SOFTWARE TESTING

Software Testing

Software Defect Management (2)

School of Computer Science & Engineering
Sun Yat-sen University

Instructor: Guoyang Cai
email: isscgy@mail.sysu.edu.cn

Approaches & Technologies

中山大學
SUN YAT-SEN UNIVERSITY

OUTLINE

- 8.1 概述
- 8.2 软件缺陷描述与分类
- 8.3 软件缺陷的处理与跟踪
- 8.4 软件缺陷报告
- 8.5 软件缺陷的度量与分析
- 8.6 软件缺陷管理工具

软件缺陷报告的概念

- 软件缺陷报告是软件测试过程中最重要的文档。
  - 在软件测试过程中，对发现的每一个软件错误，需要记录其缺陷特征和再现步骤等信息，并存储到软件缺陷数据库中。
    - 软件缺陷数据库的每一条记录称为一个软件缺陷报告，是分析、处理和管理相应的软件错误的依据。
      - 每个软件缺陷报告只描述一个缺陷或错误。
    - 软件缺陷报告记录了缺陷发生的环境。
      - 包括各种资源的现场配置情况、缺陷的再现步骤以及缺陷性质的说明等内容。
    - 软件缺陷报告记录了缺陷的处理过程和当前状态。
      - 缺陷的处理进程在一定角度上反映了测试的进程和被测软件的质量状况以及改善过程。

- 软件缺陷报告的概念
  - 软件缺陷报告的阅读
    - 缺陷报告的直接阅读者是软件开发人员和质量管理人员，来自市场和技术支持等部门的人也可能需要查看缺陷情况。
      - 软件开发人员希望获得缺陷的本质特征和再现步骤；
      - 市场和技术支持等部门希望获得缺陷类型分布以及对市场和用户的影响程度。

- 软件缺陷报告的概念
  - 软件缺陷报告的 “5C” 准则
    - Correct (准确)
      - 每个组成部分的描述准确，不会引起误解。
    - Clear (清晰)
      - 每个组成部分的描述清晰，易于理解。
    - Concise (简洁)
      - 报告只包含必不可少的信息，不包括任何多余的内容。
    - Complete (完整)
      - 报告包含复现该缺陷的完整步骤和其他本质信息。
    - Consistent (一致)
      - 按照一致的格式书写全部缺陷报告。

- 软件缺陷报告的组织结构
  - 软件缺陷报告的基本内容
    - 缺陷的标题与简述
    - 缺陷的基本信息
      - 软件名称、版本号、缺陷或错误类型、可重复性、测试平台、平台语言、缺陷或错误范围、严重程度及优先级。
    - 缺陷再现的操作步骤
      - 描述该软件缺陷或错误出现的操作顺序，要求完整、简洁、准确。对命令、系统变量、选项要用大写字母，对控件名称等加双引号。
    - 缺陷的实际结果描述、期望的正确结果描述
    - 注释文字和截取的缺陷图像
      - 对软件缺陷或错误的附加描述，一般包括缺陷或错误现象的图像，包括其他建议或注释文字。

软件缺陷报告的组织结构

软件缺陷条目列表

分类	项目	描述
可跟踪信息	缺陷 ID	唯一的、自动产生的缺陷 ID，用于缺陷的识别、跟踪、查询
软件缺陷基本信息	缺陷状态	可分为“打开或激活的”、“已修正的”、“关闭的”等等
	缺陷标题	描述缺陷的最主要信息
	缺陷的严重程度	一般分为“致命”、“严重”、“一般”、“较小”等四种程度
	缺陷的优先级	描述处理缺陷的紧急程度，可分为高优先级、正常优先级和低优先级3个级别
	缺陷的产生频率	描述缺陷发生的可能性1%-100%
	缺陷提交人	缺陷提交人的名字 (邮件地址)，一般就是发现缺陷的测试人员或其他人员

软件缺陷报告的组织结构

软件缺陷条目列表 (续)

分类	项目	描述
软件缺陷基本信息	缺陷提交时间	缺陷提交的时间
	缺陷所属项目/模块	缺陷所属的项目和模块，最好能较精确的定位至模块
	缺陷指定解决人	估计修复这个缺陷的开发人员，在缺陷状态下由开发组长指定相关的开发人员；也会自动和该开发人员的邮件地址联系起来，并自动发出邮件
	缺陷指定解决时间	开发管理员指定的开发人员修改此缺陷的时间
	缺陷验证人	验证缺陷是否真正被修复的测试人员；也会和邮件地址联系起来
	缺陷验证结果描述	对验证结果的描述 (通过、不通过)
	缺陷验证时间	对缺陷验证的时间

软件缺陷报告的组织结构

软件缺陷条目列表 (续)

分类	项目	描述
缺陷的详细描述	步骤	按照操作步骤逐步描述缺陷的操作过程
	期望的结果	按照设计规格说明书或用户需求，在完成上述步骤之后所期望的结果 (正确的结果)
	实际发生的结果	程序或系统实际发生的结果 (错误的结果)
测试环境说明	测试环境	对包括操作系统、浏览器、网络带宽、通讯协议等测试环境的描述
必要的附件	图片、日志文件	对于某些文字很难表达清楚的缺陷，使用图片等附件是必要的；对于软件崩溃现象，需要使用软件工具捕捉日志文件，作为附件提供给开发人员。

软件缺陷报告的组织结构

- 例：好的缺陷报告

重现步骤：
(a) 打开一个编辑文字的软件并且创建一个新的文档 (这个文件可以录入文字)
(b) 在这个文件里随意录入一两行文字
(c) 选中两行文字，通过选择 Font 菜单然后选择 Arial 字体格式
(d) 这两行文字变成了无意义的乱字符

期望结果：当用户选择已录入的文字并改变文字格式的时候，文本应该显示正确的文字格式，不会出现乱字符显示。
实际结果：它是字体格式的问题，如果在改变文字格式成 Arial 之前保存文件，缺陷不会出现。缺陷仅仅发生在Windows98并且改变文字格式成其它的字体格式，文字是显示正常的。参见所附的图片<有一个链接，点击即可看到>

评语：记录下最少的重复步骤，包括了期望结果，实际结果和必要的附件，还提供必要的数据、测试环境或条件，以及简单的分析。

软件缺陷报告的组织结构

- 例：含糊而不完整的缺陷报告

重现步骤：
- 打开一个编辑文字的软件.
- 录入一些文字
- 选择 Arial 字体格式
- 文字变成了乱字符

期望结果：
实际结果：

评语：缺少重建步骤，并且没有期望结果、实际结果和必要的图片

软件缺陷报告的组织结构

- 例：散漫的缺陷报告

重现步骤：

- 在 Window98 上打开一个编辑文字的软件并且编辑存在文件
- 文件字体显示正常
- 我添加了图片，这些图片显示正常
- 在此之后，我创建了一个新的文档
- 在这个文档中我随意录入了大量的文字
- 在我录入这些文字之后，选择几行文字.并且通过选择 Font 菜单然后选择 Arial 字体格式改变文字的字体。
- 有三次我重现了这个缺陷
- 我在 Solaris 操作系统运行这些步骤，没有任何问题。
- 我在 Mac 操作系统运行这些步骤，没有任何问题。

期望结果：当用户选择已录入的文字并改变文字格式的时候，文本应该显示正确的文字格式不会出现乱字符显示。

实际结果：我试着选择少量的不同的字体格式，但是只有 Arial 字体格式有软件缺陷，不论如何，它可能会出现在我没有测试的其它的字体格式

评语：无关的重建步骤，以及对开发人员理解这个错误毫无帮助的结果信息

软件缺陷报告

- 软件缺陷报告的组织结构
  - 例：手工软件缺陷报告

软件问题报告单

编号：_____

系统名称及版本			
模块名称及版本			
模块版本信息	（文件创建日期，文件大小，相关其他文件信息等）		
报告人		联系电话	
发生日期		提交日期	
问题类型：	程序□ 数据库□ 文档□ 其他□		
要求完成日期		实际完成日期	
问题描述/影响（问题发现过程与曾经处理过程）：			
问题背景描述： 如：系统使用方，工程启动时间，工程预期结束时间，工程当前状态（测试、试运行、还是已经投产）。系统的版本号，软件的来源方式等。			
问题严重性：	□很严重：严重影响系统验收，投产，或正常运行。 □严重：影响系统验收，投产，或正常运行。 □一般：影响操作。		
问题影响到下一步工作描述：			
受理人		受理时间	承诺解决时间
受理人意见：			
修改人		完成时间	
问题解决过程以及解决办法描述：			
问题解决结果并分析原因：			
报告人确认并签名：			

- 软件缺陷的度量
  - 软件缺陷度量的概念
    - 软件缺陷度量对软件开发过程中产生的缺陷数据采集和量化后统一管理，采用数学工具进行处理，分析缺陷密度和趋势等信息，有助于开发过程的改进和产品质量管理。
      - 软件缺陷度量本身并不能发现缺陷、修复缺陷。
    - 软件缺陷度量和软件测试过程密切相关，是软件质量度量的重要组成部分。
  - 软件缺陷度量的主要方法
    - 缺陷密度 (软件缺陷在规模上的分布)
    - 缺陷率 (软件缺陷在时间上的分布/缺陷密度分布)
    - 预期缺陷发现率
    - 整体缺陷清除率、阶段性缺陷清除率
    - 缺陷趋势等等

- 软件缺陷分析
  - 软件缺陷分析的概念
    - 软件缺陷分析将软件开发各个阶段产生的缺陷数据信息进行分类和汇总统计，计算分析指标，编写分析报告。
    - 软件缺陷分析利用一个符合项目要求的缺陷数据管理系统，采集完整的缺陷数据信息，进行缺陷数据分析，期望改进软件过程质量并实施缺陷预防措施。

