Chapter 3 软件过程Software Process

3.1 通用过程模型Generic Process Model

通用过程框架(generic process framework) 定义了5个框架活动(framework activity)——CPMCD，以及一系列伞活动(umbrella activity)——project tracking and control, risk management, quality assurance, configuration management, technical reviews, and others

※软件过程框架Software process framework

通用过程框架(generic process framework) 包含两部分

- 框架活动(framework activity)(CPMCD)——development dimension开发维度
- 伞型活动(umbrella activity)——management dimension管理维度(评审，QA等)

framework activity → software engineering action → task sets

action是若干个task sets，其内容：

- T：work tasks:
- P：work products: 在work tasks之后形成一个work products
- Q：quality assurance(QA) points: 产生work products后要有质量保证
- M：milestone: 形成里程碑

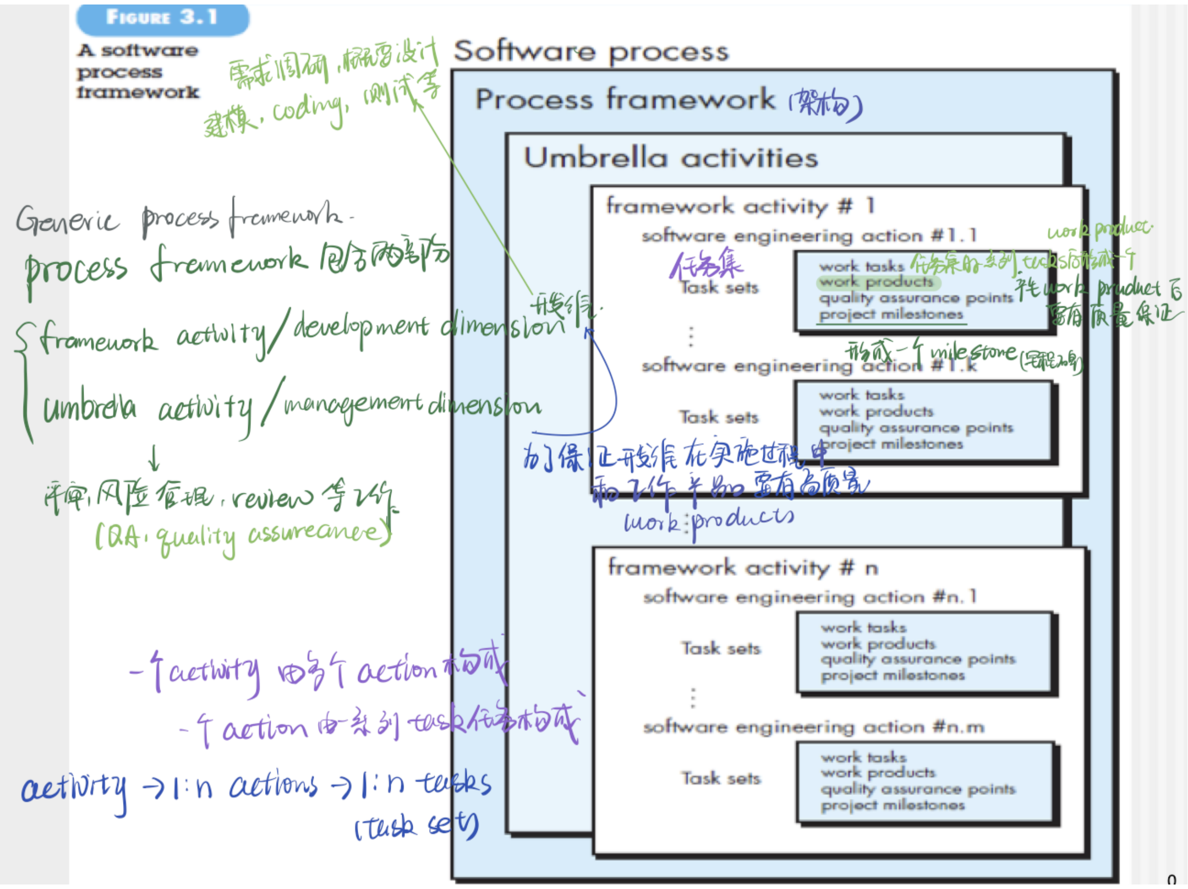


FIGURE 3.1 A software process framework

需求调研，概要设计
建模，coding，测试等

Generic process framework.

