Engenharia de Software Aula 19

Medições e Qualidade de Software

Eduardo Figueiredo
http://www.dcc.ufmg.br/~figueiredo
dcc603@gmail.com

14 Maio 2012

Disciplina 2012-2

- Medição e Qualidade de Software
- Terças e quintas: 9:25 as 11:05
- Ofertada para
  - Alunos de Ciência da Computação
  - Alunos de Sistemas de Informação
  - Alunos de Pós-Graduação

Tópicos da Disciplina

- Medição e Qualidade de Software
  - Qualidade do produto e do processo
  - Refatoração e *bad smells*
  - Métricas de software
  - Métodos quantitativos para detecção de *bad smells*
  - Avaliação da qualidade de software
  - Planejamento de experimentos e análise dos resultados

Método de Avaliação

- Medição e Qualidade de Software
  - Prova 1 (P1): 30 pts
  - Prova 2 (P2): 30 pts
  - Seminário: 20 pts
  - Exercícios e participação: 20 pts.
  - Prova substitutiva: 30 pts.
    - Substitui a nota de P1 ou P2

Agenda da Aula

- Qualidade de Software
- Medição de Qualidade
- Métricas de Produto
  - Métricas tradicionais
  - Métricas de programas orientados a objetos

Qualidade de Software

- A qualidade de software tem se aprimorado nos últimos 15 anos
  - Empresas têm adotado novas técnicas
  - Orientação a objetos tem se difundido
  - Ferramentas CASE têm sido usadas
- Na manufatura, qualidade significa atender às especificações
  - Em software, a definição não é tão simples

Adequado à Especificação

- Não é fácil definir qualidade de software como adequado à especificação
- A especificação pode estar ambígua, incompleta ou inconsistente
  - Alguns requisitos podem não aparecer na especificação
  - É difícil especificar todos os requisitos

Atributos Implícitos de Qualidade

- Alguns atributos são difíceis de serem especificados
  - Mas tem grande efeito na qualidade do sistema
- Exemplos
  - Facilidade de Manutenção
  - Proteção / Segurança
  - Eficiência, etc.

Alguns Atributos de Qualidade

Segurança	Facilidade de compreensão	Portabilidade
Proteção	Facilidade de testes	Facilidade de uso
Confiabilidade	Adaptabilidade	Facilidade de reuso
Facilidade de recuperação	Modularidade	Eficiência
Robustez	Complexidade	Facilidade de aprendizado

Equipe de Qualidade

- Idealmente, a equipe de garantia da qualidade deve ser diferente da equipe de desenvolvimento
- O processo de qualidade envolve
  - Definir padrões de processo
  - Monitorar o processo para verificar o adequado uso dos padrões
  - Emitir relatórios para a gerência de projeto e da organização

O tamanho do projeto

- Mesmo em sistemas pequenos, o gerenciamento da qualidade é importante
  - Entretanto, ele pode seguir um abordagem mais informal
- Em sistemas grandes, o gerenciamento de qualidade requer três atividades
  - Garantia de Qualidade
  - Planejamento de Qualidade
  - Controle de Qualidade

Atividades de Gerenciamento

- Garantia de Qualidade
  - Estabelece um framework com os procedimentos e padrões
- Planejamento de Qualidade
  - Seleção dos procedimentos e padrões apropriados ao projeto
- Controle de Qualidade
  - Verifica se os procedimentos e padrões estão sendo seguidos

Garantia da Qualidade (QA)

- A garantia da qualidade de software define busca saber
  - Como a qualidade pode ser atingida
  - Se a qualidade foi atingida

Planejamento da Qualidade

- É o processo de desenvolvimento de um plano de qualidade para um projeto
- Estabelece os padrões apropriados para um produto e processo

Estrutura de Planejamento

1. Descrição do produto, mercado e das expectativas de qualidade
2. Plano com as datas críticas e responsabilidades
3. Descrição dos métodos e serviços usados no desenvolvimento e gerenciamento do produto
4. Definição das metas de qualidade e respectivas justificativas
5. Descrição dos riscos e ações para minimizá-los

