



要求  
执行

# 第一部分：需求流程改进

(CH31)



# 为什么要改进？

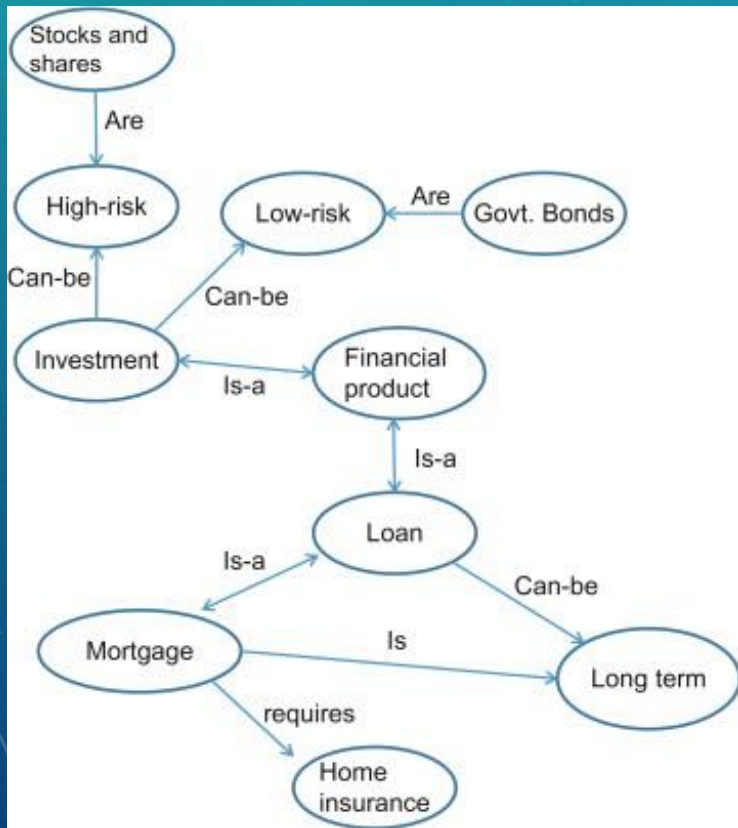


- 改进的本质在于解决问题。如果当前的需求流程运行良好，那就没有必要进行更改，不要浪费时间。
- 如果确实存在问题，那么“吸取教训”就很有道理。这将使你能够……
  - 回想一下这些问题遇到目前/之前的项目（加班、超预算、大量临时变更、大量返工等）
  - 弄清楚这些问题是如何得到解决的。（良好做法）
  - 预先防范未来项目中可能会遇到的问题（首先找出原因）。
  - 采纳良好做法比以前项目中使用的方法更高效、更有效，或者尝试不同的做法。
- 需求流程的改进将使软件开发生命周期的所有其他部分和利益相关者受益，因为需求流程与软件开发项目的其他流程以及所有利益相关者都相关。
- 它还可能对其他项目流程/利益相关者产生副作用。
- 这需要各方的支持/承诺，尤其是……

