END TERM MAJOR PROJECT REPORT
(1st February 2021 – 2nd July 2021)
On

Internship experience as a Platform Engineer at Icertis

Under the guidance of

Sachin Jere, Associate Manager

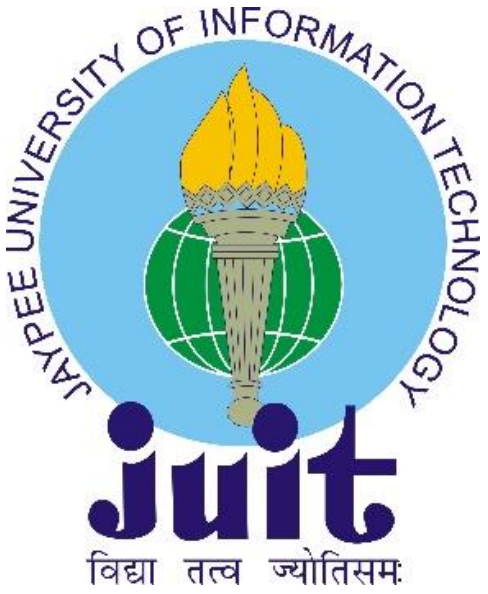
Icertis Solutions

Submitted by:

Devesh Rattan Sharma

171369

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
DISTT. SOLAN, HIMACHAL PRADESH 173234
(2017-2021)

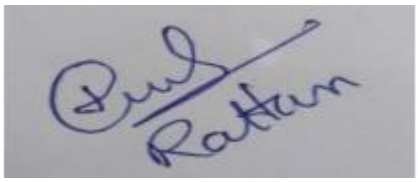


DECLARATION

I hereby declare that the work reported in this report entitled **"Internship experience as a Platform Engineer at Icertis"** is submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science and Engineering**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work

Icertis does not allow the project material to be used beyond the said guidelines, I Devesh Rattan Sharma am aware of the fact and appropriately created the project report without violating any compliance.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Devesh Rattan Sharma
Enrolment No.: 171380
B. Tech (CSE)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Sachin Jere

Sachin Jere
Associate Manager
Icertis
Date: 25th May 2021

ACKNOWLEDGEMENT

It was a privilege for me to work as a full-time intern at **"ICERTIS SOLUTIONS"** under the supervision of Mr. Sachin Jere.

This report describes the training that underwent, for the duration of 28 February 2021 - 25th June 2021 at intern at "**ICERTIS SOLUTIONS**" a world leader in providing a Contract Lifecycle Management (SaaS) Solution.

I would like to express our sincere gratitude of the all the people who have helped and supported me throughout. I am deeply indebted to Mr Sachin Jere (Associate Manager), Swapnil Mali (Senior Software Engineer) and other fellow colleagues at **ICERTIS SOLUTIONS** for organizing an effortless internship program, efficiently and providing me valuable resources and for their cooperation and willingness to share their expertise and knowledge and to devote their precious time to discuss related topics.

The help and co-operation extended by the staff at **ICERTIS SOLUTIONS** is fully acknowledged. I thoroughly enjoyed my entire internship program and would like to thank everyone at **ICERTIS SOLUTIONS** for their guidance and support.

TABLE OF CONTENTS

ABSTRACT

This report describes the internship that I did with "**ICERTIS SOLUTIONS**", Pune Office, during the period of February to June 2021. The project assigned to me was "**Internship experience as a Platform Engineer at Icertis**". The report itself sheds light on the various process and concepts I have learned as a full-time intern.

The Work from home internship started on 1st February 2021. I was assigned with the Platform Engineering department. Initially the PE interns were given a basic training around the workflow and security awareness. The second training that the interns received was a basic overview of the ICI platform, where we got a hands-on experience to use the platform, we created contracts, agreements, etc, and got a sense of the Platform.

