Software Development Unit 33, 1.4

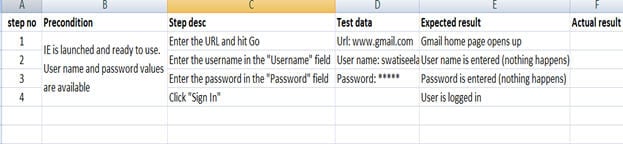
Anna Legaspi

# I can appreciate (describe) the need for testing and project-based development.

## Testing

### Manual Testing (Black box testing)

Manual testing is done by software testers who create scripts and manually test without knowledge of programming or coding. They are testing that the software behaves and functions as expected based on user stories and business rules. The scripts are normally done within MS Excel or a similar software and includes the scenario, the steps, action, expected result and the results. Screenshots of each step can also be recorded for audit purposes if necessary.



### Automation Testing (White box testing)

Automation testing is done my software developers and knowledge of coding and programming is required. This type of testing is also known as clear box, glass box or transparent box testing. It is done to test the integrity of the internal operation of the software. It is also more time consuming in comparison to black box testing. The software developers need to make sure that the logic of the software works based on the design document. There are a variety of automation testing tools software developers can use such as Selenium and Cucumber.

## Benefit and need for Testing

Testing is needed to ensure the software meets the requirements set by the project. Developing software is open to making errors and mistakes, therefore it’s important to test what’s been developed to ensure it is working as expected. The goal of most projects is to deliver a functioning software that behaves as expected.

Testers would also need to identify any bugs or defects in the software. Once raised, the developers will need to review and fix these before the product is released in production. It isn’t always possible to fix every single defect and some of these may be deferred to a later time, depending on the scale of the project. However, Identifying and fixing bugs in the testing phase is normally cheaper than making amendments to a product that has already gone into production.

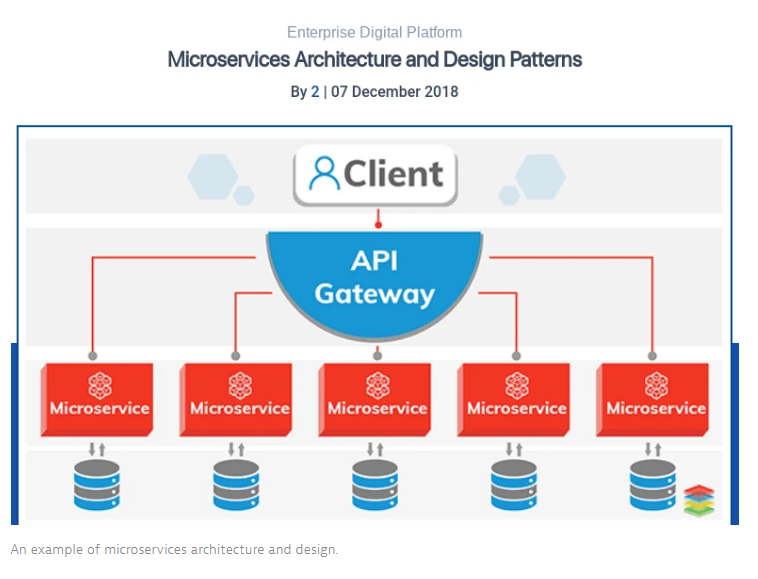
## Project based development

### Monolithic VS Microservices

There are a couple of different standards when developing project-based software: Monolithic and Microservices. Following the Monolithic approach results to a singular software that does everything within itself and its servers. Monolithic used to be a popular way, but people started having issues with managing all the requirements in one server/service. It can become cumbersome and unmanageable, especially when all the services overlapped and there is no way to unentangle Most people

On the other hand, Microservices is when there are several ‘micro’ services that provide various services to the client via an API (application programming interface) gateway. For example, a person may want to purchase an item in an online store. To complete this transaction, the client (website) will need to access a variety of services and servers (micro services) and the API Gateway allows this communication. A possible example of these services are as follows:

1. Authentication: The customer logs in to their personal account on the online website. Completing this transaction meant accessing a service that allows the authentication of the customer.
2. Products retrieval: The products available for purchase are currently stored in a database. The website needs to communicate to the database to allow it to display the products for sale.
3. Purchase: The website also needs to access the ability to charge the customer for their purchase and allow card payment services from another server.



Microservices is now the more efficient and popular way to develop projects because of its benefits. If one component of the service is down, it doesn’t affect the entirety of the project. At the same time, different team members can work on different parts of the project at the same time which can be more time efficient.

### V model

The V model is also known as the Verification and Validation model. This is because each the testing element (validation) is completed alongside the planning (verification). The benefit to this is that the there is already feedback received from the testers to how the product should look like in the end, therefore defects are found early in the project.

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Requirements Design | Acceptance test |
| System Design | System test |
| Architecture Design | Integration test |
| Module Design | Unit test |
| Coding | |