# 原因：从与项目其他部分的关系角度来看

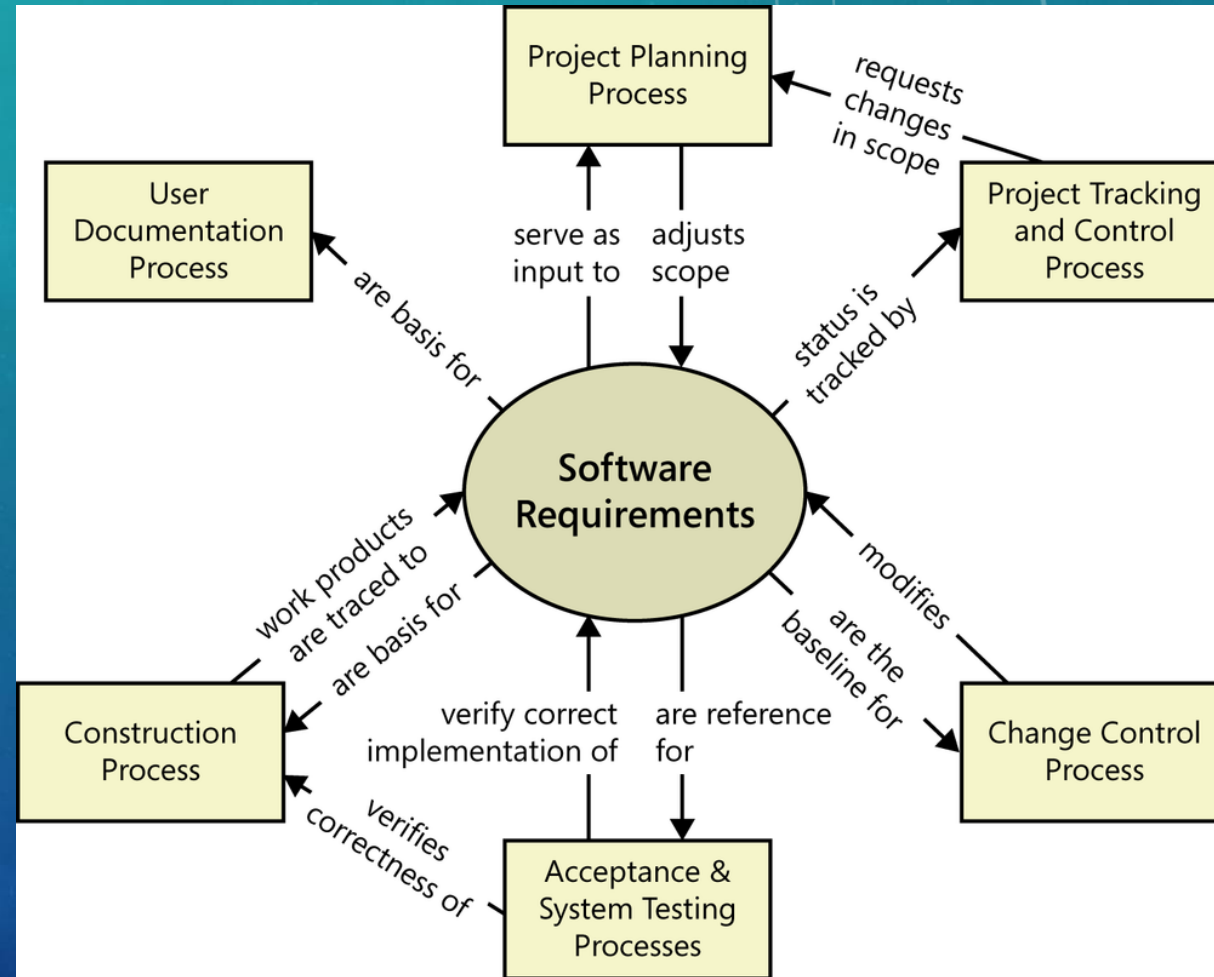
## 背景

- 语义网络代表知识，属于人工智能领域。它以网络形式展示关于对象的知识以及对象之间的关系，即语义关系。
- 例子：<https://www.sciencedirect.com/topics/computerscience/semantic-network>



# 原因：从与项目其他部分的关系角度来看

- **项目规划**需求是项目规划流程的基础。规划人员根据需求选择合适的软件开发生命周期，并制定资源和进度估算。