Controle de Qualidade

- Envolve a monitoração do processo de desenvolvimento
- Busca assegurar que os procedimentos e padrões são de fato aplicados

Abordagens de Controle

- Revisões de qualidade
  - A equipe de qualidade verifica a documentação e o processo de desenvolvimento
- Avaliação automatizada
  - O produto (ou documentação) é processado automaticamente
  - Métricas são usadas para verificar a qualidade

Qualidade de Processo x Produto

- Acredita-se que a qualidade do processo afeta diretamente a qualidade do produto
- Esta crença veio da indústria de manufatura
  - Em software, a relação entre qualidade de processo e qualidade de produto é complexa
  - Estudos mostram que a relação existe

Qualidade Baseada no Processo

Padrões de Software

Padrões de Software

- Padrões de produto
  - Aplicam-se ao produto de software que está sendo desenvolvido
  - Padrões de documentação, padrões de codificação, etc.
- Padrões de processo
  - Definem as atividades do processo e os seus resultados
  - Processos de validação, ferramentas, etc.

Exemplos: Padrões de Produto

- Formulário de revisão do projeto
- Estrutura do documento de requisitos
- Formato da assinatura de métodos
- Estilo de programação Java
- Formato do plano de projeto
- Formato do formulário de solicitação de mudanças

Exemplos: Padrões de Processo

- Conduta de revisão de projeto
- Envio de documentos para gerência
- Processo de liberação de versões
- Processo de aprovação do plano de projeto
- Processo de controle de mudanças
- Processo de registro de testes

Importância de Padrões

- Documentam o conhecimento das melhores práticas
- Indicam o caminho para se obter qualidade (aos menos experientes)
- Facilitam a comunicação entre os membros da equipe

O uso de fato de padrões

- Alguns cuidados para que os padrões sejam de fato implementados
  - Envolver a equipe de desenvolvimento na escolha dos padrões
  - Revisar os padrões regularmente para refletir mudanças de tecnologia
  - Não incluir apenas o que seguir, mas também o porque de seguir um padrão
  - Prover ferramentas para apoiar a adoção dos padrões

Medições de Software

Métrica de software

- Medições se dedicam a obter um ou mais valores numéricos para um atributo de qualidade
  - Comparando os números, é possível tirar conclusões sobre a qualidade do produto
- Uma métrica de software é qualquer medição que se refere ao sistema
  - Medições de tamanho (exemplo, LOC)
  - Número de defeitos relatados, etc.

Por que medir?

- Revisão para avaliação da qualidade é uma atividade demorada
  - Geralmente causa atraso na conclusão do projeto
- Para acelerar a revisão, ferramentas devem ser empregadas para permitir avaliações automatizadas

Uso de Medições

- Medições de software podem ser usadas de duas maneiras
  - Avaliar a qualidade do sistema e fazer previsões gerais sobre ele (exemplo, número de defeitos)
  - Para identificar partes (ou módulos) problemáticos

Adoção pela Indústria

- Muitas empresas ainda não usam medições sistemáticas para avaliar a qualidade
- Algumas razões
  - Os processos das empresas não são maduros o suficiente
  - A ausência de métricas padronizadas
  - Limitado apoio de ferramentas de medição

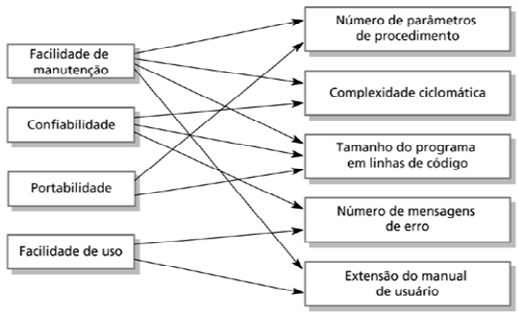
Problemas com Medições

- Geralmente é impossível medir um atributo de qualidade diretamente
  - Atributos de qualidade são fatores externos ao software
  - Métricas medem fatores internos
- Exemplos de atributos de qualidade
  - Facilidade de manutenção
  - Facilidade de uso
  - Confiabilidade

