# <http://www.cnblogs.com/taven/archive/2010/10/17/1853386.html>

<http://blog.csdn.net/alvanchen/article/details/5749872/>

# 小米唐沐：传统制造业能否跟上“小步快跑，快速迭代”节奏

# 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.

**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.



**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/)