A photograph of a winter scene in a forest. A path covered in snow leads through bare, snow-laden trees. Snow is falling, creating a soft, hazy atmosphere. The path curves to the right in the distance.

要求  
管理



# 第一部分：需求管理实践

(CH27)

# 故事背后的真相

莎莉：“我终于完成了多供应商目录查询功能的实现。这可真是一项大工程！” 戴夫：“哦，客户两周前就取消了这个功能。”

S：“你没收到修订后的SRS吗？什么意思，它被取消了？这些要求就在我最新版SRS的第6页顶部。”

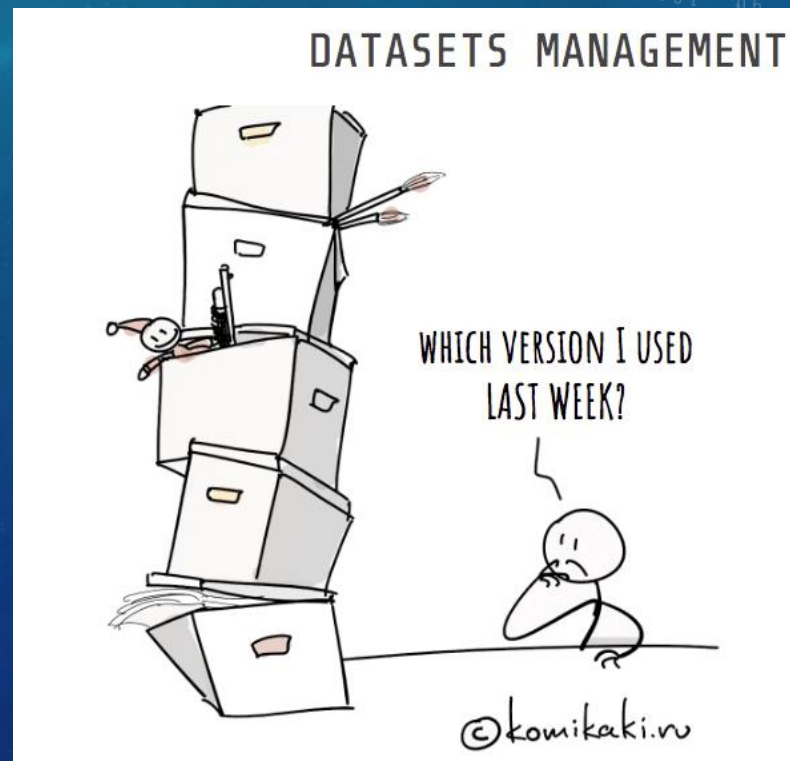
D：“嗯，我的版本里没有。我用的是SRS 1.5版。你看的是哪个版本？”

S：“我的版本也是1.5。这些文档应该完全一样，但显然不是。所以，这个功能还有存在的必要吗？还是我白白浪费了30个小时？”