- 软件缺陷分析
  - 软件缺陷分析的作用
    - 软件缺陷分析用于寻找、分析和处理缺陷的共性原因，期望实现缺陷预防。
      - 软件缺陷分析发现各种类型缺陷的发生概率，掌握缺陷集中区域，明确缺陷发展趋势，挖掘缺陷产生的根本原因，有针对性地提出缺陷预防措施，从而降低缺陷数量。
      - 缺陷预防是缺陷管理的核心任务之一，预防缺陷也是缺陷管理的最终目标。
      - 缺陷预防策略包括：
        - 测试活动尽量提前。尽早测试可以及时消除开发前期引入的缺陷，防止这些缺陷遗留并放大到后续环节。
        - 寻求缺陷根源。通过对已有缺陷进行分析，发现产生这些缺陷的技术上和流程上的不足，寻找相应改进方案，从根源上解决问题，预防类似缺陷再次出现。

- 软件缺陷分析
  - 软件缺陷分析的作用 (续)
    - 软件缺陷分析在软件可靠性评估中起着很大作用。
      - 缺陷分析用于评估当前软件的可靠性，并且预测软件产品可靠性的变化。
    - 缺陷分析报告中的统计数据及分析指标既是对当前软件质量状况的评估，也是判定软件是否能够按期发布或交付使用的重要依据。

- 软件缺陷分析
  - 软件缺陷分析步骤
    - 缺陷记录：记录缺陷不应该满足于记录缺陷的表面症状。
    - 缺陷分类：找出关键的缺陷类型，进一步分析其产生的根源，针对性地制定改进措施。
    - 缺陷预防分析：预防分析是整个缺陷分析过程的核心。
    - 编写缺陷分析报告，绘制缺陷分析图表。
      - 缺陷分析报告中的统计数据及分析指标既是对软件质量的权威评估，也是确定测试是否达到结束标准、判定测试是否已达到客户可接受状态和软件是否能够发布或交付使用的重要依据。
      - 缺陷分析图表提供了很多有价值的信息，比如分析开发和测试在人力资源的配比上是否恰当，某个严重的缺陷所造成的项目质量的波动等。

- 软件缺陷分析
  - 软件缺陷分析方法
    - ODC 缺陷分析 (参见 Chap 8.2 软件缺陷描述与分类-正交缺陷分析法)
    - *Gompertz* 成长模型
    - *Rayleigh* 分析
    - 根本原因分析
    - 缺陷注入分析
    - 四象限分析
    - DRE/DRM 分析

软件缺陷分析

软件缺陷分析方法 (续)

Gompertz 可靠性成长模型

- The standard *Gompertz* reliability growth model

$$R = ab^{c^T}$$

where $0 < a \leq 1$, $0 < b < 1$, $0 < c < 1$.

- T: time, launch number or stage number, $T \geq 0$.
- R: the system's reliability at T.
- a: the upper limit that the reliability approaches asymptotically as T, or the maximum reliability attained.
- ab: initial reliability at $T = 0$.
- c: the growth pattern indicator (small values of c indicate rapid early reliability growth and large values of c indicate slow one)
  - As in defect analysis, a is the expected total number of failures that would occur if testing was infinite. b is the rate at which the failures detection rate decreases.

软件缺陷分析

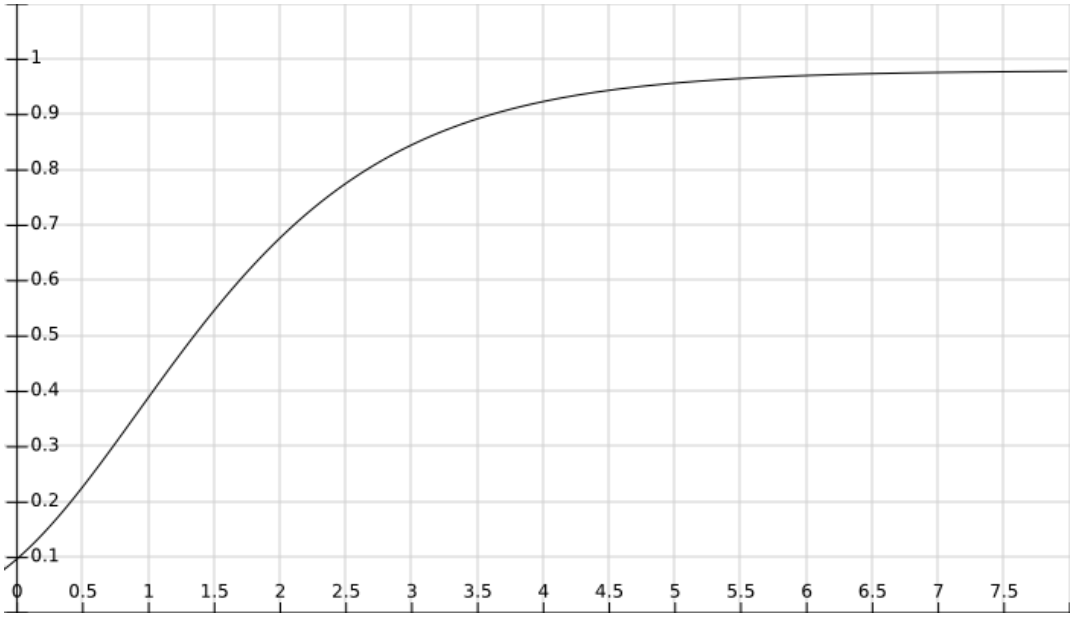
软件缺陷分析方法 (续)

Gompertz 可靠性成长模型

- The standard *Gompertz* reliability growth model

$$R = ab^{c^T}$$

where a = 0.98, b = 0.1, c = 0.4



y = 0.98*(0.1^(0.4^x))

软件缺陷分析

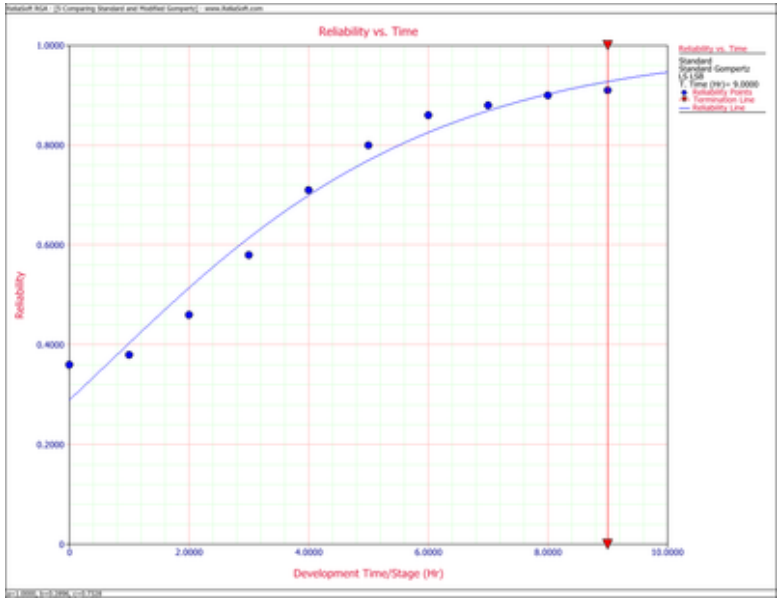
软件缺陷分析方法 (续)

Gompertz 可靠性成长模型

- The standard *Gompertz* reliability growth model

$$R = ab^{c^T}$$

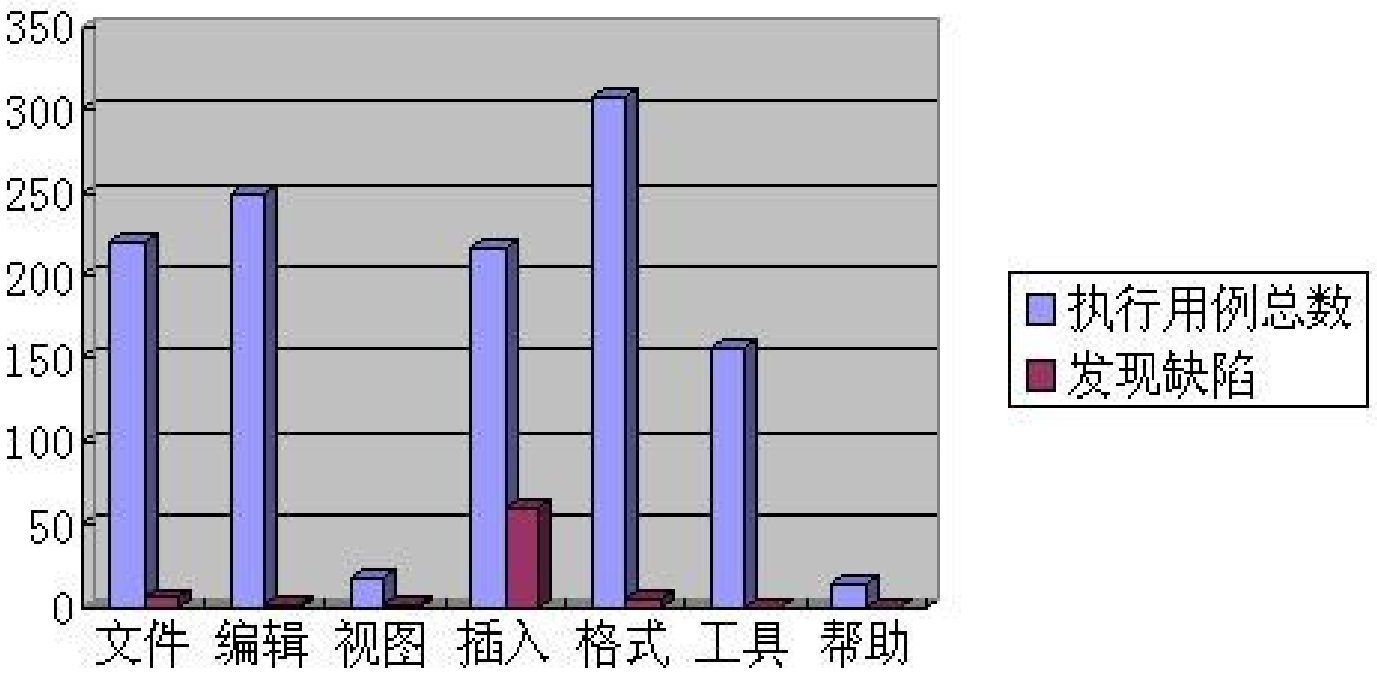
where $0 < a \leq 1$, $0 < b < 1$, $0 < c < 1$.



