Software Development Unit 33, 1.1

Anna Legaspi

# I can investigate a range of development environments

There are a variety of environments available when developing software and this normally comes in stages. Before development tools and environments are set, the project would normally have gone through planning and requirement analysis. There needs to be a set of rules and requirements that is expected from the Software and setting this will help the developers in designing it.

## Environments

### Development

This is where software developers may use a variety of programming languages or coding to create software. For example, if the project is creating a website, the website developer will need to write the code for the website inline with the agreed rules, assumptions and requirements. Software examples for web developing software are Visual Studio, Dreamweaver CC, Notepad ++. Other software to create software and games include Apache NetBeans, Visual Studio code, Unreal Engine, etc.

## Testing

Once the software has been created by the developers, testers would check the new software’s functionality and ensure that it works as intended. Testers will normally counter check what’s been developed against the Use case, a document that specifies how the system should behave when used by a specific role/user.

Depending on the scale of the project, there may be several stages of testing such as ST (System Test) and UT (User Test), etc. I’m currently involved in a project at work where I’m a UAT tester. Whilst ST test the functionality of the software, UT are there to ensure that the software is fit for purpose and works as intended against the business rules. We also need to test end-to-end to ensure that transactions go through smoothly from start to finish across all departments.

Testers also need to raise any potential issues that may arise in case the system functionality contradicts with company policies and procedures. If there is a defect or bug in the software, this will be reported, investigate and sent back to the developers to fix. These defects are normally reported in a central system such as Jira.

There may also be separate testing environments to conduct the tests. For example, ST and UT may not have the same environment at all. This retains the integrity of the environment without causing a clash in testing, especially if it’s linked with databases.

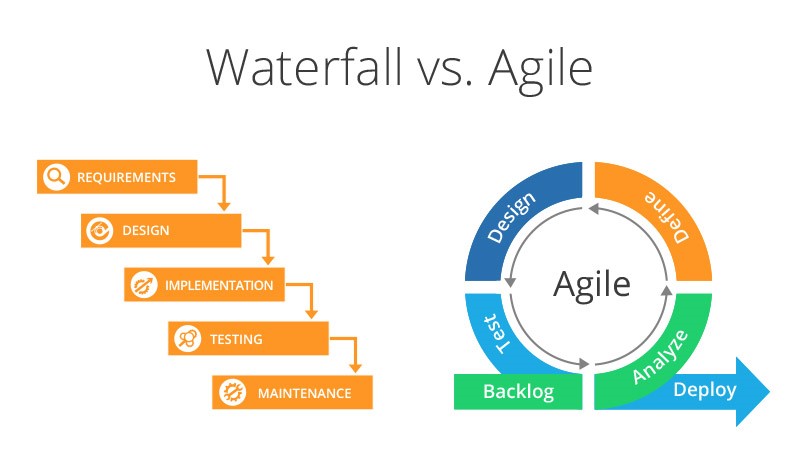
There are also different types of testing such as Manual testing and Automation testing. Manual testing is when a tester has to execute the test themselves. Automation testing is when the tool or software does the testing instead. Manual testing can be slower since a person will have to sign off the tests themselves in comparison to Automation testing where a system does it. However, humans are able to create complex scenarios and test the system thoroughly and test from various angles, including negative testing. When completing testing manually, the tester would normally need to scope and create step by step scripts whilst Automation testing is done with the help of another software.

If we follow the logic of testing a website that has just been developed, this will need to be tested in various browsers including

## Production

Once the software has been created, tested and signed off, it is now ready to be released and deployed in Production or the ‘live’ system. Since the software is launched in another separate environment, this will still need to be tested to ensure that it is working as expected.

## Project Methodologies



### Waterfall

The waterfall is a linear approach to a project that covers the following stages:

1. Requirements
2. Design
3. Implementation
4. Testing
5. Maintenance

The steps above aren’t always definite and is flexible depending on the project. However, the rule with Waterfall is that the project cannot backtrack once a phase has been completed. For instance, once the project reaches the Testing phase, they can no longer go back to the Requirements phase. If the project needed to do this for whatever reason, it becomes a bigger cost to the business to backtrack. The waterfall is an ideal method to use if the requirements for the project is clear, fixed and will not evolve.

The project I’m currently involved with at work is a Waterfall project. We are currently integrating an off the shelf product within our systems and changes are applied slowly but surely. The whole project is broken down in several phases and once a phase is completed, we can’t go back to it. Once a phase is signed off, it is deployed in production. We have thousands of existing and potential customers that still need ongoing service. This means that if we deploy the new software in production and there are issues, we still have our existing outdated systems to fall back on. However, we do conduct regression testing to ensure that any changes made in later phases doesn’t cause a defect in the previously signed off phases.

### Agile

On the other hand, Agile is tackling the project in smaller segments in a repeat cycle. It’s the waterfall method, but turned into an ongoing loop. Tasks or backlog are broken down into sprints and within each sprint, there are ongoing defects, backlog and new features introduced to the software. Once deployed in production, the business can monitor the performance of the new software and plan for any improvements in the next sprint. There is also ongoing feedback received from the product owner which makes them feel involved and if they’re not happy with a certain component, they are able to raise light on the issue early on in the development.