Methodologies

SAAD1001

Momodou Lamin Keita
NSCC Semester 2

Table of Content

The Waterfall Methodology

- ***Brief History***

The first formal description of the waterfall method is often cited as a 1970 article by Winston W. Royce. Royce presented this model as an example of a flawed, non-working model and it has been widely used for software projects ever since. Before the advent of this method, the software development in computer companies suffered from a haphazard integrated software network like cluttered knitting. However with this method they hoped to bring clarity in their projects. The waterfall methodology is broken down into 6 main phases with deliverables on each stage.

Stages

1. Requirement Gathering: This stage involves a business analyst having a series of interactions with the client to collect requirements, all of these requirements are documented.
2. Requirement Analysis: System analyst(s) will go through the documented requirements from the client and prepare the system requirement specifications
3. Design: The architecture of the application is designed to fit the requirements of the clients.
4. Coding: Developers write the code using programming languages and/or scripting languages in order to develop the application.

5. Testing: Developers will perform unit testing and integration testing using white box testing. After white box testing is done a separate team will perform system testing using black box testing.
6. Release and Maintenance: When client is satisfied will all the testing and the finished product. The project is delivered to the customer to use.

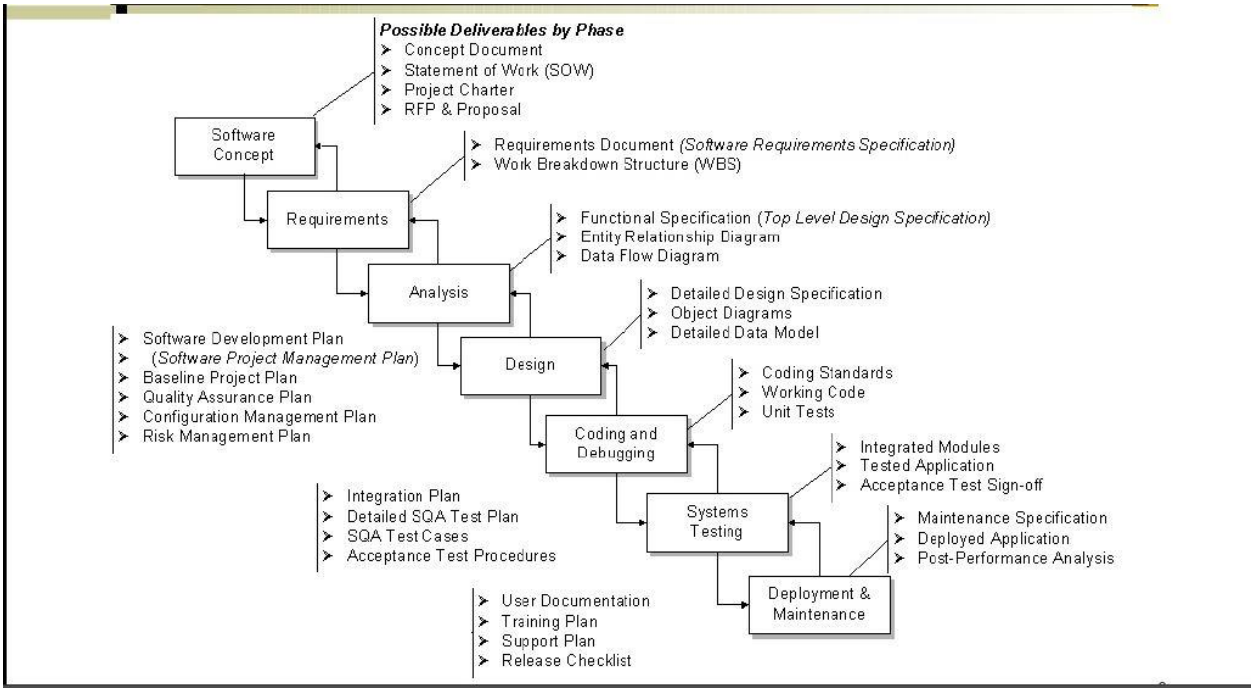


Fig1: Waterfall Methodology

- ***Context***

  The waterfall methodology can be implemented successfully when:

  1. Requirements are very well known, clear and fixed.
  2. Product definition is stable.
  3. Technology is understood.
  4. There are no ambiguous requirements.
  5. Ample resources with required expertise are available freely.