这个故事讲述了需求管理和有效沟通的重要性。拥有清晰的、有意义的需求仅仅是软件工程的一半。

版本控制是需求管理的重要方面之一，而需求管理又是需求工程的一个要素。

变更控制在需求工程中至关重要。



# 需求管理流程

**瞄准**在整个项目过程中，保持需求协议的完整性、准确性和时效性。

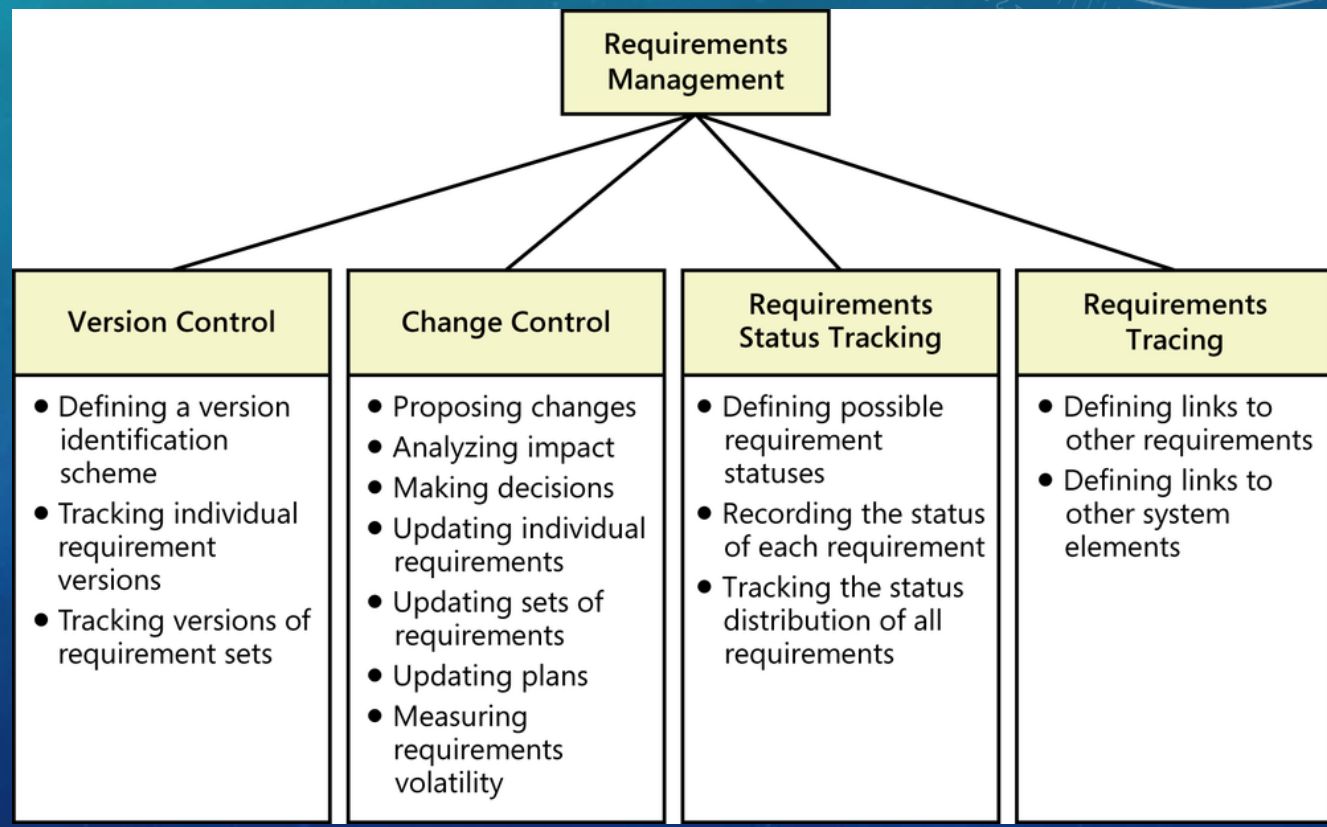
## 四个方面：

- 版本控制：避免不同团队成员使用不同的需求
- 变更控制：用于处理需求变更

- 需求状态跟踪：了解需求的当前状态

（以及通知相关人员采取行动的责任）

- 需求跟踪：监控变更对同一系统内其他元素或其他外部系统的影响。





# 需求管理 过程

## SRS 版本控制

- 版本号——例如 v1.2，但我习惯用 Vddmmyy 格式。
- 无障碍访问——所有团队成员使用同一版本
- **允许**更新——只有指定人员才能修改
- 版本历史记录 – 保留所有先前版本
- 其他属性：
  - 需求创建日期
  - 撰写该要求的作者
  - 需求的优先级/状态/来源（请参考之前关于优先级和来源的讲座——来自哪个利益相关者）
- 提出此需求的理由（完成此需求要达到什么目的）
- 需求所属的版本号或迭代次数（中期报告、交付物编号等）。
- 利益相关者可联系以咨询问题或就变更事宜做出决定（请参阅来源）
- 将采用何种验证方法或验收标准（正式或非正式的同行评审？）

Version	Date	Description of changes and person responsible for making changes
1.0	23/2/2015	Initial draft summary for comments from workshop participants (Craig Sinclair)
1.1	25/2/2015	Revision incorporating feedback from workshop facilitators (Craig Sinclair)
2.0	6/3/2015	Inclusion of submissions from two facilitators (Angus Cook, Amar Varsani) and addition of feedback points raised by 13 survey respondents (Craig Sinclair)
2.1	11/3/2015	Integration of feedback from survey respondents (Craig Sinclair and Phil Cocks)
2.2	13/3/2015	Inclusion of Stage 1 Priorities and attendance list (Craig Sinclair)
3.0	26/3/2015	Inclusion of Stage 2 Priorities voting data and executive summary (Craig Sinclair)
Final for release	30/3/2015	Revision based on feedback from lead authors and Great Southern Science Council committee members (Craig Sinclair)

Suggested citation details: Sinclair C, Cocks P, Beazley L, Rainbird K. Healthy Futures Forum 2015: Regional Health Research Priorities Workshop. Great Southern Science Council: Albany. [Available from <http://www.greatsouthernsciencecouncil.org.au>]

# 需求管理流程

## 变更控制

- 变更只能通过项目规定的变更控制流程进行。程序（更多内容请见第二部分）
- 可以通过多种方式满足新增或变更的需求：
  - 通过将优先级较低的需求推迟到后续迭代中，或者完全取消这些需求。
  - 通过增加人手或外包部分工作
  - 通过延长交付周期或增加敏捷项目的迭代次数。
  - 为了按原定日期发货而牺牲质量

## 敏捷中的变更控制

- 敏捷软件项目管理的宣言之一是：价值回应变化过度遵循计划
- 与其他三人一起
  - 重视个体和互动，而非流程和工具。
  - 比起详尽的文档，实用软件更有价值。
  - 重视与客户合作，以促进合同谈判

# 需求管理流程

敏捷开发对变更原因的解释如下：

- 很难为所有任务制定一个包含开始和结束日期的长期计划（实际上是进度表，例如以甘特图的形式），因为
  - 客户组织中的业务流程可能会发生变化。
  - 客户需要时间来澄清/确认他们的需求。
  - 开发人员对任务的理解（例如，任务是否仍然需要，或者是否会出现一些新任务，任务的复杂程度，开发人员需要多长时间才能完成任务等等）会随着时间的推移而改变。
- 解决方案是建立短期计划例如那些为期两周的安排，可能更现实一些。
- 当然，团队必须遵循这些短期计划。通过执行这些短期计划，开发人员将更好地了解后续任务，从而更好地制定下一批短期计划，以推进他们参与的项目的进一步开发。



