



四个W

第一部分：什么（CH1）

什么是软件需求吗？

用户（或最终用户）是指将要使用……的人。
软件系统。

- 术语：用户需求、软件需求、业务需求、功能需求、系统需求、产品需求、项目需求、用户故事、特性、约束等。

- 定义：需求是对应该实现的内容的规范。它们是对系统行为方式、系统属性或功能的描述。

属性。它们可能会对系统的开发过程构成限制。

- 需求类型（参见列表）

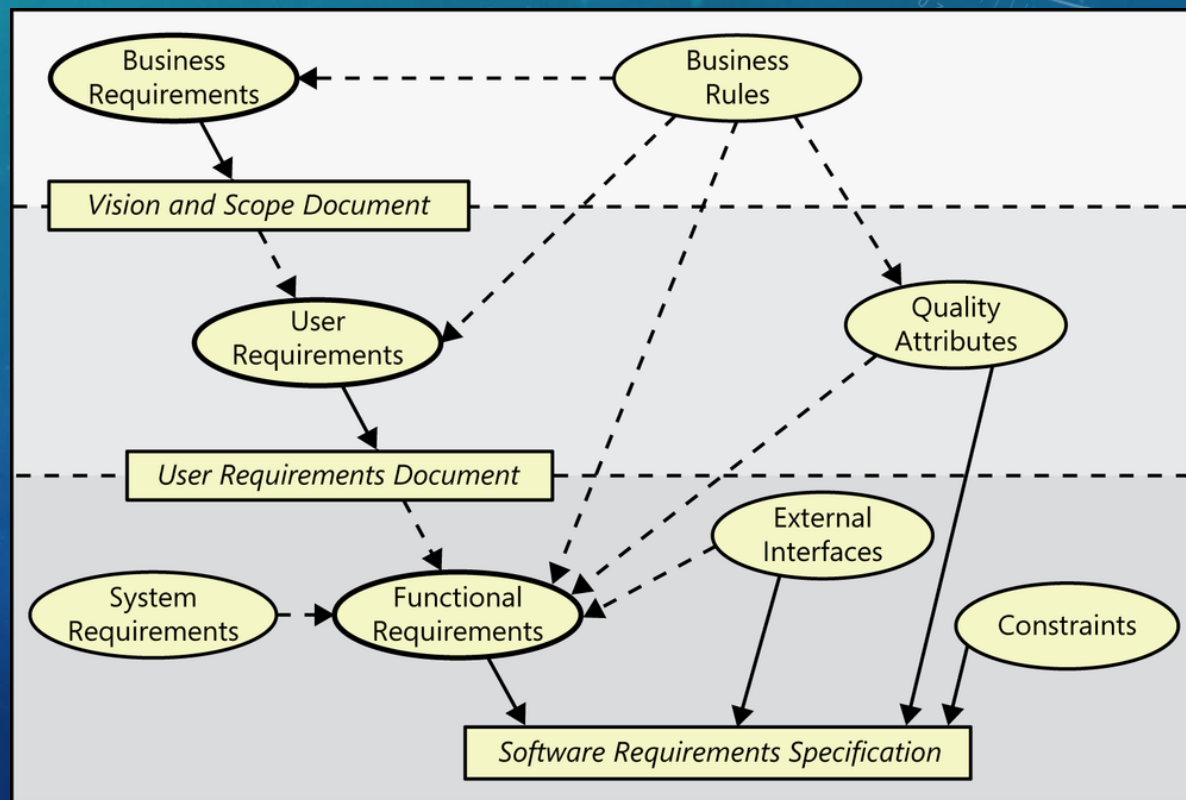
学期	定义提供端到端的IT解决方案和网络支持
业务需求	生产该产品的组织或购买该产品的客户所设定的高层业务目标。
业务规则	用于定义或约束业务某些方面的政策、指南、标准或法规。它本身并非软件需求，但却是多种软件需求的来源。
约束	对开发人员在产品设计和制造过程中可选择的方案施加的限制。
外部接口要求	对软件系统与用户、其他软件系统或硬件设备之间连接的描述。
特征	一个或多个逻辑相关的系统功能，可为用户提供价值，并由一组功能需求进行描述。
功能需求	对系统在特定条件下将表现出的行为的描述。
功能不全要求	对系统必须展现的属性或特征，或系统必须遵守的约束条件的描述。
质量属性	一种非功能性需求，用于描述产品的服务或性能特征。
系统要求	对于包含多个子系统的产品而言，最高级别的需求是软件或软硬件结合。
用户需求	特定类别的用户必须能够使用以下方式完成的目标或任务：系统，或所需的产品属性。

什么是软件需求吗？

软件需求包括三个不同的层次：业务需求、用户需求和功能需求。

- 最高层面：愿景和范围——业务目标和预期收益，说明开发该软件的原因（为什么）
- 2和层级：用户需求——要执行的任务以及软件属性和特性（内容）
- 3rd级别：软件规范 --
 - 功能需求——系统行为
 - 系统需求——系统结构
 - 外部接口——与其他系统的连接
 - 限制——开发人员可以做什么和不能做什么

例如：讲义中提供了伦敦救护车服务（LAS）计算机辅助调度系统（CAD）的示例。它解释了一些类型的需求。



什么是软件需求吗？

例如：伦敦大学现在是伦敦众多“小型”大学的统称。其中一所小型大学一直在将其学生管理系统（基本上是一个由数据库和报告工具组成的数据系统）从 Agresso 更新到 SITS。

- 业务需求——解决 Agresso 的所有问题（这两个系统功能相似，Agresso 更适合规模较小的大学。由于所谓的 IT 人员并不真正了解招生、注册、学术等方面的业务规则，因此没有演示 Agresso 的功能。） 通过流程自动化减少学生管理中的人为错误
- 用户需求——使用该系统的员工的需求，包括招生、注册、学术部门管理、学术人员等。
- 系统规范或功能——学生档案的创建和维护，为每位学生创建每个学期的模块，学生成绩发布、进度安排、奖励（1）英石，2和上部，2和较低，3rd（且不设奖项），时间表，图形用户界面等。
- 限制因素——GDPR、安全、访问权限、EDI 等。

什么是软件需求吗？

两种不同的方案（商业模式）：

- 一家软件公司为客户开发软件。（有些软件公司有自己的专业领域，并为客户开发该领域的软件。）
 - 经理、市场人员和用户都在客户端。
 - BA是公司与其客户之间的接口。
 - 例如：excoms 是一家中小企业，提供端到端的 IT 解决方案、计算机网络支持、网络安全解决方案。
- 一家软件公司，旨在根据客户的具体需求定制产品。
 - BA是连接公司CEO层和研发团队的桥梁。
 - 它也是客户与开发团队之间的接口。
 - 例如：Grassroots 是一家软件公司。它的主要产品是一款项目管理软件，该软件的设计理念源于公司自身的项目管理哲学。因此，它不仅销售产品，也销售其管理理念。

