



软件工程

Software Engineering

主讲教师：史清华

Instructor: Shi Qinghua

E-mail : sqh2006@aliyun.com

Phone: 13685317987

Software Engineering

- 要想成长为一名高级软件工程师需注重哪些方面？
 - 能否成长为一名高级软件工程师往往与所处的行业、所从事的岗位以及自主学习有密切的关系。高级软件工程师不仅要有较强的技术研发能力，同时也要有较强的行业认知能力，能够针对于不同的应用场景给出针对性的解决方案，而且要能紧跟技术发展趋势不断完善技术方案。
 - 高级软件工程师的下一步可能出任项目经理（暂且不说）
 - 对于一名初级软件开发人员来说，要想成长为一名高级软件工程师，可以按照以下路线发展：

Software Engineering

第一：根据自身的知识结构和能力特点选择主攻方向。在当前的IT产业生态中，细分领域越来越多，要想取得快速的成长和突破，一定要选择一个适合自己的主攻方向，从目前的行业发展趋势来看，大数据、物联网、边缘计算、云计算、人工智能等方向都是不错的选择。

第二：不断完成岗位升级。岗位能力升级是成长为高级软件工程师的重要依托，岗位升级的过程也是资源整合能力提升的过程，这对于软件工程师的全方面成长具有重要的意义。如果在一个岗位上长时间不能过获得突破，一定要积极寻找解决办法，比如不少知识结构欠缺比较多的初级工程师会选择通过读研、进修深造等预案来完成岗位升级。（入职百度测试开始）

第三：注重行业经验的积累。软件工程师一定要有一个较为全面的知识结构，而这个知识结构的重要组成部分就是行业知识体系，行业知识对于未来工程师的职业生命周期会有重要的影响。积累行业知识的同时也要有针对性的解决方案，不同的解决方案要面对不同的生产场景。

Software Engineering

- 软件工程课程涵盖的范围
 - 项目管理层面的内容
 - 技术研发层面的内容
 - 软件工程学范畴内相关知识点
- 软件工程课程要求
 - “底线”要求：初步了解软件工程的实际操作流程，知道较复杂工程问题的大致求解过程。
 - 通过课程设计的收获得失，促使学生进行初步的专业自我评价。

Software Engineering

中国还未成为软件技术创新大国

-----包括软件工程学领域

- 什么叫创新？有三种类型：思想创新、科学创新、技术创新。思想创新是创新的源泉（基础），科学创新是思想创新的扩展（方法）、技术创新属思想创新的务实操作（工艺）。
- 软件 engineering 创新涵盖上述创新的三个方面。
- 自主创新属技术创新，有三个方面：一是拥有自主知识产权，二是集成技术，三是引进吸收消化再创新。这是软件技术和软件 engineering 现阶段所力求达到的目标。
- 中国要进入下一个阶段----从零星的杰出成就（查毒、排版、即时通讯软件、电商平台、互联网应用等）到广泛的创新体系，所面临的一个重大挑战就是必须将更多力量从单纯硬件转至软件(思想、方法、规范等)领域。

Software Engineering

- 其实中国的现状（媒体宣传）
 - 硬件：曾把组装当制造。软件：误把应用当创新。
 - 软件工程范畴之下的基本支持技术就很薄弱
 - 例如：软件开发支撑环境及技术。
 - 近期：美国针对华为的**EDA**技术封锁。

-----中国现在最大的工业软件公司是华为技术。即使华为如此强大，设计产品时也要用三家美国公司**Synopsys**、**Cadence**、**Mentor**提供的**EDA**(电子设计自动化)工具。芯片设计极其复杂，里面几十亿的晶体管，**EDA**工具的极限设计精度是无可替代的。工业软件绝不是一般的IT软件，更不是一般互联网公司可以涉足的。仅就代码行数而言，世界上最大的软件公司不是微软，不是 **SAP**，而是全球最大军火商洛克希德·马丁公司。去掉制造业的表皮，剩下的核心技术就是工业软件，包括编程、数据建模分析、设备驱动、程序更改、传感器应用等全要依赖软件技术。

Software Engineering

- 但是言及软件工程概念而言：
 - 软件工程的基本支持技术的设计者就是根据自身对软件工程发展的预见，而设计了各种支撑系统、开发环境、软件工具、分析及驱动程序、开发模式等等。
 - 近期软件工稄研发成果：鸿蒙系统研发，相关生态系统。（所用内核是开放的、可买断的、无需从头再来的。）
- 总之：软件工程专家试图做到----能够采用高质量软件环境及工具，使软件能够按照某种能够反映软件开发规律的**规范/模式**来开发。

Software Engineering

- 那么，SE有现成思想方法或规范可遵循吗？

-----PSP/TSPi/CMMI(重量级的)

-----RUP(中等量级的)

-----Agile/XP(轻量级的)

工艺层面

技术工艺上可
保证先进性

问题：软件成熟度为CMMI 1级或2级的某些国内企业开发出的软件是真正好用的，而有的企业即使通过了CMM 5级认证，也无法保证它交付好的软件（一个可度量的软件过程不一定就是一个好的成功的软件过程）（同样规范面前人是活的，有质量差异，这就是软件工程的复杂性）

难受：软件工程现有规范背后的思想及方法是非文化派生

出路：遵守规范和培养思想并重. (top to down strategy)

- SE教育：始于思想熏陶(希望长出来的树苗一开始就是正的而不是歪的) (这也是选择英文教材的原因之一，有时候不经意的一小段话源自深刻经验，可能给人很大启发)

Software Engineering

中国和西方的现代思想观念的差异

- 西方人注重规范。
 - ❖ 例如：无论你走进全中国任何一家“肯德基”快餐店，汉堡包的味道都是一样的。
 - ❖ 东方人的传统最重视的还是人。例如：若您换一家酒店去吃饭的话，鱼香肉丝的味道可能就变了，因为厨师不是一个人！
- 起源于西方的现代科学，其精神就是讲究量化与规则
 - ❖ 从骨子里就认可规范的存在
 - ❖ 整个社会体系都认可各种规范/法规的权威，所有公民都必须遵守！（例如：发达国家个人信用一旦缺失会有大麻烦。）

Software Engineering

- 在西方不尊重规范的代价：有个德国人因为有急事，不得已驾车闯了红灯；于是，警察上门来催交了高额罚款；银行通知他的信用等级被降低了，太太回来说他们的银行卡不再能赊帐，有些消费除非有预付，否则将被取消；而孩子回家告诉大人：学校希望他的家长下个学期用现金结帐。他很后悔，说：我以后再也不闯红灯了！
- 软件工程涉及的就是“软件开发的规范”，在规范里面体现出内在的思想与软件工程之规律。
 - 因为我们的文化传统以及起步较晚的原因，我们在软件工程学方面很落后！

Software Engineering

成功的软件工程要符合大趋势

- 弄懂一种技术是相对容易的, 看清楚趋势就不容易了.
-----比尔.盖茨
- 极端的说法: 从事软件工程及电子商务的人只关心明天的世界!
- 内幕曝光: 当年丁磊是怎么会做起网游的?
 - 当网易的股票市值只剩1美元时, 没有华尔街的专家看好网易, 他们只是建议抛掉网易的股票。丁磊此时希望借网络游戏重整旗鼓, 并与EA公司进行接触, 在四处碰壁后一筹莫展时, 一个叫Fishman的程序员教会了丁磊如何运作游戏, 并拯救了网易。数年之后网易凭借游戏的贡献, 股价一度突破了100美元。

Software Engineering

- 从科利华的倒塌看软件企业发展与风险管理
在企业的成长过程中，风险一直陪伴着他，稍有不慎，便有可能让企业过早的终结生命。1970年名列《财富》杂志500家大企业排行榜的公司，到1983年有1/3已经销声匿迹。1991年，科利华成立并进入教育软件市场；96年科利华收购了名噪一时的晓军公司；98年是科利华生命中最值得炫耀的年头，疯狂炒作《学习的革命》；99年收购阿城钢铁，实现借壳上市-----02年最后因为巨额债务倒闭。
 - 原因：忽视资产风险(外债)和市场风险(互联网发展)
- 结论1：看清趋势最难！因为趋势是思想上的远见！
成功的软件工程必然包含深刻的科技思想文化内涵！
- 结论2：因为对大趋势的认同，杰出的人会走到一起。

Software Engineering

中国当时是“撞”进软件工程的新时代？

- 随着经济发展,中国当时是“撞”进软件工程的新时代,整体水平要比国外的同行晚了几个阶段。(软件工程也讲究的是功夫,我们现在不如人家)但现状很不容乐观:
 - 国际化的趋势是越来越快,在本土软件和外来软件之间的竞争中,我们怎样才能稳步发展呢? (微软与网景)
 - 目标:至少不至于几年后,还象现在这样踉踉跄跄、撞来撞去的。(-----金山与微软的故事)(华为挖爱立信)
- 如何应对:既要研究思想方法,同时消化改进成熟的规范
- 先僵化、再优化,还要注意固化
 - 先照办国外标准,再改进,最后才可能有本土化的开发模式或标准出现

Software Engineering

本土软件工程的希望之光

- 按照软件工程本身的性质和规律，依靠自身广阔市场，大步跨入先进的可能性是存在的。
 - 例如：中国的搜索引擎技术的发展。
 - 例如：软件行业的跨国兼并相对容易，小鱼有可能吃掉大鱼，及全新企业的意外出现：**IBM---微软+Intel---谷歌+苹果---腾讯**是下一个？
- 创新驱动，未来的工业**4.0**是什么样子？
- 中国电子商务：推动传统行业巨变！余额宝！
- 互联网对各个行业的“恐怖”渗透！打车软件。
- 国产**EDA**：奇捷科技，龙芯。
- 鸿蒙系统及相关生态及应用系统。（工具与应用）

Software Engineering

本土软件产业的希望之光

- 阿里巴巴：一个新的全球性互联网产业巨头！
 - “爆红”上市：得益于中国经济转型。
2012----惨淡经营； 2013----高速成长。
 - 阿里的在线生态系统：为个人和小企业只提供交易平台，不与企业竞争其他资源。
 - 阿里巴巴：成长极为坎坷困难！
 - 至于技术含量：某些领域是世界级的(数据库)，也汇集了很多优秀软件平台及商务团队成员。
 - 至于阿里系的金融杠杆与垄断嫌疑，则另说了。
 - 贡献：打破了当时的很多行业壁垒。

Software Engineering

现实中的问题：

----你能否胜任一个称职的软件项目经理？

(或者一个项目经理应该具备什么样的能力？)

- 担当项目经理会面临什么问题？例如：如何培训你的员工成为世界级的软件测试员？
- SE会面临什么问题？管理班级练习项目经验带来的困扰？
- SE的重要性
 - ❖ 新的观点：软件决定计算机系统的价值
 - ❖ 隐藏在计算机系统背后的困难
 - 非编程问题，用计算机及开发环境本身无法解决。
 - ✓ 例如：预算、进度、用户需求的优先级处理等等问题。

Software Engineering

- IT技术人终究要走上管理职位吗？
 - 理由之一：技术生命有限
 - 理由之二：从技术到管理有着天然优势
 - IT行业通常都是团队合作的成果，所以当一个人经验逐渐成熟，他就有责任也有义务带“菜鸟”一起工作。
 - 从技术人员到管理人员的发展有些并非刻意安排发展，常常是在潜移默化中就成长为一名优秀的项目经理，甚至更高级别的管理人员。
 - 理由之三：工程管理发展前景无限广阔。
- 随着时间的推移, 年龄的增长, 能力和经验的积累, 你不会满足于仅仅做一名程序员吧? 不介意做一个项目经理吧?
- 那好, 接下来请您试做一次项目经理!