A matched *Gompertz* Growth Curve

- 软件缺陷分析
  - 缺陷数据统计
    - 缺陷数据统计是缺陷跟踪管理的目标之一。
      - 缺陷数据统计图表一般包括缺陷趋势图、缺陷分布图、缺陷及时处理情况统计表等。
      - 按照缺陷严重程度及类型，可以统计整个项目生命周期中所有缺陷的分布，也可以统计某一开发阶段的缺陷分布。
    - 缺陷数据统计是软件缺陷分析报告的重要内容之一。
      - 从统计的角度出发，可以对软件过程的缺陷进行度量。
        - 软件缺陷的统计度量指标如软件功能模块缺陷分布、缺陷严重程度分布、缺陷类型分布、缺陷率分布、缺陷密度分析、缺陷趋势分布、缺陷注入率/消除率等。
      - 统计的方式可以用表格记录，也可以用图表表示，如散点图、趋势图、因果图、直方图、条形图、排列图等。

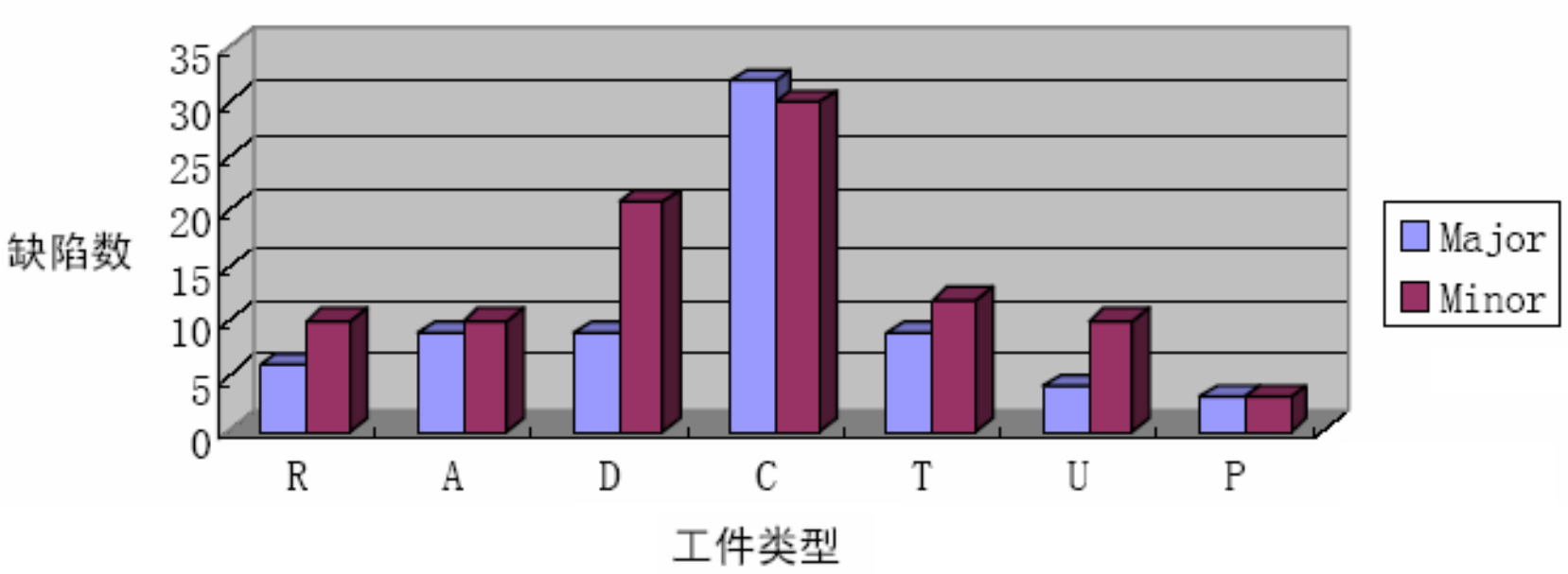
- 软件缺陷分析
  - 缺陷数据统计
    - 例：软件功能模块的缺陷统计分布图。



软件缺陷分析

缺陷数据统计

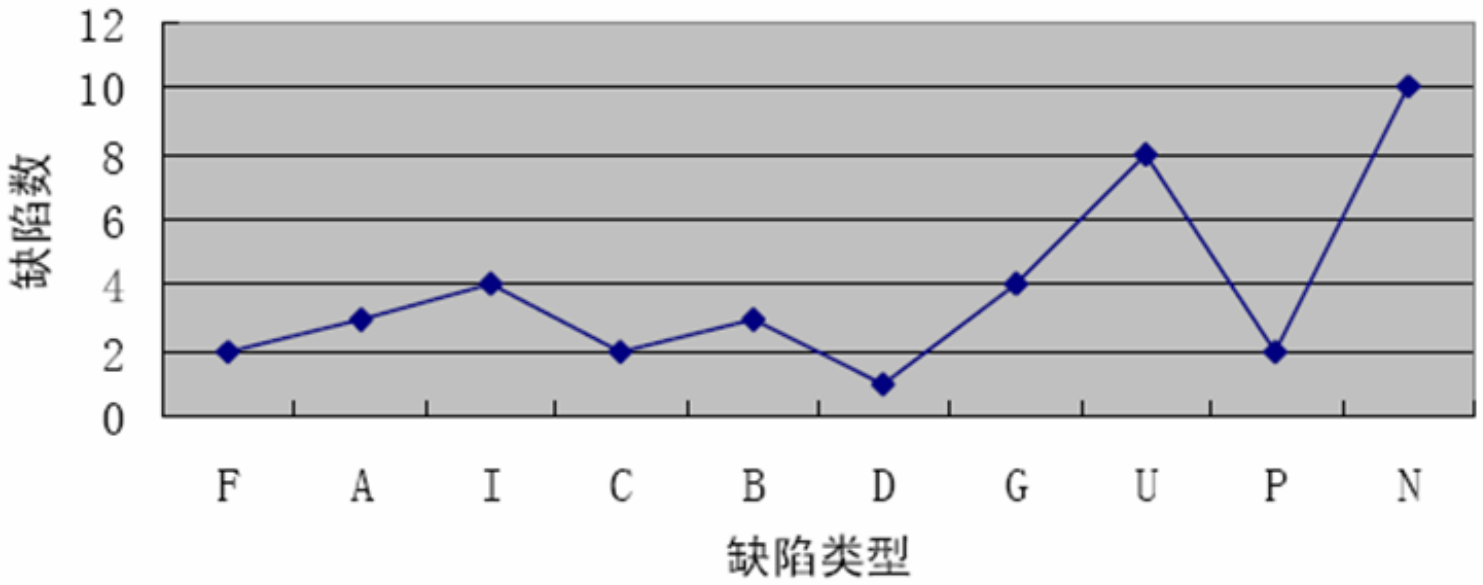
例：软件缺陷起源分类统计分布图。



参见“Chap. 8.2 软件缺陷描述与分类-软件缺陷的分类-软件缺陷分类方法的应用-按缺陷起源 (Origin) 的分类”

软件缺陷分析

- 缺陷数据统计
  - 例：XXX 评审缺陷类型统计分布图 。

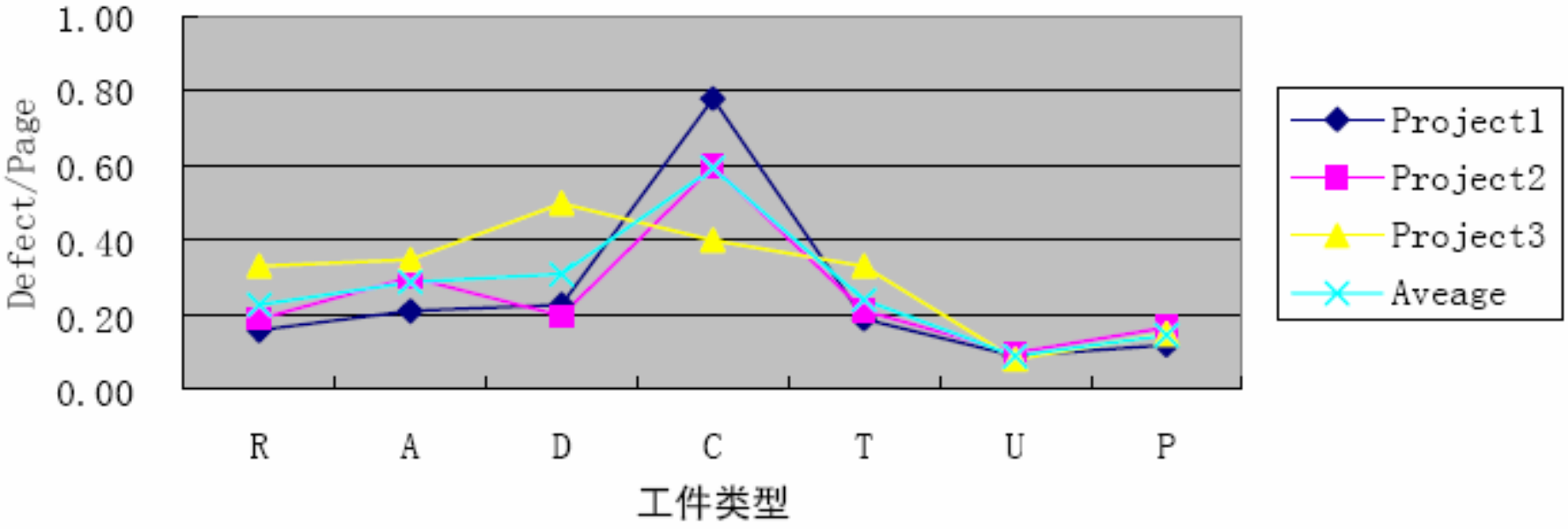


  - 参见“Chap. 8.2 软件缺陷描述与分类-软件缺陷的分类-软件缺陷分类方法的应用-按缺陷类型标准的分类”

