

# L1

## Professional software development

- Software engineering
  - SE is concerned with theories, methods, and tools for professional software development
  - SE is concerned with cost-effective software development
- What is software?
  - *computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.*
- What are the attributes of good software?
  - *good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable*
- What is software engineering?
  - *SE is an **engineering discipline** that is concerned with all aspects of software production*
    - **engineering discipline**: using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints
    - **all aspects** : Not just technical process of development. Also project management and the development tools, methods etc. to support
- What are the fundamental software engineering activities?
  - *software specification明确需求, software development设计编程, software validation检查满足需求 and software evolution满足变化*
- What is the difference between software engineering and computer science?
  - *computer science focuses on theory and fundamentals; SE is concerned with the practicalities of developing and delivering useful software*
- What is the difference between software engineering and system engineering?
  - *SE is concerned with all aspects of computer-based systems development including hardware, software and process engineering. SE is part of this more general process*
- What are the key challenges facing software engineering?

- *Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software*
- What are the cost of software engineering?
  - ■ Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.\*
- What are the best SE techniques and methods?
  - *While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of systems. e.g. games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't say one method is better than another*
- What differences has the web mad to SE
  - *The web has led to the availability of software services and the possibility of developing highly distributed service based systems. Web-based systems development has led to important advances in programming languages and software reuse.*
- Customized products
  - *e.g. embedded control systems, air traffic control software, traffic monitoring system*
- Essential attributes of good software
  - **Maintainability:** *evolve to meet the changing needs*
  - **Dependability and security:** *should not cause physical or economic damage in the event system failure. Malicious users should not be able to access or damage system*
  - **Efficiency:** *not make wasteful use of system resources such as memory, processor cycles*
  - **Acceptability:** \* must be acceptable to the type of users for wich it is designed. understandable, usable, compatible.\*
- Importance of SE
  - need to produce reliable and trustworthy systems economically and quickly
  - cheaper in the long run. For most types of systems, the majority of costs are changing the software after it has gone into use.
- General issues that affect software

- Heterogeneity 要求系统as distributed systems在不同设备下运行
- Business and social change: 所以需要快速开发新软件
- Security and trust:
- Scale: 很广的应用范围
- Application types
  - stand-alone application
  - interactive transaction-based application 远程访问
  - embedded control system 嵌入式控制系统 硬件
  - batch processing system 批处理系统
  - entertainment systems 个人娱乐
  - system for modeling and simulation 仿真建模
- .....
- SE methods and tools used 取决于:
  - application type; 客户需求; 研发团队背景
- SE fundamental
  - using a managed and understood development process. 不同类型软件使用不同过程。
  - dependability和performance都很重要
  - 明白需求
  - reuse 软件比写新软件更合适

