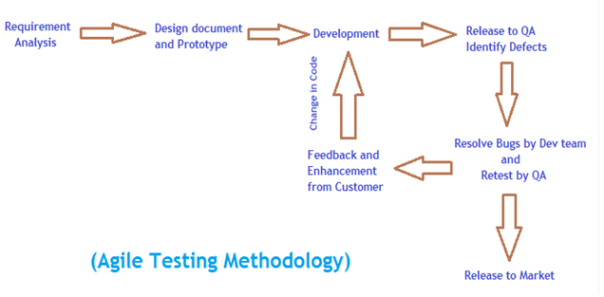
**Introduction:**AGILE is a methodology that enables continuous iteration of development and testing throughout the software development life cycle of the project. Iteration is defined as a small release of software. Agile Testing starts at the beginning of the project with rapid integration between development and testing.

We can define the term “Agile” as “moving quickly and easily”. In agile testing, the testers are closely working with the development team and testing is done in parallel as and when a piece of code has been developed.  Daily team meetings and discussions is an essential part of agile projects.

Agile testing is a Team effort. Agile team can achieve Quality and success by working as a single team to fulfill the common objective. There is no term called “My work”, “His Work”, “Your work”, “I am done with my work”. In an Agile team, we can find only the terms like “Our work”, “We have completed our Work”.

**How Agile model Helps:**1. It helps us by saving of time and money.  
2. Our highest priority is to satisfy the customer by rapid, continuous delivery of useful software.  
3. Customers, developers and testers interact with each other time to time.  
4. Working Build is delivered in a weekly or monthly basis as and when required.  
5. We can discuss the changes with our customers and add/remove new features based on feedback.  
6. We can eliminate most of the defects and crashes at the initial stage by resolving the top most reported bugs.

[](http://intensetesting.files.wordpress.com/2014/03/agile-testing-methodology.png)

Agile Testing Methodology

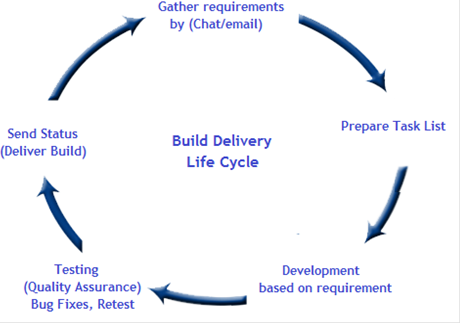
**Types of Agile Methods:**

**Scrum:**Scrum is an Agile process, where we can divide our projects into small components to be developed and then to be tested in specific time-period called as sprint (small cycles). Each feature should get developed and tested in a specified small time-slice.

**Extreme Programming (XP)**  
XP (Extreme Programming) is an agile methodology. It can be applied when we may have a system whose functionality is expected to change every few weeks/months. Sometimes, Our customers may not have a concrete idea of what the system should exactly do. In many software industries, the requirements are dynamically changing and this change is the only constant thing. This is when XP will work while other methodologies do not work.

The main goal of XP is to deliver useful software to the customer as and when it is required. Here, we have to set expectations so that in a limited period of time, customers can receive a new build of the system with the most prioritized features. Then we can make new plans for the next release. We will finish as much of working software as we can and measure our progress accordingly.

**DSDM (Dynamic System Development Method)**DSDM is based upon continuous development and Frequent Delivery. Goal of DSDM is to deliver a working software quickly, with more functionality which can be delivered at regular intervals. As the users are actively involved in the development of the system, they are more likely to accept the system. Development results are directly visible to the customer. So, there is an early indicator of whether the project will work or not. There is no chance of a nasty surprise at the end of development. Because of regular feedback from the customer, the developed system is more likely to meet the needed requirements. System is being delivered on time and on budget.

[](http://intensetesting.files.wordpress.com/2014/03/delivery-life-cycle.png)

Delivery Life Cycle

**FDD (Feature Driven Development):**This process is mostly used in larger teams working on a project using object-oriented technology. Here, as a team, we can spend a small time period at the beginning of the project to have a clear understanding of the domain in which we are working and use that understanding to create a rough plan without getting stuck in analysis and design phase.

We can create an overall development plan and a comprehensive Feature list. These can be broken into minor and major Feature sets. The development plan includes the order in which feature sets will be worked on, who will be responsible for which feature sets etc.. Then the team can start designing, building and testing the features within a particular time frame.

