

TECHNOLOGY

ISSUE 2022

DEV TESTING TEAM

Insights!



TABLE OF CONTENTS

IMPORTANCE OF TESTING/QA 04

BIG TECH TESTING PRACTICES 05

SOFTWARE MALFUNCTIONS 06

WHY QA NEEDS HUMANS 08

BENEFITS OF TESTING TEAM 10

TESTING SOFTWARE CYCLE 12



**“BE A YARDSTICK OF
QUALITY. SOME PEOPLE
AREN’T USED TO AN
ENVIRONMENT WHERE
EXCELLENCE IS EXPECTED.”**

~ STEVE JOB

WHY SOFTWARE TESTING/QA IS IMPORTANT

Software is virtually omnipresent and affects everything we do.

With all software comes bugs, a coding error in a computer program that is due to:

- unclear or constantly changing requirements & timelines
- software complexity
- programming and bug tracking errors
- communication gap
- documentation errors
- deviation from standards

Bugs can be expensive and difficult to resolve and have been known to cause death, monetary repercussions and negative effects on people's privacy, valuable data and even their safety.

Although, software malfunctions are closely monitored these days, the potential for chaos, public relation disasters and spiraling costs still remain. Testing (QA) has become more of a necessity in recent years, although, plenty of businesses still view testing and monitoring as an optional extra, despite the fact that poor quality software continues to cost US organizations over \$2 trillion annually.

What has remained unchanged is the different methods of testing/quality assurance that can be brought to the product by dedicated QA teams:

- Developers
- Testers
- Employee's

BIG TECH TESTING PRACTICES

It has been interesting to watch the debate amongst big Tech companies about whether or not to engage in quality assurance projects and in some cases to even undervalue the practice. 5 effective and powerful ways to test like tech giants:

Google - the largest search engine developer maintains a large robust testing framework

Microsoft - ratio of testers to developers is currently around 2:3, Microsoft pays testers whom they call Software developers and developers equally

Facebook - does not employ any dedicated testers at all. Instead, the social media giant relies on its developers to test their own (as well as one another's) work.

Amazon - as soon as it notices that revenue is decreasing or customers are moving away due to anomalies on the website, Amazon increases its testing efforts.

Spotify - does employ dedicated testers. They are part of cross-functional teams, each with a specific mission

When a company has a small budget it is expected that software developers will test their own code although most do this anyway.

When testing activities is limited or moved lower on the list of priorities, it leads to the inability to improve an application.

Although, bugs may annoy a user experience, it does severely compromise product reputation and stability.

SOFTWARE MALFUNCTIONS

Here we look at tech giants that are also not immune to software pitfalls and a few well-known incidents caused by simply not testing their products:

Facebook: engineers identified a “massive ranking failure” that exposed as much as half of all News Feed views to potential “integrity risks” over the past six months

Yahoo, hackers leaked 500 million credentials due to a software security breach.


History’s worst software bug, a miscalculation in the software of a radiotherapy device caused the radiation dosage to go 100x higher than it was intended with disastrous consequences

A lack of quality control caused a commercial flight crash back in 2015. Airbus confirmed that a serious assembly control issue was the cause of the tragedy.

First National Bank of Chicago system error at a major U.S. bank credited \$920 million to 823 clients’ accounts.

Googles software update for the Nest ‘smart’ thermostat went wrong and literally left users in the cold.

Cyber Attack on Nuclear Power Plant causing several years of disruption the head of an international nuclear energy consortium declared

A dramatic landscape photograph featuring a large fire in the background, silhouetted trees, and a bright sun or moon behind a hazy sky. The foreground is filled with dark, silhouetted grasses and branches. The text is overlaid on the left side of the image.

The purpose of quality assurance and the main benefits of including QA in the product development process quite simply prevents mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to end users.

REASONS WHY QA NEEDS HUMANS:

Elon Musk and Steve Hawking have expressed their fears that computer power is superior to human power.

With software quality assurance (QA). In many cases, testing software can be done quicker and more efficiently by using automated tests which can detect if a website's link is broken. But, can't determine how to easily navigate and use your application.

Software is developed to make the internet a prettier place for humans to interact and, we use these tools to our ability.

Computers cannot discern if a page is pleasing to the eye. With user acceptance testing (UAT), people try the system and reveal that the programmers' idea of easy is click hell. Human testing is then done to see exactly why the user isn't clicking the meant-to-be obvious button and instead keeps logging out.

Automated tests "Designing a piece of software might make perfect sense to the architects and programmers," says Russell Myers, former QA analyst. But if the average user finds the software confusing or unhelpful, he adds, "That's going to piss them off, and it's really going to affect usability."

There's nothing on Earth better at judgment than humans.

REASONS WHY QA NEEDS HUMANS:

Humans are better at spotting errors

Often with the testing team a font change or a misplaced format, is only obvious to them.

Computers only spot the errors they're designed to catch i.e. if the text is wrong or a word is misspelled, a computer does not know the meaning of what it sees unless its given that set of instructions.

