é efetivo.
- A estimativa de produtividade é baseada na divisão da medida de algum atributo do software pelo esforço total requerido para desenvolvimento.



Estimativa de Produtividade

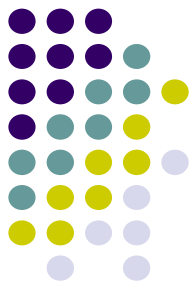
- A forma mais comum de medir produtividade é dividindo o número de linhas de código produzidas pelo número de programadores-mês requeridos para o projeto.



Técnicas de Estimativas

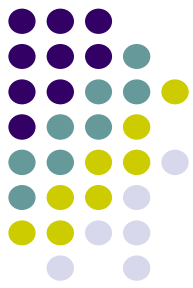
- Realizar estimativas precisas não é uma tarefa fácil.
- Existem várias técnicas que podem ajudar nesta tarefa.

Modelos Empíricos de Estimativa

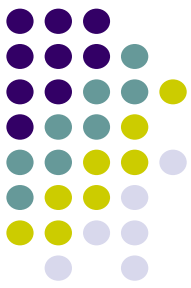


- Um modelo é desenvolvido usando informações históricas de projetos já concluídos.
- Uma fórmula matemática é usada para prever custos, baseando-se no tamanho do projeto, no número de programadores e outras características.
- A fórmula matemática normalmente apresenta um componente exponencial, indicando que o custo não aumenta de forma linear em relação ao tamanho do projeto.

Modelos Empíricos de Estimativa

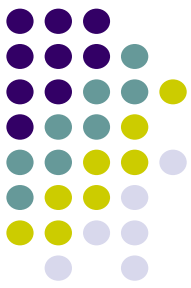


- A fórmula matemática normalmente combina atributos do produto (tamanho e complexidade, por exemplo) com atributos do processo de desenvolvimento.



O Modelo COCOMO

- Juntamente com o modelo de Putnam, é o modelo empírico mais utilizado para realizar estimativas.
- Existem três formas deste modelo:
 - COCOMO Básico.
 - COCOMO Intermediário.
 - COCOMO Avançado.

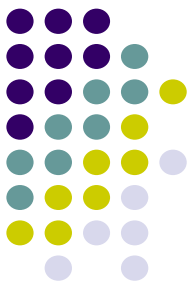


O Modelo COCOMO

- A fórmula matemática usada pelo modelo COCOMO é a seguinte:

$$\text{Esforço} = C \times PM^S \times M$$

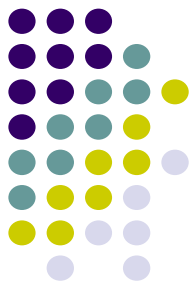
- C é um fator de complexidade.
- PM é uma métrica de produto.
- M é um multiplicador que combina diferentes atributos.
- S é um fator exponencial que indica a relação entre o custo e o tamanho do produto.



COCOMO Básico

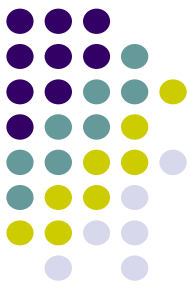
- Computa o esforço e custo do desenvolvimento considerando apenas o tamanho do programa expresso em linhas de código.
- Aplicando a fórmula:
 - PM = número de linhas do programa.
 - $M = 1$.
 - $C = 2,4$ e $S = 1,05$ em sistemas simples.
 - $C = 3,0$ e $S = 1,12$ em sistemas moderados.
 - $C = 3,6$ e $S = 1,20$ em sistemas complexos.

COCOMO Intermediário

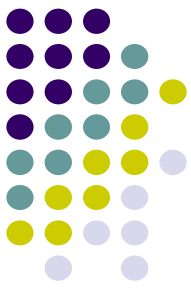


- Amplia o COCOMO básico, passando a considerar novos atributo direcionadores do custo.
- Atributos do Produto:
 - Confiabilidade, tamanho do banco de dados e complexidade.
- Atributos do Hardware:
 - Restrições de desempenho, restrições de memória, restrições de armazenamento.

COCOMO Intermediário

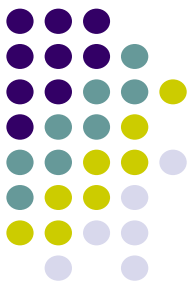


- Atributos de Pessoal:
 - Capacidade de análise, capacidade em engenharia de software, experiência em aplicações, experiência em linguagens de programação.
- Atributos do Projeto:
 - Uso de ferramentas de software, aplicação de métodos de engenharia de software, cronograma de atividades de desenvolvimento.



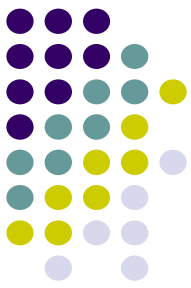
COCOMO Intermediário

- Os novos atributos adicionados são usados no cálculo do componente M da fórmula matemática.
- PM continua sendo o número de linhas do programa.
- C e S assumem os seguintes valores:
 - $C = 3,2$ e $S = 1,05$ em sistemas simples.
 - $C = 3,0$ e $S = 1,12$ em sistemas moderados.
 - $C = 2,8$ e $S = 1,20$ em sistemas complexos.



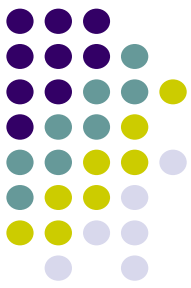
Estimativa de Prazo

- Além de estimar custo e esforço, é necessário também estimar quanto tempo levará para o software ser desenvolvido e quando a equipe será necessária para trabalhar no projeto.



Estimativa de Prazo

- O relacionamento entre equipe, esforço e tempo não é linear.
 - Dobrar a equipe não fará com que o projeto seja desenvolvido na metade do tempo.
 - Se a equipe aumenta, mais esforço pode ser necessário.



Estimativa de Prazo

- O modelo COCOMO inclui uma formula para estimativa de tempo requerido para completar o projeto (TDEV).

Projetos Simples: $TDEV = 2.5 \times PM^{0.38}$

Projetos Moderados: $TDEV = 2.5 \times PM^{0.35}$

Projetos Complexos: $TDEV = 2.5 \times PM^{0.32}$