**Role of Agile Testers:**Agile testers should have the primary ability to learn new things and Be Adaptive to changes. We can expand our skill set like understanding of Business or domain knowledge and increase our technical skills while working in an agile team. We should have a good experience in exploratory testing. We can help our development team or customers by looking the features from various perspectives and make them aware of any type of problems or issues that may arise. Instead of creating comprehensive test documentation, we can create some checklists or feature lists which can be reused throughout the Project life cycle

As an agile tester we should concentrate our testing effort in some specific areas where there are chances to get most of the defects. This is the best guideline to achieve proper testing coverage within limited time period, resources and budget.

**Conclusion:**We cannot stick to any given methodology, because the needs and conditions of the company and project may change regularly, and we need to be flexible in how to approach managing projects if we want them to be successful. A single methodology will not work all the time, so the best way is to determine which methods work at that time and adopt that methodology to suit our individual needs. This is what being “Agile” is fundamentally about. A good agile team should choose the technical practices that can best work for them.

**What’s an Agile Tester?**

We define an agile tester this way: a professional tester who embraces kartichange, collaborates well with both technical and business people, and understands the concept of using tests to document requirements and drive development. Agile testers tend to have good technical skills, know how to collaborate with others to automate tests, and are also experienced exploratory testers. They’re willing to learn what customers do so that they can better understand the customers’ software requirements.

Who’s an agile tester? She’s a team member who drives agile testing. We know many agile testers who started out in some other specialization. A developer becomes test-infected and branches out beyond unit testing. An exploratory tester, accustomed to working in an agile manner, is attracted to the idea of an agile team. Professionals in other roles, such as business or functional analysts, might share the same traits and do much of the same work.

Skills are important, but attitude counts more. Janet likes to say, “Without the attitude, the skill is nothing.” Having had to hire numerous testers for our agile teams, we've put a lot of thought into this and discussed it with others in the agile community. Testers tend to see the big picture. They look at the application more from a user or customer point of view, which means they’re generally customer-focused.

**The Agile Testing Mind-Set**

What makes a team “agile”? To us, an agile team is one that continually focuses on doing its best work and delivering the best possible product. In our experience, this involves a ton of discipline, learning, time, experimentation, and working together. It’s not for everyone, but it’s ideal for those of us who like the team dynamic and focus on continual improvement.

Successful projects are a result of good people allowed to do good work. The characteristics that make someone succeed as a tester on an agile team are probably the same characteristics that make a highly valued tester on any team.

An agile tester doesn’t see herself as a quality police officer, protecting her customers from inadequate code. She’s ready to gather and share information, to work with the customer or product owner in order to help them express their requirements adequately so that they can get the features they need, and to provide feedback on project progress to everyone.

Agile testers, and maybe any tester with the right skills and mind-set, are continually looking for ways the team can do a better job of producing high-quality software. On a personal level, that might mean attending local user group meetings or roundtables to find out what other teams are doing. It also means trying out new tools to help the team do a better job of specifying, executing, and automating customer requirements as tests.

The bottom line is that agile testers, like their agile teammates, enjoy learning new skills and taking on new challenges, and they don’t limit themselves to solving only testing issues. This isn’t just a trait of testers; we see it in all agile team members. Agile testers help the developer and customer teams address any kind of issue that might arise. Testers can provide information that helps the team look back and learn what’s working and what isn’t.

Creativity, openness to ideas, willingness to take on any task or role, focus on the customer, and a constant view of the big picture are just some components of the agile testing mind-set. Good testers have an instinct and understanding for where and how software might fail, and how to track down failures.

Testers might have special expertise and experience in testing, but a good agile tester isn’t afraid to jump into a design discussion with suggestions that will help testability or create a more elegant solution. An agile testing mind-set is one that is results-oriented, craftsman-like, collaborative, eager to learn, and passionate about delivering business value in a timely manner.