# 需求管理流程

## Portkin 的变革减少概念框架

- 变更范围
  - 缩短时间：减少从概念到可复制的时间，即把项目分成尽可能小的模块，这样完成一个模块所需的时间就会更少，而对一个模块进行更改后，重新制作该模块所需的时间也会更少。
  - 减少空间占用：将空间划分为若干个独立的模块，使每个模块不太可能受到其他模块变更的影响。
- 加入变革
  - 增加多样性：通过大量繁殖后代来增加多样性，从而提高个体应对变化的概率。
  - 知识获取：能够跟踪世界变化并调整软件本身的智能机制。





# 需求管理流程

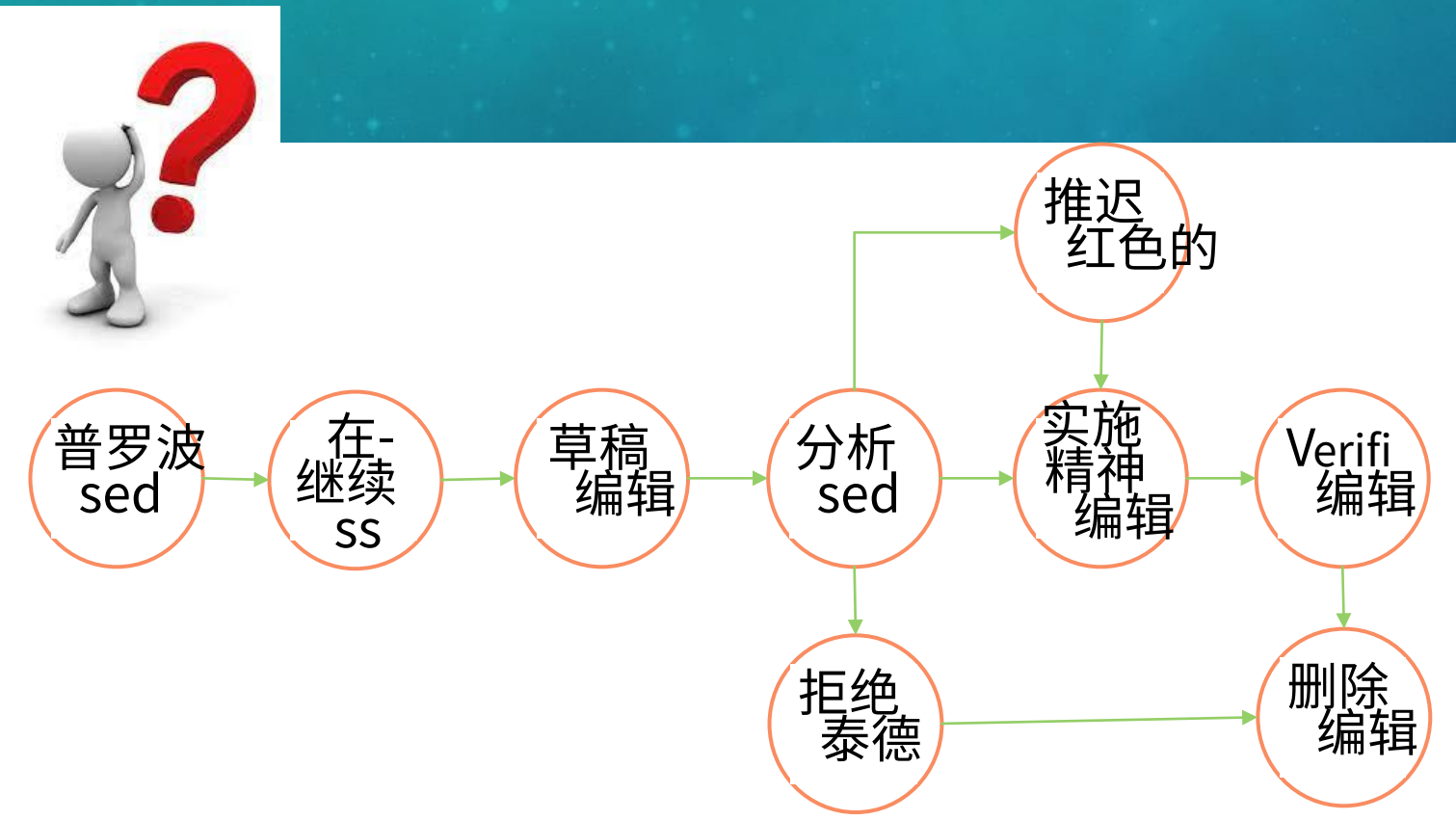
## 状态跟踪

- 一项需求可以有 9 种状态
- 仅当满足指定的转换条件时才更新需求状态

地位	定义
建议的	该需求是由授权方提出的。
进行中	业务分析师正在积极处理该需求。需求的初稿已经编写完成。
已起草	需求已进行分析并明确。
得到正式认可的	该需求已通过验证，并已分配到特定版本中。利益相关者已同意采纳该需求，开发团队也已承诺实施。
实施的	实现该需求的代码已完成设计、编写和单元测试。该需求已追溯到相关的设计和代码元素。实现该需求的软件现已准备好进行测试和审查。
已验证	该需求已满足验收标准，这意味着已确认所实现的需求功能正确。该需求已追溯到相关的测试用例。现在，该需求被视为已完成。
延期	已获批准的需求现已计划在后续版本中实施。
已删除	已实施并批准的一项要求已从基线中移除。
已拒绝	该需求已提出，但从未获得批准，且不打算在任何即将发布的版本中实现。请解释拒绝该需求的原因以及做出该决定的人员。