process framework 包含两部分

{ framework activity / development dimension
umbrella activity / management dimension

↓

评审，风险管理，review 等工作
(QA，quality assureance)

为了保证开发维在实施过程中和工作产品需有高质量
work products

一个activity 由多个 action构成、

一个action 由一系列task任务构成、

activity → 1:n actions → 1:n tasks
(task set)

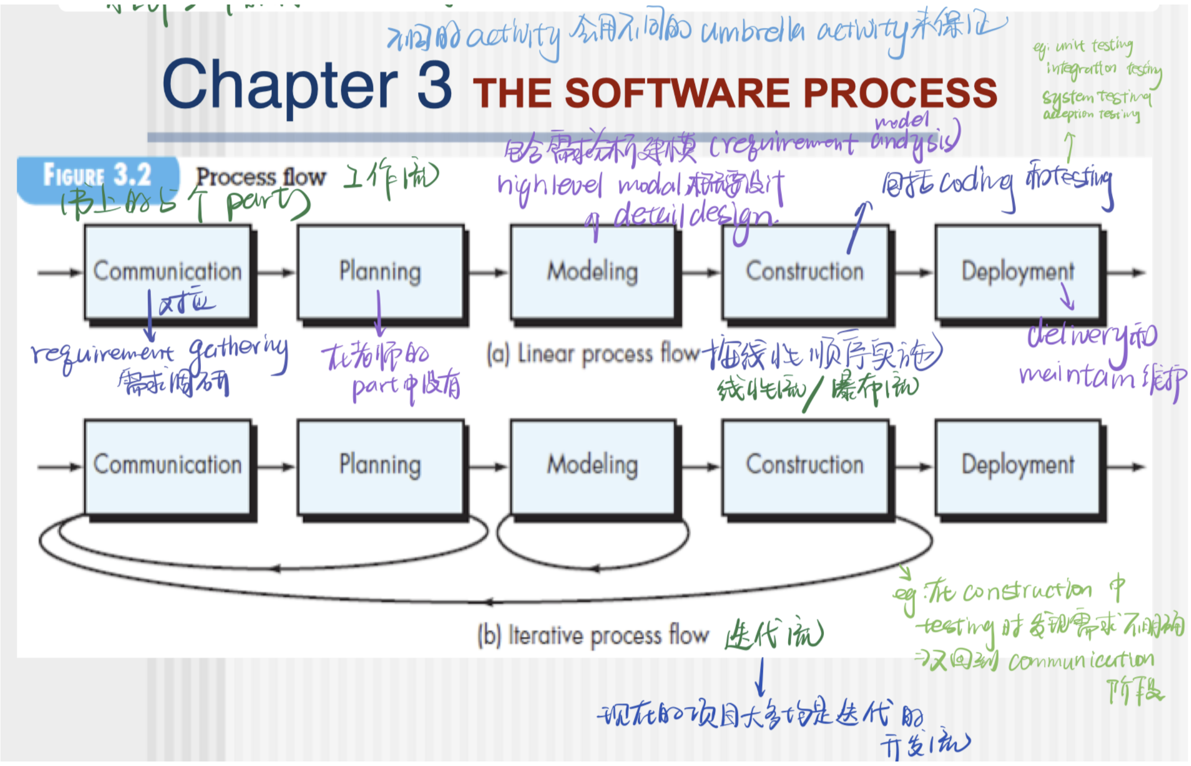
我们应该注意，尚未讨论软件过程的一个重要方面。这个方面被称为过程流程process flow - 描述了如何在序列和时间内组织在每个框架活动中发生的框架活动和动作和任务

※工作流Process flow

种类:

