Aula 4

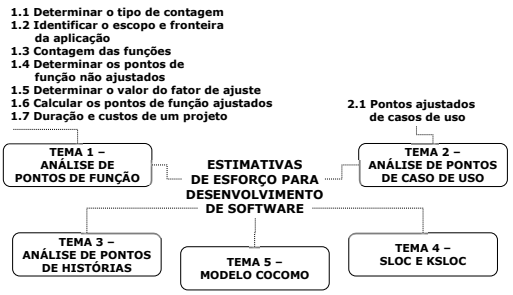
Engenharia de Software

Prof. Alex Mateus Porn

1

Conversa Inicial

2



3

Análise de pontos de função

4

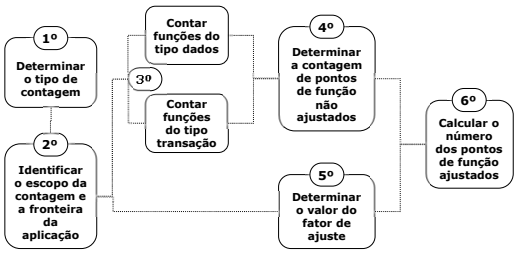
- **Técnica de medição do tamanho funcional de um software**
- **Possibilita estimar o esforço para implementação do sistema (Lopes, 2020)**
- **Tem por definição medir o que o software faz, e não como foi construído**

5

- **"Um ponto de função deve ser entendido como uma unidade de medida que procura definir o tamanho de um aplicativo (software), independentemente de como possa ser produzido e implementado. A ideia central é que a medida funcional do software esteja respaldada nos requisitos lógicos dos usuários" (Silva; Oliveira, 2005, p. 4)**

6

Processo de contagem de pontos de função



7

Determinar o tipo de contagem

- **Projeto de desenvolvimento: contagem de pontos de função associados à criação de um novo projeto**
- **Projeto de melhoria: contagem de pontos de função de funções adicionadas, modificadas ou eliminadas em um projeto existente**
- **Aplicação: contagem de pontos de função de um projeto finalizado**

8

Identificar o escopo de contagem e a fronteira da aplicação

- **Segundo Silva e Oliveira (2005):**
  - **Identificar o escopo refere-se a determinar se a contagem estará concentrada em um ou mais sistemas ou mesmo em parte de um sistema**
  - **A fronteira da aplicação aponta que ela estabelece um divisor entre os componentes do aplicativo e os componentes de outros aplicativos**

9

Contagem das funções

- **Segundo Lopes (2020):**
  - **Funções tipo dados: funcionalidades fornecidas para o armazenamento de dados da aplicação, podendo ser mantidos dentro ou fora dela**
    - ✓ **Arquivos lógicos internos (ALI): mantidos dentro da fronteira da aplicação**
    - ✓ **Arquivos de interface externa (AIE): mantidos fora da aplicação ou lidos de outra**

10

- **Complexidade das funções tipo dados**

	1 a 19 DER	20 a 50 DER	Acima de 50 DER
1 RLR	Simples	Simples	Média
2 a 5 RLR	Simples	Média	Complexa
Acima de 5 RLR	Média	Complexa	Complexa

Elaborado com base em Silva e Oliveira, 2005, p. 5

11

- **Pontos de função das funções tipo dados**

Função	Simples	Média	Complexa
ALI	7 pontos	10 pontos	15 pontos
AIE	5 pontos	7 pontos	10 pontos

Elaborado com base em Silva e Oliveira, 2005, p. 5

12

Exemplificando

- Projeto de melhoria
  - Processo de vendas
    - Tabelas "Vendas" e "Itens_Venda"
      - Vendas: "código", "cliente" e "data"
      - Itens_Venda: "código", "produto", "quantidade", "valor", "total"

13

Descrição	Tipo	RLR	DER	Complexidade	Pontos de função
Vendas	1 ALI	2 (Vendas e Itens_Venda)	6	Simples	7

14

Contagem das funções

- Segundo Lopes (2020):
  - Funções tipo transação: funcionalidades de processamento de dados do sistema
    - Entradas externas (EE)
    - Saídas externas (SE)
    - Consultas externas (CE)

