Agile Process and methodology

Prepared By:
Md: Shamim Hossain
ID: 171442594, Batch: 44th
City University of Bangladesh
Prepared for:
Rechard Philip
Senior Lecturer
City University of Bangladesh

Contents

Methodology

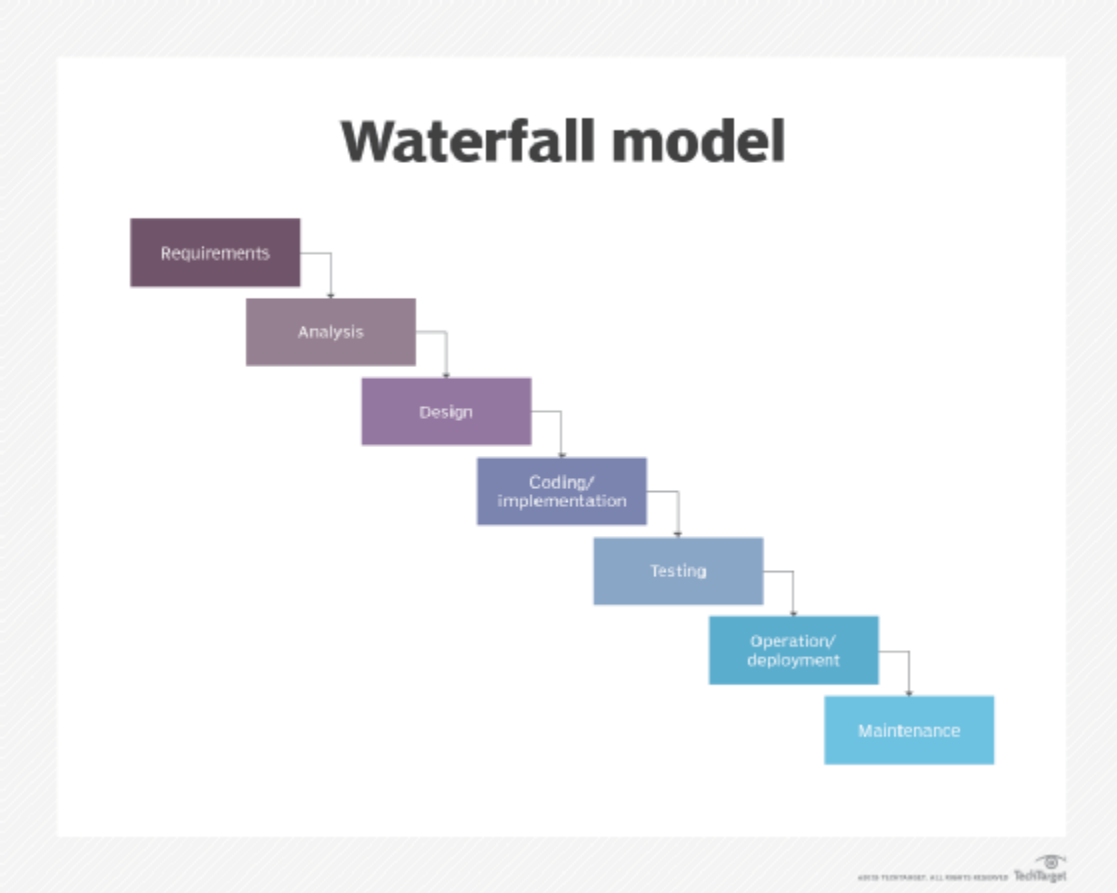
Agile methodology has taken the software development world by storm and rapidly cemented its place as "the gold standard." Agile methodologies all started based on four core principles as outlined in the Agile Manifesto. These methodologies are rooted in adaptive planning, early delivery and continuous improvement, all with an eye toward being able to respond to change quickly and easily. As a result, it's no surprise that 88% of respondents in Version One's 2017 State of Agile Report ranked "ability to adapt to change" as the number one benefit of embracing agile.

However, as more and more development teams adopt an agile philosophy, testers have struggled to keep pace. That's because the widespread adoption of Agile has led teams to issue releases and totally undocumented software on a more frequent

basis. This frequency has forced testers to shift when they conduct testing, how they work with developers and BAs and even what tests they conduct, all while maintaining quality standards.

Water Fall Development Methodology

1. to improve, update and enhance the final product. This could include releasing patch updates or releasing new versions.



Before moving to the next phase, there is usually a review and sign off process to ensure that all defined goals have been met.

