

# 让测试敏捷起来

江苏城市职业学院

态度决定高度，细节决定成败  
立足客户需求，为客户创造最大价值



# 让测试敏捷起来

1

敏捷测试概述

2

敏捷测试与传统测试的区别

3

敏捷测试中测试人员的价值与挑战

4

敏捷测试思路与探讨



# 让测试敏捷起来

## ➔ 关于预防火灾的三个方式：

- ◆ 明明只是应付变化，是事后挽救型的：出了问题才去灭火；
- ◆ 聪聪想到了变化，但只是为变化而变化，是一边灭火一边想着防火；
- ◆ 慧慧则是使自己处于应变的状态之中，随时准备变化，是一边认真地防火，一边随时准备着灭火。

## ➔ 问题讨论：

- ◆ 是谁以最小的代价达到了目标？
- ◆ 我们软件测试又怎样才能以最小的代价才能达到目标呢？



# 让测试敏捷起来

## ➔ 软件测试的目标

- ◆ 我们的软件测试目标不是找bug，而是预防bug，零缺陷是我们测试的最终目标。

## ➔ 软件测试的未来

- ◆ 现在软件测试大多数还只是停留在找bug阶段，而如果真的要做好产品的话要在bug还没出现以前就将其杜绝，这才是软件测试的未来。



# 让测试敏捷起来



怎样才能实现软件测试的目标？

答案：敏捷测试！



# 让测试敏捷起来

1

敏捷开发

2

敏捷测试的定义

3

敏捷测试的核心价值观

4

敏捷宣言



# 让测试敏捷起来

## → 敏捷开发：

敏捷开发是一种以人为核心、迭代、循序渐进的开发方法。在敏捷开发中，软件项目的构建被切分成多个子项目，各个子项目的成果都经过测试，具备集成和可运行的特征。换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

2001年二月，一组由17位在DSDM，XP，Scrum，FSD等领域的专家组成的代表团齐聚美国犹他州，寻找这些方法的共同点。最终，这些专家制定并宣布了敏捷开发宣言。由此形成了现在我们所认识的敏捷开发和后来的敏捷联盟。



# 让测试敏捷起来

## → 敏捷开发：

在软件工业界，敏捷开发已成为众多高效开发团队的制胜之道。它不仅被许多中小公司青睐，在全球一百强的企业中，敏捷也已大行其道，受到许多资深项目管理者 and 开发人员的推崇。欧美软件企业中，有近半企业已采用敏捷方法进行开发。大多数尚未应用敏捷的企业，也都对其有所了解，而且很多在计划实施。中国的外企，外包公司和许多知名企业也都开始采用了敏捷方法。例如，腾讯内部几乎所有的开发团队都在实施敏捷。

敏捷方法给这些企业也已带来了巨大的收益。据业内资深人士和长期从事敏捷咨询的服务公司透露，采用敏捷开发的团队一般会提高3-10倍的效率，软件的质量也有了更加可靠的保证。同时，敏捷开发的应用也给团队内的每个成员提供了良好的发展机会。他们的技术和合作水平都能得到响应的提高。敏捷的成功来源于其方法本身的适用性和团队对它的深入理解和合理运用。



# 让测试敏捷起来

→ 敏捷开发引发的两个问题：



到底什么是敏捷软件测试？



敏捷软件开发还需要测试工程师吗？

# 让测试敏捷起来

1

敏捷开发

2

敏捷测试的定义

3

敏捷测试的核心价值观

4

敏捷宣言



# 让测试敏捷起来

## → 到底什么是敏捷软件测试？

其实很难给敏捷测试下一个精确、完善的定义，一般来讲，只要接纳了敏捷的核心价值观(沟通，简单，反馈，勇气，尊重)，在敏捷软件开发过程中开展的测试就可以被称作是敏捷软件测试。

因此，敏捷软件测试并不是一个与敏捷软件开发同一层次的划分，而是敏捷软件开发中的一部分，与传统的测试不同，敏捷软件测试并不是一个独立的过程，相反，它与整个敏捷开发中的其他活动交织在一起，处处都能看到它的影子。由于敏捷软件测试并不倾向于一个单独的过程定义。