CW 任务 1、2

CW包含一系列以小组合作方式完成的任务。

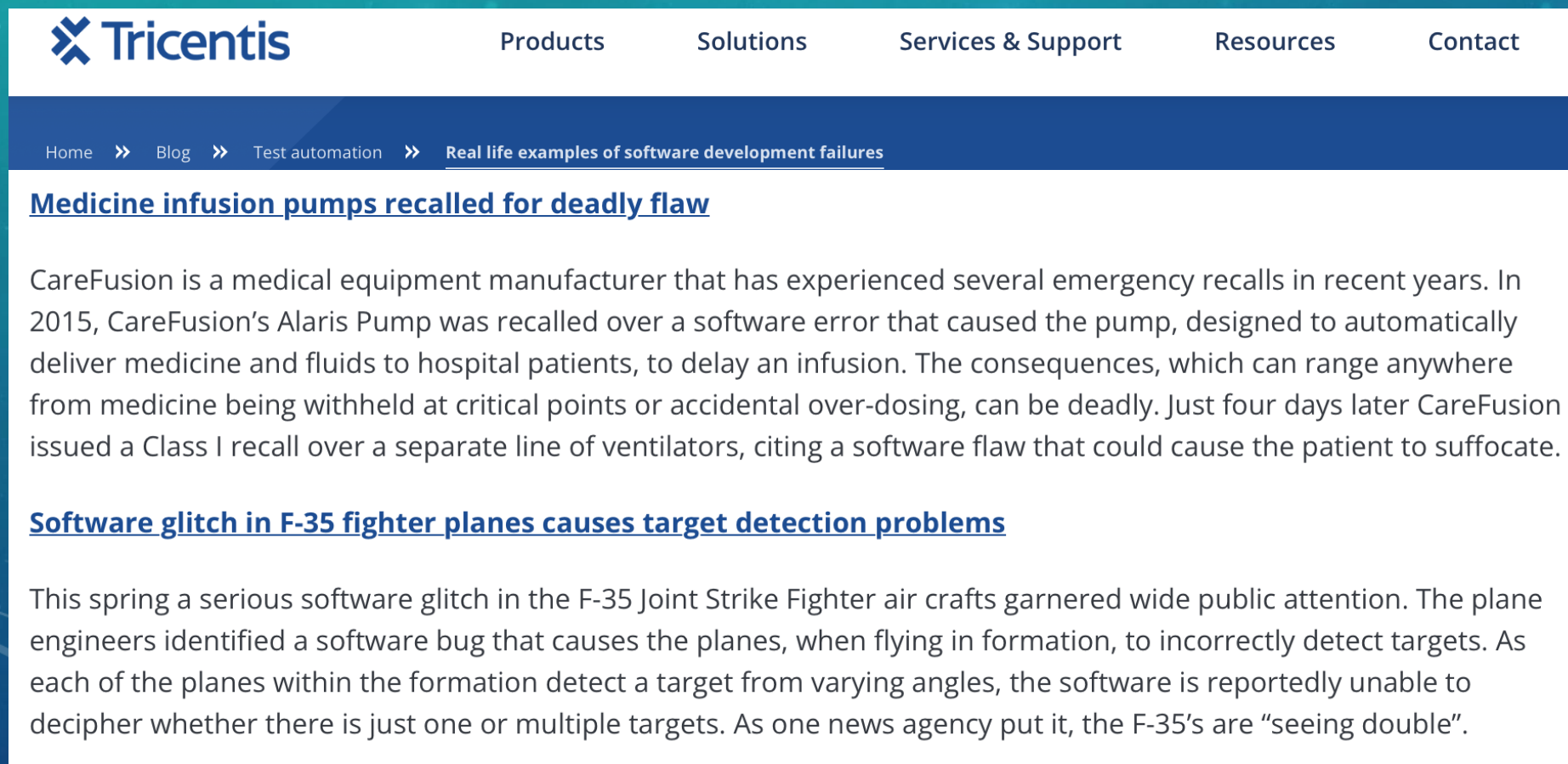
- 任务 1 至 3：
 - 小组由组员、组长和共同商定的方案组成
- 已商定的项目/方案的详细描述

学期	定义提供端到端的IT解决方案和网络支持
业务需求	生产该产品的组织或购买该产品的客户所设定的高层业务目标。
业务规则	用于定义或约束业务某些方面的政策、指南、标准或法规。它本身并非软件需求，但却是多种软件需求的来源。
约束	对开发人员在产品设计和制造过程中可选择的方案施加的限制。
外部接口要求	对软件系统与用户、其他软件系统或硬件设备之间连接的描述。
特征	一个或多个逻辑相关的系统功能，可为用户提供价值，并由一组功能需求进行描述。
功能需求	对系统在特定条件下将表现出的行为的描述。
功能不全要求	对系统必须展现的属性或特征，或系统必须遵守的约束条件的描述。
质量属性	一种非功能性需求，用于描述产品的服务或性能特征。
系统要求	对于包含多个子系统的产品而言，最高级别的需求是软件或软硬件结合。
用户需求	特定用户群体必须能够使用系统执行的目标或任务，或者所需的产品属性。

第二部分：为什么（第1章和第2章）

为什么软件需求重要吗？

- 多项研究表明，需求分析过程中引入的错误占软件产品所有缺陷的 40% 到 50%。（负面影响体现在：） <https://www.tricentis.com/blog/real-life-examples-of-software-development-failures/>）



The screenshot displays the Tricentis website's navigation bar with links for Products, Solutions, Services & Support, Resources, and Contact. Below the navigation bar, a breadcrumb trail shows the path: Home >> Blog >> Test automation >> Real life examples of software development failures. The main content area features two article titles: 'Medicine infusion pumps recalled for deadly flaw' and 'Software glitch in F-35 fighter planes causes target detection problems'. The first article describes a recall by CareFusion in 2015 due to a software error in the Alaris Pump, which could delay or withhold medicine, potentially leading to death. The second article discusses a software glitch in F-35 fighter planes that causes them to incorrectly detect multiple targets as a single one, referred to as 'seeing double'.

Tricentis Products Solutions Services & Support Resources Contact

Home >> Blog >> Test automation >> Real life examples of software development failures

Medicine infusion pumps recalled for deadly flaw

CareFusion is a medical equipment manufacturer that has experienced several emergency recalls in recent years. In 2015, CareFusion's Alaris Pump was recalled over a software error that caused the pump, designed to automatically deliver medicine and fluids to hospital patients, to delay an infusion. The consequences, which can range anywhere from medicine being withheld at critical points or accidental over-dosing, can be deadly. Just four days later CareFusion issued a Class I recall over a separate line of ventilators, citing a software flaw that could cause the patient to suffocate.

Software glitch in F-35 fighter planes causes target detection problems

This spring a serious software glitch in the F-35 Joint Strike Fighter air crafts garnered wide public attention. The plane engineers identified a software bug that causes the planes, when flying in formation, to incorrectly detect targets. As each of the planes within the formation detect a target from varying angles, the software is reportedly unable to decipher whether there is just one or multiple targets. As one news agency put it, the F-35's are "seeing double".

Software bug assists in bank heist

This story comes in two parts: one software bug related, one not. The first part to hit the news in mid-March detailed how a group of hacker-thieves hijacked the Bangladesh Bank system to steal funds. The group successfully transferred \$81 million in four transactions, before making a spelling error that tipped off the bank, causing another \$870 million in transfers to be canceled.

The software bug comes in with the \$81 million the thieves did successfully steal. According to Bangladesh Bank authorities, a printer is set up to automatically print read-outs of transactions made. The glitch in the system (whether coincidental or created by the thieves), interrupted the automatic printing process, so that it was only several days later that the transfer receipts were even discovered – giving the thieves plenty of time to cover their tracks.

