

# Chapter 1 软件的本质The Nature of Software

---

## 1.1 软件的本质

---

随笔

- high-level design (概要设计/架构设计) ——1. 软件的体系; 2. 数据库/数据设计; 3.接口设计  
(意义: 概要文档对后续的维护有重要作用, 因为它就是一个使用指南, 可以进行错误定位)
- 软件既是一种产品, 也是交付产品的一个载体 (产品和产品交付载体的双重角色)
- error 交付之前——>failure 交付之后运行的结果不符合顾客的需求, failure可能由一个或多个error导致
- 两种error错误检查的技术: **review technology** 评审技术; **testing technology** 测试技术
- 软件开发的问题:
  - 开发时间(take so long)
  - 开发成本(cost so high)
  - 开发错误检查(find all errors)
  - 维护时间投入(spend time maintaining)
  - 开发进度评估测量(measure progress)

### 1.1.1 定义软件

(要理解)

软件是什么:

1. 完成软件的特性, 功能, 性能等所需要的 **计算机程序, 指令代码——程序**
  - feature特性: *非功能需求* non-functional: 如安全性, 兼容性, 可移植性, 可扩展性等
  - function功能: 与非功能需求相对应
  - performance: feature的一部分, 但因为很重要所以单拉出来
2. 使得程序可以充分利用操纵信息所使用的 **数据结构——数据**
3. 描述程序操作使用的硬拷贝和虚拟形式的 **描述性信息——文档**

**软件与硬件的生命周期刻画曲线图:**

1. 硬件的生命周期

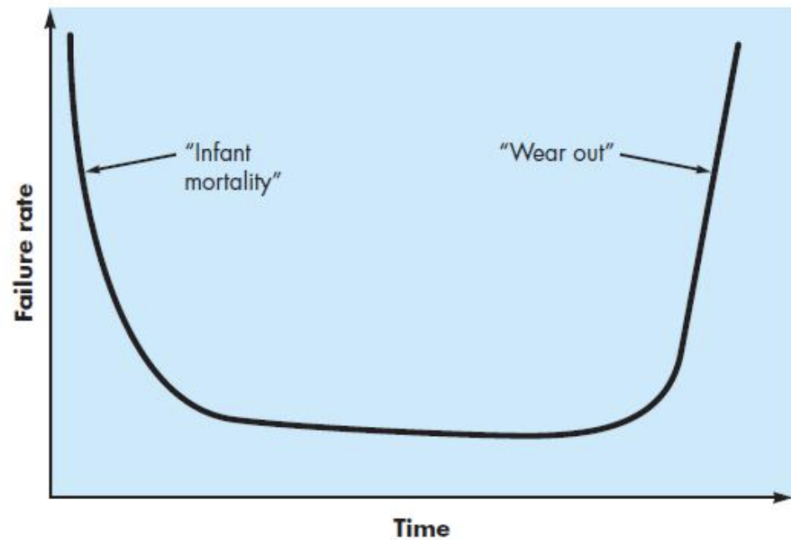
解读: 浴缸曲线 (早期的infant mortality是由于设计引起问题, 当设计问题解决后可以平稳运行, 直到硬件损耗, 期间替换硬件不影响failure rate) (**早期故障多, 后期磨损**)

FIGURE 1.1

Failure curve  
for hardware



If you want to reduce  
software deterioration,  
you'll have to do  
better software design  
(Chapters 12 to 18).



