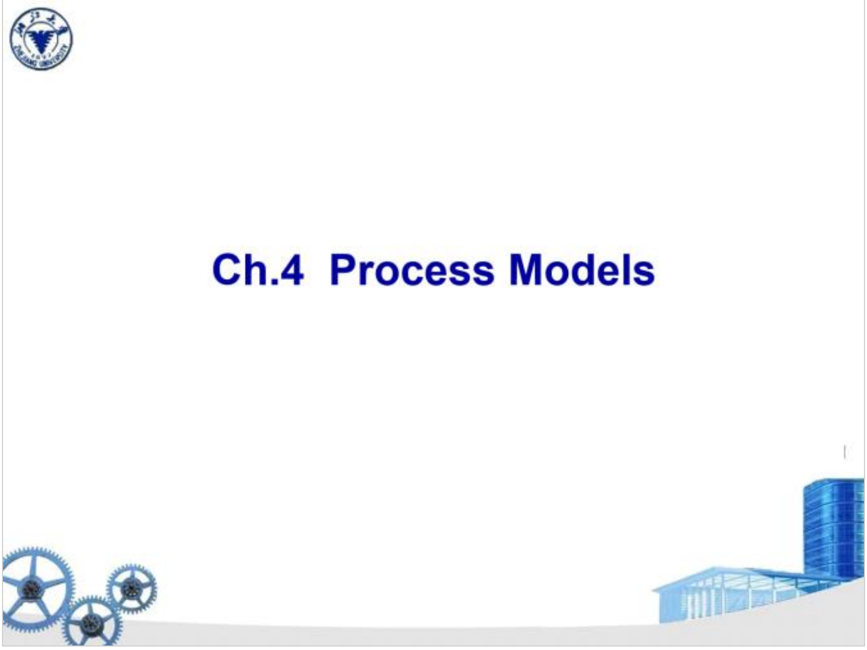
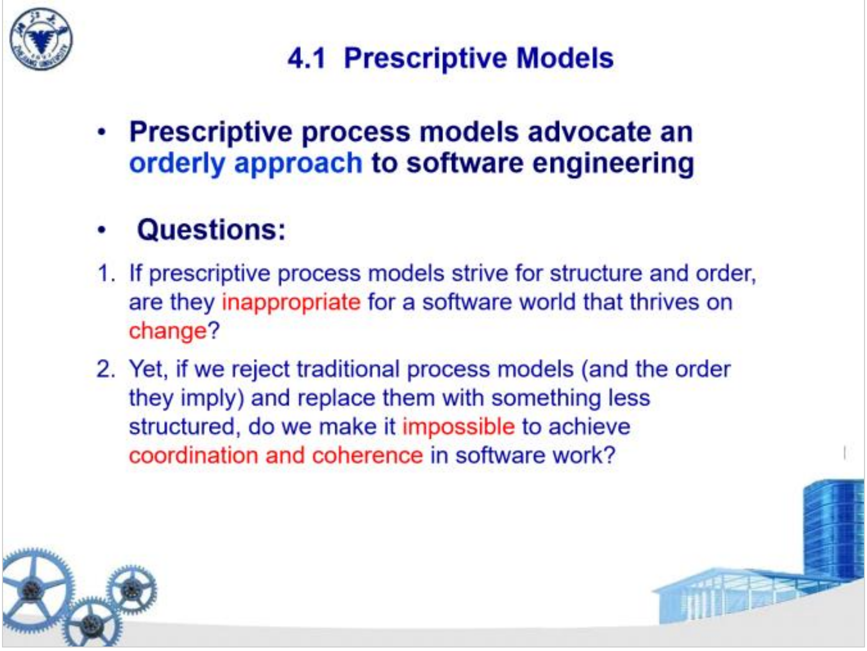
Lecture Note 4

2024年3月8日 10:38



Ch.4 Process Models



4.1 Prescriptive Models

- **Prescriptive process models advocate an orderly approach to software engineering**
- **Questions:**