The waterfall approach is ideal for projects that have specific documentation, fixed requirements, ample resources, an established timeline and well-understood technology. Alternatives to the waterfall model include joint application

development (JAD), rapid application development (RAD), sync-and-stabilize, Agile project management (APM) and the spiral model.

Advantages of the waterfall model

While agile or dynamic methods often replace the waterfall model, there are some advantages:

- Upfront documentation and planning stages allow for large or shifting teams to remain informed and move towards a common goal.
- Forces structured , disciplined organization.
- Is simple to understand, follow and arrange tasks.
- Facilitates departmentalization and managerial control based on schedule or deadlines.
- Reinforces good coding habits to define before design and then code.
- Allows for early design or specification changes to be made easily.
- Clearly defines milestones and deadlines.

Disadvantages of the waterfall model

The disadvantages of the waterfall model typically surround risk associated with a lack of revision, including:

- Design is not adaptive; often when a flaw is found, the entire process needs to start over.
- Ignores the potential to receive mid-process user or client feedback and make changes based on results.
- Delays testing until the end of the development life cycle.
- Does not consider error correction.
- Does not handle requests for changes, scope adjustments or updates well.

- Reduces efficiency by not allowing processes to overlap.
- No working product is available until the later stages of the life cycle.
- Not ideal for complex, high risk, ongoing or object-oriented projects.

Agile Development Methods

Every software development organization today seems to practice the agile software development methodology, or a version of it. Or at least they believe they do. Whether you are new to application development or learned about software development decades ago using the waterfall software development methodology, today your work is at least influenced by the agile methodology.
But what exactly is agile methodology, and how should it be practiced in software development?

[Learn how your enterprise can excel in agile development. | Take your agile career to the next level: How to improve your scrum master skills. | Keep up with hot topics in programming with InfoWorld's App Dev Report newsletter.]
Agile was formally launched in 2001 when 17 technologists drafted the Agile Manifesto. They wrote four major principles for developing better software:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Devops deployment Methodology

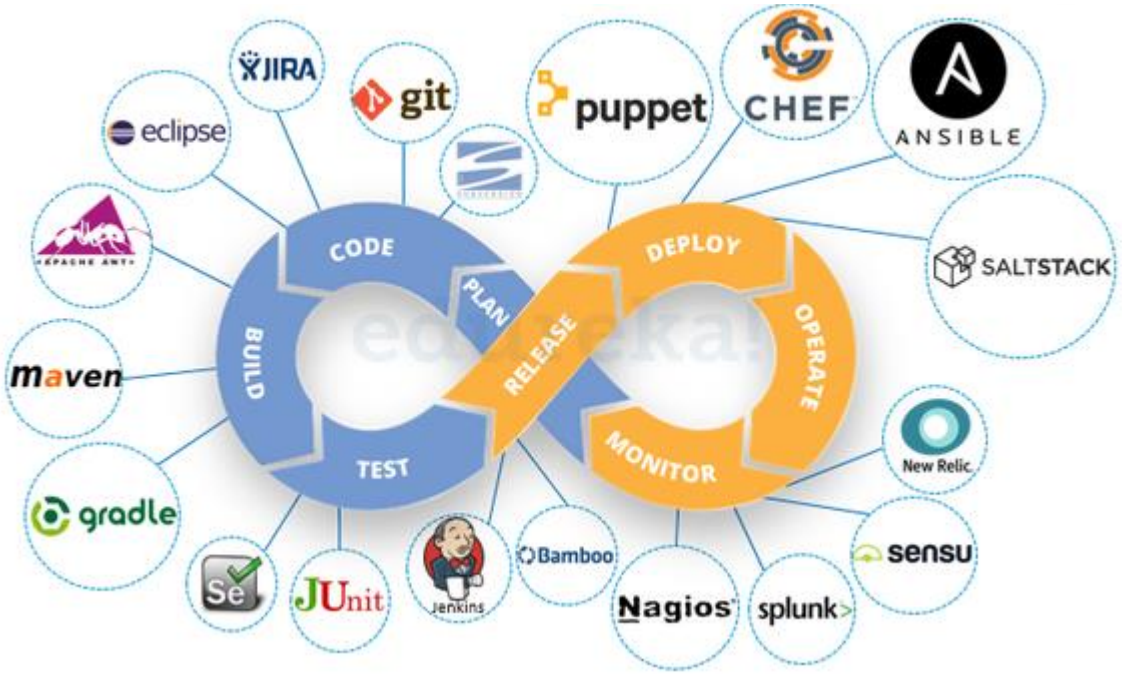
DevOps:
DevOps integrates developers and operations team to improve collaboration and productivity.