## Software engineering ethics

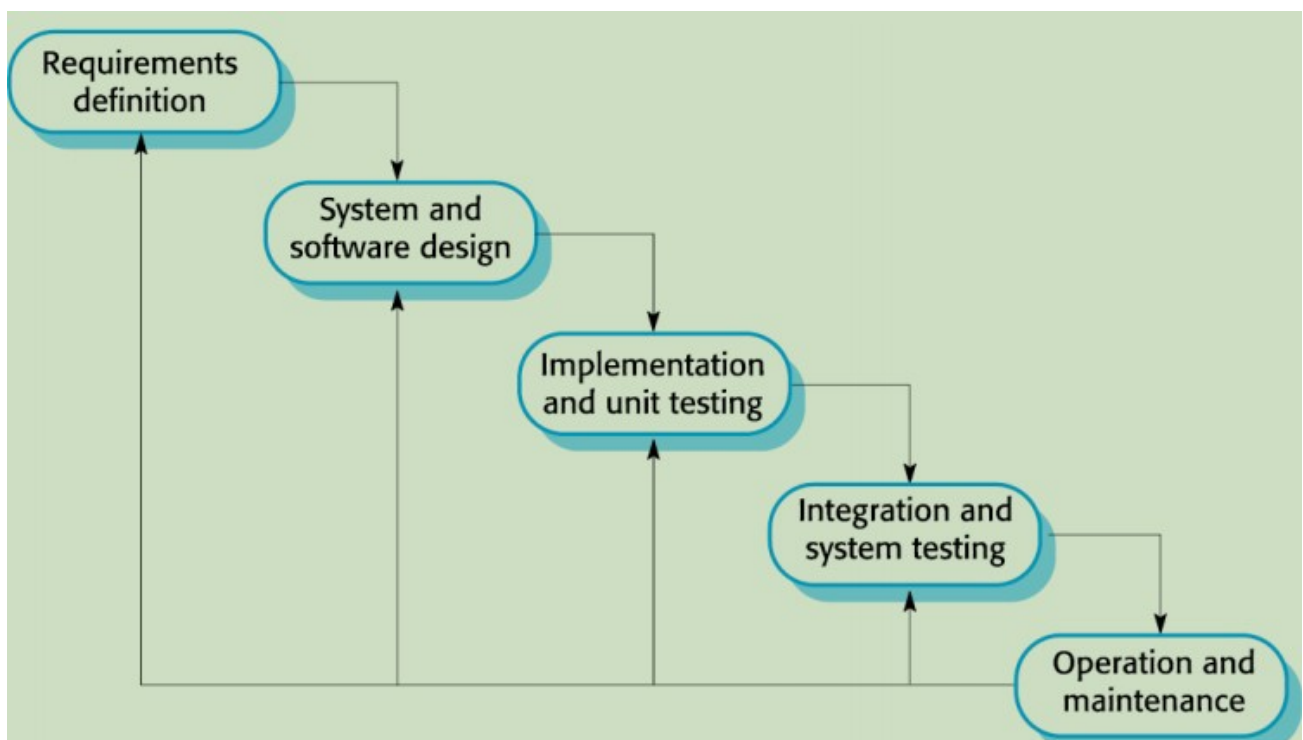
- issues of professional responsibility
  - confidentiality 尊重雇主/客户的机密
  - competence 不要接受超出能力的工作
  - intellectual property rights 确保雇主和客户的知识产权得到保护
  - computer misuse 不应该用他们的技术滥用别人的电脑