Software glitch causes SolarCity Corp to be undervalued by \$400 million in acquisition

SolarCity Corp retained an investment bank to assist in the sale of the company to Tesla Motors Inc. After the \$2.6 billion dollar agreement had been signed however, the investment bank, Lazard Ltd., discovered that they had under-valued SolarCity Corp by roughly \$400 million. Whoops. Unfortunately the error was discovered too late for SolarCity's shareholders, but Tesla did offer to make up some of the difference in stock.

Frenchman sues Uber over a software bug

It's not often you hear of a software bug resulting in divorce, but we are living in exceptional times. A common Uber app bug revealed a man's affair to his wife, leading to a divorce and a lawsuit landing in Uber's lap. The bug causes Uber notifications to be pushed to a device, even after logging out of your account on that device. In this case, the "cheating Frenchman", who had once called an Uber from his wife's phone, was exposed when she received notifications of using Uber to visit his mistress. The angry ex-husband is now suing Uber for up to \$45 million in damages.

为什么软件需求重要吗？

- 返工通常会消耗总开发成本的 30% 到 50%，而需求错误则可能占到返工成本的 70% 到 85%。
- 软件需求不佳的原因
 - 用户参与度不足（敏捷开发中也考虑到了这一点）
 - 计划不准确（<https://www.entrepreneur.com/article/329019>）
 - 用户需求悄然增长（悄然增长指的是在不知不觉中增加）。
 - 模糊的需求（例如，“用户友好”并非具体且可衡量的）
 - 镀金（指任何昂贵的非必需品、便利设施或功能）
 - 被忽视的利益相关者（指那些很少受到关注的人。在你的经验中，这个人会是谁呢？）

为什么软件需求重要吗？

高质量需求流程带来的益处

- 需求和交付产品中的缺陷更少。
- 减少开发返工。
- 更快的开发和交付速度。
- 减少不必要和未使用的功能。
- 降低升级成本。
- 减少沟通误解。
- 减少范围蔓延。
- 减少项目混乱。
- 更高的客户和团队成员满意度。
- 能够达到预期效果的产品。

高质量要求的四个原则 <https://seilevel.com/requirements/the-four-attributes-of-高质量要求>（需求博客）

1. 原子/具体的

原子是“保持元素性质的最小不可分割的物质单元”。

2. 明确的

当两个人声称他们理解同一个要求时，他们对所陈述的内容可能有两种不同的解释。

- 3 可测试的/可测量的

该要求必须提供一种方法来判断所构建的软件是否合格。必须有一个明确的标准来区分合格与不合格。

4. 必要

如果你问利益相关者，他们可能会说每一项要求都是“必要的”。

您的职业

- 通用人工智能（或称人工智能）的崛起
- 威胁还是机遇？
- 当前情况及迹象

从客户角度看软件需求

一个利益相关者是指积极参与项目、受项目过程或结果影响，或者能够影响项目过程或结果的个人、团体或组织。



木桩——打入地面用来标记边界的木杆。

利益相关者——指那些在桩子被打入地下时持有桩子的人，即那些因为自身利益而决定桩子打在哪里的人。

Outside the Developing Organization

Direct user	Business management	Consultant
Indirect user	Contracting officer	Compliance auditor
Acquirer	Government agency	Certifier
Procurement staff	Subject matter expert	Regulatory body
Legal staff	Program manager	Software supplier
Contractor	Beta tester	Materials supplier
Subcontractor	General public	Venture capitalist

Developing Organization

Development manager	Sales staff	Executive sponsor
Marketing	Installer	Project management office
Operational support staff	Maintainer	Manufacturing
Legal staff	Program manager	Training staff
Information architect	Usability expert	Portfolio architect
Company owner	Subject matter expert	Infrastructure support staff

Project Team

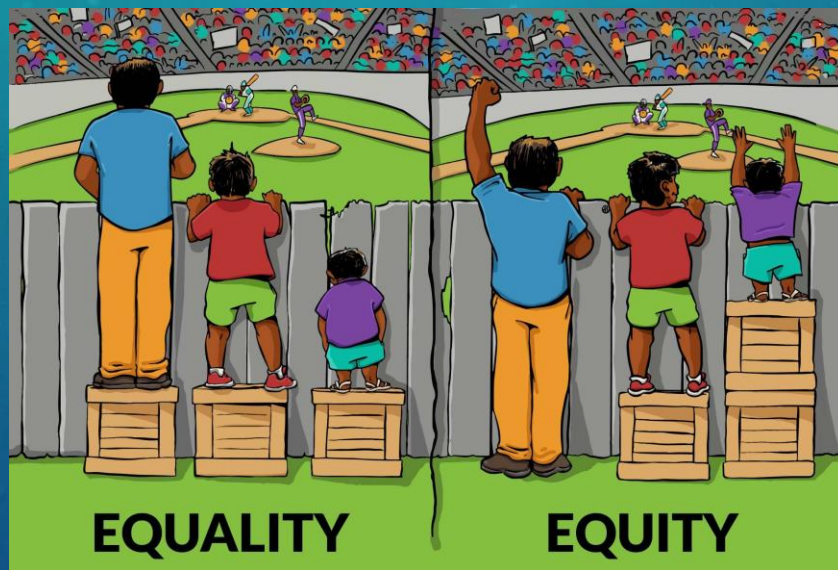
Project manager	Tester
Business analyst	Product manager
Application architect	Quality assurance staff
Designer	Documentation writer
Developer	Database administrator
Product owner	Hardware engineer
Data modeler	Infrastructure analyst
Process analyst	Business solutions architect