Then to sharpen our technical skills we were given a C# and .NET training by an external company **EduRamp** which is specialised to give this training to B2B clients. The training helped us understand the basics of C# and .NET, this made us ready for our next endeavour working for a functional team at Icertis.

Working with a functional team was the real experience that we as interns were excited about. I was assigned with a team responsible to upgrade the ICI platform and introduce new features in the platform, this was a major upgrade as the UI was to be updated. These new changes meant that the UI automation framework also needed an update, so that automated test cases could be written for the new platform. I was assigned a part of this task to update the framework and automate few test cases that our team was working on. It was a great learning experience as I could use my newly acquired skills in the real world.

Chapter I

INTRODUCTION

1.1 About Icertis

Icertis is World leader in Contract Lifecycle Management (CLM) with its ICM platform, Icertis platform manages more than 10 million contracts, in 90 different countries and 40+ supported languages. Icertis has offices in 12 countries, with Pune being the biggest hub, catering more than a 1000 Icertians. With a strong business outlook, Icertis is an employee centric company with strong foundations of FORTE values instilled in the very roots of the organization.

1.2 Internship Program

The learning journey contains two stages, first stage consisted of a basic company and product overview, and in the second stage the interns were put in a functional team to work alongside experienced engineers to get a real hands-on experience.

- Company and product overview
- C# and .NET basics
- Selenium and OpenXML

STAGE 2 – Working with a Functional team

- Worked with a team responsible for the upgrade of the ICI Platform

1.3 Software Testing and Development

In the information age, all organizations depend intensely on data for everyday use, except data itself, needs to investigate to be followed up on by human entertainers. This is the place where programming comes in software helps in everything from ticket booking to fragile operation, virtual products have given clients to do things effectively and save time.

Software itself is an immense subject and software improvement and testing is at the core of each software organization. Software advancement alludes to the cycles and models that help in making of the software, these cycles and models give a through comprehension of what can anyone do, how it ought to be finished. Software testing is huge and complex field of testing software applications.

Automation of testing system helps the software advancement firms and business to reduce on expense and settle on successful choices and improvement is simpler and stronger. Web applications are the ideal possibility for robotization as they need to serve a wide range of sorts of programs just as various customers also. Consequently, Automated testing of sites is gigantic expense saving procedure for organizations.
An automated site sees the fascinating necessities of visitors and updates content for each watcher. The item helps with supporting help contacts even more beneficially. It is responsive (adaptable pleasing) out-of-the-case, totally joined with each exhibiting channel, and flexible to the changing necessities of leads and customers.

C# and .NET is characterized by a determination and comprises of a programming language, a compiler, center libraries and a runtime (CLR). .NET framework is used to compile C# code to make it machine independent. The C# code is compiled to intermediate language (IL) which is according to the specifications of the common language infrastructure (CLI). This code is run on the CLR provided by the .NET framework which makes the code truly machine independent.

Selenium is an open source instrument that robotizes internet browsers. Selenium is Open Source UI Automation Tool. It is utilized to Automate the Web-Application however not for Desk-Top

Base Application Automation. It was made in 2004 by a ThoughtWorks Engineer. Selenium Web-driver was presented later in 2006

Chapter II

LITERATURE SURVEY

2.1 Difference Between C# and .NET

2.1 C# and .NET

S.No	C#	.NET
1.	It is a programming language	It is a framework on which language is built.
2	It is also created by Microsoft after C and C++.	It is created by Microsoft and is Network Enabled Technology..
3.	It is flexible and very easy to use than Java and C++	It supports many programming languages and contains libraries of the languages that they will use.
4.	It's vast features allow the developers to create a good number of projects and applications.	It's RunTime Environment is CLR, which is Common Language Runtime.