Software Engineering

项目管理——案例分析（虚拟）

Government Business Process Automation

某城市人民政府进行机构改革，新成立了一个部门——城市管理办公室，负责城市水电、交通、环保、绿地等基础设施以及城市应急处理工作。为了使得新成立部门高效率运转，政府决定利用信息系统辅助该部门的日常业务。经过商讨，最终决定投资8000万元建立三套信息系统，一套是办公自动化系统，一套是数据采集与交换系统，一套是城市应急指挥调度系统，三个系统作为一个项目进行招标，经过激烈的竞争，最终选取了在系统集成、MIS、GIS方面经验丰富的A公司，项目建设周期为1年。

Software Engineering

A公司经讨论确定成立一个由26人组成的项目开发小组，并制订了大致的开发计划如下：

- | | |
|-----------|-----|
| 1. 前期准备工作 | 二周 |
| 2. 项目开发工作 | 9个月 |
| 3. 系统试运行 | 2个月 |
| 4. 系统验收 | 15天 |

Software Engineering

那么，未来的项目经理，请你试着回答以下一些问题：

1. 你认为9个月完成项目开发有可能吗，依据是什么？
2. 根据你对城市管理的了解和猜测，试着综合评估一下这个项目拥有的资源。
3. 请为该项目建立一个风险表。
4. 选择你熟悉的工具确定这个项目的生存周期和项目进度表以及人员分工组织。
5. 为了确保项目质量，请谈一谈如何进行复审以及复审的时间、内容、方式。（以及如何开始启动项目测试等）
6. 请为该项目建一个配置管理库。（版本及软件零部件管理）
7. 制订一份完整的软件开发计划书，格式自定。

Software Engineering

而未来的项目经理也必须关注如下观点：

1. 工作量导向的软件研发项目管理是否完全合理？

（例如：美工的工作量就不能仅仅以工作量来简单衡量。）

（元宇宙-虚拟与增强现实-美好的沉浸感-繁重的工作量）

2. 是否有软件技术团队的规范体系建设的考虑？

（作为一种典型的“软文化”建设，形成最合适的软件技术团队的规范体系是提高凝聚力的重要手段，包括为每个团队成员勾画出更加清晰的成长路线图，让正确的理论得到正确的实施；）

3. 软件开发需要重视对异常的处理

（开发人员知道的最多的恐怕就是在函数实现的时候需要对异常情况进行错误返回，整个系统是否能够处理出现的异常（大型软件的系统性异常），这往往就没有考虑或没碰到）

Software Engineering

未来的项目经理也必须关注如下问题：

1. 软件需求和市场是什么样的关系？软件产品的技术创新一定能真正创造新的市场吗？

——有的项目根本没有确定的需求，甚至不明白用户群来自何方，需要团队来进行调研、分析、假设、验证。

2. 如何引导程序员沿着正确方向去完成软件工程？

3. 如何预测软件任务的难度？

4. 用户信任你吗？

——上述问题也是对系统分析员的提问！

Software Engineering

未来项目经理的人格特征：

1. 活下去！

-----在创业的那些年，我们从来没有想过未来，都在为明天能活下去而苦恼不已。（马化腾）

2. 有想法！

-----实现跨越的组织在看待技术以及技术所带来的变革时，有着与平庸公司截然不同的想法（吉姆柯林斯，美国管理学家，《从优秀到卓越》）

-----字节跳动做的很好。“技术出海”的全球战略。旗下产品有今日头条、西瓜视频、抖音、

Software Engineering

未来项目经理的人格特征：

3. 项目经理应该是文艺青年，而非理性青年！

-----经历跨越技术的路程之后

-----极简主义、唯美主义、直觉主义者。（苹果为什么手机只有一个按钮，乃至无按钮？）

-----我所说的，都是错的。（张小龙）

-----软件的核心价值是用户体验（张小龙）

-----创新并非来自实验室，而是市场的不断变化的需求。（腾讯公司）

Software Engineering

学生读了张一鸣的微博感受：

——越想越可怕~~~~~

1. 我从他的微博中，试图找到他成功的钥匙，得到了一些只言片语，但是仅仅是这些只言片语，我都觉得对我的启发很大。
2. 刚开始项目是99房，类似于安居客或者贝壳找房这样的模式吧，后来才是字节跳动，今日头条，他很早就认准移动互联网红利，很早就认准信息的聚合和推荐的价值。有这样的认知，做出今日头条也不诧异。（基于算法而非编辑推荐）
3. 围观他的微博，我整理了这些干货。发现他说得最多的关键词：延迟满足感，自控，理性，反省，创新，学习。

结论：张一鸣真的太厉害了。（技术与市场需求的感知能力）

 教材 (Text Book)

Software Engineering

Theory and Practice

Fourth Edition

软 件 工 程

理论与实践

影印版 第4版

Shari Lawrence Pfleeger

高等教育出版社 · Prentice-Hall, Inc.

Software Engineering

Introduction (to this lesson)

① **The position or role of the SE (软件工程地位及角色)**

oversea: two degrees

----computer science and technology

----software engineering [SE]

(see the websites of famous universities)

**domestic: ----more important than ever as an
industry (行业)**

----finding a job or obtain employment

actuality: SE have standards:

**CMM---certification (资质) of software construction
in international market.**

Software Engineering

CMM现状(2005年4月)

全球参与CMM认证的机构约有7800家(截止08年), 其中印度公司占了顶级CMMI评级的大约65%。

- 中国, 210家。2002年8月还是37家, 2003年4月73家。
 - ❖ 5级9家(包括独资、合资等)
- 印度, 380家。2002年8月他们的数据是187家, 2003年4月238家。
 - 5级66家(印度本土)
- 美国现在有2117家, 2003年4月份时为1671家
 - 5级?家

CMM现状(中国: 2006年6月)

- 5级11家(包括独资、合资等)

CMM现状(中国: 2011年6月)

- 5级11家(包括独资、合资等), 但2级/3级数量增加.

Software Engineering

② Extent and Arrangement (授课范围与安排)

A: Advantage

B: Arrangement

part 1:chapter 1 – 3(introduction、 project plan
and management)

part 2:chapter 4– 11(major steps in process)

part 3:chapter 12– 14(evaluation、 improvement)

C: Focus on---examples (重点举例范围: 需求分析,
软件设计, 软件测试)

D: Goal : excellent **Analyst and Designer
or Project Manager**

**E: manifesto: good SE exercise example document
will be awarded !**

Chapter 1 Why Software Engineering

The Background of the SE (软件工程背景)

①The notion(概念) of software (=code+data+document)

Feature A:Software is logical entity.

B:The producing is different from hardware's

C:Complexity (软件涉及每个行业的综合知识)

D:Social factors in development(观念、体制等)

②Software Crisis

A: appearance: 1970s----up to date

software pervades our world

lose face: scale↑increased → quality↓decreased

cause: scale↑increased → nature of software↑changed

Chapter 1 Why Software Engineering

B: software crisis: a series of problems met in software development and maintenance.

C: exhibition(表现):

- 对软件开发**成本和进度的估计常常不准确**。(开发成本超出预算，实际进度比预定计划一再拖延的现象并不罕见)
- 用户对“已完成”系统**不满意**的现象经常发生。
- 软件产品的**质量往往靠不住**(Bug一大堆，Patch一个接一个)。
- 软件通常**缺乏适当的文档资料**，因而可维护程度非常低
- 软件的**实际成本不断提高**。
- 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。**(Software producing---craftwork)**

Chapter 1 Why Software Engineering

D: example

U: IBM (1970s)

----OS/360, 4000 models, 1 million lines of codes,
5000 PY(人年), losing: 4亿美圆。

(Fiasco! Catastrophe!)

V: windows 2000/ XP/ Vista /7 / 8 /10安全漏洞:

----内部运行的远程登录协议导致的安全隐患: 拒绝服务、权限滥用、信息泄露。安全漏洞导致**DOS**攻击,无法提供服务; 其他两种缺陷会使攻击者控制整个计算机。

----不得不下载大堆补丁程序, 安装补丁时间比安装初始系统本身长得多。

Chapter 1 Why Software Engineering

W: 丹佛新机场启用推迟**16**个月

- 复杂的计算机软件系统，**5300**英里的光纤网络。
包裹处理系统的一个严重缺陷，导致行李箱被绞碎，居然还开着自动包裹车往墙里面撞！
- 预算超过**32**亿美圆！

X: Intel pentium

(4195835 / 3145727) * 3145727 - 4195835

=0，浮点运算没有问题；

≠0，浮点除法存在缺陷；

- 公司用来支付更换坏芯片的费用：**\$400,000,000.**

Chapter 1 Why Software Engineering

E: What's the new approach ?

assumption(设想)--system engineering--use in SE—
--emphasize administrable developing approach
(a software--several parts—detail specifications on
every part—detail development—tie parts together
-----high quality software)

③ Introduction to “ SE ”(软件工程简介)

A: importance (good SE ensure high quality life)

B: key issues in SE (at both theory and practice)

C: goal (give solutions incorporating quality software,
and consider characteristics that contribute to
the quality)

Chapter 1 Why Software Engineering

D: approach

analysis----analyze problems, examine examples of software failure or success(正反两方面).

design----give solutions.

developing team----peoples involved in developing, describing their roles and responsibilities .

develop----realize the solutions (OO--realize objects, activities, enclosing, test, etc.)

project management----divide system in parts, define process in steps, schedule control, deal with every changes, etc..

Chapter 1 Why Software Engineering

1.1 What is SE （什么是软件工程）

SE = nature (making the problem clearly)
+ solution (implement by developing tools)

Solving Problems （用系统科学的方法解决问题）

① analyzing: large problems = { subproblems } +
{ interrelationships } (Fig1.1)

② synthesis: composition of the individual solutions

③ several notions:

tool : instrument / automatic systems/programming
environment /software tool module

procedure: combination of tools, technique, method.

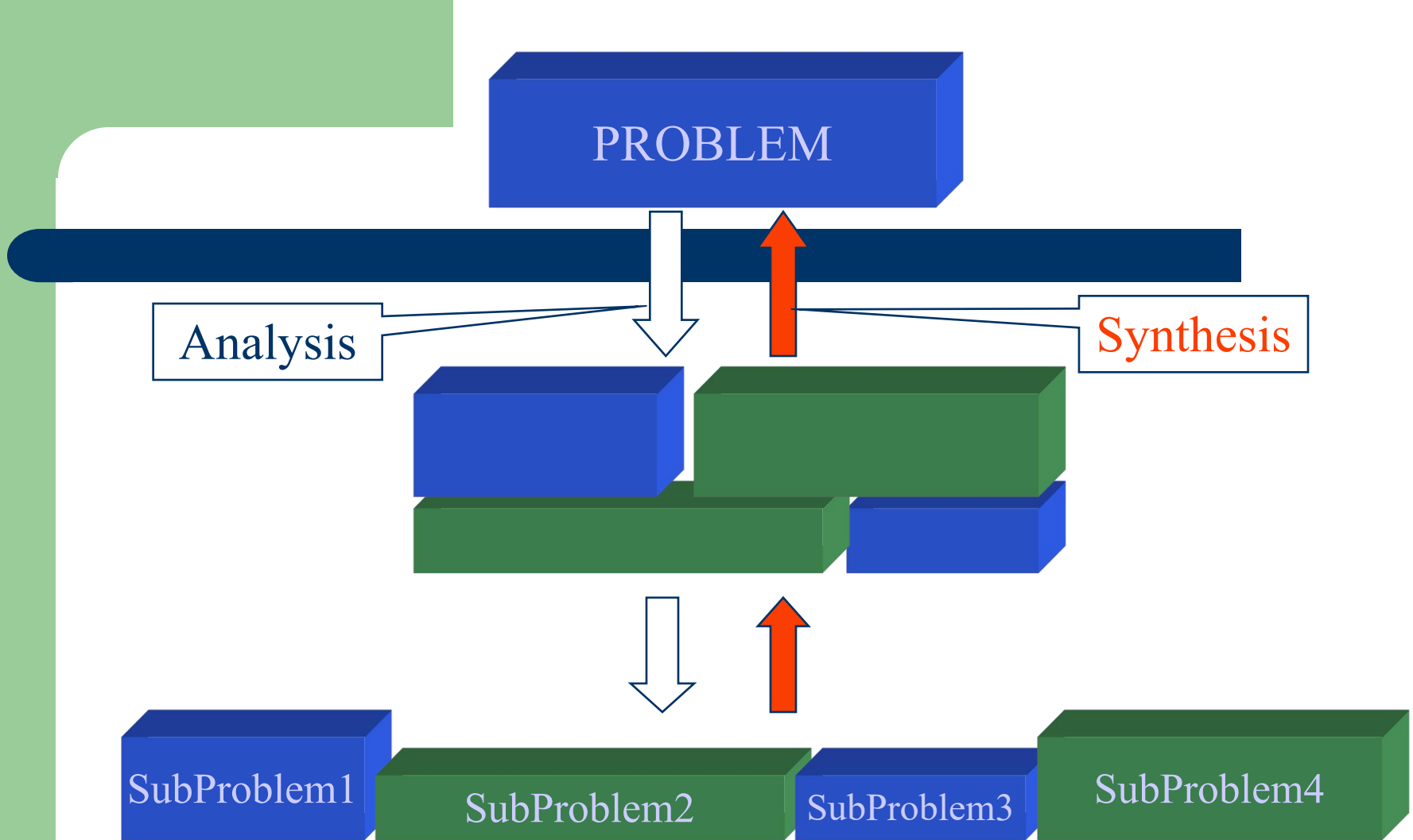


Fig 1.1 & 1.2 The Process of Analysis and Synthesis

Chapter 1 Why Software Engineering

paradigm : (model / approach / philosophy)

(模式--开发软件时特定的方法、途径或哲学)

