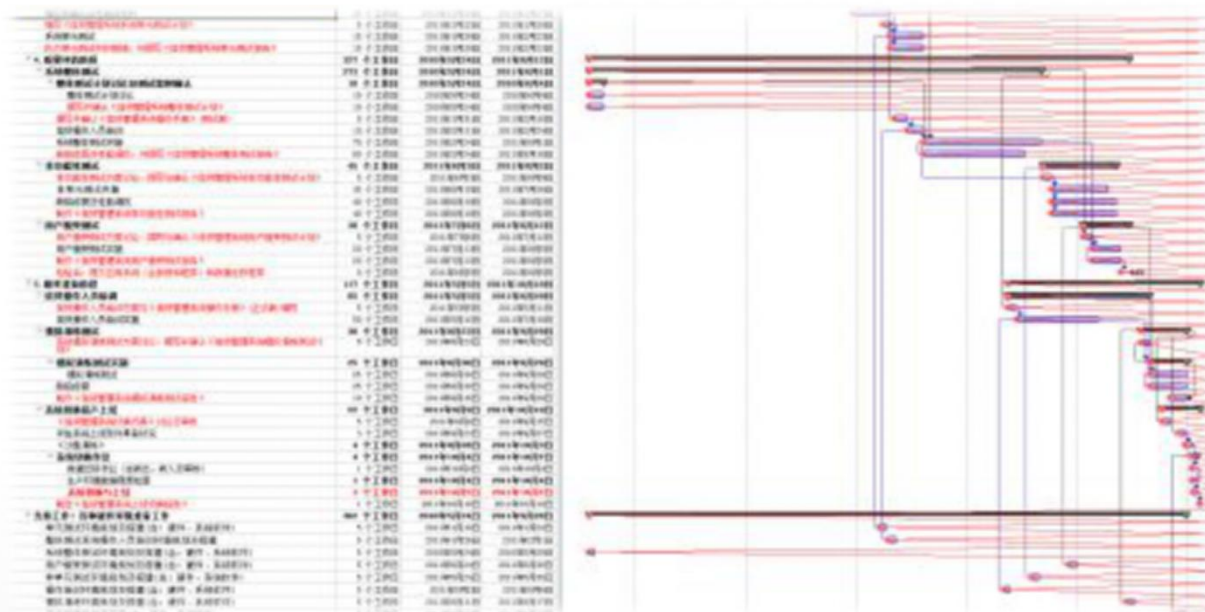


面向功能的度量

- 面向功能度量的概念
- 功能点法计算公式
- UFC相关五类组件
- 复杂性调节因素
- 面向功能的度量优缺点



面向功能的度量概念

- 用软件的功能表示软件的规模
 - 应用最广泛的是功能点（Function Point, FP）法
 - 项目开发初期就可估算出
 - 功能点计算目前主要基于经验公式
-

功能点计算方法

- $FP = UFC \times TCF = UFC \times (0.65 + 0.01 \times \sum Fi)$
 - UFC (Unadjusted Function Component) : 未调整功能点计数, 5个信息量的 “加权和”
 - TCF (Technical Complexity Factor): 技术复杂度因子
 - Fi : 14个因素的 “复杂性调节值” ($i = 1..14$)
 - 0.65, 0.01都是经验常数
-

UFC相关的五类组件

- 内部逻辑文件（ ILF, Internal Logical Files ）

- 一个用户可识别的逻辑相关的数据组，它在应用程序边界内，由用户输入来维护
- 它可能是某个大型数据库的一部分或是一个独立的文件

- 外部接口文件（ EIF, External Interface Files ）

- 一个用户可识别的逻辑相关的数据组，但只能被引用，且数据完全存于软件外部，由另一个应用程序进行维护
- 是机器可读的全部接口（如磁盘或磁带上的数据文件）
- 是另一个应用程序的内部逻辑文件

UFC相关的五类组件

▪ 外部输入 (EI, External Input)