# 让测试敏捷起来

## → 到底什么是敏捷软件测试?

敏捷测试(Agile testing)是测试的一种，原有测试定义中通过执行被测系统发现问题，通过测试这种活动能够提供对被测系统提供度量等概念还是适用的。

敏捷测试是遵循敏捷宣言的一种测试实践：

- 1、强调从客户的角度，即是从使用系统的用户的角度，来测试系统。
- 2、重点关注持续迭代的测试新开发的功能，而不再强调传统测试过程中严格的测试阶段。
- 3、建议尽早开始测试，一旦系统某个层面可测，比如提供了模块功能，就要开始模块层面的单元测试，同时随着测试深入，持续进行回归测试保证之前测试过内容的正确性。



# 让测试敏捷起来

## → 到底什么是敏捷软件测试？

在敏捷测试流程中，参与单元测试，关注持续迭代的新功能，针对这些新功能进行足够的验收测试，而对原有功能的回归测试则依赖于自动化测试。由于敏捷方法中迭代周期短，测试人员尽早开始测试，包括及时对需求、开发设计的评审，更重要的是能够及时、持续的对软件产品质量进行反馈。简单地说，敏捷测试就是持续地对软件质量问题进行及时地反馈。

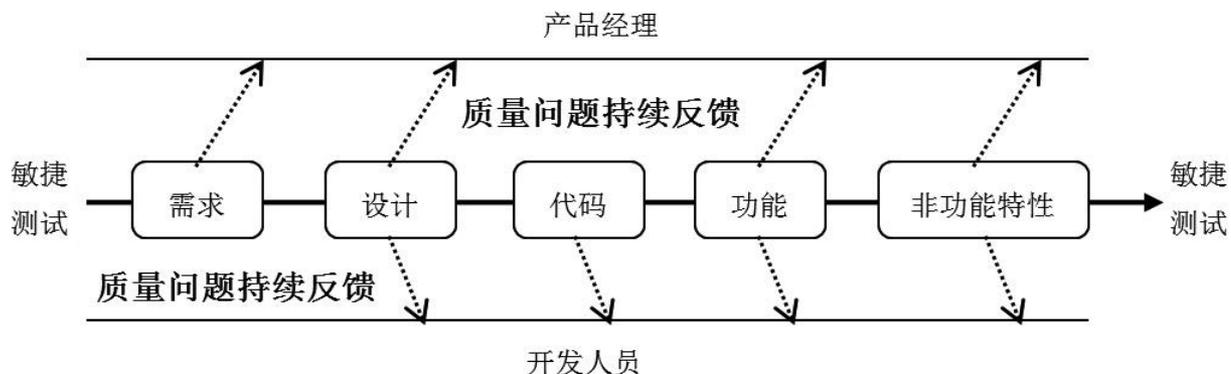


图1 敏捷测试定义的形象描述



# 让测试敏捷起来

→ 到底什么是敏捷软件测试?

## 敏捷测试流程的优化

在敏捷方法中，需求变化比较快、产品开发周期很短，功能不断累加，给软件测试带来很大的挑战，软件测试流程要做相应的调整。例如，我们原有的测试规范明确规定，首先要建立项目的主测试计划书，然后再建立每个功能任务的测试计划书，测试计划书有严格的模板，而且需要和产品经理、开发人员讨论，并和测试团队其他人员（包括测试经理）讨论，最终得到大家的认可和签字才能通过，仅测试计划经过“起草、评审和签发”一个完整的周期就需要一个月。在敏捷方法中，不再要求写几十页的测试计划书，而是在每个迭代周期，写出一页纸的测试计划，将测试要点（包括策略、特定方法、重点范围等）列出来就可以了。



# 让测试敏捷起来

## → 到底什么是敏捷软件测试？

在原有测试规范中，要求先用Excel写出测试用例，然后进行讨论、评审，评审通过以后再导入测试用例库（在线管理系统）中。在敏捷测试中，可能不需要测试用例，而是针对Use Case或User Story直接进行验证，并进行探索性测试。而节约出来的时间，用于开发原有功能的自动化测试脚本，为回归测试服务。自动化测试脚本将代替测试用例，成为软件组织的财富。原有测试规范还要求进行两轮回归测试，在敏捷测试中，只能进行一轮回归测试。综合这些考虑，敏捷测试的流程简单有效。