# 需求管理流程

- 需求在不同状态下会面临不同的问题。
- 解决这些问题将改变现状。
- 问题（见表格）



问题类型	描述
要求问题	对某个需求，还有一些地方尚未理解或确定。
丢失的要求	开发人员在设计或实现过程中发现了一个遗漏的需求。
错误要求	一项要求有误，应该予以更正或删除。
实施在问题	开发人员在实现需求时，会遇到一些问题，例如某些功能应该如何运作，或者设计方案有哪些替代方案。
复制要求	发现两个或多个等效需求。删除一个以外的所有需求。
不需要要求	这项要求已经不再必要了。



# 需求管理流程

## 需求跟踪

(参见第3部分)



# 需求管理流程

## 监控需求工作

- 监控需求以调整项目计划
- 业务分析师的工作主要体现在两个方面：

记录花费在以下方面的小时数：**需求开发活动**例如以下情况：

- 项目规划需求相关活动
- 举办研讨会和访谈、分析文件以及开展其他信息收集活动
- 编写需求规格说明、创建分析模型和确定需求优先级
- 创建和评估旨在辅助需求开发的原型
- 审查需求并执行其他验证活动

将以下活动所投入的精力计为：**需求管理工作**：

- 为项目配置需求管理工具
- 提交需求变更和提出新需求
- 评估拟议变更，包括进行影响分析和做出决策
- 更新需求库
- 将需求变更传达给受影响的利益相关者
- 跟踪和报告要求状态
- 创建需求跟踪信息





## 第二部分：管理变革（第28章）

# 需求变更

## 变革是不可避免的。

- 软件系统的需求通常每月增长 1% 到 3%。
- 变革需要谨慎管理，因为：
  - 成本——开发人员有时无法或不能对拟议的软件变更的成本（所需资源和时间）做出切合实际的估算。
  - 后门问题——开发者可能会在没有得到相关利益方的正式批准的情况下，同意用户提出的更改请求，导致只有开发者和用户知道更改内容，而其他所有人都不知道。
  - 范围蔓延——项目规模越来越大，因为它不断地添加更多功能，而没有调整资源、进度或质量目标。
  - 对其他组件的影响——后期的变更可能会对已完成的工作产生重大影响。



# 原则与政策

## 原则

- 提出的需求变更在最终确定之前都会经过仔细评估（成本和范围）。
- 适当人员对所请求的变更（后门）做出知情的商业决策。
- 变更活动将对受影响的利益相关者可见（影响）。
- 已批准的变更将传达给所有受影响的参与者（后门和影响）。
- 该项目以一致且有效的方式纳入需求变更（范围、后门）。

## 控制策略

- 所有变更都必须遵循既定流程。
- 未经批准的变更，除可行性探索外，不会进行任何设计或实施工作。
- 项目变更控制委员会（CCB）将决定实施哪些变更。
- 变更数据库的内容必须对所有项目干系人可见（或者可以是电子表格）。
- 每次变更都必须进行影响分析。
- 每一项变更都必须能够追溯到已批准的变更请求。
- 每次批准或拒绝变更请求的理由都必须记录在案。

# 流程和变更控制委员会（CCB）

## 建交银行

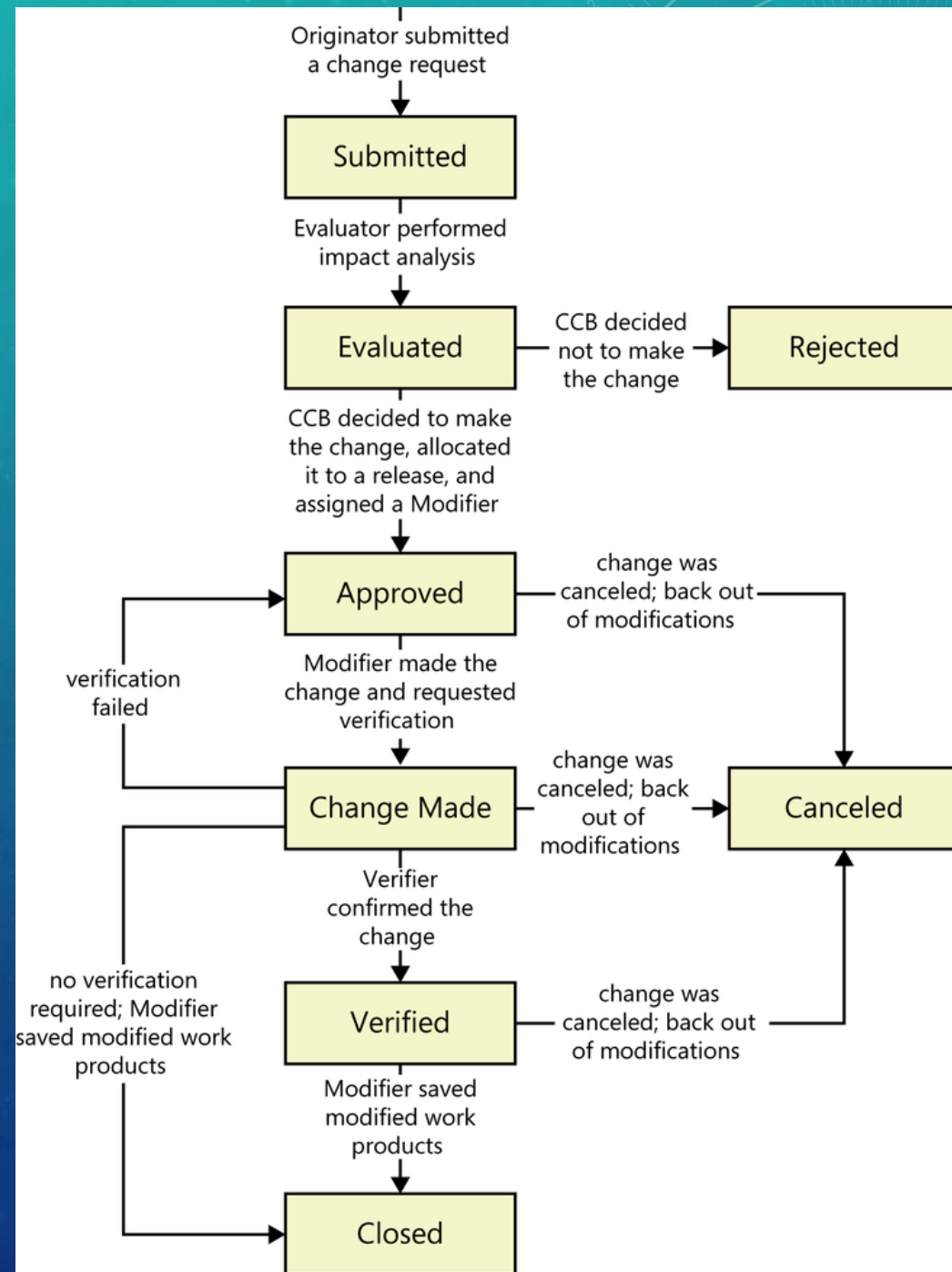
- 责任——根据拟议的变更或一系列变更及其合理性来决定变更。
- 角色——谁将参与以及参与者的职责是什么