6. The project is short.

- ***Advantages/Disadvantages***

Advantages

- The waterfall model is simple and easy to understand and implement.
- Its helps in early error detection.
- Works well when quality is more important the cost or schedule.
- Documentation is produced at every phase of the model allowing people to understand what is been done.
- Testing is done at every stage
- Phases are completed one at a time

Disadvantages

- You cannot go back to previous phases in the model once the project has gone past it. If the design phase had something go wrong. Thing can get very complicated in the implementation phase.
- High amounts of risk and uncertainty.
- Only suitable for small projects
- Constant testing of the design is needed.
- If the requirements change the waterfall model may not work.
- Adjusting the scope during the life cycle can kill a project.

- ***Companies that use Waterfall Model***
  - Toyota
  - Svitla Systems

The Agile Methodologies

- ***Brief History***

Agile came about as a "solution" to the disadvantages of the waterfall methodology. Instead of a sequential design process, the agile methodology follows an incremental approach. Notable examples include the *Unified Software Development Process* (USDP) and *Extreme Programming* (XP). Agile methodologies have short life cycle iterations with very small to no documentation. The goal of agile methodologies is to deliver a working version of the software after every iterations. Each iteration involves a number of phases including planning, requirements analysis, implementation, and testing. The iterative approach helps reduce overall risk. There a various methodologies that are collectively known as agile. The most popular ones are: DSDM, Scrum and Extreme Programming (XP).

10 Key Principles of Agile

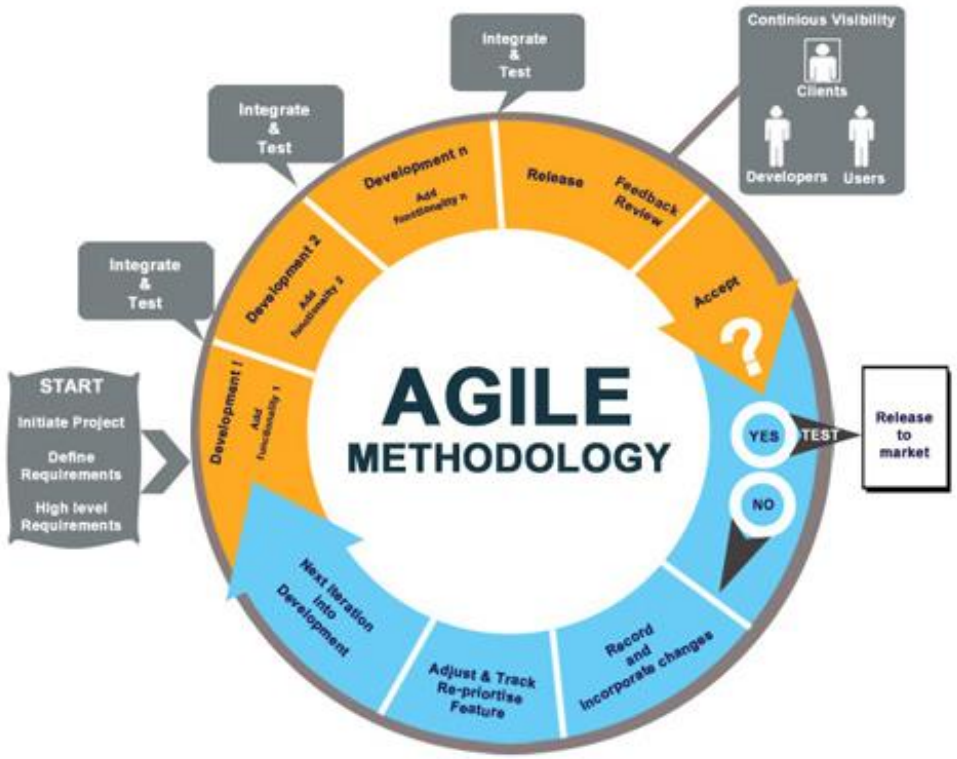


Fig 2: Agile Methodologies

- ***Context***
  - When there is a clear picture of what the final product should be
  - When clients won't be have the ability to change the scope of the project once it has begun
  - When definition, not speed, id key to success.