- 面向对象开发的各种模式（例如：**MVC**模式）
- 结构化开发的模式，基于过程的模式，某种订制的模式等等。（分布式的开发模式**Hadoop**）

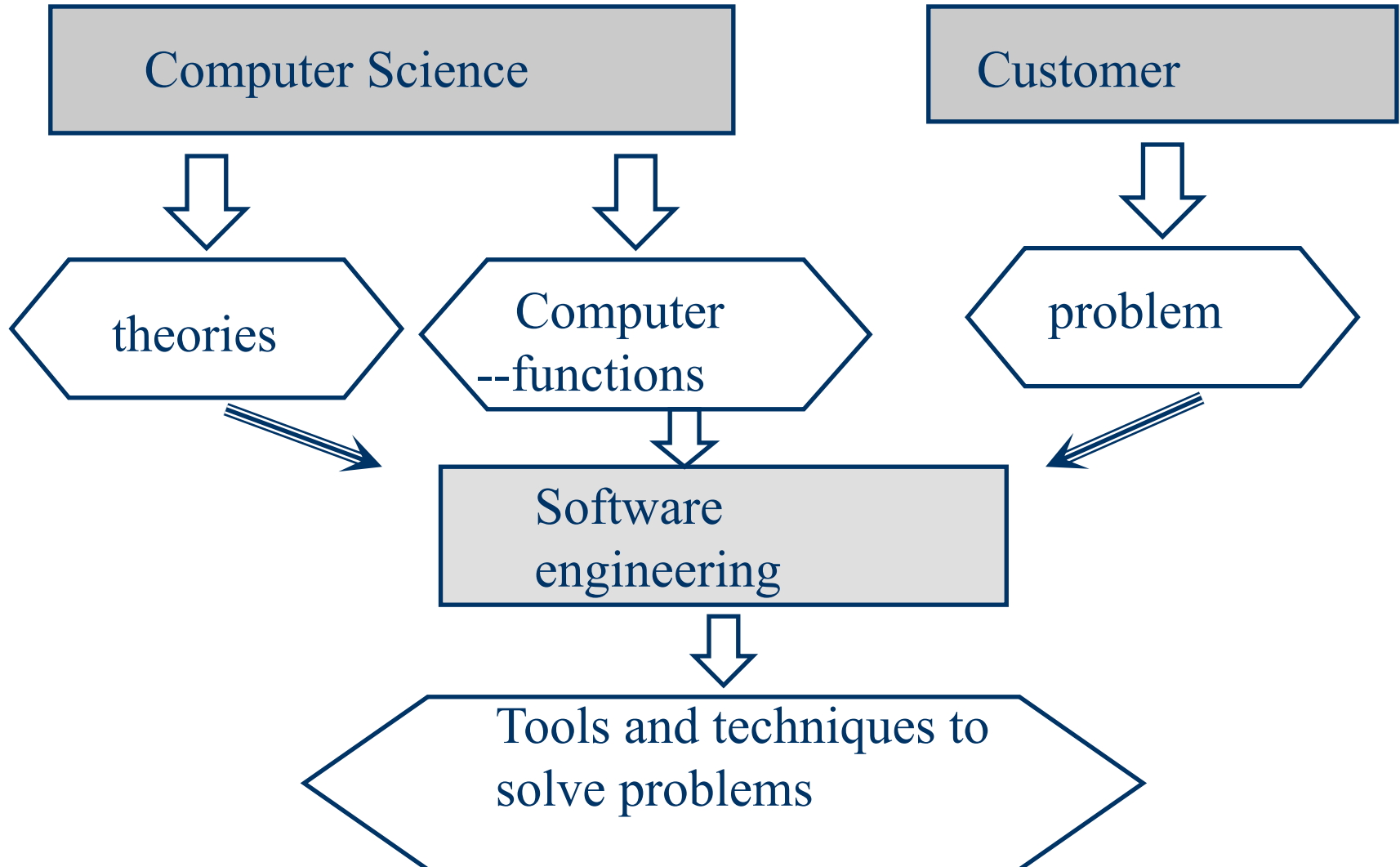
Where does the software engineer fit in ?

（软件工程师的任务）

task: 以计算机科学理论和计算机功能为基础，通过对要解决问题的本质的了解，采用相应的工具和技术，实现设计方案，推出高质量的软件产品。

SE = a **method** using tools and technique to solve problem （采用工具、技术等用来解决现实问题的综合过程）（如下页图1.3所示）

Chapter 1 Why Software Engineering



Chapter 1 Why Software Engineering

1.2 How successful have we been ?

（软件工程方面的进展）（成就及存在的问题）

Several aspects:

- ① SE is an art(学问) of generating high quality software.
(include: designing art, ingenuity, skills
goal: robust, efficient, easy developing and maintain)
- ② Great changes in today's living and working
- ③ Problems (not acceptable when deliver to users)
- ④ Influence of software problems（软件“问题”的影响）
error: a human made mistakes in software producing
(misunderstand requirement, wrong codes, etc.)

Chapter 1 Why Software Engineering

fault: problem in function implementation
(one error----several faults) (static exist)

failure: failure in running software (because of faults)
(dynamic exist)

⑤ Failure Example:

A: Sperry Corporation (P7)

----project: income tax form processing system

----cause: “inadequate planning”(项目计划不周)

----Skepticism: (fault-free software is not practical)

----Practical reason:

reliability of safety-critical software---- 10^{-9}

B: radar vs. jet fighter (PX)

----cause: “unanticipated use” (软件未能预期的使用)

计划阶段的问题

设计阶段
的问题

Chapter 1 Why Software Engineering

⑥ Zero-defect software(零缺陷软件) :

---- is impossible , because of “market forces”

⑦ Discussing about “bugs”

A: fixing(改正) is more difficult than rewriting(重写)

B: error correcting is costly (the activity in the latter period of life cycle is more costly)

C: review(复审) is very important

----is a criteria(规范) in formal developing team

----classes and statistics (分类及统计数字)

❖ Self check: (by running test cases)

---- reveal 1/5 faults

❖ Peer review (by colleagues)

---- reveal 4/5 faults

Chapter 1 Why Software Engineering

1.3 What is good software ? (什么是高质量软件)

- Purpose (about SE)-----assure software has quality and utility (**SE**确保软件具有技术高质量和实际商业价值)
- What we mean by high-quality?
 - ❖ different perspectives----sidebar 1.2
 - ❖ different context(应用背景) have different answer about quality (P10) -----字处理软件的容错程度在安全攸关的系统中，则可能是无法接受的！
- conclusion : made by special issue of IEEE software on quality)
 - ❖ Quality of the product
 - ❖ Quality of the process
 - ❖ Business value

Chapter 1 Why Software Engineering

- ① The quality of the product (最终)产品的质量
judging quality A:users (enough functionality, easy to
learn and easy to use)
B:developers (internal characteristics)

Figure1.5:

- { The left--external quality factors(因素)(from user)
- { The right--quality criteria(细则) (from developer)
- { users: track number of failures, etc.
- { developers: track number and type of faults, and
faults distributing area
(缺陷分布区间—在各个开发阶段/步骤)

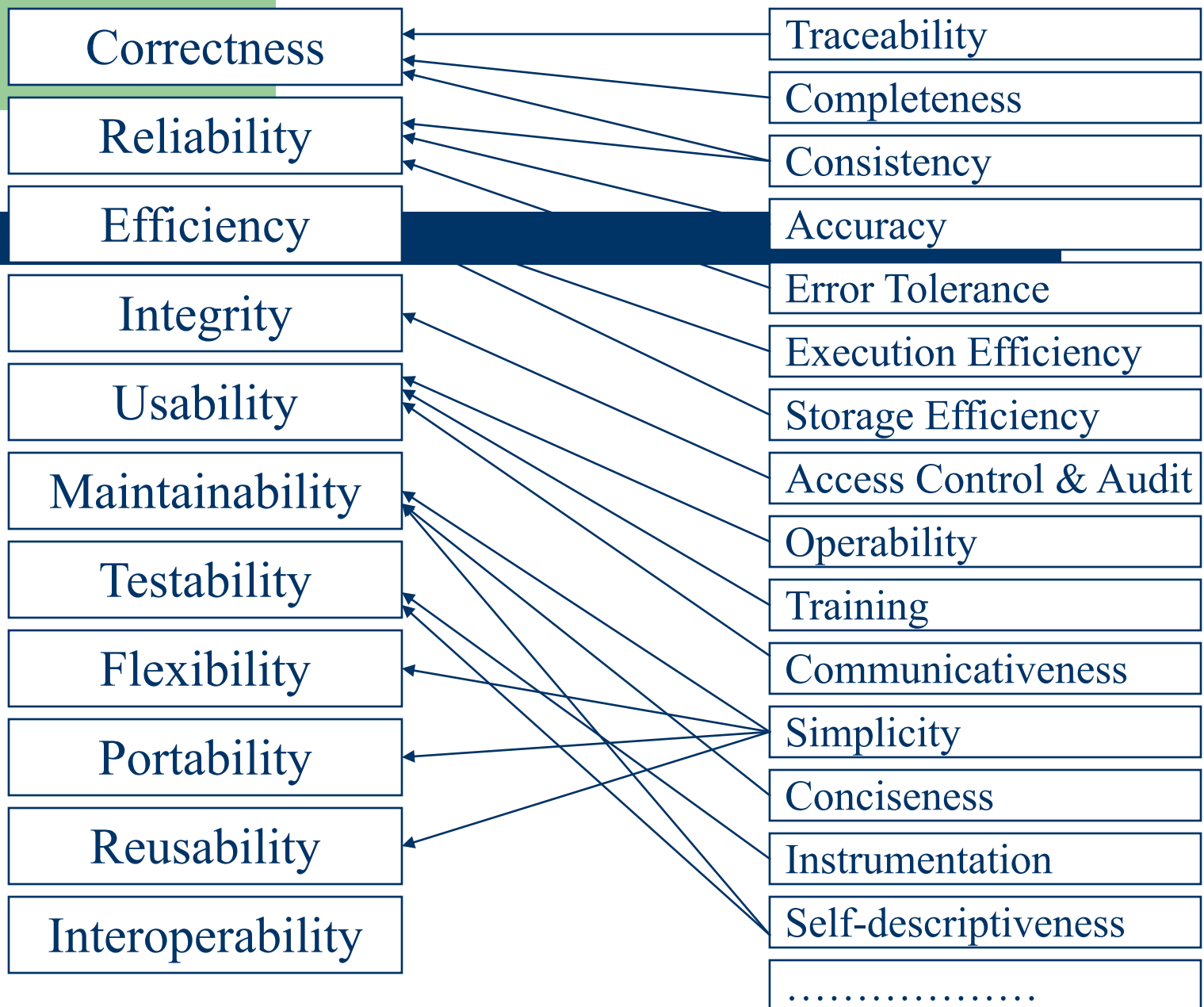


Fig 1.5 McCall's Quality Model.

Chapter 1 Why Software Engineering

- ② The quality of the process (软件开发及维护) 过程的质量
many activities----influence the quality of final product
(so, the process has the same importance)

Questions in Page 11----至少有量化的过程才能回答之!

In 1990s: modeling the process is a “pop” topic

Models: { CMM---Capability Maturity Model (5 levels)
ISO 9000—900X--1900X----该标准影响较小
SPICE---软件过程改进和能力确定模型

A: CMU-SEI的CMM/PSP/TSP;

- 美国卡耐基梅隆大学软件工程研究所 (SEI)
- 能力成熟度模型 (**C**apability **M**aturity **M**odel)
- 个体软件过程 (**P**ersonal **S**oftware **P**rocess)
- 小组软件过程 (**T**eam **S**oftware **P**rocess)

Chapter 1 Why Software Engineering

B: ISO 9000质量标准体系；

- 国际标准化组织 (ISO)
- **ISO9003**是专门为软件行业定制的

C: 以上两种质量体系都强调了软件产品的质量。

D:区别

但9000
以上的
标准较
细致些

----**ISO9000**强调的是衡量的准则,该做什么、什么算好、什么算不好;没有告诉软件开发人员如何达到好的目标,如何避免差错,它实际上是由英国学院派一批没有做过复杂系统的人制定出来的标准. (八股味道浓厚)

---- **CMM**则提供了一整套较为完善的软件研发项目管理的方法。美国先后在这上面投资了**6**亿多美元,做了很多实践工作来改进软件研发项目管理,而且其内容还在不断地改进。(例如较新的**CMMI**等。)

Chapter 1 Why Software Engineering

---- **CMMi**亦有其缺点:

----重量级方法强调以开发过程为中心,而不是以人中心,这在中国有点“水土不服”。

----**CMM/CMMI**的评估耗资不菲:一个**CMM2**级评估就可能达到数十万至数百万之巨,而且耗时很长,过程十分复杂,常常导致效果不太理想。不少企业的认证流于形式,评估完成后就只留下一大堆文档,而真正的软件开发过程却依然故我。而且,**CMM/CMMI**只告诉我们应该怎么做,而没有具体地告诉如何做。比如说,它要求必须改进需求管理,那么到底该如何做需求管理?**CMM/CMMI**没有提供答案。

E: Question--How the software development

organization update their CMM level ?

(CMM等级评价的实测过程之一: 检查 5个大的代表性项目, 几十个一般性项目, 随机抽取项目组成员然后“克隆”一个项目全过程。甚至两个子团队做同一个项目. 另外, CMM等级评估机构还有抽查活动)

Chapter 1 Why Software Engineering

③ Quality in the context of the Business Environment (商业应用背景下的软件质量) (商业质量)

A: The relate and difference between technical value and business value 技术价值与商业价值的联系与区别

technical value: technical index(speed, rightly running time, maintenance cost, etc.)

business value: 机构对软件是否与其战略利益相吻合的一种价值评估

go wrong(误区): technical quality will automatically translate into business value

goal: unify the technical quality and business value in a software (将技术价值与商业价值统一起来)

商业价值一般需要在需求阶段预判

Chapter 1 Why Software Engineering

CMM—try to unify the technical value and business value（也开启了经验管理到量化管理的转变）

Example: Hughes Aircraft: improved its productivity by 4 to 1 and saved millions of dollars (by CMM)

Raytheon: twofold increase in productivity and a \$7.70 return on every dollar invested in process improvement (be conducted by CMM)

Hope----company should has business personnel, like the project managers do in *microsoft*

Chapter 1 Why Software Engineering

技术价值与商业价值的结合案例—滴滴出行

- 1、先进的营销策略----**有钱（庞大的融资规模**35**亿美元，背后站着腾讯和阿里）、规模（滴滴自曝打车业务掌握了**99%**的市场份额，日需求**800**万单，覆盖**1000**多个城市，专车占据了**78%**的份额，日需求**600**万单，覆盖**360**个多个城市。）、补贴（滴滴力图给所有竞品造成这种印象，即它可以在自己需要的时机和场景随时发动价格战，这是有效的威慑）。完成了品牌迭代的“滴滴出行”正变成创业者的梦魇，恐惧是唯一比死亡还可怕的东西，优步、爱拼车和考拉班车就是吓死而不是战死的。
- 2、技术服务于市场----**虽有各种技术升级方案，但是市场第一：
（1）攻：用不断发掘的新场景维系平台的活力，推高估值，哪怕它不赚钱；（2）防：阻止竞争产品在某个被忽视的角落突然坐大；
- 3、深挖政策空间，摸清政府思路----**政府：维护市场秩序，但是鼓励创新。一开始交通部明确了私家车不能做专车，但形势比人强，主管部门能做的只是极力维护“正常”的市场秩序而不是对创新痛下杀手，所以已经完成原始积累的滴滴反而是比创业公司更合适的对话方；而滴滴也非常需要官方的认可、合作或收编以架起通往**IPO**的捷径

Chapter 1 Why Software Engineering

B: The business value of process improvement
(改进过程所带来的商业价值)(或者说对商业质量的影响)

**X: traditional formula : Return On Investment = capital
+benefit+allowance for risk**

**Y: software producing: (view investment in terms
of effort(将工作量视为投资), rather than cost or dollars)**

**Investment = normal effort + effort for process
improvement**

Nine investments(effort) concerned “return”(回报) :
training; schedule; risk; quality; process;
costs; productivity; customer; business.