15

- Conforme Vazquez, Simões e Albert (2009):
  - EE: processo elementar que manipula dados ou informações de controle originados fora da fronteira da aplicação
    - Tem como função manter (incluir, alterar ou excluir dados) um ou mais ALI

16

- Complexidade das entradas externas

	1 a 4 DER	5 a 15 DER	Acima de 15 DER
1 ALR	Simples	Simples	Média
2 ALR	Simples	Média	Complexa
Acima de 2 ALR	Média	Complexa	Complexa

Elaborado com base em Silva e Oliveira, 2005, p. 5

17

- SE: processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação
  - Tem como função apresentar informação ao usuário por meio de lógica de processamento que não seja apenas a recuperação de dados ou informações de controle

18

- **CE: processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação**
  - **Tem como principal intenção apresentar informação ao usuário por meio de uma simples recuperação de dados ou informação de controle de ALIs e/ou AIEs. A lógica de processamento não deve conter fórmulas matemáticas ou cálculos**

19

- **Complexidade das saídas e consultas externas**

	1 a 4 DER	5 a 19 DER	Acima de 19 DER
1 ALR	Simples	Simples	Média
2 a 3 ALR	Simples	Média	Complexa
Acima de 3 ALR	Média	Complexa	Complexa

Elaborado com base em Silva e Oliveira, 2005, p. 5

20

- **Pontos de função das funções tipo transação**

Função	Simples	Média	Complexa
EE	3 pontos	4 pontos	6 pontos
SE	4 pontos	5 pontos	7 pontos
CE	3 pontos	4 pontos	6 pontos

Elaborado com base em Silva e Oliveira, 2005, p. 5

21

Exemplificando

- **Projeto de melhoria**
  - **Criação dos processos elementares**
    - **Incluir, alterar e excluir vendas**
    - **Consultar clientes por idade**
    - **Consultar total de compras por clientes**
    - **Consultar produtos por ordem de preços**

22

Exemplificando

DESCRIÇÃO	TIPO	ALR	DER	COMPLEXIDADE
Incluir vendas	EE	1	8	Simples
Alterar vendas	EE	1	8	Simples
Excluir vendas	EE	1	8	Simples
Consultar clientes por idade	CE	1	9	Simples
Consultar produtos por ordem de preços	CE	1	6	Simples
Consultar total de compras por cliente	SE	3	12	Média

23

Exemplificando

- **3 EE simples x 3 = 9 pontos de função**
- **2 CE simples x 3 = 6 pontos de função**
- **1 SE média x 5 = 5 pontos de função**
- **Total de 20 pontos de função tipo transação**

24

Determinar a contagem de pontos de função não ajustados

- Total de pontos de função tipo dados
- Total de pontos de função tipo transação
- Soma-se os dois resultados

25

- Analisando nosso exemplo:
  - 7 pontos de função tipo dados
  - 20 pontos de função tipo transação
  - Total de 27 pontos de função não ajustados

26

Determinar o valor do fator de ajuste

- Itens de influência

Comunicação de dados	Funções distribuídas
Performance	Utilização do equipamento
Volume de transações	Entrada de dados on-line
Interface com o usuário	Atualizações on-line
Processamento complexo	Reusabilidade
Facilidade de implantação	Facilidade operacional
Múltiplos locais	Facilidade de mudanças

Elaborado com base em Silva e Oliveira, 2005, p. 7

Fator de ajuste = (pontos de influência x 0,01) + 0.65

27

Exemplificando

Funções distribuídas	– 2 pontos
Interface com o usuário	– 5 pontos
Reusabilidade	– 3 pontos
Facilidade de implantação	– 3 pontos
Múltiplos locais	– 4 pontos

Fa = (17 x 0.01) + 0.65
Fa = 0.82

28

Calcular o número de pontos de função ajustados (AFP)

- Multiplicar o total de pontos não ajustados pelo fator de ajuste
- Voltando ao nosso exemplo:
  - 27 pontos de função não ajustados
  - Fator de ajuste de 0,82
  - 27 x 0,82 = 22,14 pontos de função ajustados

