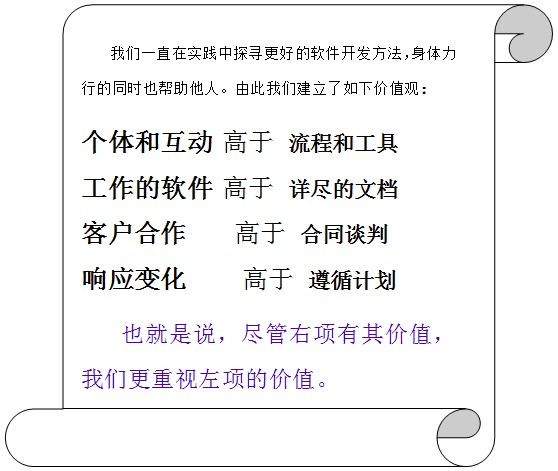
# 敏捷开发不提倡加班,敏捷测试更不提倡加班

<https://sanwen8.cn/p/120Pzrn.html> 瀑布开发模式与敏捷开发模式比较

“小步快跑，快速迭代”。为了实现单点突破就要允许不完美，但要快速迭代向完美逼近。每天都能发现修正一两个小问题，不到一年产品就打磨出来了。

**敏捷开发的4句宣言**



**什么是敏捷开发？**

敏捷开发(Agile Development)是一种以人为核心、迭代、循序渐进的开发方法。

怎么理解呢？首先，我们要理解它不是一门技术，它是一种开发方法，也就是一种软件开发的流程，它会指导我们用规定的环节去一步一步完成项目的开发；而这种开发方式的主要驱动核心是人；它采用的是迭代式开发；

**为什么说是以人为核心？**

我们大部分人都学过瀑布开发模型，它是以文档为驱动的，为什么呢？因为在瀑布的整个开发过程中，要写大量的文档，把需求文档写出来后，开发人员都是根据文档进行开发的，一切以文档为依据；而敏捷开发它只写有必要的文档，或尽量少写文档，敏捷开发注重的是人与人之间，面对面的交流，所以它强调以人为核心。

**什么是迭代？**

迭代是指把一个复杂且开发周期很长的开发任务，分解为很多小周期可完成的任务，这样的一个周期就是一次迭代的过程；同时每一次迭代都可以生产或开发出一个可以交付的软件产品。

