SOFTWARE TESTS

s3

Welles,Nick N.H.M.

2022 - 2023

Table of content

[DOT Framework 2](#_Toc121917789)

[Research in ICT 2](#_Toc121917790)

[DOT (Development Oriented Triangulation) 2](#_Toc121917791)

[The “What” 2](#_Toc121917792)

[The ”Why” 2](#_Toc121917793)

[The “How” 2](#_Toc121917794)

[Software testing 4](#_Toc121917795)

[What is software testing 4](#_Toc121917796)

[Why software testing is so important 4](#_Toc121917797)

[How do we test software 5](#_Toc121917798)

[Acceptance testing 5](#_Toc121917799)

[Integration testing 5](#_Toc121917800)

[Unit testing 5](#_Toc121917801)

[Functional testing 5](#_Toc121917802)

[Performance testing 6](#_Toc121917803)

[Regression testing 6](#_Toc121917804)

[Stress testing 6](#_Toc121917805)

[Usability testing 7](#_Toc121917806)

[End to end testing 7](#_Toc121917807)

[Which tests are important for me 7](#_Toc121917808)

[Sources 8](#_Toc121917809)

# DOT Framework

## Research in ICT

The DOT framework is a set of research methods for practice-based research in ICT. The research is centred around the product, and not in essence on making new information which is the point in most logical research.

A small overview of possible ICT research:

* investigating the stakeholders' wishes
* choosing the most appropriate technology for a (part of) ICT system
* investigating the usefulness of a certain technology or framework
* testing the quality of an ICT product

## DOT (Development Oriented Triangulation)

The DOT Framework consists of tree levels that you can include in your research.

* The "**What**" of your research (the domains)
* The "**Why**" of your research (the trade-offs)
* The "**How**" of your research (the strategies and methods)

### The “What”

You can research both the particular application setting of your undertaking and the more broad information accessible that can help your venture. The two sorts of research will help you, and are important to make your own "innovation", your new product/counsel/report. We can communicate this by three areas. The main space is the "application domain". This is the space of the particular setting that the ICT project happens. The subsequent space is of "available work". All suitable hypothesis, models and different antiques that you can utilize are essential for the accessible work space. Thirdly is the "development area", where your genuine advancement happens and the research is all finished.

### The ”Why”

In many cases you need to guarantee that your product ultimately depends on contemporary quality principles. In those cases you need to utilize all the skill accessible to make your product. You do this by doing research in the "available work" area. As there are not many techniques that at the same time enhance fit and aptitude joining the two sorts of strategies in the project is frequently required.

Likewise there is a trade-off between improving outline and sureness. Frequently, generally at the outset you need to acquire a decent outline over what is required or what is accessible. At different times you need to test explicit parts of your work, ensuring it works. In these cases you attempt to arrange your research to upgrade "certainty" about your theories or objectives.

Ultimately, much of the time your research could require an information situated way to deal with legitimize your decisions, demonstrate culmination, etc. At times, be that as it may, it is smart to pick a more motivation based strategy, similar to the Talk technique.

### The “How”

During the project you try to learn as much as possible about the meaning of the available work, the application context and the innovation space. The DOT Framework has 5 available research strategies.

* Library

Library research is finished to investigate what is now finished and what rules and speculations exist that could be useful to you further your plan. Since the appearance of the web library research is additionally called work area research.

* Field

Field research is finished to investigate the application setting. You apply a field methodology to get to know your end clients, their necessities, wants and limits as hierarchical and actual settings in which they will utilize your product.

* Lab

Lab research is finished to test parts or ideas of your product, of the end result. You use lab research to learn in the event that the work resolve the manner in which you expected them, or to test various situations.

* Showroom

Showroom research is finished to test your thoughts comparable to existing work. Showing your model to specialists can be a type of display area research or illuminating how your product is not the same as the opposition. Likewise testing your product to overall rules is a type of display area research.

* Workshop

Workshop research is finished to investigate open doors. Prototyping, planning and co-creation exercises are ways of acquiring experiences in what is conceivable and the way in which things could work.

Source: <https://ictresearchmethods.nl/The_DOT_Framework>

# Software testing

## What is software testing

Software testing is the most common way of assessing and checking that a software product or application does what it should do. The advantages of testing incorporate forestalling bugs, decreasing development costs and further developing performance.

