

# Engenharia de Software II

## Aula 13

<http://www.ic.uff.br/~bianca/engsoft2/>

# Ementa

- Processos de desenvolvimento de software (Caps. 2, 3 e 4 do Pressman)
- Estratégias e técnicas de teste de software (Caps. 13 e 14 do Pressman)
- **Métricas para software (Cap. 15)**
- Gestão de projetos de software: conceitos, métricas, estimativas, cronogramação, gestão de risco, gestão de qualidade e gestão de modificações
- Reengenharia e engenharia reversa

# Métricas para Software

- Métricas para o modelo de análise
  - Métricas baseadas em função
  - Métricas de qualidade de especificação
- Métricas para o modelo de projeto
- Métricas de código fonte
- Métricas para teste

# Métricas para o Modelo de Análise

- Examinam o modelo de análise com um dos seguintes objetivos:
  1. Prever o “tamanho” ou “complexidade” do sistema resultante.
    - O tamanho é um indicador do esforço de codificação, integração e teste.
  2. Avaliar a qualidade do modelo de análise e da correspondente especificação de requisitos.

# Métricas Baseadas em Função

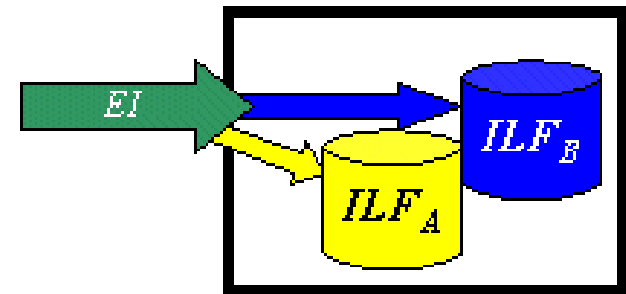
- Métrica **ponto por função** (ou *FP=function points*)
  - Usada para a medir a funcionalidade entregue por um sistema.
- Usando **dados históricos**, o FP pode ser usado para:
  - Estimar o custo ou esforço necessário para projetar, codificar e testar o software.
  - Prever o número de erros que vão ser encontrados durante o teste.
  - Prever o número de componentes e/ou o número de linhas de código.
- Pontos por função são derivados usando uma relação empírica baseada em medidas de **contagem direta** de características do software.

# **Características consideradas no cálculo de FP**

- Número de entradas externas (EIs)
- Número de saídas externas (EOs)
- Número de consultas externas (EQs)
- Número de arquivos lógicos internos (ILFs)
- Número de arquivos de interface externa (EIFs)

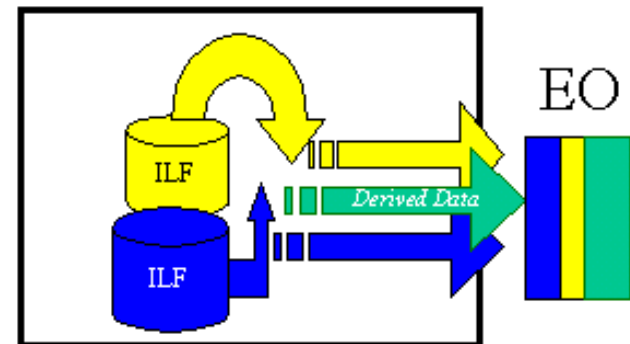
# Entradas Externas (EIs)

- Se originam de um usuário ou outra aplicação.
- Fornecem dados distintos orientados à aplicação do software ou informação de controle.
- São freqüentemente usadas para atualizar arquivos lógicos internos.
- Devem ser distinguidas de consultas, que são contadas separadamente.