CW 任务 3

- 任务3:
 - 项目所有利益相关者名单
 - 对利益相关者角色的解释将发挥重要作用。

从客户角度看软件需求

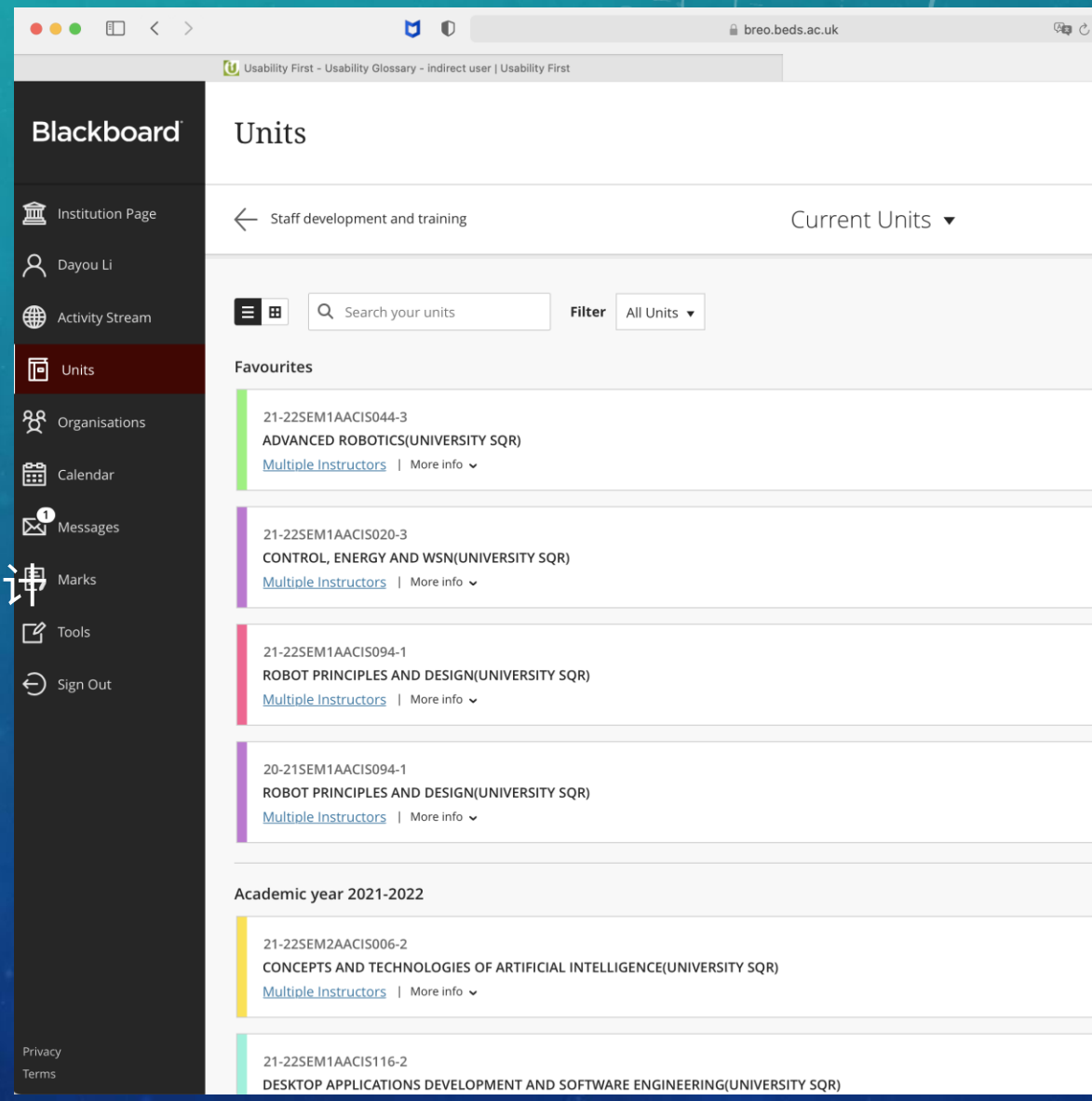
避免忽视一些重要的社区，但要重点关注……**核心套装**你需要了解哪些人的意见，才能确保你理解项目的所有要求和限制，以便你的团队能够交付正确的解决方案。



从客户角度看软件需求

顾客是利益相关者的一部分。

- 一个 **顾客**是指从产品中获得直接或间接利益的个人或组织。软件客户可以请求、付费、选择、指定、使用或接收软件产品生成的输出结果。
- 客户包括直接用户、间接用户、执行发起人、采购人员和收购方。
- 有些利益相关者并非客户，例如法律人员、合规审计人员、供应商、承包商和风险投资家。
- **最终用户**是客户的一个子集。
- 用户需求应该来自最终用户，也就是直接或间接使用产品的最终用户。



从客户角度看软件需求

优秀的需求源于开发人员和客户（特别是实际用户）之间的有效合作——一种伙伴关系。



讨论：
团队合作、协作和协同作用之
间有什么区别？

从客户角度看软件需求

(在美国) 法律原因

软件客户权利法案 你有**正确的到**

1. 希望业务分析师能说你的语言。
2. 希望业务分析师了解你的业务和目标。
3. 要求业务分析师以适当的形式记录需求。
4. 接收有关需求实践和交付物的解释。
5. 改变你的需求。
6. 期待一个相互尊重的环境。
7. 听取关于您的需求及其解决方案的想法和替代方案。
8. 描述使产品易于使用的特性。
9. 了解如何通过重来调整需求，从而加快开发速度。
10. 获得满足您的功能需求和质量期望的系统。



软件客户需求责任清单 你有**责任到**

1. 向业务分析师和开发人员普及你的业务知识。
2. 花时间提供和明确需求。
3. 在提供需求意见时，要具体、准确。
4. 当被问及时，对需求做出及时的决定。
5. 尊重开发商对需求成本和可行性的评估。
6. 与开发人员合作，设定合理的需求优先级。
7. 审查需求并评估原型。
8. 制定验收标准。
9. 及时沟通需求变更。
10. 尊重需求开发过程。

从客户角度看软件需求



与开发团队一起做决策

- 软件项目中可能需要做出数百个决策；这些决策往往是项目能否顺利推进的关键。您可能需要解决一些冲突、接受（或拒绝）一项变更提案，或者批准特定版本的需求。
- 开发与客户之间的协议
 - 就待开发产品或其特定部分的需求达成一致，是客户与开发商合作关系的核心。这项协议涉及多方：
 - 客户一致认为这些要求满足了他们的需求。
 - 开发人员一致认为他们理解这些要求，并且这些要求是可行的（可操作的、可实现的）。
 - 测试人员一致认为这些要求是可验证的。
 - 客户方管理层同意这些要求将实现其业务目标。

CW 任务 4

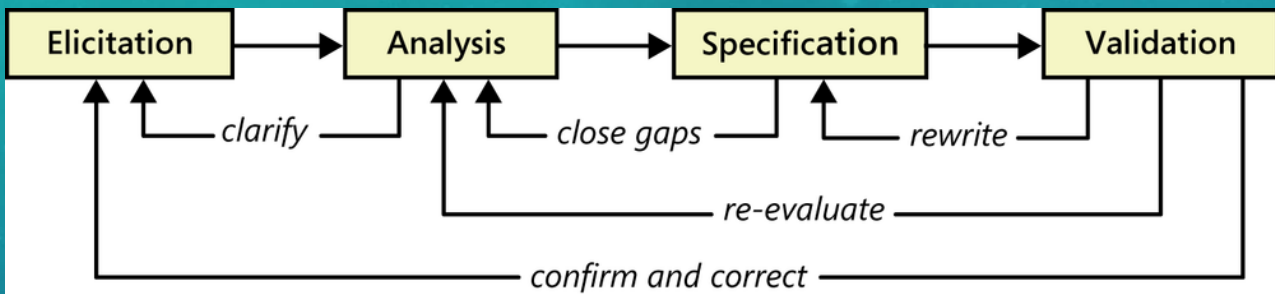
- 确定所有利益相关者对各种类型需求的认识。
- 对需求进行分类
- 解释要求和类型。

第三部分：如何（CH3）

如何制定软件需求？

软件需求开发过程

四个迭代、交织和渐进的阶段，或者，17 个步骤



1. Define business requirements
2. Identify user classes
3. Identify user representatives
4. Identify requirements decision makers
5. Plan elicitation
6. Identify user requirements
7. Prioritize user requirements

8. Flesh out user requirements
9. Derive functional requirements
10. Model the requirements
11. Specify nonfunctional requirements
12. Review requirements
13. Develop prototypes
14. Develop or evolve architecture
15. Allocate requirements to components
16. Develop tests from requirements
17. Validate user requirements, functional requirements, nonfunctional requirements, analysis models, and prototypes