1. Linear
2. Iterative
3. Evolutionary
4. Parallel

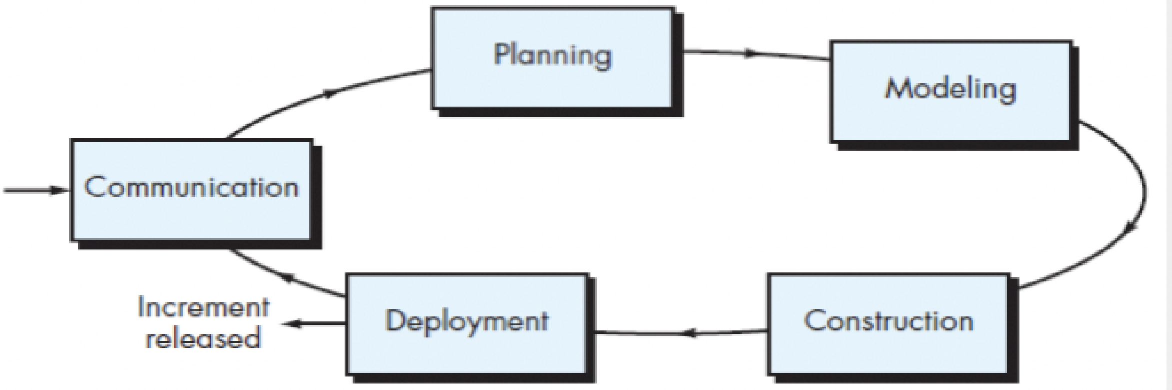
- **线性工作流和迭代工作流**



迭代流的几个迭代箭头：

1. communication和planning反复迭代(planning后，由于1.需求变更;2. 技术和业务更深层的理解;3. 有错误等，重复迭代)→CUE
2. modeling部分的迭代
3. construction后(coding&testing)，可能在testing时发现需求不明确，又回到了前面四个阶段迭代

- **演化工作流(Evolutionary)——螺旋上升**



(c) Evolutionary process flow

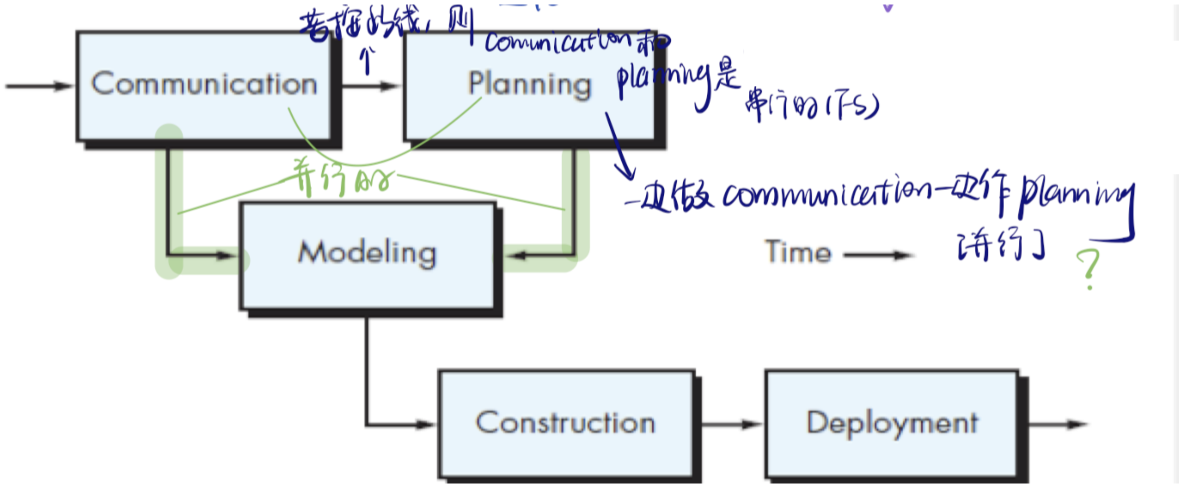
解读：本质还是迭代开发

Planning不回到Communication: 在planning阶段发现有communication的问题，但不回去，而是记录下来直到把本次迭代做完，不论有多少问题，都先进行完本次开发

increment released: **和单纯的迭代流不同，每次迭代完必须要有一个可展示的产品**

每一次的输入（下次迭代要解决的问题）：下次开发的新功能+本次开发中记录下来的问题+用户使用这个increment的产品后反馈要解决的问题

- **并行工作流(Parallel)**



planning和modeling同时做，在communication时导出需求，可以进行planning；同时明确的部分可以进行modeling

3.2 定义框架活动Framework Activity

框架活动的关键问题：考虑到要解决的问题的性质、从事工作的人员的特征以及赞助项目的利益相关者，哪些行动适合框架活动?

