



敏捷开发中的需求活动

徐思涵
南开大学

Slides adapted from materials by Prof. Xiaohong Chen (East China Normal University)



写用户故事



写用户故事

敏捷需求开发方法

- 敏捷需求开发用简洁自然的用户故事描述软件系统的需求
- 用户故事在不断迭代的过程中得到修正和细化
- 开发小组围绕用户故事进行需求磋商、需求规划和估算，同时制订产品开发和测试规划，以及产品迭代开发和发布规划，形成以产品需求到产品发布的完整闭环。

写用户故事

用户故事

敏捷开发中，用户故事是干系人表达需求的主要方式。他们从使用者的角度表达对系统功能的要求，并用简短的自然语言将用户故事写在故事卡片上。

- 故事描述：故事描述有三个要素，分别表达谁(用户角色)需要这个功能，需要什么(功能)，以及为什么(目标/价值)需要提供这个功能。
- 故事测试：干系人围绕用户故事进行讨论，确定故事测试的细节，并用简短自然语言说明，以表达如何判定未来的实现是否符合期望。

常常需要反复迭代才能获得好的用户故事，从初始故事描述开始，反复讨论并不断补充具体内容，如测试用例和应用场景等，直到得到大家认可的用户故事。

写用户故事

如何写用户故事

作为 <角色> 我想要 <功能/意图> 以便于 <目的/实现价值>

↓ ↓ ↓

说明谁需要 说明需要什么 回答为什么需要

举例：

作为 患者家属 我想要 通过互联网挂号 以便于 节省去医院和排队的时间

作为 初次挂号者 我想要 创建一个账号 以便于 使用挂号系统提供的挂号服务

写用户故事

如何写用户故事

作为 初次挂号者 **我想要** 创建一个账号 **以便于** 使用挂号系统提供的挂号服务

围绕这个用户故事，干系人进行讨论并获得如下要求：

- 创建一个新账号，然后重新登录系统，并查看信息是否已经保存；
- 如果用户名已经存在，则不能成功创建账号。

- 这是“创建账号”需求的两个**实例化场景**，涉及用户故事的**测试要求**，告诉开发人员**如何判定一个用户故事是否被完整实现**。
- 从中可以获得“创建账号”这个用户故事的**验收标准**，包括：如果能够登录系统，则账号创建成功；如果用户名已存在，则创建账号不成功。
- 开发人员需要了解用户故事的不同场景，**常常在写用户故事的时候同时给出故事测试**，例如，当干系人讨论用户故事并希望明确其细节时，在编程实现用户故事前，或编程实现后进行故事测试时等环节，让写用户故事的人提供这些故事测试的内容。

写用户故事

如何写用户故事

在阅读用户故事的时候，同时思考类似以下问题，是编写故事测试的有效方式：

- 关于这个用户故事，开发人员还需要知道什么？
- 我对这个用户故事的实现有什么特殊想法吗？
- 是否存在一些特殊情况让这个故事可能包括不同行为？
- 用户故事在什么情况下可能会出错？

写用户故事

写用户故事场景

在获得用户故事描述及其附加条件之后，可以**构造用户故事场景**，它们就像脚本一样，包含了用户故事所包含的特定行为。一些敏捷开发框架，如极限编程（XP）、行为驱动开发(BDD)和测试驱动开发(TDD)等，都给出了**场景描述脚本语言**，使用户故事场景的编写能够更加规范，**支持故事测试的自动化**。**场景描述脚本语言和故事描述结构一起，成为完整的用户故事描述语言。**

Gherkin语言是一种支持场景描述的用户故事描述语言。它的中文版用户故事以功能特性(Feature)关键词开头，后面接着能表达该功能的名称，然后是功能描述，形式为：

作为<角色>我想要<功能/意图>以便于<目的/实现价值>

写用户故事

写用户故事场景

场景以关键词场景(scenario)开头，后面接着该场景的名字，用于概述场景测试的结果，然后是场景的描述。场景描述需要表达①先决条件； ②行动； 和③结果。结构化表示为

- ◆ 假如(Given) <前提条件1> 而且(and) ... 而且(and) <前提条件n>
- ◆ 当(when) <交互动作>
- ◆ 那么(Then) <后置条件1> 而且(and) ... 而且(and) <后置条件m>

其中先决条件相当于该场景的触发条件，交互动作指进入该场景后执行的行动，结果是指行动产生的效果。其场景刻画的模式为： ①系统处于特定状态； ②系统交互动作； ③系统所处的新状态。

写用户故事

带场景的用户故事举例

功能 创建合法账号

作为 系统初次使用者 **我想要** 创建一个账号 **以便于** 使用挂号系统提供的挂号服务

场景 成功创建账号

假如 输入了用户名

而且 输入了密码

当 用户单击确认创建

那么 显示“成功创建账号”

场景 用户名已存在，不能创建账号

假如 输入了用户名

当 系统发现用户名已存在

那么 提示“用户名重复”

场景: 成功创建账号

假如 输入了<用户名>

而且 输入了<密码>

当 用户单击确认创建

那么 显示<创建结果>

例子:

<用户名>|<密码>|<创建结果>

138123456|123456|成功

158123456|654321|成功

12|12345678|不成功

138123456|234567|不成功

在带场景的用户故事中，每个功能特性至少包含一个场景，每个场景表示了某个特定情景下该功能特性的行为实例或执行路径，还可以表示功能特性的边界情况。

为了支持场景的自动验证，场景后面还可以附加场景测试用例，以关键词例子(Example)开头

文A

写用户故事

怎样写好故事

一般认为，好的用户故事应具有以下六个特性，简称INVEST原则。

- **独立性**(Independent): 每个用户故事可独立存在，避免相互依赖。**独立性保证用户故事间没有顺序约束，其更改也不会相互影响。**
- **可协商性**(Negotiable): 用户故事的内容是可以协商的，写故事不是签合同，故事卡内容作为功能描述，是所有干系人反复讨论产生的。
- **业务价值性**(Valuable): 用户故事主要用来表达**客户或业务价值的实现，不是表达要采用的技术手段**。像“作为数据库访问者我想要所有的数据库连接都要通过某个连接池”这样的描述是技术手段，不代表业务价值，不是一个好的用户故事。
- **可估算性**(Estimable): 开发团队能够根据用户故事大致估算出工作量。**如果开发人员觉得很难估计，说明这个用户故事缺少必要细节或者粒度太粗需要拆分**。例如，用户故事“作为病人我想要使用手机挂号以便于不用排队”，可以分解成“病人可以创建账号、填写挂号信息、编辑挂号信息”这三个更具体的用户故事。
- **粒度适中性**(Small): 虽然一个用户故事的粒度不能太粗(见可估算性)，但也不能太细节，例如，用户故事“作为病人我想要输入看病的日期”就太过细节了。粒度细的用户故事需要合并。
- **可测试性**(Testable): **用户故事需要是可测试的**，特别是表达功能需求的用户故事。有些关于质量需求的用户故事可能难以测试，例如“用户不需要花很长时间等待窗口出现”，这里的“很长时间”不具有可测试性。

写用户故事

怎样写好故事

表 11.1 用户故事需求质量评估准则

质量评估准则	具 体 含 义	评 估 边 界	
完整性	关键字段完整	故事描述中角色、意图不缺失；场景中上下文、事件和结果不缺失	单个故事
	语言成分完整	故事和场景中每个字段的描述信息中关键语言成分不缺失	单个故事
	功能完整	所有故事尽量覆盖一个完整的系统功能，即没有步骤/功能上的缺失	故事集合
一致性	模板统一	采用统一的格式表达故事和场景	故事集合
	无重复	不存在两个相同的故事，也不存在两个相同的场景	故事集合
	无冲突	任意两个故事和任意场景不存在矛盾	故事集合
可测试性	原子性	一个字段描述一种情况，不可拆分	单个故事
	最小化	每个关键字段仅包含必要的信息	单个故事
	无二义性	不使用模糊词汇，表达明确的信息	单个故事
	可估算	故事不能粒度太粗，可用场景实例化	单个故事
	独立性	故事之间无相互依赖关系	故事集合

写用户故事

怎样写好故事

表 11.2 违反准则的用户故事案例

ID	故事描述	违反的准则
US1	我想要 当文件不能打开时,可以看到出错提醒	关键字段完整:角色缺失
US2	作为 手机用户 我想要 一致界面 以便于 不会产生混淆	语言成分完整:意图缺少动词
US3	作为 找工作者 我想要 增加个人信息	功能完整:应该还有“删除”和“修改”个人信息的故事
US4	作为 管理员, 当 一个用户已经注册,可以收到邮件	模板统一:没有使用团队认可的用户故事模板
US5_1	作为 新闻网站访问者 我想要 查看最新的新闻 以便于 知道当前最新的新闻	无重复:有两个干系人分别写出了相同的用户故事
US5_2	作为 新闻网站访问者 我想要 查看最新的新闻 以便于 知道当前最新的新闻	
US6_1	作为 购买者 我想要 搜索图书 以便于 找到我想买的书	无冲突:相同“意图”达到不同“目的”
US6_2	作为 购买者 我想要 搜索图书 以便于 查看是否存在该图书	
US7_1	作为 购买者 我想要 输入注册信息 以便于 成为合法用户	无冲突:不同“角色”相同“意图”和/或相同的“目的”;这两个故事可以考虑合并
US7_2	作为 快递员 我想要 输入注册信息 以便于 成为合法用户	
US8	作为 用户 我想要 单击地图上的一个位置,并执行与该位置关联地标的搜索 以便于 反馈到关联地标的路线	原子性:包含多个功能点
US9	作为 医生 我想要 查看本周的预约时间(分为手术和出诊)	最小化:包含多余信息
US10	作为 注册用户 我想要 查看医生详细信息 以便于 选择合适的医生	无二义性:使用了模糊词
US11	作为 求职者 我想要 发布自己的简历 以便于 招聘方可以看到	可估算:US11 故事粒度太粗分解成 US11_1、US11_2、US11_3 三个可评估的小故事
US11_1	作为 求职者 我想要 创建新简历	
US11_2	作为 求职者 我想要 修改简历	
US11_3	作为 求职者 我想要 删除简历	
US12_1	作为 采编网用户 我想要 编辑稿件	独立性:存在先后关系
US12_2	作为 管理员 我想要 增加一个用户 以便于 增加的用户可以使用采编系统	