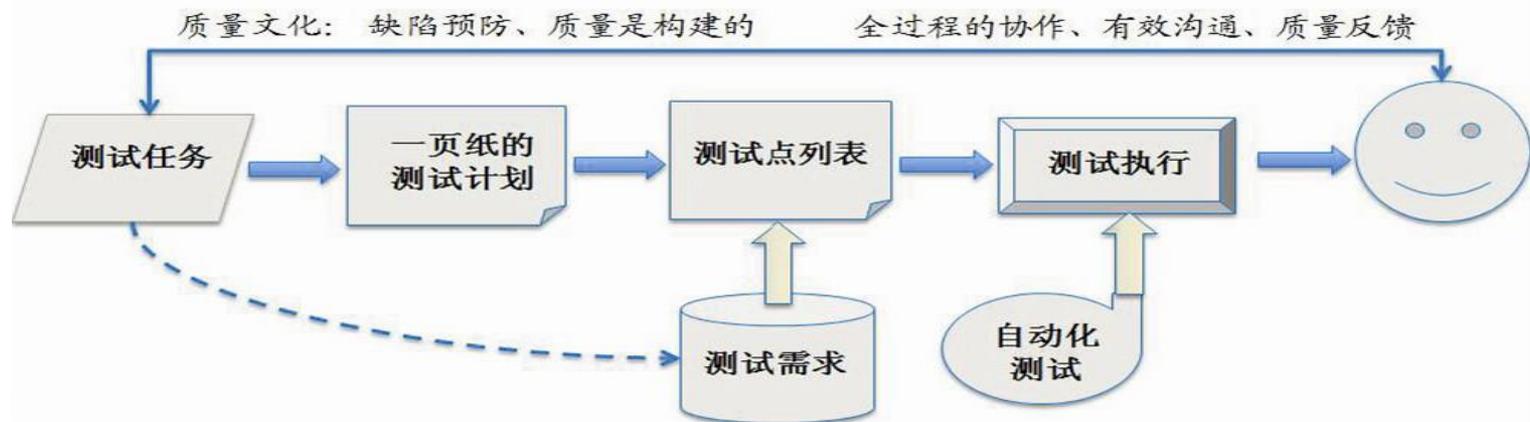


图2 敏捷测试流程简要图

# 让测试敏捷起来

## → 到底什么是敏捷软件测试?

在敏捷测试流程中，如前所述，测试是一个持续的质量反馈过程，测试中发现的问题要及时反馈给产品经理和开发人员，而且某些关键方面也要得到我们足够的关注，主要有：

- ◆ 测试人员不仅要全程参与需求、产品功能设计等讨论，而且要面对面地、充分地讨论（包括带语言、视频的即时通讯），仅仅通过邮件是不够的。
- ◆ 参与代码复审（Code Review），并适当辅助开发人员进行单元测试。
- ◆ 在流程中增加一个环节“产品走查（Product Walk-through）”——测试人员和产品经理、开发人员等在一起，从头到尾将新功能看一遍，可直观、快速地发现问题。



# 让测试敏捷起来

1

敏捷开发

2

敏捷测试的定义

3

敏捷测试的核心价值观

4

敏捷宣言



# 让测试敏捷起来



敏捷的核心价值观：



- ◆ 沟通 (Communication)
- ◆ 简单 (Simplicity)
- ◆ 反馈 (Feedback)
- ◆ 勇气 (Courage)
- ◆ 尊重 (respect)



# 让测试敏捷起来

## → 敏捷的核心价值观：

### ◆ 沟通

沟通对创造团队的气氛和有效的协作很重要。每个人都是团队中的一员，并且每天面对面的交流。团队成员从需求分析到代码编写都一直工作在一起，共同创造最好的解决问题的方法，这样做的目标是让所有开发人员对系统有共同的理解，并且和用户对系统的理解相对应。

### ◆ 简单

简单是指让系统简单到只是足够解决好今天的问题，其他的功能可以在以后加入。这使得开发人员可以集中注意在设计 and 开发今天的需求上，而不是明天的需求。这样做的好处是可以使至今为止的投资回报率最大化，而不是投资在可能会改变的将来的需求上。达到了简单，需要交流的东西就会少很多，这样也可以增进交流的质量。