## 2. 软件的生命周期

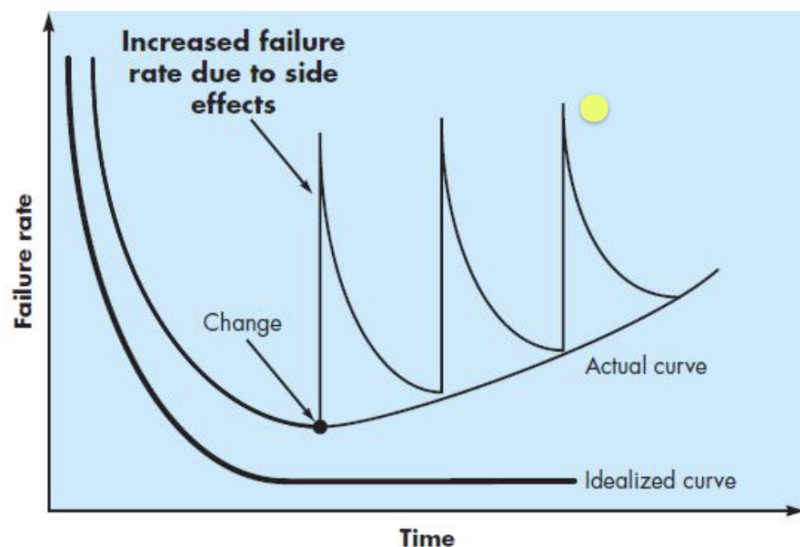
解读：理想曲线——开发后可以一致运行；

实际曲线——考虑：1. 功能增强的update；2. 功能修改和错误修复modify & bug fixed

不断改变后failure rate的突然上升与总体上升：修改change后导致其他新的问题出现，有"传染性"错误的可扩展性，这是功能调整增强的代价（错误的扩展性，修复某问题的副作用）

FIGURE 1.2

Failure curves  
for software



### ※DevOps: 现在开发常用的开发框架

development in Operations: 软件开发人员(Dev)和订正人员(Ops)之间的沟通合作

### ※微服务：

将整个Web应用组织为一系列小的Web Service，这些小的Web Service可以独立的编译部署，并通过各自的API接口相互通信，彼此相互协作，作为一个整体为用户提供服务，但可以独立进行扩展。

优点：一个微服务出错不会影响其他

## 1.1.2 软件应用领域

(软件的领域划分)

- System Software系统软件：操作系统软件（应用软件运行在这之上）——算法核心
- Application Software应用软件：解决特定需求的独立应用程序。
- Engineering/scientific Software工程/科学软件：例如matlab
- Embedded Software嵌入式软件：例如家电内部的，汽车内部的——操作系统特殊，硬件交互性强，错误较难以定义
- Product-line Software产品线软件：为不同用户提供特定功能
- Web/mobile App: browser-based
- AI Software人工智能软件：机器人，博弈，模式识别（语音图像），自动驾驶，自然元素处理（文本）

## 1.1.3 遗留软件Legacy Software

(重要)

特点：年龄大(太老了)，业务关键性（数据难以移植，风险大，很少重新开发）

描述：**1.老旧软件； 2.不断变更以适配需求——臃肿； 3.质量差，代码可读性差，文档混乱**

※遗留系统演化原因：(考试会根据这些设计情景，比如银行系统)

- A需要进行适应性调整，满足新的计算环境和计算需求（比如：密码升级到指纹） **new tech adapted**
- U升级以实现商业需求 **enhanced implementation**
- E扩展使得可以与更多的系统交互，如现代系统或数据库 **extend**
- A架构需要重新部署来适应新环境（例如下面以前没有mobile，或部署微服务框架等） **re-architected**

## 1.2 软件的演变

四大类软件正在发展成主流：

※练习：Web技术的发展

标红的部分

- **网页应用WebApp**

by looking up information, to learn the history of web development.(web1.0~web3.0)

- **移动设备Mobile Application**

在大多数情况下，移动应用程序包含用户界面、与基于 Web 的资源的互操作性和本地处理能力(对本地的存储和信息进行计算处理)，还可提供长久的存储能力；移动网络应用程序和移动应用程序之间存在细微的区别：前者浏览器，后者可以直接获取设备硬件信息提供本地处理与存储能力

在大多数例子中，移动应用建立一个user interface，以及和网页资源的可交互性(interoperability)和本地处理能力

- **云计算Cloud Computing 考试看看**

云平台:IaaS层（资源层）-虚拟机（跨IaaS与PaaS）->PaaS层（平台层：资源监测；预警；优化决策）-->SaaS层(软件，可视化界面)

- **产品线软件**

(简单看看即可)

练习:

- Web1.0-3.0
- DevOps
- 微服务
- SOA
- 云计算