Choice Questions

- **software is**:

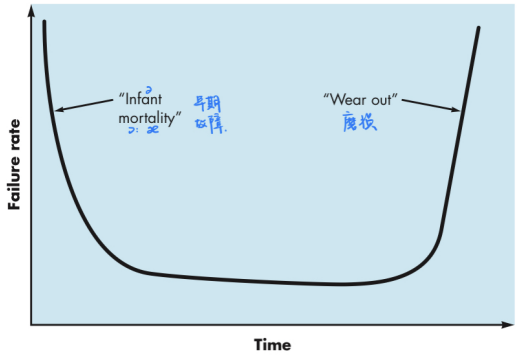
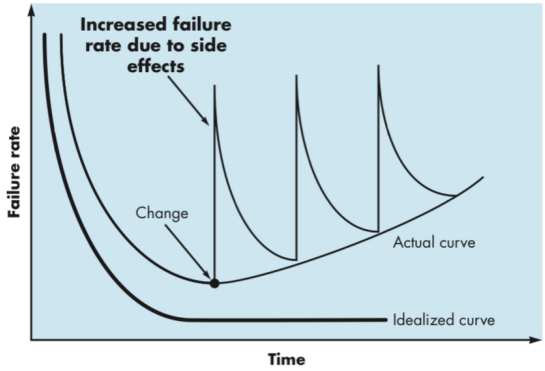
  (1) instructions (computer programs) that when executed provide desired features,function and performance;

  (2) data structure that enable the programs to adequately manipulate information;

  (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.

- **failure curve** for hardware: bathtub curve

  failure curve for software: idealized cure & actual curve

hardware	software
	

- 7 categories of present continuing **challenges** for software engineers:

  (1) system software

  (2) application software

  (3) engineering/scientific software

  (4) embedded software (嵌入式软件)

  (5) product-line software

  (6) web/mobile applications

  (7) artificial intelligence software

- **Clouding Computing**:

  与传统的服务器相比，云平台可以将物理资源虚拟化为虚拟机资源池，灵活调用软硬件资源，实现对用户的按需访.

  - Iaas:能够按需向用户提供的计算能力、存储能力或网络能力等IT基础设施类服务
  - Paas:通过互联网为用户提供一整套开发、运行和运营应用软件的支撑平台
  - Saas:一种通过互联网提供软件服务的软件应用模式

- legacy systems evolve(逐渐形成) reasons:

  (1) environments or technology

  (2) new business requirements

  (3) other more modern systems or database

  (4) evolving computing environment (可运行)

- **Software Engineering** :(1) The application of a systematic(系统的),disciplined(有条理的), quantifiable(可量化的) approach to the development,operation and maintenance of software; that is, the application of engineering to software(将工程应用于软件).(2) The study of approaches as in (1).研究1的方法

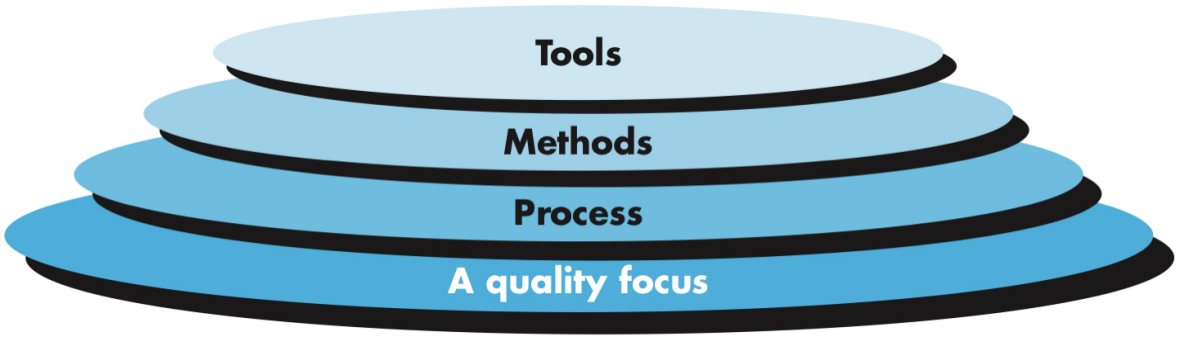
- The bedrock(基石) that supports software engineering is a *quality focus*.

