

What is Software Testing? Definition, Basics & Types...

[Simon Hall May 2021.]

Sources: Multiple: Developers & Leaders, Google... And... Gartner.

Simon Hall

May 2021.



SIMON HALL MAY 2021.

What is Software Testing? Definition, Basics & Types...

Software Testing

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing as a White Box and Black Box Testing. In simple terms, Software Testing means the Verification of Application Under Test (AUT). This tutorial introduces testing software to the audience and justifies its importance.

Why is Software Testing Important?

Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

What are the consequences of not doing Software Testing?

The cost to fix bugs or defects post-development is a significant consideration associated with improper software testing. These costs may include long work hours, lost productivity, and profit loss during software downtime.

Risks of Not Testing Software Properly

Life is full of risks, but with careful planning and strategic testing, there are ways to mitigate those risks. The job of software testers hinges solely on risk mitigation; insufficient software testing (or a complete lack thereof) increases the chance of developer complications, stakeholder disapproval, and bad consumer experiences. Software testing is, therefore, a top priority among development teams. Here are some of the most common risks of not testing software properly before deployment.

Excessive Expenses

Software bugs require prompt attention to rectify. This is easy during pre-launch testing, especially when conducting agile testing, or continuous testing that is concurrent with software development. However, bug remedies become increasingly difficult the more deeply ingrained into the software they become. The cost to fix bugs or defects post-development is a significant consideration associated with improper software testing. These costs may include long work hours, lost productivity, and profit loss during software downtime.

Delayed Product Launches

Customers demand software updates that solve their problems and make their lives easier or more pleasurable. They buy countless software products every day to meet their needs. Unfortunately, companies that don't launch updates quickly get left in the dust alongside My-Space, Napster, and Tamagotchi toys.

SIMON HALL MAY 2021.

However, if a product is to launch quickly without defects, it must be tested throughout the development process – or at the very least before it hits the market. While in-house software testing can help catch many bugs before launch, without third-party software

testing, many defects risk oversight until it is too late. In other words, yes, testing could extend launch time by a few weeks, but not testing at all could stall launch indefinitely. That's because inadequate pre-launch testing forces teams scramble to pick up the pieces of broken software applications.

Dissatisfied Users

Software users have certain expectations about the products they use; they expect them to run fast, work properly, and protect their private information. If software bugs prevent them from having a positive user experience (UX), they will likely churn perhaps never to return. Notably, poor user experiences are more likely to garner negative public feedback than positive experiences. What's worse: negative feedback could block potential leads from ever trying the software in the first place.

According to a survey by Dimensional Research (sponsored by ZenDesk), 95 percent of people who had a negative interaction with a product or service will share their bad experience with friends, family, and social media. Only 87 percent shared positive experiences. Additionally, people are more likely to share negative experiences across multiple platforms (social media, review sites, etc.) compared to positive ones. As Inc. explains, a single negative review requires as much as 40 positive reviews to counteract it when considering variables like human behavior, basic mathematics, and simple logic.

The best way to prevent user dissatisfaction is to test software products with an experienced test team. Software testers know how to “break” a system before the public gets their hands on it. They understand how a system is supposed to work and how it is not supposed to work. They test the software by pushing the wrong buttons, entering incorrect information, and navigating the system through non-standard means. Finally, they provide clear documentation outlining their processes and the errors they find so developers can correct any concerns before they become overly problematic.

Reduced Brand Reputation

A single bad experience has a ripple effect. Not only is a user less likely to use a product after having a negative experience (and shouting that experience out to the world), they are also more likely to perceive the brand, as a whole, in a bad light. As the negative reviews compile, so does the negative perception of the brand supporting the product.

