从一个小小案例谈起

某个老师（T）想要考察一个同学（S）的学习情况和技术水平，于是交给该学生一个任务。

T：我有一个朋友想要一个图象浏览软件，能够查看多种格式的图象，包括BMP、TIFF、JPG、PNG，并且能够支持一般的放大、缩小、漫游。你能做这样一个软件吗？

S：就是类似以前ACDSEE这样的软件吗？

T：差不多，不过不需要商业软件那么强大的功能，我这个朋友计算机是外行，最好能做的比较方便，傻瓜型的，但是例如象ACDSEE自动翻页这种功能还是要的。

S：我以前学过BMP和JPG的图象格式解析，我想没有问题

T：好的，给你30天时间，下周你再来一趟，跟我讲一下你的工作计划和进度。

这位同学非常明白老师的意图，回去后想了一下，并列出了一个清单：

工 作 清 单 （最简版）

一 基本功能：

1。 读取、显示、另存四种格式图片（ BMP、TIFF、JPG、PNG ）

2。 单个图片操作：放大、缩小、漫游

3。 列表显示当前目录下所有四种格式图片文件名

4。 PAGEUP（PAGEDOWN）自动调出当前目录上一张（下一张）图片

二 其它说明：

1。界面尽量简洁，容易操作

2。暂时不要图片预览和打印等附加功能

三 开发工具：VC ++

四 开发环境：普通PC机；Windows 7 /10

五 工作目标：1. 研究一下四种图片的存储格式

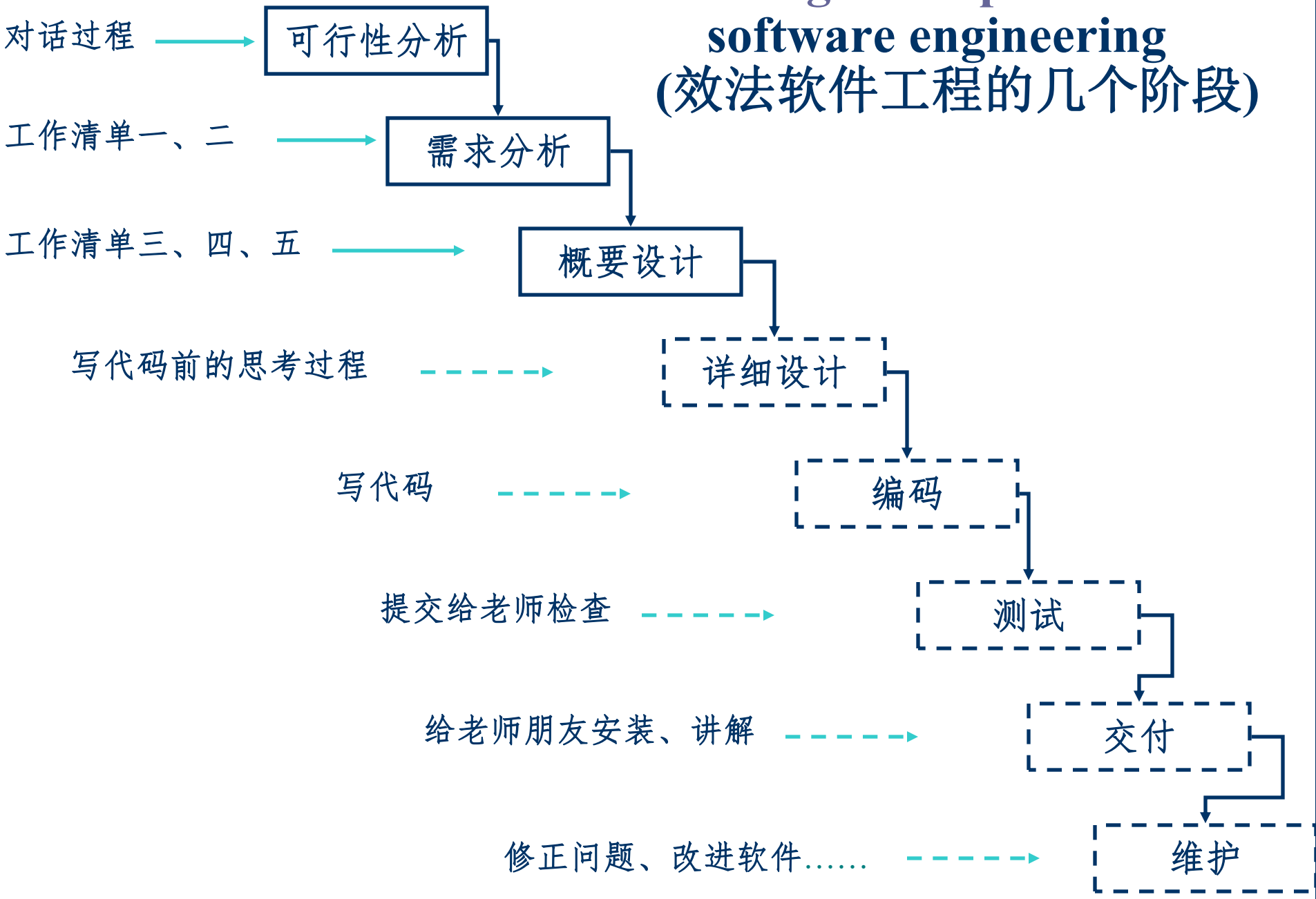
2. 设计一个解析器类，能解析这四种格式文件

3. 设计一个文档类，实现读取、另存和目录浏览功能

4. 设计一个视图类，实现显示、缩放、漫游功能

5. 设计一个控制/协调类，实现系统总控。

The 8 generic phases of software engineering (效法软件工程的几个阶段)



假设：实际情况01

一切顺利，学生S按期交付了软件，经过一两周的试用、修改、完善后，三方都比较满意，该软件在老师的朋友那里成为一个得心应手的工具

Waterfall Model

(评点：一切顺利，基本没有返工；即用自上而下展开的、瀑布似的、经典的、相对固定的过程就完成了任务)

假设：实际情况02

一周后，学生去见老师，并提交了工作清单，他发现老师的这位朋友（C）和老师在一起。

S: 这是工作清单，我已经研究清楚了四种文件的格式，可以写代码了。

T：很好，不过我这位朋友有一些新想法，你不妨听听。

C：你好。我新买了一个扫描仪，你的程序可不可以直接扫描图片进来。

S: 你可以自己扫描呀，买扫描仪的时候一般都会送正版软件的。

C：是的，可是我一直不太会用，你知道我计算机水平不高，学一些新东西很累，也没有时间，如果你的软件能直接链接扫描仪，我只要学会你的软件就行了，我愿意多支付一些费用……还有，我想建一个图片库，你知道，我工作时需要几百个图片，我经常找不到，软件最好还带模糊查询等功能。（举手之劳的事情，用户就是不想做！）

（等于再开发一套已经部分存在的软件，加上扩展功能！）

实际情况2（续）

S：.....!!!!!（明明很简单，非得绕圈！）

C:还有一些，现在一时想不起来，我想起来的话会再跟你联系，时间上还可以长一些。

S:.....!!!!! !!!!! !!!!!（需求不确定！）

T:要不这样吧，你先做一个样子出来给C看看，一边制作，一边修改。

C:这样最好，看见一个基本样子我就知道我到底想要什么了。

事情就这样定下来了，S愤怒的撕掉了自己的工作清单.....，回去后S花1天时间用DELPHI做了个样子，只能读BMP和JPG文件，做了些菜单和工具栏，用ACCESS建了一个图片库。就这个“假”的程序，S和C讨论了一天，S又修改了几次，又讨论了几次，一周后，这个“假”的程序表面看起来和真的一模一样。

实际情况2（续）

于是S打算用VC++重写这个程序，但是他很快发现继续用DELPHI写更方便，因为至少界面不用重做了，于是……，两个月后，这个事情终于结束了。

S顺利的完成了他的毕业设计《图像压缩优化算法设计》，C一直使用这个软件管理他的图片，并庆幸花了较少的钱得到了这么有用的东西.

套路问题

Prototyping Model

(通过建立简单原型完成了任务)

备注: 因为软件规模小,这里的原型已经分不清是需求原型、原型界面、还是设计原型等。

问题1: 下一个项目如果规模比较大且复杂，将采用什么方法呢?

问题2: 是否需要对采用的“开发过程”本身得有更详细的界定和理解，积累更多经验以应对更复杂的情况呢?

- 现代大型软件工程中模型与流程的作用：
  - 现代大型软件系统的开发中，工程化的开发控制取代个人英雄主义，这样才能保证软件系统开发成功；
  - 一个编程高手并不一定是一个优秀程序员，一个优秀程序员却是能将出色的编程能力和开发技巧同严格的软件工程思想有机结合（编程只是软件生命周期中的其中一环），优秀程序员应该掌握软件开发各个阶段的基本技能，如市场分析，可行性分析，需求分析，结构设计，详细设计，编程实现、软件测试，软件评价等等。
  - 简单概括对软件工程的看法:“创意无限，流程保证”。
    - 在一个较大型软件开发过程中，若其中一个小组在使用旧的开发环境开发完毕后，提出使用新的开发环境完成整个系统的下一个新版本，如何决定？（该小组一般要完成多项验证工作）

Chapter 2 Modeling the Process and life cycle

Contents:A: process(过程定义) + modeling software development(软件开发建模)

B: several software process models

C: static and dynamic modeling techniques

D: examples

2.1 The Meaning of Process (过程的含义)

1. The definition for process（过程的定义）

①process: (P45) 软件开发活动中产生某种期望结果的一系列有序任务，涉及活动、约束和资源.