软件缺陷分析

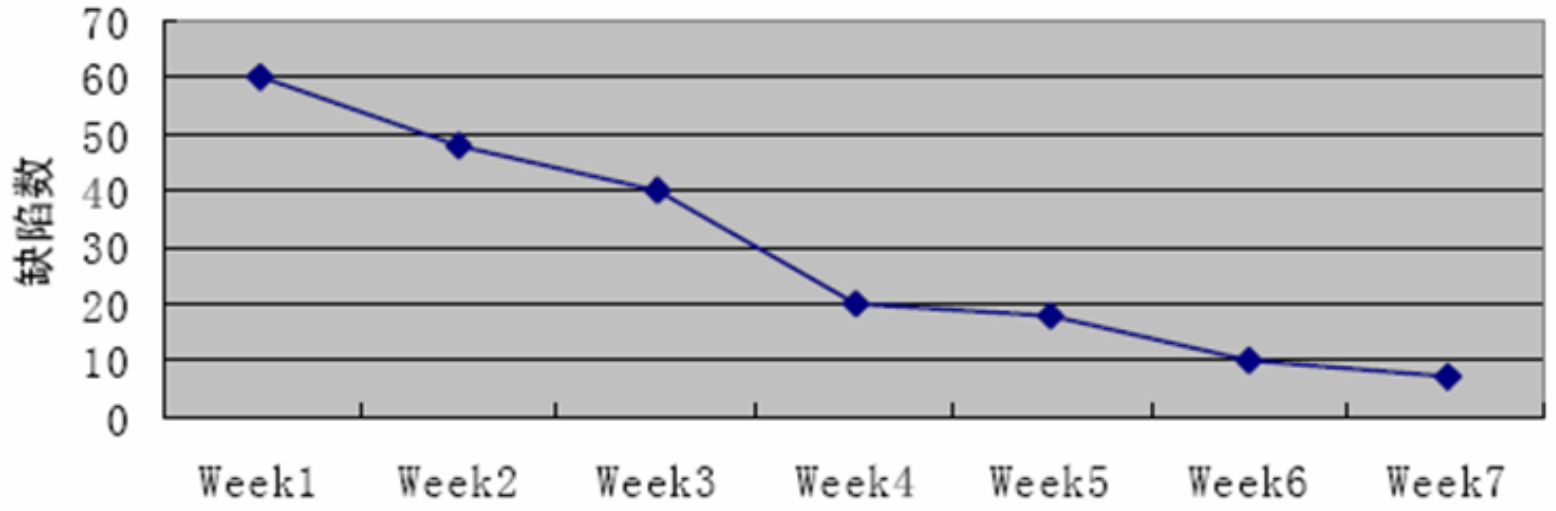
缺陷数据统计

- 例：多项目静态评审缺陷统计分布图 (项目缺陷率)。



- 参见“Chap. 8.2 软件缺陷描述与分类-软件缺陷的分类-软件缺陷分类方法的应用-按缺陷起源 (Origin) 的分类”