Goal: reducing the cost by minimize the effort

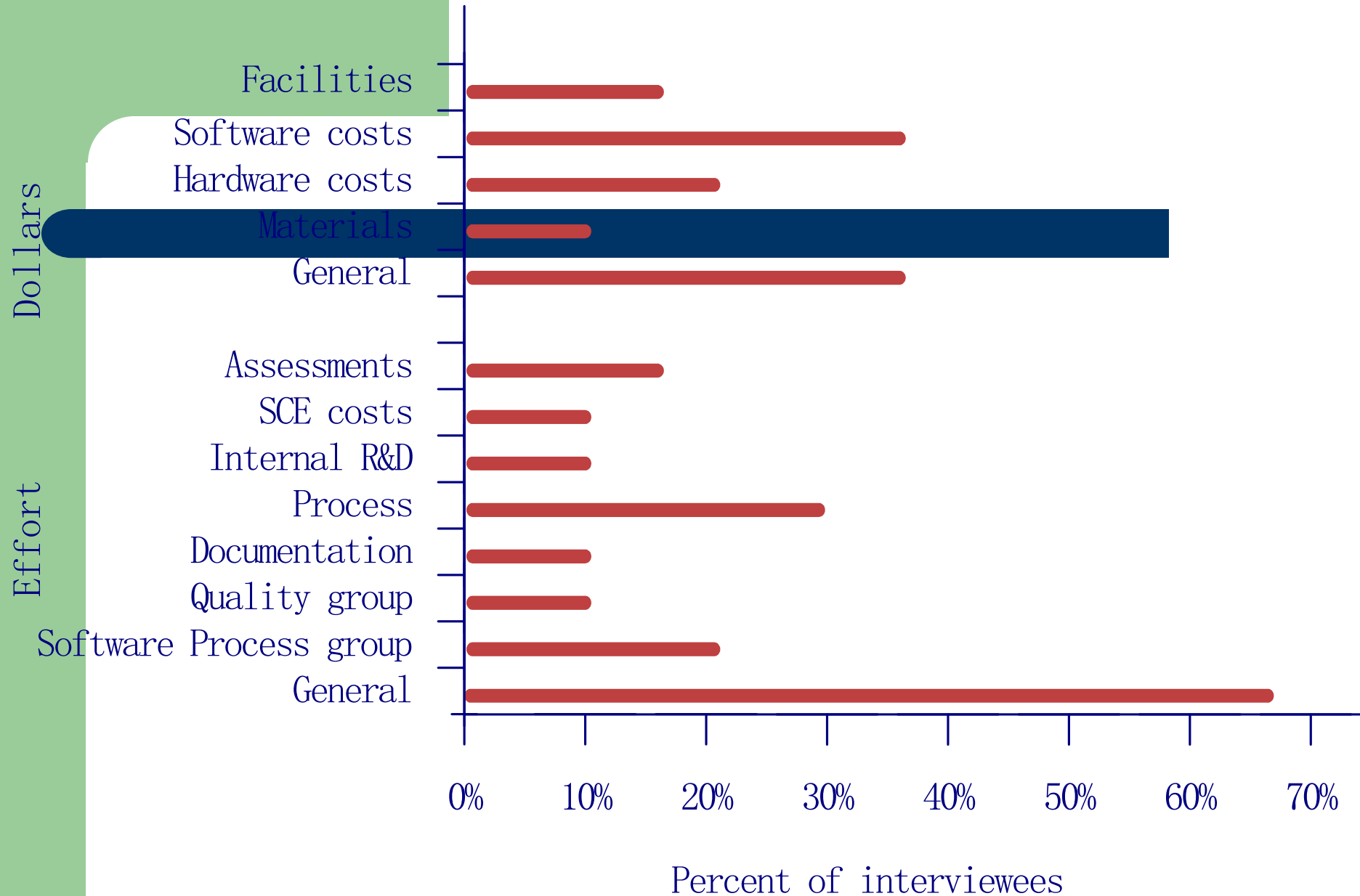


Fig 1.6 Terms included in industry definition of return on investment

Chapter 1 Why Software Engineering

C:The definition of ROI in software industry (Fig1.6)
(一般行业)Dollars —————> effort(针对软件业)

1.4 Who does software Engineering ? (参与者)

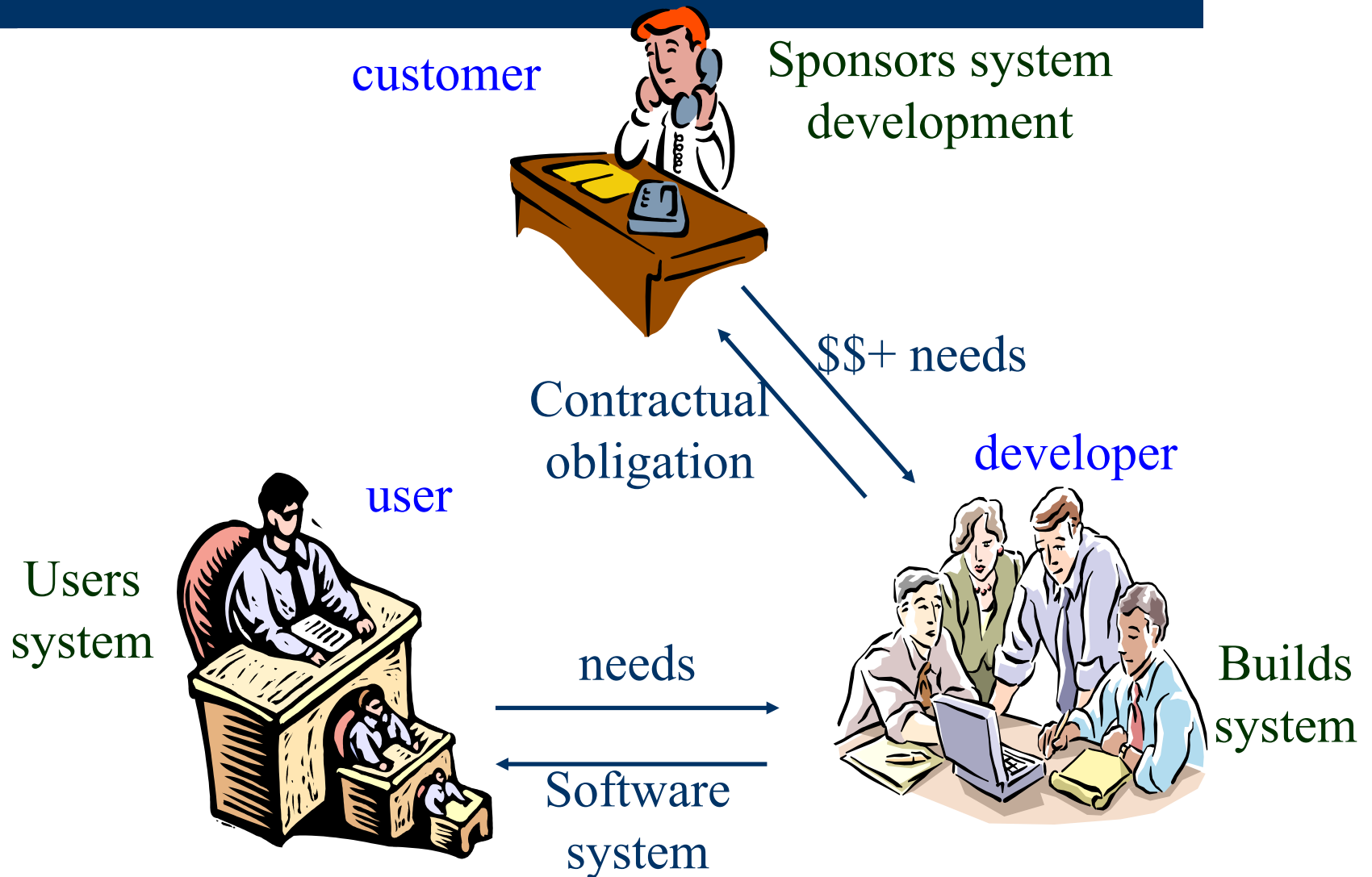
Focus: communication
customer ←————→ developer

①participants: customer (客户): 公司、机构或个人
developer (开发者): 公司、机构或个人
user (使用者): 软件(直接)使用者

Notes A: sometimes the customer is not a user

B: sometimes the customer,user and
developer belong to same organization

C: subcontractors (concern with COTS)



Chapter 1 Why Software Engineering

② Conclusion:

SE is a system engineering


- { A: include the analysis and synthesis about technical problems (project development)
- { B: includes the management about the process and the anticipants (project management)

核心思想: 是把软件产品（就像其他工业品一样）看作是一个工程产品来处理，把需求计划、可行性研究、工程审核、质量监督等工程化的概念引入到软件生产中，期望达到一般工程项目的三要素：进度、经费、质量。

Chapter 1 Why Software Engineering

1.5 A System Approach(一种系统的开发方法)

Cause:

hardware/software ^{interact}  users, other software tasks,
other pieces of hardware,
existing databases, or
other computer systems

Conclusion: Boundary(边界) is important(description to
what is included in the project and what is
not in that)

Example: Writing a program to print paychecks (P16)

software = { objects and activities } + { relationships }

an activity = inputs+actions+outputs

Chapter 1 Why Software Engineering

① The Elements of a System (系统的要素)

A : Activities and Objects

activity : something happens in a system
(transfer one thing to another thing)

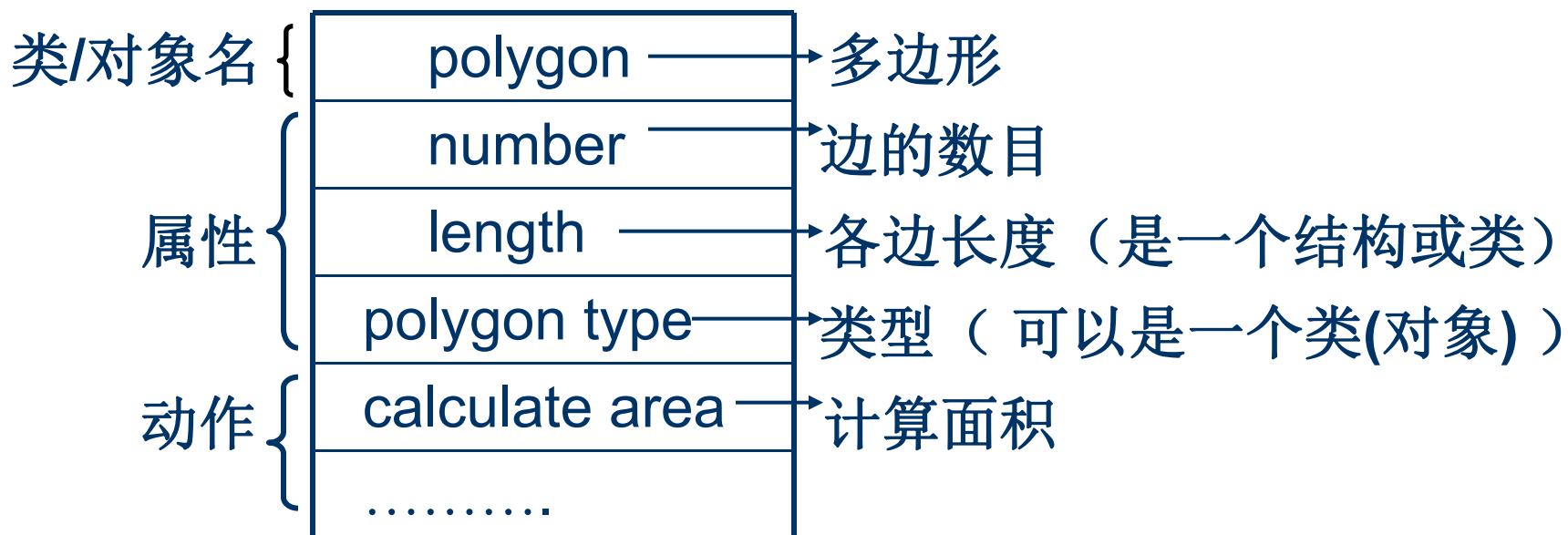
object : the element(元素) involved in the activities
which includes : characteristics + actions

[note: the object now discussing is different from that one in OOP]

[现在讨论的对象的概念与OOP中的对象有所不同, 有些太大, 而有些较小, 需要分解/合并才可演化成为后来OOP中的对象, 有些则现在大多只是些属性类聚合而成的实体]