29

Duração de um projeto

- Cálculo do esforço total de desenvolvimento
- Total de pontos de função produzidos por hora
- Total de horas trabalhas por mês
- Exemplo:
  - Java – 15 horas por ponto de função
  - PHP – 11 horas por ponto de função
  - Python – 7 horas por ponto de função

Esforço = 7 x 22,14 = 154,98 horas

30

Custo de um projeto

- Cálculo do custo do projeto

$$Custo = E * Custo_{horaDev}$$

Fonte: Wazlawick, 2013, p. 164

154,98 x R$27,00 = R$4.184,46

31

Pontos de casos de uso

32

- "O método se baseia na análise da quantidade e complexidade dos atores e casos de uso, o que gera os UUCP, ou pontos de caso de uso não ajustados. Depois, a aplicação e os fatores técnicos e ambientais levam aos UCP, ou pontos de caso de uso ajustados" (Wazlawick, 2013, p. 171)

33

Complexidade dos atores

- Segundo Wazlawick (2013):
  - Atores humanos que interagem com o sistema por meio de interface gráfica
    - Complexo: 3 pontos de caso de uso

34

- Sistemas que interagem por um protocolo como TCP/IP e atores humanos que interagem com o sistema apenas por linha de comando
  - Média complexidade: 2 pontos de caso de uso
- Sistemas que são acessados por interfaces de programação (API)
  - Baixa complexidade: 1 ponto de caso de uso

35

Complexidade dos casos de uso

- De acordo com Wazlawick (2013):
  - Definida em função do número estimado de transações
    - Casos de uso simples: com até 3 transações valem 5 pontos
    - Casos de uso médios: de 4 a 7 transações valem 10 pontos
    - Casos de uso complexos: acima de 7 transações valem 15 pontos

36

- Definida em função da quantidade de classes necessárias para implementar as funções do caso de uso
  - Casos de uso simples: com 5 classes ou menos
  - Casos de uso médios: com 6 a 10 classes
  - Casos de uso complexos: com mais de 10 classes

37

- Definida pela análise de seu risco
  - Simples: casos de uso como relatórios que têm apenas uma ou duas transações e baixo risco
  - Médios: casos de uso padronizados que têm um número conhecido e limitado de transações
  - Complexos: casos de uso não padronizados que têm um número desconhecido de transações e alto risco

38

Pontos ajustados de casos de uso

- Fatores técnicos

FATOR TÉCNICO	PESO	FATOR TÉCNICO	PESO
Sistema distribuído	2	Performance	2
Eficiência de usuário final	1	Complexidade de processamento	1
Projeto visando código reusável	1	Facilidade de instalação	0,5
Facilidade de uso	0,5	Portabilidade	2
Facilidade de mudança	1	Concorrência	1
Segurança	1	Acesso fornecido a terceiros	1
Necessidades de treinamento	1		

Fonte: Elaborado com base em Wazlawick, 2013, p. 172

TCF = 0,6 + (0,01 * Total da soma dos pontos de fatores técnicos)

39

Pontos ajustados de casos de uso

- Fatores ambientais

FATOR AMBIENTAL	PESO	FATOR AMBIENTAL	PESO
Familiaridade com o processo de desenvolvimento	1,5	Experiência com a aplicação	0,5
Experiência com orientação a objetos	1	Capacidade do analista líder	0,5
Motivação	1	Estabilidade de requisitos obtida historicamente	2
Equipe em tempo parcial	-1	Dificuldade com a linguagem de programação	-1

Fonte: Elaborado com base em Wazlawick, 2013, p. 172

EF = 1,4 – (0,03 * Total da soma dos fatores ambientais)

40

Total de pontos de casos de uso ajustados

UCP = UUCP * TCF * EF

41

Pontos de histórias

42

- "Um ponto de história não é uma medida de complexidade funcional, como os pontos de função ou pontos de caso de uso, mas uma medida de esforço relativa à equipe de desenvolvimento" (Wazlawick, 2013, p. 177)

43

- Uma história de usuário corresponde a uma explicação informal e geral sobre um recurso de software, escrita de acordo com a perspectiva do usuário final (Kniberg, 2007)
- Consiste em identificar de forma subjetiva, com a equipe de desenvolvimento, quanto tempo um número qualquer de pessoas que se dedicassem unicamente a uma história de usuário levariam para terminá-la com uma versão executável do sistema