敏捷开发中的需求 规划

文A

敏捷开发中的需求规划

团队的组织

参与敏捷需求活动的干系人包括三类：客户、开发人员和管理者。

- **客户：** 包括客户代表、销售人员、领域专家、市场营销人员、客户经理等。他们的任务是编写用户故事并向开发团队提供反馈，主要负责的工作是：编写用户故事、参与需求优先级排序、参与需求测试等。
- **开发团队：** 包括开发者和测试者。他们参与需求收集和磋商，并不断与客户交流互动，每次需求迭代时都向客户展示软件(原型)，在展示的过程中收集客户的反馈。
- **管理者：** 其职责是确定最佳敏捷团队，促进团队成员之间的协作。在小规模的敏捷团队(一般包含6~8名成员)中，管理者包括产品负责人、敏捷开发负责人等。产品负责人代表客户的利益，承担的任务包括：与开发团队沟通产品愿景、管理项目财务与投资回报、决定正式发布时间等。敏捷开发负责人则负责促进开发团队和产品负责人的沟通，全面掌握团队开发活动进展并把握节奏。大规模的敏捷项目团队可能由几个或十几个小规模的敏捷团队组成，每个团队通常有3~9人。超大型的项目团队可能由几十个甚至是上百个小团队组成。这些团队间一般通过会议方式定期进行沟通和协同。

敏捷开发中的需求规划

团队沟通的技巧

- **主动倾听：** 倾听是有效沟通的首要条件。听者要以发言人为焦点，鼓励发言人充分完整地表达想法。听者还要向发言人陈述自己对发言人描述内容的理解，以确认真正理解了发言人的想法，如果没有，则需要通过进一步的交流澄清理解。
- **交互反馈：** 成员之间需要相互反馈对于对方所表达意思的理解，形成团队的信息流。当信息和反馈在成员间自由流动时，团队成员才可以集中力量，实现共同目标。
- **群体决策：** 群体决策是指群体成员共同商议，取得对一个问题的确定性方案。当群体成员数量增多时，群体决策存在较大难度。此时，**管理者可以行使决策权，优先选择解决重要的、时间紧迫的事情。**

敏捷开发中的需求规划

敏捷会议

面对面会议是敏捷开发中团队成员交流和协作的主要途径。除了针对开发过程中的特定问题组织不定期的讨论会以外，敏捷团队还会定期组织不同主题的会议。

- **计划会议：** 团队成员讨论并商定在迭代周期内要完成的工作范围。具体包括：①选择可以在一个迭代周期内完成的任务；②明确迭代周期内完成的任务，即将选出的任务分解成工作，并估算出需要的时间。
- **每日站会：** 每天在同一时间和地点举行，时长一般为15分钟左右，用于团队同步每天的工作。一般就如下三个问题进行快速交流：上次会议之后你完成了什么？今天计划完成什么？是否遇到可能影响工作完成的阻碍？
- **审核会议：** 在迭代周期的最后一天进行。团队成员审查已完成的工作和未完成计划的工作；通过产品演示向干系人介绍已完成的工作；与干系人合作，共同研究下一步工作。
- **回顾会议：** 在审核会议后举行。团队趁此机会发现如何改善工作流程和敏捷框架的实现。一般围绕如下三个问题进行：在当前迭代期间哪些方面进展顺利？哪些方面不顺利？在下一个迭代周期中哪些方面需要改善和提高？

敏捷开发中的需求规划

为用户故事确定优先级

确定用户故事优先级，需要考虑以下因素：

- ①用户故事是否存在不能如期完成的风险(如设计全新算法的难度与风险)；
- ②推迟实现一个用户故事对其他用户故事是否有影响；
- ③用户故事对于用户或客户的重要性；
- ④用户故事与其他用户故事的内聚性，一个用户故事本身可能不重要，但它却是一个重要用户故事的补充，它应具有较高的优先级；
- ⑤商业价值、开发成本、学习代价等。

一般来说，开发人员有一个实现用户故事的排序，客户也有自己的排序。当双方对排序有不同意见时，往往听取客户的意见。

敏捷开发中的需求规划

为用户故事确定优先级

- 在客户和开发者对用户故事排列优先级时，**需要知道每个故事的实现要花费多长时间。故事点是一个代表用户故事实现所需时间的模糊单位，常定义为“理想工作日”**(没有任何中断的、可以全身心投入工作的一天)。
- 对于每个用户故事，估算其故事点是一个迭代的过程：
 - 首先，给定一组故事集，并召集所有参与估算的客户和开发人员，组织一个讨论会；
 - 客户从故事集中**随机选择一个故事**，读给开发人员听。开发人员根据需要尽可能多提问题，并由客户回答。如果客户不知道答案，团队可以推迟对这个故事的估算； 当这个故事没有更多的问题时，每个开发人员都会在卡片上写下一个故事点数的估算值；
 - 随后，会议负责人会公开所有人员的估算值，当估算值不相同时，估算值高的和估算值低的人要解释他们这样估算的原因；
 - 接下来，团队成员会展开讨论，如果对故事有疑问，客户会澄清疑问，并将澄清的信息填入故事片卡上； 团队讨论之后，开发人员再次将估算值写下来。
 - 当每个人都完成估算后，将估算值再次公开展示给所有人。
 - **在大多情况下，第二轮估算值会比较接近。**如果估算值还存在差异，应该重复让估值高和估值低的人解释他们的想法。估算的目标是得到一个估算值，估算过程很少超过三轮。

敏捷开发中的需求规划

规划的迭代与发布

敏捷软件项目一般以2~6个月为一个发布周期。需要根据团队进度和用户故事估算，根据发布周期，制订发布计划和迭代计划。具体需要进行以下工作：

- ①根据项目一次迭代周期时长，确定每次发布的迭代次数；
- ②估算开发团队的开发速度，即每次迭代可以完成的故事点数量；
- ③对故事进行优先级排序，确定故事的开发次序；
- ④把故事分配到具体的迭代中，使每次迭代的故事点数量不超过开发团队的开发速度。