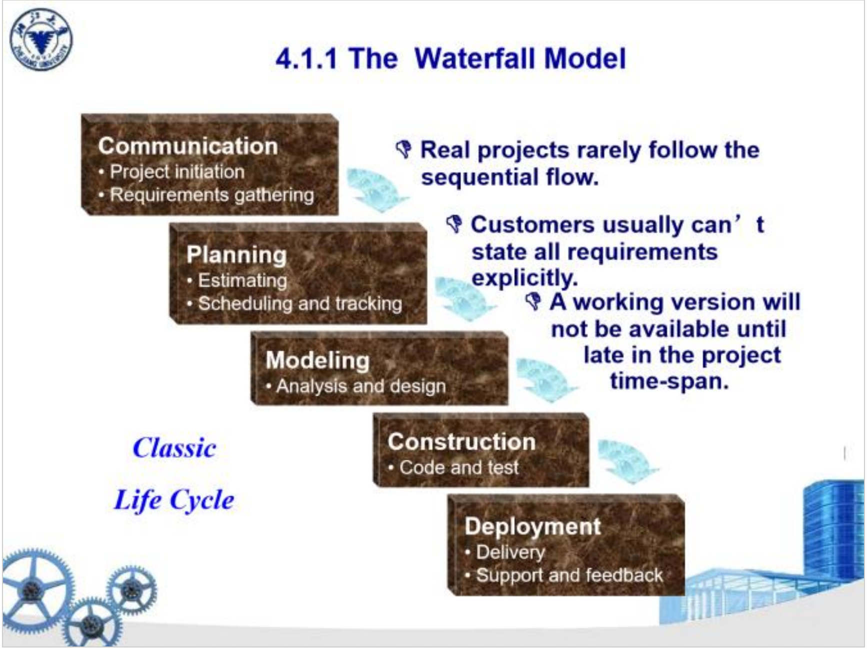
1. If prescriptive process models strive for structure and order, are they inappropriate for a software world that thrives on change?
2. Yet, if we reject traditional process models (and the order they imply) and replace them with something less structured, do we make it impossible to achieve coordination and coherence in software work?

惯例模型：常用过程模型

定义有序化方法

特定流程能否适应变化?

不行的话对过程进行改进?



4.1.1 The Waterfall Model

Communication
- Project initiation
- Requirements gathering

Planning
- Estimating
- Scheduling and tracking

Modeling
- Analysis and design

Construction
- Code and test

Deployment
- Delivery
- Support and feedback

- **Real projects rarely follow the sequential flow.**
- **Customers usually can’t state all requirements explicitly.**
- **A working version will not be available until late in the project time-span.**