# Saídas Externas (EOs)

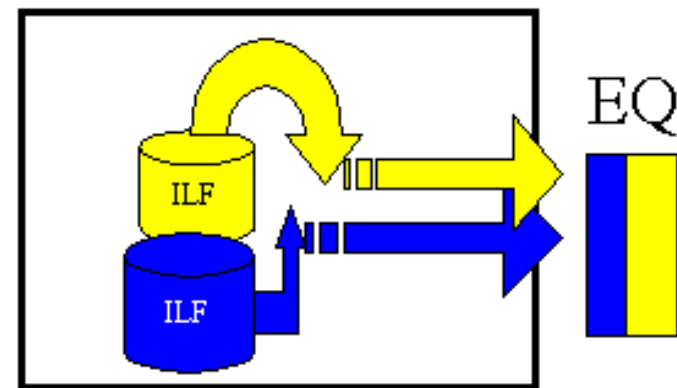
- São derivadas de dentro da aplicação e fornecem informação para o usuário.
- Referem-se a relatórios, telas, mensagens de erro, etc.
- Itens de dados individuais dentro de um relatório não são contados individualmente.





# Consultas Externas (EQs)

- São entradas on-line, que resultam na geração de alguma saída imediata do software.
- A entrada não modifica nenhum arquivo.
- A saída não contém dados derivados.



# Arquivos Lógicos Internos (ILFs)

- Agrupamentos lógicos de dados que residem dentro das fronteiras da aplicação.
- Mantidos por entradas externas.

# Arquivos de Interface Externa (EIFs)

- São agrupamentos lógico de dados que residem externamente à aplicação.
- Fornecem dados que podem ser úteis à aplicação.