项目规划还会指出，某些功能可能无法在现有资源和时间范围内交付。规划流程可能会导致项目范围的缩减。
- **项目跟踪与控制**项目跟踪包括监控项目状态与需求状态的对比情况（请参阅需求状态跟踪）。
- **变更控制**所有后续变更和新增内容均应通过既定的变更控制流程进行。经批准的变更将被添加到需求列表中。





# 原因：从与项目其他部分的关系角度来看

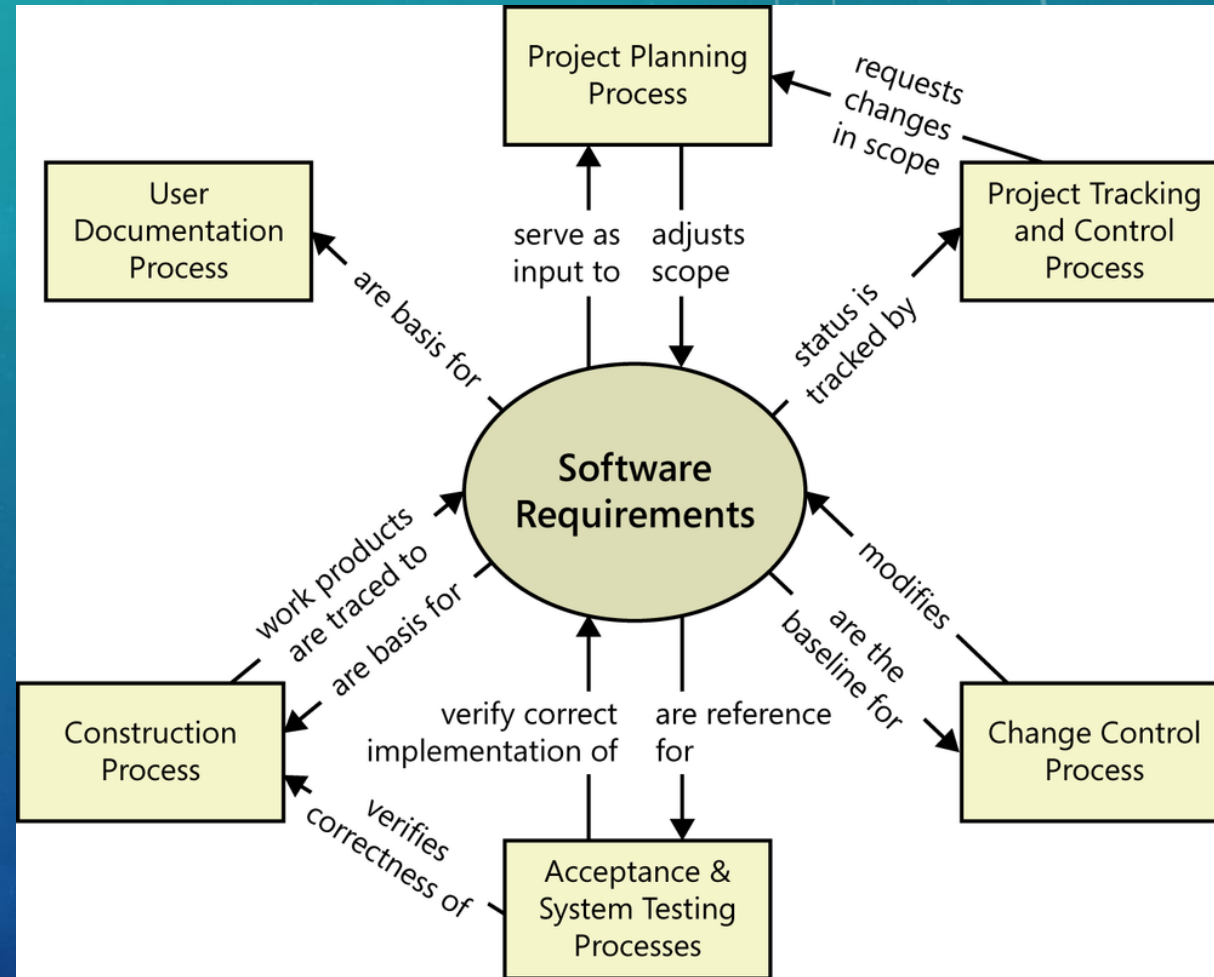
- 验收和系统测试用户需求和功能需求分别是验收测试和系统测试的重要输入。后者为……提供信息。