Classic Life Cycle

瀑布式模型——线性模型

简单

但是：

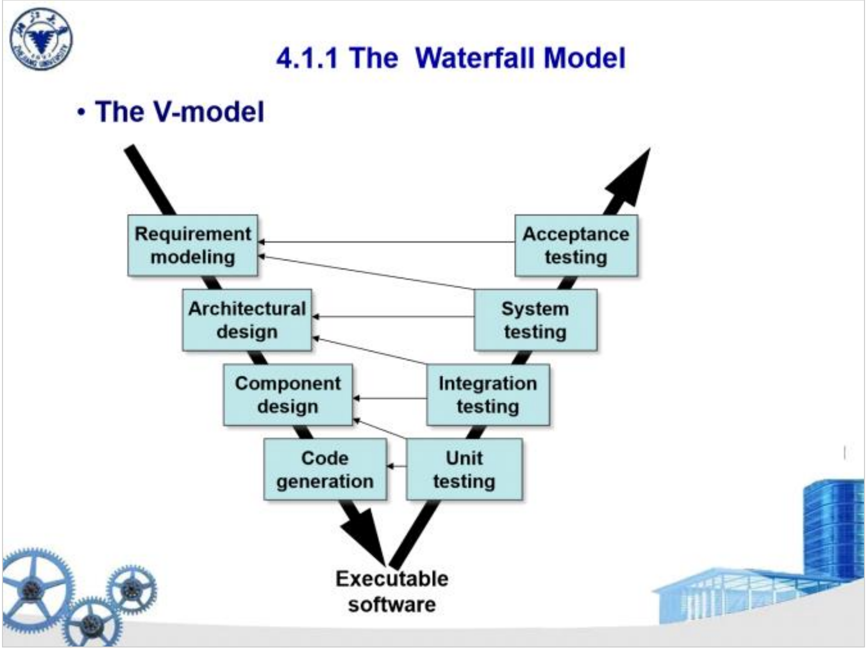
1、现实的过程很少序列化地变化

2、周期太长，用户不能很好地讲清楚自己需要什么

3、反馈不及时，用户到最后才能看到结果

适用于当需求明确、更改较少的情景

Eg. 工程实施类项目



4.1.1 The Waterfall Model

- The V-model

Requirement modeling

Architectural design

Component design

Code generation

Acceptance testing

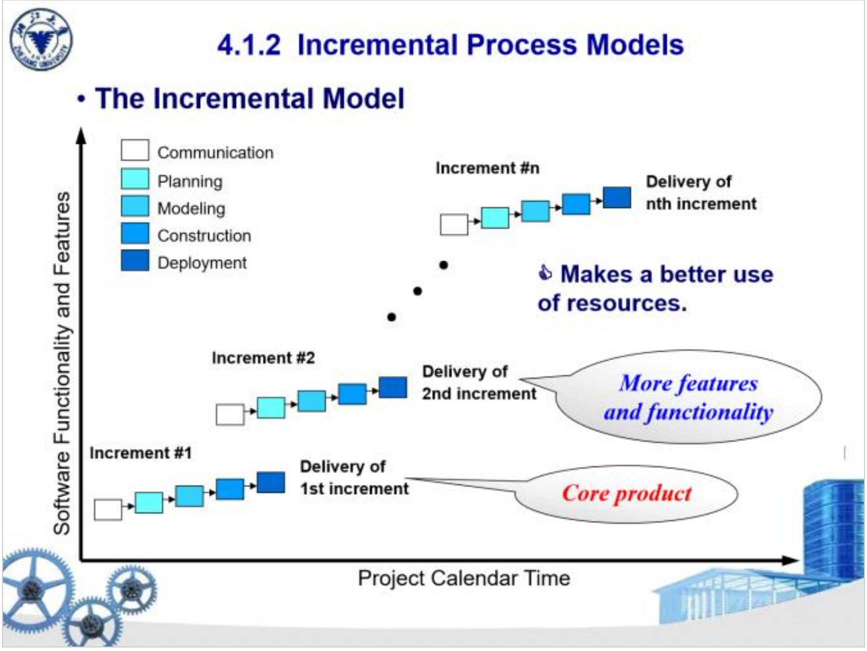
System testing

Integration testing

Unit testing

Executable software

V模型（本质也是瀑布模型）
对testing和design进行细化
每个testing分别对应之前的



4.1.2 Incremental Process Models

- The Incremental Model

□ Communication
□ Planning
□ Modeling
□ Construction
□ Deployment

Software Functionality and Features

Project Calendar Time

Increment #1 — Delivery of 1st increment — *Core product*

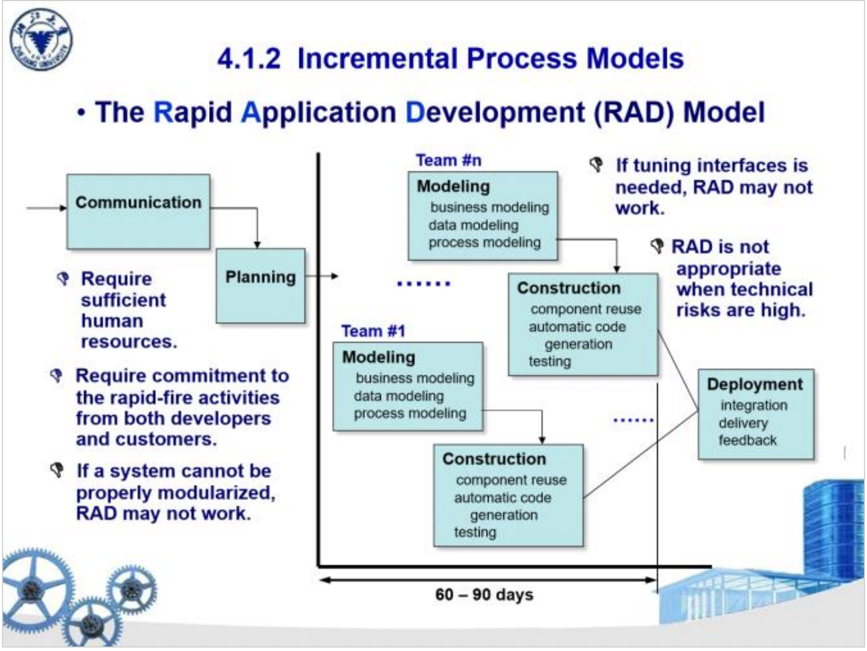
Increment #2 — Delivery of 2nd increment — *More features and functionality*