Let's take a look back to 2013 when Target experienced a significant security breach across its national outlet stores on Black Friday. The data breach affected over 100 million customers who had their debit and credit card information stolen from the retailer. The ordeal caused Target's profits to drop a whopping 46 percent in the fourth quarter. Additionally, the fallout forced Chief Executive, Gregg Steinhafel, to step down from his position and cost the company over \$300 million to resolve. There continue to be repercussions of the breach, including class-action lawsuits against the company

SIMON HALL MAY 2021.

and the necessity of partners like Visa and Mastercard to monitor their user's financial services more diligently than ever. Many victims still find mortgages and loans challenging to acquire, as well.

Project managers must juggle many balls to keep software development on time and budget but should never compromise quality by omitting software testing. The risks of not testing software properly far outweigh any benefits of a quick – yet shoddy – software launch.

Software Testing Basic concepts

Mocks, Stubbs & Fakes in testing...

- Used in unit testing.
- Replace class dependencies.
- Allow isolation of class under test.

Fake

- A Fake is a simplified implementation of a dependency.
- Usually coded directly, without the use of a framework.
- Does not provide direct validation of how the class uses the dependency.
Disadvantage.
 - If it is important to test whether a class is interacting with a dependency, “in a specific way”, it may be worth considering a Mock.
- Used when the class being tested requires specific logic in the dependency. May involve implementing the bare minimum of the dependency.

Stub

- Provides a way to inject data into the class.
- Usually created using a mocking framework.
- Does not provide validation of how the class uses the dependency.
- Used when data is required by the class; but the process used to obtain it isn't relevant to what's being tested.
- Usually use a framework to create the stub then in the test method explicitly define the data required. Most mocking frameworks also allow you to create Stubbs.

Mock

- Mechanism for validating how a dependency is used by the class.
- Created using a mocking framework.
- Can provide data or logic required by the class.
- Used when the test requires that the class performs specific actions on the dependency.
- A mock is the only one that can cause a test to fail via assertions.

SIMON HALL MAY 2021.

What Are the Different Types of Testing?

There are many different types of testing. Here is a quick breakdown of the most common testing types:

- Accessibility testing

Accessibility testing is the practice of making your web and mobile apps usable to as many people as possible. It makes apps accessible to those with disabilities, such as vision impairment, hearing disabilities, and other physical or cognitive conditions.

- Acceptance testing

In software testing, the ISTQB defines acceptance testing as: Formal testing with respect to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether to accept the system.

- Black box testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

- End-to-end testing

End to end testing (E2E testing) refers to a software testing method that involves testing an application's workflow from beginning to end. This method basically aims to replicate real user scenarios so that the system can be validated for integration and data integrity.

- Functional Testing

Functional testing is a quality assurance process and a type of “black-box” testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered.

- Interactive testing

IAST (interactive application security testing) analyses code for security vulnerabilities while the app is run by an automated test, human tester, or any activity “interacting” with the application functionality.

- Integration testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing.

SIMON HALL MAY 2021.

- Load testing

Load testing is the process of putting demand on a software system and measuring its response.

- Non functional testing

Non-functional testing is the testing of a software application or system for its non-functional requirements: the way a system operates, rather than specific behaviours of that system. ... For example, software performance is a broad term that includes many specific requirements like reliability and scalability.

- Performance testing

In software quality assurance, performance testing is in general a testing practice performed to determine how a system performs in terms of responsiveness and stability under a particular workload.

- Regression testing

Regression testing is re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change. If not, that would be called a regression.

- Sanity testing

A sanity check or sanity test is a basic test to quickly evaluate whether a claim or the result of a calculation can possibly be true. It is a simple check to see if the produced material is rational.

- Security testing

SECURITY TESTING is a type of Software Testing that uncovers vulnerabilities, threats, risks in a software application and prevents malicious attacks from intruders.

- Single user performance testing

Client-side performance testing, also known as single user performance testing, evaluates client-side KPIs like response time and device vitals. ... SLAs for acceptable user experience should be set (e.g., time to login, time to accomplish a task) and measured in all scenarios.