Chapter 1 Why Software Engineering

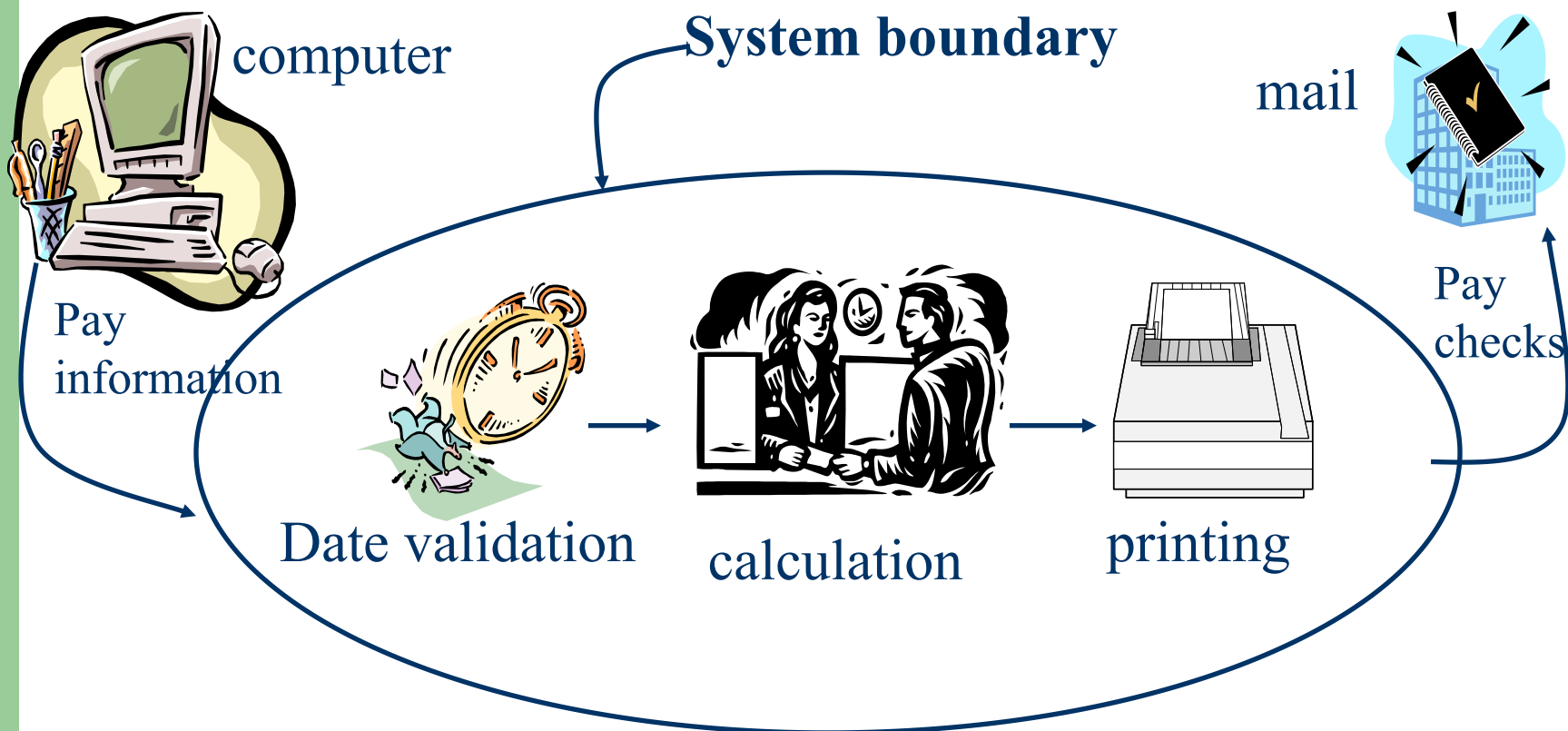
Example: (如果采用类似OOP中对实体的描述方法的话:)



但是大凡做软件，却不是从设计上述似是而非的类开始的，而是从一个更宽阔的视角开始的：

Chapter 1 Why Software Engineering

图1.9 工资单系统的定义----主要由活动和边界组成, 若再细分活动成系列动作时,可以得到对象极其关系的定义



Chapter 1 Why Software Engineering

B: Relationship and the System Boundary

relationship : a description among entities and activities(对实体和活动中数据项及动作的相互关系的描述)

boundary : the scope(范围) of items and processing

A system = **entities(实体) + activities(活动)**
+ relationships(关系) + boundary(边界)

C: Example: Fig1.8----Respiratory system (P18)

② Interrelated Systems (相互关联影响的系统)

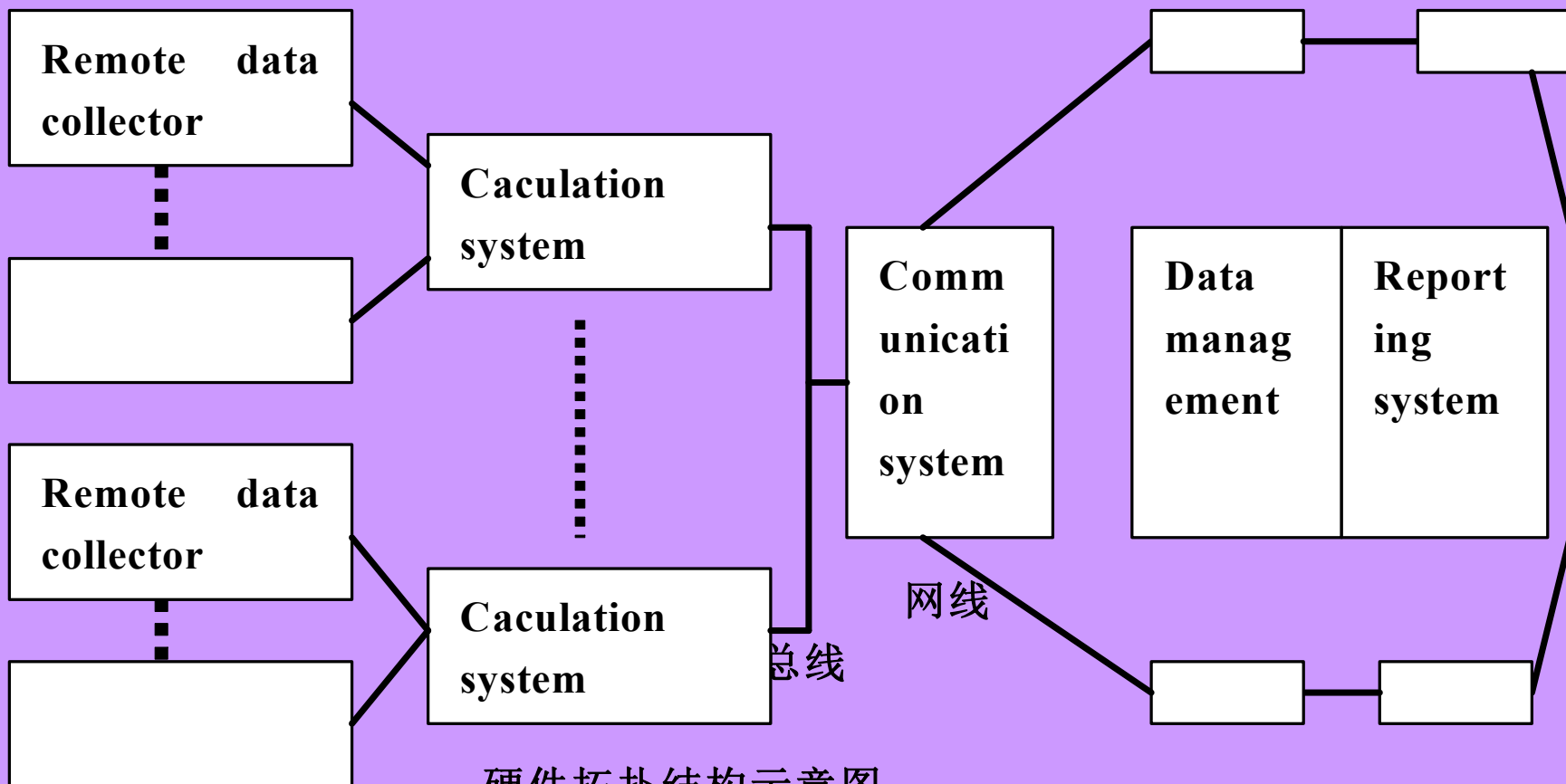
A: Describing boundary(刻画边界) is important

assumption : large system = {small subsystems with their own boundaries and relationships}
-----is reasonable

Chapter 1 Why Software Engineering

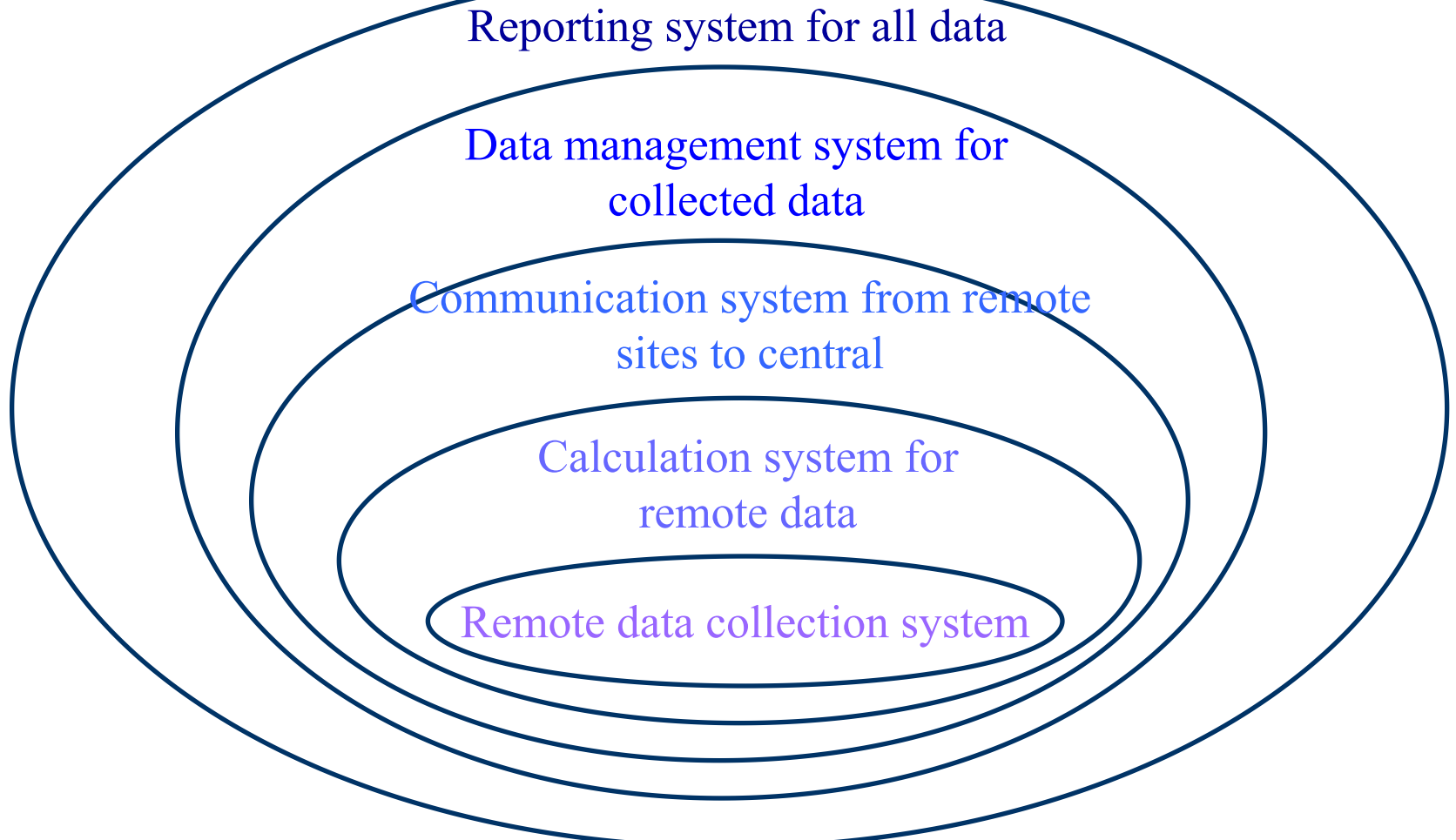
B: Example : water-monitoring system(水文监控系统)

按照数据层级设计



硬件拓扑结构示意图

Chapter 1 Why Software Engineering



Chapter 1 Why Software Engineering

Note : developing from the inner layer, but the objects and activities is hard to considering (from the beginning , we must give the frame of the objects and activities)

软件设计：团队工作基本上应该同步开始，虽然开发与测试的软件组装顺序是由内而外，但即使最外层的软件也必须基本同步建设好，这对软件工程的管理者带来了挑战，需要良好的规划设计。

本节作者想说的话：较大型软件工程需要框架规划设计。

C: Incremental development approach(增量式开发方法)

system $A \rightarrow A' \rightarrow A'' \rightarrow B$ (A', A'' is semi-finished product (中间产品)) (不明确设计目标的情况下)