- 小项目:
    1. 在线讨论需求和开发笔记;

一个人(远程）有直接的需求进行一个小的软件项目：电话交流就好，

task：

1. 和stakeholder通过电话甲流;
2. 将笔记组织成一个简单的需求书面(written)声明;
3. 和stakeholder用电子邮件进行审阅和通过

- 大项目:

多个股东，股东的需求不同甚至有冲突

action:

1. I奠基inception
2. E诱导elicitation (诱导需求，功能需求和非功能需求function requirement& non-function requirements)→requirement gathering
3. E细化elaboration (建模，用UML各种diagram对需求内容进行抽象)
4. N协商negotiation (甲方讨论需要和不需要的需求，需求规约，需求分析） **与3存在迭代**
5. S规约specification
6. V确认validation

开发过程控制:MS-project

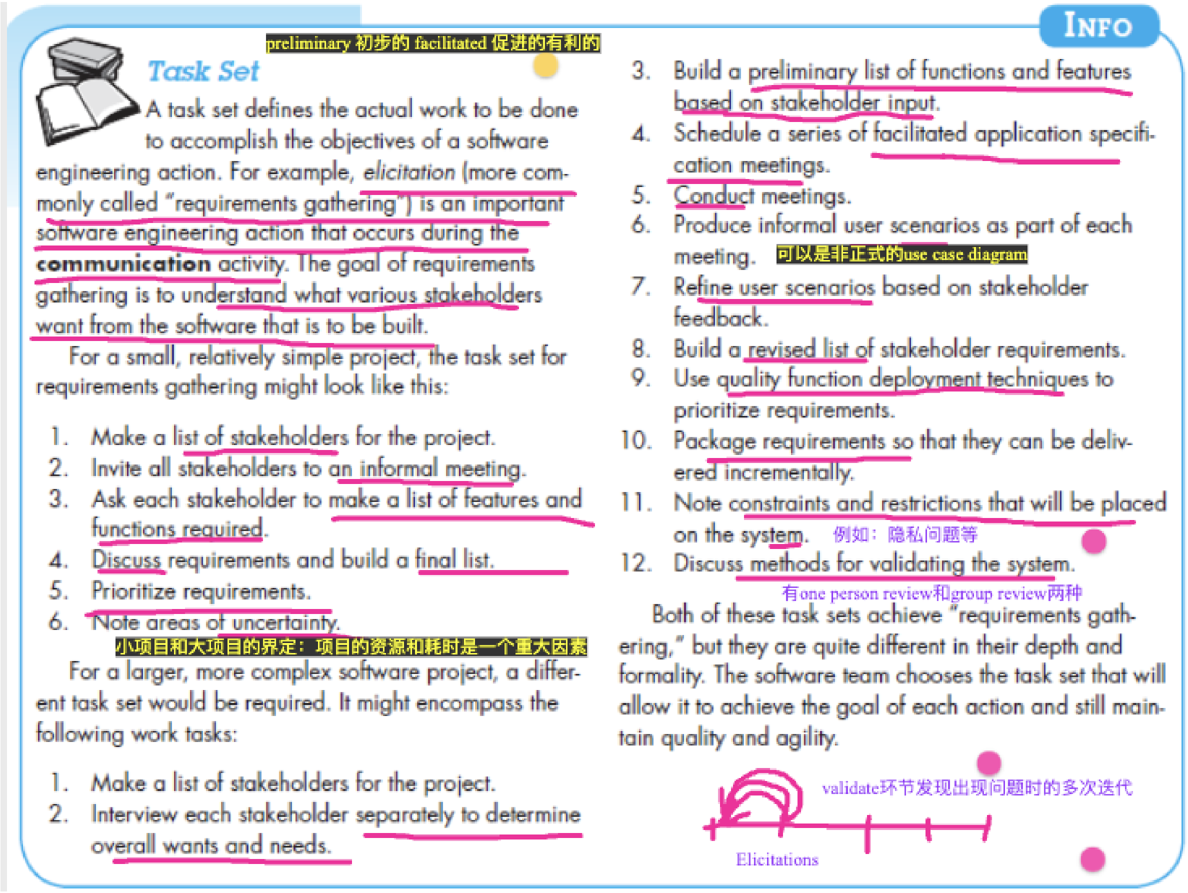
3.3 定义任务列表Identify a Task Set

- 软件工程工作任务work tasks
- 相关工作产品products

- 质量保证点 quality assurance points
- 项目里程碑project milestone的集合

例子：

elicitation:



INFO

preliminary 初步的 facilitated 促进的有利的

Task Set

A task set defines the actual work to be done to accomplish the objectives of a software engineering action. For example, *elicitation* (more commonly called "requirements gathering") is an important software engineering action that occurs during the **communication** activity. The goal of requirements gathering is to understand what various stakeholders want from the software that is to be built.