## L2 Software Processes

### Software process models

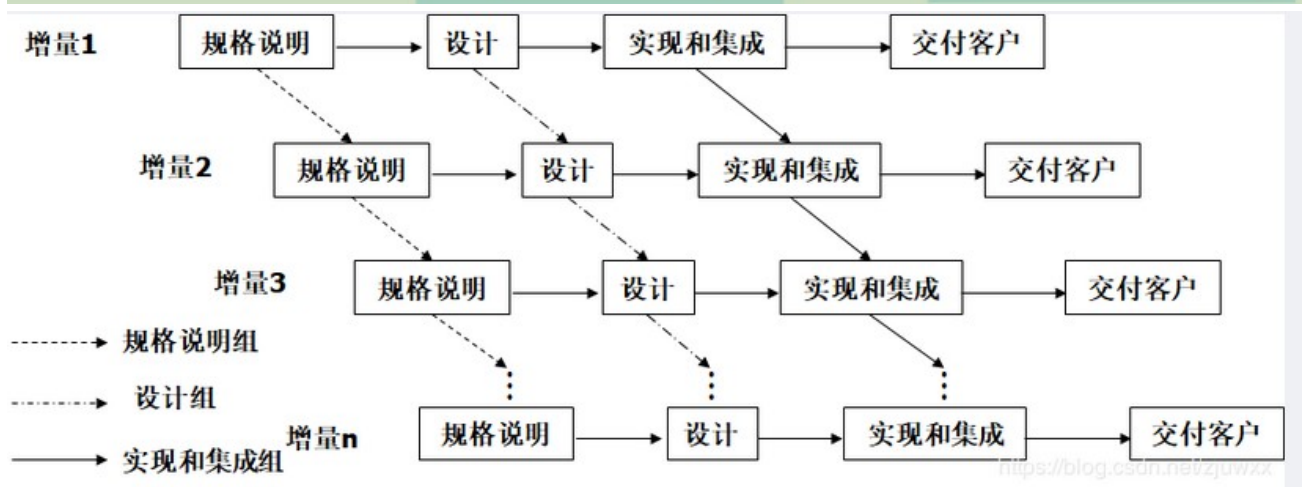
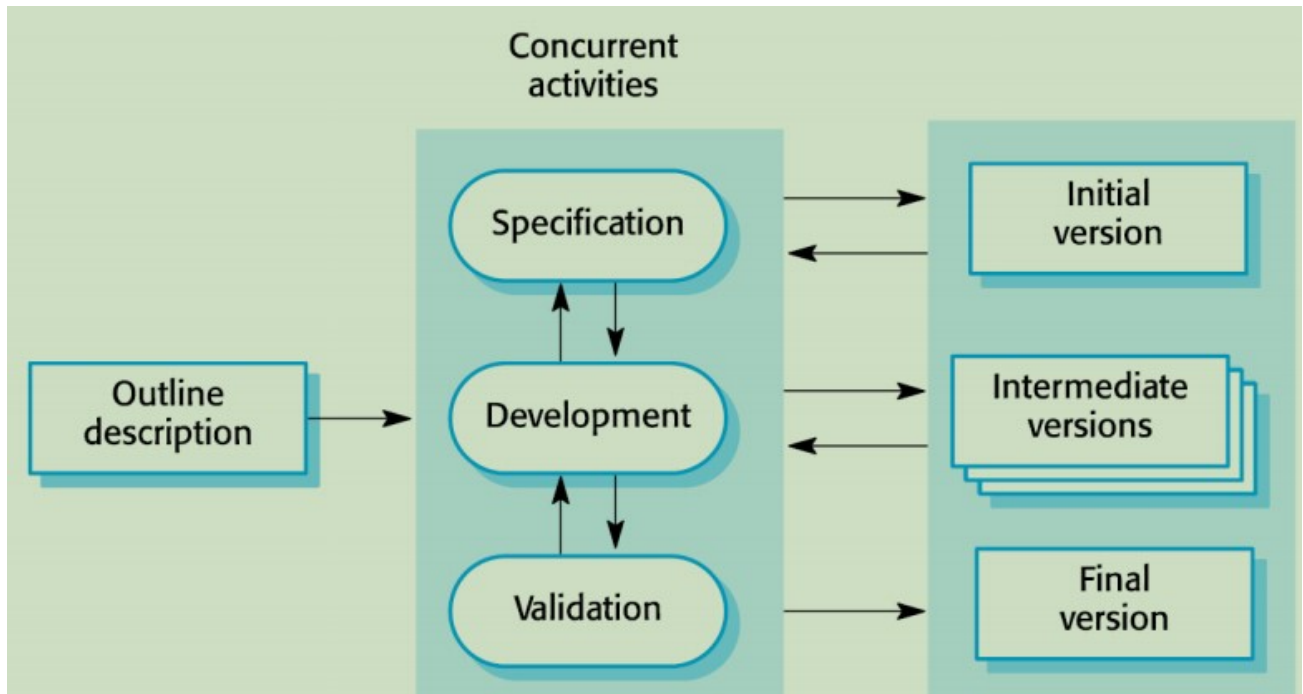
- A set of activities required to develop a software system
  - specification, design and implementation, validation, evolution

- **Software process model**: an abstract representation of a process
- **plan-driven processes**
  - *processes where all of the process activities are planned in advance and process is measured against this plan*
- **Agile process**
  - *planning is incremental and it is easier to change the process to reflect changing customer requirements*
- **The waterfall model**
  - **plan-driven model** : separate and distinct phases of specification and development 规范和开发不同的阶段
  - drawback:
    - difficulty of accommodating change after process is underway
    - 理论上, a phase has to be complete before moving onto the next
    - 只适用于非常了解需求, 并且变化有限; 主要应用于大型工程项目 where a system is developed at several sites



- **Incremental development**
  - **May be plan-driven or agile**: specification, development and validation are interleaved
  - 使用增量模型时, 第一个增量构件往往实现软件基本需求, 提供最核心的功能。把软件产品分解为许多增量构件, 约束条件: 把新构建集成到现有构件中时, 所形成的产品必须是可测试的