②characteristics: A: activities (可继续分解为系列动作)

B: resource, constraints(软硬件工具，进度)

C: subprocess (大型过程可看作子过程的链接)

Chapter 2 Modeling the Process and life cycle

D: more explaining

③notion: process=life cycle

2. The importance of process（过程的重要性）

统一布置一篇论文的写作格式要求

①generality (通用性) (keep/impose consistency(一致性) and structure(结构性) on a set of activities)

(一致性和结构性可以使我们知道是否已经做好了工作，还能使别人以同样的方式做工作，因而具有相对通用性)

(过程有助于保持大量不同人员开发的产品和服务之间的一致性和质量)

总结经验，完善规范，指导新版本

②guidance (自我指导性) (analyze, check, understand, control and improve activities)

③example: process improvement(P46-47)

----making chocolate cake

Chapter 2 Modeling the Process and life cycle

④Focus on : documentation---making recipe (capture experiences and pass them along to others)

3. Several stages in process（过程的几个阶段）

①several stages: 1—9(P47)（9个阶段已在第一章陈述过）

②explaining:(P47)

A: each stage is itself a process that can be described as a set of activities
(每个阶段本身是一个过程，它同样由一系列活动组成)

B: each activity involves inputs, subactivities constraints, outputs, and resources.
(每个活动涉及输入，子活动，约束，输出，和资源)

C: 过程阶段化的目的: 尽量通用化，以确保最终产品的高质量.

过程虽然有规定性内容，但很多具体描述是不一样的。

Chapter 2 Modeling the Process and life cycle

本节将从模型演化的角度介绍各个过程模型

2.2 Software Process Models（软件过程模型）

1. Reasons for modeling a process（建模的理由）

A: To form a common understanding

(开发团队在记录开发过程的描述时，自然的对软件所涉及到的活动, 资源,约束等达成共识，这共识就是”模型”)

------一般设备、可运行的拓扑环境、进度及费用等等。

开发环境，复审方式，测试手段等

B: To find inconsistencies, redundancies, omissions

(发现过程层面的缺陷(过程实施时的不一致性、多余部分、缺省部分、不完善部分)，从而让过程更有效。)

过程改进

C: To evaluate appropriate activities for reaching process goal (模型应该反映开发的诸多目标，并评价侯选活动的有效性和正确性, 以构建高质量软件)（即：保证各种功能性能特色，还有通过评价各种过程的量化指标数据等，以发现最佳过程。）

D: To tailor (定制) a general process for the particular situation in which it will be used

Chapter 2 Modeling the Process and life cycle

2.Waterfall model（瀑布模型）

⓪ definition: stages are depicted in fig2.1, and be companied documents .

① feature: A:one stage should be completed before the next begins(阶段间的依赖性和连续性)

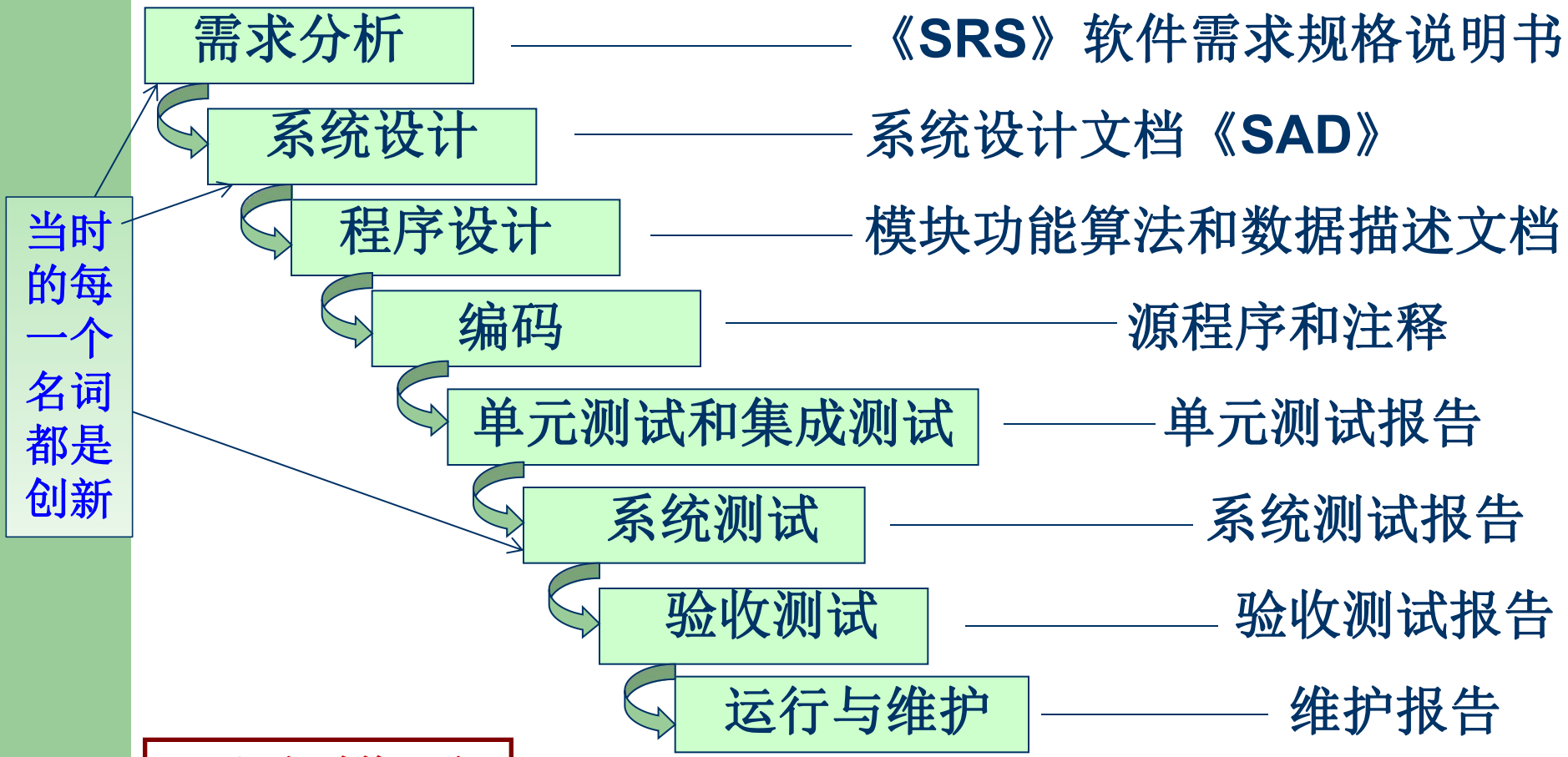
B:推迟实现的观点(尽可能推迟程序的物理实现)

C:质量保证的观点 (每个阶段必须完成规定的文档，每个阶段结束前都要对所完成的文档进行评审)

D:high level view about development

② useness: A:describe software development activities, associated with them were milestone(里程碑), deliverables(提交物); (manager can gauge评价 the progress)

Chapter 2 Modeling the Process and life cycle



Chapter 2 Modeling the Process and life cycle

B:simplicity

(easy to explain to user)(P49-s3)

C: the basis of other complex model

(add feedback loop and extra activities)

③ drawback:

A: can't deal with iteration, not fit actuality

(面临软件变动时, 该模型无法处理实际过程中的重复开发问题（下页图2.2）----软件是一个创造的过程, 不是一个制造的过程)

B: transform is hard (当时的文档转换有困难)

(from one kind of document to another)(P50)

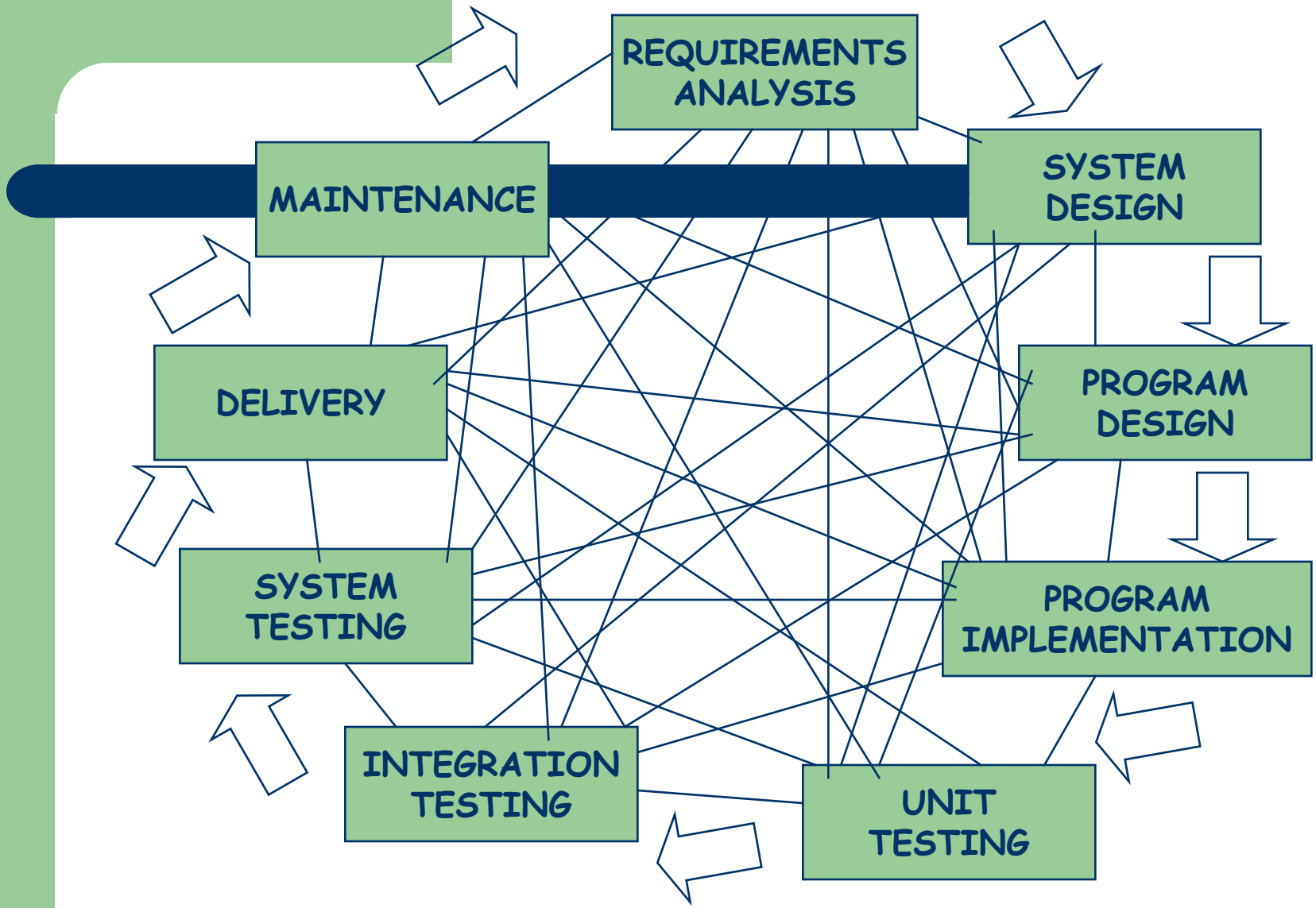


Fig2.2 The software development process in reality

Chapter 2 Modeling the Process and life cycle

④ improvement(对瀑布模型的改进)