Increment #n — Delivery of nth increment

Makes a better use of resources.

增量过程模型
时间维和功能维

第一个增量分为五步，先做核心的功能
第二个增量再增加一些功能
第二个子流程的开始时间不一定要等到第一个结束之后
可以应对变化 / 让用户快速看到阶段结果 / 可以利用资源



4.1.2 Incremental Process Models

- The Rapid Application Development (RAD) Model

Communication

Planning

Team #n
Modeling
business modeling
data modeling
process modeling

Construction
component reuse
automatic code generation
testing

Team #1
Modeling
business modeling
data modeling
process modeling

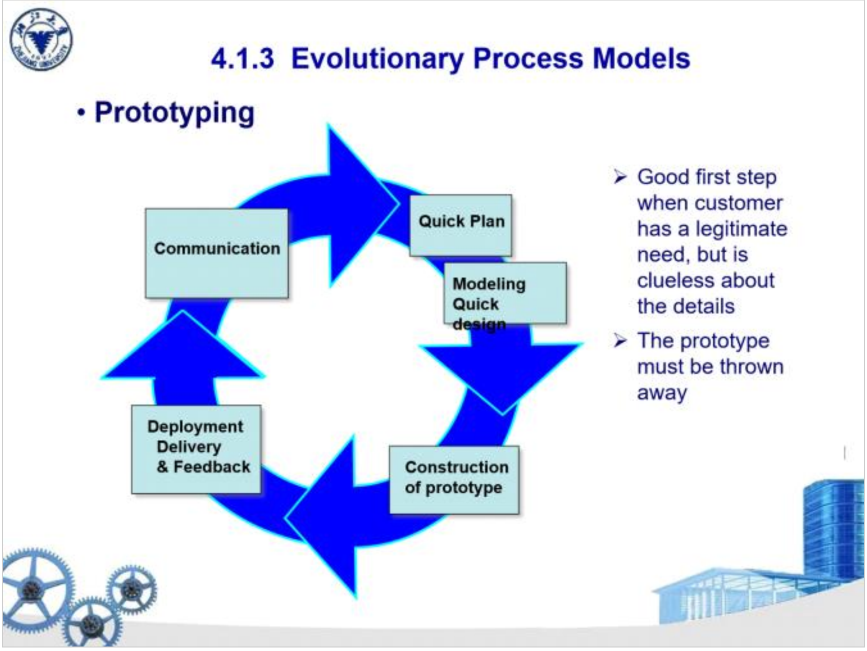
Construction
component reuse
automatic code generation
testing

Deployment
integration
delivery
feedback

60 – 90 days

- Require sufficient human resources.
- Require commitment to the rapid-fire activities from both developers and customers.
- If a system cannot be properly modularized, RAD may not work.
- If tuning interfaces is needed, RAD may not work.
- RAD is not appropriate when technical risks are high.

快速应用开发模型
planning阶段还是瀑布式的
之后分为很多team，可以并行做，最后打包在一起
增量式开发同时引入并行
开发时间短
但必须满足：
1、系统可以很好模块化，不能有太多交互
2、需要足够的开发人员
3、需要比较投入