需求流程，用于判断需求是否已得到验证。

- 建造需求是设计和实施工作的基础。

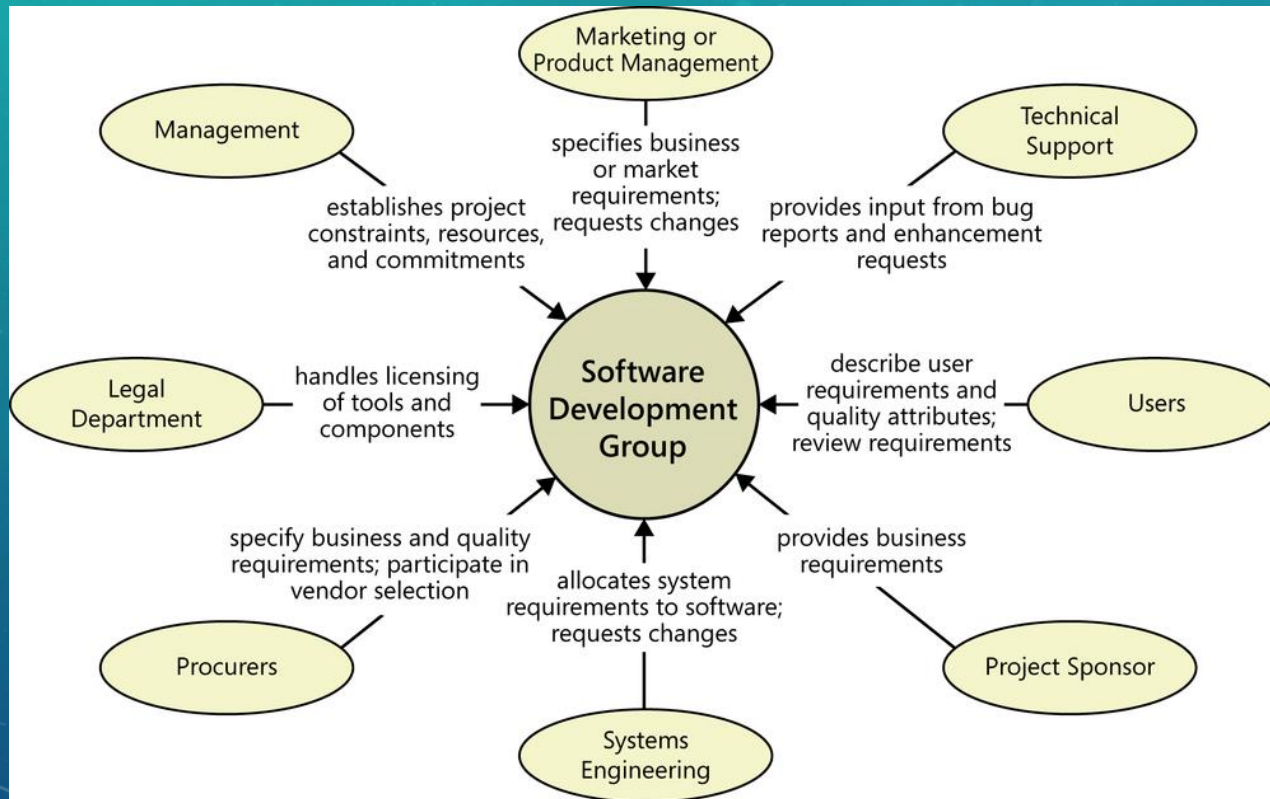
施工/开发（或者说设计和编码）实现需求。

- 用户文档产品需求为用户文档（如用户手册）提供输入。



# 如何改进？——与利益相关者的关系

不同利益相关者对需求的贡献或意见



需要确保利益相关者对改进工作的承诺，特别是对所有管理工作的承诺。

利益相关者组织。

反抗：

- 那些忙得根本没时间完成项目工作的人。
- 改进后的新流程可能会被视为阻碍变革的因素（别费劲了，太麻烦了）。
- 开发人员/管理人员认为编写和审查需求是浪费时间的官僚行为。

