PUCRS
Pontifícia Universidade Católica
do Rio Grande do Sul

ESCOLA
POLITÉCNICA

Construção de Software
Complexidade

46504-04 - Construção de Software

Prof. Msc. Eduardo Arruda
eduardo.arruda@pucrs.br

Material original elaborado pelo Prof. Dr.
Marcelo Yamaguti

Fundamentos de Construção de Software

- Minimizar a complexidade
  - Limitação de lidar com muita informação ao mesmo tempo
  - Minimizar a complexidade auxilia o teste
  - Buscar código simples e legível em vez de código astuto
  - Como obter:
    - Projeto modular
    - Técnicas de codificação
    - Técnicas de qualidade de construção

Métricas de complexidade

Métrica	Descrição
Fan-in / Fan-out	Fan-in é uma medida do número de funções ou métodos que chamam outra função ou método. Fan-out é o número de funções que são chamadas por uma função. Um valor alto para fan-in significa que a função está fortemente acoplada ao resto do código e as alterações nela terão efeitos abrangentes. Um valor alto para fan-out sugere que a complexidade geral da função pode ser alta por causa da complexidade da lógica de controle necessária para coordenar os componentes chamados.
Comprimento do código	Esta é uma medida do tamanho de um programa. Geralmente, quanto maior o tamanho do código de um componente, mais complexo e sujeito a erros esse componente provavelmente será. O comprimento do código tem se mostrado uma das métricas mais confiáveis para prever a tendência a erros em componentes.
Complexidade ciclomática	Essa métrica mede a quantidade de caminhos de execução independentes a partir de um código fonte.
Comprimento dos identificadores	Esta é uma medida do comprimento médio dos identificadores (nomes para variáveis, classes, métodos, etc.) em um programa. Quanto mais longos forem os identificadores, maior será a probabilidade de eles serem significativos e, portanto, mais compreensível será o programa.
Profundidade do aninhamento condicional	Esta é uma medida da profundidade do aninhamento de instruções if em um programa. As instruções if profundamente aninhadas são difíceis de entender e potencialmente sujeitas a erros.
Índice de nevoeiro	é uma medida do comprimento médio de palavras e frases em documentos. Quanto mais alto for o valor do índice Fog de um documento, mais difícil será de entender o documento.