For a small, relatively simple project, the task set for requirements gathering might look like this:

1. Make a list of stakeholders for the project.
2. Invite all stakeholders to an informal meeting.
3. Ask each stakeholder to make a list of features and functions required.
4. Discuss requirements and build a final list.
5. Prioritize requirements.
6. Note areas of uncertainty.

小项目和大项目的界定：项目的资源和耗时是一个重大因素

For a larger, more complex software project, a different task set would be required. It might encompass the following work tasks:

1. Make a list of stakeholders for the project.
2. Interview each stakeholder separately to determine overall wants and needs.
3. Build a preliminary list of functions and features based on stakeholder input.
4. Schedule a series of facilitated application specification meetings.
5. Conduct meetings.
6. Produce informal user scenarios as part of each meeting. 可以是非正式的use case diagram
7. Refine user scenarios based on stakeholder feedback.
8. Build a revised list of stakeholder requirements.
9. Use quality function deployment techniques to prioritize requirements.
10. Package requirements so that they can be delivered incrementally.
11. Note constraints and restrictions that will be placed on the system. 例如：隐私问题等
12. Discuss methods for validating the system.

有one person review和group review两种

Both of these task sets achieve "requirements gathering," but they are quite different in their depth and formality. The software team chooses the task set that will allow it to achieve the goal of each action and still maintain quality and agility.

validate环节发现出现问题时的多次迭代

Elicitations

e.g. 书上没有的例子：

Coding过程的Action :1个

task set:

1. 分析理解详细设计（详细的设计规约，理解接口）；
2. 算法以及数据结构；
3. 准备环境写代码
4. (单元测试）自我测试self-testing（可能考虑的：算法复杂度时间性能）；
5. SQA软件质量保证；
6. refactor重构（测试后对大项目可能要重构，规范性的）
7. code review；
8. code end(里程碑任务milestone)

提高系统性能的几个方法：

1. 后端的数据库设计(冗余设计，表空间划分，范式)(高访问量的几张表不能放在一个表空间里)；
2. 架构设architecture design；
3. 前端和后端的接口（交易频繁的分到不同接口支流中）；
4. 前端本身的memory使用，和后端的通讯；
5. 模拟工具测试得到性能

(TODO)

