CPT203重点摘要（个人向）V0.2

Week1：

1，What is a computer software? • ComputerSoftware is a collection of instructions, data, or computer programs that are used to run machines and carry out particular activities. Software products may be developed for a particular customer or may be developed for a general market. 计算机软件是用于运行机器和执行特定活动的指令、数据或计算机进程的集合。软件产品可 能是为特定客户开发的，也可能是为一般市场开发的。

2，What is Software System? 什么是软件系统

• A softwaresystemis a system that consists of a number of separate computer software,configuration files, system documentation and user documentation. (“Wikipedia, Software system”)

软件系统是由许多独立的计算机软件、配置文档、系统文档和用户文档组成的系统。（“维基百科，软件系统”）

3，What is software process

• The systematic approach that is used in software engineering is sometimes called a software process.

• 软件工程中使用的系统方法有时称为软件过程

• A software process is a sequence of activities that leads to the production of a software product.

• 软件过程是导致软件产品生产的一系列活动。

• There are four fundamental activities that are common to all software processes:

• Software specification（规范）

• Software development（开发）

• Software validation（验证）

• Software evolution（演化）

4,Software failures：

• Software failures are a consequence of two factors: 软件故障是由两个因素共同造成的：

• Increasing demands 不断增长的需求

- Low expectations (you don’t expect the software to grow into a giant beast) 期望值低 （你不会指望软件会成长为一头巨大的野兽）

5，Professional software usually has the following properties（专业软件通常具有以下属性）

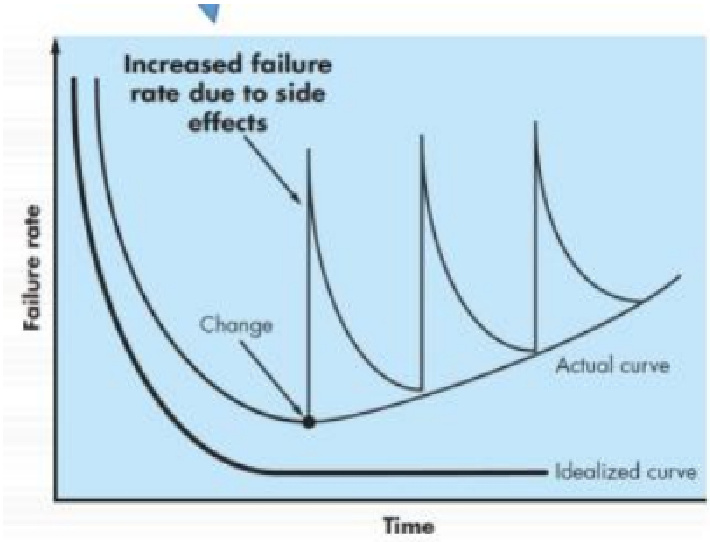
- Strict user requirements 严格的用户要求
- Required accuracy and data integrity 所需的准确性和数据完整性
- Higher security standard 更高的安全标准
- Stable performance for heavy load 重负载性能稳定
- Required technical support, etc. 所需的技术支持等

Product Characteristics	Description
Maintainability	Software should be written in such a way so that it canevolve to meet the changing need of customers. This is a critical attribute because software changing is an inevitable requirement of a changing business environment. 软件的编写方式应使其能够不断发展以满足客户不断变化的需求。这是一个 关键属性，因为软件更改是不断变化的业务环境的必然要求。
Dependability and Security	Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not beable to access or damage the system 软件可靠性包括一系列特性，包括可靠性、安全性和安全性。在系统发生故 障时，可靠的软件不应造成物理或经济损失。恶意用户不应能够访问或损坏 系统
Efficiency	Software should make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness , processing time, memory utilization, etc. 软件应浪费系统资源，例如内存和处理器周期。因此，效率包括 响应性 、 处理时间、内存利用率等。

Product Characteristics	Description
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use. 软件必须为其设计所针对的用户类型所接受。这意味着它必须易于理解、可 用并与他们使用的其他系统兼容

6,deteriorate(恶化)

记住这张图和恶化可能的原因即可



原因：Before the curve can return to the original steady-state failure rate, another change is requested, causing the curve to spike again.

