Vskills CERTIFIED

www.vskills.com

Scrum Methodology Sample Material VS-1422

1. AGILE AND SCRUM BASICS

The most commonly used software development model was waterfall model. However, in most of the cases, new functionalities get added, and also earlier requirements may change. The Waterfall model is not structured to accommodate such continuous changes in requirements. Further, the user will not have clarity on the functionality of the product till the product becomes available in its entirety.

In the iterative incremental model, the development starts with a limited number of finalized and prioritized requirements. The deliverable is a working increment of the product. A set of activities ranging from requirements to code development is called an iteration. Based on the functionality of the increment and any or all of the new, modified, pending requirements, the next lot of requirements is given to the subsequent iteration. The outcome of the subsequent iteration is an enhanced working increment of the product. This is repeated till the product accomplishes the required functionalities.

The user is usually not involved in the development work and it may cause communication gaps resulting in incorrect functionalities. The involvement is positive for the development team, but is demanding on the time of the team and can add delays. Further, any informal requirement changes during an iteration may lead to confusion and may also create scope creeps. With this premise, Agile development came into existence.

"Agile Development" is an umbrella term for several iterative and incremental software development methodologies. The most popular agile methodologies include Extreme Programming (XP), Scrum, Crystal, Dynamic Systems Development Method (DSDM), Lean Development, and Feature-Driven Development (FDD).

While each of the agile methodologies is unique in its specific approach, they all share a common vision and core values (see the Agile Manifesto). They all fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system. They all involve continuous planning, continuous testing, continuous integration, and other forms of continuous evolution of both the project and the software. They are all lightweight, especially compared to traditional waterfall-style processes, and inherently adaptable. What is more important about agile methods is that they all focus on empowering people to collaborate and make decisions together quickly and effectively.

1.1. Agile Basics

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.
agile in a nutshell

It works by breaking projects down into little bits of user functionality called user stories, prioritizing them, and then continuously delivering them in short two week cycles called iterations.
agile in a nutshell

To be agile merely means to be quick. The definition of the word is rather easy to understand; however, quick is a comparative term used to describe the "degree of comparison between similar adjectives" (i.e., good or better). In other words, the word "quick" is appraised by comparing it to other adjectives with similar meaning with an end result that has the potential to be highly subjective. The assessment of the word "quick" is relative and is based on individualized perceptions. The best way to describe "quick" as it pertains to agile project management is "quicker than traditional project management methods." Agility refers to the capability to think and reach conclusions quickly.

Agile development is based on iterative incremental development, in which requirements and solutions evolve through team collaboration. It recommends a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a theoretical framework and does not specify any particular practice that a development team should follow. Scrum is a specific agile process framework that defines the practices required to be followed.

Early implementations of agile methods include Rational Unified Process (1994), Scrum (1995), Crystal Clear, Extreme Programming (1996), Adaptive Software Development, Feature Driven Development (1997), and Dynamic Systems Development Method (DSDM) (1995). These are now collectively referred to as agile methodologies, after the Agile Manifesto was published in 2001.

1.2. Agile Manifesto

The Agile Manifesto was published by a team of software developers in 2001, highlighting the importance that needs to be given to the development team, accommodating changing requirements, customer involvement.

The Agile Manifesto is as follows:

"We are uncovering better ways of developing 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."
...Manifesto for Agile Software Development, Authors: Beck, Kent, et al. (2001)

Manifesto Item	Description
Individuals and interactions	Importance needs to be given to: self-organization and self-motivation of the team members continuous interaction for work, clarifications, information among the team members
Working Software	Delivery of working software at short duration intervals helps gain customer

Manifesto Item	Description
	trust and assurance in the team.
Customer collaboration	Constant involvement of customer with the development team ensures communication of necessary modifications.
Responding to change	Focus on quick response to the proposed changes, which is made possible with short duration iterations.

1.3. Agile Principles

As part of the Agile Manifesto and in addition to the four agile values, agile practitioners follow 12 guiding principles. These values and principles are important in gaining the correct understanding of agile project management.

The agile guiding principles are as

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software or project.