3.4 过程模式 Process Pattern

※ 非常重要

定义：

过程模式Process Pattern描述了在软件工程工作中遇到的与过程有关的问题，确定了遇到该问题的环境，并提出了一个或多个行之有效的解决问题的方法。

流程模式为你提供了一个模板--一种在软件流程的背景下描述问题解决方案的一致方法。

pattern: 已证实有用的，抽象成一个模板步骤

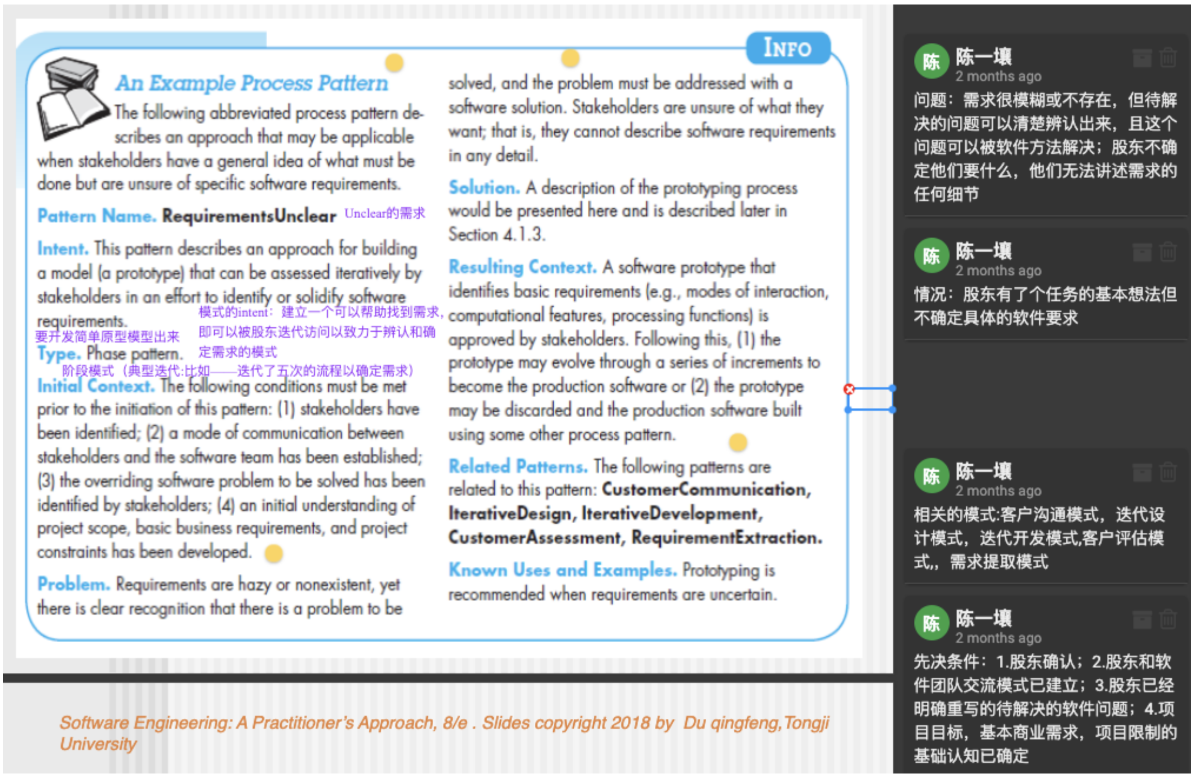
具体来说：

描述与完整流程模型（例如原型设计）相关的问题（和解决方案）；可用于描述与框架活动（例如，计划）或框架活动（例如，项目估算）内的行动相关联的问题（和解决方案）

※ 描述模板template

内容	中文	描述	例子
Pattern Name	模式名字	形容软件过程的context	TechnicalReviews
Force	环境	所需要的环境，硬件，网络，版本管理工具等	
Type	类型	stage pattern: 解决activity相关的（EstablishingCommunication）；task pattern：解决action或task的(RequirementGathering)；phase pattern: 整个框架活动的序列，涉及各种activity(SpiralModel, Prototyping)	stage pattern; task pattern; phase pattern

内容	中文	描述	例子
Initial Context	启动条件	在模式启动之前：（1) 已经发生了哪些组织或团队相关的活动？（2)进程的入口状态是什么？（3) 已有哪些软件工程信息或项目信息？（Activity Happended? State ? SE info)	规划模式Stage pattern的Initial Context（1）客户和软件工程师建立了协作沟通；(2) 成功完成了通信模式的多个任务模式[指定]；(3)项目范围、基本业务需求、项目约束条件已知。
Problem	问题	Pattern可以用来解决什么问题	
Solution	解决方案	描述如何成功执行Pattern	
Resulting Context	接口（出口条件)	为接下来提交什么信息(哪些activity必须出现；过程的出口状态是怎样的；开发了什么软件工程信息)	
Related Pattern	相关的Pattern	提供与此直接相关的所有流程模式的列表。这可以表示为层次结构或以其他示意性形式表示（同级或上下）	比如：同一个action下的两个task上下相关；又比如关于unit test不知道怎么做，去寻找相关、包含或并行的stage pattern
Know Uses and Examples	用过的案例		



An Example Process Pattern

The following abbreviated process pattern describes an approach that may be applicable when stakeholders have a general idea of what must be done but are unsure of specific software requirements.