Humans communicate

Testers, not only find bugs; they provide valuable feedback on documentation and training

Software development is sometimes spread across multiple teams, which can make synchronizing project work complicated if one team is outsourced and another team is off-shore.

BENEFITS OF A SKILLED TESTING(QA) TEAM:

MEET END USER DEMANDS AND EXPECTATIONS

- User experience can make or break a product
- Slow, unproductive, and glitchy software will fail to bring the company value.

Quality assurance ensures that end users are satisfied, and that the application closes all functionality gaps.

BUILD TRUST AND PRODUCE A QUALITY PRODUCT

- Having a high-quality product makes the process of building long-term and mutually beneficial business relationships easier.
- By prioritizing quality, it helps our end user generate value and greater profits.

Long term, stability and sustainability of our services and applications converts happy clients into returning customers.

BENEFITS OF A SKILLED TESTING(QA) TEAM:

SAVE COSTS, DETECTS AND FIXES ISSUES EARLY

- Bugs are an inevitable part of the software development cycle.
- Avoid chaos by discovering bugs early in the development process reduces late-stage delays, hurried fixes, clogging up developers' schedules, and less impact on resources.

Organized testing processes and having a clear testing implement allows better allocation of resources to catch bugs reducing cost and time.

SET AND MAINTAIN HIGH QUALITY STANDARDS AND FOCUSES ON PREVENTION

A consistent quality assurance strategy, and agile methodology best practices does bear fruit resulting in more time for exploratory testing and searching for improvement possibilities.

Given today's quick paced and fierce competition in IT, investing in quality assurance is not only worthwhile, but indispensable and it is useful to know that Frega does use QA team best practices to achieve optimal results.

"One App at A Time,
Together We Will Get There"
@ Frega Dev Testers

TECHNOLOGY



TESTING IN SOFTWARE DEVELOPMENT CYCLE

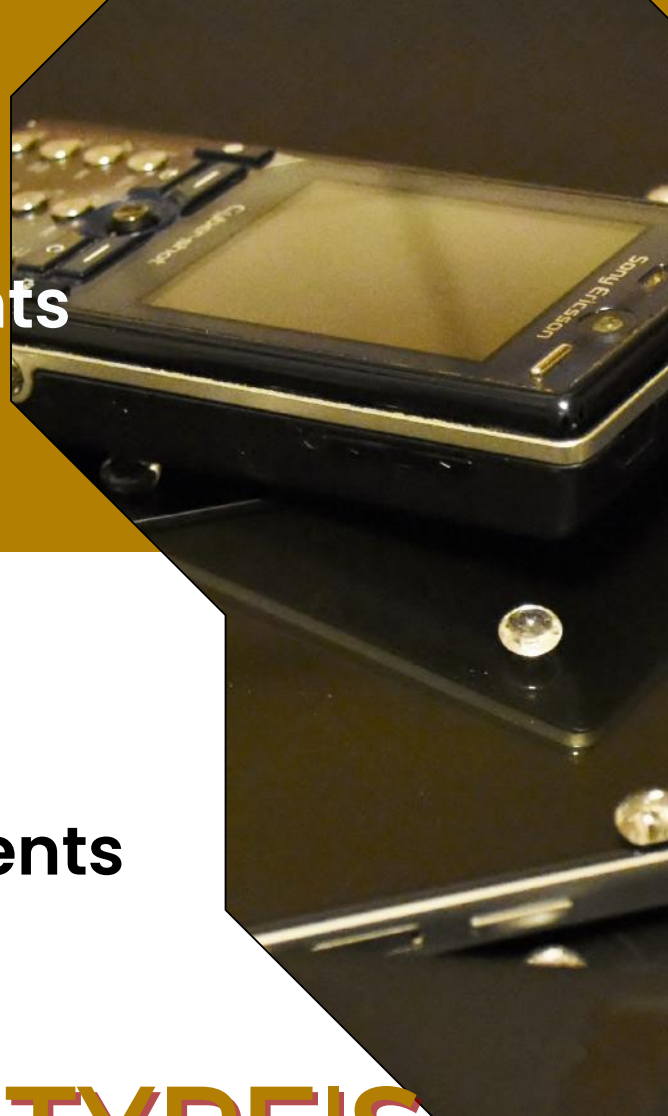
Testing is an expected requirement of the software development process that requires attention to detail and a methodical approach.