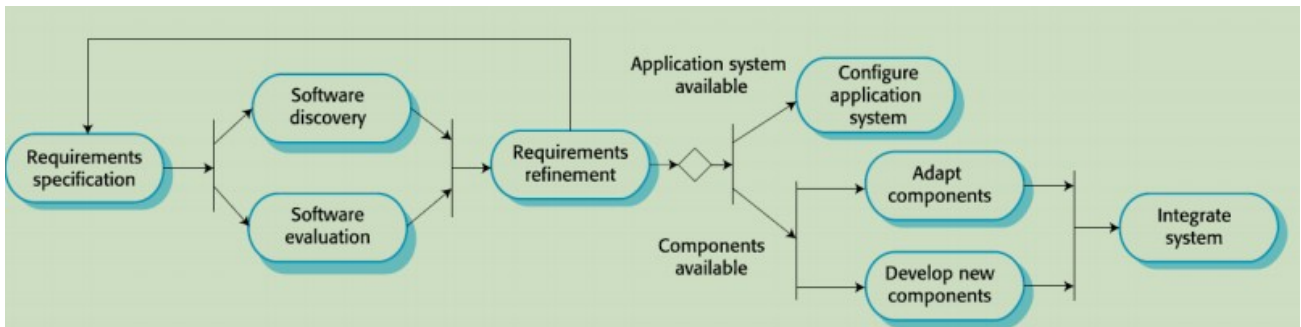
- **benefits:** 适应需求改变的成本减少；容易得到顾客对已完成部分的反馈；可以更快速地交付和部署有用的软件。
- **problems:** 容易退化成边做边改模型，从而软件过程的控制失去整体性；if systems are developed quickly, 每个版本的文档not cost-effective; difficult to incorporating; may corrupt its structure



## • Integration and configuration

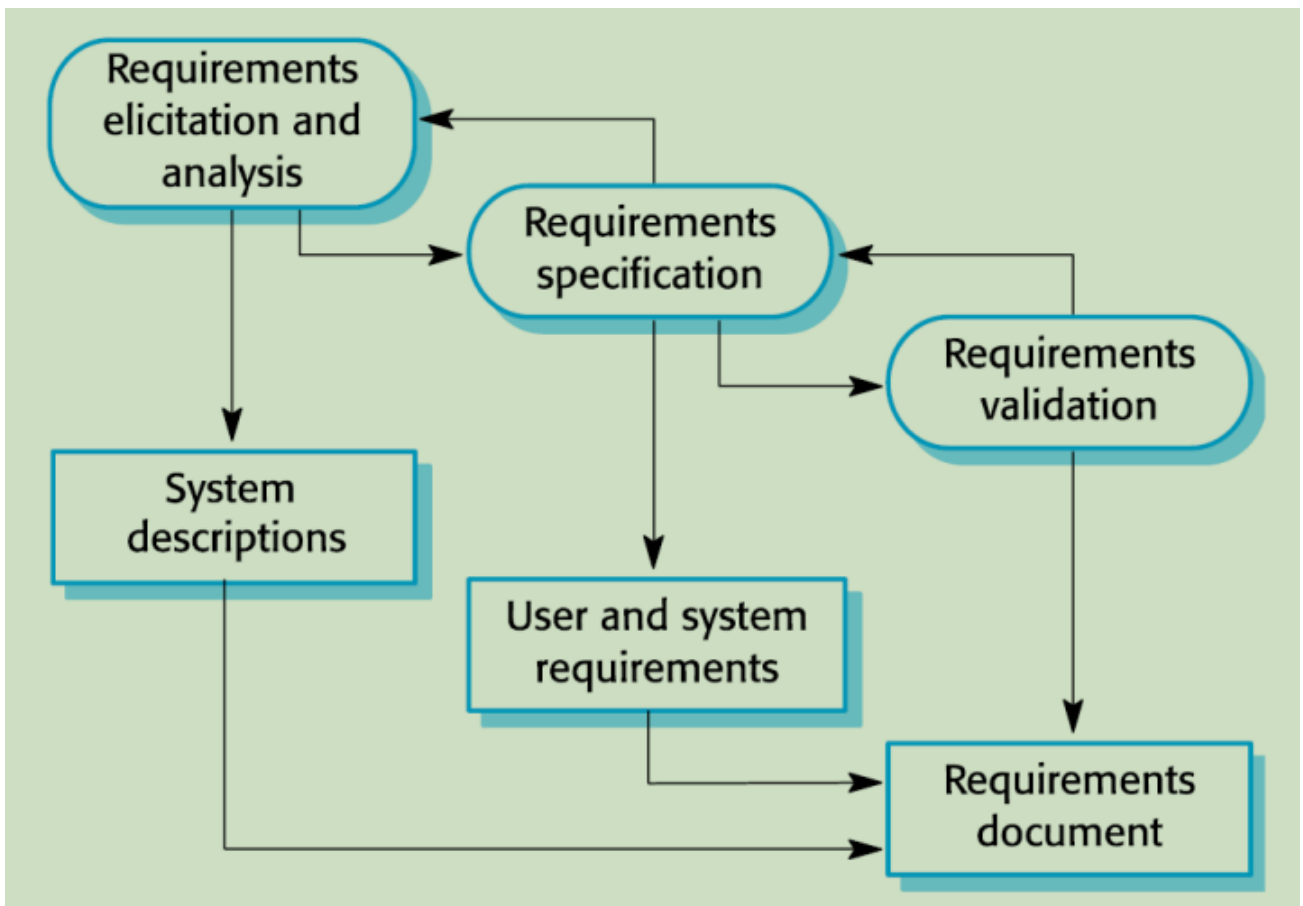
- **May be plan-driven or agile:** the system is assembled from existing configurable components. 由现有的配置组件构成
- key process stages:
  - requirements specification需求分析 → software discovery and evaluation软件发现和评估 → requirements refinement需求细化 → application system configuration应用系统配置 → component adaptation and integration组件适应和整合