# 如何改进？——与利益相关者的关系

这种情况确实令人担忧。

大家都一致认为，最近几个项目进展并不顺利。作为首席业务分析师，乔安娜深知需求问题至少是部分问题的根源。各个项目的业务分析师在教育背景和经验水平上差异很大。他们各自采用不同的方法来开发和管理需求，只是尽力而为。他们组织需求的方式也各不相同。有些团队遵循有效的需求变更流程，从而减少了项目混乱；而另一些团队则对每一个变更请求都反应过度，不假思索地做出反应。整个团队都感到十分沮丧。

乔安娜曾尝试指导经验不足的业务分析师；有些人比其他人更乐于接受她的建议。乔安娜所在团队的一些团队在需求分析方面做得不错，这些项目遇到的问题也比其他团队少。乔安娜意识到，如果能提升所有团队的需求分析水平，那将是一件好事。或许现在正是认真改进需求分析实践的好时机。但是，其他业务分析师和他们的团队成员会配合吗？管理层真的致力于减少痛点吗？这次真的会有所改变吗？还是会像之前的改进计划一样，因大家的漠不关心而夭折？



# 如何改进？——与利益相关者的关系

## 管理层的承诺是关键

- 获得他们支持的迹象：

1. Asking that requirements for a project be documented in an appropriate form.
2. Working with the business analyst to provide business requirements for each project.
3. Expecting requirements to be reviewed by appropriate stakeholders, including themselves when appropriate.
4. Asking stakeholders to agree on requirements before implementing each portion of the solution.
5. Ensuring that project plans include time and resources for requirements tasks.
6. Collaborating with other key stakeholders to gain their participation in requirements activities.
7. Establishing effective mechanisms and policies to handle requirements changes.
8. Investing in training, tools, books, and other resources for those involved in requirements activities.
9. Funding and staffing activities to improve the organization's requirements processes.
10. Making the time available for team members to spend on requirements process improvement activities.

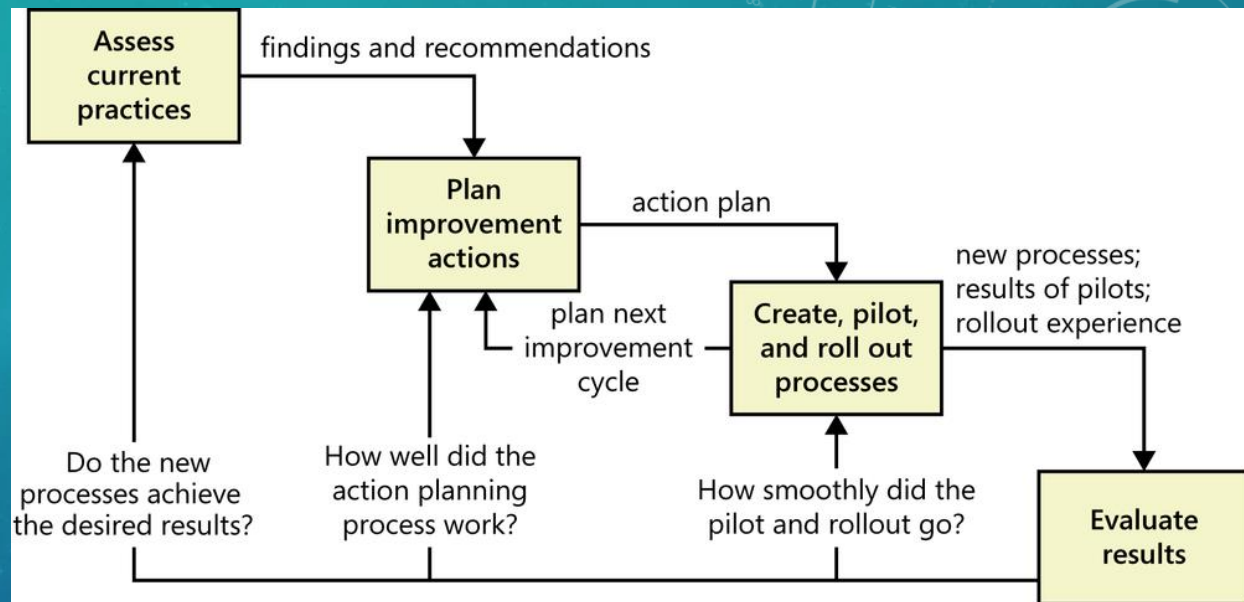
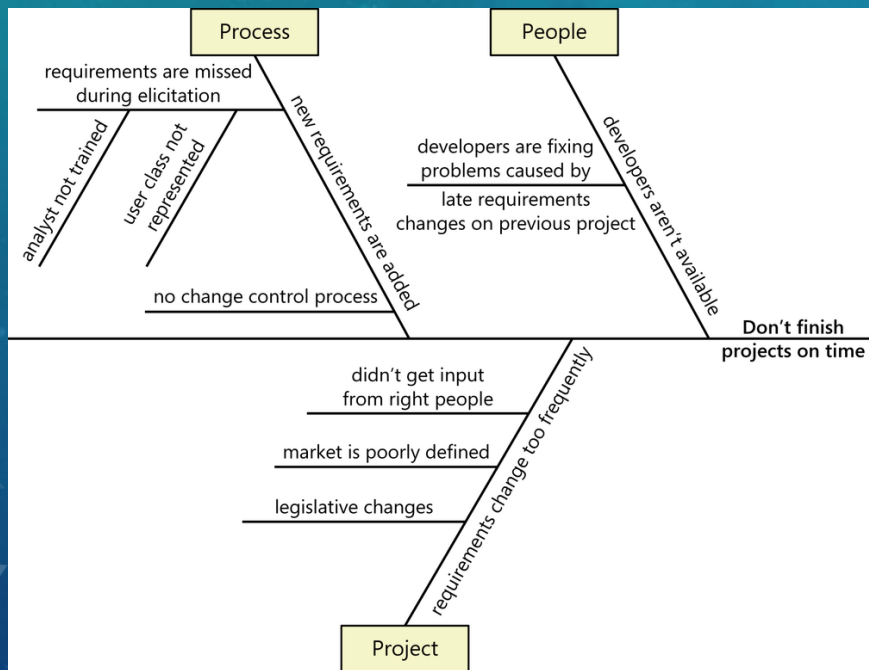
# 如何改进？——改进原则

