Engenharia de Software II

Aula 20

http://www.ic.uff.br/~bianca/engsoft2/

Ementa

- Processos de desenvolvimento de software
- Estratégias e técnicas de teste de software
- Métricas para software
- **Gestão de projetos de software**
  - **Conceitos (Cap. 21)**
  - **Métricas (Cap. 22 – Seções 22.1 e 22.2)**
  - **Estimativas (Cap. 23)**
  - **Cronogramação**
  - **Gestão de risco**
  - **Gestão de qualidade**
  - **Gestão de modificações**
- Reengenharia e engenharia reversa

Estimativa de Projetos de Software

- Antes do projeto começar, o gerente e a equipe precisam estimar:
  - o trabalho a ser feito,
  - os recursos necessários,
  - o tempo que vai decorrer do início ao fim.
- A partir destas estimativas, é possível estabelecer um cronograma que defina:
  - as tarefas e marcos da engenharia de software,
  - quem é o responsável por conduzir cada tarefa,
  - as dependências intertarefas.
- Vale à pena planejar:
  - O tempo de planejar é menor que o tempo perdio em refazer.

Observações sobre estimativa

- Sempre que fazemos estimativas, devemos aceitar um grau de incerteza como inevitável.
  - Abordagens modernas assumem que as estimativas serão revistas à medida que o projeto avança.
- A disponibilidade de informação histórica tem forte influência no risco da estimativa.
  - Experiência anterior ajuda imensamente.
  - Métricas de software de projetos anteriores ajudam na geração de estimativas quantitavias.

O Processo de Planejamento de Projeto

- O objetivo do processo de planejamento é fornecer um arcabouço que permita ao gerente fazer estimativas razoáveis.
  - Recurso, custos e cronograma.
- As estimativas devem considerar tanto o melhor quanto o pior caso.

Conjunto de Tarefas para Planejamento de Projeto

1. Estabeleça o escopo do projeto.
2. Determine a viabilidade.
3. Analise riscos (Cap. 25)
4. Defina recursos necessários.
   a) Recursos humanos
   b) Recursos reúsaveis de software
   c) Recursos ambientais

Conjunto de Tarefas para Planejamento de Projeto (cont.)

5. Estime custo e esforço
   a) Decomponha o problema.
   b) Desenvolva duas ou mais estimativas usando tamanho, ponto por função, tarefas de processo ou casos de uso.
   c) Harmonize as estimativas.
6. Desenvolva um cronograma do projeto (Cap. 24)
   a) Estabeleça um conjunto significativo de tarefas.
   b) Defina uma rede de tarefas.
   c) Use ferramentas de cronogramação para desenvolver um diagrama de tempo.
   d) Defina mecanismos de rastreamento de cronograma.

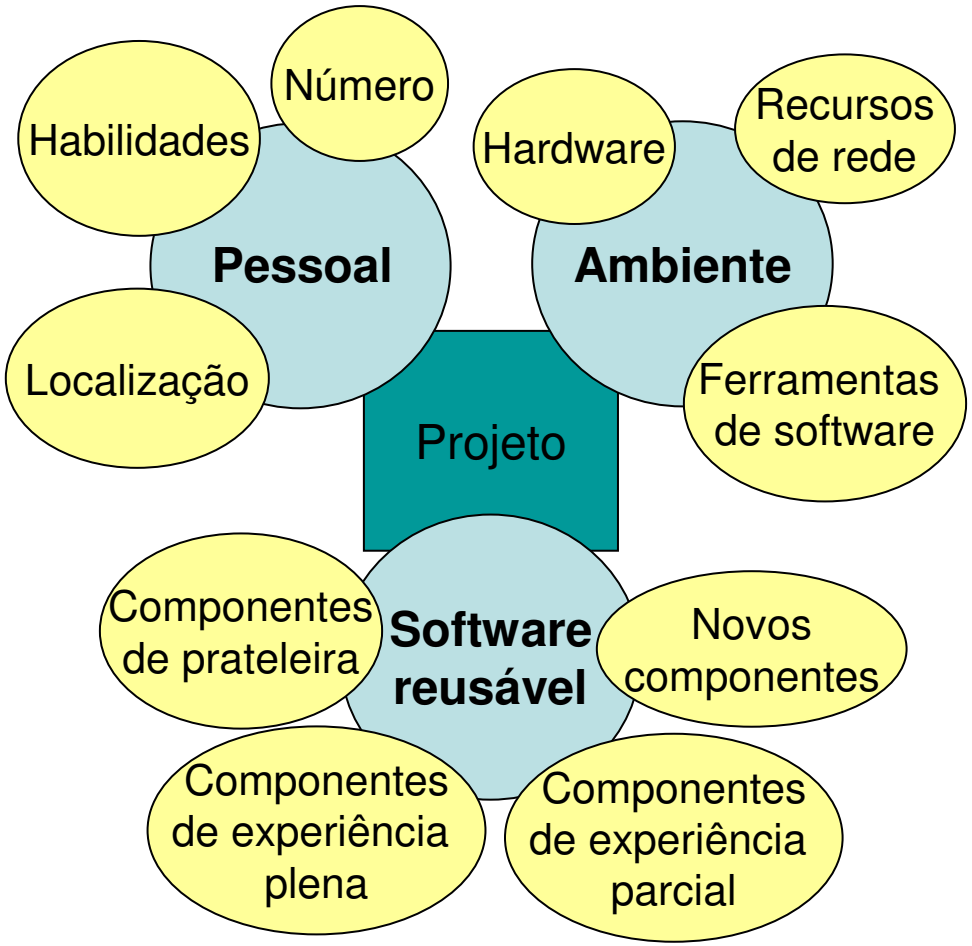
Escopo do Software

- Escopo
  - Funções e características a serem entregues aos usuários finais.
  - Dados que entram e saem.
  - O conteúdo que é apresentado aos usuários como conseqüência do uso do software.
  - Desempenho, restrições, interfaces e confiabilidade.
- O escopo pode ser definido usando uma de duas técnicas:
  1. Depois da comunicação com os interessados, uma descrição narrativa é desenvolvida.
  2. Os usuários finais desenvolvem um conjunto de casos de uso.