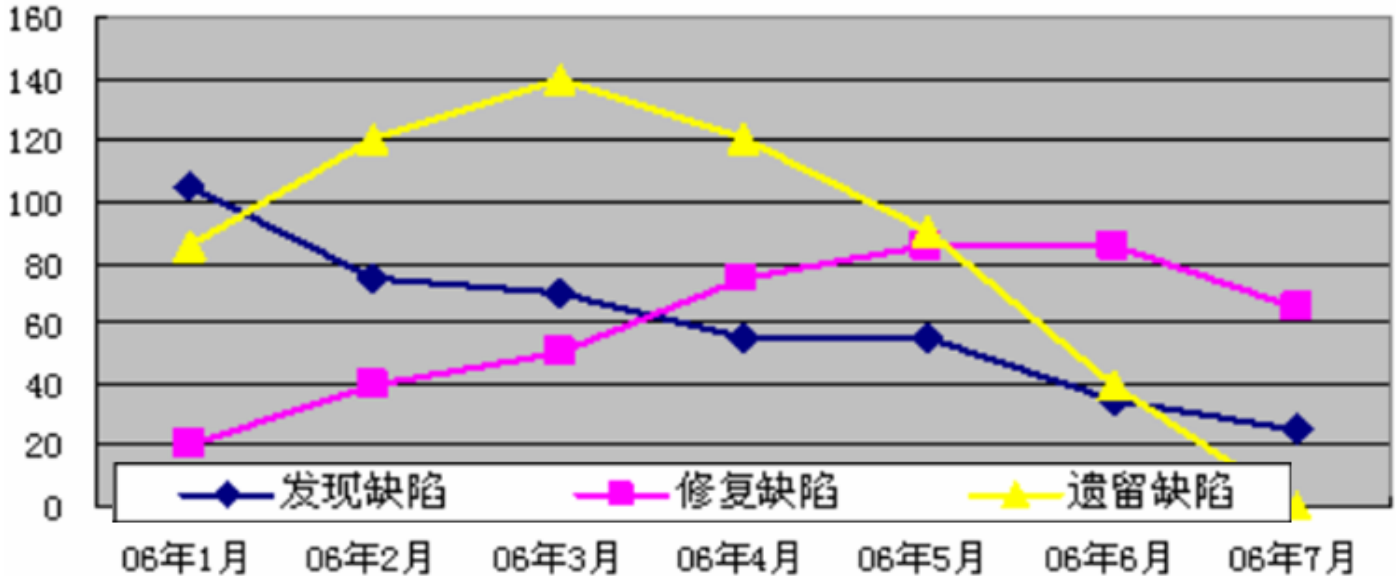
- 软件缺陷分析
  - 缺陷数据统计
    - 例：软件缺陷发展趋势图。



软件缺陷分析

缺陷数据统计

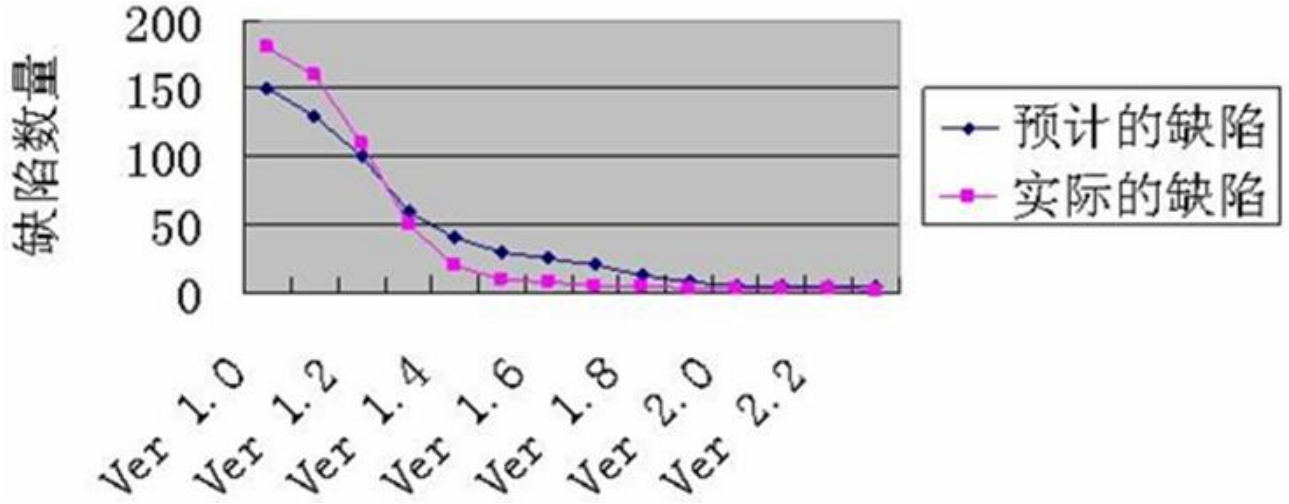
例：软件缺陷发现、修复、收敛趋势图。



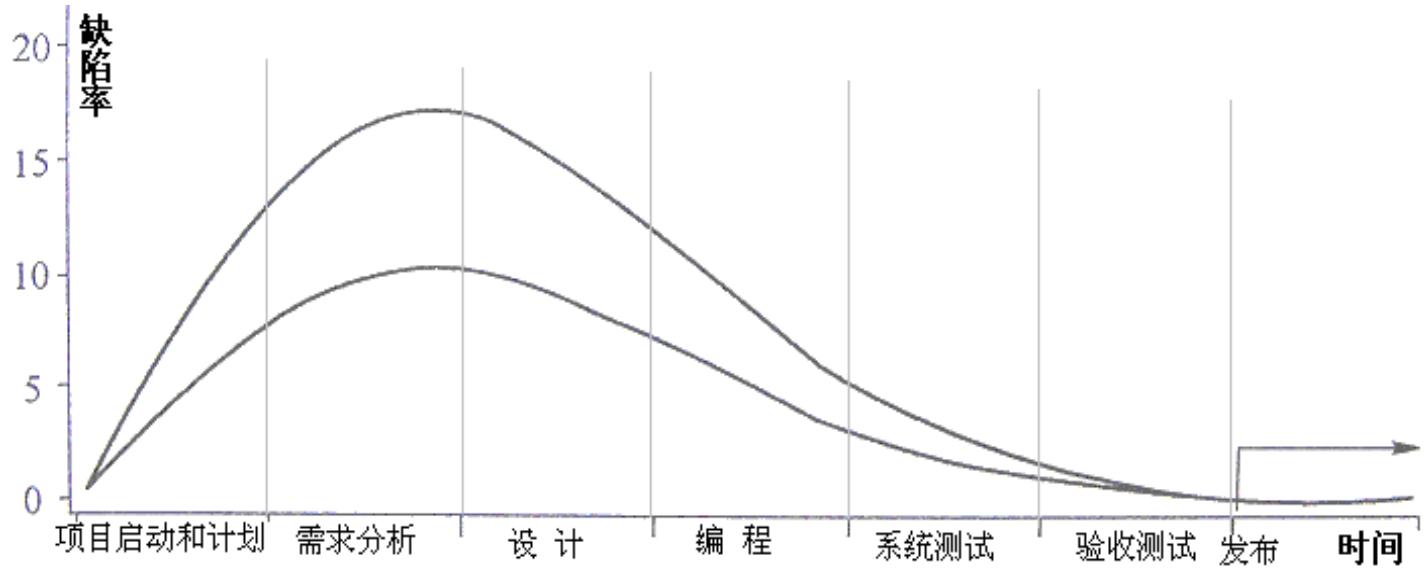
迭代公式：R = R + F – M

R: 遗留数，F: 发现数，M: 修复数

- 软件缺陷分析
  - 缺陷数据统计
    - 例：基于软件版本的缺陷趋势图。



- 软件缺陷分析
  - *Rayleigh* 软件缺陷模型 (*Putman* 1978, *Gaffney* 1984)



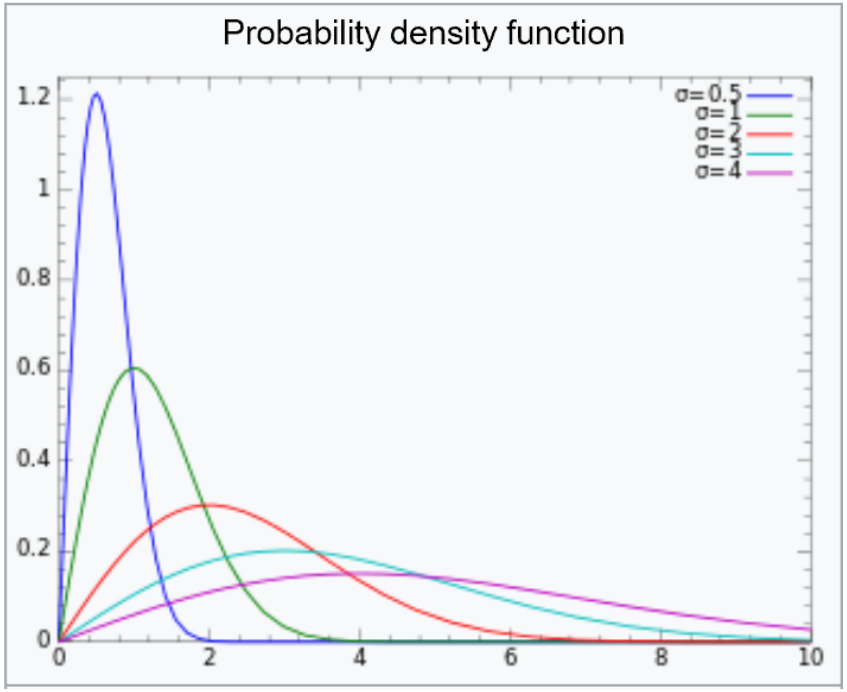
软件缺陷分析

Rayleigh 软件缺陷模型

- The probability density function (PDF) of the *Rayleigh* distribution

$$f(x;\sigma) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, x \geq 0.$$

where σ is the *scale parameter* of the distribution.



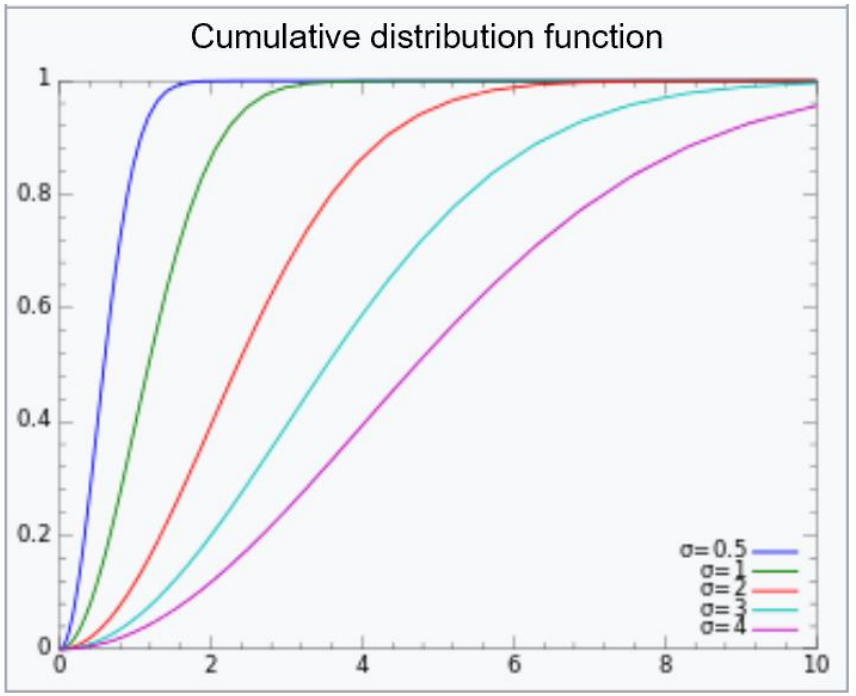
软件缺陷分析

Rayleigh 软件缺陷模型

- The cumulative distribution function (CDF) of the *Rayleigh* distribution

$$F(x;\sigma) = \int_0^x f(x;\sigma)dx = 1 - e^{-\frac{x^2}{2\sigma^2}}, for\ x \geq 0.$$

- We found that
  - $\lim_{x\to\infty} F(x;\sigma) = 1.$
  - $F(\sigma;\sigma) = 1 - e^{-\frac{x^2}{2\sigma^2}} = 0.4.$



软件缺陷分析

Rayleigh 软件缺陷模型

- *Rayleigh* 模型可用于预测软件开发全生命周期的缺陷分布，是一种常用的可靠性模型。1982年 IBM 的 *Gaffney* 报告指出，软件质量评估的缺陷计数应该基于 IBM 定义的开发过程的6个阶段：概要设计、详细设计、编码、单元测试、集成测试和系统测试。上述6个阶段所发现的缺陷率，随着这些阶段在软件生命周期时间的分布符合 *Rayleigh* 分布。
- *Rayleigh* 分布的 PDF 曲线 $f(x)$ 先快速上升到峰值，然后以减速率下降，σ是曲线到达最大值的时间。如果在开发前期排除的缺陷较多，σ 值将向左移动，那么在后期的测试和维护阶段的缺陷率就会较低，从而降低缺陷修复的总体成本。
- *Rayleigh* 分布的 CDF $F(x; \sigma)$ 中令 $x = \sigma$，得到 $F(\sigma, \sigma) = 0.4$，即当 $f(x)$ 达到最大值时，已经发现的缺陷数约为缺陷总数的 40%。

软件缺陷分析

Rayleigh 软件缺陷模型

例：一个项目的实测缺陷数目分布如下表。

测试时间 x	1	2	3	4	5	6	7	8	9	10	11	12	13
发现缺陷数	20	38	55	52	41	22	10	5	4	2	2	2	1

- 分析上述表格，发现 x = 3 时缺陷数达到峰值 55。构造σ=3 的 *Rayleigh* 分布，

$$f(x;\sigma=3)=\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}=\frac{x}{9}e^{-\frac{x^2}{18}},x\geq 0.$$

- 估计缺陷总数：

$$K = [f(1) + f(2) + f(3)]/0.4 = [20+38+55]/0.4 \approx 282.$$

- 实际发现缺陷数 = 254.