- 流程改进应该是渐进式和持续性的。
- 只有当人们和组织有改进的动力时，他们才会做出改进。
  - 项目未能按时完成，原因是需求超出预期。
  - 由于需求被误解或含糊不清，开发人员不得不加班加点。
  - 系统测试工作白费了，因为测试人员不了解产品应该做什么。
  - 功能虽然具备，但由于运行缓慢、易用性差或其他质量缺陷，用户并不满意。
- 由于客户提出了许多本可以在需求收集阶段就发现的改进要求，该公司经历了高昂的维护成本。
- 项目过程中需求变更未能得到适当落实，因此交付的解决方案未能满足客户需求。
- 由于多个业务分析师同时处理需求，且没有版本控制流程，导致需求修改丢失或被覆盖。
- 客户无法联系以澄清和完善需求。
- 与需求相关的问题未能及时解决，导致返工。
- 流程改进应该**目标导向**
- 将你的改进活动视为小型项目



# 如何改进？——改进过程周期

- 第一步：评估现有做法
  - 根本原因分析（鱼骨图分析）——根本原因分析是指连续多次追问问题“为什么”存在，每次都深入探究前一个“为什么”问题的答案背后的根本原因。（问题和原因是两回事。）

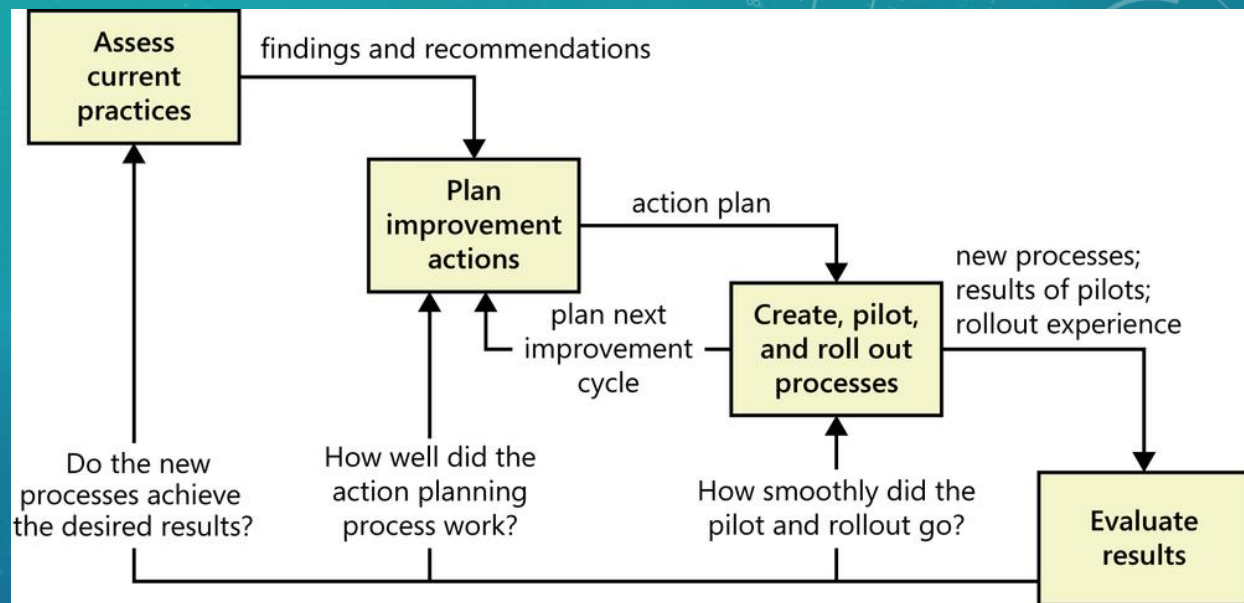




# 如何改进？——改进过程周期

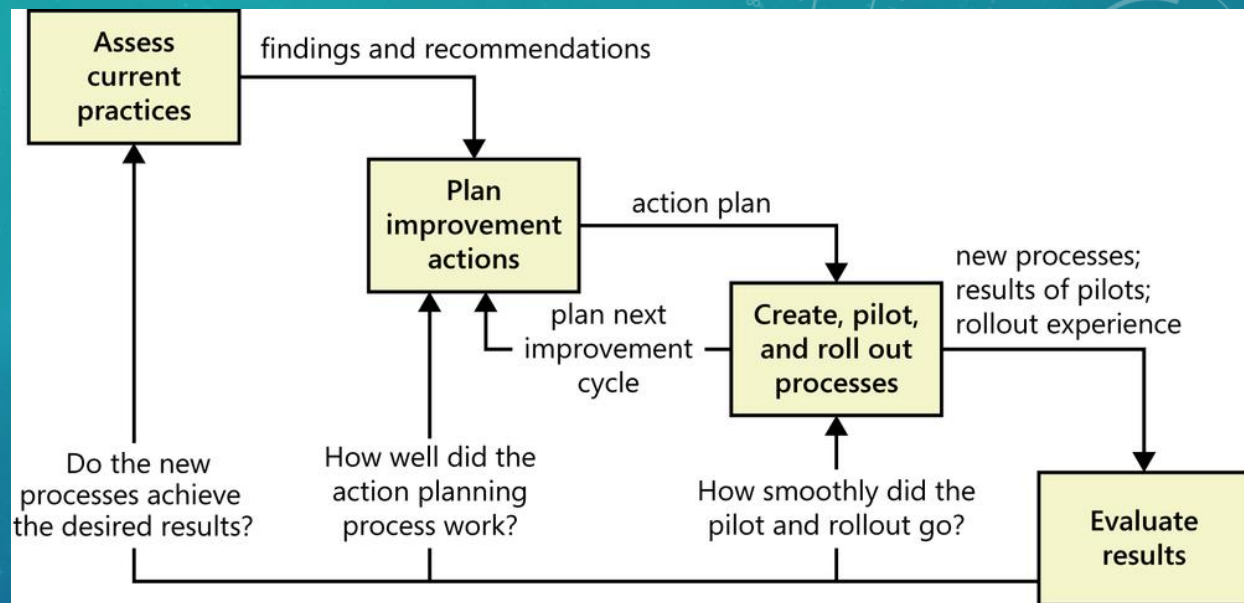
- 步骤二：规划

- 拟定一份改进方案。
- 审阅并修改提案。
- 在项目 A 中试点改进措施。
- 根据试点项目的反馈意见，修改变更内容和行动方案。
- 评估问题跟踪工具，并选择一款工具来支持变更控制流程。
- 采购问题跟踪工具，并对其进行定制以支持变更控制流程。