- 来自于软件外部的数据输入
- 控制信息(不更新ILF) / 业务逻辑信息 (更新ILF)
- 可来自于一个数据输入屏幕或其他应用程序。

▪ 外部输出 (EO, External Output)

- 经过处理的数据，由程序内部输出到外部
- 从ILF、EIF中取出数据经过一定的组合、计算后得出的输出数据, 如生成报表, 派生数据, 可能更新ILF

▪ 用户查询 (EQ, External Query)

- 一个输入输出的组合过程，从一个或多个ILF、EIF中取出数据输出到程序外部
- 输入过程不更新ILF，输出过程不进行任何数据处理

UFC相关的五类组件

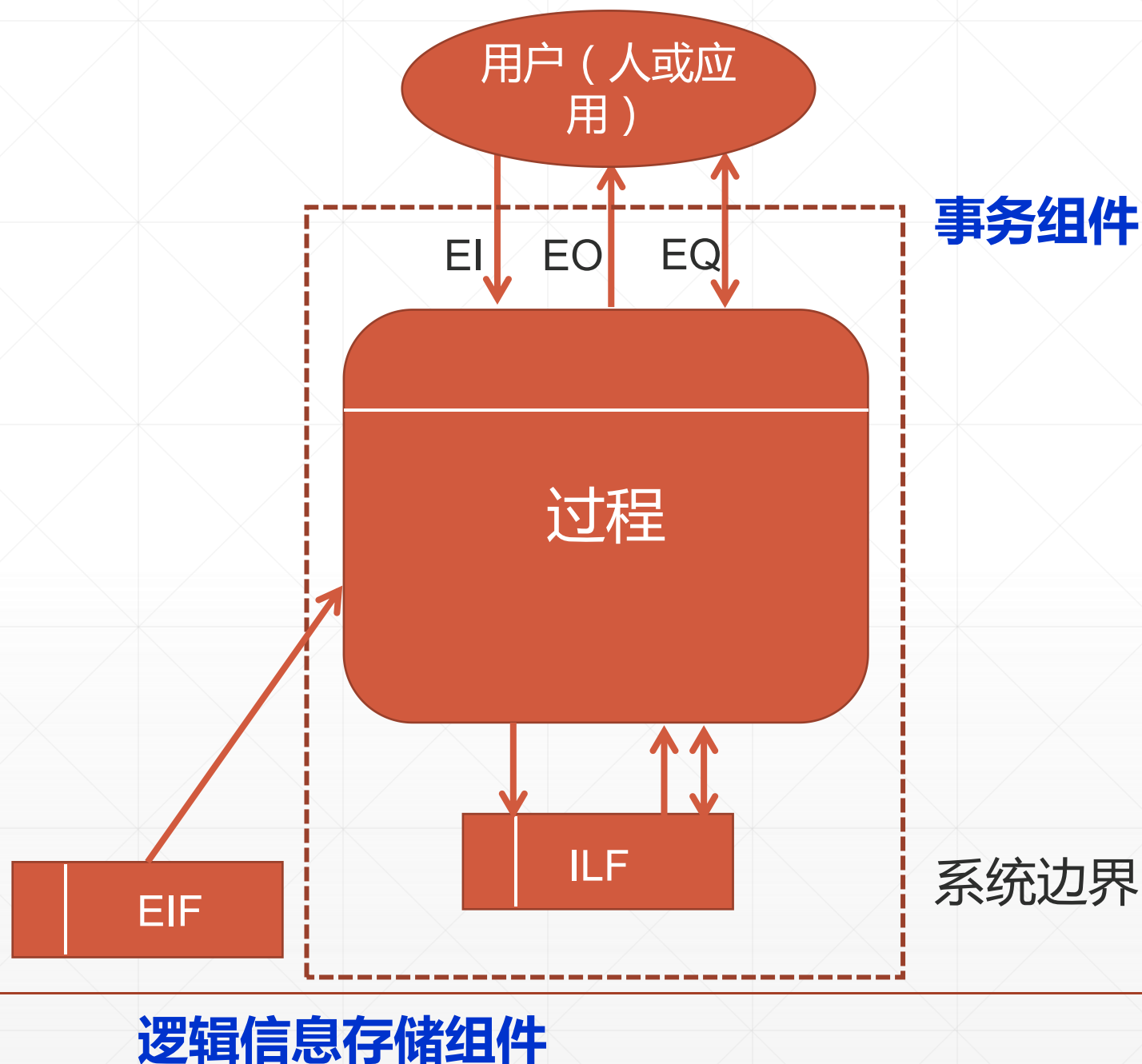
EI : External Input (外部输入)