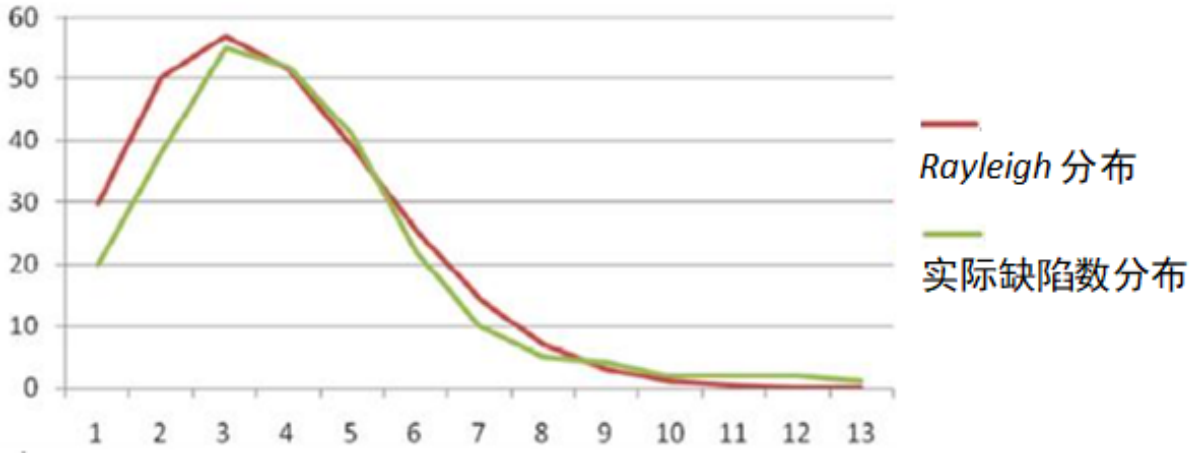
软件缺陷分析

Rayleigh 软件缺陷模型

- 项目的实测缺陷数分布和拟合的 *Rayleigh* PDF 分布如下表。

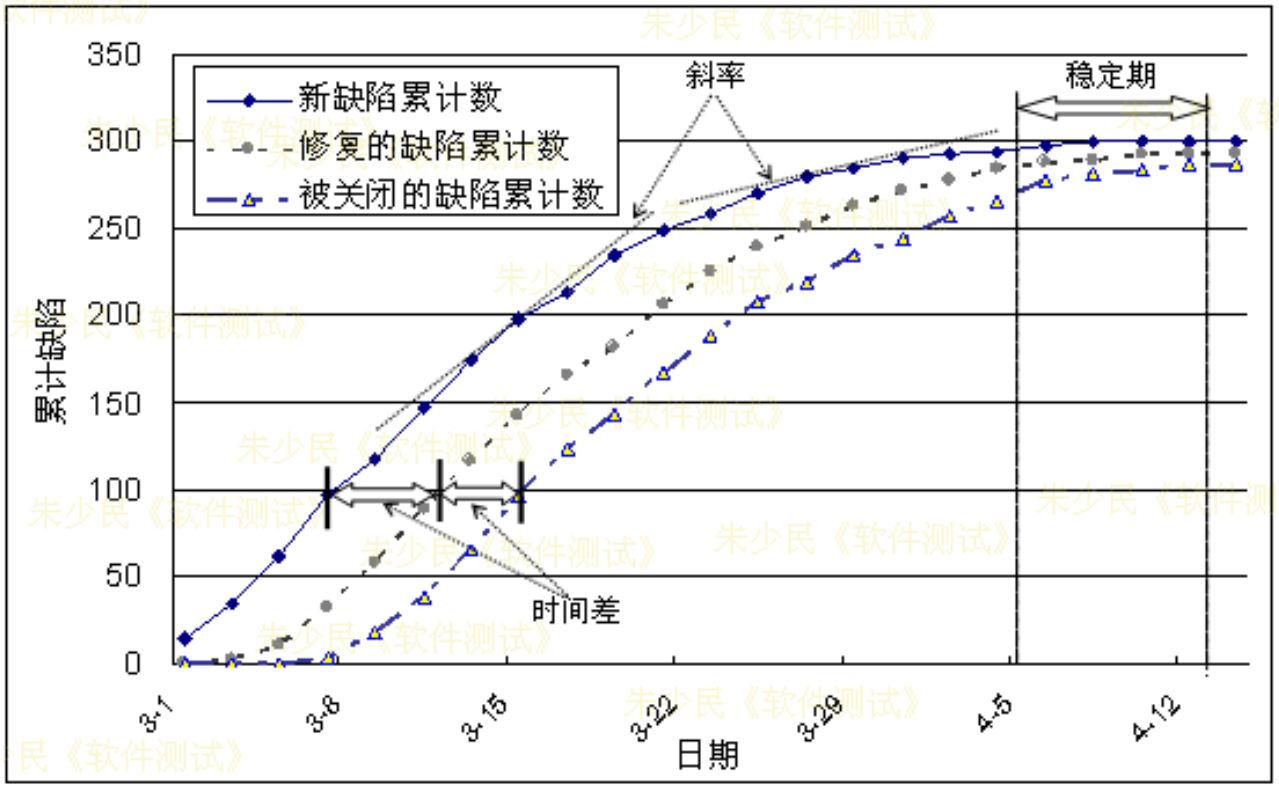
测试时间 x	1	2	3	4	5	6	7	8	9	10	11	12	13
发现缺陷数	20	38	55	52	41	22	10	5	4	2	2	2	1
$K \cdot f(x;\ \sigma=3)$	26.6	50.2	57.0	51.5	39.0	25.4	14.4	7.2	3.1	1.2	0.4	0.1	0.03

- 项目的实测缺陷数分布和 *Rayleigh* 分布模拟值的比较。

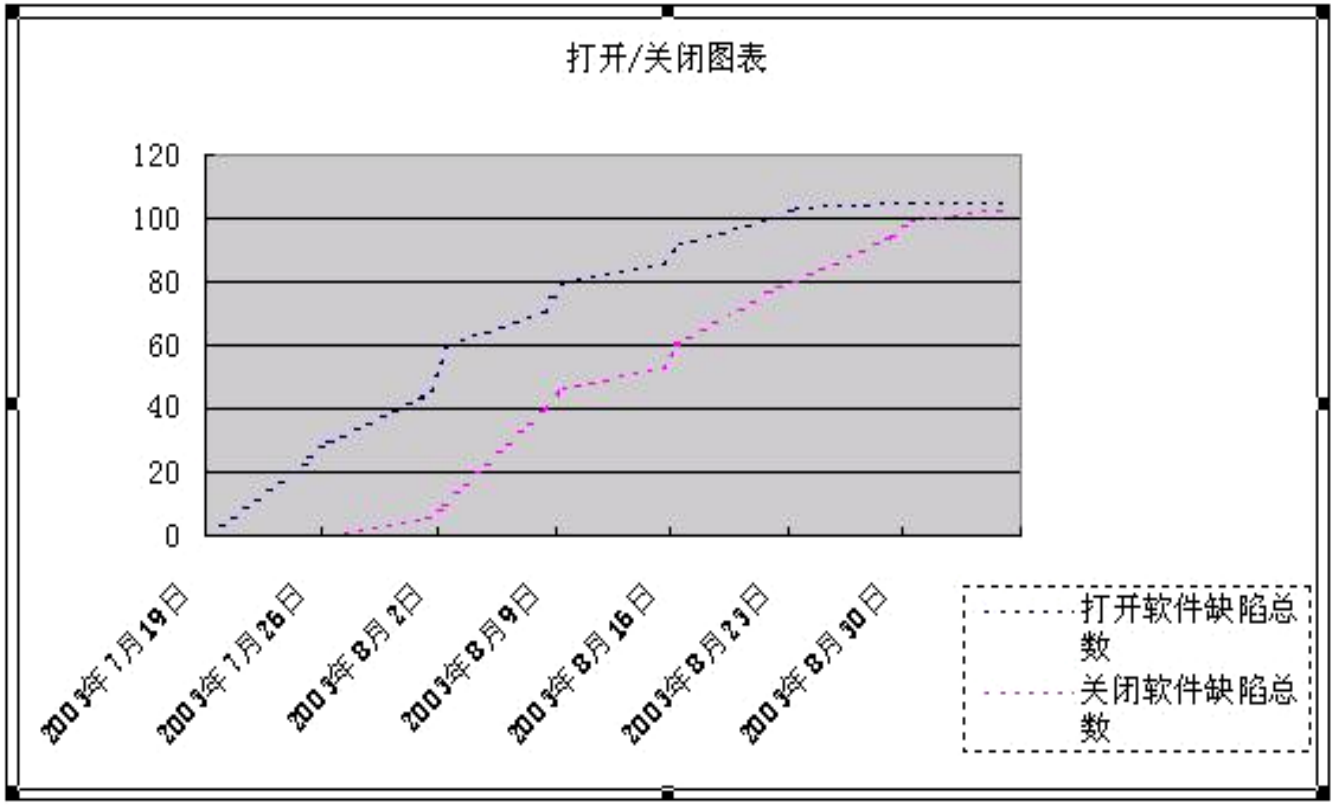


软件缺陷分析

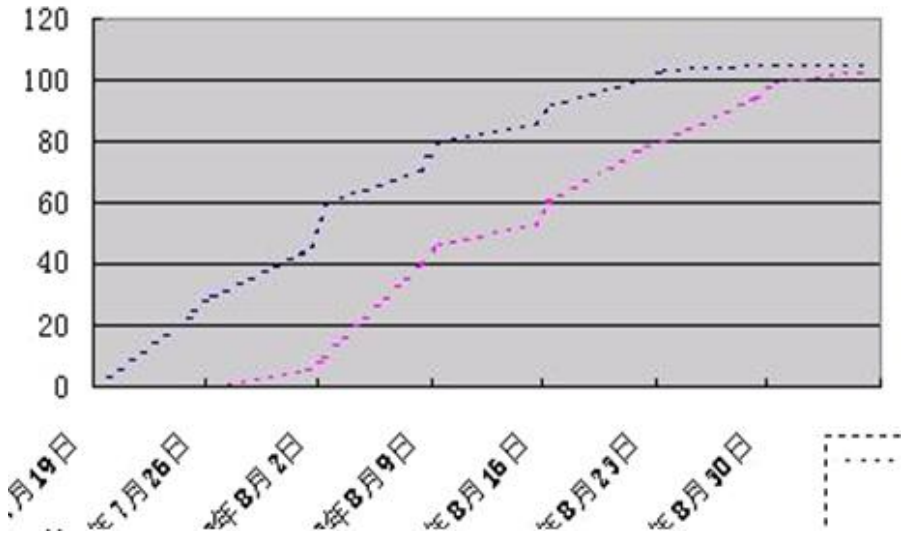
- 缺陷跟踪的方法与图表
  - 例：累积趋势分析。累积数据是将前面产生的数据不断累加起来所构成的时间序列；累积曲线具有更明显的趋势特征。