Repeat for iteration 2

Repeat for iteration 3

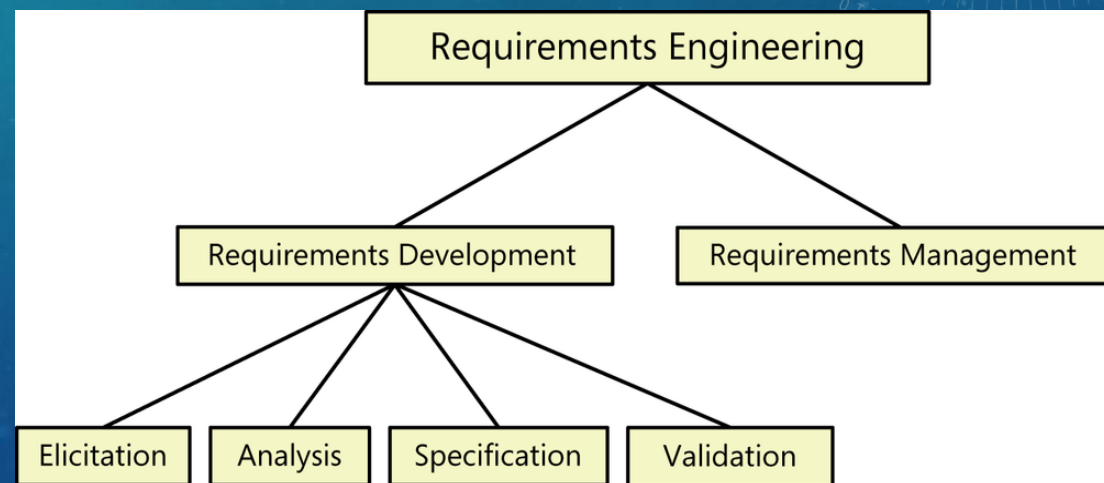
Repeat for iteration N

如何制定软件需求？

诱导涵盖从最高层开始的所有需求发现活动。这些活动包括访谈、研讨会、文档分析、原型设计等，旨在：

- 确定产品的**预期用户类别**以及其他利益相关者（需求应来自他们）

识别“其他利益相关者”的示例
思考一下：财务部门的员工是否会使用SITS？
他们会根据需要记录学生的缴费情况。



（这张图不太好。需求工程也包括需求实现。）

如何制定软件需求？

- 了解用户任务和目标，以及这些任务与业务目标相一致。并非所有任务用户提出的需求与系统需求相关，而有些需求则可能无关。（这些需求是什么以及为什么）

“无关任务”示例

思考一下：记录外部考官的意见是否与使用 SITS 有关？

不，因为这些评论与学生无关。

- 了解环境新产品将在何种情况下使用（条件）

“环境”示例（不一定是物理环境）思考：管理员在 SITS 开发方面的 IT 技能是什么？

- 理解功能需求以及他们的质量预期各类用户的需求（功能性需求和非功能性需求）。

如何制定软件需求？

分析需求引导需求从一个层级发展到另一个层级

- 这是一个以需求为原材料投入，以对每个需求更丰富、更精确的理解为输出的过程。
- 分析活动：
 - 分析从用户处收到的信息**区分**他们的任务目标来源于功能需求、质量预期、业务规则、建议的解决方案和其他信息。
 - 将高层需求分解为适当层级的需求**细节**
 - 从其他需求信息中导出功能需求
 - 了解相对重要性**质量**属性
 - 将需求分配给系统架构中定义的软件组件（**这种情况可能发生在第二种情形下，即产品已经开发完成**）
 - 协商实施**优先事项**（你知道这在敏捷开发中很重要）
 - 识别需求中的差距或**不必要的要求**它们与既定范围相关

如何制定软件需求？

要求**规格**包括以持久且组织良好的方式表示和存储收集到的需求知识。主要活动是：

“将收集到的用户需求转化为适合目标受众理解、审阅和使用的书面需求和图表。”

- 这是一项“整理工作”，也是一个好习惯。它有助于我们在软件开发的后期阶段快速找到相关文件。
- 这相当于机器学习中的数据预处理。
- 排序非常重要（分类+按重要性排序）。

5	1	1	1	1
2	2	2	2	2
3	3	3	3	3
8	8	8	8	5
1	5	5	5	8

使用的是什么算法？你能提供一下吗？
实施？



如何制定软件需求？

要求**验证**确认您拥有正确的需求信息，以便开发人员能够构建满足业务目标的解决方案。核心活动包括：

- 在开发团队接受需求之前，请审查已记录的需求，以纠正任何错误/疏漏或您不清楚的任何内容。
- 制定验收测试和标准，以确认基于需求的产品能够满足客户需求并实现业务目标。（您必须了解敏捷开发中使用的验收测试。验收测试是一种“黑盒”测试，即输入数据并检查输出是否符合预期，这被称为验证。）

良好做法

在四个阶段中

• 诱导

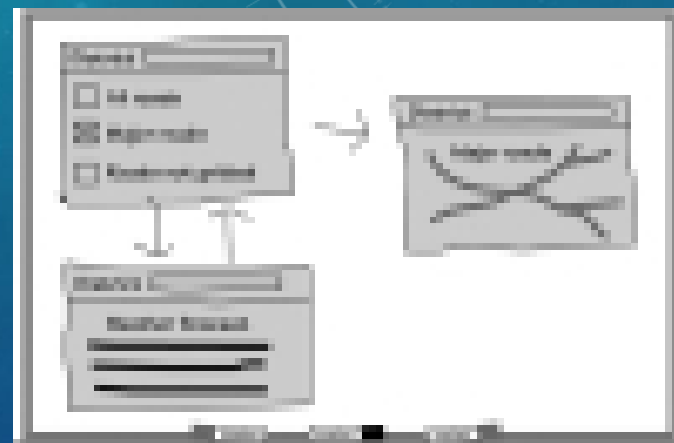
- 明确愿景和范围
- 根据用户使用软件的方式，识别用户类别
- 选择产品负责人（每类产品的关键联系人）
- 开展焦点小组访谈
- 确定用户需求
- 识别系统事件和响应（以了解各项需求之间的关系）
- 进行信息收集访谈（必须做好充分准备）
- 举办引导式需求获取研讨会（就像我为DPS做的那样）
(自然语言可能存在缺陷和不清晰之处) 至少有 30% 到 40% 的损失会返还给您)
- 尽可能重用现有需求。



良好做法

• 分析

- 对应用程序环境进行建模
- 创造原型
- 分析可行性
- 优先考虑的需求
- 创建数据字典
- 对需求进行建模（基于场景的建模、数据建模、面向流程的建模、基于类的建模）以及行为建模）
- 分析接口（不同需求集之间的接口）
- 将需求分配到子系统（现在可能为时尚早，但您可以将需求分类到不同的集合中。）



良好做法

• 规格

- 采用需求文档模板（这样可以使需求统一化、更加透明）。
- 确定需求来源（无论是否来自最终用户）
- 为每个要求添加唯一标签
- 记录业务规则
- 明确非功能性需求

• 验证

- 审查要求
- 测试要求
- 定义验收标准
- 利用基于场景的建模、数据建模、面向流程的建模、基于类的建模和行为建模来模拟需求。



良好做法

从其他角度看好的实践

需求管理（（处理变更）

- 建立变更控制流程
- 进行变更影响分析
- 建立基线并控制需求集的版本
- 维护变更历史记录
- 跟踪需求状态
- 跟踪需求问题
- 维护需求可追溯性矩阵
- 使用需求管理工具

知识

- 培训业务分析师
- 向利益相关者讲解需求
- 对开发人员进行应用领域方面的教育
- 定义需求工程流程
- 创建词汇表



"The client kept changing the requirements on a daily basis, so we decided to freeze them until the next release."