A: prototyping (原型化): a subprocess of making prototype (see fig2.3)

prototype(原型): 一种部分开发的产品，用来让用户和开发者共同研究，提出意见，为最终产品定型(see P51)

B: advantage(原型化的优点):

X: prototyping requirement or design ➡ enable improving

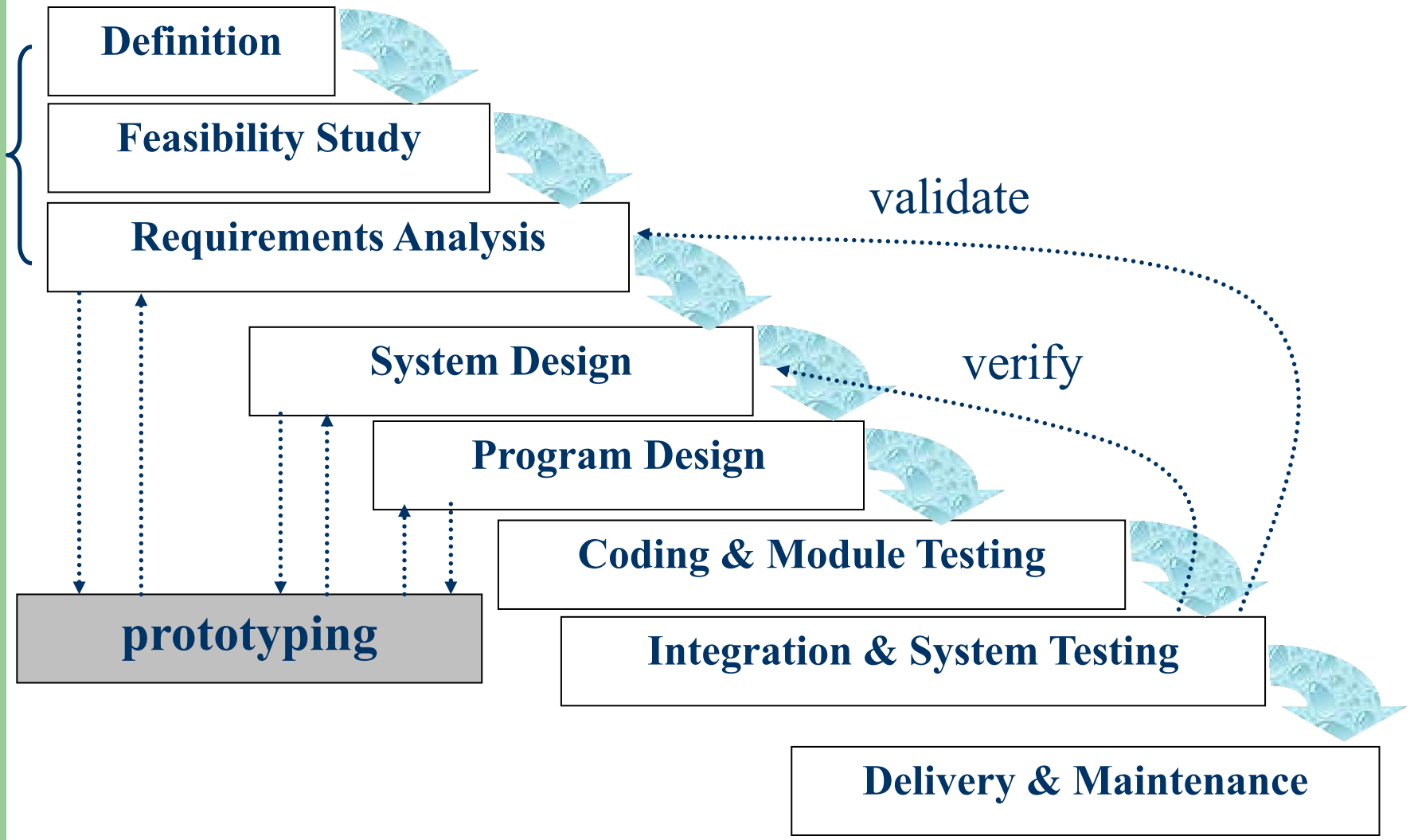
Y: alternative solutions ➡ enable selecting

⑤ the difference between Validation and Verification

Validation（核准）: check <SRS>

Verification（检验）: check “design description”

Chapter 2 Modeling the Process and life cycle



Chapter 2 Modeling the Process and life cycle

3. V model （V模型）

①definition: a variation of the waterfall model(瀑布模型的变种) that demonstrates how the testing activities are related to analysis and design

②The difference from basic waterfall model (Fig2.1)

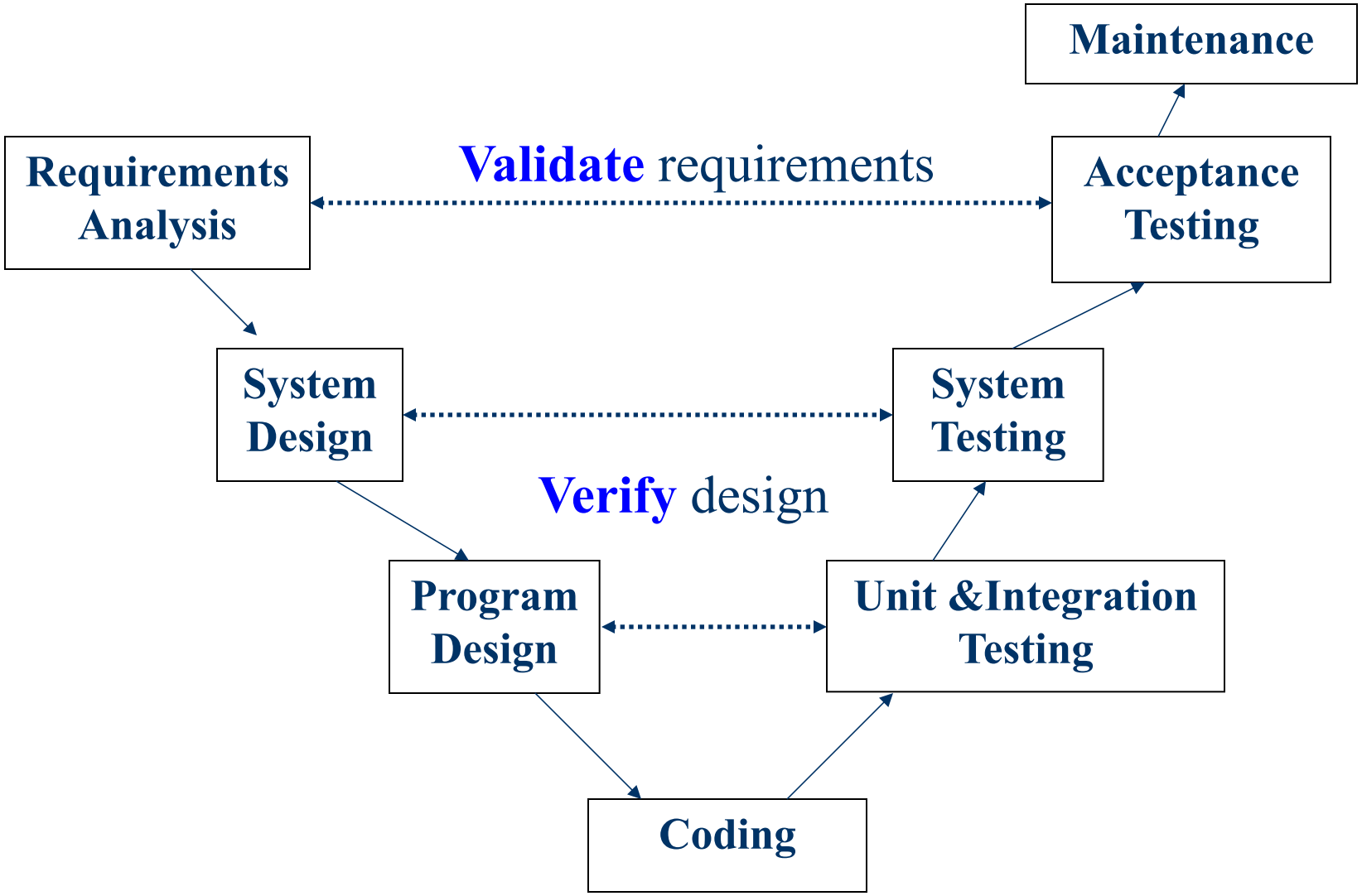
A:V model make some iteration more explicit

B:waterfall model —emphasize→ documents/artifacts（文档/提交物）

V model —emphasize→ activity and correctness 开发活动及正确性，允许各种重复活动。(增加了各种针对性的措施)(fig 2.4)

从此和传统系统工程思想分道扬镳

V Model


Maintenance
Requirements Analysis
Validate requirements
Acceptance Testing
System Design
System Testing
Verify design
Program Design
Unit &Integration Testing
Coding

Chapter 2 Modeling the Process and life cycle

4. Prototyping Model (原型化模型)

① explaining: it can itself be the basis for an effective process model because it enable users construct system quickly as a absolute engineering model