3 几种常用的敏捷方法比较  
3.1 极限编程（XP）  
极限编程（简称XP）是由KentBeck于1996年提出的，极限编程要求把它列出的每一个思想和方法都做到极限、做到最好。  
极限编程的核心价值是我们在开发中必须注意的：Communication（沟通）、Simplicity（简单）、Feedback（反馈）、Courage（勇气）、此外还有第五个价值： Modesty（谦虚）。因为计划赶不上变化，使用极限编程的软件开发人员只需要在开发的初期做出一些文档。极限编程把软件测试放在首位，这样以后出现漏洞的几率就会降到最低。  
极限编程是一种近螺旋式的开发方法，它把复杂的开发分解为相对比较简单的小软件；通过沟通、反馈和其它的方法，客户和开发人员就可以清楚的了解到开发进度、变化、困难和急需解决的问题等，并及时地调整开发过程。  
3.2 SCRUM （迭代式增量软件开发过程）  
SCRUM的宗旨是发挥构件技术和面向对象的开发方法，对迭代式面向对象方法进行改进，适用于需求不确定的产品的开发。是迭代的增量化过程，便于工作管理和产品研发。更综合了各种开发的经验。  
SCRUM把项目分成N个为期15-30天的迭代阶段，称之为“冲刺”（sprint）。每个“冲刺”之前，你明确这一个“冲刺”需要实现的功能，然后让开发人员去完成。但是，在“冲刺”时，SCRUM的核心是所有开发都围绕着迭代，需求是固定的。SCRUM方法中只有3中角色：SCRUM主管、开发团队、产品负责人。  
3.3 动态系统开发方法（DSDM）  
开发一种面向领域的快速开发方法是产生动态系统开发方法的原因，动态系统开发方法在技术支持、应用推广、研究改进培训认证和培训认证等方面都比其他方法要完善，适用于对时间要求很紧的开发项目，动态系统开发方法应用范围不再仅仅局限于IT行业。  
DSDM方法提倡以业务为核心，快速而有效地进行系统开发，并提出了探索式开发方法的概念。强调软件使用者一开始就预见所有需求是不可能的。该方法中，只要进能入下一步，当前的算法就是可行的。  
3.4 水晶方法（Crystal）  
水晶方法是Alistair Cockburn于上世纪90年代末提出的，水晶方法目的是发展一种提倡“机动性的”方法。  
Crystal是根据项目重要性和规模来区别项目的，并给出相应的办法。所以，crystal是多种方法的组合.它阐明了要把对话和交流放在第一位的观点。Crystal方法中有两条准则：（1）应用反思工作室促使方法学的自适应，（2）使用的增量式循环不超过4个月。  
3.5 特性驱动开发（FDD）  
特性驱动开发是一个强调快速迭代、特性驱动的软件开发方法，适用于周期短的开发。它既能保证文档和质量，又能保证软件的快速开发，并提出划分的每一个功能开发时间不超过两星期，要求两星期内生产出可见的、能运行的代码。  
特性驱动开（www.t262.com）发方法认为简单的过程和良好的定义就能很好地被执行，它强调的是实用、简化、易于被开发人员接受，是一个特性驱动快速迭代的过程，适用的项目为软件需求经常变动。  
3.6 自适应软件开发（ASD）  
自适应软件开发方法的理论来源是复杂自适应系统理论，目的是通过提高自适应性用来应对互联网时代下的软件需求难于预测并高速变化的软件开发，它与水晶方法正在相互借鉴和融合。  
在一个环境中，结果是不可预测的，把计划看成是一个自相矛盾的。在计划中，偏离计划就是错误的，要纠正。而在一个适配性环境里，偏离计划恰恰是在引导开发人员走向正确的目标。在不可预测的环境中，需要我们用各式各样的方法来应对不确定性。在管理中，重点在于鼓励大家交流沟通，而不是告诉大家需要做什么，从而使开发人员能自己提出具有创造性的解决方案。

<http://blog.csdn.net/alvanchen/article/details/5749872/> 敏捷开发流程总结

# “小步快跑，快速迭代” 上线-反馈-修改-上线

# Agile Testing

[Test Script](http://softwaretestingfundamentals.com/test-script/)

[Test Case](http://softwaretestingfundamentals.com/test-case/)

## AGILE TESTING Fundamentals

This article on Agile Testing assumes that you already understand Agile software development methodology (Scrum, Extreme Programming, or other flavors of Agile). Also, it discusses the idea at a high level and does not give you the specifics.

本文介绍敏捷测试假设您已经了解了敏捷软件开发方法（Scrum，极限编程或其他敏捷敏捷）。另外，它在高层讨论这个想法，不给你细节。

**VERY SHORT DEFINITION**

Agile Testing is a [method of software testing](http://softwaretestingfundamentals.com/software-testing-methods/) that follows the principles of agile software development.

敏捷测试是一种遵循敏捷软件开发原理的[软件测试方法](http://softwaretestingfundamentals.com/software-testing-methods/)。



**MANIFESTO FOR AGILE SOFTWARE TESTING**

This is adapted from agilemanifesto.org and it might look a little silly to copy everything from there and just replace the term  development with testing but here it is for your refreshment. You need to however realize that the term development means coding, testing and all other activities that are necessary in building a valuable software.

We are uncovering better ways of testing

software by doing it and helping others do it.

Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation **Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on

the right, we value the items on the left more.

