

CRITÉRIOS DE QUALIDADE EM PROJETO DE SOFTWARE

- Estimativas -> métricas
- Padrões

CRITÉRIOS DE QUALIDADE EM PROJETO DE SOFTWARE

- Estimativas -> métricas

Segundo Deming,

“não se pode gerenciar o que não se consegue medir.”

O uso de métricas de software torna-se essencial para medir (tamanho, custo, prazo) e, consequentemente gerenciar melhor o desenvolvimento do software.

CRITÉRIOS DE QUALIDADE EM PROJETO DE SOFTWARE

- Estimativas -> métricas

Quanto mais precisa for a estimativa, maior a qualidade do nosso projeto. Por isso a importância de se estimar bem para a **qualidade de um projeto de software.**

TIPOS DE MÉTRICAS DE SOFTWARE

- Contagem de Linhas de Código Fonte (LOCs)
- Análise de Pontos por Função -> projetos estruturados
- **Análise por Casos de uso - > projetos O.O.**
- Outras Técnicas

ESTIMATIVAS BASEADAS EM CASOS DE USO

PONTOS POR CASO DE 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;

PONTOS POR CASO DE USO

- 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;

PONTOS POR CASO DE USO

- Sistema 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;

PONTOS POR CASO 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

PONTOS POR CASO DE USO

- No caso do exemplo, vamos supor os seguintes valores:

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

PONTOS POR CASO DE USO

- **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;

PONTOS POR CASO DE USO

- No caso do exemplo, vamos supor os seguintes valores:

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

PONTOS POR CASO DE USO

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

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

- No caso do exemplo:

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

PONTOS POR CASO DE USO

- 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;

PONTOS POR CASO DE USO

- **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
T1	Sistema distribuído	2
T2	Tempo de resposta	2
T3	Eficiência	1
T4	Processamento complexo	1
T5	Código reusável	1
T6	Facilidade de instalação	0.5
T7	Facilidade de uso	0.5
T8	Portabilidade	2
T9	Facilidade de mudança	1
T10	Concorrência	1
T11	Recursos de segurança	1
T12	Acessível por terceiros	1
T13	Requer treinamento especial	1

PONTOS POR CASO DE USO

- No caso do exemplo, vamos supor os valores:

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

PONTOS POR CASO DE USO

- **Passo 5:** Cálculo do TCF (Technical Complexity Factor)

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

- No caso do exemplo:

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

PONTOS POR CASO DE USO

- **Passo 6: Cálculo do Efactor**
 - 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	Descrição	Peso
E1	Familiaridade com RUP ou outro processo formal	1.5
E2	Experiência com a aplicação em desenvolvimento	0.5
E3	Experiência em Orientação a Objetos	1
E4	Presença de analista experiente	0.5
E5	Motivação	1
E6	Requisitos estáveis	2
E7	Desenvolvedores em meio-expediente	-1
E8	Linguagem de programação difícil	-1

PONTOS POR CASO DE USO

○ No caso do exemplo:

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	-1	0	0
			Efactor	26

Valores já calculados: **UUCP = 222, TCF = 0.795, Efactor = 26**

PONTOS POR CASO DE USO

- **Passo 7:** Cálculo do ECF (Environmental Complexity Factor)

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

- No caso do exemplo:

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

Valores já calculados: **UUCP = 222, TCF = 0.795, Efactor = 26, ECF = 0.62**

PONTOS POR CASO DE USO

- **Passo 8: Cálculo dos UCP (Use Case Points)**

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

- No caso do exemplo:

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

PONTOS POR CASO DE USO

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

ESTIMATIVA DE CUSTO DE DESENVOLVIMENTO

- O custo da hora-desenvolvimento varia de acordo com a especialização do profissional que irá realizar a tarefa.
- Para analistas, este valor se situa entre 80 e 100 reais por hora.
- Para programadores, entre 30 e 60 reais a hora.
- Na média, para horas de desenvolvimento de cada caso de uso, pode-se considerar R\$ 50,00

ESTIMATIVA DO CUSTO DE DESENVOLVIMENTO.

- É obtida a partir da multiplicação do número de casos de uso estimados, pelo valor médio da hora de desenvolvimento.
- Exemplo: para um sistema de 300 UCP, teríamos:

$$300 * 50,00 = 15.000,00$$

- Assim neste caso teríamos um custo de desenvolvimento de R\$ 15.000,00 (quinze mil reais)

ESTIMATIVA DO CUSTO DE DESENVOLVIMENTO

- Para cada empresa que desenvolve software, estes valores devem ser ajustados em função do que realmente ocorreu nos projetos já terminados e entregues ao usuário.
- Com o tempo, pode-se chegar a estimativas da proporcionalidade do envolvimento de programadores e analistas no projeto, fazendo-se cálculos mais realistas.

ESTIMATIVA DE CUSTO DO PROJETO

- Devem ser somados todos os custos envolvidos, desde o início do projeto até o seu final:
 - Custo de treinamento
 - Custo de hw
 - Custo do sw de apoio (licenças de BD, Ferramenta CASE, etc.)
 - Custo do desenvolvimento
 - Outros