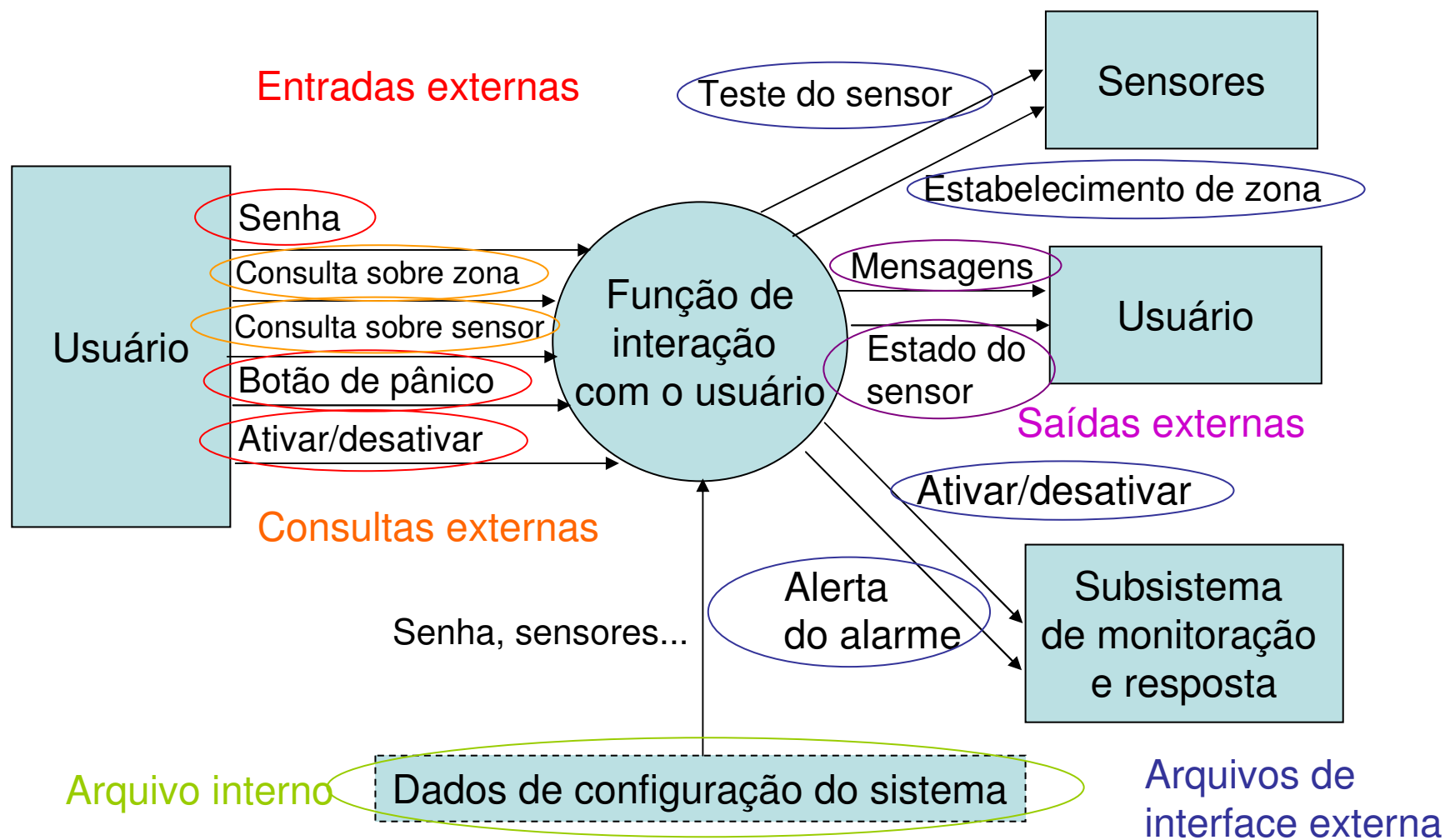
# Cálculo de Pontos por Função

		Fator de Ponderação				
Contagem		Simple	Médio	Complexo	Total	
EIs		×	3	4	6	=
EOs		×	4	5	7	=
EQs		×	3	4	6	=
ILFs		×	7	10	15	=
EIFs		×	5	7	10	=
Total de Contagem						

# Cálculo de Pontos por Função

- **FP = Total de Contagem  $\times$  [0,65 + 0,01  $\times$   $\Sigma(F_i)$ ]**
- Os  $F_i$  são fatores de ajuste baseados nas respostas a perguntas como:
  - O sistema requer *backup* e *recovery*?
  - Existem funções de processamento distribuído?
  - O desempenho é crítico?
  - O sistema será executado em um ambiente operacional existente, intensamente utilizado?
  - O sistema requer entrada de dados on-line?
  - A entrada de dados on-line exige várias telas?
  - Os ILFs são atualizados on-line?
  - O código é projetado para ser reusado?
- Cada uma das questões é respondida usando uma escala de 0 a 5.

# Exemplo: Diagrama de Fluxo de Dados



# Exemplo: Pontos por Função

	Fator de Ponderação				
	Contagem	Simple	Médio	Complexo	Total
EIs	3	× 3	4	6	= 9
EOs	2	× 4	5	7	= 8
EQs	2	× 3	4	6	= 6
ILFs	1	× 7	10	15	= 7
EIFs	4	× 5	7	10	= 20
Total de Contagem =					50

# Exemplo

- Supondo-se que, respondendo às perguntas, estimou-se que  $\Sigma(\mathbf{F}_i)$  é 46, temos:

$$\mathbf{FP} = 50 \times [0,65 + 0,01 \times 46] = 56$$

- Supondo-se que dados anteriores indiquem que 12 FPs são produzidos para cada pessoa-mês de esforço, pode-se fazer um planejamento para o projeto.



# Métricas de Qualidade de Especificação

- Supondo que há  $n_r$  requisitos em uma especificação:
  - $n_r = n_f + n_{nf}$
  - onde  $n_f$  é o número de requisitos funcionais,
  - onde  $n_{nf}$  é o número de requisitos não-funcionais.
- Especificidade (não-ambigüidade) pode ser determinada por:
  - $Q_e = n_{ui}/n_r$
  - onde  $n_{ui}$  é o número de requisitos para quais revisores diferentes dão a mesma interpretação.
- Completeza pode ser determinada por:
  - $Q_c = n_u/[n_i \times n_s]$
  - onde  $n_u$  é o número de requisitos funcionais únicos,  $n_i$  é o número de entradas especificadas e  $n_s$  é o número de estados especificados.
- Completeza global pode ser determinada por:
  - $Q_3 = n_c/[n_c \times n_{nv}]$
  - onde  $n_c$  é o número de requisitos que já foram validados como corretos e  $n_{nv}$  é o número de requisitos que ainda não foram validados.