**AGILE TESTING VALUES EXPLAINED 敏捷测试概念解释**

个体和交互胜过过程和工具

* **Individuals and interactions over processes and tools:**This means that flexible people and communication are valued over rigid processes and tools. However, this does not

这意味着灵活的人和沟通被看重于死板的过程和工具。

mean that agile testing ignores processes and tools. In fact, agile testing is built upon very

然而,这并不意味着敏捷测试忽略了流程和工具。 事实上,敏捷测试是建立在非常简单的、强大的、合理的流程进行日常会议的过程或准备每日构建。

simple, strong and reasonable processes like the process of conducting the daily meeting or preparing the daily build. Similarly, agile testing attempts to leverage tools, especially for test

类似地，敏捷测试试图利用工具，特别是测试自动化，尽可能多。

automation, as much as possible. Nevertheless, it needs to be clearly understood that it is

不过， 需要清楚地认识到这一点

驱动这些工具的测试人员和工具的输出依赖于测试人员(而不是相反的)。

the testers who drive those tools and the output of the tools depend on the testers (not the other way round).

有效的软件重于完整的文档。

* **Working software over comprehensive documentation:**This means that functional and

这意味着功能和可用的软件被认为是全面而无用的文档。

usable software is valued over comprehensive but unusable documentation. Though this is more directed to upfront requirement specifications and design specifications, this can be true for test

尽管这更多的是针对预先需求的规范和设计规范，但是对于测试计划和测试用例来说也是如此。

plans and test cases as well. Our primary goal is the act of testing itself and not any elaborate

我们的主要目标是测试本身，而不是仅仅指向那个目标的任何复杂的文档。

documentation merely pointing toward that goal. However, it is always best to have necessary documentation in place so that the ‘picture’ is clear and the ‘picture’ remains with the team if/ when a member leaves.

但是，最好有必要的文档，这样“图片”就清楚了，如果一个成员离开，“图片”就留在团队里。

* **Customer collaboration over contract negotiation:**This means that the client is engaged frequently and closely in touch with the progress of the project (not through complicated progress reports but through working pieces of software). This does put some extra burden on the customer who has to collaborate with the team at regular intervals (instead of just waiting till the end of the contract, hoping that deliveries will be made as promised). But this frequent engagement ensures that the project is heading toward the right direction and not toward the building of a frog when a fish is expected.

客户合作重于合同谈判: 这意味着客户端频繁和密切联系项目的进展(不是通过复杂的进展报告,但工作的软件)。 这确实把一些额外的负担在顾客定期与团队合作(而不是等到结束的合同,希望交付将承诺)。 但这频繁接触确保项目走向正确的方向,而不是向建设一只青蛙鱼时的预期。

响应变化胜过遵循计划:这意味着接受变化是自然和回复他们不害怕他们。 总是很高兴事先有一个计划,但是没有很好的坚持一个计划,不管代价,即使情况发生了变化。 让说你写一个测试用例,这是你的计划,如果一定要求。 现在,如果需求变更,您不哀悼的浪费你的时间和精力。 相反,您及时调整您的测试用例来验证更改的要求。 当然,只有傻瓜才会尝试新软件上运行相同的测试用例和测试标记为失败。

* **Responding to change over following a plan:**This means accepting changes as being natural and responding to them without being afraid of them. It is always nice to have a plan beforehand but it is not very nice to stick to a plan, at whatever the cost, even when situations have changed. Lets say you write a test case, which is your plan, assuming a certain requirement. Now, if the requirement changes, you do not lament over the wastage of your time and effort. Instead, you promptly adjust your test case to validate the changed requirement. And, of course, only a FOOL would try to run the same old test case on the new software and mark the test as FAIL.

**PRINCIPLES BEHIND AGILE MANIFESTO** 敏捷宣言背后的原则

Behind the Agile Manifesto are the following principles which some agile practitioners unfortunately fail to understand or implement. We urge you to go through each principle and digest them thoroughly if you intend to embrace Agile Testing. On the right column, the original principles have been re-written specifically for software testers.