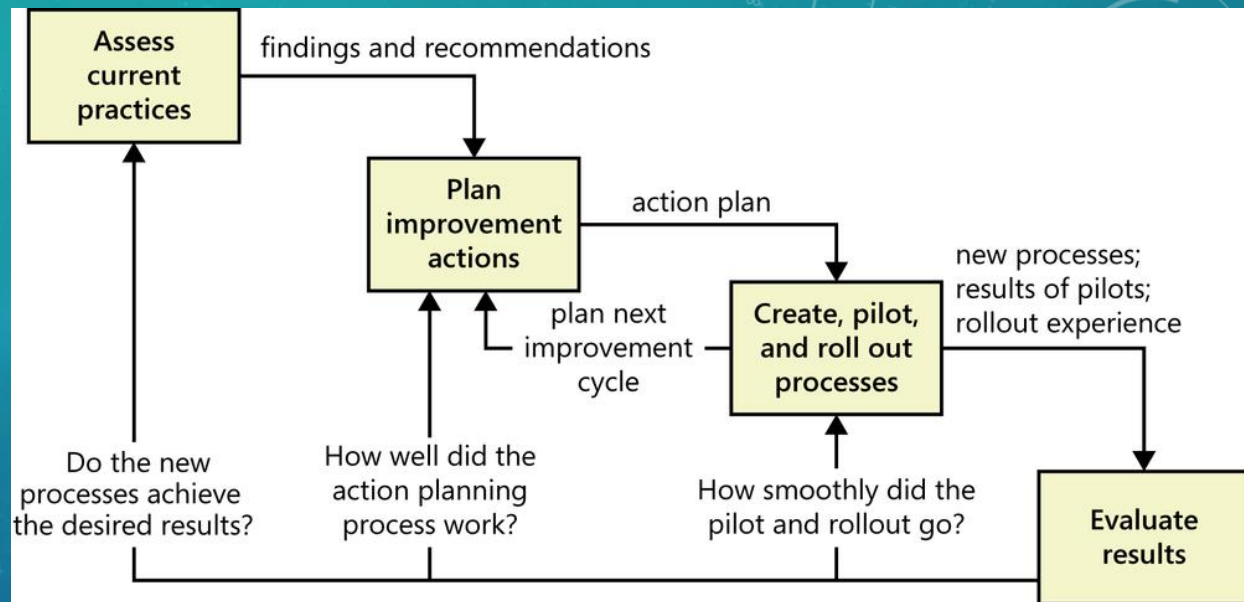
# 如何改进？——改进过程周期

- 步骤 3：实施试点项目是个好主意。
  - 选择一些试点参与者，让他们公平地尝试新方法并提供有益的反馈。这些参与者可以是支持者，也可以是怀疑者，但都不应该强烈反对改进工作。
- 量化团队将用于评估试点结果的标准。
- 确定哪些利益相关者需要了解试点项目及其开展原因。
- 考虑在不同的项目中试行部分新流程。这能让更多人参与尝试新方法，从而提高认知度、收集反馈并获得认可。
- 将新措施和工具推广到公司各部门。



# 如何改进？——改进过程周期

- 第四步：评估
  - 关键步骤是评估新流程是否产生了预期的结果。
  - 作为评估的一部分，询问试点参与者，如果他们不得不回到以前那种工作方式，他们会有什么感受。



循环！！

这并非线性过程，而是一个迭代过程。

- 步骤 3 可以用于制定下一步改进计划。
- 第 4 步可以改善之前所有步骤的效果。



## 第二部分：风险管理（第32章）

# 什么是风险？

- 这可能性某事坏事发生（剑桥词典）
- 风险是指可能造成损失或以其他方式威胁项目成功的状况。
- 风险与回报——风险回报关系基于这样的概念：投资本金损失的风险越高，本金增加或本金收益率提高的潜在回报就越大。
- 在软件工程中，风险会导致项目失败。
- 获奖的条件是按时按预算交付软件产品，从而获得丰厚的利润。
- 日志中的节点宕机风险

**ID:**

*<sequence number>*

**Submitter:**

*<individual who brought this risk to the team's attention>*

**Date Opened:**

*<date the risk was identified>*

**Date Closed:**

*<date the risk was closed out>*

**Risk Statement:**

*<statement of the risk in the form "condition-consequence">*

**Scope of Impact:**

*<project teams, business areas, and functional areas the risk could affect>*

**Probability:**

*<the likelihood of the risk becoming a problem>*

**Impact:**

*<numerical rating of the potential damage if the risk does become a problem>*

**Exposure:**

*<probability multiplied by impact>*

**Risk Management Plan:**

*<one or more approaches to control, avoid, minimize, or otherwise mitigate the risk>*