4.1.3 Evolutionary Process Models

- **Prototyping**

Communication

Quick Plan

Modeling Quick design

Construction of prototype

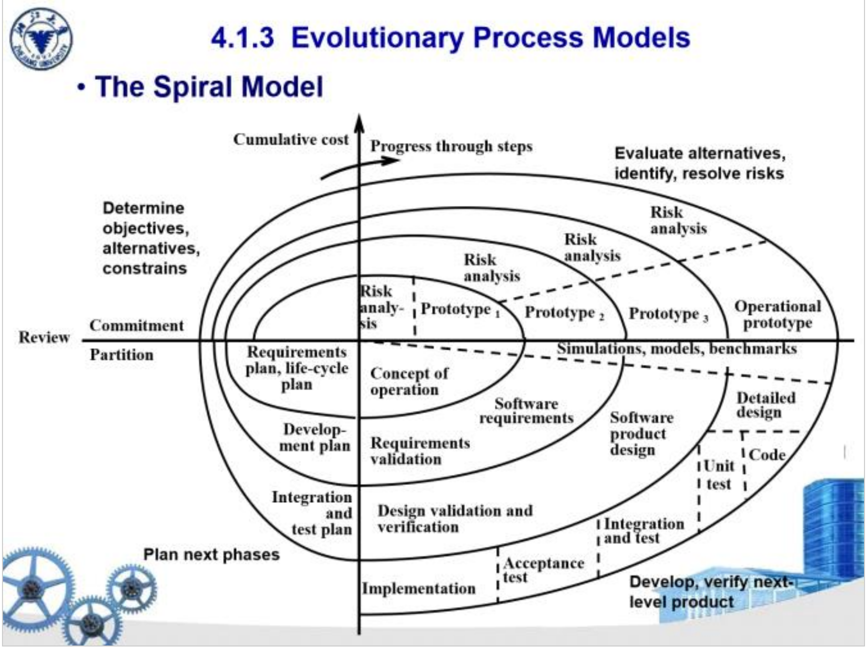
Deployment Delivery & Feedback

- Good first step when customer has a legitimate need, but is clueless about the details
- The prototype must be thrown away