在敏捷宣言背后，是一些敏捷实践者不幸未能理解或实现的原则。如果您打算接受敏捷测试，我们鼓励您仔细检查每一个原则并彻底地消化它们。在右边的列中，最初的原则被专门为软件测试人员重新编写。

|  |  |
| --- | --- |
| **We follow these principles:**  我们遵循这些原则 | **What it means for Software Testers:**  对于软件测试人员来说意味着什么 |
| Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.  我们的首要任务是通过早期和持续交付有价值的软件来满足客户。 | Our highest priority is to satisfy the customer through early and continuous delivery of high-quality software.  我们的首要任务是通过早期和持续交付高质量的软件来满足客户。 |
| Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.  欢迎改变需求，即使是在开发后期。敏捷过程利用变更来满足客户的竞争优势。 | Welcome changing requirements, even late in testing. Agile processes harness change for the customer’s competitive advantage.  欢迎改变需求，即使是在测试后期。敏捷过程利用变更来满足客户的竞争优势。 |
| Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.  经常交付工作软件，从几个星期到几个月，优先考虑时间较短的时间。 | Deliver high-quality software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.  频繁地交付高质量的软件，从几个星期到几个月，优先考虑时间更短的时间。 |
| Business people and developers must work together daily throughout the project.  业务人员和开发人员必须在整个项目中每天工作。 | Business people, developers, and testers must work together daily throughout the project.  业务人员、开发人员和测试人员必须在整个项目中每天工作。 |
| Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.  围绕有动机的个人构建项目。给他们提供他们需要的环境和支持，并相信他们能完成工作 | Build test projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.  围绕有动机的个人构建测试项目。给他们提供他们需要的环境和支持，并相信他们能完成工作。 |
| The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.  在开发团队中传递信息的最有效和有效的方法是面对面的交谈。 | The most efficient and effective method of conveying information to and within a test team is face-to-face conversation.  在测试团队中传递信息的最有效和有效的方法是面对面的交谈。 |
| Working software is the primary measure of progress.  工作软件是进步的主要衡量标准。 | Working high-quality software is the primary measure of progress.  工作高质量的软件是进步的主要衡量标准。 |
| Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.  敏捷过程促进可持续发展。发起人、开发人员和用户应该能够无限期地保持恒定的速度。 | Agile processes promote sustainable development and testing. The sponsors, developers, testers, and users should be able to maintain a constant pace indefinitely.  敏捷过程促进可持续发展和测试。发起人、开发人员、测试人员和用户应该能够始终保持恒定的速度。 |
| Continuous attention to technical excellence and good design enhances agility.  持续关注技术卓越和好的设计提高了敏捷性。 | Continuous attention to technical excellence and good test design enhances agility.  持续关注技术卓越和良好的测试设计可以提高敏捷性。 |
| Simplicity–the art of maximizing the amount of work not done–is essential.  简单——最大限度地发挥工作量的艺术——是必不可少的。 | Simplicity–the art of maximizing the amount of work not done–is essential.  简单——最大化工作量的艺术——是必不可少的 |
| The best architectures, requirements, and designs emerge from self-organizing teams.  最好的架构、需求和设计来自自组织的团队。 | The best architectures, requirements, and designs emerge from self-organizing teams.  最好的架构、需求和设计来自自组织的团队。 |
| At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.  每隔一段时间，团队就会反思如何变得更有效率，然后相应地调整自己的行为。 | At regular intervals, the test team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. |

**THE RECURRING QUESTION 一直存在的问题**

So what happens to all the traditional software testing methods, types and artifacts? Do we throw them away?

那么，所有传统的软件测试方法、类型和工件会怎样呢?我们要扔掉它们吗?

**THE ANSWER**

Naaah! You will still need all those software testing methods, types and artifacts (but at varying degrees of priority and necessity). You will, however, need to completely throw away that traditional attitude and embrace the agile attitude.

Naaah !您仍然需要所有这些软件测试方法、类型和工件(但是在不同程度的优先级和必要性)。然而，你需要完全抛弃传统的态度，接受敏捷的态度。

[Test Script](http://softwaretestingfundamentals.com/test-script/)

[Test Case](http://softwaretestingfundamentals.com/test-case/)