角色	描述和职责
椅子	变更控制委员会主席；通常拥有最终决定权。 如果变更控制委员会 (CCB) 未能达成一致意见，则由该机构做出决定；并确定每项变更请求的评估人和修改人。
评估员	CCB主席要求分析拟议变更的影响（成本、范围、影响、资源）的人员
修饰符	负责根据已批准的变更请求对工作成果进行修改的人员
鼻祖	提交新变更请求的人员（业务分析师）
要求接收者	最初接收新提交的变更请求的人员（CCB秘书）
验证者	负责判定变更是否正确的人员
建交银行	负责决定批准或拒绝特定项目拟议变更的小组



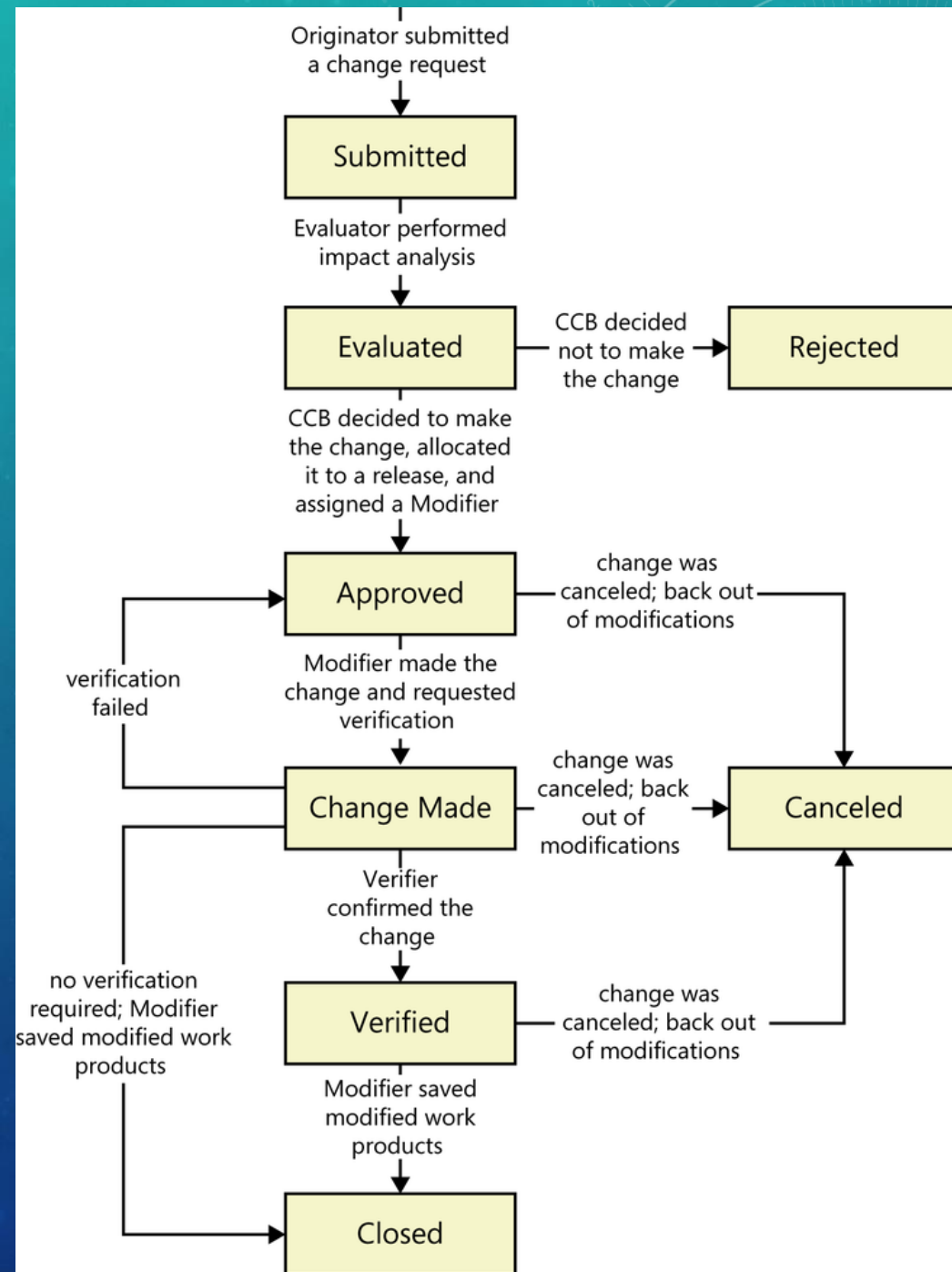
# 流程和变更控制委员会（CCB）

- CCB流程（矩形表示变更状态）
  - 准入条件——流程执行开始前必须满足的条件
- CCB的各项任务：
  - 评估变更请求：技术可行性、成本、与项目业务需求和资源限制的一致性、影响分析、风险和危害分析
  - 决定是否接受或拒绝变更，如果接受，则确定优先级、目标实施日期，以及将变更分配到哪个迭代或版本。
  - 实施变更
  - 验证——依据一系列标准



# 流程和变更控制委员会（CCB）

- CCB流程（继续）
  - 退出标准，即表明流程成功完成的条件，例如：
    - 请求状态为：已拒绝、已关闭或已取消。
    - 所有修改后的工作成果都会更新并存储在正确的位置。
    - 相关利益方已被告知变更详情和变更请求的状态。
- 报告——



# 流程和变更控制委员会（CCB）

## 来自成员

- 项目或项目群管理（范围、资源）
- 业务分析或产品管理（成本）
- 开发（可行性）
- 测试或质量保证（可追溯性）
- 市场营销部门、应用程序所服务的企业或客户代表（影响）
- 技术支持或服务台（影响）

## 决策

- CCB会员资格
- 要使用的决策规则
- 建管会主席能否推翻建管会的集体决定
- 建行会主席或管理层是否需要批准该小组的决定





# 变更影响分析

## 影响分析包括三个步骤：

- 第一步：了解变更可能带来的影响。需求变更通常会产生较大的副作用，导致：其他方面的修改需求、架构、设计、代码和测试。变更可能导致冲突可能存在其他要求，或者会损害质量属性，例如性能或安全性。

(帮助评估者理解接受……的含义的问题清单)  
(拟议变更)

- ☐ Will the change enhance or impair the ability to satisfy any business requirements?
- ☐ Do any existing requirements in the baseline conflict with the proposed change?
- ☐ Do any other pending requirements changes conflict with the proposed change?
- ☐ What are the business or technical consequences of not making the change?
- ☐ What are possible adverse side effects or other risks of making the proposed change?
- ☐ Will the proposed change adversely affect performance or other quality attributes?
- ☐ Is the proposed change feasible within known technical constraints and current staff skills?
- ☐ Will the proposed change place unacceptable demands on any resources required for the development, test, or operating environments?