# **Métricas para Modelo de Projeto**

- Métricas de projeto arquitetural
- Métricas para o modelo de projeto OO

# Métricas de Projeto Arquitetural

- Focalizam as características da arquitetura com ênfase na estrutura e efetividade dos módulos ou componentes.
- São “caixa-preta”.
- Três indicadores da complexidade do projeto:
  - Complexidade estrutural
  - Complexidade dos dados
  - Complexidade de sistema
- DSQI: indicador da qualidade de estrutura do projeto.

# Complexidade Estrutural

- Serve para arquiteturas hierárquicas.
- A complexidade estrutural é dada por:

$$S(i) = f_{out}^2(i)$$

onde  $f_{out}(i)$  é o *fan-out* do módulo  $i$ ; *fan-out* é o número de módulos diretamente subordinados ao módulo  $i$ .

# Complexidade dos Dados

- Fornece uma indicação da complexidade na interface interna de um módulo  $i$  e é definida como:

$$D(i) = v(i)/[f_{out}(i) + 1]$$

onde  $v(i)$  é o número de variáveis de entrada e saída que são passadas para e do módulo  $i$ .

# Complexidade do Sistema

- É definida como a soma da complexidade estrutural e de dados.

$$C(i) = S(i) + D(i)$$

- À medida que complexidade global aumenta, maior a probabilidade de aumento do esforço de integração e teste.

# DSQI

- DSQI = *Design Structure Quality Index* = Índice de qualidade da estrutura do projeto
- **DSQI =  $\sum w_i D_i$**   
onde  $w_i$  é um peso e os valores  $D_i$  é são dados por:
  - $D_1$  : Estrutura de programa
  - $D_2$  : Independência modular
  - $D_3$  : Módulos não dependentes de processamento anterior
  - $D_4$  : Tamanho da base de dados
  - $D_5$  : Compartimentalização da base de dados
  - $D_6$  : Característica de entrada/saída do módulo

# DSQI

- $D_1$  : Estrutura de programa
  - Se um método específico foi utilizado para o projeto então  $D_1 = 1$ , senão  $D_1 = 0$ .
- $D_2$  : Independência modular
  - $D_2 = 1 - (S_2/S_1)$  senão  $D_2 = 0$ .
    - $S_1$  é o número total de módulos do programa.
    - $S_2$  é o número de módulos cujo funcionamento correto depende da fonte de entrada de dados ou que produz dados a serem usados em outro lugar.
- $D_3$  : Módulos independentes de processamento anterior
  - $D_3 = 1 - (S_3/S_1)$  senão  $D_3 = 0$ .
    - $S_1$  é o número total de módulos do programa.
    - $S_3$  é o número de módulos cujo funcionamento correto depende de processamento anterior.



# DSQI

- $D_4$  : Tamanho da base de dados
  - $D_4 = 1 - (S_5/S_4)$ 
    - $S_5$  é o número total de itens únicos na base de dados.
    - $S_4$  é o número de itens na base de dados.
- $D_5$  : Compartimentalização da base de dados
  - $D_5 = 1 - (S_6/S_4)$ 
    - $S_6$  é o número de segmentos da base de dados.
    - $S_4$  é o número de itens na base de dados.
- $D_6$  : Característica de entrada/saída do módulo
  - $D_6 = 1 - (S_7/S_1)$ 
    - $S_7$  é o número de módulos com uma única entrada e saída.
    - $S_1$  é o número total de módulos do programa.

# Métricas para o modelo de projeto OO

- Métricas CK
- Métricas MOOD
- Métricas de Lorenz e Kidd

# Métricas CK

- A classe é a unidade fundamental de um sistema OO.
  - Métricas para uma classe individual, para a hierarquia de classes e para as colaborações entre classes são de grande valor.
- O conjunto CK (Chidamber e Kernerer) contém seis métricas para projeto OO.
  - Métodos ponderados por classe (WMC)
  - Profundidade da árvore de herança (DIT)
  - Número de filhos (NOC)
  - Acoplamento entre as classes de objetos (CBO)
  - Resposta de uma classe (RFC)
  - Falta de coesão de métodos (LCOM)