良好做法

从其他角度来看（续） 项目管理

- 选择合适的生命周期——哪种生命周期适合整个项目？
- 需求规划方法——如何以及何时开始和继续需求流程？
- 估算所需工作量——人力资源部门需要哪些资源？
- 根据需求更新项目基础计划——根据需求更新项目基础计划
- 确定需求决策者——由谁来决定？
- 当需求发生变化时，需要与客户重新协商承诺。
- 管理需求风险——针对不断变化的情况/环境制定应急计划
- 追踪需求分析工作量——不能无休止地投入精力在需求分析上
- 回顾过去吸取的教训——为了避免犯同样的错误

良好做法

价值与难度之间的权衡

- 高价值（良好实践的价值在于它们能为需求工程过程带来的益处）

价值	困难		
	高的	中等的	低的
高的	<ul style="list-style-type: none">• 定义需求工程过程• 根据需求制定基本计划• 重新协商承诺	<ul style="list-style-type: none">• 培训业务分析师• 计划要求方法• 精选产品推广大使• 确定用户需求• 进行信息收集访谈• 明确非功能性需求• 确定需求优先级• 明确愿景和范围• 建立变更控制流程• 审查要求• 将需求分配给各个子系统• 使用需求管理工具• 记录业务规则	<ul style="list-style-type: none">• 对开发人员进行应用领域方面的教育• 采用需求文档模板• 识别用户类别• 对应用程序环境进行建模• 确定需求来源• 建立基线并控制需求集的版本• 确定需求决策者

良好做法

• 中等价值

价值	困难		
	高的	中等的	低的
中等的	<div><div>• 维护需求可追溯性矩阵</div><div><div>• 举办引导式信息征询研讨会</div><div>• 估算需求工作量</div><div>• 重用现有需求</div></div></div>	<div><div>• 向利益相关者讲解需求</div><div><div>• 开展焦点小组访谈</div><div>• 创建原型</div><div>• 分析可行性</div><div>• 定义验收标准</div><div>• 对需求进行建模</div><div>• 分析接口</div><div>• 进行变更影响分析</div><div>• 选择合适的生命周期</div><div>• 识别系统事件和响应</div><div>• 管理需求风险</div><div>• 回顾以往经验教训</div><div>• 跟踪需求努力</div></div></div>	<div><div>• 创建数据字典</div><div>• 观察用户执行其工作</div><div>• 测试要求</div><div>• 跟踪需求状态</div><div>• 进行文档分析</div><div>• 跟踪需求问题</div><div>• 为每个要求添加唯一标签</div><div>• 创建词汇表</div></div>

良好做法

- 低价值

价值	困难		
	高的	中等的	低的
低的		<ul style="list-style-type: none">• 分发问卷• 维护变更历史记录• 模拟需求	<ul style="list-style-type: none">• 查看问题报告

- 总之，在整个项目周期内，需要在价值和难度之间取得平衡。
- 但在某些阶段
焦点可以放在任何一方。



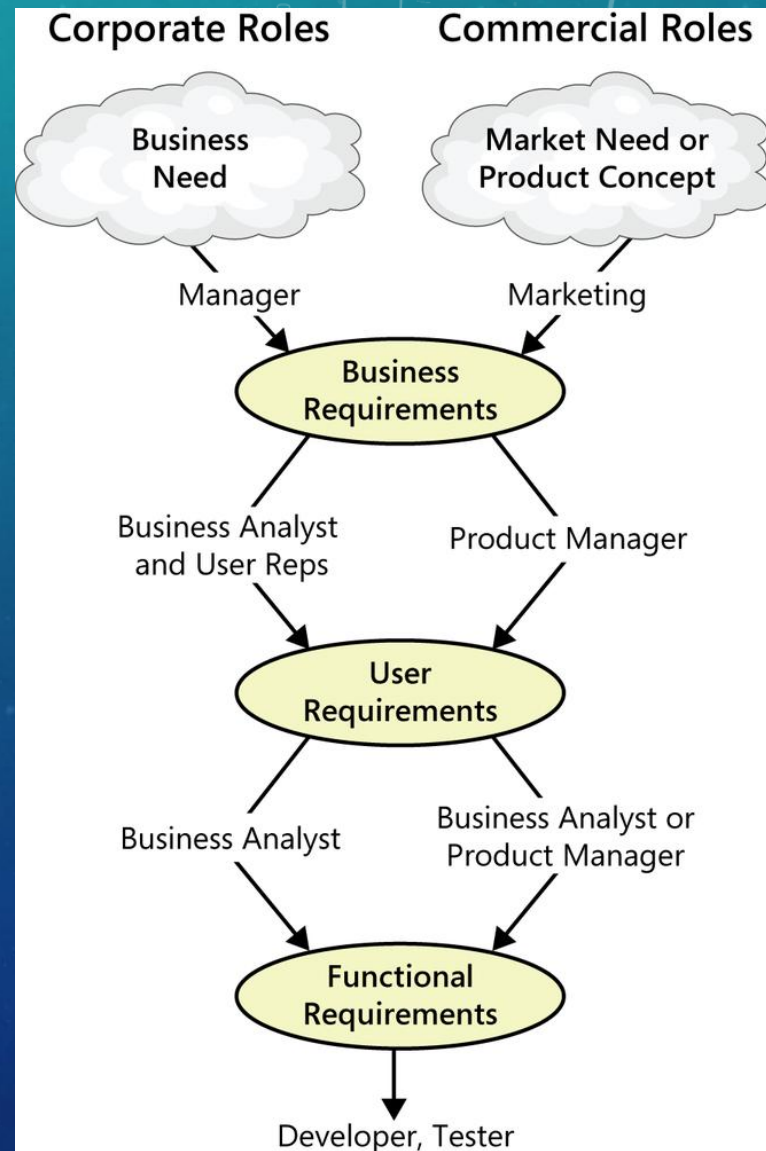
第四部分：世界卫生组织（CH4）

WHOSE 工作?

参与方式

各个利益相关者如何参与收集这三个层次的需求?