There are many types of software testing each with explicit goals and strategies:

* **Acceptance testing**
* **Integration testing**
* **Unit testing**
* **Functional testing**
* **Performance testing**
* **Regression testing**
* **Stress testing**
* **Usability testing**
* **End to end testing**

For each situation, approving base prerequisites is a basic evaluation. Similarly as significant, exploratory testing assists an analyser or testing group with revealing hard-to-foresee situations and circumstances that can prompt software blunders.

Indeed, even a basic application can be dependent upon a huge number and assortment of tests. A test the board plan assists with focusing on which sorts of testing offer the most benefit - given accessible time and assets. Testing adequacy is advanced by running the least number of tests to track down the biggest number of imperfections.

## Why software testing is so important

Late delivery or software defects can damage a brand’s reputation which leaves the brand with frustrated customers. Testing your software can save you a few dollars or millions per year in development and support. Early testing can uncover problems before the product is on the market. The sooner the problem occurs, the sooner they can address issues, such as:

* Architectural flaws
* Poor design decisions
* Invalid or incorrect functionality
* Security vulnerabilities
* Scalability issues

Security is viewed as the most vulnerable and delicate part. There are a lot of circumstances where the data and subtleties of the clients are taken and they are utilized for the advantages.  
It is viewed as the motivation behind why individuals search for the very much tried and reliable products.

With testing you can ensure the quality of the product. Products should be serving the user in one way or the other. It is a must that it is going to bring the value, as per the promise. With the guide of Quality Assurance, you can track down a wide cluster of situations and mistakes, for the reproduction of the blunder. It is truly straightforward and the engineers need to fix a similar in the blink of an eye. Moreover, software analysers ought to be working with the development group parallelly, which is valuable in the speed increase of the development procedure.

## How do we test software

As mentioned before in ‘What is software testing’, there are multiple ways to ensure the products complies to certain goals and strategies. In this part of the research we will read though them what you need to do in order to complete the types of tests.

### Acceptance testing

With acceptance testing you verify whether the whole system works as intended. the acceptance test is requested by the client. Tests directed here are monotonous, as they would have been shrouded in system testing.

Then, why is this testing is conducted by customers?

* To gain confidence in the product that is getting released to the market.
* To ensure that the product is working in the way it has to.
* To ensure that the product matches current market standards and is competitive enough with the other similar products in the market.

### Integration testing

Integration testing is characterized as a sort of testing where software modules are coordinated consistently and tried collectively. A common software project comprises of numerous software modules, coded by various developers. The reason for this degree of testing is to uncover absconds in the association between these software modules when they are coordinated.

### Unit testing

Unit testing is a kind of software testing where individual units or parts of a software are tried. The object is to approve that every unit of the software code proceeds true to form. Unit Testing is finished during the development (coding period) of an application by the engineers. Unit Tests segregate a part of code and confirm its rightness. A unit might be a singular capability, technique, strategy, module, or item.

This type of testing is significant in light of the fact that software designers once in a while take a stab at saving time doing negligible unit testing and this is fantasy on the grounds that unseemly unit testing prompts significant expense Imperfection fixing during Framework Testing, Integration Testing and, surprisingly, Beta Testing after application is constructed. In the event that legitimate unit testing is finished in early development, it sets aside time and cash eventually.

### Functional testing

This type of testing validates the software framework against the utilitarian necessities/details. The motivation behind Utilitarian tests is to test each capability of the software application, by giving suitable info, checking the result against the Useful prerequisites.

The prime objective of functional testing is actually taking a look at the functionalities of the software framework. It principally focuses on:

* Mainline functions: Testing the main functions of an application
* Basic Usability: It involves basic usability testing of the system. It checks whether a user can freely navigate through the screens without any difficulties.
* Accessibility: Checks the accessibility of the system for the user
* Error Conditions: Usage of testing techniques to check for error conditions. It checks whether suitable error messages are displayed.

### Performance testing

Performance testing is a software testing process utilized for testing the speed, reaction time, steadiness, dependability, versatility, and asset utilization of a software application under a specific responsibility. The fundamental reason for performance testing is to recognize and take out the performance bottlenecks in the software application. It is a subset of performance designing and is otherwise called "Perf testing".

Features and functionality upheld by a software framework are not by any means the only concern. A software application's performance, similar to its reaction time, unwavering quality, asset use, and versatility, do matter. The objective of Performance Testing isn't to track down bugs however to dispense with performance bottlenecks.