预约讲课联想到的号外话题：

比亚迪公司（同等学历硕士班）讲课
外企秋招面试有感而发（纯技术类公司，前两道题是技术问题。）

选择题：下列行为中，具有侵犯著作权行为的是_____。

- A：** 商场为烘托气氛播放了一些在音像店购买的正版音乐**CD**。
- B：** 未与原作者协商，将已经出版的书籍翻译成盲文出版。
- C：** 为了备份，将自己的正版软件光盘复制了一张。
- D：** 模仿某知名软件的功能和界面，开发一套相类似的系统。

Chapter 1 Why Software Engineering

1.6 An Engineering Approach(一种工程化的方法)

System's nature → begin to construction → artist(工匠) (use workbench)
craft(工艺)(use artistic approach)

① Building a House (Page 22--23)

conference → drawing / plan → construct

- ① document (detail plan / timing)
- ② building and change and modification(P22)
- ③ testing / verifying

→ move in → (documentation, etc.) maintenance

Summarize the construction process:

about 8 steps (see p22--23)

Chapter 1 Why Software Engineering

② Building a System (P23)

Software project progress
in a way similar to the
house-building process



customer:Howells
user:Mr.Howell's parents
developer:McMullen

1.5节：一般软件制作时的规划与部件划分问题的思考
本节：大型软件开发过程中的工程化途径与方法

Chapter 1 Why Software Engineering

简单
业务
模型

Phase 1:

Requirement
analysis and
definition

需求分析

展示：皇家机动车服务站问题定义，迷你图书馆用例图，范文的领域图，单独的类图。

meeting(by users, customers)
and developers

A:problem definition (问题定义)

B:feasibility research (可行性研究)

C:requirement analysis (需求分析)
(文档: 《**SRS**》 软件需求规格说明书)

--logical architecture(系统逻辑结构))

--Software Requirement Specification

(learn about boundary, entities,
activities) (what we will do)

D: review(复审) (by all anticipants)

Chapter 1 Why Software Engineering

Phase 2:

System Design
系统设计

- A: appearance/interface 用户界面
- B: 《**SAD**》 software architecture diagrams and reports 系统结构图 (how to do it)(拓扑,功能,数据架构)
- C: review(by developer and customer)

Phase 3:

Program Design
程序设计

- A: description for modules(模块功能算法与数据描述)[documentation]
- B: review (by developers)

Phase 4:

Program Implementation
程序实现

- A: coding(编程) and debugging [source codes and comments]
- B: review (by developer / coders)

Chapter 1 Why Software Engineering

Phase 5:

Unit Testing
单元测试

A: test for functionality and performance 模块功能与性能测试
(generate **test report**)(程序设计)
B: review (by test team)

Phase 6:

Integration Test
集成测试

A: integrated test by software architecture diagram(结构图 **SAD**)
(generate **test report**)
B: review (by test team)

Phase 7:

System Testing
系统测试

A: test the all functions by **<SRS>**
(按 **SRS** 对系统总体能力等进行测试)
(the last testing before delivery)
B: review (by developer and customer)

Chapter 1 Why Software Engineering

Phase 8:

**System
Delivery** 系
统提交

A:delivery交付(several ways in the stage:directly,parallel,gradually)
(include: user training)

B:review (to **user manual** and operator manual)

Phase 9:

Maintenance
维护

A:a process for modifying software or meet new requirement
(修改软件的过程,为改错或满足新需求)
(correct, adopt, complete, prevent)
(generate **maintenance report**)

B:review (by maintenance team)

Chapter 1 Why Software Engineering

对上述大型软件工程化制作途径的说明01:

以上阶段只是大致的划分，软件团队实际执行时比这要复杂，还有若干辅助性过程和阶段，而有些阶段实际是相互重叠、相互影响的。

例如测试：

- 1、关于测试阶段，实际上在需求分析和系统设计阶段就可能开始制定和完善测试计划了。
- 2、测试阶段要做的工作其实有很多，例如系统测试时仅针对“非功能性需求”，就有许多类型的测试：例如：负载与压力测试、稳定性测试、安全性测试、可用性测试、所谓的“国际化，全球化和本地化”测试、地缘政治测试、卸载测试等等。移动**APP**等测试又有许多特有的测试分类。

Chapter 1 Why Software Engineering

对上述大型软件工程化制作途径的说明02:

以上阶段只是大致的划分，软件团队实际执行时比这要复杂，还有若干辅助性过程和阶段，而有些阶段实际是相互重叠、相互影响的。

针对第二章的过程模型说明：

- 1、面对开发团队实际情况（能力、习惯等），对上述九个阶段有必要进行剪裁和组合。形成不同的过程模型。
- 2、面对开发项目的实际情况（特点、要求等），对上述九个阶段有必要进行剪裁和组合。形成不同的过程模型。

每一位工程师或参与者可能擅长于软件开发的某一特定方面, 承担某些特定职责, 扮演某个特定的角色

1.7 Members of the development team (P25)

Activities	Personnel	Role
①Requirement:	Analyst Customer	analyzing (break things into component parts)
②System Design:	Analyst Designer	generating system-level description
③Program Design:	Designer Programmer	detailed description (algorithm,data, etc.)
④Program Implementation:	programmer	coding
⑤Unit Testing:	programmer tester	catch faults

Chapter 1 Why Software Engineering

- ⑥ Integration Testing: Test team → check functions
(according to the “system design”)
- ⑦ System Testing: { test team
customer
trainer } → check requirement
(according to the <SRS>)
- ⑧ System Delivery : trainer → training /educating
- ⑨ Maintenance: maintenance team → find faults
meet new requirement
changes

The maintenance team includes: analysts + designers
+ programmers + testers + trainers + users/customers

Chapter 1 Why Software Engineering

Explaining :

①class projects:

- rare documentation, stable requirement
- demonstrate ability, but not necessarily to solve a problem for a real customer

②facing real customer

- ❖ different context: scale, complexity, frequent changes, great desire for document and maintenance
- ❖ project management is hard: develop team will relatively stable

③Librarians: managing documents, do simple test

Configuration management: 维持一个软件的不同版本之间各种文档的对应关系。

Chapter 1 Why Software Engineering

④ Different SE ➡ different number and species of personnel in developing different software

⑤ 软件架构师: 属于高级程序员，侧重开发过程和模式的选择和论证（在国内和分析员差不多,其工作重点与分析员有所不同（分析师与设计师兼顾吧），但就开发来说，其工作似乎更重要些，而分析员的工作更偏重于市场与用户需求）。

Chapter 1 Why Software Engineering

1.8 How has software engineering changed ?

(软件工程发生的变化及其起因)

Comparing { Building a house:
Building a software: } Different judgment

A: why is there a difference ? Why are we as software engineers having such a difficult time producing quality software ? (假如能轻易列出软件工程的开发步骤, 那么, 为什么软件工程师生产高质量软件如此艰难?)

B: result (运用软件工程手段生产出失败商业软件的危害) :

X: not meet the user's demand (due to changes)

Y: waste the time and endeavor

C: difficulty and complexity in real developing
(examples P27-28)

D: conclusion: flexibility is essential !

(each project is unique.)

**(以灵活的方式运用软件工程工具与技术是至关重要的)
(研究软件工程的较大规模分析设计技术也至关重要!)**

E: Interface(软件接口): has become a “large problem”

**F: Conclusion: SE \longrightarrow more complex(un-independent systems, coordination between interface)(大型系统接口不是仅事关程序员的技术问题，而是结构性问题和其他逻辑性解读、精确性、完整性等等问题) + too many changes + 高速/并发的系统环境(同步性等等)。
上述情况最终带来了一系列问题。**

Chapter 1 Why Software Engineering

1.The nature of the change

① Early application system (回顾早期的应用程序)

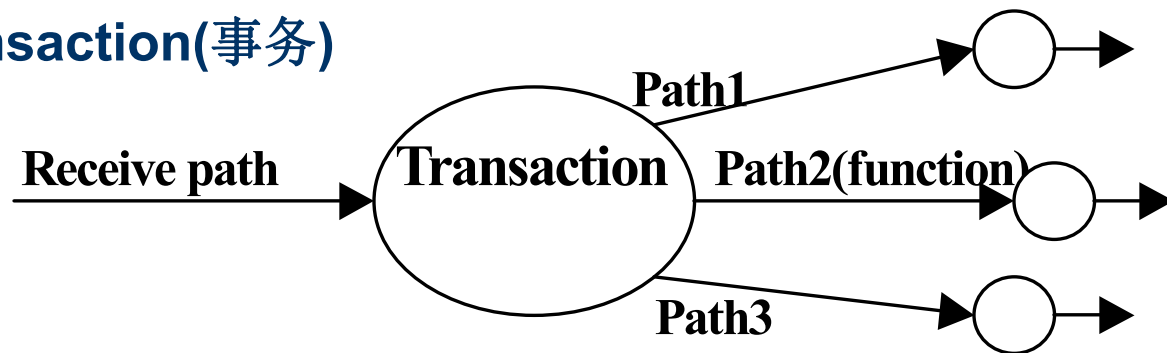
A: input is linear (single processor, mainframe)

B: output is alphanumeric

C:system X:transformation(变换)



Y:transaction(事务)



Chapter 1 Why Software Engineering

D: Conclusion: “waterfall” approach is suitable

② Today’s software—based system: far different and more complex

A: run on multiple systems

(may be is c/s or b/s or other architecture with distributed functionality) (以及跨平台系统等)

B: normal functions , network control , security , user-interface , data or object management etc.

C: conclusion: great changes → the traditional SE approach(waterfall model----假定开发活动为线性) is no longer flexible/suitable

Chapter 1 Why Software Engineering

③ **seven key factors** that have altered SE practice
(使SE实践发生变化的七个关键因素) [by wasserman]

A: **criticality of time-to-market** for commercial products
(商业软件的投放市场时间的紧迫性)

B: **shifts in economics of computing** (lower HW, higher development/maintenance costs)(计算经济学的改变)

C: availability of **powerful desktop computing**
(强力的桌面计算平台的可用性)

---desktop computing puts development power in the users, engineer should or can build more complex system than ever . (BIG DATA处理、云计算、物联网与智慧服务等等) (移动平台5G+计算)

D: **extensive local- and wide-area networking**
(局域网和广域网的延伸)—CS/BS—parallel working

Chapter 1 Why Software Engineering

E: availability and adoption of OO technology
(面向对象技术的出现和采用)—software reuse

F: graphical user interfaces
(使用窗口、图标、菜单和指针的图形用户界面)

friendly face

G: unpredictability of waterfall model of development
[schedule and fee/expense]

(瀑布模型用于软件开发的不可预测性)

瀑布模型沿袭了传统系统工程的“大规模批发制造”的理念，假定生产活动为线性，这与现代软件的生产方式相矛盾

不再是有足够的灵活性和适应性来满足并行开发或并行运行这样的商业软件需求

Chapter 1 Why Software Engineering

④ Examples : (P29)

A:time to market

B:shifts in economics

C:desktop computing

D:networking

E: (OO) technology—provide reuse repositories

F:graphical user interface

⑤ conclusion:

modern SE → requiring a different development model (from “waterfall” model), and try to develop systems in parallel

Chapter 1 Why Software Engineering

2. Wasserman's Discipline of software engineering [SE的Wasserman规范(或基本概念)]

- ① **Abstraction** (英文见课本): 基于某种层次归纳水平的问题描述。它使我们将注意力集中在问题的关键方面而非细节。

某人逛动物园 第一次被狮子咬了

第二次被老虎咬了———猛兽

第三次逛大森林碰见豹子时自己就提醒:这可能是猛兽!

软件高层抽象: 展现对象的团体

A: 程序员的团体;

B: 程序员创建的对象团体。

次高层抽象: 各个单元 (java: package; C++: 名空间)

较低的抽象层次3, 4, 5: 已在OO技术里面介绍过。

(抽象概念新进展: ALPHA狗的抽象推理能力—人工智能)

Chapter 1 Why Software Engineering

In OO method: involve identifying classes of objects and identifying the attributes and relations transformation (P30)

example(P30-31): sensor (simple hierarchy)

A:the items or entity have properties(not idiographic content), the properties have commonality .