Pattern Name. **RequirementsUnclear** Unclear的需求

Intent. This pattern describes an approach for building a model (a prototype) that can be assessed iteratively by stakeholders in an effort to identify or solidify software requirements. 要开发简单原型模型出来 模式的intent：建立一个可以帮助找到需求，即可以被股东迭代访问以致力于辨认和确定需求的模式

Type. Phase pattern. 阶段模式（典型迭代:比如——迭代了五次的流程以确定需求）

Initial Context. The following conditions must be met prior to the initiation of this pattern: (1) stakeholders have been identified; (2) a mode of communication between stakeholders and the software team has been established; (3) the overriding software problem to be solved has been identified by stakeholders; (4) an initial understanding of project scope, basic business requirements, and project constraints has been developed.

Problem. Requirements are hazy or nonexistent, yet there is clear recognition that there is a problem to be solved, and the problem must be addressed with a software solution. Stakeholders are unsure of what they want; that is, they cannot describe software requirements in any detail.

Solution. A description of the prototyping process would be presented here and is described later in Section 4.1.3.

Resulting Context. A software prototype that identifies basic requirements (e.g., modes of interaction, computational features, processing functions) is approved by stakeholders. Following this, (1) the prototype may evolve through a series of increments to become the production software or (2) the prototype may be discarded and the production software built using some other process pattern.

Related Patterns. The following patterns are related to this pattern: **CustomerCommunication, IterativeDesign, IterativeDevelopment, CustomerAssessment, RequirementExtraction.**

Known Uses and Examples. Prototyping is recommended when requirements are uncertain.

Software Engineering: A Practitioner's Approach, 8/e . Slides copyright 2018 by Du qingfeng,Tongji University

陈一壤
2 months ago
问题：需求很模糊或不存在，但待解决的问题可以清楚辨认出来，且这个问题可以被软件方法解决；股东不确定他们要什么，他们无法讲述需求的任何细节

陈一壤
2 months ago
情况：股东有了个任务的基本想法但不确定具体的软件要求

陈一壤
2 months ago
相关的模式:客户沟通模式，迭代设计模式，迭代开发模式,客户评估模式,，需求提取模式

陈一壤
2 months ago
先决条件：1.股东确认；2.股东和软件团队交流模式已建立；3.股东已经明确重写的待解决的软件问题；4.项目目标，基本商业需求，项目限制的基础认知已确定