Performance testing is finished to give partners data about their application in regards to speed, dependability, and versatility. All the more critically, Performance Testing uncovers what should be further developed before the product goes to advertise. Without Performance Testing, the software is probably going to experience the ill effects of issues, for example, running delayed while a few clients use it at the same time, irregularities across various working frameworks, and unfortunate ease of use.

### Regression testing

This type of test confirms that a new program or code change has not unfavourably impacted existing elements. Relapse Testing is only a full or incomplete choice of currently executed experiments that are re-executed to guarantee existing functionalities turn out great.

There is a need for regression testing at whatever point the code is changed, and you want to decide if the changed code will influence different pieces of the software application. Besides, relapse testing is required when another element is added to the software application. Relapse tests may likewise be performed when a utilitarian or performance imperfection/issue is fixed.

### Stress testing

Stress testing verifies stability and reliability of software application. The objective of Stress testing is estimating software on its robustness and mistake taking care of capacities under incredibly weighty burden conditions and guaranteeing that software doesn't crash under crunch circumstances. It even tests past typical working focuses and assesses how software functions under outrageous circumstances.

Consider the following real-time examples where we can discover the usage of Stress Testing:

* During festival time, an online shopping website might observer a spike in rush hour gridlock, or when it reports a deal.
* At the point when a blog is referenced in a main paper, it encounters an unexpected flood in rush hour gridlock.

The objective of stress testing is to investigate the way of behaving of the framework after a disappointment. For stress testing to find success, a framework ought to show a fitting mistake message while it is under outrageous circumstances. To direct stress testing, now and again, huge informational indexes might be utilized which might get lost during stress testing. Analysers shouldn't lose this security-related information while doing pressure testing.

### Usability testing

A usability test is a method to test how easy and user-friendly a software application is. A little arrangement of target end-clients, use software application to uncover convenience deserts. Convenience testing predominantly focuses around client's simplicity of utilizing application, adaptability of application to deal with controls and capacity of application to meet its targets.

There are many software applications/websites, which miserably fail, once launched, due to following reasons:

* Where do I click next?
* Which page needs to be navigated?
* Which Icon or Jargon represents what?
* Error messages are not consistent or effectively displayed
* Session time not sufficient.

More of a usability testing method can be found in the [UX Design](../UX%20Design.docx) document.

### End to end testing

End-to-end software testing is a crucial step in the software development process, as it helps ensure that a software application works correctly from start to finish. This type of testing involves testing the entire software application, from the user interface to the back-end systems, to make sure that it functions as intended.

One key advantage of end-to-end testing is that it helps identify potential issues that may not be detectable through other types of testing. For example, while unit testing focuses on individual components of the software, end-to-end testing looks at how these components work together to provide the desired functionality. This can help identify issues that may arise when different components interact, such as compatibility issues or performance bottlenecks.

Another benefit of end-to-end testing is that it can help ensure that the software application meets the needs of the end user. By testing the application from the user's perspective, end-to-end testing can help identify any user experience issues, such as usability problems or confusing interfaces. This can help improve the overall quality of the software and make it more user-friendly.

# Which tests are important for me

The most important tests for my own project are unit tests, integration tests, security tests, regression tests, end to end tests, stress tests and performance tests.

1. Unit tests to see if the component is actually expected to handle the data as expected.
2. Security tests to make sure you can't manipulate another user's data as a user.
3. Regression tests to see if the rest of the application still works as it should after implementing new features or fixing bugs.
4. End to end tests to see if the front-end can actually pick up data from the database through API calls.
5. Stress tests to see if the website can still display and sell products under a heavy load.
6. Performance tests to see if the website loads quickly at all times and works properly to the client's satisfaction.

# Sources

<https://www.ibm.com/topics/software-testing>   
<https://www.indiumsoftware.com/blog/why-software-testing/>   
<https://www.guru99.com/usability-testing-tutorial.html>   
<https://www.guru99.com/stress-testing-tutorial.html>   
<https://www.guru99.com/regression-testing.html>   
<https://www.guru99.com/performance-testing.html>   
<https://www.guru99.com/functional-testing.html>  
<https://www.guru99.com/user-acceptance-testing.html>  
<https://www.guru99.com/unit-testing-guide.html>  
<https://www.guru99.com/integration-testing.html>  
<https://www.guru99.com/end-to-end-testing.html>