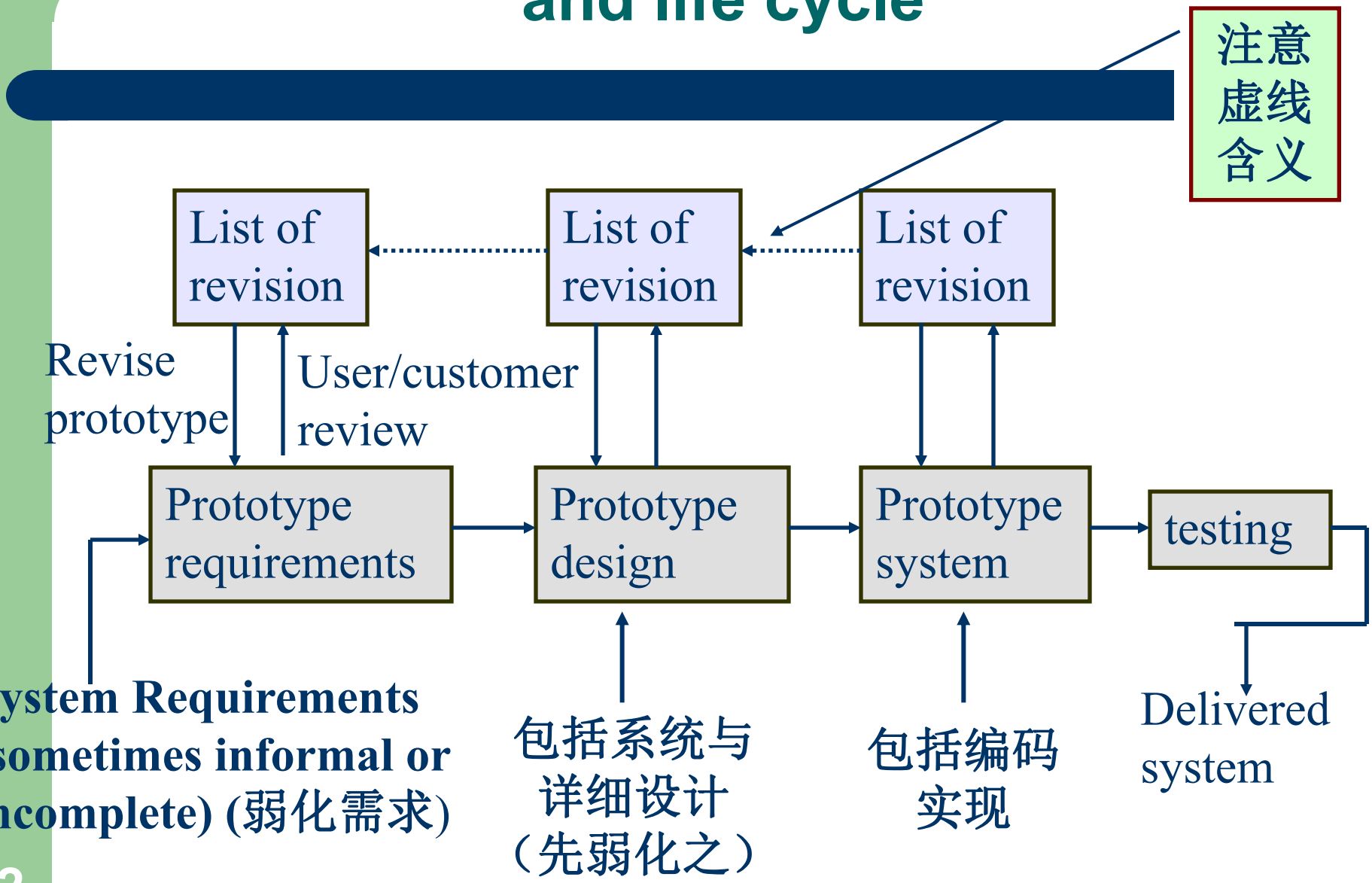
（该模型本身是有效的过程模型的基础。因为它允许用户以独立的工程模型的方式, 每一阶段都基于原型的建立, 以快速构造系统, 逐步完成各阶段任务）

② goal: reducing risk and uncertainty in development （降低开发时的风险和不确定性） (P54)

③ working principle: --see fig 2.5 (P53)

④ 当软件规模较大时，此模型的大型变更很复杂，于是干脆提出了分阶段开发模型。

Chapter 2 Modeling the Process and life cycle



Chapter 2 Modeling the Process and life cycle

5. Operational specification(可操作规格说明模型)

①definition:demonstrating or evaluating requirements by using software package (get requirement and design)

②explaining:A:the difference with traditional model

B:it is similar to “prototyping model”

6. Transformation Model(可变换模型) (Fig 2.7)

①goal:reduce the opportunity for error(by eliminating several development steps) (using automated support system / tools)

②steps: A,B,C(----see P55)

③default/impediment: needing formal specification

Chapter 2 Modeling the Process and life cycle

7.Phased Development: Increments and Interations

(分阶段开发模型：增量式和迭代式) (对原型化模型的改进)

⓪ definition: 系统被设计成部分提交, 每次用户只能得到部分功能, 而其他部分处于开发过程中.

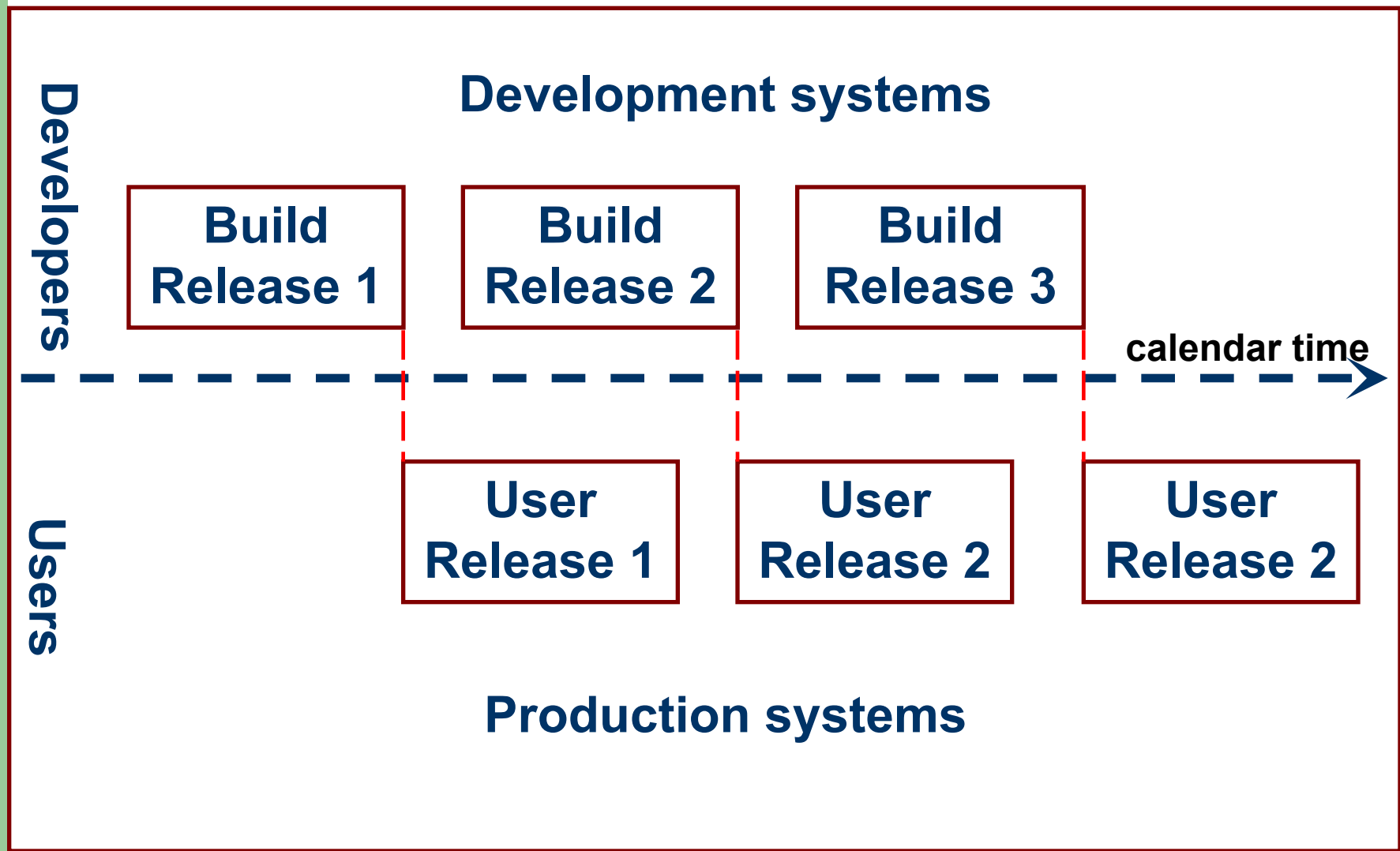
①cycle time(循环时间): 软件开发时整理需求文档时间与系统提交时间之差(P55)

②reducing cycle time→use phased development model

production system(产品系统): 用户正在使用的版本 (fig2.8)

development system(开发系统): 准备代替现有产品系统的下一个版本

Phases Development: Increments and Iterations



Chapter 2 Modeling the Process and life cycle

③two methods:

A: incremental development (增量式开发)(fig2.9)

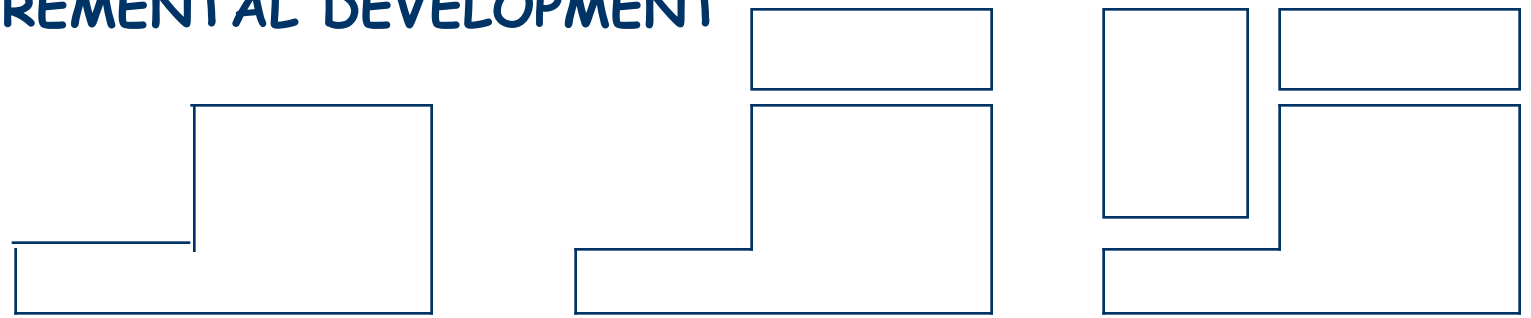
----系统需求按照功能分成若干子系统，开始建造的版本是规模小的、部分功能的系统，后续版本添加包含新功能的子系统，最后版本是包含全部功能的子系统集.