演化式过程模型

原型系统：先出个原型系统给用户看，用户提出反馈

不是最终的交付服务，是个demo，比较适合需求分析 / 关键技术公关



4.1.3 Evolutionary Process Models

- **The Spiral Model**

Cumulative cost

Progress through steps

Evaluate alternatives, identify, resolve risks

Determine objectives, alternatives, constrains

Risk analysis

Risk analysis

Risk analysis

Risk analy-sis

Prototype 1

Prototype 2

Prototype 3

Operational prototype

Review

Commitment

Partition

Requirements plan, life-cycle plan

Concept of operation

Simulations, models, benchmarks

Software requirements

Software product design

Detailed design

Develop-ment plan

Requirements validation

Code

Unit test

Integration and test plan

Design validation and verification

Integration and test

Acceptance test

Implementation

Plan next phases

Develop, verify next-level product

螺旋模型：大型软件开发

四个阶段：

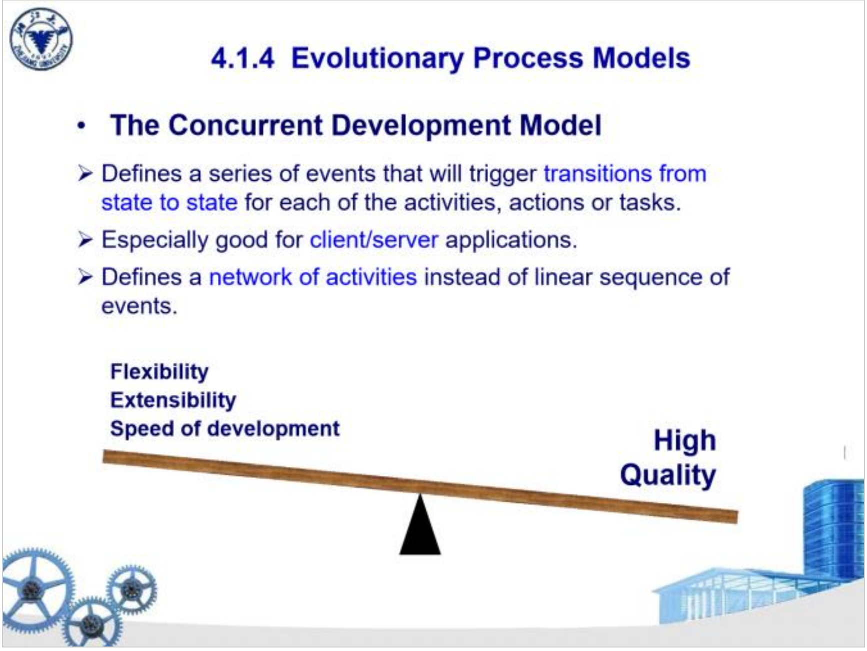
1、定义目标、可选方案、约束

2、评价方案，风险分析

3、实施阶段，对模型开发验证仿真

4、规划下一个循环

Eg. 第一个循环做立项，第二个循环做产品分析



4.1.4 Evolutionary Process Models

- **The Concurrent Development Model**

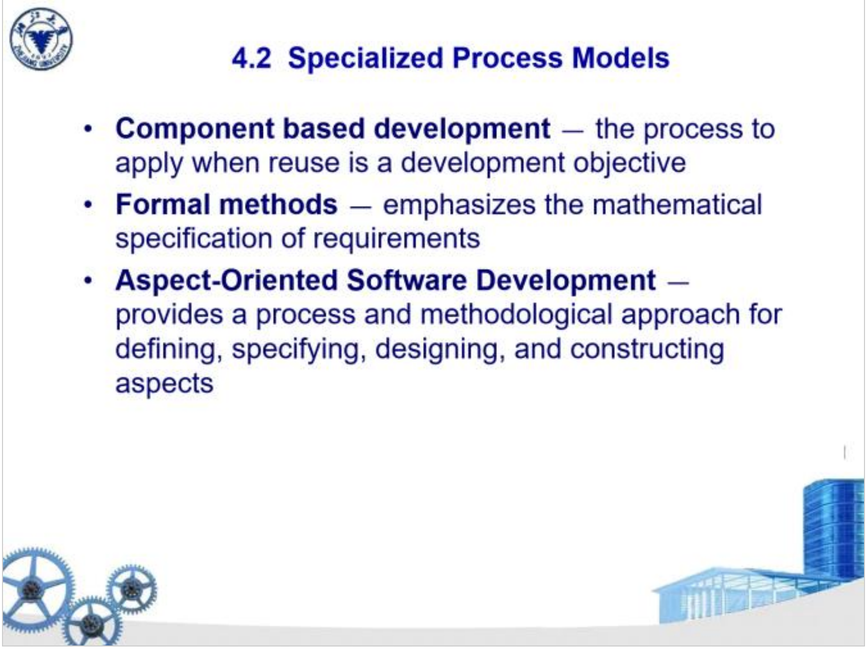
- Defines a series of events that will trigger transitions from state to state for each of the activities, actions or tasks.
- Especially good for client/server applications.
- Defines a network of activities instead of linear sequence of events.