# 让测试敏捷起来

## → 敏捷的核心价值观：

### ◆ 反馈

没有一个固定的方向会长期有效，改变是避免不了的。有改变就需要反馈。XP试图产生尽可能多的来自不同方向的反馈，越早知道，越容易调整：

代码改变之后从系统得到的反馈：通过单元测试和阶段性集成测试；

从客户那里得到的反馈：通过检查每个迭代交付的可工作的软件；

从团队本身得到的反馈：通过团队成员的交谈改进流程。

反馈对交流和简单都很重要。反过来，简单使反馈变得容易。



# 让测试敏捷起来

## → 敏捷的核心价值观：

### ◆ 勇气

勇气是面对恐惧的有效回应。勇气是指真实地说出进展和估计；不用文档记录失败的借口，即使我们计划着成功；不去害怕任何事情，因为没有人是单独工作的。勇气和其他价值观一起会倍显强大，说真话的勇气可以培养交流与信任。抛弃失败方案的勇气可以鼓励做到简单。寻求真相的勇气可以创造反馈。

### ◆ 尊重

有了之前四个价值观后，就会获得团队其他成员的尊重，团队中没有人应该有不受欢迎或者被忽视的感觉，这可以保证工作的主动积极性并鼓励大家忠诚于团队，忠诚于项目目标。此外，开发人员应该尊重客户之所长，反之亦然。管理层应该尊重团队行使职责的权利，也相应地得到对团队的威信。



# 让测试敏捷起来

1

敏捷开发

2

敏捷测试的定义

3

敏捷测试的核心价值观

4

敏捷宣言



# 让测试敏捷起来

## 敏捷宣言：

- ◆个体和交互 重于 过程和工具
- ◆可用的软件 重于 完备的文档
- ◆客户合作 重于 合同谈判
- ◆响应变化 重于 遵循计划



# 让测试敏捷起来

## → 敏捷宣言：

### ◆ 个体和交互重于过程和工具：

方法和工具是死的，人是活的，如何没有优秀个人和团队协作，再强大的方法和工具都是摆设。一个使用普通工具的优秀人员会比使用优秀工具的普通人员做得更好。

虽然我们致力于个体和交互，但并不是不需要过程与工具了。敏捷测试方法本身也有一些方法和过程，每日构造等敏捷实践也需要工具的支持，需要哪些过程和工具由自组织团队制定，而不是由领导制定。



# 让测试敏捷起来

## → 敏捷宣言：

### ◆ 可用的软件重于完备的文档：

在合同中有时会看到分别在需求、设计、开发、测试阶段提供什么文档，支付多少金额等内容，而这些文档对用户来说是不是真的有价值呢？面面俱到的文档对客户来说并不重要，用户需要的是一个能够运行起来，能够实质解决工作中问题的可以工作的软件。面面俱到的文档对开发团队也不重要，上百页的报告没有人愿意写，更没有人愿意去读，对开发团队来说最好的两份文档就是代码和团队。

虽然我们致力于提供可供做的软件，但并不是不要文档。我们在开发过程中仍然需要进行内部交流，也需要和客户交流，我们仍旧可能需要制作原型，书写一些主要需求和算法，只要自组织团队认为足够就行了。



# 让测试敏捷起来

## → 敏捷宣言：

### ◆ 客户合作重于合同谈判：

寻求客户合作的价值重于对合同的谈判。软件开发的最终目标是提供给客户满意的软件，而只有客户才更清楚怎么样才能满意，敏捷开发提倡客户和开发团队密切的在一起工作，并尽量经常行得提供反馈。各种不同的敏捷方法都会利用短期迭代，通过尽早提供软件来达到与客户频繁沟通和反馈的，这也可以把问题及早暴露出来，以免隐藏的问题在后期造成更大的影响。

虽然我们致力于客户协作，但为了双方利益和需要仍旧需要进行合同谈判。



# 让测试敏捷起来

## → 敏捷宣言：

### ◆ 响应变化重于遵循计划：

计划赶不上变化，敏捷项目承认开发过程中的不确定性，所以不会在开发中制定长时间的复杂计划，它们的成功都是建立在现实反馈的基础上的。通过迭代开发，每次迭代都是基于上一迭代的完善基础之上进行的，通过不断的响应变化来消除开发中的不确定性。