在曲线可以恢复到原始稳态故障率之前，会请求另一次更改，从而导致曲线再次达到峰值。

Week2-3：

1，Design activities:

a) Architectural design

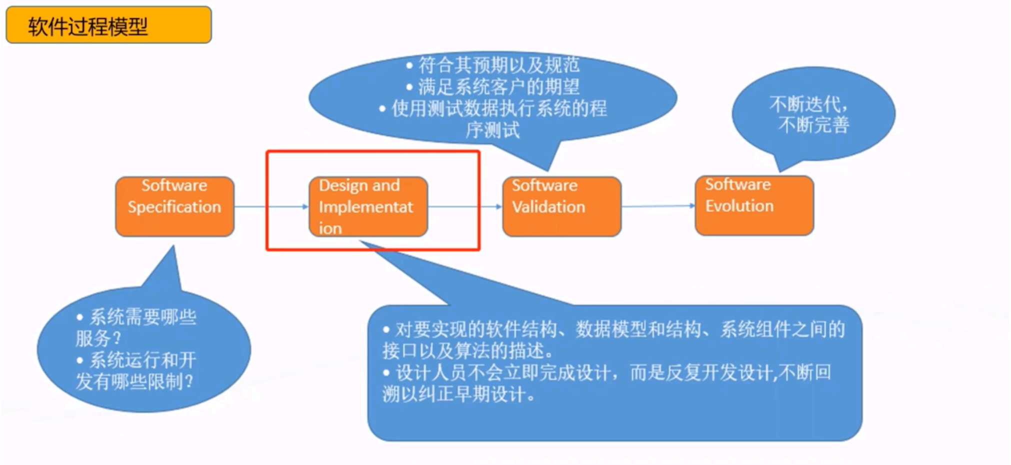
b) Interface design

c) Component design

d) Database design,

2，软件开发过程（牢记2个模型，waterfall，agile）

然后记住这个图


软件过程模型

- 符合其预期以及规范
- 满足系统客户的期望
- 使用测试数据执行系统的程序测试

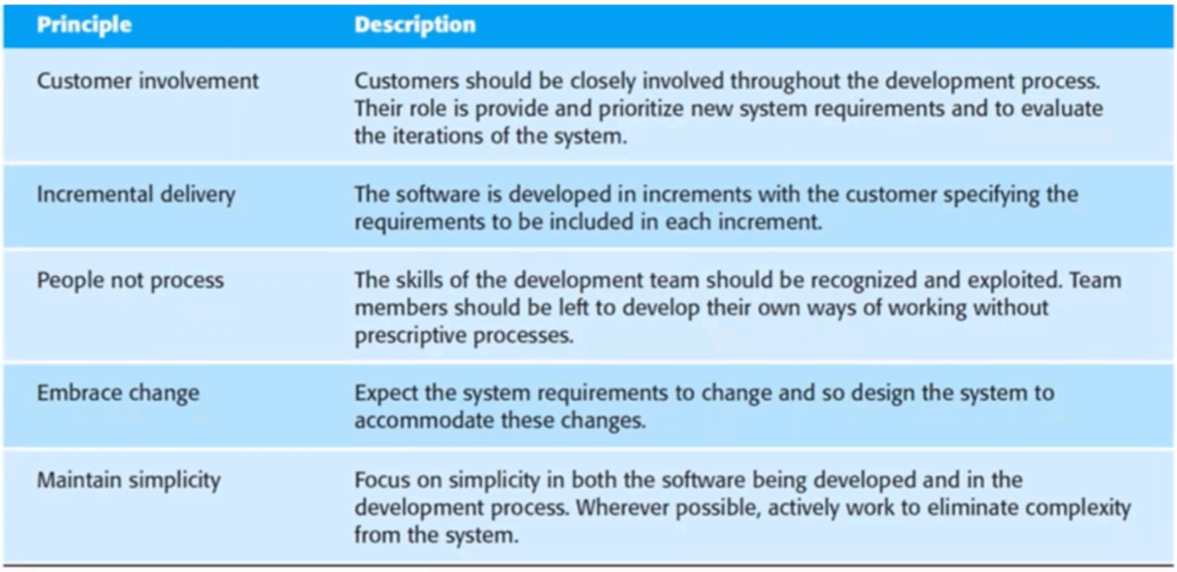
不断迭代，不断完善

- 系统需要哪些服务？
- 系统运行和开发有哪些限制？

- 对要实现的软件结构、数据模型和结构、系统组件之间的接口以及算法的描述。
- 设计人员不会立即完成设计，而是反复开发设计,不断回溯以纠正早期设计。

因为我记得她考过一次软件模型的四个主要组成部分，答案就是这个图。

然后就是记住他们的使用场景，瀑布模型适合需求固定的情况下使用，敏捷开发适合需求多变的情况下使用。此外，敏捷开发还应当记住如下几个原则：



Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

3，敏捷开发存在的问题：

Challenges - In practice, the principles underlying

agile methods are sometimes difficult to realize:

1，Its success depends on having a customer who is willing and able

to spend time with the development team and who can

represent all system stakeholders.

2,Individual team members may not have suitable personalities for

the intense involvemen.

3，Prioritizing changes can be extremely difficult, especially in

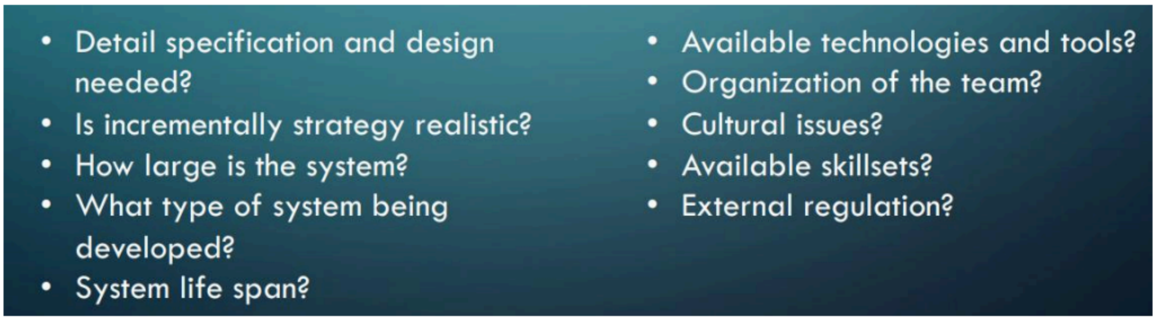
systems for which there are many stakeholders.

4，Maintaining simplicity requires extra work.