B:we can deal with the class or entity, rather than its elements or details (in devising a solution)

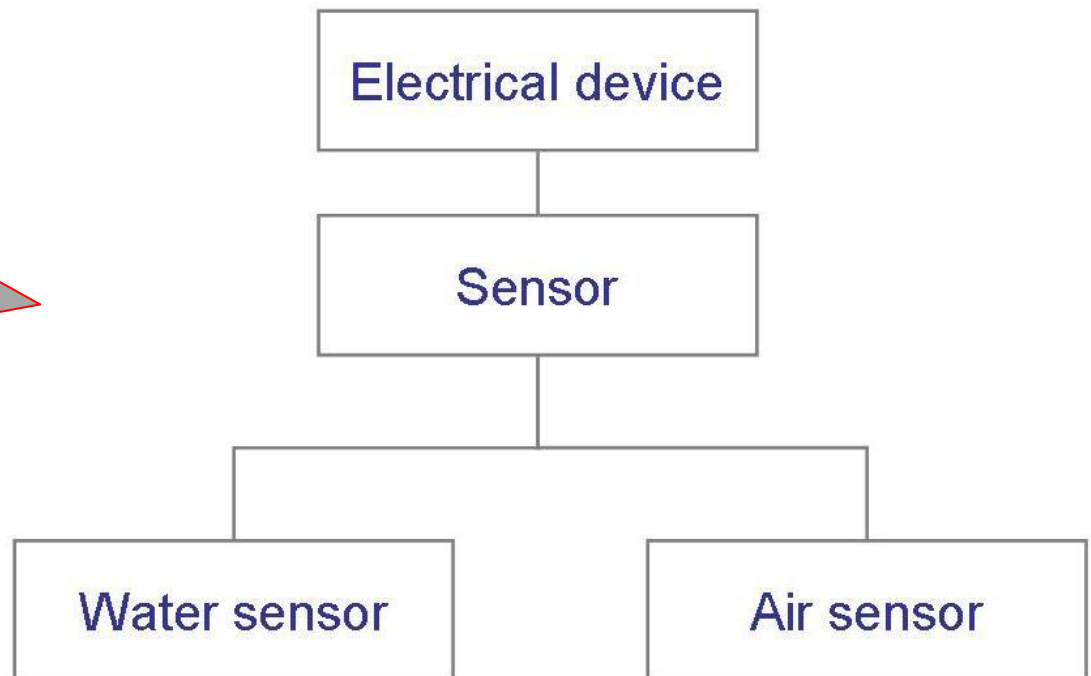
C:we can form the classes and its hierarchies of abstractions in classes

(抽象类的层次)-----see Fig1.13

软件分析与设计要学会抽象,以类为单位考虑问题

- A description of the problem at some level of generalization
 - Hide details

一开始不会如此明确，但因为大型软件设计的功能扩展或预扩展等需要，会逐渐实现这样的抽象



Chapter 1 Why Software Engineering

- 软件分析与设计中关于抽象的广泛讨论与总结：
 - 抽象与转换的区别：初级程序员的软件开发经历（软件不是简单的手工系统到软件的简单对应转换，而是经过抽象的、效率更高的软件，内含了对系统的优化。）
 - 项目早期的软件设计：让某个实体做某件事情。（因为是早期，因而不具体考虑到属性等等细节问题）
 - 抽象可以分成不同的层次，于是软件的设计也能够由粗到细。
 - 软件分析与设计要学会抽象，学会把问题逐步抽象成为类的集合，软件任务就是通过类之间的交互来完成。

Chapter 1 Why Software Engineering

② Analysis and design methods and notations

(分析、设计方法和符号描述系统) (建模原语/表示法)

A: class assignment(班级练习): informal description

B: developing COTS in team work:

-----use standard notation, just like a architect does

(采用标准的符号表示系统)

importance:

X: help communication (利于交流)

**Y: build model and check them for completeness
and consistency(利于建模并检查其完整性和一致性)**

**Z: reuse requirements and design components
(易于对需求和设计部件进行重用)**

C: 软件工程应用符号描述系统: 力图实现分析设计的可视化。

Chapter 1 Why Software Engineering

③ User interface prototyping (用户界面原型化)

A: **prototyping**: 建立系统的小型版, 通常具有有限的关键功能, 以利于用户评价和选择

B: Role(作用) X: identify the key requirement (关键需求)

C: Note(说明)

难以评价不同体系结构的优劣

Y: prototyping → to design a good user interface

④ Software architecture (软件体系结构)

A: definition: 定义一组体系结构单元及其相互关系集来描述软件系统。

B: explain: 一个系统可由不同的体系结构来组成之。

C: importance X: ease(简化) the implementing and testing

Y: improve the effectiveness(效益) of maintenance and changing(变更)

Chapter 1 Why Software Engineering

D: construction: subunits + maps(describe all relations between subunits)

(构成: 一系列子结构单元加上它们彼此的联系集)

focus on: independence (模块独立性)

E: ways of unit partition (单元分解的方法): (1--5 see P32)

X: 基于功能的模块化分解: 基于指派到模块的功能。

Y: 基于数据的分解: 基于外部或通用数据结构。

Z: 面向事件的分解

M: 由外到内的分解

N: 面向对象的设计: 基于标识的对象的类以及它们之间的相互关系。

Chapter 1 Why Software Engineering

⑤ Software process（软件过程）

A: definition: 软件开发活动中的各种组织及规范方法
(organization and discipline in developing)

B: explain

X: 因应用类型和组织文化之间的巨大差异,故难以对软件过程本身进行预先指定,也就是说:使过程本身规范化是不可能的.软件过程不可能以抽象和模块化的方式作为软件工程的基础(Wasserman 1996--see P33)

Y: small/medium-sized system

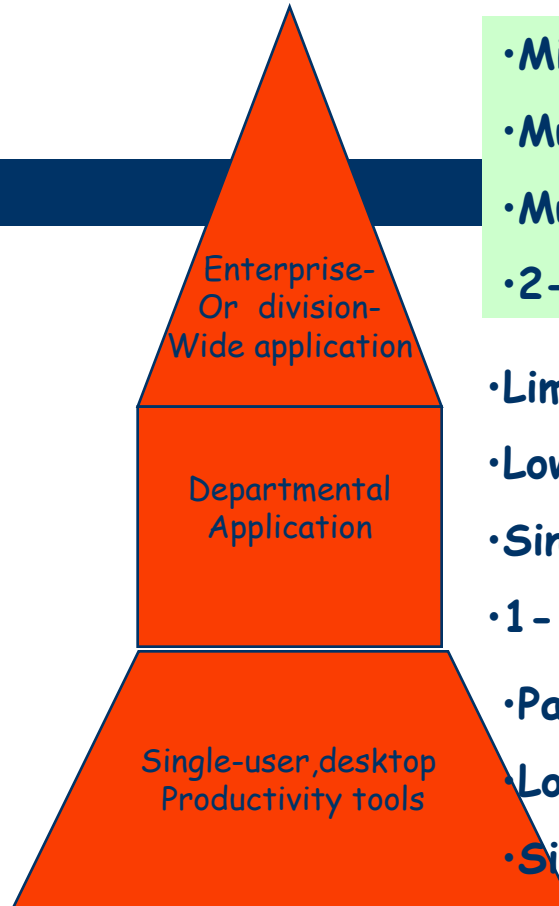
----rapid developing method / tool. Low risk management cost.

Z: large/complex system

----normal developing method , large amount of control. Project managing, configure manage, test tools, review, etc. .

**Controlled
development**

**Rapid
application
development**



- Mission-critical
- Multiuser
- Multiplatform
- 2- to 3-tier development

- Limited scope/vision
- Low/medium risk
- Single/multiplatform
- 1- to 2-tier development
- Packages/minimal development
- Low cost/low risk
- Single platform

Fig 1.14 Differences in development

Chapter 1 Why Software Engineering

⑥ Reuse (重用)

A:definition: commonality → reusable items → new project

重复采用以前开发的软件系统中具有共性的部件, 用到新的开发项目中去. (注: 这里的重用绝不仅仅是源代码的重用)

B: several barriers in reuse(1,2,3,4,5,6 see P34)

⑦ Measurement (度量/测度) (通用的评价方法和体系)

A:definition:quantitative way(approach) → describe process,resource,methods → improving quality

B:importance (用类数学的语言来描述和评价行动与结果)

empirical,practical model → mathematical model

(base on elements and relations)

c: 两个方面: 量化描述系统 + 量化审核系统。

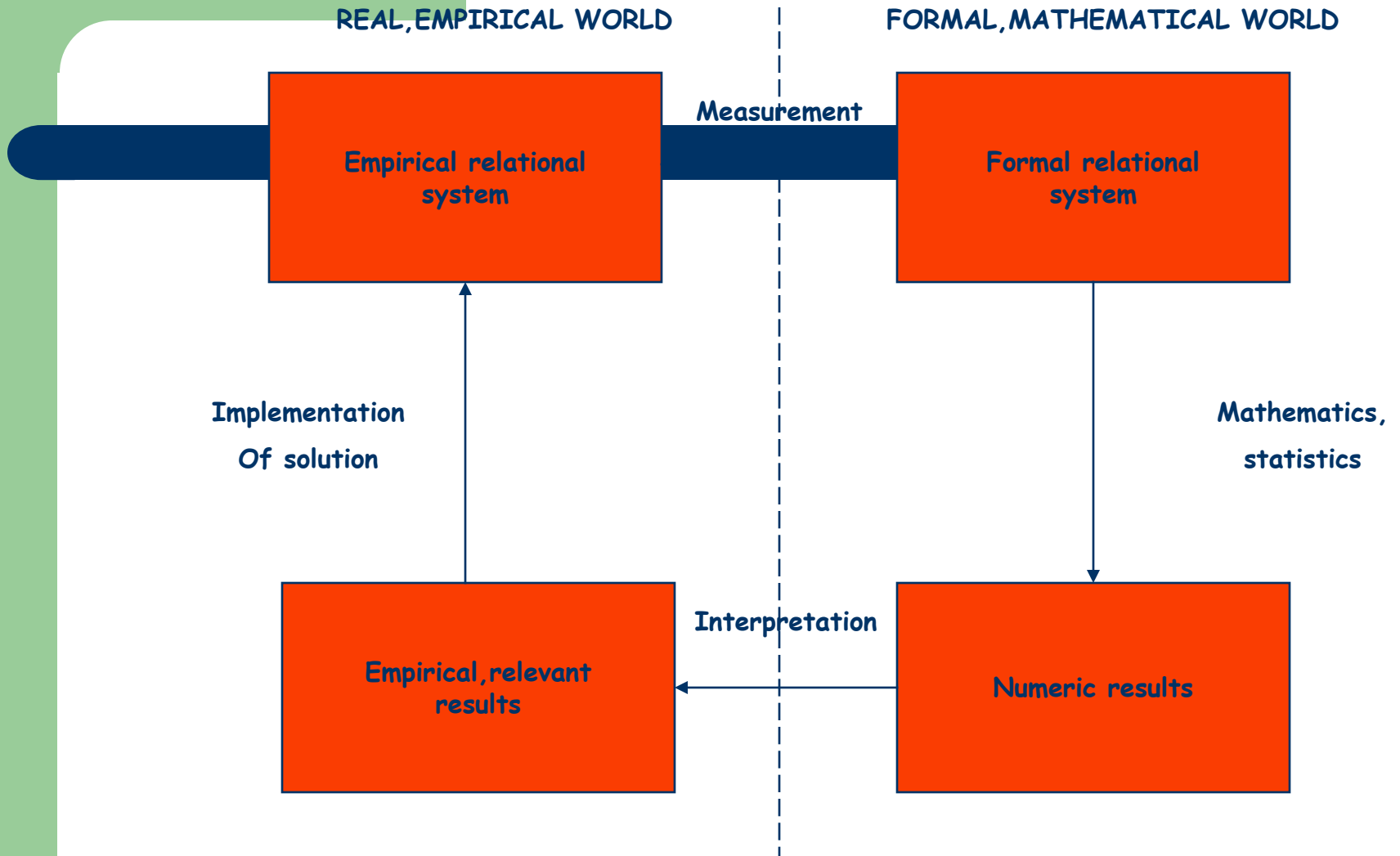


Fig 1.15 Using measurement to help find a solution

Chapter 1 Why Software Engineering

⑧ Tools and integrated environments

A: introduction（开发框架或工具的选择）

frameworks \Rightarrow **compare environments** \Rightarrow **decide which is the best**

B: tool integration

unifying development life cycle \Rightarrow **difficult**

integrating particular activities \Rightarrow **practical**

C: wasserman's five issues in tool integration(P35)

⑨ Conclusion

eight concepts \Rightarrow **making SE as a scientific discipline**

Chapter 1 Why Software Engineering

1.9 Information System Example(信息管理系统实例)

The advertising time selling system of Piccadilly

Television (Piccadilly 电视台广告时段销售信息系统)

1. Government's management (政府管理办法)

problem {
A: franchise (广播特权的发放)
B: content (keep balance in it)(广播内容的规定)
C: restrictions and rules(广播节目种类、时段以及广告的约束与限制规定)

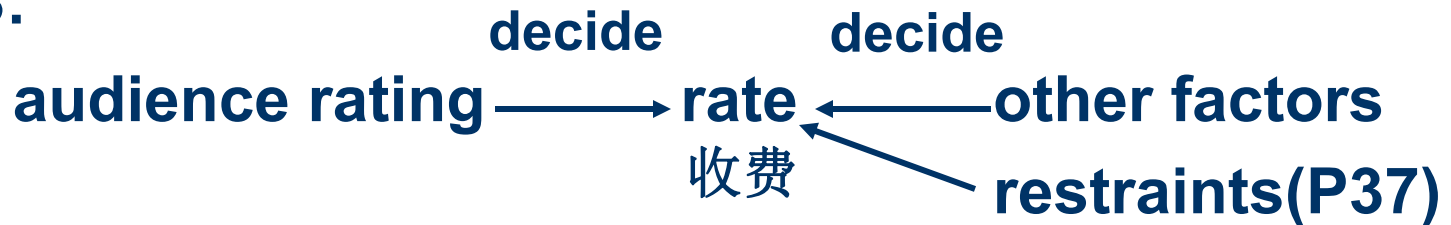
└─ the problem is both interesting and difficult

2. Background Analysis

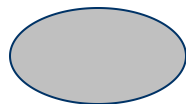
A: Piccadilly Television → high audience rating

Chapter 1 Why Software Engineering

B:



C: system content diagram(P37---fig1.17)

