

Ferramentas e Técnicas de Fan-Out

Pedro Gonçalves, A82313

Roberto Cachada, A81012

Introdução

- Qualidade de Software.
 - ISO 25010.
 - Standard onde foram definidos quais os fatores pelos quais um software deve ser avaliado.
 - Formas muito teóricas e gerais para medir a qualidade de um software.
- Métricas de Software.
 - Servem para medir a qualidade de um software.
 - Criadas para medir de modo mais prático a qualidade de um software.
 - TIOBE Quality Indicator.

Fan-Out

- Consiste no grau de interdependência entre os diferentes componentes de um sistema, ou seja, quantos módulos são necessários para que um dado componente funcione corretamente.
- Métrica pode ser aplicada tanto ao nível da função como ao nível do módulo.
- Quanto mais elevado o fator de Fan-Out de um componente, maior é a dificuldade na modificação do mesmo.
- Esta métrica serve para avaliar o fator de manutenção de um software.

Tipos de Fan-Out

- Fan-Out Externo:
 - Consiste na chamada de módulos não pertencentes ao sistema, isto é, consiste na reutilização de software já existente.
- Fan-Out Interno:
 - Consiste na chamada de módulos pertencentes ao sistema.
- Para efeitos de cálculo do TQI (TIOBE Quality Indicator), o Fan-Out interno possui um efeito 4 vezes mais negativo num sistema comparativamente ao Fan-Out externo.

Cálculo do Fan-Out

- Fan-Out Médio:
 - Consiste na soma do Fan-Out dos diferentes ficheiros de um sistema e dividir esse resultado pelo número de ficheiros.
- TQI Score:
 - Método de cálculo recomendado pelo TIOBE.
 - Segue a seguinte fórmula:

$$score = \min(\max(120 - (8 * internalFanOut + 2 * externalFanOut), 0), 100)$$

Cálculo do Fan-Out

- Consoante o resultado do cálculo do TQI Score, é atribuída uma classificação ao sistema, numa escala de A a F, tendo em conta a tabela apresentada em baixo.

Fan Out	TQI Score	Category
<= 6	>= 90%	 A
<= 8	>= 80%	 B
<= 10	>= 70%	 C
<= 14	>= 50%	 D
<= 16	>= 40%	 E
> 16	< 40%	 F

Cálculo para diferentes linguagens

- C/C++ → Número de *includes*;
- JavaScript → Número de *imports*;
- C# → Número de dependências únicas para cada ficheiro;
- Java → Número de declarações *import*. *Wildcard imports* devem contar como 5 declarações *import* normais;
- Python → Número de *imports/import-from*;
- Swift → Número de declarações *import*.

Ferramentas

- TICS Framework (C, C++, Dart, Go, Lua, Java, JavaScript, JSP, Kotlin, Objective-C, PL, Python, Scala, Swift, TypeScript, VB.NET)
- TICSCil (C#)
- mlint (Matlab)
- JStyle (Java)
- JHawk (Java)

JHawk

- Apenas tínhamos disponíveis a versão demo
- Não foi possível testar todas as funcionalidades
- É possível ver o número de pacotes importados por cada classe

Classes

Name	No. Methods	LCOM	AVCC	NOS	HBUG	HEFF	UWCS	INST	PACK	RFC
Chronometer	8	0,29	1,00	18	0,11	1062,88	10	2	0	8
cliente	12	0,08	1,75	149	2,11	140715,34	25	13	6	14
servidor	4	0,08	2,50	83	1,23	55723,58	12	8	2	5

No. of packages imported

Conclusão

- Percebemos melhor a importância do Fan Out no desenvolvimento de software
- É uma métrica que dá uma ideia do nível de complexidade do sistema que está a ser desenvolvido
- Isso permite perceber o quão difícil será fazer manutenção do código

Ferramentas e Técnicas de Fan-Out

Pedro Gonçalves, A82313

Roberto Cachada, A81012