虽然我们致力于响应变化，但并不是像上面漫画所说的不需要计划了，反而我们需要更多的规划。



# 让测试敏捷起来

## → 敏捷宣言：

### ◆ 响应变化重于遵循计划：

规划是困难的，而且作出的计划常常会出错，面对这样的问题，开发小组往往会走上两个极端：要么根本不做任何规划，要么在计划中投入大量的精力直到自己确信计划是正确的。不做规划的小组对一些最基本的问题，例如“你们什么时候能完成？”以及“我们可以在6月份安排产品发布吗？”都无法回答。而做了大量计划的小组会自欺欺人地认为某个计划是“正确的”。他们的计划也许更全面，但这并不一定意味着它更准确或更有用。这两种极端都是敏捷需要避免的。

“我们实施敏捷，不再需要计划和文档了”的论调是及其错误的。



# 让测试敏捷起来

## → 敏捷宣言：

### ◆ 响应变化重于遵循计划：

敏捷不是不需要计划，相反它需要更多的规划。不确定性是影响计划正确的主要因素，对大部分不确定而言，在获取知识、减少不确定性的唯一办法是通过执行—作一些事情、构建一些东西或模拟一些东西—然后获得反馈。许多项目管理方法是“规划、规划、规划—执行”，而敏捷开发方法是“规划—执行—调整”、“规划—执行—调整”。一个项目的不确定性越高，敏捷开发方法对取得成功就越是至关重要，不断学习和调整是敏捷开发的核心。



# 让测试敏捷起来

1

敏捷测试概述

2

敏捷测试与传统测试的区别

3

敏捷测试中测试人员的价值与挑战

4

敏捷测试思路与探讨



# 让测试敏捷起来

## → 敏捷测试与传统测试的区别：

传统的测试模式基于如下的一些理念：

- ◆ 测试是质量的最后保护者；
- ◆ 严格的变更管理；
- ◆ 预先的计划和细节的准备；
- ◆ 重量级文档；
- ◆ 严格的各阶段测试入口和出口标准；
- ◆ 回归测试阶段重量级的自动化测试；
- ◆ 企图流程改善和执行；
- ◆ 测试团队和开发团队是可分割的。



# 让测试敏捷起来

## → 敏捷测试与传统测试的区别：

那么对照传统的测试模型，敏捷测试颠覆了以上观念：

- ◆ 测试是质量的最后保护者——开发和测试人员是紧密合作，大家都有责任对软件负责；
- ◆ 严格的变更管理——变更是可接受的，拥抱变更；
- ◆ 预先的计划和细节的准备——计划随时进展时常调整；
- ◆ 重量级文档——只需要绝对必要的文档；
- ◆ 严格的各阶段测试入口和出口标准——各迭代之间已经没有明显的入口和出口标准；
- ◆ 回归测试阶段重量级的自动化测试——所有阶段都需要自动测试，每个人都需要做，是项目集成的一部分；
- ◆ 企图流程改善和执行——流程不再需要严格执行；
- ◆ 测试团队和开发团队是可分割的——团队合作是无缝隙合作；



# 让测试敏捷起来

## 敏捷测试

自组织团队与客户紧密协作，通过高度迭代式、增量式的软件开发过程响应变化，并在每次迭代结束时交付经过编码与测试的有价值的软件。

## 传统测试

与客户确定合同后在初期制定并遵循基于活动的完整计划，在重型过程和工具指导下，通过完成大量文档进行知识传递，最后交付需求。

胜手



# 让测试敏捷起来

1

敏捷测试概述

2

敏捷测试与传统测试的区别

3

敏捷测试中测试人员的价值与挑战

4

敏捷测试思路与探讨



# 让测试敏捷起来

1

敏捷测试中测试人员的价值

2

敏捷测试中测试人员的挑战

3

敏捷测试成功的关键要素



# 让测试敏捷起来

## → 敏捷测试中测试人员的价值

### 敏捷测试中测试人员扮演的角色：

- ◆ 测试是项目的“车头灯”，它告诉大家现在到哪了，正在往哪个方向走。
- ◆ 测试为项目组提供信息，使得项目组基于可靠的信息作出正确的决定。
- ◆ “BUG”是让用户感觉烦恼的东西，测试人员不作出发布的决定。
- ◆ 测试员不保证质量，整个项目组对质量负责。
- ◆ 测试不是抓虫子的游戏，它的目的不是纠缠在错误中，而是帮助找到目标。