EO : External Output (外部输出)

EQ : External Query (外部查询)

ILF : Internal Logical File (内部逻辑文件)

EIF : External Interface File (外部逻辑文件)



UFC相关的五类组件

功能组件复杂度加权因子表

功能组件类型	简单	中等	复杂
外部输入数EI	3	4	6
外部输出数EO	4	5	7
外部查询表EQ	3	4	6
内部逻辑文件数ILF	7	10	15
外部接口文件数EIF	5	7	10

UFC的计算方法

- 外部输入EI数×加权因子(简单=3,平均=4,复杂=6)
- 外部输出EO数×加权因子(简单=4,平均=5,复杂=7)
- 外部查询EQ数×加权因子(简单=3,平均=4,复杂=6)
- 内部逻辑文件ILF数×加权因子(简单=7,平均=10,复杂=15)
- 外部接口EIF数×加权因子(简单=5,平均=7,复杂=10)

UFC = 上述计算值的总和 (加权和)

UFC计算例子

- 假设每个功能要素的复杂度都是平均的。若有一个由25个数据登记表、15个报告、10个外部查询、20个逻辑内部表单和5个接口文件组成的学生管理系统，其未调整功能点计数为：

$$\begin{aligned} \text{UFC} &= (25 \times 4) + (15 \times 5) + (10 \times 4) + (20 \times 10) \\ &+ (5 \times 7) = 450 \text{ (个功能点)} \end{aligned}$$

14个复杂性调节因素 F_i

1. 系统需要可靠的备份和复原吗？
2. 系统需要数据通信吗？
3. 系统有分布处理功能吗？
4. 性能是临界状态吗？
5. 系统是否在一个紧张使用的操作系统下运行？
6. 系统需要联机数据项吗？
7. 联机数据项是否在多屏幕或多操作之间进行切换？

复杂性调节因素值 F_i

0-没有影响

1-偶有影响

2-轻微影响

3-平均影响

4-较大影响

5-严重影响

14个复杂性调节因素 F_i

- 8. 需要联机更新主文件吗？
- 9. 输入、输出、查询和文件很复杂吗？
- 10. 内部处理复杂吗？
- 11. 代码需要被设计成可重用吗？
- 12. 设计中需要包括转换和安装吗？
- 13. 系统的设计支持不同组织的多次安装吗？
- 14. 应用的设计方便用户修改和使用吗？

复杂性调节因素值 F_i

0-没有影响

1-偶有影响

2-轻微影响

3-平均影响

4-较大影响

5-严重影响

$$FP = UFC \times (0.65 + 0.01 \times \sum F_i)$$

面向功能的度量

- 优点

- 与程序设计语言无关, 在开发前就可以估算出软件项目的规模

- 不足

- 没有直接涉及算法的复杂度, 不适合算法比较复杂的软件系统;
 - 功能点计算主要靠经验公式, 主观因素比较多
-

面向功能的度量

一个功能点所需的代码行粗略估算

程序语言	平均值	中值	低值	高值
C	162	109	33	704
C++	66	53	29	178
ASP	86	83	20	184
HTML	43	42	35	53
Java	63	53	9	214
Javascript	58	63	42	75
SQL	40	37	7	110