- ***Advantages/Disadvantages***

Advantages

- The Agile methodology allows for changes to be made after the initial planning. Re-writes to the program, as the client decides to make changes, are expected.

- Because the Agile methodology allows you to make changes, it's easier to add features that will keep you up to date with the latest developments in your industry.

- At the end of each sprint, project priorities are evaluated. This allows clients to add their feedback so that they ultimately get the product they desire.

- The testing at the end of each sprint ensures that the bugs are caught and taken care of in the development cycle. They won't be found at the end.

- Because the products are tested so thoroughly with Agile, the product could be launched at the end of any cycle. As a result, it's more likely to reach its launch date.

Disadvantages

- With a less successful project manager, the project can become a series of code sprints. If this happens, the project is likely to come in late and over budget.

- As the initial project doesn't have a definitive plan, the final product can be grossly different than what was initially intended.

- ***Companies that use Agile Model***
  - BP
  - General Electric

The Scrum Methodology

- ***Brief History***

Scrum is an agile framework for completing complex projects. Scrum originally was formalized for software development projects, buts it works well for any complex innovative scope of work. Scrum consist of three roles: The product owner, the Scrum master and the team. Scrum's incremental, iterative approach trades the traditional phases of "waterfall" development for the ability to develop a subset of high-value features first, incorporating feedback sooner.