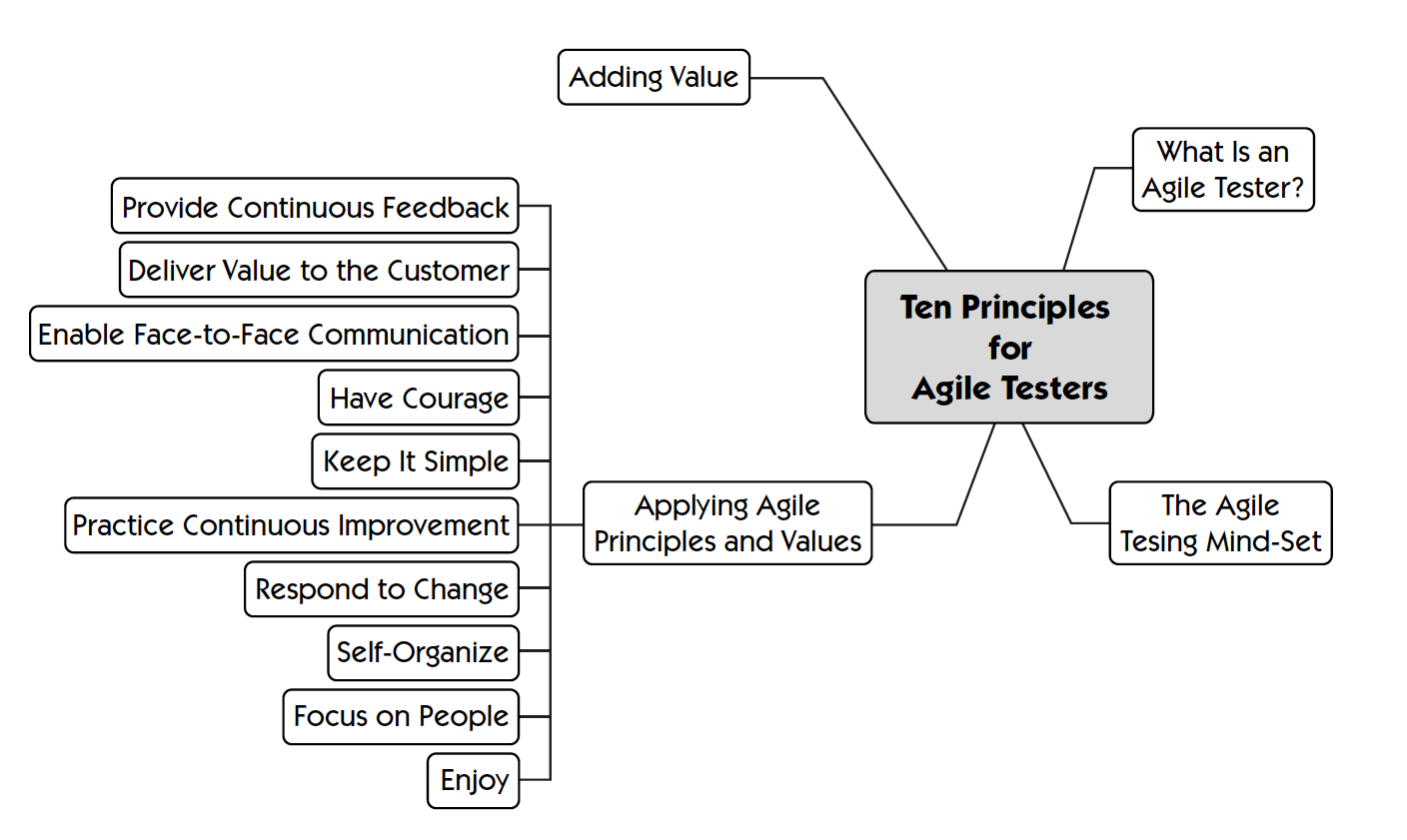
**Applying Agile Principles and Values**

Individuals can have a big impact on a project’s success. We’d expect a team with more experienced and higher-skilled members to outperform a less talented team. But a team is more than just its individual members. Agile values and principles promote a focus on the people involved in a project and how they interact and communicate. A team that guides itself with agile values and principles will have higher team morale and better velocity than a poorly functioning team of talented individuals.

The four value statements in the Agile Manifesto, which we presented at the start of the first chapter, show preferences, not ultimatums, and make no statements about what to do or not to do. The Agile Manifesto also includes a list of principles that define how we approach software development. Our list of agile “testing” principles is partially derived from those principles. Because we both come from the Extreme Programming culture, we’ve adopted many of its values and underlying principles. We’ve also incorporated guidelines and principles that have worked for our teams. Your team’s own values and principles will guide you as you choose practices and make decisions about how you want to work.

The principles we think are important for an agile tester are

1. Provide continuous feedback.
2. Deliver value to the customer.
3. Enable face-to-face communication.
4. Have courage.
5. Keep it simple.
6. Practice continuous improvement.
7. Respond to change.
8. Self-organize.
9. Focus on people.
10. Enjoy.

[](http://agile.dzone.com/sites/all/files/TenPrinciples_AgileTesters.png)