5，It is difficult for some organization to accept informal processes

defined by development teams.

4，如何选择 plan-driven 和 agile?



- Detail specification and design needed?
- Is incrementally strategy realistic?
- How large is the system?
- What type of system being developed?
- System life span?
- Available technologies and tools?
- Organization of the team?
- Cultural issues?
- Available skillsets?
- External regulation?

Week4（Scrum）：

暂略，目前没看到考过，也有可能是我题没做完的原因

Week5-6为画图，参考我的绘图课件即可，Week8开始用老师的思维导图应当是更加合适的

Week8（DesignConcept)

优秀的design应该做到如下几点：

1，可读可理解；

2，满足所有可能存在的需求

3，应当有对整个流程完整描述的图像

常见的designconcept：

• Abstraction：

抽象通常使用突出基本功能，隐藏不必要细节来简化系统复杂度，这也是他的定义，牢记他的细节和级别高低负相关，即较低层次上，提供了概念/方法/解决方案的更详细描述，在高层次上他使用问题环境的语言以广义术语陈述概念/方法/解决方案，一个很简单的抽象例子就是：A car -> a Xiaomi Su7 -> a Xiaomi Su7 Ultra

• Modularity：

clusters similar or relative functions together, sets up boundaries and provides interfaces for communication

将相似或相对的功能聚集在一起，建立边界并提供通信接口，我们牢记他的优缺点即可：

- 模块化提高了制造效率并节省了时间
- 模块化允许系统部件的开发彼此独立进行，从而减少了开发时间。
- 但是，系统中过多的模块会增加模块集成的复杂性

- Functional Independence（内聚和耦合，暂略）：
  - Coupling
  - Cohesion
- Object-oriented design