文A

敏捷开发中的需求规划

规划的迭代与发布

- 项目干系人团队一般以会议的形式共同讨论并选择合适的迭代周期时长。例如，在Scrum中，迭代时长通常为1~4周。
- 迭代时长越短，则项目可以越频繁地调整，项目进度也会更加透明。
- 当不确定迭代的长度时，通常选择短迭代而不是长迭代，因为使用长迭代更容易犯错。
- 此外，还应该尽可能地坚持固定的迭代长度。因为有了一致的迭代长度，项目会有固定的节奏，有利于团队开发速度。
- 通常情况下，开发人员和客户会协商一个日期范围来发布软件产品，而不是指定一个固定的日期，以降低风险。

敏捷开发中的需求规划

规划的迭代与发布

项目干系人团队会安排迭代计划会议来制订迭代计划，包括如下任务。

- **讨论故事：** 会议开始时，客户从最高优先级的故事开始，读给开发人员听。然后由开发人员提问，直到开发人员能充分理解故事。
- **分解任务：** 将故事表达的特性细化到任务，使开发团队能明确故事的完成步骤。例如故事“用户可以根据不同的字段搜索科室”，可以分解为三个具体任务：编写基本搜索界面、编写高级搜索界面和编写搜索结果界面。任务分解的技巧如下：①如果故事的某个任务很难估算，就把那个任务从故事中分离出来；②如果任务能分配给多名开发人员协作完成，就将任务分解；③如需要让客户了解故事的部分完成情况，可以把这部分单独提出来作为一个任务。
- **承担任务：** 故事的所有任务都确定之后，需要有开发人员执行每个任务。可以把任务写在白板上，开发人员在他们认领的任务旁写上自己的名字，并估算任务完成的时间。如表11.3所示是一个记录开发人员承担任务的估算表。

表 11.3 分配责任人的任务列表

任 务	责 任 人	估算时间(故事点,单位:个)
编写基本搜索界面	王建欣	1
编写高级搜索界面	李依然	2
编写搜索结果界面	王建欣	1
...



敏捷需求质量检查 和测试

文A

敏捷需求质量检查和测试

故事质量检查

带场景用户故事质量的概念模型，包括如下三组概念：

- **用户故事：** 用户故事包含故事描述、特征属性和状态变迁。其中，
 - 故事描述包括故事名、角色、意图和目的。
 - 特征属性是实体属性的集合，其中每个属性-值对<实体属性，取值>是一个原子状态，多个原子状态的合取构成组合状态，原子状态和组合状态统称为状态。
 - 状态变迁表示为三元组<先决条件，行动，结果>其中，“先决条件”和“结果”表示状态，“行动”表示事件。其含义是：当处于“先决条件”状态，如果发生事件“行动”，则进入“结果”状态。
- **关系：** 有合作关系、依赖关系或重复关系三种。存在合作关系的故事所描述的业务需求具有较高的相关度，它们或者就是用于合作完成一个规模较大的用户故事，因而会有相同的特征属性。依赖关系指用户故事之间存在先后顺序，即一个用户故事的完成依赖于另一个用户故事的实现。重复关系指两个故事具有相似的故事描述、特征属性及状态变迁。
- **问题：** 指故事中存在不满足质量评估准则的情况,包括问题类型、问题定位、修改推荐等。问题类型指违背了质量评估准则中的哪条准则。问题定位指将用户故事看成由“Gherkin关键词+句子”组成的文档，当发现对应行出现问题，则反馈该位置。修改推荐是针对该问题提出的修改建议。