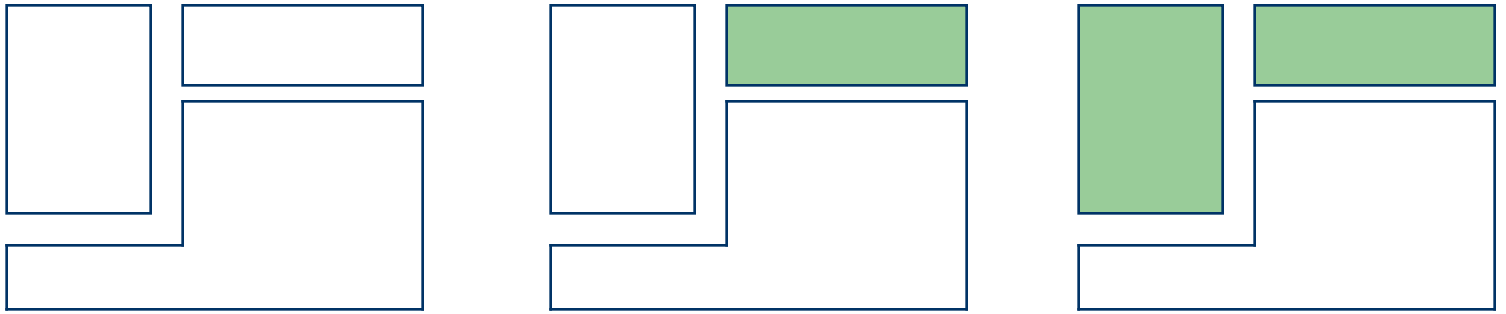
B: iterative development (迭代式开发)

----系统开始就提供了整体功能框架，后续版本陆续增强各个子系统，最后版本使各个子系统的功能达到最强性能.

④ example for incremental development and iterative development (P57)

INCREMENTAL DEVELOPMENT





ITERATIVE DEVELOPMENT

Fig2.9 The incremental and iterative models

Chapter 2 Modeling the Process and life cycle

⑤ combination of iterative and incremental development

----a new release may include new functionality，but existing functionality from the current release may have been enhanced。

⑥ reasons for this forms of Phased Development

A: training----observe user's response

B: market----will be created early

C: fix problems early

D: different expertise for different release/version

⑦ 分阶段开发模型的最大优势: 每个软件版本的周期减少了!

Chapter 2 Modeling the Process and life cycle

- 补充内容------当前市场热门开发过程话题：

  统一过程（UP）可以用三句话来表达：它是用例驱动的、以基本架构为中心的、迭代式和增量性的软件开发过程框架，它使用对象管理组织（OMG）的UML 并与对象管理组织（OMG）的软件过程工程原模型（SPEM）等相兼容。（UP的学术定义）

  - “统一过程”将重复一系列生命期，这些生命期构成了一个系统的开发期寿命。每个生命期都以向客户推出一个产品版本而结束。
  - 每个周期包括四个阶段：开始阶段、确立阶段、构建阶段和移交阶段。每个阶段可以进一步划分为多次迭代。

- 统一过程（UP）定义了下列三个支持工序（discipline）：

（1）配置变更管理工序，用来管理系统和需求变更的配置。

（2）项目管理工序，用来管理项目。

（3）环境配置工序，用来配置项目的环境，包括所涉及到的过程和工具。

- 统一过程定义了下列六个核心工序（这个和一般过程相似）

（1）业务模型工序，通过业务模型获取相关知识以理解需要系统自动完成的业务。（简单时称为问题定义或领域知识）（针对较大项目）

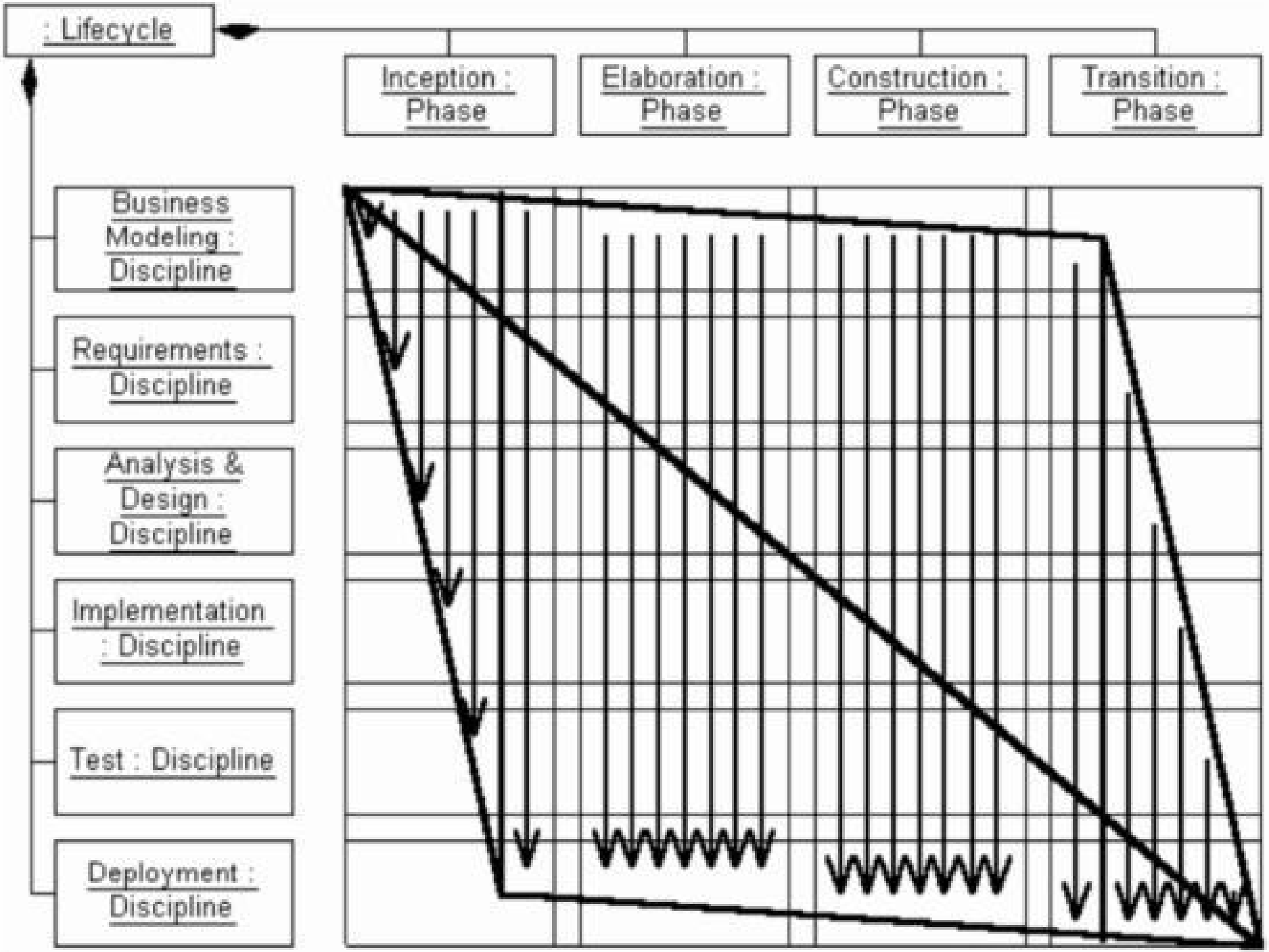
（2）需求工序，通过用例模型获取相关知识以理解自动完成业务的系统需求。

（3）分析设计工序，通过分析/设计模型以分析需求，设计系统结构。

（4）实现工序，基于实现模型实现系统。

（5）测试工序，通过测试模型进行针对需求的系统测试。

（6）部署工序，通过部署模型部署系统。


: Lifecycle
Inception : Phase
Elaboration : Phase
Construction : Phase
Transition : Phase
Business Modeling : Discipline
Requirements : Discipline
Analysis & Design : Discipline
Implementation : Discipline
Test : Discipline
Deployment : Discipline

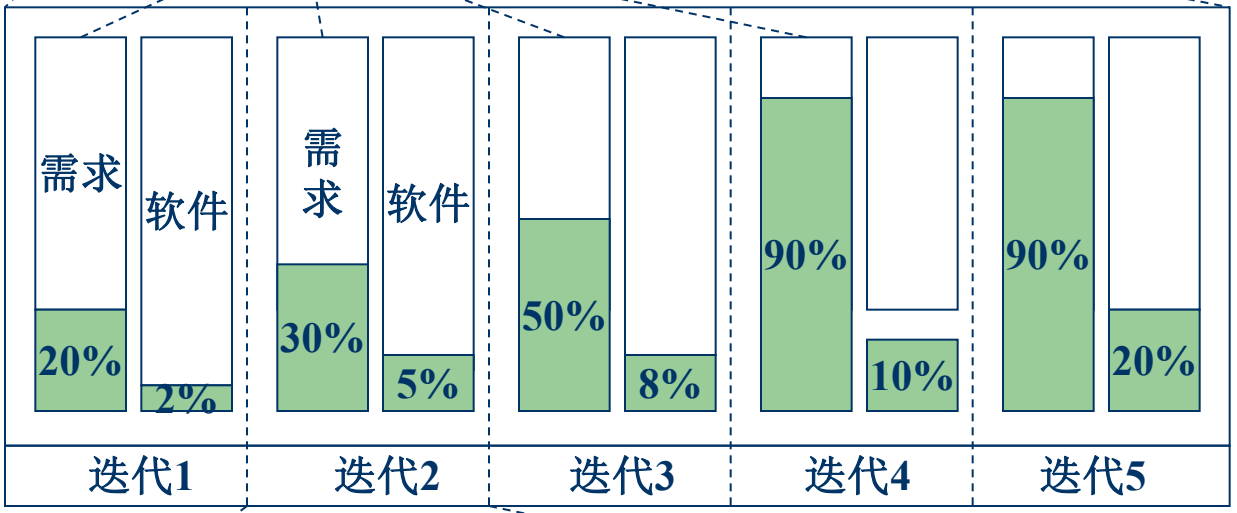
- 进化式迭代开发（Iterative development）
  - 迭代开发是统一开发过程(RUP)的关键实践
  - 开发被组织成一系列固定的短期小项目
  - 每次迭代都产生经过测试、集成并可执行的局部系统
  - 每次迭代都具有各自的需求分析、设计、实现和测试
  - 随着时间和一次次迭代，系统增量式完善
- 注意：上述提法不是RUP的定义。是对UP的定制化描述。（UP的市场化定义）
- RUP是IBM提供支持和包装的UP系统。