Flexibility
Extensibility
Speed of development

High Quality

并行模型

找到没有依赖的工程，实现并行



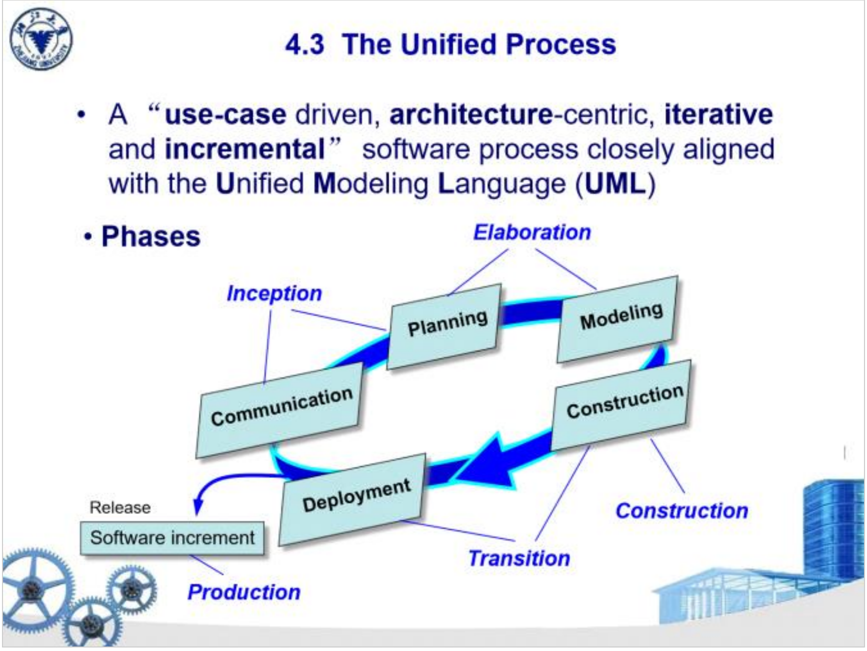
4.2 Specialized Process Models

- **Component based development** — the process to apply when reuse is a development objective
- **Formal methods** — emphasizes the mathematical specification of requirements
- **Aspect-Oriented Software Development** — provides a process and methodological approach for defining, specifying, designing, and constructing aspects

基于构件的软件开发：找有没有现成的，之后思考怎么把这些东西组装

形式化方法：比较精准，适合规模不大但对可靠性要求高的软件开发

面向刻面：一个刻面一个刻面完成

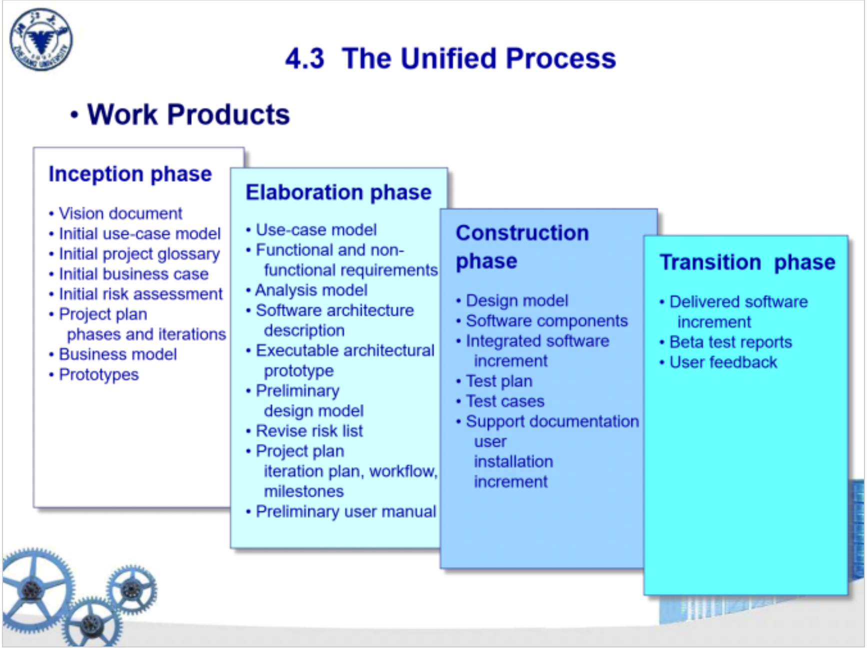


4.3 The Unified Process

- A “**use-case** driven, **architecture**-centric, **iterative** and **incremental**” software process closely aligned with the **U**nified **M**odeling **L**anguage (**UML**)
- **Phases**

统一开发过程——面向对象

对过程中的阶段进行重新界定



4.3 The Unified Process

- **Work Products**

Inception phase

- Vision document
- Initial use-case model
- Initial project glossary
- Initial business case
- Initial risk assessment
- Project plan phases and iterations
- Business model
- Prototypes