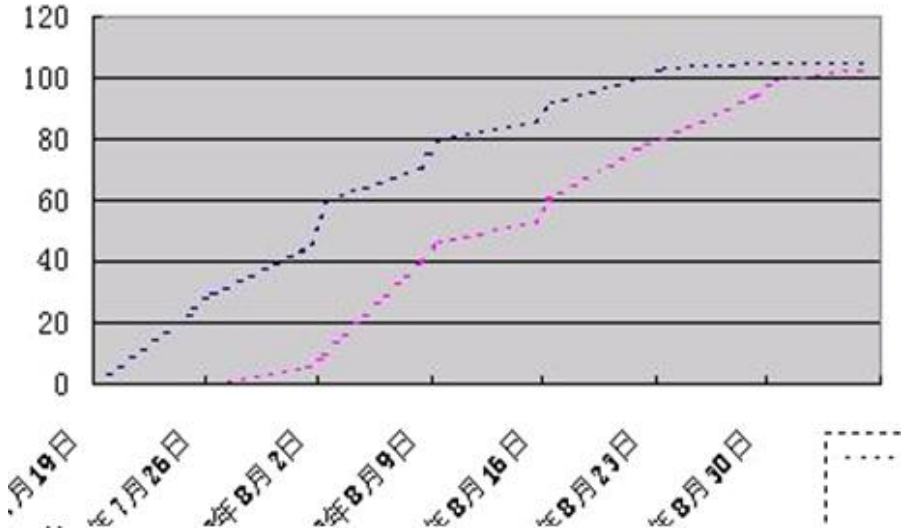
- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析。



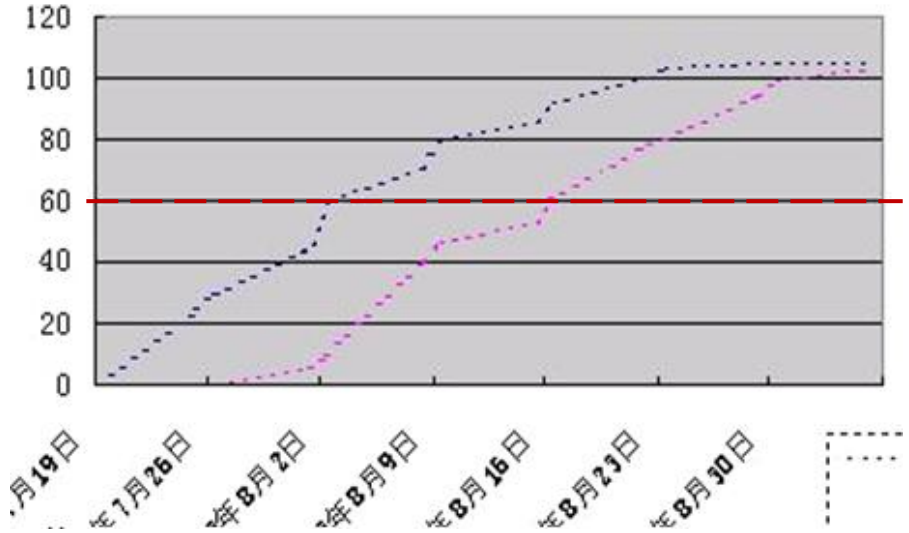
- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析 (续)
      - 项目目前的质量情况取决于累积打开曲线和累积关闭曲线的趋势；项目目前的进度取决于累积关闭曲线和累积打开曲线起点的时间差。



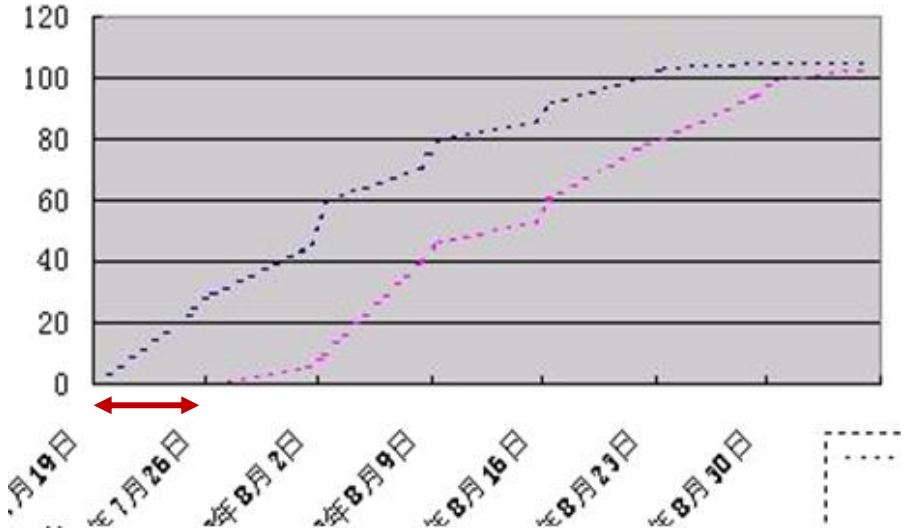
- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析 (续)
      - 开发人员是否完成软件缺陷修复：累积关闭曲线是否快速上升。
      - 测试人员是否积极验证软件缺陷：累积关闭曲线是否紧跟在累积打开曲线后面。



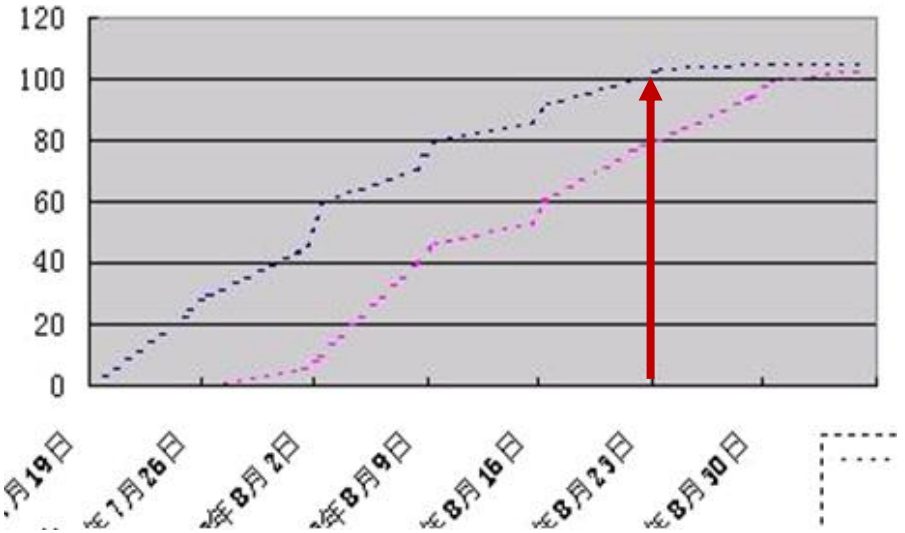
- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析 (续)
      - 当累积的打开曲线 (如图的顶部曲线) 在一条渐近线限制下稳定下来，通常可以认为该测试已经完成。



- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析 (续)
      - 打开日期在关闭日期之前，可以看到关闭曲线大约落后了一个星期。这种滞后起源于件缺陷被引入到产品并将该产品发送到测试小组，以及测试配置和回归测试所引起的延迟。这种延迟集中到测试的最后一天。



- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析 (续)
      - 在当前测试阶段找到软件缺陷的能力在减弱。发现软件缺陷的极限在8月23号左右；接下来系统测试第二个周期发现少数几个软件缺陷，在最后的周期中没有发现缺陷。

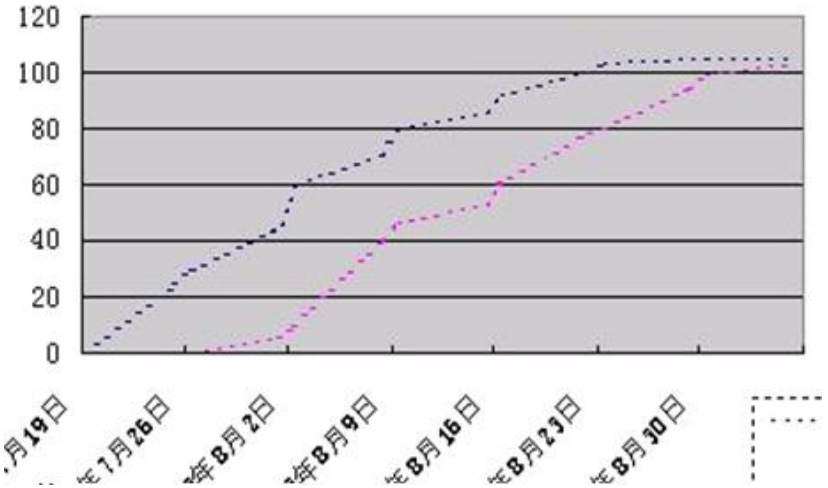


软件缺陷分析

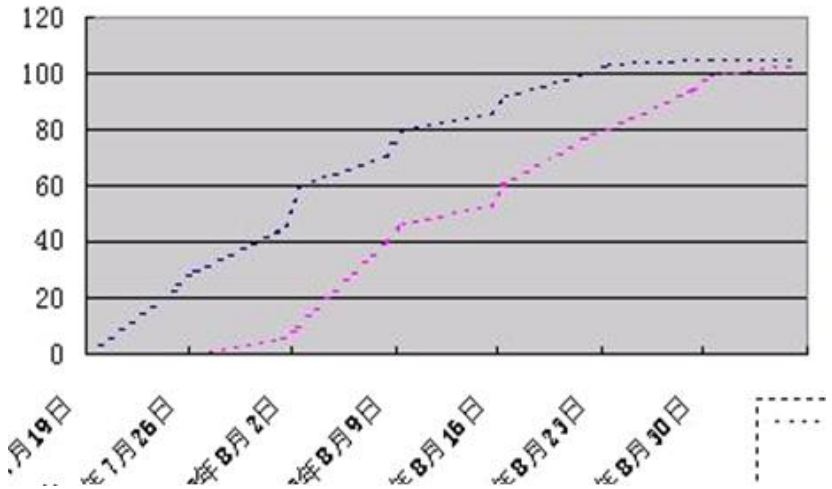
缺陷跟踪的方法与图表

例：软件缺陷打开/关闭累积图分析 (续)