# 让测试敏捷起来

## → 敏捷测试中测试人员的价值

- ◆ 在需求和功能设计讨论上，测试人员可以站在客户角度来阐述自己的观点，扮演“用户代表”角色，强调用户体验，真正体现测试人员和开发人员的互补作用。
- ◆ 测试人员不仅扮演“用户代表”角色，而且通过需求讨论、代码复审等各种活动及时地提供质量反馈，包括代码质量、接口一致性等，保证在产品构造的整个过程中质量受到足够的关注，以提高质量改进的持续性和可视性。
- ◆ 测试人员应积极参与单元测试，即使不参加单元测试，也应督促开发人员进行单元测试，确保单元测试达到80%以上覆盖率，确保开发出具有良好可测试性的代码。
- ◆ 在敏捷方法中，往往将一个大的系统开发分解成多个小的子系统（模块或组件），集成测试和端到端（End-to-End）测试显得更为重要，测试人员在这些测试上能发挥更大的作用。
- ◆ 产品发布前，验收测试和回归测试依然不可缺少，这更是测试人员的用武之地。
- ◆ 一个迭代周期结束后，对缺陷根本原因进行分析、总结规律，帮助开发人员建立良好的习惯，预防缺陷，从根本上提高产品质量。



# 让测试敏捷起来

1

敏捷测试中测试人员的价值

2

敏捷测试中测试人员的挑战

3

敏捷测试成功的关键要素



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 为什么以前的开发模式不再适用？

- ◆ 以前的开发模式要求有详细的测试计划，但是缺乏足够的时间来写，而且测试计划受很多因素的影响经常改变。
- ◆ 以前的开发过程会专门留出一个阶段来测试，但是你不能定义进入和退出的标准，测试阶段会随之而过。
- ◆ 以前的开发模式强调变更控制，但是现在的软件需求变更非常频繁，变更成了家常便饭。
- ◆ 以前的开发模式要求测试要对软件做出权威的判断，但是测试很难做出权威的关于软件正确性的判断。

### 测试的作用

- ◆ 有价值的东西要么提供产品，要么提供服务。那么测试提供什么产品或服务呢？有人认为测试提供调试通过的、经过测试的软件。
- ◆ 这是错误的回答。测试不提供产品，测试提供信息，关于开发过程中的软件的状态的信息，以便基于这些信息做出决定。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试中，测试人员面临七大挑战：

- ◆ 测试员是否不再需要了？
- ◆ 测试不完整的软件
- ◆ 可接受性测试是否过于简单了？
- ◆ 把测试员作为项目组的一部分
- ◆ 测试什么时候完成？
- ◆ 我们还需要bug跟踪系统吗？
- ◆ 用什么质量标准来度量敏捷项目？
- ◆ 回归测试和回归测试工具



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之一：测试员是否不再需要了？

既然有开发人员做单元测试了，我们还需要测试员吗？有些项目团队采用了敏捷开发方式后把测试员都给解雇了，然后过了不久他们就后悔了。

测试可以是除QA或测试员外的人来做，例如业务分析员，有些项目团队让开发人员来做接受性测试。

但是有专门的测试员带来两个好处：

- 1、专注于用户使用而不是软件的技术实现；
- 2、专注于发现软件的错误而不是证明完整性。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之二：测试不完整的软件

频繁的迭代产生的测试版本很多时候是不完整的，测试员如何测试这些不完整的代码呢？

“故事”应该从业务价值方面来定义。一个“故事”应该在一个迭代周期内完成。好的“故事”是不容易定义出来的，但是差的故事对测试人员的影响比对开发人员的影响还要大。有时候测试人员需要帮助定义“故事”。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之三：可接受性测试是否过于简单了？

测试人员如果只是做可接受性测试，只是验证“故事”是否完整，岂不是太简单了？这样怎么能做好测试呢？

其实，每一个迭代都需要额外的测试，而不仅仅是局限于验证“故事”的完整性。

在迭代测试中还要按需进行下面类型的测试：

探索性测试：同时学习系统、计划和执行测试，寻找bug、遗漏的特性和改进的机会；

组合交互测试：专注于特性之间的交互；

场景测试：模拟真实世界的场景进行测试；

疲劳测试：长时间地执行软件；

业务循环测试：基于月末、季度末等业务循环的边界来执行场景；

压力测试：对系统施加强大的压力进行测试。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之四：把测试员作为项目组的一部分