进化式分析和设计——早期迭代的主要形式

假设该项目最终有20个迭代

需求讨论会

在进化式迭代开发中，通过一系列需求讨论会，需求在一组早期迭代中进化。或许经过四次迭代和讨论会后，可以定义和精化90%的需求，然而，只构建了10%的软件。



为期三周的迭代

一星期

二星期

三星期

启动会议，向团队明确迭代目标，1小时

团队进行敏捷建模和设计，在白板上绘制UML草图，5小时

开始编码和测试

在此期间进行大部分OOA/D并应用UML

如果有太多工作，分解迭代目标

为形成迭代基线，最后检入代码并冻结代码

演示和为期两天的需求讨论会

在讨论会上进行用例建模

下一次迭代计划会议，2小时

Chapter 2 Modeling the Process and life cycle

8. Spiral model(螺旋模型) (by boehm (1998))

① explaining: combine development activities with risk management to minimize and control risk

（此法将开发活动与风险管理结合起来, 以降低和控制风险）

（该模型的适用范围：较大型软件工程项目）

② spiral model

A: four tasks: plan(计划), goals/alternatives(目标/可选方案), risk evaluating(风险评估), develop and test(开发和测试)

B: four iterations: operating conception(操作概念), requirement(软件需求), designing(软件设计), implementation(系统实现与部署运行)

Chapter 2 Modeling the Process and life cycle

C: “risk analysis” and “prototyping” (in every cycle)

D: note: the first iteration is:

nominal plan⟶normal plan (concept of operation)

（简要设想） （正式计划/操作概念/行业名词）

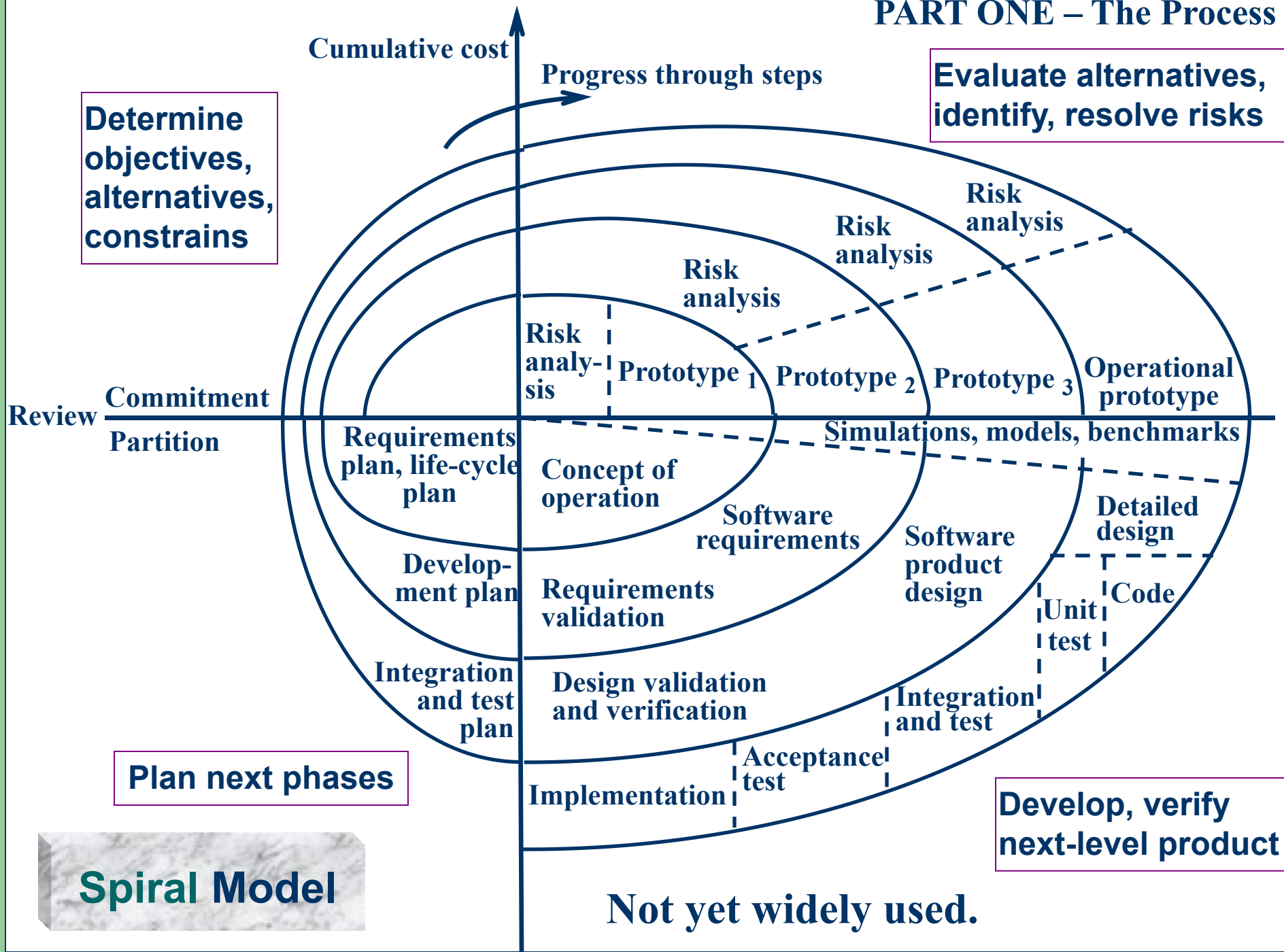
操作概念: 结合业务模型，总结出若干基本的软件操作或流程, 涉及角色，动作，制约关系等等----构成早期领域模型（下页图）

大型系统：很多时候一开始是类似领域模型的文档集合。

③ conclusion:

A: developing task----a collection of process models

B: activities is relatively independent (they don’t rely on a particular model)



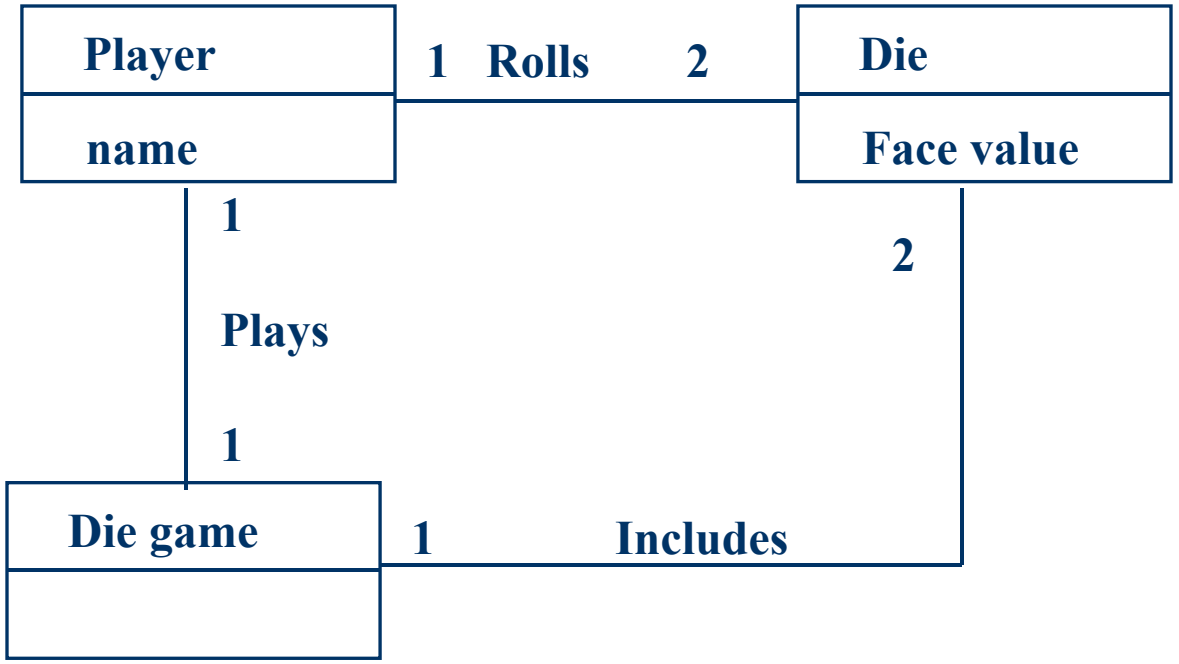
Spiral Model

Not yet widely used.

操作概念/领域模型—简单实例

- 定义用例
  - 骰子游戏：游戏者请求掷双骰子。系统展示结果：如果骰子的总点数是7，则游戏者赢，否则游戏者输。
- 领域模型

(操作概念)



领域模型不是对软件对象的描述，而是对现实世界中的概念的表示

Chapter 2 Modeling the Process and life cycle

9. Agile Methods (敏捷方法) ⇨ 重量级方法的叛逆者

① cause: In late 1990s, some developers resisted the rigor form in software developing, formulated their own principles in which software producing should be more flexible and more quickly.

② four tenets in agile manifesto: (敏捷宣言的四条原则)

A: they value individuals and interactions over processes and tools .

(个体和交互的价值胜过过程和工具)（个人的卓越创意）

B: they prefer to invest time in producing working software rather than in producing comprehensive documentation .

(可以工作的软件胜过面面俱到的文档) （文档非软件）

Chapter 2 Modeling the Process and life cycle

C: they focus on customer collaboration rather than contract negotiation .

(客户合作胜过合同谈判)

D: they concentrate on responding to change rather than on creating a plan and then following it .