According to the DevOps culture, a single group of Engineers (developers, system admins, QA's. Testers etc turned into DevOps Engineers) has end to end responsibility of the Application

(Software) right from gathering the requirement to development, to testing, to infrastructure deployment, to application deployment and finally monitoring & gathering feedback from the end users, then again implementing the changes.
This is a never ending cycle and the logo of DevOps makes perfect sense to me. Just look at the above diagram – What could have been a better symbol than infinity to symbolize DevOps?
Now let us see how DevOps takes care of the challenges faced by Development and Operations. Below table describes how DevOps addresses Dev Challenges.

Going further, below table describes how DevOps addresses Ops Challenges.

However, you would still be wondering, how to implement DevOps. To expedite and actualize DevOps process apart from culturally accepting it, one also needs various DevOps tools like Puppet, Jenkins, GIT, Chef, Docker, Selenium, AWS etc to achieve automation at various stages which helps in achieving Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment, Continuous Monitoring to deliver a quality software to the customer at a very fast pace.
Now take a look at the below DevOps diagram with various DevOps Tools closely and try to decode it.



These tools has been categorized into various stages of DevOps. Hence it is important that I first tell you about DevOps stages and then talk more about DevOps Tools.
DevOps Lifecycle can be broadly broken down into the below DevOps Stages:

- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Virtualization and Containerization

These stages are the building blocks to achieve DevOps as a whole.

Basic introduction of Agile Software Development Methodology

When it comes to Software (SW) development, there are many frameworks and methodologies for smart management. Some of these frameworks are scrum, kanban, XP, TDD, FDD DDD etc.

Each of these frameworks and methodologies have their own pros and cons based on the nature of project they are applied to. Agile project management however is a relatively new methodology focusing on transparency, prioritization short feedback loop Inspect and adapt. The other known methodologies are the waterfall, prototyping, iterative and incremental development, spiral development, extreme programming etc.

Introduction to Agile

The common meaning of Agile is “able to move quickly and easily”. It is a programming methodology to do work in easy and convenient way. It is very useful to complete wider scope projects. It is a serialized process of delivering product to owner rather than to delivering it after completion.

2.1 What Agile Really is ?

Agile can be considered as SW development methodology, set of processes, principles or model to promote better planning, development evaluation, ontime delivery, continuous improvement and early response to change. Agile just provide a framework originally never set methods to achive this, but over the time many framework or change agents develop in the form on Scrum, XP, Kanban, Scrumban , etc.

Agile is a repetitive and additional method of managing the design and develop. Its aim is to offer the new services or products development in a highly flexible and common way.

2.2 Orgin of Agile and its Manifesto

Origin of Agile goes back to Agile Manifesto while Agile started as SW development process. However, now it is embraced in other industries like marketing, sales, support, banking and production. The known frameworks of the agile methodology are Scrum, Kanban, XP etc.

Agile manifesto describes 4 important values that are relevant in all the projects of any size and kind.Check our agile manifesto infographics on its 15th anniversary, which was on February 11th 2016.

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

2.3 Why Agile ?

- Identify difficulties, makes them visible
- Embrace the change
- Plan, but always keep in mind that detailed plans are always wrong
- Improve feedback loops
- In case of any failure, the impact will be very slight i.e. impact will be only on that specific bit of the project.

2.4 Agile History

- 1993: First Scrum team created by Jeff Sutherland
- 1995: Scrum formalized by Jeff Sutherland & Ken Schwaber

- 2001: First Scrum book by Ken Schwaber and Mike Beedle (Agile Software Development with SCRUM)
- Agile manifesto was Introduced by 17 software developers meeting at snowbird resort in Utah in Feb – 2001.

References:

Methodology: https://www.qasymphony.com/blog/agile-methodology-guide-agile-testing/
Waterfall development Methodology: //searchsoftwarequality.techtarget.com/definition/waterfall-model
Agile development Methodology: https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html
Devops deployment Methodology: //www.quora.com/What-is-DevOps-methodology
Introduction and definition of Agile Methodology: https://www.yodiz.com/blog/agile-software-development-methodology-definition-principle-of-agile/