- Smoke testing

In computer programming and software testing, smoke testing is preliminary testing to reveal simple failures severe enough to, for example, reject a prospective software release.

SIMON HALL MAY 2021.

- Stress testing

Stress testing is a software testing activity that determines the robustness of software by testing beyond the limits of normal operation. Stress testing is particularly important for "mission critical" software; but is used for all types of software.

- Unit testing

In computer programming, unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use.

- White box testing

White-box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In “white-box” testing an internal perspective of the system, as well as programming skills, are used to design test cases.

- And many more...

Many of these types of testing can be done manually — or they can be automated.

E.G. Large UK or International Supermarket Chain – running back-end in Java.

Technology Team Department (Probably multi-site & Global... Some elements probably outsourced...) responsible for scoping out requirements with leaders, running projects/sprints, developing new content/upgrading existing codebase/manipulating & integrating data and systems from newly acquired companies (Like Acutest was in 2017 by Capita).

The Essential Enterprise Resource Planning Systems/MiddleWare/Applications/APIs need to be knitted in/integrated/joined with Group systems for full & proper Board Level Management Information & Insights, HR & Benefits... So everyone gets paid & had healthcare & option of gym memberships & other cool benefits/So Compensation bands can be aligned...

SIMON HALL MAY 2021.

Technology Team Department - Consists of:

Divisional & Group Board's Commercial Leadership & Chief Visionary Officers (Set the direction of Travel... that determines strategy => Outcomes & Return On Investment (ROI).

|

Tech Leaders

Business Analysts

|

Agile Coaches

|

Development Teams for Modules

|

Software Testing QA

|

DevOps

- All... Get together to swarm/Mob Resolve for big releases or problems in Slack.
- Lots of small teams of 3 developers (1 Dev, 1 Coding Manager, 1 Project Manager)

Development Team roles...

PM

A software project manager defines the requirements of the project, builds the project team, lays out a blueprint for the whole project including the project scope and parameters, clearly communicates the goals of the project to the team; the targets to be achieved, allots budget to the various tasks to be completed, and ensures that the expectations of the Board of Directors and Stakeholders are met through timely completion of the project.

- Run Jira.
- Set up Sprints.

Scrum Master

The scrum master is the team role responsible for ensuring the team lives agile values and principles and follows the processes and practices that the team agreed they would use. The responsibilities of this role can include: Clearing obstacles. Establishing an environment where the team can be effective.

- Master of ceremony.
- References what blockages there are.

Developers

Software developers typically do the following: Analyse users' needs and then design, test, and develop software to meet those needs. Recommend software upgrades for customers' existing programs and systems. Design each piece of an application or system and plan how the pieces will work together.

SIMON HALL MAY 2021.

Google – “GitFlow”, used for branch Management.

SDET Software Developers in Test

An SDET, in layman terms, is a developer who instead of working in the product development team, works as part of the test team. In essence, SDETs are responsible not only for writing code but are required to test the code as well. SDETs are required to continuously write, test, and fix the written code.

Great source of SDET Training material freely available

<https://www.youtube.com/c/pavanoltraining/playlists>

Includes in-depth vlog-Instruction_&_Code_Alongs_like.THIS...

QA Manual Testing Course for beginners

<https://www.youtube.com/watch?v=QJqNYhiHysM>

SDET Unicorns - Why is it so Hard to Hire SDETs?

<https://devqa.io/sdet-hiring-software-developers-in-test/>

SDETs Test for functionality.

1. Releasing a component or Upgrade for the Authors.
2. When Author created component need to release a new version of codebase to production. Companies use Content Management Systems (CMS) when they need to release a new version of codebase for production. One such CMS is Jenkins.
3. Jenkins DevOps use to release / Build this new release.
They also look Build Git Branches requested by Development.

A DevOps tools approach to agile software development helps developers and operations teams build, test, deploy and monitor applications with speed, quality and control. Successful DevOps software implementations generally rely on an integrated set of solutions, or a toolchain, to remove manual steps, reduce errors, increase team agility and scale beyond small, isolated teams.