- advantage & disadvantage: 降低成本和风险; 更快交付和部署; 需求妥协不可避免 requirements compromises are inevitable; loss of control over evolution of reused system elements
- Reuse-oriented SE



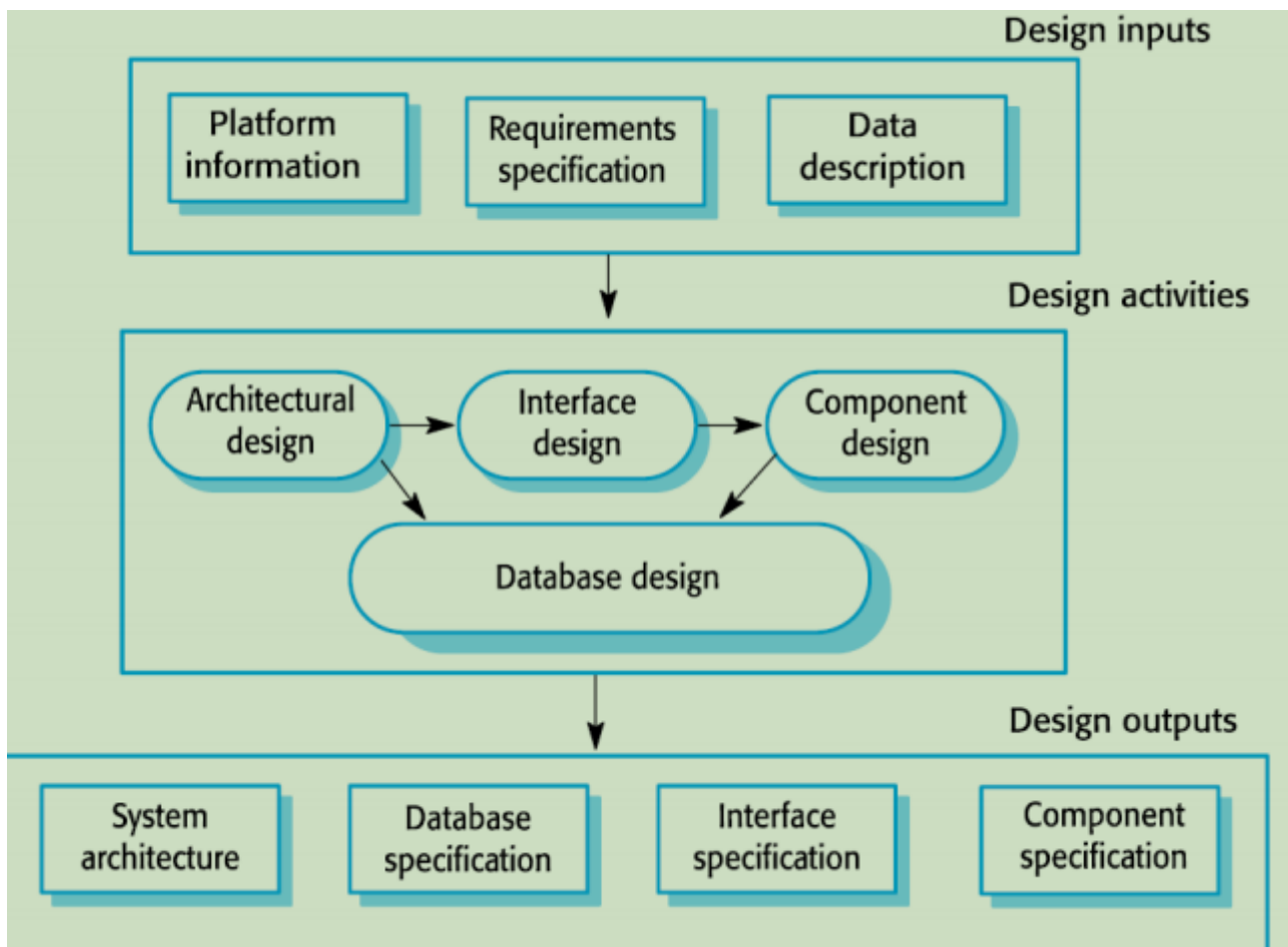
## Process activities

- **basic process activities:**
  - *specification, development, validation, evolution*
  - *organized differently in different development processes*
- **Requirements engineering process**
  - the process of establishing *the required services* and *the constraints* on the system's operation and development
  - *Requirements analysis*: what do the system stakeholders require or expect from the system?
  - *Requirements specification*: define the requirements in detail
  - *Requirements validation*: checking the validity of the requirements
  - *software design*: design a software structure that realises the specification
  - *implementation*: translate this structure into an executable program
  - 设计和实现经常交替 inter-leaved



## general design process model

- design activities
  - Architectural design: *identify the overall structure of the system, the principal components, their relationships and how they are distributed*
  - Database design: *design the system data structures and representation in a database*
  - interface design
  - component selection and design: *search for reusable components. If unavailable, design it.*



## SW validation

- **Verification and validation**

- show that a system conforms to its specification and meets the requirements of the system customer.
- checking and review process and system testing.