- ☐ Must any tools be acquired to implement and test the change?
- ☐ How will the proposed change affect the sequence, dependencies, effort, or duration of any tasks currently in the project plan?
- ☐ Will prototyping or other user input be required to validate the change?
- ☐ How much effort that has already been invested in the project will be lost if this change is accepted?
- ☐ Will the proposed change cause an increase in product unit cost, such as by increasing third-party product licensing fees?
- ☐ Will the change affect any marketing, manufacturing, training, or customer support plans?

# 变更影响分析

- 步骤 2：识别所有可能的需求、文件、模型和文档必须进行修改如果团队采纳了所要求的更改。

- ☐ Identify any user interface changes, additions, or deletions required.
- ☐ Identify any changes, additions, or deletions required in reports, databases, or files.
- ☐ Identify the design components that must be created, modified, or deleted.
- ☐ Identify the source code files that must be created, modified, or deleted.
- ☐ Identify any changes required in build files or procedures.
- ☐ Identify existing unit, integration, and system tests to be modified or deleted.
- ☐ Estimate the number of new unit, integration, and system tests needed.
- ☐ Identify help screens, training or support materials, or other user documentation that must be created or modified.
- ☐ Identify other applications, libraries, or hardware components affected by the change.
- ☐ Identify any third-party software to be acquired or modified.
- ☐ Identify any impact the proposed change will have on the project management plan, quality assurance plan, configuration management plan, or other plans.