**Contingency Plan:**

*<course of action to follow if the risk management plan is not effective>*

**Owner:**

*<individual responsible for resolving the risk>*

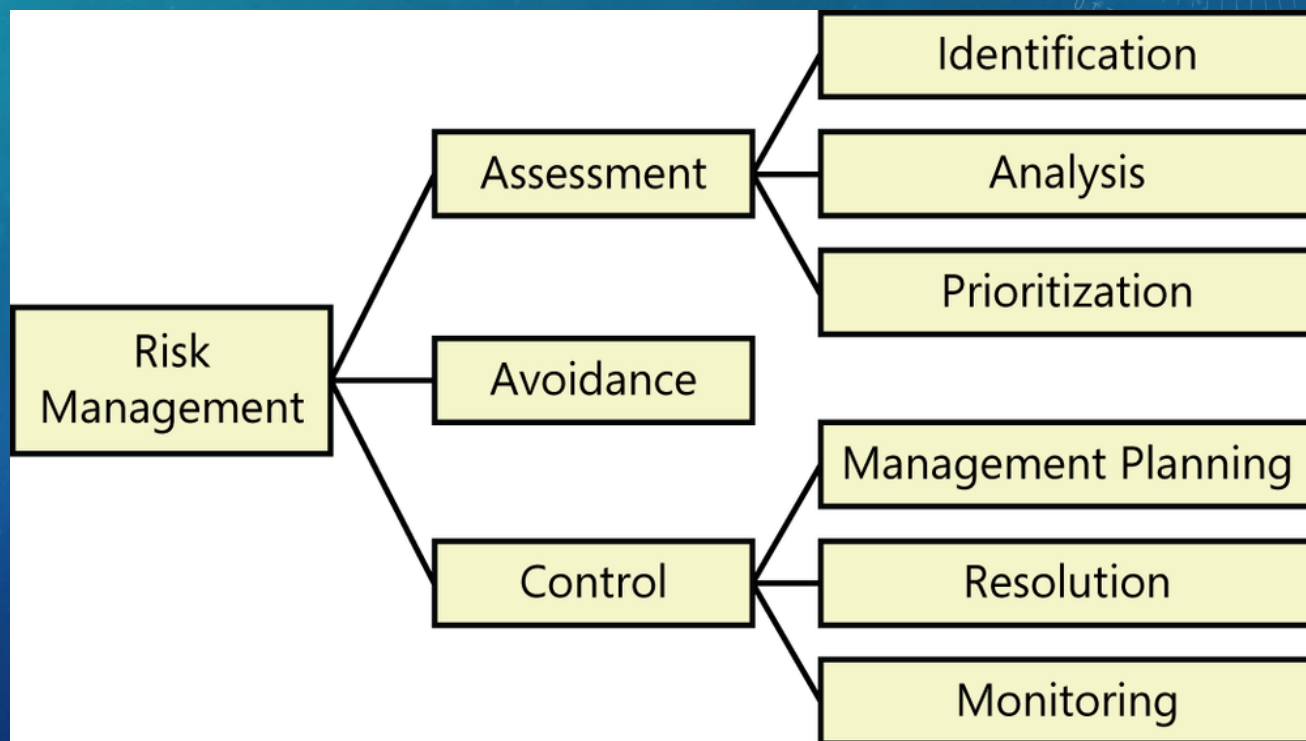
**Date Due:**

*<date by which the mitigation actions are to be implemented>*

# 风险管理

软件风险管理是风险与回报之间的平衡。 **风险管理流程**

- 第一步：识别风险。发现、识别并描述可能影响您的项目或其结果的风险。
- 第二步：分析风险。确定每项风险发生的可能性和后果。
- 步骤 3：评估或排序风险。通过确定风险程度来评估或排序风险，风险程度是可能性和后果的结合。
- 第四步：应对风险。评估风险等级最高的风险，并制定计划来应对或调整这些风险，以达到可接受的风险水平。（应急计划和风险评估）  
（迁移）
- 步骤五：监控和审查风险。使用项目风险登记册来监控、跟踪和审查风险。





# 需求工程中的风险

## 在需求获取过程中

- 范围蔓延
  - 原因——利益相关者在项目初期并没有就产品应该是什么以及应该做什么达成共识，而是逐渐地加入进来。
  - 处理方法——制定产品愿景和项目范围文件，并以此为参考来解读需求，严格遵循CCB（内容控制框架）。
- 利益相关者之间的期望差距
  - 原因——客户互动混乱
  - 处理方法——为每个客户群体指定一名代表，并用通俗易懂的语言描述每个用例。
- 需求规格说明的不完整性和错误性
  - 原因：客户自身对业务目标不够清晰
  - 如何处理——将用户需求与业务目标进行映射，并让客户决定验收标准



# 需求工程中的风险

## 需求获取（续）

- 对创新产品的市场反应估计错误
  - 原因——这些产品是同类产品中的首创。
  - 如何处理——通过市场调研、原型制作和用户焦点小组访谈来获取早期反馈
- 忽略非功能性需求
  - 原因——过分关注功能需求
  - 处理方法——向客户询问质量属性并记录非功能性需求
- 部分客户对需求文档不满意。
  - 原因：缺乏沟通和对需求的共识
  - 处理方法——重点关注主要客户，他们通常是用户



# 需求工程中的风险

## 需求获取（续）

- 未明示的要求
  - 原因在于——客户通常抱有隐性期望，并认为业务分析师了解这些隐性期望。
  - 处理方法——使用开放式问题来获取客户更多想法
- 解决方案以需求的形式呈现
  - 原因——客户对解决方案有自己的想法
  - 如何处理——UML用例图
- 业务部门与开发团队之间缺乏信任
  - 原因——感受到可能被裁员的威胁
  - 如何处理——？





# 需求工程中的风险

## 在需求分析中

- 需求优先级排序不当
  - 原因在于——完全记录客户反馈，因此缺乏分析。
  - 如何处理——恰当的权衡决策
- 技术上难以实现的功能
  - 原因——可行性研究不足
  - 如何处理——跟踪需求实施情况，以便尽早识别那些进度落后的需求。
- 学习不熟悉的技术、方法、语言、工具或硬件需要更长时间
  - 原因：不熟悉的技术、方法、语言、工具或硬件
  - 如何处理——尽早识别高风险组件，以便开发团队学习新知识。



# 需求工程中的风险

## 在需求规范中

- 不同的需求理解
  - 原因：开发人员和客户对需求的理解不同
  - 如何处理——客户、开发人员和测试人员的同行评审
- 未决问题（待定）
  - 原因——时间压力
  - 如何处理——明确谁负责解决这些问题
- 含糊不清的术语
  - 原因——自然语言
  - 如何处理——业务和技术术语定义词汇表
- 需求设计
  - 原因——客户认为他们有新的解决方案
  - 如何处理——请客户强调他们的需求



# 需求工程中的风险

## 在需求验证中

- 未经验证的需求
  - 原因——冗长的需求文档
  - 如何处理——确保遵循 V 模型并进行正式/非正式的同行评审。
- 缺乏检查经验
  - 原因——新人员加入检查组是不可避免的。
  - 如何应对——培训

## 在需求管理中

- 不断变化的需求 – CCB
- 需求变更控制 – CCB
- 未实现的需求——需求跟踪
- 扩大项目范围 – CCB