IBM has the agile practices, cloud-native tools and DevOps tools you need to help improve the entire DevOps lifecycle, from planning and builds to software testing and application monitoring.

N.B. If... some tests have failed component/upgrade cannot be released to Production.

SIMON HALL MAY 2021.

What is Production in Software Development?

Production is the final environment in your software development process. It is the work that is ready to be publicly available, and only the most thoroughly tested code should end up here. While most people associate the production environment with the product being live, this is not necessarily the case.

- There are work arounds.
- Create a bug fix branch & release as soon as possible.
- Sometimes things fail due to environment differences in staging.
- If there is a data difference...

At this Supermarket chain a large challenge is that they can't containerize the data. There is a risk of an error in Production if a developer ran short of time didn't write tests/didn't test.

TDD is good in principle but not always feasible. Sometimes teams just have to push out a product.

As a tester you set those/these Quality Assurance Gates.

There is a product called SonarQube which will analyse code line by line & will examine lines of code hit by the test & tests.

SonarQube

N.B. If 70% of lines of code covered are ok. This is deemed to be good enough.

When things go wrong a Jira ticket is raised in Slack to fix => & only after launch.

Tests

1. 1st Assert; that "a thing" [function, data, process, network, your car's engine, your broadband... etc...] is as Asserted/expected...
2. Jenkins is a DevOps Build Manager / Automation Server. It is used for Continuous Integration Continuous Development Pipeline management.
E.G. Upgrading to the next version of CMS.

DevOps Control

- Manage servers & upkeep.
- Releases.
- Delivery of pipelines to environments.
- Release functionality to production environment.

N.B. WIX is a Content Management System Online!

E.G. National/International Supermarket uses/hosts a Content Management System (CMS) called Magnolia.

They use SonarQube.

What is SonarQube?

SonarQube is an open-source platform developed by "SonarSource" for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

SIMON HALL MAY 2021.

SonarQube & alternatives (below) analyse code line by line & will examine lines of code hit/covered/targeted by the test & tests.

If 70% of lines of code covered are “All Okey Dokey!”. This is considered “Good Enough”!

Developer: Open-source software, “SonarSource”

Classifications: Static program analysis

Alternatives to SonarQube:

- Coverity.
- Checkmarx.
- Veracode Application Security Platform.
- Kiuwan Code Security & Insights.
- Embold.
- Klocwork.
- Codacy.
- GitLab.

When things go wrong... Tickets are raised in Jira and sent to the relevant folks in Slack. N.B. Jira is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management.

E.G. Regulatory System in a Big Bank...

Due to a Developer (Pressed to hit deadline) did not carry out proper rigorous Unit Testing for function... The knock-on effect was/is/could-be INCONSISTENT DATA (Wrong Data... Possibly the system not being available).

E.G. Securities or Commodities Exchange

LSE had a problem that delayed trading by 30mins

Hargreaves Lansdown suffers system outage amid record trading volumes...

“Siobhan Riding and Madison Darbyshire in London NOVEMBER 9 2020”:

<https://www.ft.com/content/6b2f26f6-5dff-4218-beb0-59f544aebd5b>

E.G. Ariane 5

On June 4th, 1996, the very first Ariane 5 rocket ignited its engines and began speeding away from the coast of French Guiana. 37 seconds later, the rocket flipped 90 degrees in the wrong direction, and less than two seconds later, aerodynamic forces ripped the boosters apart from the main stage at a height of 4km. This caused the self-destruct mechanism to trigger, and the spacecraft was consumed in a gigantic fireball of liquid hydrogen.

SIMON HALL MAY 2021.

The disastrous launch cost approximately \$370m, led to a public inquiry, and through the destruction of the rocket's payload, delayed scientific research into workings of the Earth's magnetosphere for almost 4 years. The Ariane 5 launch is widely acknowledged as one of the most expensive software failures in history.