Métricas de complexidade de requisitos

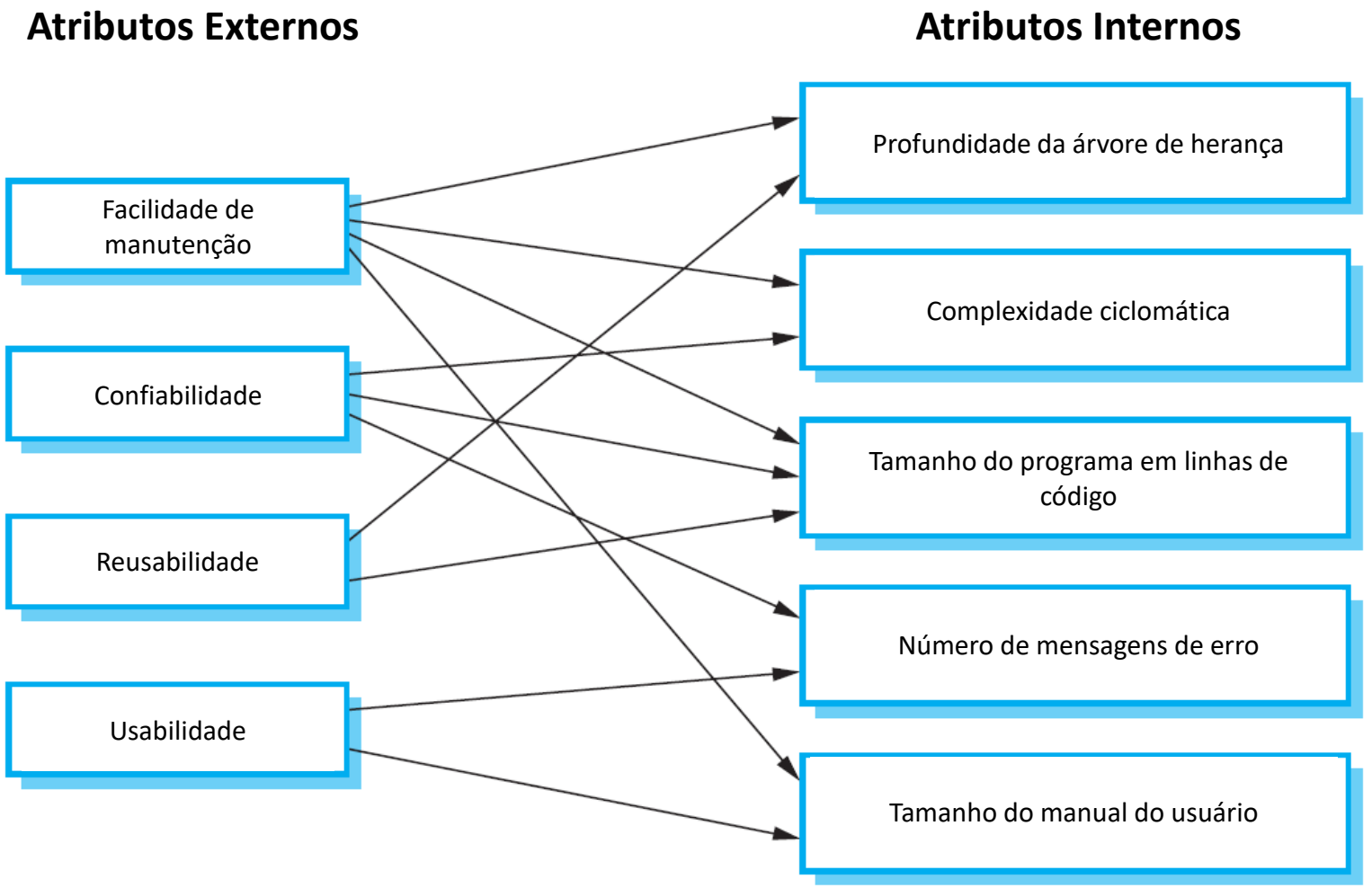
Pontos de Função

Métrica	Descrição
Número de entradas externas	Cada entrada externa se origina de um usuário ou é transmitida de outro aplicativo e fornece diferentes dados orientados a aplicativos ou informações de controle. As entradas são frequentemente usadas para atualizar arquivos lógicos internos. As entradas devem ser diferenciadas das consultas, que são contadas separadamente.
Número de saídas externas	Cada saída externa é derivada de dados dentro do aplicativo que fornece informações ao usuário. Neste contexto, a saída externa refere-se a relatórios, telas, mensagens de erro, etc. Os itens de dados individuais em um relatório não são contados separadamente.
Número de consultas externas	Uma consulta externa é definida como uma entrada online que resulta na geração de alguma resposta imediata do software na forma de uma saída online (geralmente recuperada de um arquivo lógico interno).
Número de arquivos lógicos internos	Cada arquivo lógico interno é um agrupamento lógico de dados que reside dentro dos limites do aplicativo e é mantido por meio de entradas externas.
Número de arquivos de interface externa	Cada arquivo de interface externa é um agrupamento lógico de dados que reside fora do aplicativo, mas fornece informações que podem ser úteis para o aplicativo.

Métricas de complexidade

Métrica	Descrição
Métodos ponderados por classe	Este é o número de métodos em cada classe, ponderado pela complexidade de cada método. Portanto, um método simples pode ter uma complexidade de 1 e um método grande e complexo pode ter um valor muito mais alto. Quanto maior o valor dessa métrica, mais complexa é a classe. Objetos complexos são mais propensos a serem difíceis de entender. Eles podem não ser logicamente coesos, portanto, não podem ser reutilizados efetivamente como superclasses em uma árvore de herança.
Profundidade da árvore de herança	Representa o número de níveis discretos na árvore de herança onde as subclasses herdam atributos e operações (métodos) das superclasses. Quanto mais profunda a árvore de herança, mais complexo é o projeto. Muitas classes de objetos podem ter que ser compreendidas para compreender as classes de objetos nas folhas da árvore.
Número de filhos	Esta é uma medida do número de subclasses imediatas em uma classe. Ele mede a amplitude de uma hierarquia de classes. Um valor alto para esta métrica pode indicar maior reutilização. Isso pode significar que mais esforço deve ser feito na validação de classes base devido ao número de subclasses que dependem delas.
Acoplamento entre classes de objetos	As classes são acopladas quando os métodos em uma classe usam métodos ou variáveis de instância definidas em uma classe diferente. Esta métrica mede quanto existe de acoplamento. Um valor alto para ela significa que as classes são altamente dependentes e, portanto, é mais provável que a mudança de uma classe afete outras classes no programa.
Resposta para uma classe	Esta métrica mede o número de métodos que poderiam ser executados em resposta a uma mensagem recebida por um objeto dessa classe. Quanto mais alto for o valor dessa métrica, mais complexa será a classe e, portanto, mais provável será que ela inclua erros.
Falta de coesão nos métodos	Essa métrica é calculada considerando pares de métodos em uma classe. Ela é a diferença entre o número de pares de métodos sem atributos compartilhados e o número de pares de métodos com atributos compartilhados.

Relação entre os atributos externos e internos



SOMMERVILLE, I. Engenharia de software. 9ª ed. São Paulo: Pearson Brasil, 2011. Página 670.

Referências

- Estude para aprofundamento no conteúdo:
  - PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software-8ª Edição. McGraw Hill Brasil, 2016.
  - SOMMERVILLE, I. Engenharia de software. 9ª ed. São Paulo: Pearson Brasil, 2011. – Capítulos 2 e 3.