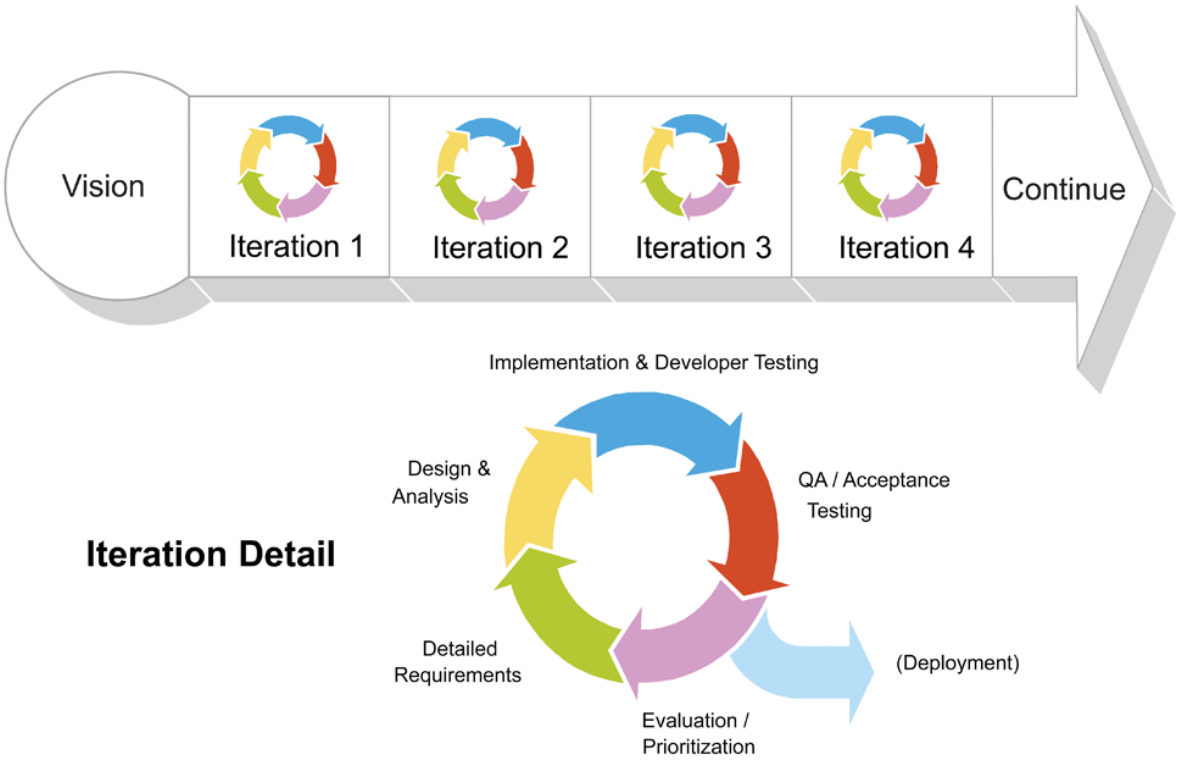


Fig 3: Scrum Methodology

- ***Context***

Scrum Roles

- ***Product Owner***

1. Single person responsible for maximizing the return on investment (ROI) of the development effort
2. Responsible for product vision
3. Constantly re-prioritizes the Product Backlog, adjusting any long term expectations such as release plans
4. Final arbiter of requirements questions
5. Accepts or rejects each product increment
6. Decides whether to ship
7. Decides whether to continue development
8. Considers stakeholder interests
9. May contribute as a team member

- ***Scrum Development Team***

1. Cross-functional (e.g., includes members with testing skills, and often others not traditionally called developers: business analysts, domain experts, etc.) Self-organizing / self-managing, without externally assigned roles
2. Negotiates commitments with the Product Owner, one Sprint at a time
3. Has autonomy regarding how to reach commitments

4. Intensely collaborative
5. Most successful when located in one team room, particularly for the first few Sprints
6. Most successful with long-term, full-time membership. Scrum moves work to a flexible learning team and avoids moving people or splitting them between teams.
7. 3-9 members (originally 7 ± 2 members)

- ***Scrum Master***

1. Facilitates the Scrum process
2. Helps resolve impediments
3. Creates an environment conducive to team self-organization
4. Captures empirical data to adjust forecasts
5. Shields the team from external interference and distractions to keep it in group flow (a.k.a. the zone)
6. Enforces time boxes
7. Keeps Scrum artifacts visible
8. Promotes improved engineering practices
9. Has no management authority over the team (anyone with authority over the team is by definition not its Scrum Master

- ***Advantages/Disadvantages***

Advantages

- good and *organized* involvement of customer and stake holders
- easier to stay within the 80%
- continuous feedback
- more flexible to changes
- team motivation
- very transparent when tracing progress
- makes it more obvious where delays come from (think unclear requirements)

Disadvantages

- Team needs to embrace the methodology. this needs a very open, communicative mindset
- easy to slack and let discipline go
- very high pressure on the product owner scrum role
- Does not scale well - for me a scrum team is at its best at about 7 people. Scrum of scrums are...awkward
- the peer pressure within the team can be immense
- usually established organisations have a hard time accepting scrum
- quality measures have to be explicitly built in

- ***Companies that use Waterfall Model***
    - Universities
    - Military
    - Automotive Companies