Viabilidade do Software

- Viabilidade: Podemos construir um software para satisfazer esse escopo?
  - Viabilidade tecnológica
    - O projeto é exeqüível tecnicamente? Os defeitos podem ser reduzidos ao ponto de satisfazer as necessidades da aplicação?
  - Viabilidade financeira
    - O desenvolvimento pode ser completado a um custo que a organização de software, seu cliente ou o mercado possam pagar?
  - Viabilidade de tempo
    - O prazo para o projeto chegar ao mercado vai vencer a concorrência?
  - Viabilidade de recursos
    - A organização tem os recursos necessários para obter sucesso?

Recursos



- Cada recurso é especificado por quatro características:
  - Descrição
  - Declaração de disponibilidade
  - Época em que o recurso será necessário
  - Prazo no qual o recurso será aplicado

Janela de tempo

Recursos Humanos

- O planejador começa pela avaliação do escopo e seleção das aptidões necessárias.
  - Posição na organização
    - Gerente, desenvolvedor experiente
  - Especialidade
    - Telecomunicações, base de dados, cliente/servidor
- Projetos pequenos
  - Um único indivíduo pode realizar todas as tarefas de desenvolvimento, consultando especialistas.
- Projetos grandes
  - A equipe pode estar geograficamente distribuída.
- A quantidade de pessoas necessária só pode ser determinada quando é feita uma estimativa do esforço (ex: pessoa-mês).

Recursos de Software Reusáveis

- A engenharia de software baseada em componentes enfatiza a reusabilidade.
  - Os componentes devem ser catalogados, padronizados e validados.
- Existem quatro categorias de componentes:
  - Componentes de prateleira
    - Foram adquiridos de terceiros, estão prontos e validados.
  - Componentes de experiência plena
    - Desenvolvidos para projetos anteriores que são similares ao projeto atual, podem ser usados sem muitas modificações.
  - Componentes de experiência parcial
    - Desenvolvidos para projetos anteriores, mas vão exigir substancial modificação.
  - Componentes novos
    - Precisam ser construídos especificamente para o projeto atual.

Recursos de Ambiente

- O ambiente que apóia o projeto de desenvolvimento de software incorpora hardware e ferramentas de software.
- O planejador de um projeto deve verificar se os recursos de hardware e software estão disponíveis para o projeto.

Estimativa do Projeto de Software

- Opções para conseguir estimativas de custo e esforço:
  - Adiar a estimativa até que o projeto esteja mais adiantado.
    - Não é uma opção prática.
  - Basear as estimativas em projetos anteriores.
    - Só é possível se o projeto for bastante semelhante a esforços anteriores.
  - Usar técnicas de decomposição.
    - Usa a abordagem “dividir e conquistar”.
  - Usar um ou mais modelos empíricos.
    - Serve como complemento às técnicas de decomposição.
- É comum usar mais de um método para estimativa a fim de comparar os resultados.

Técnicas de Decomposição

- Na decomposição, o problema de estimar o projeto de software é sub-dividido em problemas menores.
- Antes que uma estimativa de custos e esforço possa ser feita, o planejador precisa entender o escopo do software e fazer uma estimativa do seu “tamanho”.
  - Dimensionamento do software

Técnicas de Decomposição (cont.)

- O planejador do projeta começa com uma declaração do escopo e decompõe o software em funções que possam ser estimadas:
  - LOC, FP ou outra variável de dimensão é estimada para cada função.
  - Métricas referenciais de produtividade são então aplicadas à variável de estimativa (LOC/pm ou FP/pm).
    - Referências por domínio são mais adequadas.

Estimativa de tamanho

- O planejador começa estimando um intervalo de valores para cada função.
- Usando dados históricos ou intuição, são estimados:
  - Um valor de tamanho otimista (S_{ot})
  - Um valor mais provável (S_m)
  - Um valor de tamanho pessimista (S_{pess})
- O valor esperado é calculado como sendo:
  - $S = (S_{ot} + 4S_m + S_{pess})/6$

Exemplo de Estimativa Baseada em LOC

- Sistema CAD

Função	Otimista	Médio	Pessimista	Estimativa
Recursos de controle e interface				2300
Análise geométrica bidimensional				5300
Análise geométrica tridimensional	4600	6900	8600	6800
Gestão de base de dados				3350
Recursos de computação gráfica				4950
Função de controle de periféricos				2100
Módulos de análise de projeto				8400
Estimativa das linhas de código				33200

Considerando 620 LOC/pm, a estimativa de esforço é 54 pessoas-mês.
Considerando 13 $/LOC, a estimativa de custo é $431000.

Estimativa Baseada em Processo

- Além de decompor o projeto em funções, decompõe-se também nas atividades do processo.
- O planejador estima o esforço que será necessário para realizar cada atividade do processo pra cada função.
- É muito provável que o custo de trabalho varie para cada tarefa.
  - As primeiras tarefas exigem pessoal mais bem remunerado que o pessoal que faz geração de código e testes.

Estimativas com Casos de Uso

- Dificuldades de se utilizar casos de uso para estimativas:
  - Não tem um formato padrão.
  - Representam uma visão externa e são escritos em diferentes níveis de abstração.
  - Não indicam a complexidade e as características das funções que são descritas.
  - Não descrevem comportamento complexo que envolve muitas funções.