把测试员作为项目组中的一员不是牺牲了他们作为一个组织的完整性吗？

测试员一直被认为是受压迫的对象，经常坐在一起互相诉苦、互相支持。现在是时候结束这种情况了。测试员应该跟开发人员和分析师坐在一起，当项目组中有更多的正式或非正式的沟通时才有可能达到敏捷。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之五：测试什么时候完成？

没有专门分配的时间来完成测试，我们怎么知道什么时候测试应该结束？

敏捷测试员需要根据项目和产品的风险来调整测试。基本上测试的优先级应该跟“故事”的优先级一致。BUG列表也提供了测试完整性的提示。

一个好的测试员是永远都能找到需要完成的测试来做的。

为什么需要跟开发人员结对进行测试呢？因为开发人员对潜在的错误有一定的洞察力，测试员对约束和错误的时机有一定的洞察力。而他们在一起能使自动化测试更加成功。

测试员应该测试的代码，重用单元测试的框架，使软件更加可测。

利用“灰自动化可接受性测试，使用与开发环境一样的编程语言来编写可接受性盒”测试。设法弄清楚系统各模块之间的关系，分析变更的影响，看什么是需要测试的，什么是可以不测试的。弄清楚bug，bug的表面现象是什么？产生bug的根本原因是什么？弄清楚风险，使用解决风险的测试策略，调整测试目标。

# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之六：我们还需要bug跟踪系统吗？

有些人说敏捷团队不需要跟踪bug，只需要把发现的bug尽快修正就行了。

这种做法只适用于开发过程的测试，如果是一个完整迭代的测试，你就需要bug跟踪系统，因为有些bug不是在这个迭代马上修改的。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之七：用什么质量标准来度量敏捷项目？

其中一个最好的质量标准是发布后逃逸的bug数量。不幸的是，这是个事后的衡量标准。

采用每个迭代后计算逃逸bug数量的方法能标识代码的质量。

我们还可以从bug学习到很多东西：

- 1、是否有些类型的bug在可接受性测试中发现的，其实是可以在单元测试就发现呢？如果是，把它加入到单元测试。
- 2、我们是否能让bug的发现过程或bug的诊断更简单？
- 3、我们是否能让程序员不那么容易犯这种普通的错误？



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之八：回归测试

伴随着频繁的迭代，我们需要频繁地重新测试，单元测试是不够的。我们怎样有效地进行用户层面的回归测试呢？

你不一定需要在每次的迭代都做完整的回归测试。可以每个迭代运行一部分的测试。需要某种程度上的用户层次的自动化回归测试。



# 让测试敏捷起来

## → 敏捷测试中测试人员的挑战

### 敏捷测试的挑战之八：回归测试工具

大部分的商业测试工具在敏捷环境下都不是很好用。大部分有这些缺点：

#### 1、指定的语言

大部分商业测试工具会指定某种语言，例如：**WinRunner**（TSL）、**SilkTest**（4test）、**Robot**（**Test** Basic），但是一些新的工具也开始使用标准语言，例如：**Astra QuickTest**（VB Script），**XDE Tester**（Java）  
参考<http://www.stickyminds.com/se/S2326.asp>

#### 2、与源代码控制的结合不好

很多工具没有与源代码控制工具集成，使用临时文件和目录（WinRunner），

关键信息存储在Repositories中，例如**Rational**

#### 3、很难与持续集成配合使用

缺乏外部调用的API，不允许作为一个库被使用，因此很难与持续集成整合在一起。一些新的工具则有所改进，例如**TestComplete**

#### 4、不能在所有机器上部署（受License限制）



# 让测试敏捷起来

1

敏捷测试中测试人员的价值

2

敏捷测试中测试人员的挑战

3

敏捷测试成功的关键要素



# 让测试敏捷起来

## → 敏捷测试成功的关键要素

### 敏捷软件测试的七个关键成功要素：

- ◆ 使用团队整体参与的方法
- ◆ 采用敏捷测试思维
- ◆ 自动化回归测试
- ◆ 提供并获取反馈
- ◆ 构建核心实践的基础
- ◆ 与客户合作
- ◆ 保持大局观等