  The foundation for software engineering is the *process* layer.

  Software engineering *methods* provides the technical how-to's for building software.

  Software engineering *tools* provide automated or semi-automated support for the process and the methods.

- Software engineering layers:



- A process is a collection of activities, actions, and tasks that are performed when some work product is to be created.

- A generic **process** framework for software engineering encompasses(包括) five **activities**:
    - Communication.
    - Planning. (framework-activities-actions-tasks)
    - Modeling.
    - Construction. (coding and testing)
    - Deployment. (delivering and feedback)

- **Umbrella activities** :

  (1) software project tracking and control

  (2) risk management

  (3) software quality assurance 软件质量保证

  (4) technical reviews 技术评审

  (5) measurement 测量

  (6) software configuration management 软件配置管理

  (7) reusability management 可重用性管理

  (8) work product preparation and production

- Differences among processes adopted for projects:

  (1) Overall flow of activities, actions, and tasks and the interdependencies among them.

  activities, actions, tasks的总体流程以及它们之间的相互依赖关系。

  (2) Degree(程度) to which actions and tasks are defined within each framework activity.

  (3) Degree to which work products are identified and required.

  (4) Manner in which quality assurance activities are applied.

  (5) Manner in which project tracking and control activities are applied.

  (6) Overall degree of detail and rigor with which the process is described.

  (7) Degree to which the customer and other stakeholders are involved with the project.

  (8) Level of autonomy given to the software team.

  (9) Degree to which team organization and roles are prescribed.

- the essence(本质) of software engineering practice:
  (1) Understand the problem (communication and planning).
  (2) Plan a solution (analysis and design modeling).
  (3) Carry out the plan (code generation).
  (4) Examine the result for accuracy (testing and quality assurance).

- 7 general principles of se:
  - the reason it all exists
  - KISS(Keep it Simple, stupid!)
  - Maintain the Vision
  - What you Produce, others will consume
  - Be open to the future
  - Plan ahead for reuse
  - Think

- **Process flow**(figure 3.2): linear process flow, iterative ~, evolutionary ~, parallel ~

- 6 distinct actions for **communication** activity:
  inception, elicitation, elaboration, negotiation, specification, validation
  开始、启发、完善、协商、规范、确认

- template for describing a **process pattern**:
  - pattern name
  - forces *适用的environment*
  - type(stage pattern - *establishing communication, incorporate requirements gathering* / task pattern - *requirements gathering* / phase pattern - *spiral螺旋型的 model, prototyping*)
  - initial context *描述模式应用条件*
  - problem
  - solution
  - resulting context
  - related patterns
  - known uses and example

- risk management 3个阶段:
  1. risk identification 风险识别
  2. risk mitigation 风险缓解
  3. risk tracking 风险追踪

- 3-year half-life: half of what you need to know today will be obsolete within 3 years

> *Requirement （communication-modeling）*

- Requirement engineering :

1. **inception** 开始
2. **elicitation**(establish business goals) 启发
3. **elaboration**(a refined requirements model,提取分析类) 精化
4. **negotiation** 协商
5. **specification** 规范
6. **validation** 验证，对规约进行测试
7. **management** 管理，对需求进行版本控制、管理

(有的可以并行)

- Before the first meeting, each attendee is asked to make:
    - a list of objects
    - a list of services(processes or functions)
    - a list of constraints 约束(eg. cost,size,business rules)
    - performance and other non-function requirements (eg. speed, accuracy)

- **Requirement model** must achieve 3 primary objectives:
    1. to describe what the customer requires
    2. to establish a basis for the creation of a software design
    3. to define a set of requirements that can be validated once the software is built

- 判断potential class是否纳入分析模型:
    1. retained information
    2. needed services (operations change attributes)
    3. multiple attributes
    4. common attributes
    5. common operations
    6. essential requirements

- defining operations:
    1. manipulate data(add,delete...)
    2. computation
    3. inquire state
    4. monitor an object for the occurrence of a controlling event 监视对象是否发生控制事件

- find potential class:
    1. the preceding list is not all inclusive, additional classes would have to be added to complete the model;
    2. some of the rejected potential classes will become attributes for those classes that were accepted;
    3. different statements of the problem might cause different "accept or reject" decisions to be made.