- **system testing**

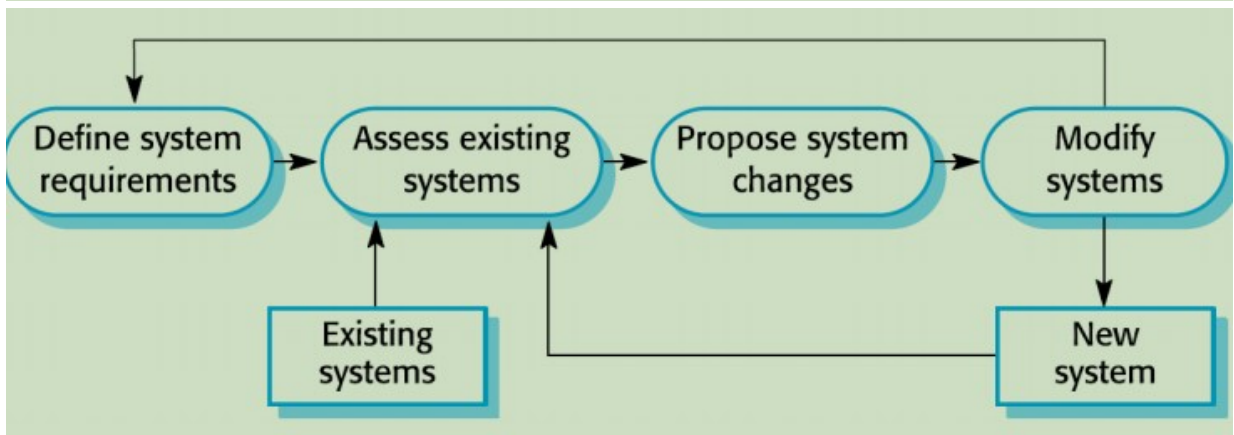
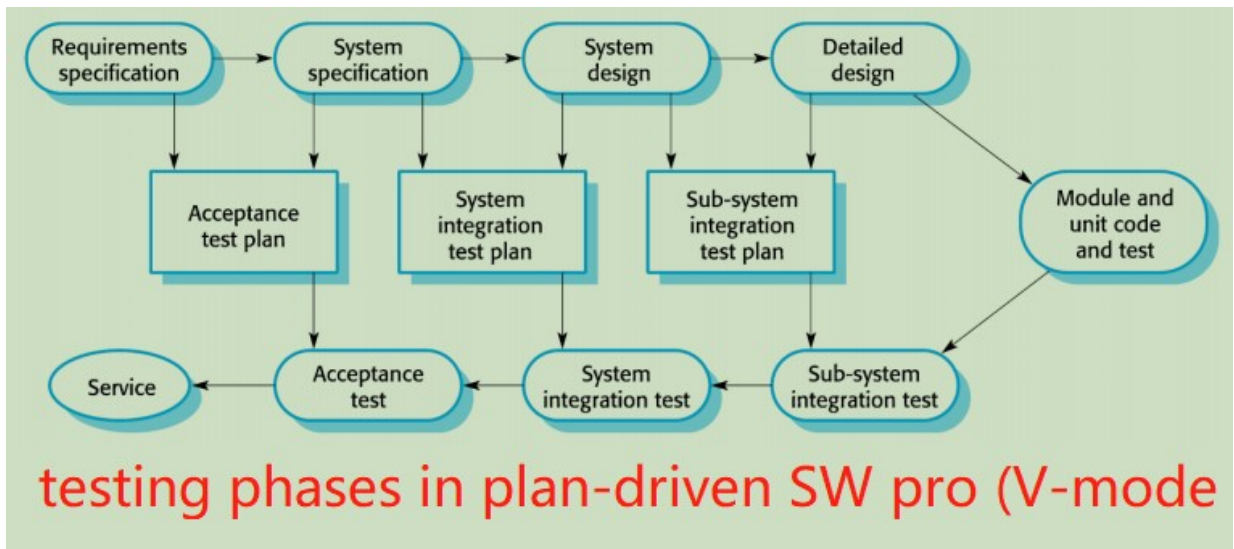
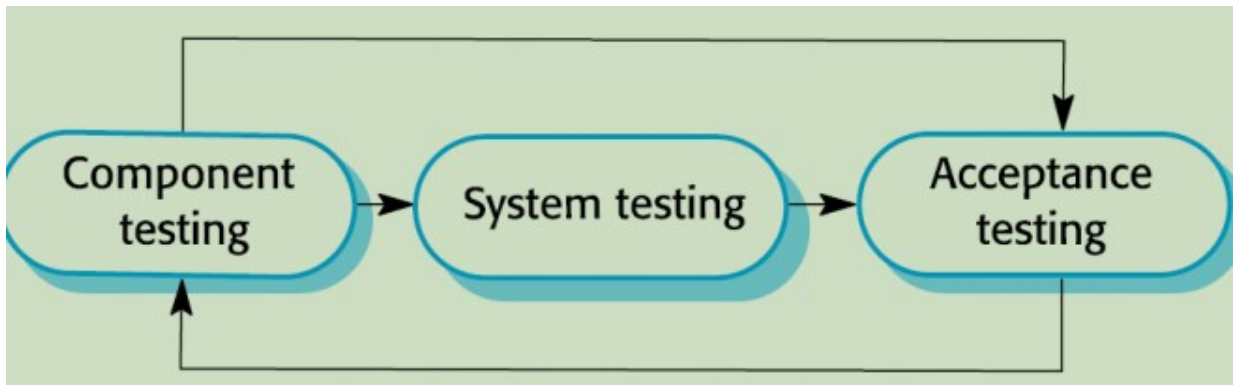
- executing the system with test cases derived from specification of the real data to be processed by the system