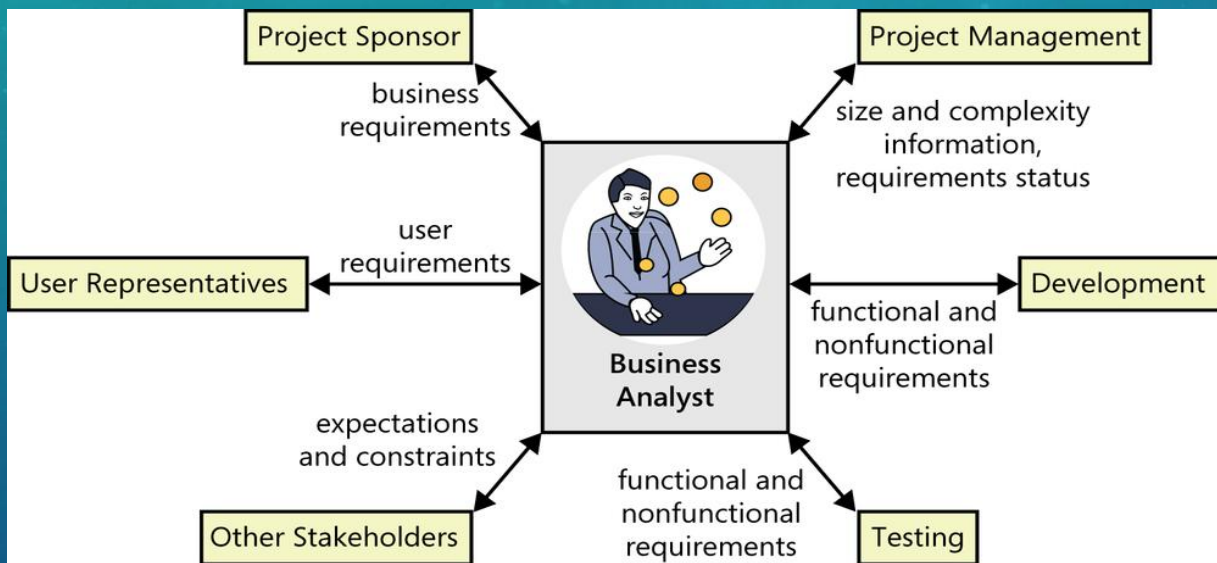
- 经理（企业职位）负责制定业务需求
- 市场营销（商业角色）旨在识别市场机会
- 商业分析师（BA）用户代表（Rep）是将业务需求转化为用户需求的关键人员。
- 项目经理应该
 - 为了满足用户需求，并且
 - 制定功能需求（业务分析师和项目经理）
- 开发人员和测试人员将继续按部就班地制定系统规范。



WHOSE 工作?

商业分析师 (BA) 角色

- 负责收集、分析、记录和验证项目干系人的需求。
- 客户群体与软件开发团队之间的主要沟通桥梁



WHOSE 工作?

计划和进度安排的区别：计划是制定由一系列行动组成的计划；进度安排是制定由一系列行动组成并与时间相关的时间表。

业务分析师的任务

全面了解应用程序并将这种理解传递给开发人员

- 明确业务需求——业务分析师可以自行定义，也可以通过业务分析师寻找其他相关人员，例如发起人、产品经理或市场经理。业务分析师可能会建议使用愿景和范围文档的模板。
- 制定需求方法计划——计划如何在整个项目过程中收集、分析、记录、验证和管理需求。业务分析师与项目经理合作，确保该计划与整体项目计划保持一致。（这应该归类为日程安排还是计划安排？）
- 确定项目干系人和用户类别——与业务发起人合作，为每个用户类别选择合适的代表，解释业务活动希望从他们那里获得什么，并就每个类别的适当参与程度达成一致。
- 收集需求——通过运用多种信息收集技巧
- 分析需求——通过使用需求模型来识别模式、发现需求中的差距、揭示冲突的需求，并确认所有需求的范围，从而得出客户请求的逻辑结果——衍生需求。

WHOSE 工作?

BA的任务（续）

- 文件要求——将要求写下来，“以清晰地描述” 解决方案这将解决客户的问题”（真的吗？）
通过使用标准模板
- 沟通需求——使用文本以外的方式表达需求，以确保所有参与方都能理解您所传达的信息。
（沟通的关键在于让别人接受你的想法，而不仅仅是理解你。）
- 主导需求验证工作——确保所有需求都可行、符合范围、合法等，并制定验收标准以测试可能的解决方案。
- 促进需求优先级排序——经纪人各利益相关方和开发商之间需要进行协作和协商，以确保他们做出合理的优先决策。
- 管理需求——跟踪这些需求的状态，验证它们在产品中的满足情况，以及管理变更达到需求基线（即所有相关方都认可的基本需求）

WHOSE 工作?

临场应变能力——你一旦有了想法，就能迅速根据这个想法采取行动。在这里，这意味着你能快速发现不确定因素，并立即采取行动加以澄清。

学士学位需要具备的基本技能

- 倾听——积极倾听包括排除干扰、保持专注的姿势和眼神交流，以及重述要点以确认理解。
- 访谈和提问——与不同背景的个人和群体互动，并提出正确的问题
- 临场应变——发现矛盾、不确定性、模糊之处和假设，以便在发现问题时立即进行讨论。
(不确定性 = 模糊性、歧义性、不准确性、未明确说明等)
- 分析型思维——既能进行高层次的抽象思考，也能进行低层次的抽象思考，并且知道何时从一种抽象层次过渡到另一种抽象层次，即具体化和概括化。
- 系统思维——着眼全局
- 学习——终身学习者，能够快速掌握新知识，建立背景知识，以开展需求工程工作。(背景？真的是背景吗？要说是顾客的背景吗？)
- 引导——发起并领导讨论

WHOSE 工作?

学士学位持有者需要掌握的基本技能（续）

- 领导力——运用各种技巧，影响利益相关者群体朝着特定方向前进，以实现共同目标。
- 观察法——观察用户执行工作或使用当前应用程序的过程，有助于发现用户可能不会主动提及的细微之处。
- 沟通——能够清晰地表达复杂的想法，无论是书面还是口头形式，都能面向多个受众，包括需要验证需求的客户和需要清晰、精确的需求进行实施的开发人员。
- 组织层面——建立信息架构（类、类别等），以支持项目信息在整个项目过程中不断增长。
- 建模——结构化分析模型（数据流图、实体关系图及类似图表、统一建模语言（UML）表示法等）
- 人际关系——让利益冲突的人们像一个团队一样合作，并用目标受众的语言与他们沟通，而不是让技术团队与业务利益相关者合作。
- 创造力——可能会带来新想法（并非解决方案）

WHOSE 工作?

必要的分析师知识

商业分析师需要具备广泛的知识，其中大部分是：**通过经验获得。**

- 了解当代需求工程实践及其在各种软件开发生命周期中的应用。
- 对项目管理、开发生命周期、风险管理和质量工程有深入的理解
- 产品管理概念包括新产品开发、商业论证、规划、验证、预测、定价、产品发布和产品营销等。
- 具备架构和运行环境方面的基本知识，以便能够参与有关优先级和非功能性需求的技术讨论。
- 企业、行业和组织都是强大的资产——**应用程序背景**

谁的工作?

哪些人适合担任业务分析师?