# 让测试敏捷起来

## → 敏捷测试成功的关键要素

### 使团队整体参与：

当整个团队负责测试和质量问题，你会拥有很多不同的技能集合和经验等级来处理测试可能发生的问题。测试自动化对于技能高潮的开发人员来说不是大问题，当测试拥有优先权的时候，任何人都可以参与测试任务，团队才会涉及可测试的代码。

### 采用敏捷测试测试思维：

敏捷测试思维的一个重要部分是不停想办法改进工作。成功的敏捷测试人员会持续的磨练技能。读好书、博客和文章以获得新想法和技能。

应该使用敏捷准则和价值观指导你。不断尝试最简单的方法来满足测试需要，勇敢地寻求帮助和实验新想法。关注与产生价值，尽可能多的直接交流，灵活的对应变化。敏捷开发以人为中心，我们应该享受工作。



# 让测试敏捷起来

## → 敏捷测试成功的关键要素

### 自动化回归测试：

敏捷开发利用测试来知道开发，为了编写代码是测试通过，需要快速、简单的运行测试，没有短期反馈周期和安全的回归测试，团队将很快陷入技术债务，缺陷不断增加，速度越来越慢。

自动化回归测试是整个团队的工作。在开始时很艰苦，应确保在时间、培训、和激励上提供了足够的支持。

### 提供并获得反馈：

反馈是敏捷测试的核心价值。敏捷的短期迭代可以提供持续的反馈以帮助团队正常运转。测试人员沟通过自动化测试结果、探索性测试的发现和系统实际用户的观察结果的形式帮助提供反馈。

一个最有价值的技能是学习如何寻求自己工作的反馈，询问开发人员是否得到了足够的信息以理解需求并且能够知道编码，询问客户是够理解质量标准。



# 让测试敏捷起来

## → 敏捷测试成功的关键要素

### 构建核心实践的基础：

- ◆ 持续集成
- ◆ 测试环境
- ◆ 管理技术债务（自动化测试）
- ◆ 增量工作
- ◆ 编码和测试是同一个过程的组成部分
- ◆ 实践之间的协作



# 让测试敏捷起来

## → 敏捷测试成功的关键要素

### 与客户协作：

测试人员对敏捷团队的最大贡献之一就是帮助客户清理需求并设定优先级，通过与其行为和用户场景的具体示例描绘需求，并把这些示例转换为可执行的测试。

开发人员、客户、测试人员三方协作。当需求丢失或者被误解，客户、开发人员和测试人员需要一起解决问题。

### 保持大局观：

测试人员应该有大局观，从客户的角度看待问题。开发人员通常关注于实现当前的故事。



# 让测试敏捷起来

1

敏捷测试概述

2

敏捷测试与传统测试的区别

3

敏捷测试中测试人员的价值与挑战

4

敏捷测试思路与探讨



# 让测试敏捷起来

## → 敏捷测试思路与探讨

轻量的敏捷开发模型是为了提供给软件开发团队一种迅速应对客户需求变化，能够高效完成项目工作，降低整体风险的开发模式。敏捷的测试也是服务于这个目标的测试团队对测试工作的敏捷定义。

传统测试模式下成长起来的测试团队要如何转向敏捷，从个人和团队两个层面又要做出那些转变呢？有什么方法和标准衡量敏捷测试团队的绩效？如何帮助团队的每个人规划正确的发展路线？团队在部署敏捷的过程中又会遇到哪些问题呢？



# 让测试敏捷起来

## → 敏捷测试思路与探讨

什么样的项目才是最适合于敏捷方法，在此关于这方面进行一个探讨。在我看来，最适合敏捷方法的项目是那些有着激进的时间期限限制，那些有着高度的复杂程度，以及那些有着高度新颖性(独特性)的项目。

当我们在做一些新的事情，至少是对于开发团队是新的事情，的时候我们会比较愿意使用敏捷方法。如果这是一件团队以前曾经重复做过很多次的事情，他们很可能就不需要用敏捷的方法来做了。对我来说，这种时候就应该考虑引入类比制造的方法了。如果我们每天建造同一种车，我们很快就会了解到造车中的每一个细微差别。我们不需要一个敏捷的方法因为在这种情况下新颖性非常低。



态度决定高度，细节决定成败  
立足客户需求，为客户创造最大价值

Thank You!