The Open-Source Methodology

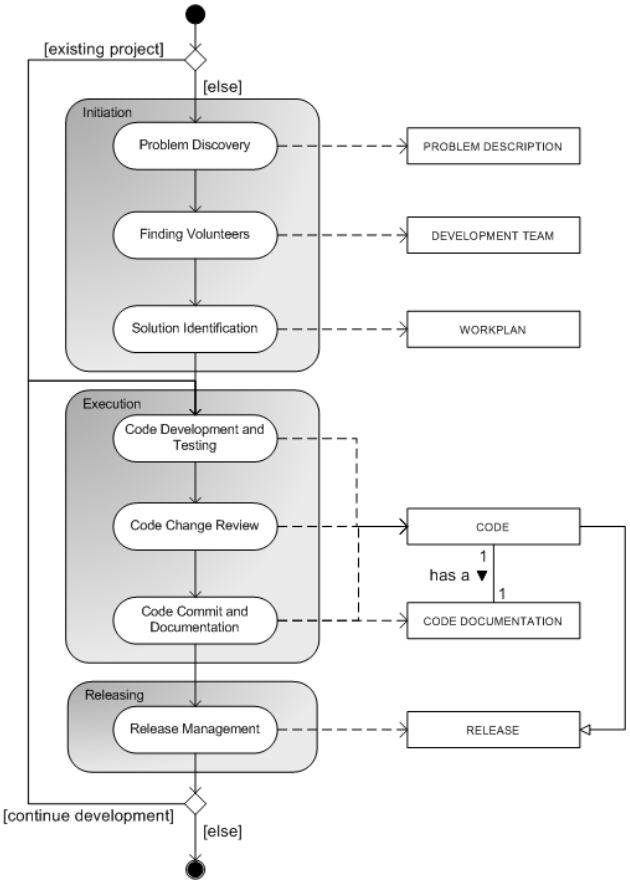
- ***Brief History***

Open source development is the process by which open-source software, or similar whose source code is publicly available is developed. All software produced under this methodology has its source code under an open-source license to study, change and improve its design. The open-source software development has been a large of the creation of the World Wide Web.

Fig1: Open Source Methodology

- ***Context***

The Open-source methodology can be divided into several stages



- ***Advantages/Disadvantages***

Advantages

- Open-source software is free to use, distribute, and modify.
- Lower cost
- Open-source software is more secured as the code is accessible to everyone
- Open source is not dependent on the company or author that originally created it.
- Open source standards are accessible to everyone; thus, it does not have the problem of incompatible formats that exist in propriety software

Disadvantages

- Not being straightforward to use.
- There are very few applications that run both open source and proprietary software. Switching to an open-source platform involves a compatibility analysis of all the other software that run on propriety platforms.
- Too many parallel development all on-going at the same time.
- Open source software might require third party driver.

- ***Companies that use Waterfall Model***
    - UNIX based Operating systems: (Ubuntu, Linux Mint, Debian etc.)

References

- "Software Development Methodologies-HSM, SSADM." Software Development Methodologies-HSM, SSADM. Web. 23 Apr. 2015. <http://www.slideshare.net/naksarpong/software-development-methodologyhsm-ssadm>
- "18-Methodologies." 18-Methodologies. Web. 23 Apr. 2015. <http://www.slideshare.net/Samuel90/18methodologies?related=1>
- "Software Design Methodologies." Software Design Methodologies. Web. 23 Apr. 2015. <http://www.slideshare.net/anniyappa/software-design-methodologies>.
- "Waterfall Model Ppt Final." Waterfall Model Ppt Final. Web. 23 Apr. 2015. http://www.slideshare.net/shivakrishnakontham/waterfall-model-ppt-final
- "Waterfall Model." Waterfall Model. Web. 23 Apr. 2015. <http://www.waterfall-model.com/>/
- "Agile & Waterfall Methodologies - A Side-By-Side Comparison." Base36. 13 Dec. 2012. Web. 23 Apr. 2015. <http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/>
- Web. 23 Apr. 2015. <http://splendidpixels.com/images/mobile902.jpg>.
- "Agile Methodology." Agile Methodology. Web. 23 Apr. 2015. <http://agilemethodology.org/>.
- "What Is Agile? (10 Key Principles of Agile)." All About Agile What Is Agile 10 Key Principles of Agile Comments. Web. 23 Apr. 2015. <http://www.allaboutagile.com/what-is-agile-10-key-principles/>.
- "Scrum Reference Card." Scrum Reference Card. Web. 23 Apr. 2015. <http://scrumreferencecard.com/scrum-reference-card/>.
- "Advantages and Disadvantages of the Scrum Project Management Methodology." Small Business. Web. 23 Apr. 2015. <http://smallbusiness.chron.com/advantages-disadvantages-scrum-project-management-methodology-36099.html>.
- "Learn About Scrum." What Is Scrum? An Agile Framework for Completing Complex Projects. Web. 23 Apr. 2015. <https://www.scrumalliance.org/why-scrum>.

- Web. 23 Apr. 2015. <http://www.quora.com/What-are-the-pros-and-cons-of-the-Waterfall-and-Agile-Scrum-project-management-approach>.
- Wikipedia. Wikimedia Foundation. Web. 23 Apr. 2015. <http://en.wikipedia.org/wiki/Open-source_software_development#Communication_channels>.