In paper [1][2][3], Developers write test cases alongside standard code in unit testing, which is a

common procedure. Automation systems like JUnit for C# have popularised this approach by allowing unit test suites to be run frequently and automatically. Regardless of how unit testing is perceived in practise, software engineering researchers see room for innovation and are investigating innovative strategies such as automatic unit-test generation. We used a global online marketing research tool to perform a survey of 225 app developers, covering various programming languages and 29-countries, in order to match such research with the needs of practitioners. The results of the survey show that unit testing is an integral part of software production and that there is a need for further research into unit testing automation. The findings assist us in identifying areas of interest for which additional testing is needed (e.g., unit test maintenance), as well as providing insight into the suitability of online marketing research tools for software's engineering survey**.**

Chapter III

SOFTWARE DEVELOPMENT

3.1 Software Development Lifecycle

A Software Development Lifecycle Model tells us about the types of activities executed at every stage in a software development project, and how the ventures relate to one another aptly and chronologically.

- SDLC is the elision of Software Development Life Cycle.
- It is furthermore called as Software Development Process.
- SDLC is a complex set apart assignments executed at each parade in the product up gradation measure.
- ISO/IEC 12207 is a global typical for programming wheel of life measures. It is said to be the streamer that sets every one of the assignments needed for fabricating and looking after programming.

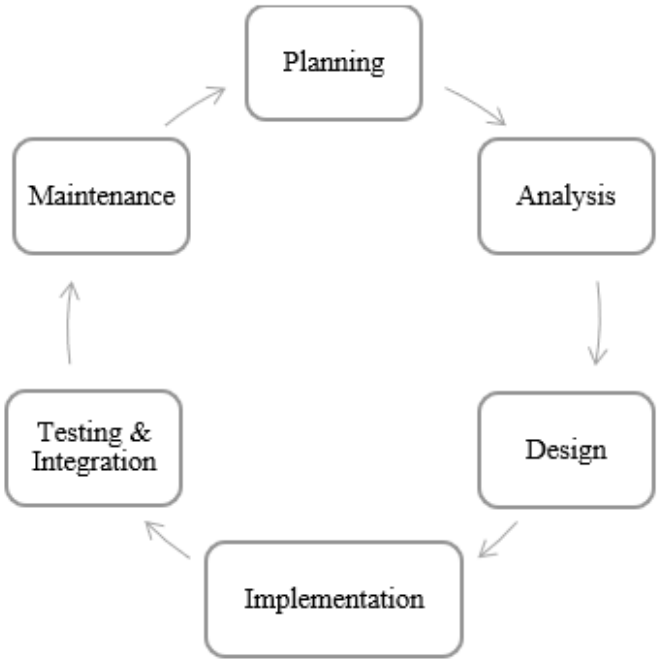


Fig 2.1 Graphical representation of the processes of SDLC

3.2 Software Development Models

Software Development is generally of two types:

1. Sequential
2. Iterative and Incremental

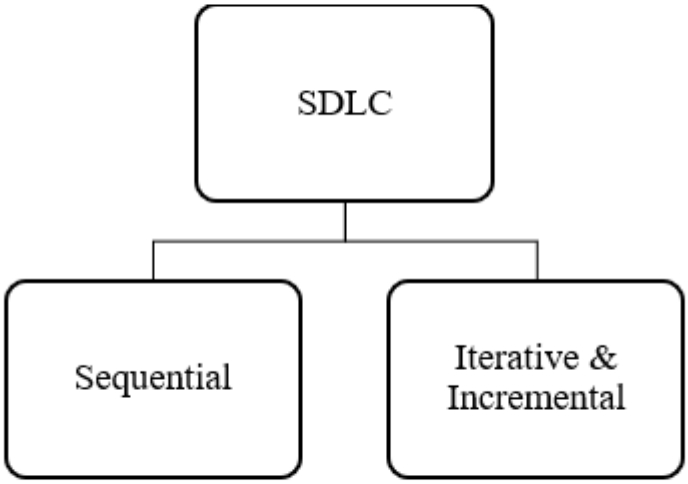


Figure 2.2 Types of SDLC

3.1.2 Sequential Software Development

A sequential development model expresses the software development process as a continuous, progressive flow of activities. This implies that any stage in the advancement, interaction should start when the past stage is finished. In principle, there is as such no cover of stages, yet by and by, it is useful to have early input from the accompanying stage.