What went wrong?

The fault was quickly identified as a software bug in the rocket's Inertial Reference System. The rocket used this system to determine whether it was pointing up or down, which is formally known as the horizontal bias, or informally as a BH value. This value was represented by a 64-bit floating variable, which was perfectly adequate.

However, problems began to occur when the software attempted to stuff this 64-bit variable, which can represent billions of potential values, into a 16-bit integer, which can only represent 65,535 potential values. For the first few seconds of flight, the rocket's acceleration was low, so the conversion between these two values was successful. However, as the rocket's velocity increased, the 64-bit variable exceeded 65k, and became too large to fit in a 16-bit variable. It was at this point that the processor encountered an operand error; and... populated the BH variable with a diagnostic value.

Not enough space to reach space.

In layman's terms, this can be thought of as attempting to fit 10 million litres of ice cream into a camping fridge on a hot summer's day. It'll be fine for the first few tubs, but after a certain threshold, you'll be unable to fit anything else in, the fridge door will be stuck wide open, and everything will start melting really, really fast.

The backup Inertial Reference System also failed due to the same error condition, meaning that at T+37 the BH variable contained a diagnostic value from the processor, intended for debugging purposes only. This was mistakenly interpreted as actual flight data; and... caused the engines to immediately over-correct by thrusting in the wrong direction, resulting in the destruction of the rocket seconds later.

It worked on the last device

Several factors make this failure particularly galling. Firstly, the BH value wasn't even required after launch, and had simply been left in the codebase from the rocket's predecessor, the Ariane 4, which did require this value for post-launch alignment. Secondly, code which would have caught and handled these conversion errors had been disabled for the BH value, due to performance constraints on the Ariane 4 hardware which did not apply to Ariane 5.

SIMON HALL MAY 2021.

A final contributing factor was a change in user requirements - specifically in the rocket's flight plan. The Ariane 5 launched with a much steeper trajectory than the Ariane 4, which resulted in greater vertical velocity. As the rocket sped to space faster, there was a higher certainty that the BH value would encounter the conversion error.

Ultimately, the European Space Agency assembled a team to recover logs from the two Inertial Reference Systems, which were spread over a debris field of approximately 12 square kilometres. Their work was impeded by treacherous marshland terrain, hazardous chemicals dispersed from the rocket, and immense public scrutiny from the media, all because of a single type casting error.

This (referenced) Google-able series on the Worst Software Bugs in History... is in honor of Bug Day 2017. A more recent one:

{ <https://www.wired.com/2005/11/historys-worst-software-bugs/> }.

Seventy years ago, Grace Hopper discovered the first computer bug — a moth was stuck between relays in the Harvard Mark II computer she was working on. The notion of bugs was described in other fields previously, but the moth discovery was the first use of the term “debugging” in the field of computers.

Bugsnag automatically monitors your applications for harmful errors and alerts you to them, giving you visibility into the stability of your software. You can think of them as mission control for software quality.

<https://www.bugsnag.com/blog/bug-day-ariane-5-disaster>

More examples available at this url:

<https://www.cigniti.com/blog/37-software-failures-inadequate-software-testing/>

Failing to carry out Rigorous Testing on Safety Critical Systems is “LESSSSS GOOD”!

SIMON HALL MAY 2021.

TESTING IN ACADEMIA – E.G. Computational Scientist

All Research is bespoke... Has 1 specific purpose... to Test... hypothesis or hypotheses related to very specific and widely varying topics.

Typically carried out using R.

What is R?

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. Polls, data mining surveys, and studies of scholarly literature databases show substantial increases in popularity; as of April 2021, R ranks 16th in the TIOBE index, a measure of popularity of programming languages.