敏捷需求质量检查和测试

故事质量检查

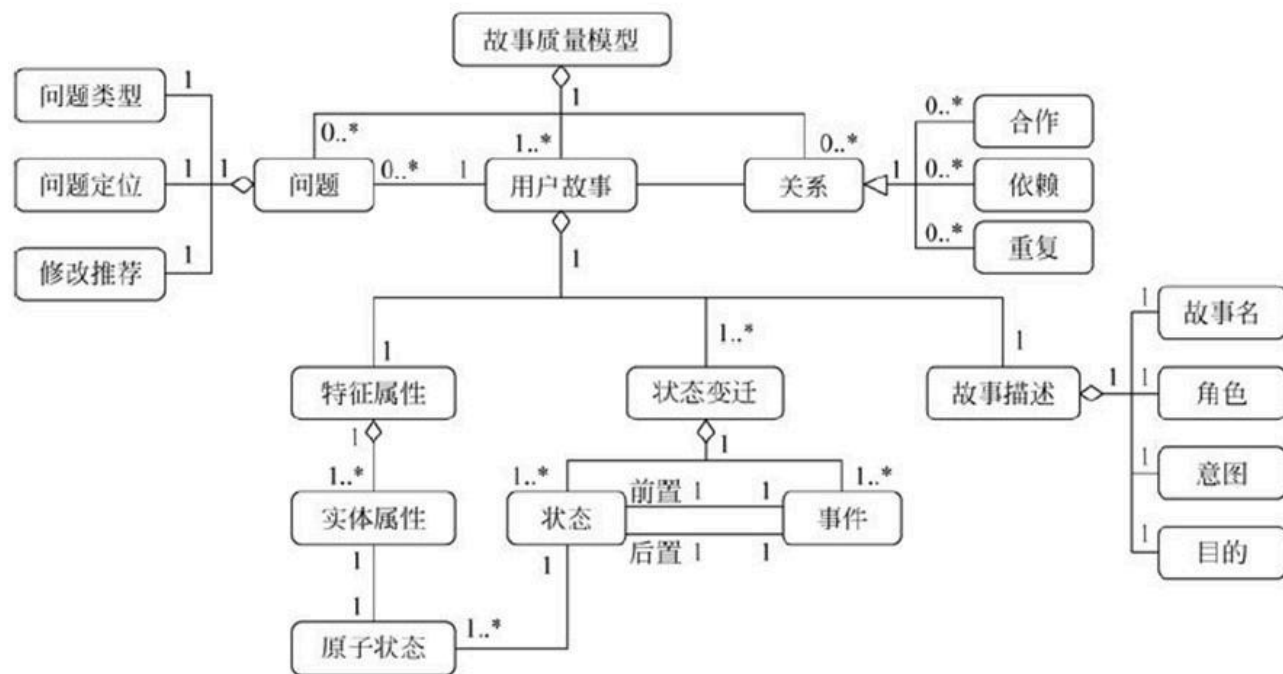


图 11.4 故事质量概念模型

敏捷需求质量检查和测试

基于故事质量模型的问题发现过程

- **故事结构分析：** 以带关键词的用户故事文档为输入，根据关键词的含义，识别用户故事中的概念，**判断是否违背了概念完整性和模板统一准则。**
- **故事句法模式分析：** 采用自然语言分析技术分析故事中每个概念的对应句型和语法成分，**识别出场景对应的状态变迁序列和特征属性信息，同时发现违背成分完整、原子性及最小化准则的故事。**
- **故事语义分析**包括以下三方面。
 - **相似性分析：** 先对输入的一组故事进行相似度计算，发现重复的故事和相似度高的故事。相似度值为1时，认为两个故事是重复的故事。
 - **场景冲突分析：** 对每个故事和每对相似度高的故事中的状态变迁序列进行检查，分析场景是否存在冲突。借助同义词库、模糊词库识别故事中是否存在模糊性和用词上的不一致。
 - **依赖和合作关系分析：** 根据概念之间的关联信息，量化用户故事存在合作关系和依赖关系的关联性。例如，一个故事的收益(以便于后面的字段)可能叙述了该故事与其他故事之间的依赖关系。如故事“作为持卡人我想要验证持卡人身份以便于可以办理取款业务”表达了“验证持卡人身份”这一故事与“取款”用户故事之间存在可能依赖关系。识别这些关系并反馈给敏捷团队，帮助判断是否违背了独立性以及完整性。

敏捷需求质量检查和测试

需求测试

- 需求测试主要判断故事是否符合客户团队的期望。当迭代开发过程开始，开发人员即对当前迭代周期的故事编写代码，客户团队则为每个故事定义相应的测试，并将测试内容写在故事卡片的背面，由测试人员输入自动化测试工具。
- **行为驱动开发**(Behavior-Driven Development,BDD)是一种自动化架构，它**将场景转换成验收测试**。比较常用的行为驱动开发架构有Cucumber、JBehave和SpecFlow等。
- 图展示了Cucumber自动化测试的过程，包括四个任务：首先使用Gherkin语言来描述场景；然后由程序员编写“步骤定义”，即针对场景需求编写测试的执行步骤；随后步骤定义被转换成可执行的代码(Cucumber支持Ruby、Java、.NET、JavaScript等多种语言)；最后执行这些代码，进行场景测试并反馈结果。

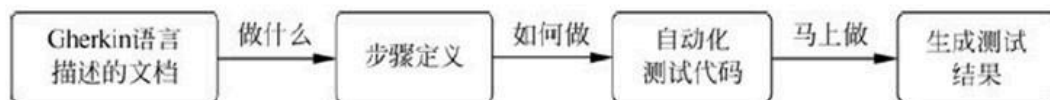


图 11.6 Cucumber 的需求测试过程

敏捷需求质量检查和测试

需求测试

- 步骤定义完成从业务领域到编程实现的转换，职责是将每个Gherkin场景中的简单描述翻译成可以执行的代码。例如针对“Given 账户余额是\$100”，其步骤定义需要做以下两件事情：为场景中的角色创建一个账户，将账户的余额设置为100美元，可表达为

```
Given / 账户余额是$100/ do
# 这里写将100美元存入用户账户的代码
end
```

- 将写好的步骤定义保存为一个可执行文件，包括一个用于匹配单个或多个步骤的正则表达式(两个/之间的部分)和一个在步骤匹配时需要运行的代码块(do .. end之间的部分)。Cucumber存储代码块，如果遇到匹配的步骤则启动执行，输出检查每个步骤的测试结果。例如，反馈这个场景是否测试成功(Passed或Failed)，如果测试失败，则反馈未通过测试的场景描述信息。

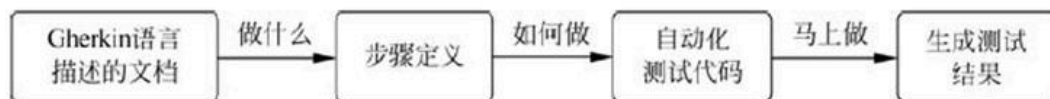


图 11.6 Cucumber 的需求测试过程

敏捷需求质量检查和测试

非功能属性测试

用户故事也可以表达非功能方面的需求。通常情况下，非功能需求和设计约束以约束的形式写在用户故事卡片上。例如，用户故事“作为消费者我想要得到任何限电计划的通知”有一项非功能需求：“所有公共事业单位的通知应该在1分钟内传达”。该非功能需求作为上述用户故事需求的约束，写在用户故事卡片上。