Advantages

- Cost Effective
- Less time taking
- Suitable for different geographical positions
- It's linear
- Maximized customer satisfaction
- No pre knowledge required

Disadvantages

- Rigid
- Absence of central authority
- Lack of intuitiveness
- No centrality of the client

Waterfall Model

In the Waterfall model, the maturing ventures are completed one after another. In this model, test ventures only occur after all other maturing activities have finally been out righted.

Its diagrammatic portrayal looks like a course of cascades.

Figure 1Stages in Waterfall model

V-Model

Disparate the Waterfall model, the V-model mingles the test operations all-round the development process, executing the principle of before time testing.

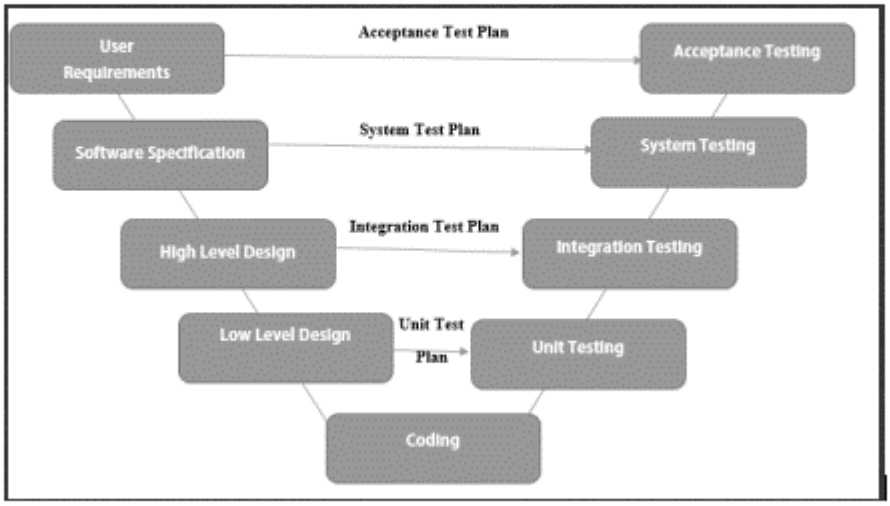


Fig 2.5 Steps Involved

Incremental and Iterative Development

Incremental Model is a cooperation of programming progression where essentials disengaged into various autonomous modules of the item improvement cycle. In this imitation, each and every module goes through the necessities, aim, execution, and preliminary stages. Each resulting coming of the section adds extent to the prior release. The cooperation gets moving until the whole skeleton achieved.