Elaboration phase

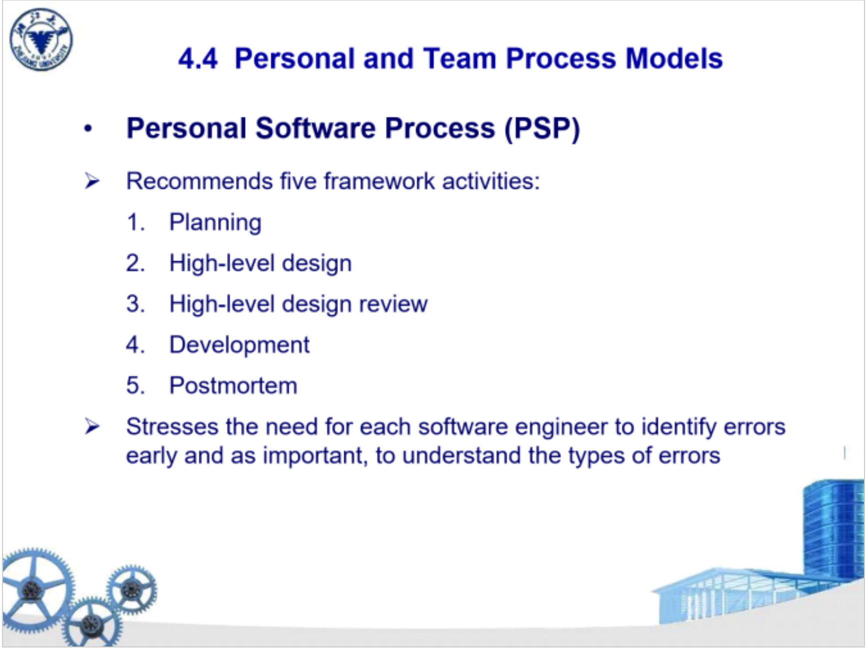
- Use-case model
- Functional and non-functional requirements
- Analysis model
- Software architecture description
- Executable architectural prototype
- Preliminary design model
- Revise risk list
- Project plan iteration plan, workflow, milestones
- Preliminary user manual

Construction phase

- Design model
- Software components
- Integrated software increment
- Test plan
- Test cases
- Support documentation user installation increment

Transition phase

- Delivered software increment
- Beta test reports
- User feedback



4.4 Personal and Team Process Models

- **Personal Software Process (PSP)**

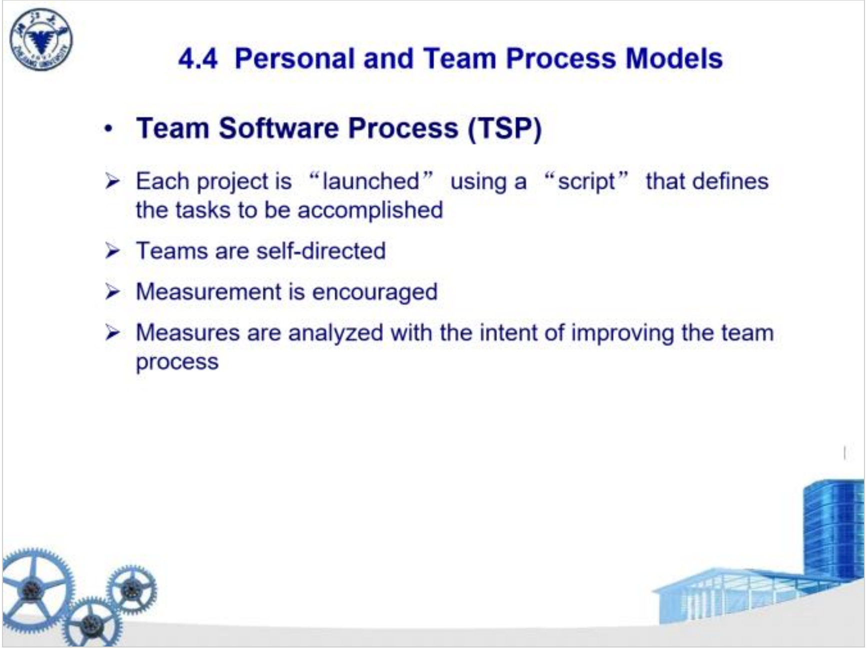
➢ Recommends five framework activities:

1. Planning
2. High-level design
3. High-level design review
4. Development
5. Postmortem

➢ Stresses the need for each software engineer to identify errors early and as important, to understand the types of errors

面向人的过程模型

个人软件过程（PSP）
最后要有个自查（postmortem）



4.4 Personal and Team Process Models

- **Team Software Process (TSP)**

➢ Each project is “launched” using a “script” that defines the tasks to be accomplished

➢ Teams are self-directed

➢ Measurement is encouraged

➢ Measures are analyzed with the intent of improving the team process

团队软件开发过程
团队如何组织，文化建设，相互投入