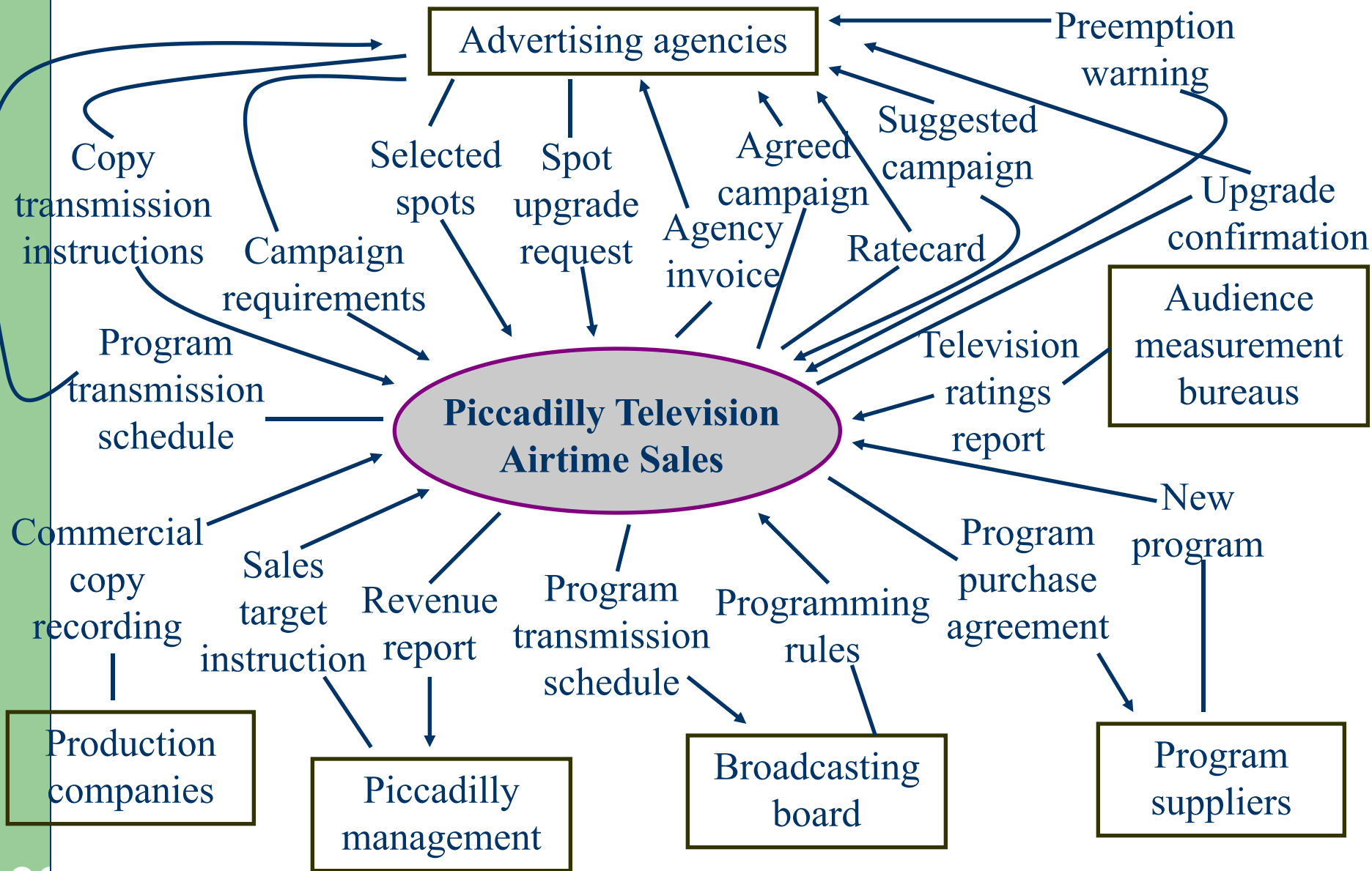


: the Piccadilly system and its boundary

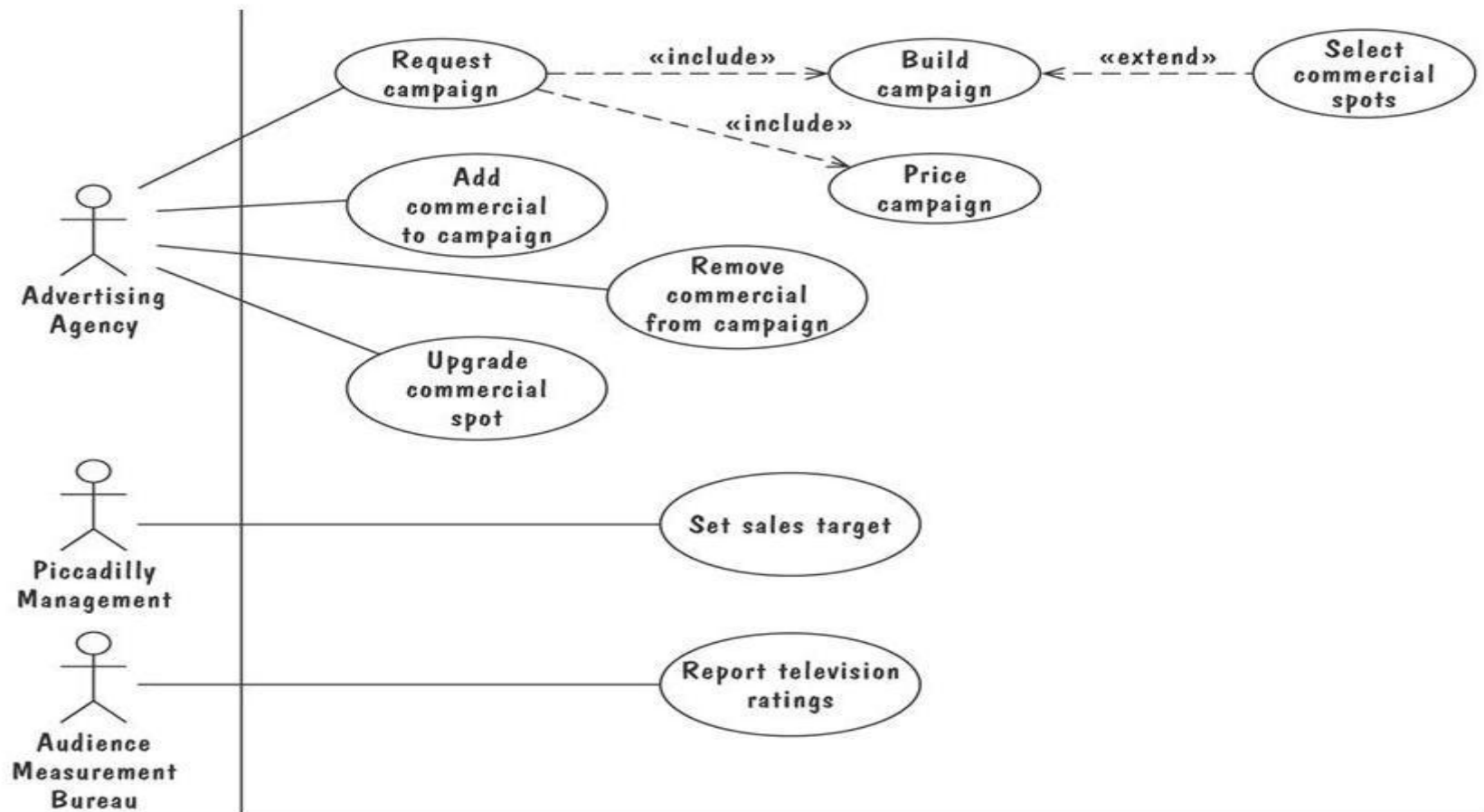


} : the other items that can affect the working of the Piccadilly system

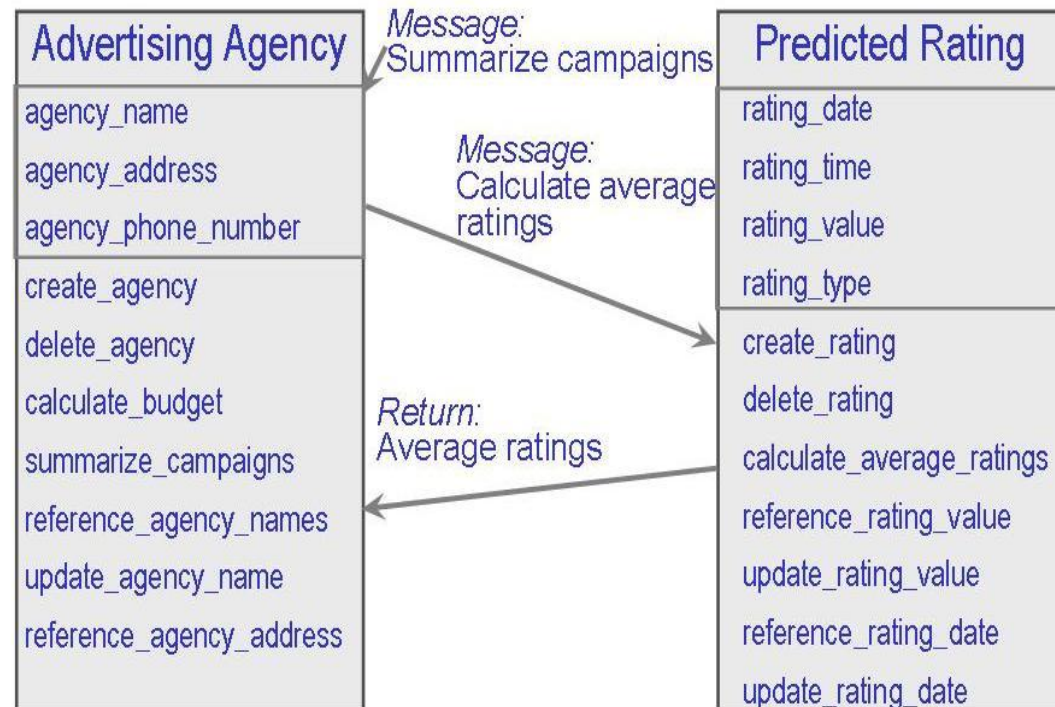
D: In later chapters, the activities and elements inside the shaded oval will be visible.



- High-level diagram captures the essential functionality
 - Shows nothing about the ways in which each of these use cases might succeed or fail



- **Domain elements and relationships that the Piccadilly database will maintain**
- **A closer examination will reveal that there are considerable commonality**



Chapter 1 Why Software Engineering

1.10 Real-Time Example (实时控制系统实例)

1. Incident and Cause (loss: \$500 million)

inquiry board → result: software failure (P38-s2,s3)

2. Ariane Series's history

segment(段)

A:Ariane-4 rocket's success

B:Ariane-5's goal (carry heavier payload into orbit)

C:real cause:the customer misspecify the requirement

D:inquiry board's investigation

software in SRI → incident (cause: wrong design to
exception(意外))

conclusion: software should be assumed to be faulty

Chapter 1 Why Software Engineering

E: Detailed investigation will be done in later chapters(以后自学或上网查资料)

F: ①system engineering's view { analyzing
synthesis



Software Engineering

好消息：北美前景最好的**25种职业**！

-----人家的今天，就是我们的明天！

我们生活在一个科技时代，未来的职业趋向可以充分说明这一点。美国劳动局最近的调查表明，四分之一的新职业集中在计算机、医疗保健及私营教育行业。

到**2017年**，北美最热门、需求量最大的职业当数以下**25种**，其中有些要求大学学历，有些必须具备相关工作经验：**(以下年薪指税后美元额度)**

Software Engineering

1. 计算机软件工程师(Computer Software Engineers) ——
应用软件工程师:年薪\$74,980 ;系统软件工程师:\$79,740

计算机应用软件工程师的工作是根据用户需求,设计、维护计算机应用软件,他与系统软件工程师一起维护公司计算机系统。

2. 侦探(Detectives)——年薪\$53,990

搜集事实、罪证。

3. 家庭保健护理(Home Health Aides)——\$18,330

此项工作负责每天为用户做卫生保健,如:洗澡、修饰,一般只需短期培训或在工作中培训,未来几年这一职业的需求量会很大,唯一的缺点是薪资太低。

4. 程序员(Programmers)——\$65,910

根据公司需要,计算机程序员负责编写、测试程序。

Software Engineering

5. 药剂师 (Pharmacists)——\$84, 900

医师或保健人员开具药品处方后，由药剂师分发，并告知病人如何服用。

6. 安保顾问 (Security Consultants)——约\$45,000 (不能提供准确详细的薪资)

提供个人、公司、组织或团体人身安全、资产保护及专业安保等服务，包括保护客户免遭盗贼、计算机黑客或恐怖分子袭击。

7. 专业猎头 (Employment, Recruitment and Placement Specialists)——\$41, 190

专门为各行业的客户提供中高级管理人才、专业技术人才及特殊岗位的招聘和相关猎头咨询服务，为企业提供免费招聘网络平台等。

Software Engineering

8. 社会服务助理 (Social and Human Service Assistants)——\$24, 270

替卫生保健人员及其客户提供咨询服务，尤其是为社区、个人提供精神支持、安慰，处理日常事务，这项工作需要在职培训，特别适合喜欢处理一般公共事务的人。

9. 计算机系统分析员——\$96, 460

帮助解决计算机问题，开发新系统。但由于编程语言以及编程技术的不断更新，为此用人单位越发注重系统分析师个人对新知识的接受能力和熟练掌握程度。

10. 社会工作者 (Social Workers)——\$39, 440

帮助个人改进社会关系、居住环境、处理个人及家庭问题。

Software Engineering

课堂作业：

- 1、请谈一下关于软件工程的整体认识和印象。
- 2、系统分析师和系统构架设计师有何区别？
- 3、应用软件工程（实施分阶段原理等）会增加系统工作量吗？

来自教材：练习题 2，练习题 3。

你指导**24**年前的腾讯公司是什么样吗？



最初办公地点：深圳华强北赛格科技园



腾讯90年代的格子间



中午吃快餐



Software Engineering
