

SOFT
DISCIPLINA: Engenharia de Software
AULA NÚMERO: 17

DATA: ____/____/____
PROFESSOR: Andrey

APRESENTAÇÃO

O objetivo desta aula é apresentar, discutir o conceito de métricas de software orientadas a função.

DESENVOLVIMENTO

Métricas baseadas em Função

Estimativa usando Pontos por Caso do Uso

Foi proposto em 1993 por Gustav Karner; Baseou-se na Análise por Pontos de Função; Trata de estimar o tamanho de um sistema de acordo com:

- o modo como os usuários o utilizarão;
- a complexidade de ações requerida por cada tipo de usuário;
- uma análise em alto nível dos passos necessários para a realização de cada tarefa;

O Método de Use Case Points foi criado para que seja possível estimar o tamanho de um sistema já na fase de levantamento de Casos de Uso; Ele utiliza-se dos próprios documentos gerados nesta fase de análise como subsídio para o cálculo dimensional;

Sistema que será usado como exemplo:

- Site de suporte de produtos para uma grande companhia de software;
- A estimativa foi feita a partir dos casos de uso de nível muito alto (*business modelling*), que foram criados em tempo de levantamento de requisitos;
- Os atores, nessa vez, foram os diferentes tipos de usuários identificados nesses casos de uso;

Passo 1: Cálculo do UAW (*Unadjusted Actor Weight*)

Tipo de Ator	Peso	Descrição
Ator Simples	1	Outro sistema acessado através de uma API de programação
Ator Médio	2	Outro sistema acessado interagindo através da rede
Ator Complexo	3	Um usuário interagindo através de uma interface gráfica

No exemplo:

Tipo de Ator	Peso	Nº de atores	Resultado
Ator Simples	1	0	0
Ator Médio	2	0	0
Ator Complexo	3	4	12
Total UAW			12

Passo 2: Cálculo do UUCW (*Unadjusted Use Case Weight*)

Para fins de cálculo, dividimos os casos de uso em três níveis de complexidade:

- **Simple (peso 5):** Tem até 3 transações, incluindo os passos alternativos, e envolve menos de 5 entidades;
- **Médio (peso 10):** Tem de 4 a 7 transações, incluindo os passos alternativos, e envolve de 5 a 10 entidades;
- **Complexo (peso 15):** Tem acima de 7 transações, incluindo os passos alternativos, e envolve pelo menos de 10 entidades;

No exemplo:

Tipo	Peso	Nº de Casos de Uso	Resultado
Simple	5	7	35
Médio	10	13	130
Complexo	15	3	45
Total UUCW			210

Passo 3: Cálculo do UUCP (*Unadjusted Use Case Points*)

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

No caso do exemplo:

$$\text{UUCP} = 12 + 210 = 222$$

Calculando fatores de ajuste:

O método de ajuste é bastante similar ao adotado pela Análise por Pontos de Função e é constituído de duas partes:

- **Cálculo de fatores técnicos:** cobrindo uma série de requisitos funcionais do sistema;
- **Cálculo de fatores de ambiente:** requisitos não-funcionais associados ao processo de desenvolvimento;

Passo 4: Cálculo do Tfactor

Para cada requisito listado na tabela, deve ser atribuído um valor que determina a influência do requisito no sistema, variando entre 0 e 5;

Fator	Requisito	Peso	Influência	Resultado
T1	Sistema distribuído	2	1	2
T2	Tempo de resposta	2	3	6
T3	Eficiência	1	3	3
T4	Processamento complexo	1	3	3
T5	Código reusável	1	0	0
T6	Facilidade de instalação	0,5	0	0
T7	Facilidade de uso	0,5	5	2,5
T8	Portabilidade	2	0	0
T9	Facilidade de mudança	1	3	3
T10	Concorrência	1	0	0
T11	Recursos de segurança	1	0	0
T12	Acessível por terceiros	1	0	0
T13	Requer treinamento especial	1	0	0
Tfactor				19,5

Passo 5: Cálculo do TCF (Technical Complexity Factor)

$$\text{TCF} = 0.6 + (0.01 \times \text{Tfactor})$$

No caso do exemplo:

$$\text{TCF} = 0.6 + (0.01 \times 19.5) = 0.795$$

Fator	Descrição	Peso	Influência	Resultado
E1	Familiaridade com RUP ou outro processo formal	1.5	5	7.5
E2	Experiência com a aplicação em desenvolvimento	0.5	0	0
E3	Experiência em Orientação a Objetos	1	5	5
E4	Presença de analista experiente	0.5	5	2.5
E5	Motivação	1	5	5
E6	Requisitos estáveis	2	3	6
E7	Desenvolvedores em meio-expediente	-1	0	0
E8	Linguagem de programação difícil	2	0	0
			Efactor	26

Passo 7: Cálculo do ECF (Environmental Complexity Factor)

$$\text{ECF} = 1.4 + (-0.03 \times \text{Efactor})$$

No caso do exemplo:

$$\text{ECF} = 1.4 + (-0.03 \times 26) = 0.62$$

Passo 8: Cálculo dos UCP (Use Case Points)

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{ECF}$$

No caso do exemplo:

$$\text{ECF} = 222 \times 0.795 \times 0.62 = 109.42 \text{ ou } 109 \text{ Use Case Points}$$

Passo 9: Cálculo do tempo de trabalho estimado

Para simplificar, utilizaremos a média de 20 horas por Ponto de Casos de Uso . No caso do exemplo:

$$\text{Tempo estimado} = 109 * 20 = \mathbf{2180 \text{ horas de trabalho}}$$

Analisando as duas técnicas:

APF	PCU
Mais antiga e mais utilizada no mundo	Relativamente nova e pouco utilizada
Padrão internacional desde 2002	Ainda não alcançou o nível de padronização e nem foi incorporada em ferramentas populares
Não requer o uso de notação padrão, mas é baseada no modelo funcional e independente de tecnologia	Baseada no modelo de casos de uso
Largamente discutida na literatura	Tem aumentado o uso e a publicação de estudos na literatura
É suportada pelo IFPUG e diversos grupos nacionais de usuários e bases históricas de medidas realizadas	Ainda não possui bons históricos de produtividade
Possui regras de contagem padronizadas	Há dúvidas de qual o nível apropriado de detalhe que cada caso de uso deve possuir
Alto nível de maturidade	Em fase de amadurecimento
Oferece treinamento e certificação	Ainda não oferece treinamentos e certificação

Modelos empíricos: Constructive Cost Model COCOMO (Bohem)

O modelo COCOMO original se tornou um dos mais estudados e usados pela indústria. Segundo o modelo o sistema é classificado segundo suas características como:

- orgânico,
- semidestacado e
- embutido.

Para cada tipo são atribuídos coeficientes empiricamente determinados que são usados nas fórmulas:

$$E = a(KLOC)^b$$

$$T = c \cdot E^d$$

Onde “E” é o esforço dado em pessoas-mês, “T” é o tempo de duração do projeto dado em meses e “KLOC” é o valor estimado para o tamanho do sistema dado em 1000 linhas de código.

Em 96, o modelo Cocomo evoluiu para um modelo mais abrangente chamado COCOMO II onde a estimativa é feita com base no número e complexidade dos objetos do sistema como: relatórios, telas e componentes.

ATIVIDADE

1. Cite e descreva os tipos de complexidade de casos de uso segundo a técnica de pontos por caso de uso.
2. Cite os fatores de ajuste técnico e ambiental

BIBLIOGRAFIA BÁSICA

PRESSMAN, R. S.. *Engenharia de Software*. Makron Books. 1995