(响应变化胜过遵循计划)

③ 说明：上述四条原则反映了敏捷方法的软件过程倾向性。

强调: 人与人之间的交互是复杂的，并且其效果从来都是难以预期的，但却是工作中最重要的方面 .

目标：尽可能早的，持续的对有价值的软件系统的交付活动，以客户满意为最终目标。

Chapter 2 Modeling the Process and life cycle

④敏捷开发过程的方法

A: Extreme Programming (XP) (极限编程)

极限编程(XP)是于1998年由Smalltalk社群中的大师级人物Kent Beck首先倡导的，是敏捷方法中最主要的流派。

B: Crystal （水晶法）

C: SCRUM（并列争球法）

D: Adaptive Software Development(ASD) (自适应软件开发)

E: Feature Driven Development(FDD) (特征驱动软件开发)

Chapter 2 Modeling the Process and life cycle

⑤ XP简介

A：四个变量：成本、时间、质量和范围，通过研究变量之间的相互作用，将项目开发分析的更加透彻，成功讲述一个项目成功的原则

B：XP制定了四个准则：

----沟通：客户与开发者之间持续的交流意见

----简单性：鼓励开发者选择最简单的设计或实现来应对客户的需求

----反馈：指在软件开发过程中的各个活动中，包含的各种反馈循环工作

----勇气：指尽早的和经常性的交付软件功能的承诺

C:十二条制作原则：计划游戏、小版本、隐喻、简单设计、测试、重构、结队编程、代码集体所有、持续集成、每周工作40小时、现场客户、编码标准

---小版本：系统设计要支持尽可能早的交付。（测试要简单有效。）（软件子集如何定位？）

---简单设计：只处理当前需求，使设计保持简单。（因为假设需求是变化的）（不要预防式设计）

---编码标准：编码支持其他实践，例如测试和重构等。

D: XP是一个非常庞大的知识库，每一项都是一门值得深究的学问。提出这些要求和原则后，XP又提出了一系列的解决方案，即策略，其中包含：管理、设施、计划、开发、设计和测试策略。在真正实现XP时，XP方法又提供了将策略成功应用的实践。

- 选择题示例：
- 关于小版本(小型发布)的说明：敏捷开发方法中，对计划的发布版本应该（　　）。　--------答案：B

A：按产品特性交付：需要交付的特性都必须交付，必要时要推迟发布时间

B：按日期交付：按照预定发布时间进行发布，必要时候裁剪部分功能特性。

C：临时决定：我们会平衡一下，临时根据市场要求和开发进展来确定，可能会同时调整交付时间和特性。

D：在迭代模式下，没有必要计划版本。每个迭代都应该完成可发布的版本，按照市场需要发布迭代版本即可。

Chapter 2 Modeling the Process and life cycle

2.3 Tools and Techniques for Process modeling

- There are many choices for modeling tools and techniques, once you decide what you want to capture in your process model
  - 含义之一: 建模工具与技术是在过程模型之内的具体运用，且有诸多选择。
- The appropriate technique for you depends on your goals and your preferred work style
  - 团队中采用的工具和技术与其工作目标及工作风格有密切关系。
  - 一座建筑即使有模型，施工时有的喜欢水泥，有的喜欢木头。即使采用水泥，但施工工艺有所不同。

Chapter 2 Modeling the Process and life cycle

1. Introduction

①modeling tools and techniques:
(建模工具和技术)
modeling notations:
(建模标准符号表示系统)

} have close relation

②the type of model(模型的分类) (in notation system)

A: static model(静态模型): (depict(描述) a process, showing that the input are transformed to outputs)

B: dynamic model(动态模型): (enact(推演) a process, user and developer can see the simulated result)

----含义一：系统中的动态转换关系的描述。（信号灯转换模式）

----含义二：基于要素的整个系统动态仿真推演。(本文特指此处)

Chapter 2 Modeling the Process and life cycle

2. Example:Lai Notation (a static model)

（范例：Lai符号描述系统）(早期符号描述系统示例:观摩)

①explaining:

A: notes: comprehensive process notation ….(P64)

(综合的过程符号描述系统, 允许人们在任何详细的层次上对任何过程建模, 该模型范式中可由人员完成角色, 由资源支持活动, 最后导致软件工件/制品的产生)

B: the model: shows relations among roles, activities, and artifacts, and state table show information about the completing of each at a given time(该过程模型可以用角色、活动、加工项(工件)来显示彼此之间的关系，用状态表显示每个加工项(工件)在特定时间的完成情况)

Chapter 2 Modeling the Process and life cycle

②elements（过程的元素） (seven elements—see P64)

A: 活动：过程中要发生的事件。各种前后关系、触发条件、规则、团队成员等等。也可以理解为子过程。

B: 序列。活动顺序等等。

C: 过程模型。小型工程可以认为是开发方式等描述。

D: 资源。活动所需的各种资源标注。

E: 控制。针对活动的外部影响等。

F: 策略。各种指导原则，包括约束等。

G: 组织。各种层次化结构等描述。包括物理的和软件逻辑的结构。

③process description（过程描述）：（用模板勾勒框架）

several levels of abstraction –several templates

(such as ADT(artifact define template)：制品定义模板)

④example: driving an automobile（机动车驾驶）

A: table 2.1: description of the key resource (of a car)

artifact definition form----one of three subsystems

(role, activity, artifact)

B: fig2.11—process of starting a car(process / activity)

C: fig2.12----transition diagram (role(规则) about a car)

D: conclusion: Lai notation is useful in requirement

Name		Car	
Synopsis		This is the artifact that represents a class of cars	
Complexity type		Composite	
Data type		(car c, user-defined)	
Artifact-state list	Parked	(state_of(car.engine)=off) (state_of(car.gear)=park) (state_of(car.speed)=stand)	Car is not moving, and engine is not running.
	Initiated	(state_of(car.engine)=on) (state_of(car.key_hole)=has-key) (state_of(car-driver(car.))=in-car) (state_of(car. gear)=drive) (state_of(car.speed)=stand)	Car is not moving, but the engine is running.
	moving	(state_of(car.engine)=on) (state_of(car.key_hole)=has-key) (state_of(car-driver(car.))=driving) (state_of(car. gear)=drive)or (state_of(car. gear)=reverse) (state_of(car.speed)=stand)or (state_of(car.speed)=slow)or (state_of(car.speed)=medium)or (state_of(car.speed)=high)	Car is moving forward or backward.

Sub-artifact list	Doors	the four doors of a car
	Engine	the engine of a car
	Keyhole	the ignition keyhole of a car
	Gear	the gear of a car
	Speed	the speed of a car
Relation list	This is the relation between a car and a key	
	This is the relation between a car and a driver	

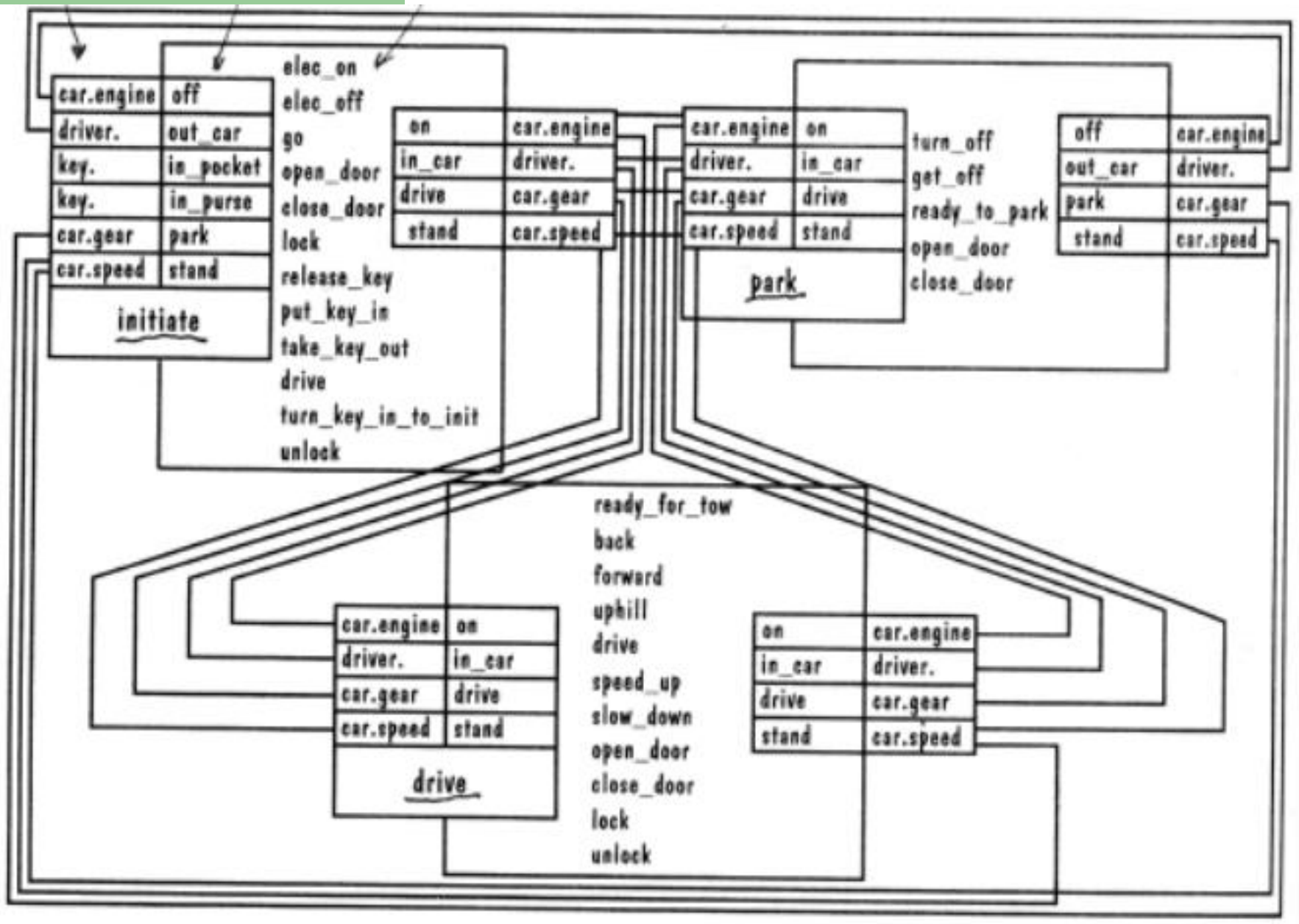


Fig2.11 The process of starting a car (Lai 1991)