The goal is to satisfy the customer by the continuous delivery of valuable software. Valuable software refers to software that is valued by the customer. The term "value" is a very important word and it represents the main focus of the agile framework. The understanding that must be obtained from this guiding principle is that the agile framework has a customer and a value- based focus.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

This guiding principle demonstrates the agile framework's flexibility and adaptability when dealing with change. Traditional project management is generally managed around a change management process and is sometimes viewed as being costly and enabling scope creep (i.e., small changes in a plan or project that necessitate other changes which lead to still more changes, etc.). In contrast, the agile process is more accepting of changes at any time, even very late in development. There is no such thing as scope creep in the agile realm because changes are always accepted. If, however, the change does not add value, then there is the possibility that it won't be included in the product. It is important to understand that the focus on the agile project is mainly about value- added changes. The logic behind the acceptance of change is based on several factors

- ✓ Changes are considered to be the norm on agile projects.
- ✓ Flexibility and adaptability enable and support a customer's competitive advantage by welcoming changes that add value at any time.
- ✓ Accepting change is faster than approving or denying changes.

Remember that agility is all about speed! Once again, readers need to understand that only those changes that "add value" are candidates for inclusion into the product. The agile team must fully understand that they should limit change discussions and actions to value- added propositions only.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.

Early feedback is decidedly much better than proceeding on a project and getting feedback late. What is worse is that the project could end up heading in the wrong direction. Not only is going down the wrong path quite costly, it is also a great waste of time and effort. An iteration on agile projects should be between two weeks to one month, again with the preference always being the shorter time frame. Why do we want a shorter time frame? There are a couple of reasons

- ✓ Short delivery cycles result in regular feedback from the project stakeholders. This keeps the project from losing momentum by keeping everyone actively engaged. On a traditional project, slow feedback can result in a lack of engagement by the project team and has the potential to contribute to project delays.
- ✓ Frequent delivery of working software is beneficial in that requirements can be quickly added or modified. Here again, speed is very relevant.
- ✓ How do frequent small changes increase the amount of value-added for the customer? Small and frequent changes can minimize the likelihood that the customer will have the need to create a large number of change requests. This practice ensures that the project is satisfying the customer's requirements on a regular and consistent basis.
- ✓ The goal of the agile project is to deliver value throughout the project. Accepting changes is one practice that supports the goal of consistently adding value to the product.

Business people and developers must work together daily throughout the project.

In order to deliver a project quickly, face- to- face interactions is the fastest way to communicate on agile projects. E- mails, phone calls, and documentation are considered to be less efficient and slower methods of communication. Daily face- to- face interactions between customers and developers enable a faster rate of transferring knowledge. This results in all project stakeholders being on the same page with no surprises. Product delivery on the agile project has a higher rate of success in meeting or exceeding customer expectations based on these interactions.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Motivated and talented people make a big impact on an agile project. The successful delivery of the product depends on empowered team members. Agile methods are based on self- directed and self- organized teams who can be trusted to get the job done. There is no micromanagement on agile projects. The management style or lack thereof is based on team collaboration. This mind-set results in the project getting completed faster and effectively. The best team members on an agile project are those who can work with little supervision and are self- motivated.

The most efficient and effective method of conveying information to and within a development team is face- to- face conversation.

Such conversations are the fastest way to communicate on agile projects. The flow of communication is more effective and efficient for face to face in comparison with other methods. Inquiries and inconsistencies can be addressed very quickly. This results in minimal delays and faster delivery of the product. Small team sizes make for ease in communicating face to face. In the case of larger project teams, face to face can be challenging, however, the method of communication can be tailored to meet the needs of the project.

Working software is the primary measure of progress. Progress on the agile project is determined by how well the software works.

This is a results- based focus that cannot be easily disputed. Working software shows the customer results that can be approved and accepted. It also shows progress made toward the end goal of product delivery. When the software works, only then can it be approved by the customer as being completed.

Agile processes promote sustainable development.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely. Agile methods support the project team's need to have a life outside of work. This means that the expectation for the team is to maintain a sustainable pace based on a typical 40-hour workweek. Long work-days are not looked upon favorably on agile projects. By working at a sustainable pace, the team can be more productive, which results in less project tension. The sustainable pace is a win– win situation for everyone involved in the project. This in turn is beneficial at the corporate level in that companies do not want overworked teams who are stressed out, burnt out, and unhappy.

Continuous attention to technical excellence and good design enhance agility.