参考OOP即可

Week9(software design):

1，architecture pattern：

MVC pattern：

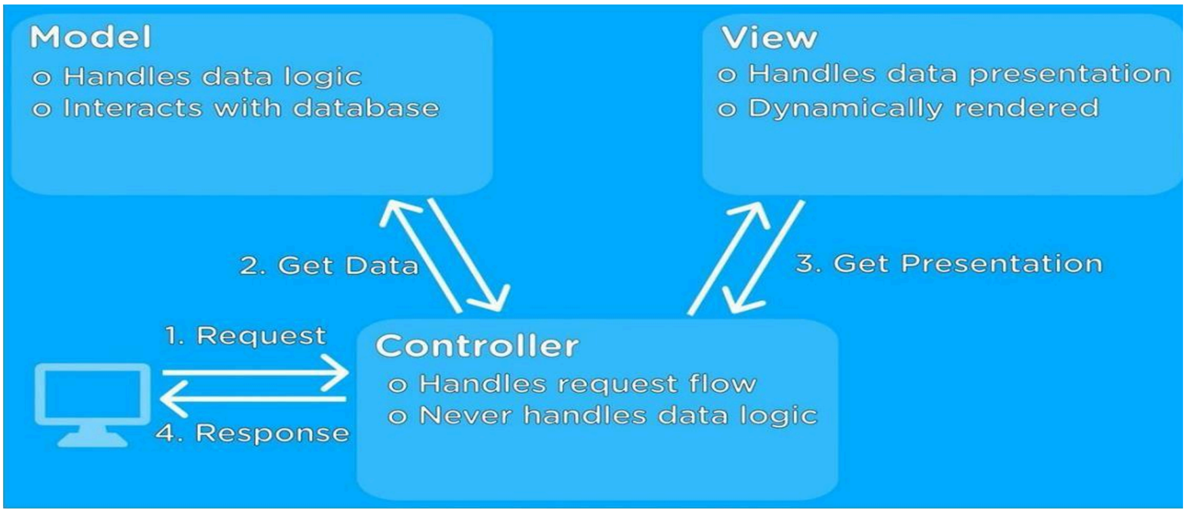
主要包括如下内容：

Model（模型）：直接管理应用程序的数据、逻辑和规则的模式的中心组件

View（视图）：信息的任何输出表示，例如图表或示意图。

Controller（控制器）： 接受输入并将其转换为模型或视图的命令，实现视图与模型之间的互连

值得注意的是只有控制器和主机直接交互



使用场景：有多种方式查看和与数据交互时使用

优点：允许数据独立于表示更改，支持用不同的方式展示相同的数据，在一个表示中的更改可以覆盖到所有表示中

缺点：代码可能更加复杂

The Layered Pattern

优点

- **Advantages**:
  - 只要接口保持不变，可以替换整个层。
  - 每一层都可以提供冗余设施（例如，认证），以增加系统的可靠性。

缺点

- **Disadvantages**:
  - 实际上，提供层之间的清晰分离往往很困难，高层可能需要直接与低层交互，而不是通过其直接下层。
  - 性能可能是一个问题，因为服务请求在每一层处理时会有多级解释

2，userinterface：

What it is：

an effective communication medium between a human and a computer

人与计算机之间的有效通信媒介

Why is it important?：

A poorly designed user interface will force user to commit mistakes. Users can get easily frustrated using a poorly designed interface regardless of computational power or content.

□ 设计不佳的用户界面会迫使用户犯错误。□ 无论计算能力或内容如何，用户都很容易使用设计不佳的界面而感到沮丧。

The golden rules：

Place the user in control

Reduce the user’s memory

Make the interface consistent

Interface Design Issues：

Response time: System response time has 2 important characteristics: length and variability.

Help facilities: Help must be available for all system functions. Include help menus, print documents.

Error handling: describe the problem in a language the user can understand. Never blame the user for the error that occurred.

Application accessibility: especially for the physically challenged.

Internationalization: The Unicode standard has been developed to address the daunting challenge of managing dozens of natural languages with hundred of characters and symbols.