Solução: Modelos de Qualidade

- A solução é medir atributos que podem estar relacionados aos atributos de qualidade
- Idealmente, deve haver um relacionamento claro e válido entre atributos de qualidade e atributos internos

Um Modelo de Qualidade



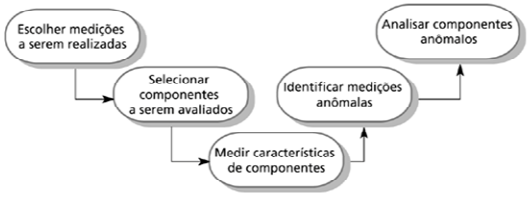
Validade dos Modelos

- Três condições devem ser verificadas em modelos de qualidade
  - O atributo interno deve ser precisamente medido
  - Deve haver relacionamentos entre o que podemos medir e o que queremos saber
  - Os relacionamentos são compreendidos e válidos

O Processo de Medição

- O processo de medição deve fazer parte do processo de controle da qualidade
  - Utilizam dados históricos de projetos anteriores
- As atividades do processo
  - Escolher medições a serem realizadas
  - Selecionar componentes a serem avaliados
  - Medir características dos componentes
  - Identificar medições anômalas
  - Analisar componentes anômalos

Modelo do Processo



Escolher Medições

- Uma abordagem para escolher as medições é o GQM
  - *Goal-Question-Metric*
- As questões são formuladas para atender um objetivo
- As métricas são escolhidas para responderem as questões

O Modelo de Medição GQM

- Objetivos
  - Definem o que a organização quer melhorar (exemplo: produtividade)
- Questões
  - Refinamento dos objetivos em áreas de incertezas (exemplo: linhas de código produzidas podem ser aumentadas?)
- Métricas
  - Medições necessárias para responder as questões (exemplo: LOC por desenvolvedor)

Selecionar Componentes

- Pode não ser necessário (ou desejável) medir todo o sistema
- Estratégias de escolha
  - Escolher um subconjunto representativo de todos os componentes
  - Escolher os componentes particularmente críticos na aplicação

Medir os Atributos de Qualidade

- Os componentes selecionados são medidos
- As medidas são associadas aos atributos de qualidade
  - Geralmente envolve uma representação dos componentes
- Ferramentas de medição podem estar incorporadas a outras ferramentas (ou ambientes) de desenvolvimento

Analisar Medições

- Uma vez feita as medições, é preciso compará-las a medições anteriores
  - Dados históricos são utilizados
- A análise deve procurar valores incomuns
  - Ou seja, valores muito altos ou muito baixos para cada métrica

Analisar Componentes

- Se um componente tem valores anômalos, este deve ser examinado
  - A inspeção é responsável por decidir se existe (ou não) problema no componente
- Um valor incomum para um componente não necessariamente significa que o componente tem baixa qualidade

Métricas de Produto

Métricas de Produto

- Quantificam atributos internos do software
- Exemplos de atributos
  - Tamanho
  - Acoplamento dentre componentes
  - Coesão de um componente, etc.

Tipos de Métricas

- Métricas Dinâmicas
  - São coletadas por medições realizadas durante a execução do programa
- Métricas Estáticas
  - São coletadas por medições realizadas na documentação de projeto ou código fonte do programa

Dinâmicas x Estática

- Métricas dinâmicas ajudam a avaliar atributos de qualidade como eficiência e confiabilidade
  - São medidas após o sistema ter sido implementado
- Métricas estáticas ajudam a avaliar atributos como complexidade e facilidade de manutenção
  - Podem ser medidas na fase de projeto

Algumas Métricas Estáticas

- Fan-in / Fan-out
- Tamanho do código
- Complexidade Ciclomática
- Tamanho do Vocabulário
- Profundidade de Aninhamento

