



THE UNIVERSITY OF
MELBOURNE

SWEN90016

软件流程 & 项目管理

质量保证
计划、控制、监测



1. 了解测试 **敏捷**
2. 测试 **正式的** - 为您的团队准备的清单





QA 在敏捷中的作用是什么？

- 一种) 在开发之后，敏捷团队在多个冲刺中进行了单独的测试
- b) 由于开发周期很快，因此没有时间进行测试
- C) 敏捷旨在快速适应变化并最大限度地减少时间，因此无需测试
- d) 测试在每个冲刺中完成
- e) 开发和测试之间的持续集成



每个冲刺都有自己的测试阶段。

每次发布新功能时都可以运行测试。

在敏捷测试中，一小块工作软件交付给客户
冲刺结束。

测试人员和开发人员在敏捷测试中紧密合作。测试由
整个团队。

用户接受在每个冲刺结束时执行。

用户故事描述了需求……

**验收标准提供了用户故事何时完成的
定义。**

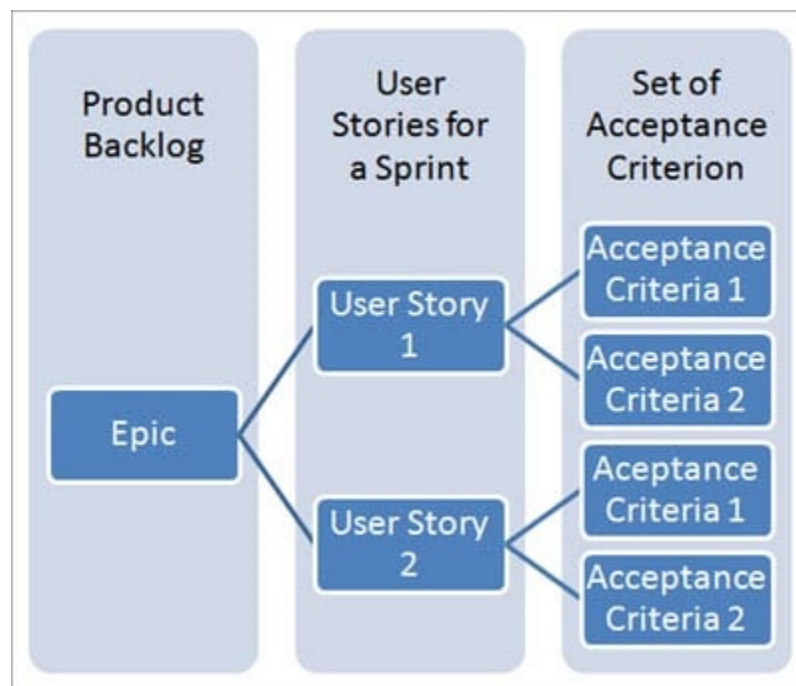


**作为客户……我希望能够分摊我的付款……
所以我可以使用多张借记卡付款**

活动：在分组讨论中，为上述用户故事集思广益三 (3) 条验收标准

例子：

- 用户可以在支付页面选择“分期付款”
- 用户可以选择不同类型的付款方式
- 用户可以指定他们想要拆分的数量
- 当用户输入他们的分期付款时，付款页面会自动计算需要支付的费用



<https://www.softwaretestinghelp.com/userstory-acceptance-criteria/>



例子:

- 作为一个 **用户** 我可以在付款页面选择“分期付款” **以便** 我可以选择使用多张卡 **什么时候** 我来买单
- 作为一个 **用户** 我可以选择不同类型的付款方式 **以便** 我可以付钱
- 作为一个 **用户** 我可以指定要在付款选项之间分配的金额 **以便** 我可以使用多张卡 **什么时候** 我来买单。
- 作为一个 **用户** 付款页面应自动计算需要支付的金额 **以便** 用户知道还需要支付多少 **什么时候** 用户输入他们的分期付款

<https://www.softwaretestinghelp.com/userstory-acceptance-criteria/>

敏捷 QA 桌面审计障碍：

邀请多技能的观众进行案头审计：业务分析师、另一位开发人员 and 一名测试人员

在允许将代码提交到共享之前，在开发人员的办公桌上查看代码
混帐存储库，GitHub。

一旦代码提交到 GitHub，它的测试套件就会立即由持续集成工具运行

CI 工具显示运行代码的通过/失败状态



Sprint Review QA 评估:

- 构建具有最少功能的小块工作软件
- 向利益相关者展示产品块 **早期的**
- 失败 **快速地** 并尽可能便宜，并获得及时的反馈
- 捕捉 **技术债务项目** 在产品待办列表中，（可选 FDD 格式）
- 产品负责人设定优先级 **技术债务项目**

What
Where
Who
When
Why



将 QA 需求写成用户故事

作为敏捷 Scrum 团队：

5个。

我们想要一个质量计划，以便我们的 Sprint 具有很强的质量管理重点

作为质量保证 设计 团队：

湾

我们需要一个 QA 清单，以便在定义的时间评估关键类别和属性

作为系统管理员：

C。

我想要一个密码策略指南，以便我们的应用程序具有有用的过程

我想要一个密码策略清单，以便我们的应用程序高度安全



另一个例子

用户的故事： 作为用户，我希望能够恢复我的密码
帐户，这样我就可以在忘记密码的情况下访问我的帐户。

设想： 忘记密码

鉴于： 用户已导航到登录页面 **什么时候：** 用户选择 忘
记密码 选项

和： 输入有效的电子邮件以接收密码恢复链接 **然后：** 系统发送链接到
输入的邮箱 **鉴于：** 用户通过电子邮件收到了链接

什么时候： 用户浏览了电子邮件中收到的链接 **然后：** 系统允许用户
设置新密码

[https://www.softwaretestinghelp.com/
userstory-acceptance-criteria/](https://www.softwaretestinghelp.com/userstory-acceptance-criteria/)

正式方法



一种 工具

Quality Attributes	Definition According to McCall et al.
Correctness	The extent to which a program satisfies its specifications and fulfils the user's mission objectives.
Reliability	The extent to which a program can be expected to perform its intended function with required precision.
Efficiency	The amount of computing resources and code required by a program to perform a given function.
Integrity	The extent to which access to software or data by unauthorised persons can be controlled.
Usability	The effort required to learn, operate, prepare input, and interpret output of a program.
Maintainability	The effort required to locate and fix an error in an operational program.
Testability	The effort required to test a program to ensure that it performs its intended function.
Flexibility	The effort required to modify an operational program.
Portability	The effort required to transfer a program from hardware and/or software environment to another.
Reusability	The extent to which a program (or parts thereof) can be reused in other applications.
Interoperability	The effort required to couple one system with another.



Checklist for software requirements specification artifact

Organisation and Completeness

- ☐ Are all internal cross-references to other requirements correct?
- ☐ Are all requirements written at a consistent and appropriate level of detail?
- ☐ Do the requirements provide an adequate basis for design?
- ☐ Is the implementation priority of each requirement included?
- ☐ Are all external hardware, software, and communication interfaces defined?
- ☐ Have algorithms intrinsic to the functional requirements been defined?
- ☐ Does the specification include all of the known customer or system needs?
- ☐ Is the expected behaviour documented for all anticipated error conditions?

Correctness

- ☐ Do any requirements conflict with or duplicate other requirements?
- ☐ Is each requirement written in clear, concise, unambiguous language?
- ☐ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ☐ Is each requirement in scope for the project?
- ☐ Is each requirement free from content and grammatical errors?
- ☐ Is any necessary information missing from a requirement? If so, is it identified as “to be decided”?
- ☐ Can all of the requirements be implemented within known constraints?
- ☐ Are any specified error messages unique and meaningful?

Quality Attributes

- ☐ Are all performance objectives properly specified?
- ☐ Are all security and safety considerations properly specified?
- ☐ Are other pertinent quality attribute goals explicitly documented and quantified, with the acceptable tradeoffs specified?

Traceability

- ☐ Is each requirement uniquely and correctly identified?
- ☐ Is each software functional requirement traceable to a higher-level requirement (e.g. system require-



一种 工具



在您的分组讨论中，**讨论** 质量流程和做这些活动

2. 创建一个适当的正式清单来审查小组分配

- 是否已决定由谁提交？
- 有没有人从头到尾审查过所有部分？
- 字数是否符合规范？
- 部分格式是否与作业模板一致？
- 你写的内容是否反映了讲座和教程？

3. 描述 **结果** 这篇评论？

完毕！