The official R software environment is a GNU package. It is written primarily in C, Fortran, and R itself (thus, it is partially self-hosting) and is freely available under the GNU General Public License. Pre-compiled executables are provided for various operating systems. Although R has a command line interface, there are several third-party graphical user interfaces, such as RStudio, an integrated development environment, and Jupyter, a notebook interface.

E.G. Biological Computational Scientist Research

Data required is very specific case.

Large amounts of time are spent “Rangling” data in preparation for query/queries via R Scripts.

SIMON HALL MAY 2021.

As a major cause of error due to lack of testing in Commerce can come from lack of testing resulting in DATA PROBLEMS. Testing is less important. That said getting the scripts “Bang On” and having meaningful results means that this “data wrangling” and the correct R scripted queries are critical to meaningful research analysis. Academics are funded. Get it wrong & funding will dry up.

Analysis starts with plotting data... Such as... What are the Statistical properties & possible outliers? From here the actual interpretation of results is analysed and reported on. Progress reports to Funding Providers & Senior Academics are weekly (at least).

Computational scientists vary in the tools they use... Some use all the aids available... Some “purist/s” use text editors like EMACS (ESS – EMACS Speaks Statistics) & the relevant plugins for R. Some use RSTUDIO (IDE).

The data they analyse varies in size greatly.

Big... 2-3 Million Rows.

Medium... A couple of Thousand Rows.

Small... Table... 10 lines.

For the Medium & Big data Super Computers are utilized to run multiple queries and operations SYNCHRONOUSLY (At the same time).

Sometimes a huge number of CORES (CPU) are required... Sometimes not.

Big... Imperial College London... UpTo 2000 Cores (One can’t use them all at once).

Medium... 24 Modern Super PC.

Small... 2 PC.

Complex calls that are independent of each other at somewhere like CERN could use up substantial share of Core capacity.

Typically there is a “Core Manager” / “Queue Manager” prioritising & allocating Core capacity depending on priority & availability.

E.G. Processing data coming from the large HADRON Collider – Higgs Boson was detected like this.

SIMON HALL MAY 2021.

Computational Scientists use an Open Source... Open Science Grid... On which Super Computers are available to use when the Cores are idle.

Imperial College 2000 Cores
Brazil 280 Cores
These are large & distributed.

Super Computers can run 100s of processes that take a short time to run SYNCHRONOUSLY & some... not many that take very lengthy time to process.

These are Super High Memory (RAM... Space to move things around... Context shift & multi task.) processes!

Big... Up to 1 or 2 Terabytes.
Medium... 1-4 Gigabytes.
Small... A couple of hundred Megabytes.

E.G. Big... Like Genomic Data used for DNA sequencing.

Storage (Text File, Like filing cabinet space)
Big... Greater than 0.5 Gigabytes.
Medium... From a couple – 50 – 100 Megabytes.
Small... 100 Kilobytes.

Processing Power requires more Cores & Access time.
Memory RAM GBs/TBs – has ability to concurrently handle BIG DATA. Like the size of the prep area in your kitchen enables only a certain volume of activity relative to a Restaurant kitchen.

E.G. Citibank Consumer Banking...
Has large distributed systems & BIG DATA, Huge Cloud Storage (Multi-site/available & has several failsafe back up facilities). A simple cashpoint transaction interface with the API & change to the persistent database layer doesn't use much processing power Memory RAM. They'll have a great deal of "Rack-Space" and in fact I read recently that there's even competition for the land to build these centre/s on and... that a lot of old data centres are being updated as such (supply & demand, costs & margin... Service and safety though?).

N.B. There's a land grab going on in some places for new data centres.

SIMON HALL MAY 2021.

What does take up huge processing power is Artificial Intelligence & learning applied to this BIG DATA!

In plain English...

It is unlikely that basic transaction load on cash points or loans... will require a lot of Cores.

Unless they are running machine learning or AI then it's SIZEABLE!

Interesting to get a detailed insight very left of field and an insight in to the "horse-power" that underpins it. And... The market forces at play!