(有助于识别所有可能受此变更影响的软件元素和其他工作成果的问题)



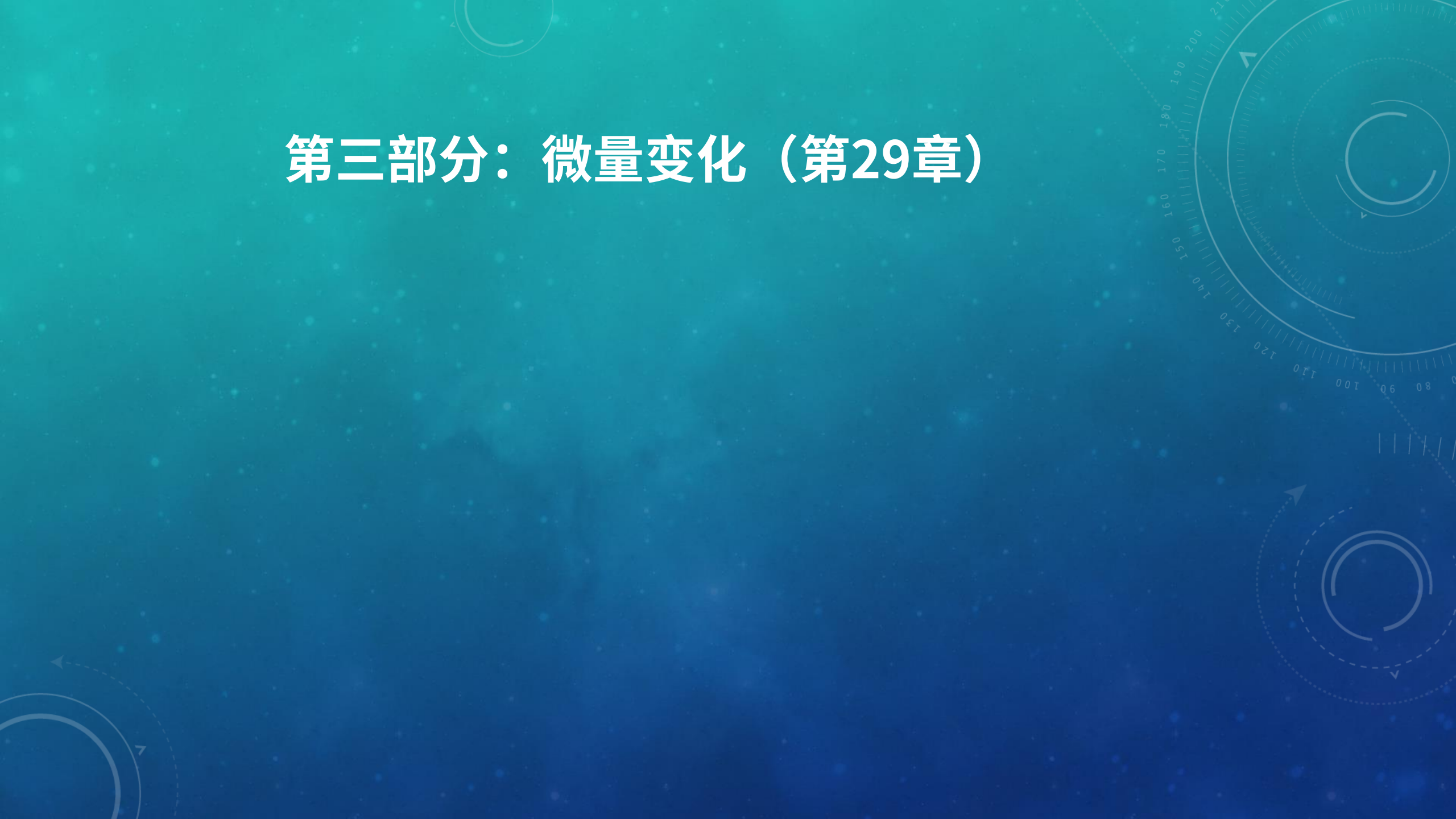
# 变更影响分析

- 步骤 3：确定实施变更所需的任务，并进行估算努力需要完成这些任务。

Hours	Task
_____	Update the SRS or requirements repository
_____	Develop and evaluate a prototype
_____	Create new design components
_____	Modify existing design components
_____	Develop new user interface components
_____	Modify existing user interface components
_____	Develop new user documentation and help screens
_____	Modify existing user documentation and help screens
_____	Develop new source code
_____	Modify existing source code
_____	License and integrate third-party software
_____	Modify build files and procedures
_____	Write new unit and integration tests
_____	Modify existing unit and integration tests
_____	Perform unit and integration testing after implementation
_____	Write new system and acceptance tests
_____	Modify existing system and acceptance tests
_____	Modify automated test suites
_____	Perform regression testing
_____	Develop new reports
_____	Modify existing reports
_____	Develop new database elements
_____	Modify existing database elements
_____	Develop new data files
_____	Modify existing data files
_____	Modify various project plans
_____	Update other documentation
_____	Update the requirements traceability matrix
_____	Review modified work products
_____	Perform rework following reviews and testing
_____	Other tasks
_____	<b>Total Estimated Effort</b>