44

- Multiplica-se o número de pessoas pela quantidade de dias para se obter o total de pontos de história
- Por exemplo:
  - 2 pessoas levariam 6 dias para implementar determinada história de usuário
  - Atribui-se à história 2 x 6 = 12 pontos de história

45

SLOC e KSLOC

46

SLOC e KSLOC

- SLOC (*Source Line of Code*)
  - Estima, com base na opinião de especialistas e no histórico de projetos passados, quantas linhas de código o projeto terá
- KSLOC (*Kilo Source Line of Code*)
  - Uma unidade KSLOC vale mil unidades SLOC. Além disso, também são usados os termos MSLOC para milhões de linhas e GSLOC para bilhões de linhas de código

47

Técnica de estimativa KSLOC

- Conforme Wazlawick (2013):
  - KSLOC otimista: número mínimo de linhas que se espera desenvolver se todas as condições forem favoráveis
  - KSLOC pessimista: número máximo de linhas que se espera desenvolver em condições desfavoráveis
  - KSLOC esperado: número de linhas que efetivamente se espera desenvolver em uma situação de normalidade

KSLOC = (4 x KSLOCesperado + KSLOCotimista + KSLOCpessimista) / 6

48

COCOMO

49

Implementações de complexidade

- Para Wazlawick (2013, p. 134):
  - Implementação básica: quando se tem somente o número estimado de linhas de código
  - Implementação intermediária: quando fatores relativos ao produto, ao suporte computacional, pessoal e ao processo são conhecidos

(...)

50

(...)

- Implementação avançada: quando for necessário subdividir o sistema em subsistemas e distribuir as estimativas de esforço por fase e atividade

51

Tipo de projetos

- Segundo Wazlawick (2013, p. 134):
  - Modo orgânico: quando o sistema é de baixa complexidade e a equipe está acostumada a desenvolver esse tipo de aplicação
  - Modo semidestacado: maior grau de novidade para a equipe, em que a combinação do risco tecnológico e pessoal seja média
  - Modo embutido: alto grau de complexidade, embarcados e para os quais a equipe tenha considerável dificuldade de abordagem

52

Estimativas

- As três implementações permitem determinar três informações básicas
  - O esforço estimado em desenvolvedor/mês, dado pela notação E
  - O tempo linear de desenvolvimento sugerido em meses corridos, dado pela notação T
  - O número médio de pessoas recomendado para a equipe, dado pela notação P

53

Estimativa de esforço do modelo básico

- Esforço estimado em desenvolvedor/mês

$$E = ax * KSLOC^{bx}$$

- Tempo linear de desenvolvimento

$$T = cx * E^{dx}$$

- Número médio de pessoas recomendado para a equipe

$$P = E / T$$

54

Constantes para o cálculo de estimativas do modelo básico

TIPO	ax	bx	cx	dx
Orgânico	2,4	1,05	2,5	0,38
Semidestacado	3,0	1,12	2,5	0,35
Embutido	3,6	1,2	2,5	0,32

Elaborado com base em Wazlawick, 2013, p. 135

55

Estimativa de esforço dos modelos intermediário e avançado

- Considera 15 fatores influenciadores de custo

Fatores influenciadores de custo	Acrônimo	Muito baixa	Baixa	Média	Alta	Muito alta	Extra alta
Relativos ao produto							
Nível de confiabilidade requerida	RELY	0,75	0,88	1,00	1,15	1,40	
Dimensão da base de dados	DATA		0,94	1,00	1,08	1,16	
Complexidade do produto	CPLX	0,70	0,85	1,00	1,15	1,30	1,65
Suporte computacional							

Elaborado com base em Wazlawick, 2013, p. 136

56

Estimativa de esforço dos modelos intermediário e avançado

- Cálculo dos influenciadores de custo

$$EAF = RELY * DATA * CPLX * TIME * ... * SCED$$

- Estimativa de esforço

$$E = ai * KSLOC^{bi} * EAF$$

TIPO	ai	bi
Orgânico	2,8	1,05
Semidestacado	3,0	1,12
Embutido	3,2	1,2

Fonte: Wazlawick, 2013, p. 137

57