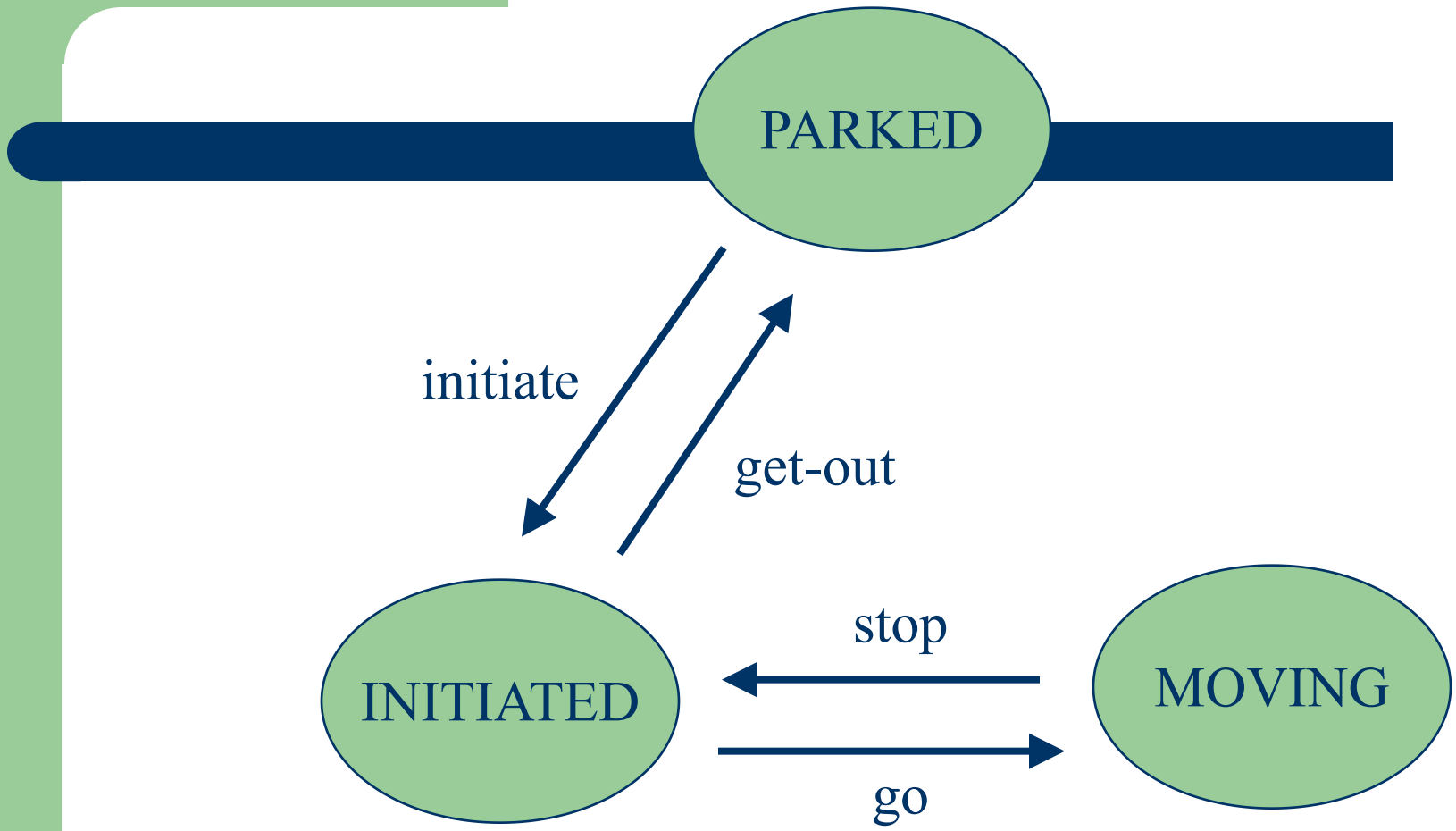


Fig2.12 Transition diagram for a car (Lai 1991)

Chapter 2 Modeling the Process and life cycle

3. Dynamic Process Modeling（动态过程建模）

----系统动力学：展示资源流(非一般性输入)如何通过活动成为输出。

① meaning: (a model can enact(推演) a process)

resource, artifact → enact a process → output → [evaluate → change resources,etc]

② role: simulation → change resources → best scheme(about process)

③example: system dynamic model(系统动态模型)

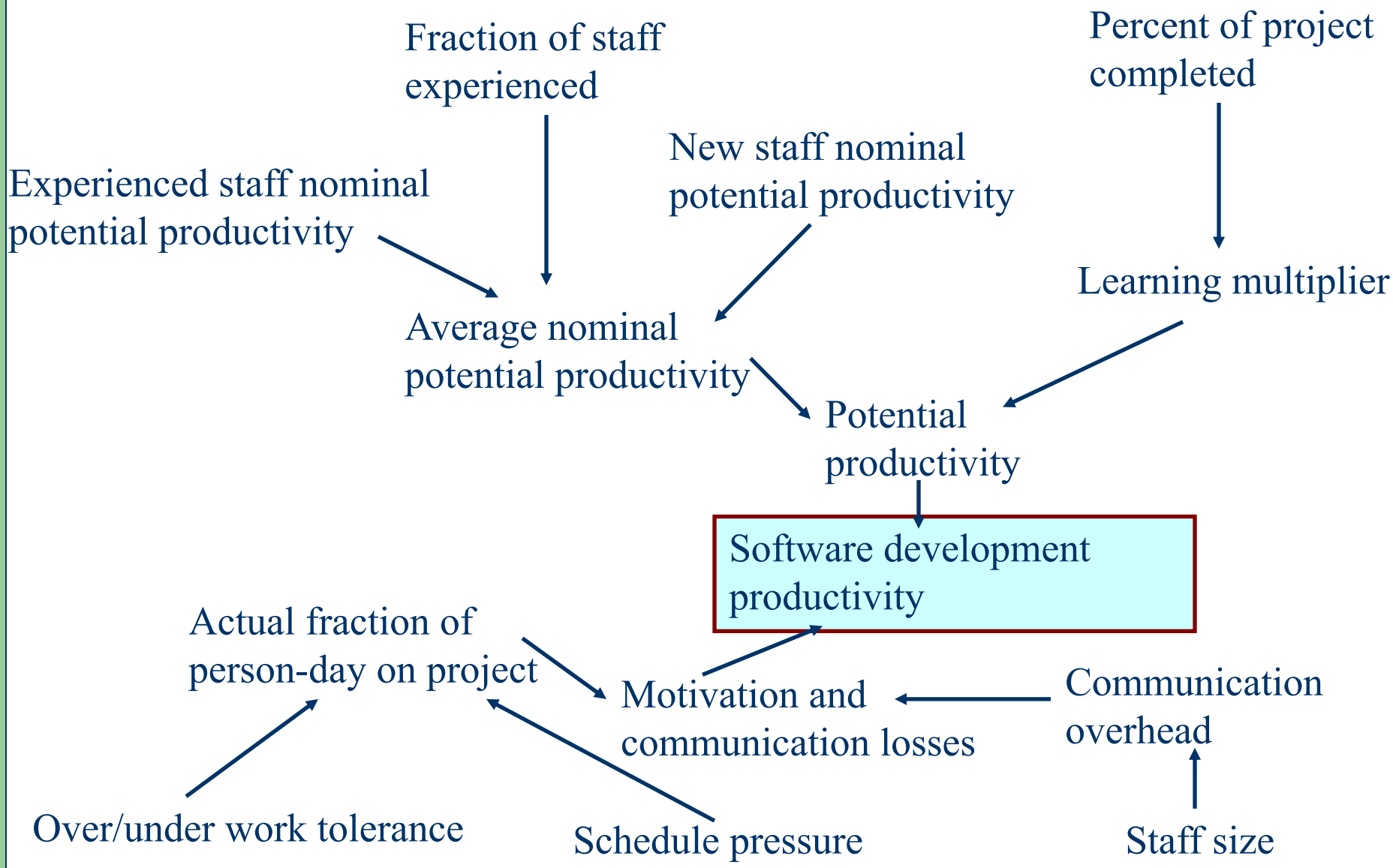
A:fig2.13:model of factors contributing to productivity

X: resource: the productivity of the experienced staff

fraction(of staff experienced)

the productivity of the new staff

(The fourth: percent of project completed)


Fraction of staff experienced
Percent of project completed
Experienced staff nominal potential productivity
New staff nominal potential productivity
Learning multiplier
Average nominal potential productivity
Potential productivity
Software development productivity
Actual fraction of person-day on project
Motivation and communication losses
Communication overhead
Over/under work tolerance
Schedule pressure
Staff size

Chapter 2 Modeling the Process and life cycle

基于经验数据，研究报告和直觉等要素

Y: restraints: work tolerance
schedule pressure
staff scale

identify relationship → quantify relationship → simulation → best choices

B: 系统动力学模型----fig2.14 （超过100个因果链接）

four major areas that affect productivity:
- software production
- human resource management
- planning
- control

------dynamics model can be extensive and complex

Chapter 2 Modeling the Process and life cycle

C: Be caution in using

----quantified relationship is heuristic or vague

2.4 Practical Process Modeling

1. Marvel case studies:

①introduction: MSL—Marvel specification language

A:role:X:define a process

Y:generate a Marvel process enactment environment

B:main constructs and description:

X:constructs----classes, rules, tool envelopes

Y:description----a rule based, OO orientation,

a set of envelops

Chapter 2 Modeling the Process and life cycle

②example (P66 and fig 2.15, 2.16)

③advantage (P68)

A: generating “process enact environment”

B: “information modeling” and “modeling behavior”

2.Desirable Properties of Process Modeling Tools and techniques

----five categories of desirable properties: (P68,69)

- 如何称得上一名优秀的程序员？
- 系统架构师应该具备的素质是什么？
- 来自教材：练习题 4，练习题 11。

①②③④⑤⑥ ⑦ ⑧⑨⑩