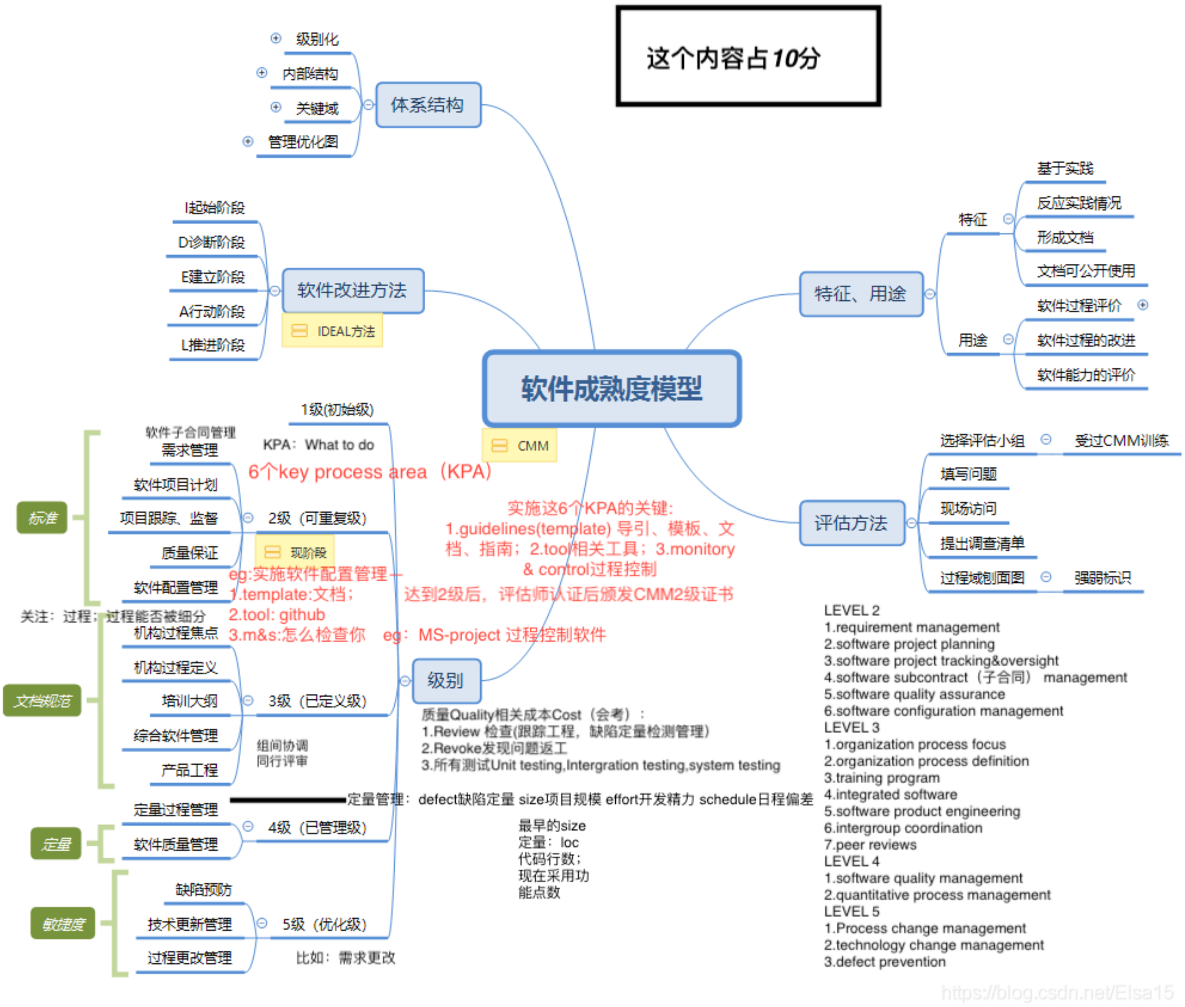
3.5 过程评估与改进Assessment and Improvement

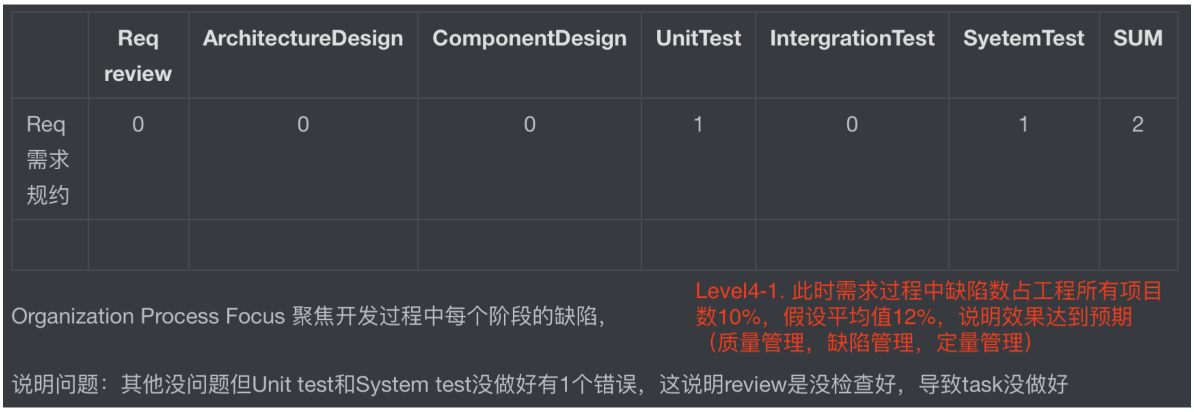
Process patterns必须与可靠的软件工程实践相结合。 此外，可以对过程本身进行评估，以确保它满足一组基本过程标准，这些标准已被证明对于成功的软件工程至关重要。

(CMMI)

Initial Repeatable Defined Managed Optimize

软件成熟度模型CMMI





	Req review	ArchitectureDesign	ComponentDesign	UnitTest	IntergrationTest	SyetemTest	SUM
Req 需求 规约	0	0	0	1	0	1	2

Organization Process Focus 聚焦开发过程中每个阶段的缺陷，

Level4-1. 此时需求过程中缺陷数占工程所有项目数10%，假设平均值12%，说明效果达到预期（质量管理，缺陷管理，定量管理）

说明问题：其他没问题但Unit test和System test没做好有1个错误，这说明review是没检查好，导致task没做好

注：上图的机构过程焦点的例子在下图

add-read

ppt-c3