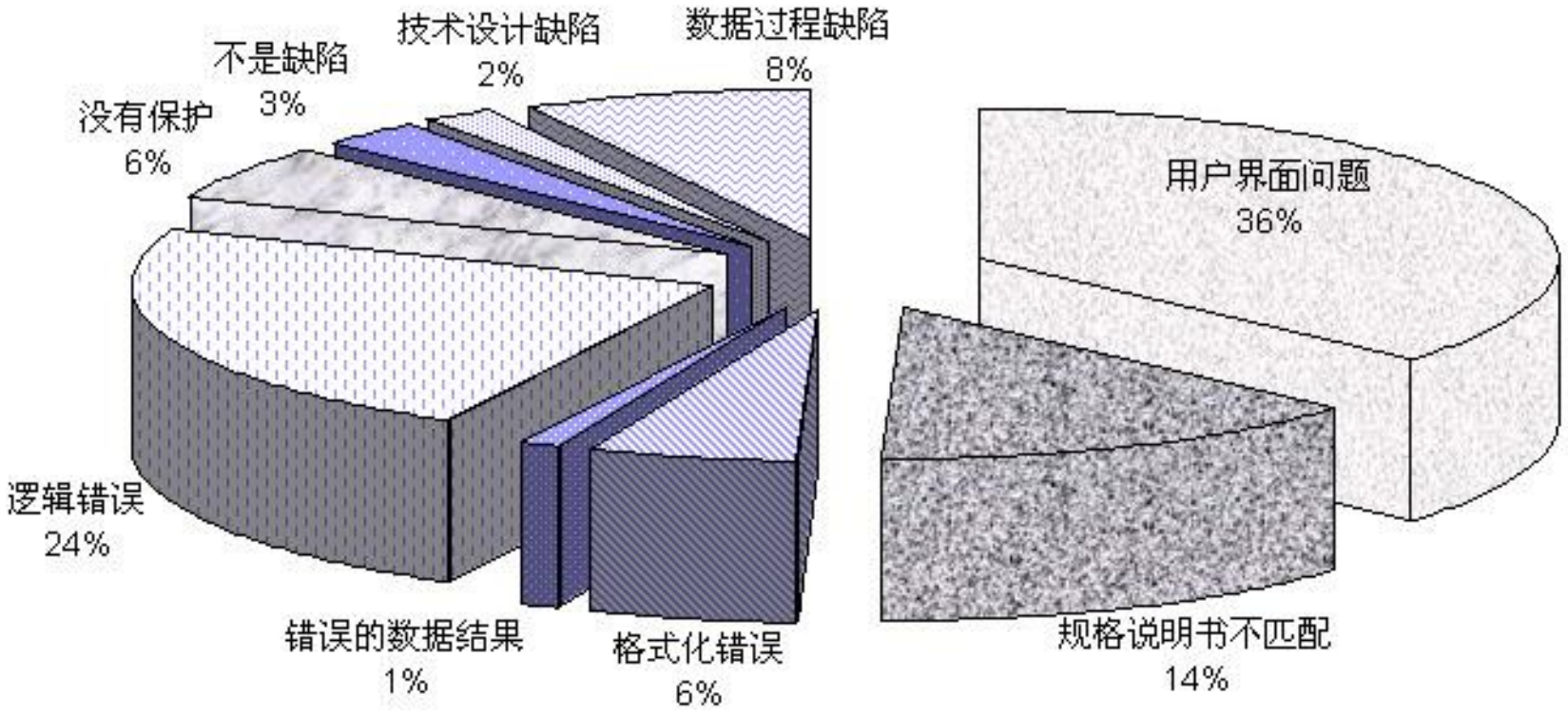
- 开发人员是否完成软件缺陷修复：在测试和修复的过程中，发现这两条曲线在不断的收敛，当这两条曲线收敛成一个点时，开发人员基本上完成了修复软件缺陷的任务。并且注意到关闭曲线紧跟在打开曲线的后面，这表明项目小组正在快速地推进问题的解决。



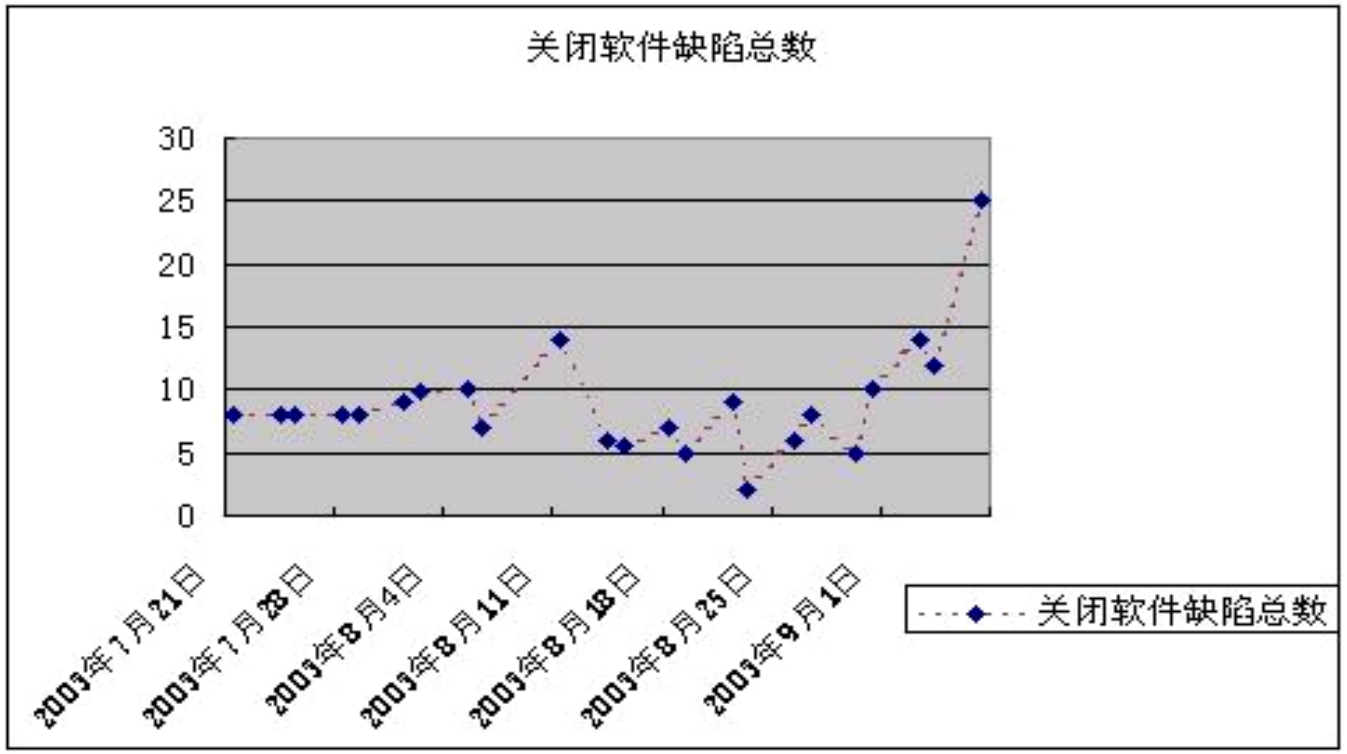
- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷打开/关闭累积图分析 (续)
      - 当测试人员从一个测试阶段过渡到另一个测试阶段时，发现累积打开曲线有一个凸起，这样的凸起必须引起充分的注意，说明开发人员修复软件缺陷引入了新的缺陷或者有些软件缺陷被遗漏到下一个阶段才被发现。项目管理人员需要召开紧急会议分析当前项目情况，找到解决办法。



- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：软件缺陷根源图表。



- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：关闭软件缺陷周期图表。



- 软件缺陷分析
  - 缺陷跟踪的方法与图表
    - 例：关闭软件缺陷周期图表 (续)
      - “关闭周期”的直观意义：关闭周期将开发人员对软件缺陷的响应量化到软件缺陷报告中，一个稳定的关闭周期图表显示了从一天到另一天相对较少的变化。如果软件缺陷打开的日期被推迟，日常关闭曲线将被拉向0。此外，一个可接受的日常关闭曲线不向任何边界明显倾斜。
      - 一个稳定而可接受的关闭周期图表指出了一个理解良好、运行平稳的缺陷管理过程，理想情况是一个向下或水平趋向的关闭周期曲线。

软件缺陷管理工具

- 商用工具
  - 国外工具
    - TrackRecord，Compuware
    - ClearQuese，IBM Rational
  - 国产工具
    - QAMonitor，北航
- 共享软件
  - BugRat in the Giant Java Tree:

    http://www.gjt.org/pkg/bugrat
  - Bugzilla, Buggit, Mantis 等

Lecture 24. Software Defect Management (2)

End of Lecture