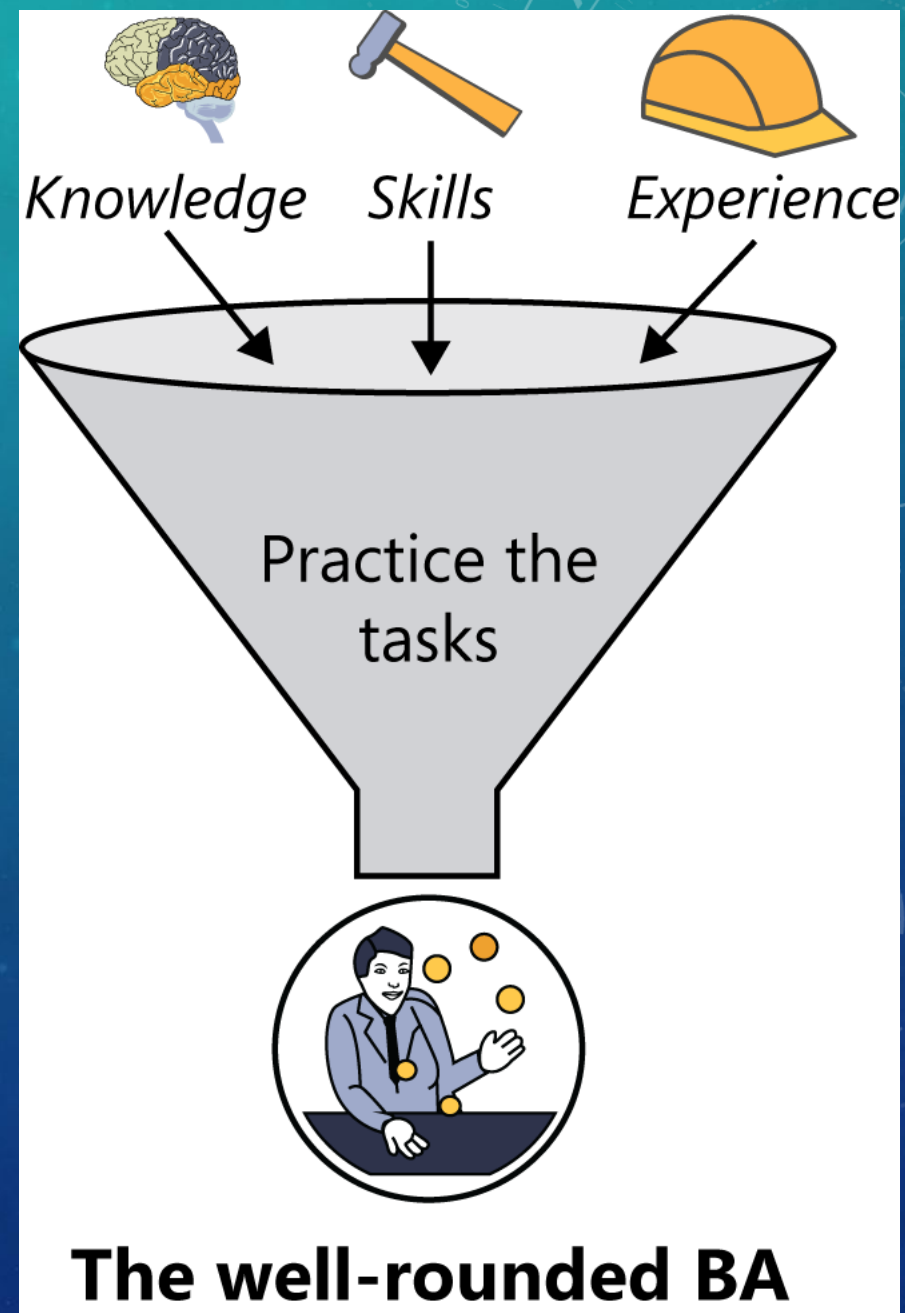
- 前用户
- 前开发人员或测试人员
- 前任（或现任）项目经理
- 该领域的专家
- 新兵（新招募人员）

这是一份地位较高且回报丰厚的工作。

How much do business analysts earn in London?

IT Business Analyst in London Area Salaries

Job Title	Location	Salary
UBS IT Business Analyst salaries - 16 salaries reported	London Area	£82,640/yr



第五部分：案例研究

1. 软件需求的必要性

“你好，菲尔？我是人力资源部的玛丽亚。我们公司的人事系统出了点问题，是你帮我们编程的。一位员工刚刚把名字改成了闪亮星光（Sparkle Starlight），但是系统无法识别这个更改。你能帮帮我们吗？”

“她嫁给了一个叫星光的人？”

“不，她没结婚，只是改了名字，”玛丽亚回答说。“问题就在这儿。看起来我们可以改名字。”只有当某人的婚姻状况发生变化时才会如此。

“嗯，是啊，我从没想过有人会改名字。”我不记得你跟我说过这种可能性。“当我们谈到这个系统时，”菲尔说。

“我以为你知道人们可以随时合法地更改姓名。”玛丽亚回答说，我们必须在周五之前解决这个问题，否则斯帕克尔就拿不到工资了。你能在那之前修复这个漏洞吗？”

“这不是漏洞！”菲尔反驳道。“我之前并不知道你需要这个功能。我正在忙着开发新的绩效评估系统。我大概能在月底前修复，但周五之前肯定不行。抱歉。下次请早点告诉我这些事情，并且最好写下来。”

“我该怎么跟斯帕克尔说呢？”玛丽亚问道，“如果她不能兑现支票，她会很失望的。”

“嘿，玛丽亚这不是我的错，”菲尔抗议道。“如果你一开始就告诉我，你可以随时更改别人的名字，就不会发生这种事了。”“你不能怪我没读懂你的心思。”玛丽亚又生气又无奈地吼道：“是啊，就是这种事让我讨厌电脑。修好之后赶紧给我打电话，好吗？”

2 客户对软件需求的看法

康托索制药公司的高级经理格哈德正在与康托索IT部门经理辛西娅会面。“我们需要建立一个化学品追踪信息系统，”格哈德开口说道，“这个系统应该能够追踪我们仓库和实验室里所有现有的化学品容器。这样一来，化学家们就可以直接从隔壁的同事那里借用一些化学品，而不用每次都购买新的容器。这应该能为我们节省一大笔钱。此外，健康与安全部门也需要更高效地生成关于化学品使用和处置的政府报告。你们能在五个月后合规性审计之前完成这个系统的搭建吗？”

“我明白这个项目的重要性，格哈德，”辛西娅说。“但在我能确定时间表之前，**我们需要了解化学品追踪系统的要求。**”

格哈德很困惑。“你什么意思？我刚才不是已经告诉你我的要求了吗？”

“实际上，你描述了一些项目的总体业务目标，”辛西娅解释说。“但这并没有给我足够的信息来知道要开发什么软件，或者可能需要多长时间。”**我想安排我们的一位业务分析师与一些用户合作，了解他们对系统的需求。**

“化学家们都很忙，”格哈德抗议道，“他们没时间在你们开始编程之前把每个细节都敲定。你们的人难道就不能弄清楚要开发什么吗？”

辛西娅回答说：“如果我们只是凭感觉猜测用户需要用系统做什么，那就做不好。我们是软件开发人员，不是化学家。我明白，如果我们不花时间去理解问题，没有人会对结果满意。”

“**“我们没时间做这些，”格哈德坚持道。“我已经把需求告诉你们了。现在请你们赶紧把系统建好。随时向我汇报进度。”**

3. 软件需求流程

“欢迎加入团队，莎拉，”项目经理克里斯汀说道。“我们期待你能帮助我们完成这个项目的需求分析。我知道你之前是做业务分析师的。你觉得我们应该如何着手呢？”

“嗯，”莎拉回答说，“我在想……” **我应该采访一些用户，看看他们想要什么。**然后我会把他们告诉我的内容写下来。这应该能给开发人员提供一个良好的切入点。我们以前也主要这么做。你认识一些我可以联系的用户吗？

“嗯……你觉得这对于这类项目来说足够好吗？”克里斯汀问道。**我们之前试过这种方法，但效果不太好。**我希望您能根据您以往的业务分析经验，分享一些比仅仅采访几个用户更好的最佳实践。您有没有发现哪些特别有效的技巧？

莎拉一时不知所措。**除了与用户沟通之外，我不太清楚还有哪些具体的方法可以满足用户需求。**我正努力根据他们所说的内容编写清晰的规范。在上一份工作中，我只是尽我所能，凭借我的商业经验做到最好。让我看看我能了解到些什么。

第一讲结束