# 第三部分：微量变化（第29章）



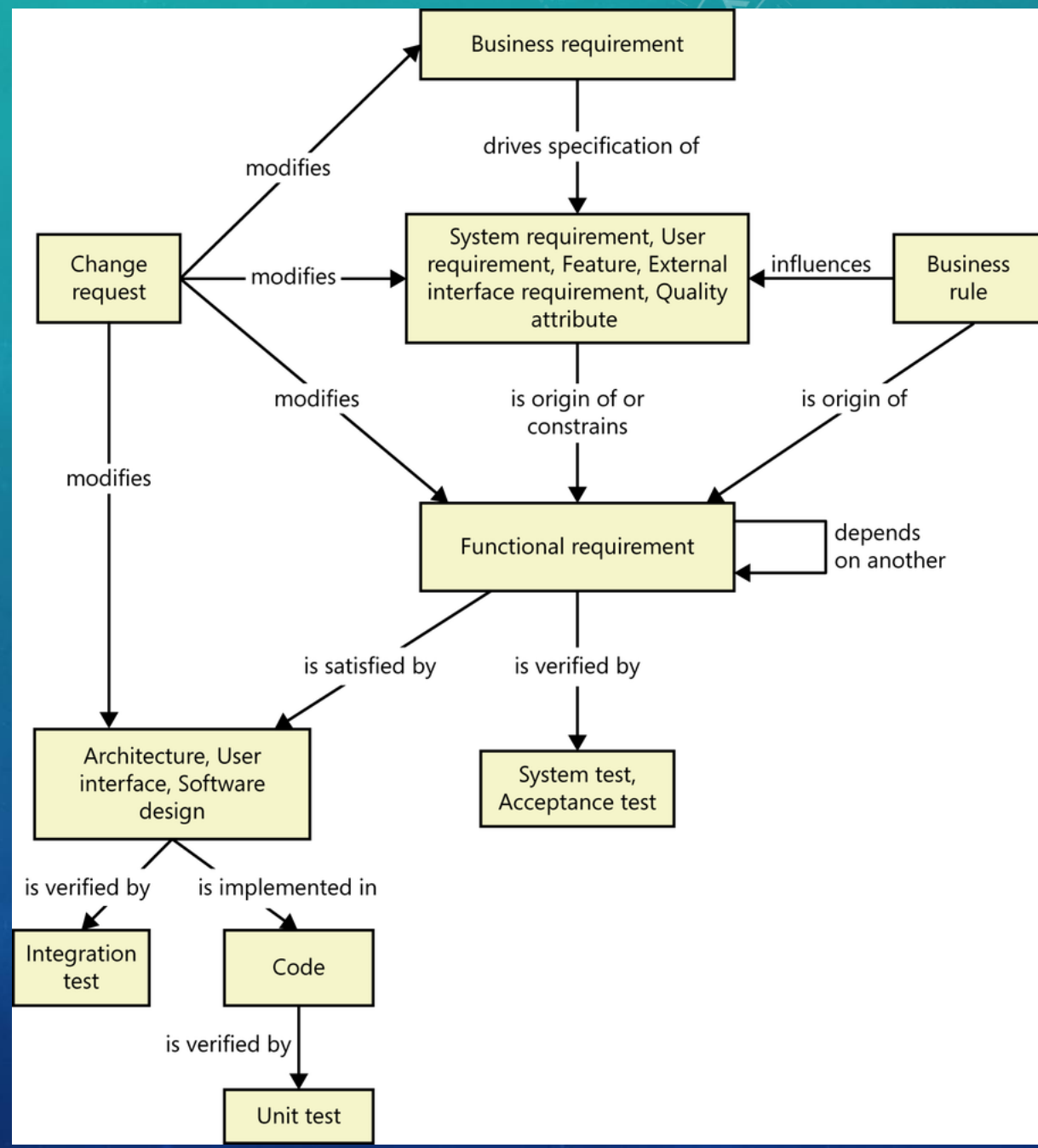
# 好处

- 需求变更看似简单，但往往会产生深远的影响（例如，范围界定可能并不明显）。
- 追踪变化会带来以下好处：
  - 查找缺失的需求。
  - 通过变更影响分析，比较变更引入前后的情况，找出不必要的需求。
  - 认证和合规性——追踪信息可以证明所有要求都已得到落实。
  - 维护——可靠的跟踪信息表明更改已正确完成。
  - 项目跟踪——跟踪信息能够准确记录计划功能的实施状态。
  - 重组。
  - 重用——追踪信息，识别相关需求、设计、代码和测试的软件包
  - 测试——测试、需求和代码之间的联系可以指出可能存在缺陷的领域。



# 显示可追溯性

- 图表显示了不同类型需求之间的可追溯性关系，以及上游变化对下游组件的影响。
- 例如，测试人员对测试结果感到困惑，可能意味着功能需求需要进行一些更改。
- 另一个例子是，如果架构不再满足功能需求，则功能需求可能会发生一些变化。





# 显示可追溯性

- 需求可追溯性矩阵展示了每个功能需求如何反向链接到特定的用例，以及如何正向链接到一个或多个设计、代码和测试元素，例如：

用户要求	功能要求	设计元素	代码元素	测试
UC-28	catalog.query.sort	课程目录	CatalogSort()	搜索.7 搜索.8
UC-29	目录查询导入	课程目录	CatalogImport() CatalogValidate()	搜索.12 搜索.13 搜索.14

# 需求跟踪活动（良好实践）

- 向团队和管理层解释需求跟踪的概念和重要性。
- 例如，解释上一张幻灯片中的图表中选定的链接关系。
- 解释可追溯性矩阵，例如上一张幻灯片上的表格。
- 确定需要维护可追溯信息的软件部分。首先从关键核心功能、高风险部分或在产品生命周期内维护和演进最多的部分入手。
- 确定提供每种类型链接信息的个人，以及协调追踪活动和管理数据的人员（很可能是业务分析师）。
- 修改开发流程，提醒开发人员在实施需求或批准变更后更新链接。
- 定义标签规则，为每个系统元素赋予唯一的标识符，以便将它们链接在一起。
- 随着开发的进行，让每个参与者在完成小部分工作后提供所需的跟踪信息。