In order to deliver high value to the end client, it is often necessary for the development team to make changes to the design. This means that the design must be relatively easy to maintain. Technical excellence and a good design make for ease in understanding and making changes to the design. This in turn supports the ability to respond to change very quickly. A good product design and technical excellence enhance agile methods because continuous attention is given to the design of the software. The value of this guiding principle is having a design that is easily maintainable based on its technical excellence.

Simplicity—the art of maximizing the amount of work not done—is essential.

The author believes that this guiding principle is the trickiest one to understand. At face value, the principle talks about work not done. The question then becomes: "Why is there a concern about work not done?" From the agile perspective, work not done is more reliable because there is nothing that could go wrong with it. The development team does not code work that is not done so it's perfect because it has not been touched. It is believed that over 60% of software features are included in a product but are rarely if ever used. * This is the reason why agile methods focus on simplicity. By focusing only on the necessary components of the product, there is less risk. Too much complexity increases project risk. The takeaway of this guiding principle is that simplicity is better than complexity. The development team should only build what is required by the customer. Gold plating (the incorporation of costly and unnecessary features or refinements into a product) is a big no- no. By focusing on simplicity, this in turn speeds up the actual delivery of the product.

The best architectures, requirements, and design emerge from self-organizing teams.

According to agile methods, when people are given the chance to self- manage themselves, they produce better work. This includes the best architectures, requirements, and designs. The best work is developed by those who are the originators of the work. Why is this so? Self- managing teams with the ability to make their own decisions take pride in ownership of their work. One reason for this is because these teams are free to work in the manner that suits them without unnecessary interference from others such as the customer and the business people. These teams don't have to sell their ideas to others and this saves a considerable amount of time. Simply put,

the agile team is responsible for the outcome of the product and they are the best ones to have the freedom to create it. For clarity, readers need to understand that we are speaking of the development team. This is not inclusive of the customer or the business side of the project.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

This guiding principle is simply about lessons learned and when they occur on the agile project. Agile methods support the idea that it is best not to wait until the end of a project to address lessons learned. Lessons learned need to be captured and addressed during the project. For example, in the case of Scrum, lessons learned should be addressed at the end of each Sprint (iteration). Agile methods capture lessons learned as the project ensues rather than at the end of the project. The agile project team then tunes and adjusts its behavior for subsequent iterations. This gives the project team a chance to act on the lessons learned rather than just discussing the actions and hoping for a chance to apply the lessons on future projects.

The Agile Manifesto is based on the following principles:

Principle	Description
Satisfaction and Delivery	Customer satisfaction through early and continuous working software.
Welcoming Change	Welcome changing requirements, even at later stages of development.
Deliver Frequently	Deliver working software frequently (weekly rather than monthly).
Communication is the Key	Ensure close association of developers with business people on daily basis.
Environment and Trust	Build projects around motivated individuals. Give them necessary support and trust them.
Face-to-face Communication	Encourage face-to-face conversation to ensure efficient and effective communication.
Software as Measure of Progress	Working software is the primary measure of progress.
Sustainable Development	Promote sustainable development with the ability to maintain a constant pace throughout the development.
Attention to Details	Continuous attention to technical excellence and good design.
The Power of Less	Simplicity is essential.
Self-organizing Teams	Regular attention of the team on becoming effective in changing circumstances.

DECLARATION OF INTERDEPENDENCE

The Declaration of Interdependence (DOI) was developed strictly for the project management side of the agile projects and it is not a part of the Agile Manifesto. As a matter of fact, the DOI was created in 2003, two years after the Agile Manifesto. There are six principles associated with the DOI. According to Griffiths (2012), the DOI is directed at project leader ship and is not tested on the PMI- ACP exam. * It is important to mention that the difference between the DOI and the Agile Manifesto is that the DOI is for the project management side of an agile project and the latter guides the entire agile project.