Fan-in e Fan-out

- Fan-in
  - Conta o número de funções que chama uma determinada função
  - Valor alto significa grande impacto em mudanças (propagação)
- Fan-out
  - Conta o número de funções chamadas pela função
  - Valor algo significa grande complexidade da função

Tamanho e Complexidade

- Tamanho
  - Tamanho tem se mostrado como métricas mais confiáveis e úteis
  - Em geral, quanto maior, mais complexo e propenso a erros será o componente
- Complexidade Ciclomática
  - Mede a complexidade de controle do programa (*if*, *while*, *for*, etc.)
  - Está relacionada a facilidade de compreensão

Vocabulário e Aninhamento

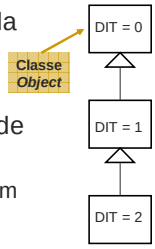
- Tamanho do Vocabulário
  - Conta a quantidade de identificadores (exemplo, nome de classes) do programa
  - Mais identificadores podem significar que eles são mais significativos
- Profundidade de Aninhamento
  - Conta estruturas internas como *if* e *while* aninhados
  - Estruturas aninhadas são mais difíceis de se compreender

Métricas de Programas OO

- Métricas de Chidamber-Kemerer (CK)
  - Métodos Ponderados por Classes (WMC)
  - Profundidade da Herança (DIT)
  - Número de Filhos (NOC)
  - Acoplamento entre Objetos (CBO)
  - Falta de Coesão em Métodos (LCOM)
- Número de Operações Sobreescritas

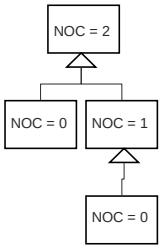
Profundidade de Herança (DIT)

- Representam o número de níveis que uma classe herda métodos e atributos
- Quanto maior a profundidade
  - Mais complexo o projeto
  - Mais difícil de se entender um módulo



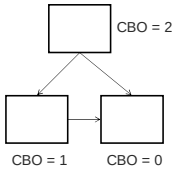
Número de Filhos (NOC)

- Conta o número de subclasses diretas
  - Mede a largura da hierarquia de uma classe
- Valor alto, pode indicar maior reuso



Acoplamento entre Objetos (CBO)

- Semelhante a Fan-out
  - Conta classes chamadas por uma classe
- Quanto mais acoplado uma classe
  - Mais difícil de entender e de manter



Métricas para Métodos

- Métodos Ponderados por Classes (WMC)
  - Atribui pesos aos métodos de uma classe
  - Uma forma é “pesar” por linhas de código
  - Valores altos indicam complexidade
- Número de Operações Sobrescritas
  - Conta as operações de uma classe que são sobrescritas por subclasses
  - Valores altos indicam problema na hierarquia de herança

Análise de Medições

- Nem sempre é óbvio o que os dados significam
  - Entender uma grande quantidade de números é muito difícil
- Estatísticos devem ser consultados, se estiverem disponíveis
- A análise de dados deve levar em conta as circunstâncias locais

Exemplo (Arquivo XLS)

- Medição de várias versões
  - Métricas estáticas coletadas em várias versões do sistema Health Watcher
- Métricas
  - CBO: Coupling Between Objects
  - LCOM: Lack of Cohesion over Methods
  - DIT: Depth of Inheritance Tree
  - NOC: Number of Children
  - LOC: Lines of Code, etc.

Resumo

- Gerenciamento de qualidade de software assegura que o software atende aos padrões adequados
- Padrões encapsulam conhecimento de boas práticas
- Revisões são amplamente usadas para avaliação de qualidade
  - Revisões são demoradas e podem atrasar o projeto

Resumo

- Medição podem ser aplicadas tanto ao processo quanto ao produto de software
- Métricas de produto podem ser usadas para identificar componentes potencialmente problemáticos
- Métricas são difíceis de serem usadas e compreendidas
  - Nem sempre são padronizadas

Bibliografia da Aula

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 2011.
  - Cap. 24 Gerenciamento de Qualidade