The Software testing stages of progression can be simplified and categorised as the following:

- Unit Testing
- Integration Testing
- Functional Testing
 - System Testing
 - User Acceptance Testing

UNIT

Test separate components



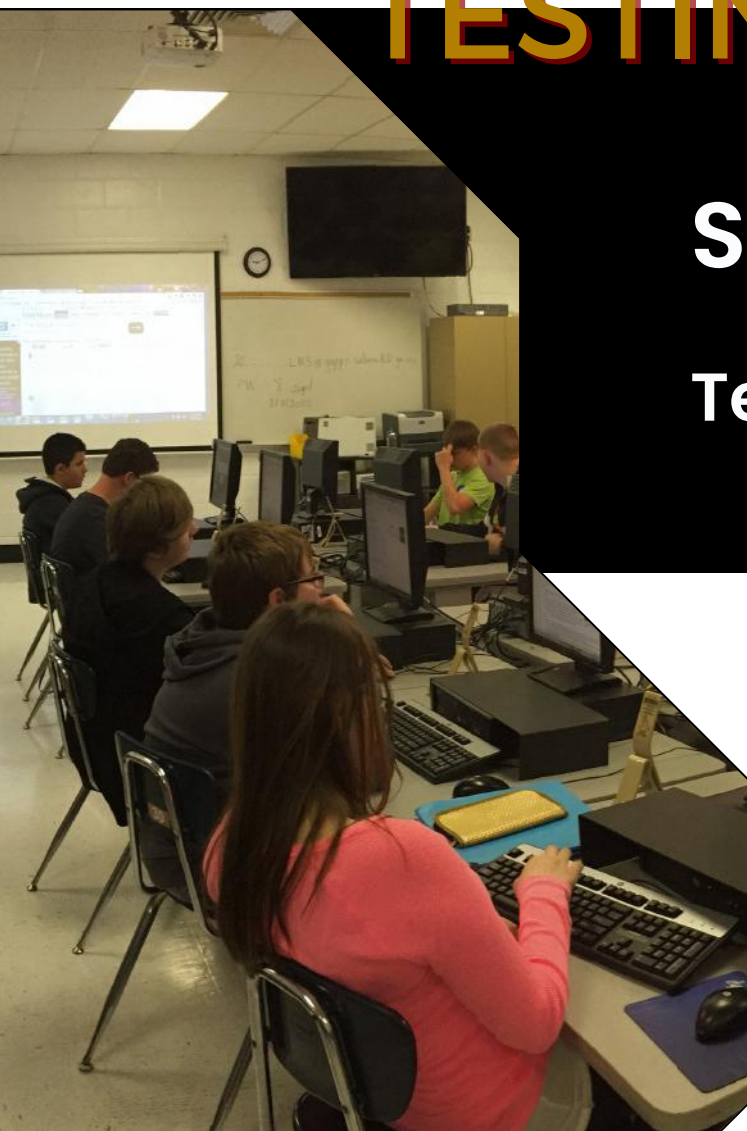
INTEGRATION

Test integrated components

TESTING TYPE'S

SYSTEM

Test the entire system



ACCEPTANCE

Test the final system

UNIT TESTING

Here we test the smaller components of the software called a single unit.

Units can be defined as individual functions, code components classes, or as large as single features of the software.

Often, it is the smallest testable component of the software application that the testers compile load and execute.

Unit testing ensures that each unit of the software is functioning as it should.

White box, testing is used where we see and test the code

Carried out at the initial phase of testing and then can be performed anytime.

Uncover the issues within the functionality of individual modules only.

Does not expose integration errors or system-wide issues.

UNIT TESTING

This stage is normally completed by our Programmers or those whom are familiar with coding protocols as it:

- is done before Integration testing enabling early detection of issues and easily reducing any impact
- checks for the positive behavior i.e. the correct output In case of valid input, and also the failures that occur with invalid input.
- finds issues/bugs at an early stage reducing the overall project costs.
- tests small pieces of code or individual functions so the issues/errors found in these test cases are independent and do not impact the other test cases.
- unit test cases simplify and make testing of code easier as only the latest change in the code is tested.
- saves time and cost, and it is reusable and easy to maintain.

INTEGRATION TESTING

The main function or goal of this testing is to test the interfaces between the units/modules.

The meaning of Integration testing is that it integrates and combines each unit tested module one by one and then test the behavior collectively.

This stage of testing is necessary:

- to ensure that the collective system follows the correct data flow
- that the integrated system is prepared for system testing
- detects the errors related to the interface.
- helps uncover bugs that may arise when different modules interact with each other to form the overall system

INTEGRATION TESTING

Must be carried out after unit testing and before system testing.

Frega's software integrations:

FregaRooms - powered by Zoom integration



FregaMoney - PFS Card Services



KnowledgePlus - Algolia



SYSTEM TESTING

Where our collective systems in integrated testing is combined further into a single integrated system.

Here testers evaluate the functional requirements of the project once more and also see whether the system follows compliance according to the given requirements.

Besides that, testers can also evaluate how overall components interact with each other. Therefore, they can perform specific testing procedures such as performance, load, reliability and security testing on the integrated system.

Black box testing, from a user's point of view



USER ACCEPTANCE (UAT) TESTING

User acceptance testing, where our team will test components of our Frega software to see if it meets the expected requirements.

Here is where we interface with our Tech Support via the Jira Tes board:

- to record our results and recommend that further enhancements be made
- or that testing has met its desired outcome

This is normally the final interaction with end-users or stakeholders that all the agreed-upon requirements are a part of the product.

However, testing still continues with each software update.





WEBLET



*T*ogether

*E*veryone

*A*chieves

*M*ore