有时也可直接采用用户故事描述非功能需求，例如，将上述非功能需求表达成如下用户故事：“作为消费者我想要来自公共事业单位的通知在1分钟之内传达以便于我可以快速采取适当行动”。

表 11.4 敏捷非功能需求的测试方法

质量需求类别	系统品质测试
易用性	测试用户使用系统实现其目标的难易程度。通常由 3~5 人组成测试小组，以预设的方式执行产品系统。关注以下四方面：用户执行特定任务的时间；经历了多少错误或失误；使用系统的难易程度；使用系统的反馈（苦差事、可接受、很快乐等）
可靠性	根据可靠性测试用例，让系统在负载或可能出现的环境压力加剧的情况下，也能够正常运转
安全性	安全性是一种特殊类型的可靠性测试，可采用白盒测试和黑盒测试两种方式。白盒测试检查实际的代码，通过测试用例测试安全性。黑盒测试模拟现实黑客尝试破坏系统的方式，使用一些脚本和工具，向系统注入各种有缺陷的输入，以发现安全问题
性能	性能测试通常使用专门工具辅助完成。系统级性能测试可以通过用户或其他系统负载模拟器、度量和监视的工具来进行。这些模拟的负荷被施加到服务器、网络、系统组件和其他对象上，以测试系统在不同负载类型之下的还原性，建立总体性能指标

4

敏捷需求管理工具

文A

敏捷需求管理工具

工具概述

- 在敏捷需求开发过程中，规模较小的敏捷开发团队一般只使用纸笔和看板来管理用户故事，可以随手操作和随时调整。但是，用手工看板管理需求的方式不适用于大型异地研发团队的敏捷需求管理。
- 一些敏捷开发团队使用工作簿(spreadsheets)、数据库(databases)并基于维基(Wikis)等通用工具管理用户故事，但这些通用工具不专门针对敏捷项目管理，不适用于迭代式的开发过程。目前，一些敏捷开发团队选择使用专用的敏捷管理工具，实现敏捷开发过程管理。
- 敏捷需求管理工具的基本功能包括：支持不同利益相关者编写用户故事；与本地和异地团队交流；对用户故事需求进行估算和规划；对故事的开发进度报告和度量；等等。有的敏捷需求管理工具还支持需求验证和需求建模等。先介绍敏捷需求管理工具的基本功能，然后介绍其扩展功能，最后介绍一些常用的工具。

敏捷需求管理工具

用户故事编写

敏捷需求管理工具的常用功能包括编写用户故事、迭代式地管理用户故事及编写进度报告等。图展示了敏捷工具Rally的用户故事编辑界面，支持用户故事的写作、给用户故事附加更多细节(如存在的缺陷、用于测试的用例、用户故事分解形成的任务及必要的说明和注释等)，以及规划用户故事的版本、评估用户故事的点数等功能。

Edit User Story » US39: Test Story 2

General

ID: US39

Name: Test Story 2

Tags: Rally Review
Choose Tags

Description: Normal

Attachments: Browse...

Owner: No Entry

Hierarchy

Parent:

Schedule

Blocked: ☐

Release: Review Release Rally

Plan Est: 4.0 Points

To Do: 0.0 Hours

Iteration: Review Rally Iteration 1

Task Est: 0.0 Hours

Notes

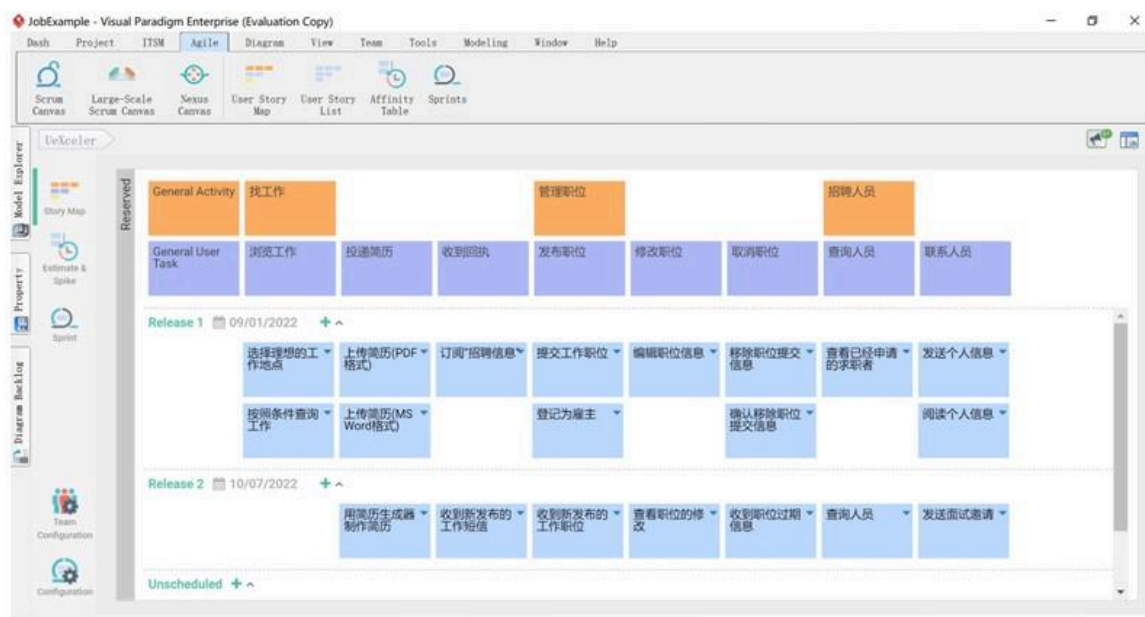
Notes:

Save & Close Save & New Save Cancel

敏捷需求管理工具

用户故事地图

一些敏捷实践采用用户故事地图(user story mapping)组织用户故事，表达发布计划、迭代计划及对故事进行验收测试等信息。



包括三个层次：活动(activity)、任务(task)、故事(user story)。其中，活动是相同角色的群体在一个时间段内完成任务的组合，目的是达成一个特定的目标。任务是实现软件功能的手段，完成活动所涉及的具体操作。用户故事表达实现任务的功能特性，与其所属的任务相对应，置于用户故事地图的第三层。

文

敏捷需求管理工具

用户故事追踪

为了追踪进度，一些工具将迭代期间的故事状态、缺陷和其他正在处理的任务可视化。如图所示是一个敏捷项目管理工具中的迭代状态。在这个界面，任何干系人都可以看到迭代中的用户故事所处的状态(已定义、进度中、已完成、已验收、受阻等)、剩余任务的工作量估计以及迭代的整体工作完成趋势。

ID	Name	Schedule State	Owner	Project
US98	Get a new Card	D	Dan	Re-Charge Team
US97	Shop for Items	A	Dudley	Re-Charge Team
US96	Save cart for future checkout	C	MattS	Re-Charge Team
US94	Payment - Gift Certificates	A	Dawn	Shot-in-the-Arm T...
US93	Ask for help	D	MattS	Re-Charge Team
US92	Daily deals	A	MattS	Re-Charge Team
US91	Customize product list	D		Shot-in-the-Arm T...
US90	Manage cards	D	MattS	Re-Charge Team
US89	Request higher limit	D	MattS	Re-Charge Team
US88	Add Single Item	A	Drew	Shot-in-the-Arm T...

敏捷需求管理工具

用户故事关系追踪

用户故事之间可能存在合作、重复、依赖等关系。识别用户故事之间的关系有助于发现用户故事的冲突和整合用户故事。有工具支持对用户故事间关系的识别，通过关系推荐和人工确认的方法生成用户故事关系图。

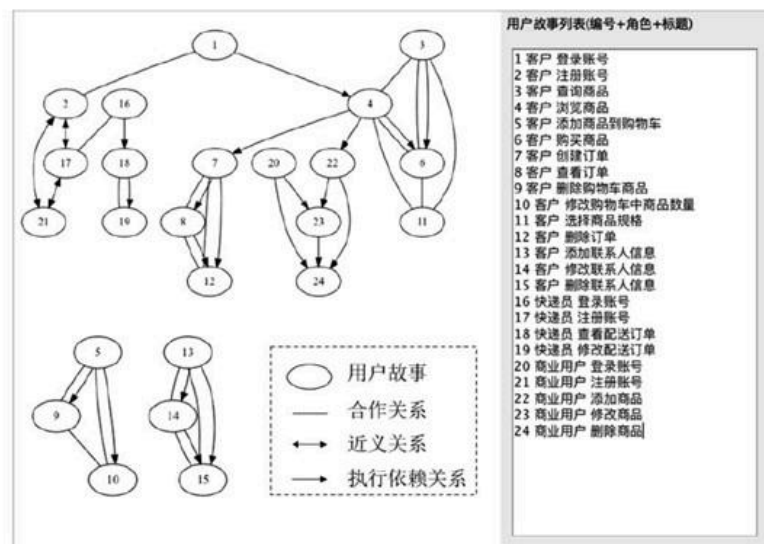
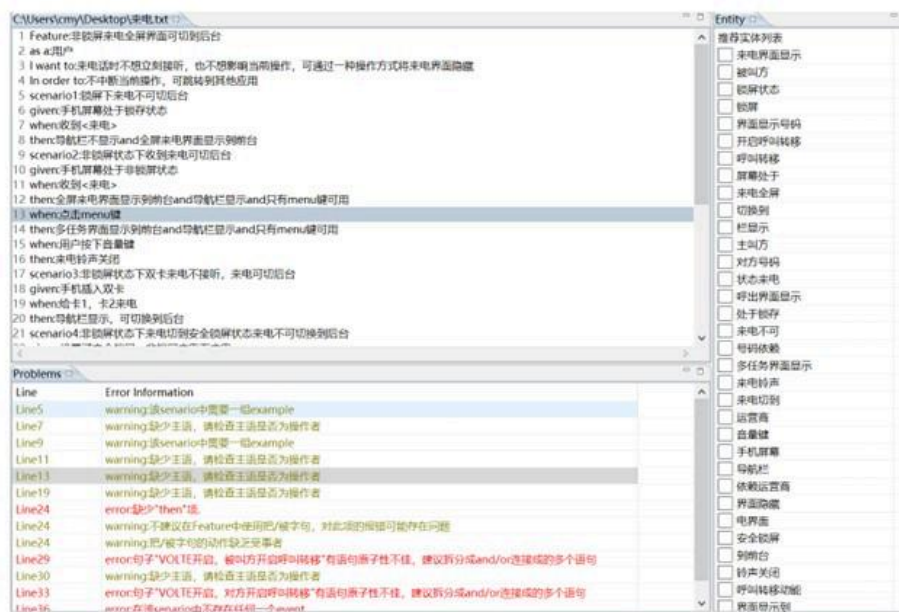