The six principles associated with the DOI, are

- ✓ We increase return on investment by making the continuous flow of value our focus. This principle means that the agile project provides exactly what the business has asked for and nothing more. The logic behind this principle is that when results are provided that meet the needs of the business, then this makes a case for project continuation and approval. The continuous flow of value means that the project delivers the desired business results instead of simply delivering results that may be useless to the customer.
- ✓ We deliver reliable results by engaging customers in frequent interactions and shared ownership. Not only is engaging the customer good business, it helps develop and build relationships. Frequent interactions with the customer support the delivery of a product that meets or exceeds expectations. When the customer is engaged and sharing ownership of the project, the results are undoubtedly reliable.
- ✓ We expect uncertainty and manage for it through iterations, anticipation, and adaptation. Agile methods do not rely on plans to guide projects. Change is the expectation for agile methods and it is managed through iterations, anticipation, and adaptation. Software projects are known for constant changes and a good way to manage modifications is to develop software in iterations. The fact that agile methods are highly adaptable means that change is anticipated at any time during the project. The agile project adapts to change for value- added reasons and very little up-front planning is needed.
- ✓ We unleash creativity and innovation by recognizing that individuals are the ultimate sources of value, and creating an environment where they can make a difference. Agile methods recognize the value of individuals on the team. The logic behind this principle is that each individual team member must be treated well; his or her needs should be satisfied and each should get the support needed to be successful on the agile project. In addition, the agile project environment should be the best possible one that can be provided.
- ✓ We boost performance through group accountability for results and shared responsibility for team effectiveness. Shared responsibility for project success is the idea behind this principle. Self- managed teams are typically more satisfied and enthusiastic when working together to resolve issues. Team empowerment not only results in ownership of project glitches, it encourages team members to work very hard for resolutions. When the entire team shares the responsibility for project success, everyone on the team is a success.
- ✓ We improve effectiveness and reliability through situational specific strategies, processes, and practices. It should be well understand that agile methods are flexible and projects are unique. This means that project conditions must be evaluated for the best solutions instead of attempting a one- size- fits- all approach. Based on what is available for the project, agile methods require adaptation for the best solution based on the environment and the circumstances that are present.

Certifications

Accounting, Banking and Finance
- Certified AML-KYC Compliance Officer
- Certified Business Accountant
- Certified Commercial Banker
- Certified Foreign Exchange Professional
- Certified GAAP Accounting Standards Professional
- Certified Financial Risk Management Professional
- Certified Merger and Acquisition Analyst
- Certified Tally 9.0 Professional
- Certified Treasury Market Professional
- Certified Wealth Manager

Big Data
- Certified Hadoop and Mapreduce Professional

Cloud Computing
- Certified Cloud Computing Professional

Design
- Certified Interior Designer

Digital Media
- Certified Social Media Marketing Professional
- Certified Inbound Marketing Professional
- Certified Digital Marketing Master

Foreign Trade
- Certified Export Import (Foreign Trade) Professional

Health, Nutrition and Well Being
- Certified Fitness Instructor

Hospitality
- Certified Restaurant Team Member (Hospitality)

Human Resources
- Certified HR Compensation Manager
- Certified HR Stafffing Manager
- Certified Human Resources Manager
- Certified Performance Appraisal Manager

Office Skills
- Certified Data Entry Operator
- Certified Office Administrator

Project Management
- Certified Project Management Professional

Real Estate
- Certified Real Estate Consultant

Marketing
- Certified Marketing Manager

Quality
- Certified Six Sigma Green Belt Professional
- Certified Six Sigma Black Belt Professional
- Certified TQM Professional

Logistics & Supply Chain Management
- Certified International Logistics Professional
- Certified Logistics & SCM Professional
- Certified Purchase Manager
- Certified Supply Chain Management Professional

Legal
- Certified IPR & Legal Manager
- Certified Labour Law Analyst
- Certified Business Law Analyst
- Certified Corporate Law Analyst

Information Technology
- Certified ASP.NET Programmer
- Certified Basic Network Support Professional
- Certified Business Intelligence Professional
- Certified Core Java Developer
- Certified E-commerce Professional
- Certified IT Support Professional
- Certified PHP Professional
- Certified Selenium Professional
- Certified SEO Professional
- Certified Software Quality Assurance Professional

Mobile Application Development
- Certified Android Apps Developer
- Certified iPhone Apps Developer

Security
- Certified Ethical Hacking and Security Professional
- Certified Network Security Professional

Management
- Certified Corporate Goverance Professional
- Certified Corporate Social Responsibility Professional

Life Skills
- Certified Business Communication Specialist
- Certified Public Relations Officer

Media
- Certified Advertising Manager
- Certified Advertising Sales Professional

Sales, BPO
- Certified Sales Manager
- Certified Telesales Executive

& many more job related certifications

Contact us at :

Vskills

011-473 44 723 or **info@vskills.in**

www.vskills.com