## Testing

- component testing: 独立测试各个部件
- system testing: 系统作为整体测试



- customer testing: 使用客户数据测试, 检查是否满足需求



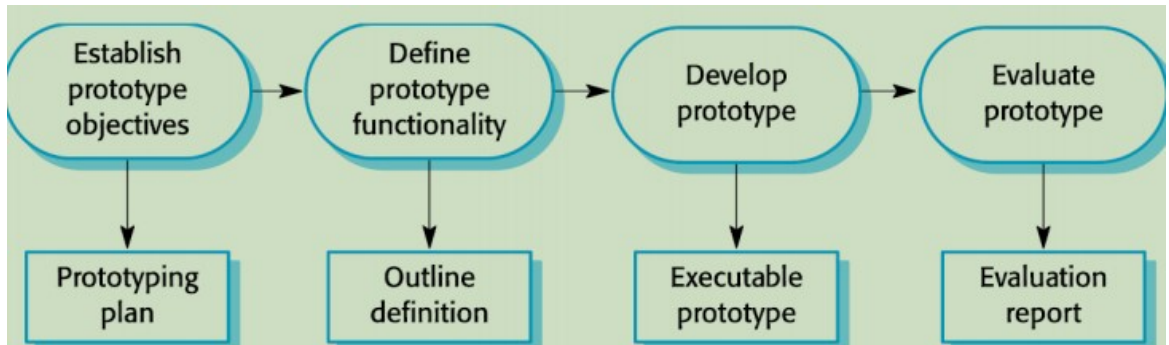
## Copint with change

- Change is inevitable** in all large SW projects
- the costs of change include
  - rework*. reducing its cost:
    - 改变预期anticipation; 改变容忍度tolerance
  - costs of implementing new functionality*

## SW prototyping

- an initial version of a system used to demonstrate concepts and try out design options*

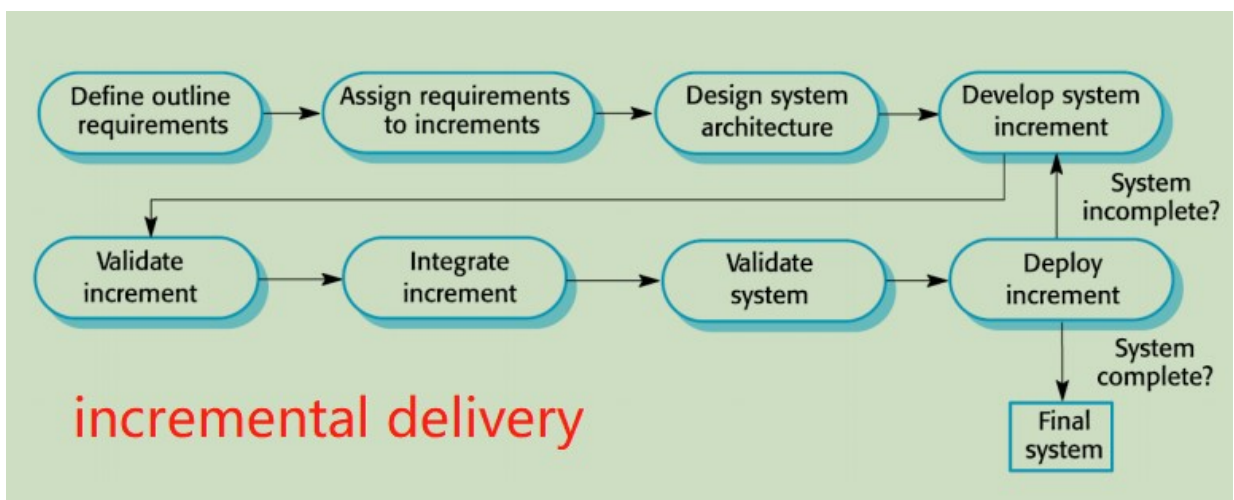
- can be used in:
  - requirements engineering process;
  - design process: explore options and develop a UI design;
  - testing process: run back-to-back design
- Benefits:
  - 提高usability; closer match用户需求; 提高设计质量; 提高maintainability; 减少开发工作



- prototype should be **discarded after development 被丢弃** as they are not a good basis for a production system

## Incremental delivery

- advantages
  - 顾客更早收到交付; 早期的增量作为原型引出后面的增量; 低风险失败; 优先级大的可以先做
- problem



## Process improvement

- process maturity approach 成熟度: 反映良好技术和管理实践
- agile approach: 关注在迭代开发中的reduction

## process improvement cycle

- 1process measurement: 度量属性, 这些度量形成基线, 决定过程改进是否有效。
- 2process analysis: 评估当前流程, 确定流程弱点和瓶颈bottlenecks
- 3process change: 过程变更可以解决一些已确定的过程弱点
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$