图 11.10 用户故事关系识别案例

敏捷需求管理工具

用户故事质量检查

用户故事可能存在不完整、不一致和不可测试等质量问题。采用需求验证方法，能够发现用户故事中存在的质量问题，并报告可能存在的问题。



文

敏捷需求管理工具

常用工具

表 11.5 常用敏捷需求管理工具比较

	Platform based ¹	Web based ¹	Online ¹	Cloud Based ¹	Burn Down Chart ¹	User Story Boards ¹	Milestones ¹	Resource Management ¹	Time Tracking ¹	Bug Tracking ¹	Tasks ¹	Integration ¹	Report ¹	Version Control ¹	Workspaces ¹	User role ¹	Evaluating ¹	Free Version ¹
Jira ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Active Collab ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Agile for Scrum ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Spira Team by Inflectra ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Pivotal Tracker ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Visual paradigm ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Icesrum ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
SprintGrou nds ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
VersionOne ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Taiga ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Agilant ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Wrike ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Trello ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Axosoft ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Planbox ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²
Rally ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²	√ ²

常用的敏捷需求管理工具有Jira、Agile for Scrum、VersionOne、Rally、Planbox、Visual Paradigm等。这些工具在使用方式、项目规划和项目追踪以及是否为免费版本等方面均有不同。

- 使用方式：有的工具基于平台(Platform based)，而有的工具基于网络(Web based)并支持在线(Online)使用，也有的工具基于云服务(Cloud based)
- 在项目追踪上，不同工具采用不同种类的图形化表示，帮助项目团队监察用户故事开发进度及开发过程中的问题。
- 在项目规划上，有的工具采用任务(Tasks)、报告(Report)或者对用户故事需求进行整合(Integration)的方式形成需求文档；有的通过创建工作空间(Workspaces)、用户故事角色划分(User Role)以及评估(Evaluating)和版本控制(Version Control)等方式促进干系人间的交互与协同。

思考题

思考题

根据以下描述，编写用户故事，估算用户故事的故事点数，确定优先级，并结合版本规划绘制用户故事地图。

在医生的办公室里，接待员、护士和医生使用病人记录和计划安排系统。当病人第一次来这里看病时，接待员使用该系统来输入病人信息，并且安排所有的预约。护士使用系统来跟踪病人每次看病的结果并输入护理病人的信息，如医疗和诊断。护士也可以访问这些信息以打印病人诊断结果或病人看病历史。医生主要用这个系统来查看病人的病史，偶尔也输入病人医疗信息，但通常让护士输入这些信息。

请基于你的用户故事，回答以下问题：

1. 用户故事的角色都有哪些？
2. 你觉得哪些用户故事应该写故事测试？针对需要写测试的用户故事，请使用Gherkin语言写出完整的测试场景。
3. 有比较大的用户故事吗？这些大故事是否需要拆分？如果有，请列出。
4. 是否有用户故事粒度太细，需要合并？如果有，请列出。
5. 是否有用户故事重复？如果有，请列出。
6. 是否有用户故事存在场景冲突问题？如果有，请列出。
7. 哪些用户故事之间存在依赖关系？请列出。
8. 哪些用户故事之间存在合作关系？请列出。
9. 你计划安排几名开发者，经过几轮迭代完成该系统的开发？请使用Visual Paradigm或ProcessOn等绘图工具绘制用户故事地图，展示你的规划。

参考文献

- [1] Manifesto for Agile Software Development [EB/OL] .
[2023.04.30] .<http://agilemanifesto.org/iso/en/manifesto.html>.
- [2] Douglass,Bruce Powel.Agile Systems Engineering [M] .Burlington,MA: Morgan Kaufmann,2015.
- [3] Jyothi V E and Rao K N.Effective Implementation of Agile Practices: Ingenious and Organized Theoretical Framework [J] .(IJACSA) International Journal of Advanced Computer Science and Applications,Vol.2,No.3,March 2011.
- [4] Jeffries R.Essential XP: Card,Conversation,and Confirmation [J/OL] .XP Magazine,2022,8 [2022 12 05] .<https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>.
- [5] Patton J.User Story Mapping.Discover the Whole Story,Build the Right Product [M] .Sebastopol,CA,USA: O Reilly Media,2014.
- [6] Beck K.Extreme Programming Explained: Embrace Change [M] .Boston,MA: Addison Wesley,2004.
- [7] Wynne M,Hellesoy A.The Cucumber Book: Behaviour Driven Development for Testers and Developers [M] .Pragmatic Bookshelf,2012.
- [8] Beck K.Test Driven Development: by Example [M] .Boston,MA: Addison Wesley,2002.
- [9] 王春晖,金芝,赵海燕,等.人机协作的用户故事场景提取与迭代演进 [J] .软件学报,2019,30(10): 3186 3205.
- [10] 王春晖,金芝,赵海燕,等.一种用户故事需求质量提升方法 [J] .计算机研究与发展,2021,58(04): 731 748.
- [11] Ghezzi C.Formal Methods and Agile Development: Towards a Happy Marriage [C] .The Essence of Software Engineering.Springer,2018.
- [12] Zkan D,Mishra A.Agile Project Management Tools: A Brief Comparative View [J] .Cybernetics and Information Technologies,2019,19(4): 17 25.
- [13] Rally [EB/OL] . [2023 04 30] .https://www.broadcom.com/products/software/value_stream_management/rally.
- [14] Visual Paradigm [EB/OL] . [2023 04 30] .http://www.visual_paradigm.com/cn/.