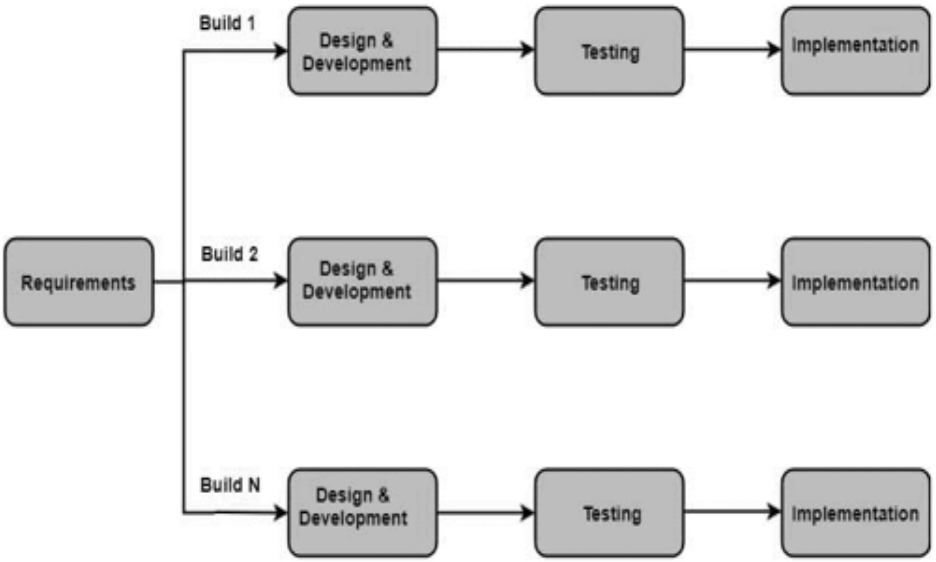


Fig 2.6 Incremental Model

In Iterative Model, you can begin with a portion of the product determinations and build up the main adaptation of the product. After the main form on the off chance that there is a need to change the product, another rendition of the product is made with another emphasis. Each arrival of the Iterative Model will result in a definite and fixed period that is called cycle.

This replica sanctions the coming by to antecedent stages, in which the diversities made discretely. The rearmost yield of the venture recharged towards the varnish of the Software Development Life Cycle measure.

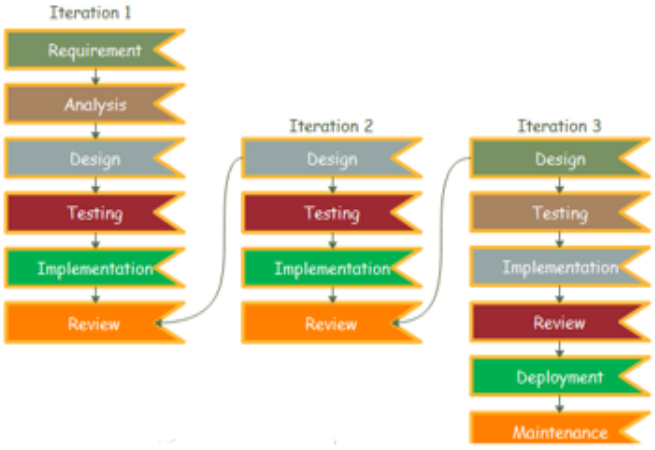


Fig 2.7 Iterative Model

3.3 Agile Methodology

Agile methodology for software development is a combination of incremental and iterative processes. With Agile the main objective of the developers is on adaptability and customer satisfaction by rapid delivery of working software products. These products are developed in rounds of weeks, which are then iterated over and over, keeping customer's priorities in mind.

Every agile iteration involves, cross functional teams working simultaneously on different areas like:

- Planning
- Requirement Analysis
- Design
- Coding/Development
- Unit Testing
- Acceptance testing

Agile model relies on the belief that every project has different requirements and has to be handled differently, therefore, the methods have to suit the project requirements.

In the earlier days of software development, there were different methodologies that were synonymous to agile, but there were few differences because each of these methodologies were developed by different groups with varying project requirements.

Popular methods like Extreme Programming (XP), Scrum, feature driven development, etc, are now collectively known as Agile Methodologies, after the agile manifesto came into existence in 2001.

Agile Values are:

- **Working Software:** A working prototype of the software is considered to be the best means of communication between the developers and customers as it allows customers to see the product and request features, while developers can observe the different interactions
- **Responsive:** The product has to responsive to changing customer needs, and teams should be able to quickly develop the feature requested.
- **Customer Interaction:** As the requirements cannot be gathered completely during the initialisation of the project and due to requirement changes and feature requests it is considered that customer should be involved in the product to get proper requirements and criteria for the product.
- **Individuals and Interaction:** The agile methodologies tells that interactions between individuals is a good way to develop software, which allows increased motivation, self-organization and tools like pair programming helps in developing a quality product.

3.4 Agile Principles

Agile principles are the guiding rules/practices that helps in implementation and execution with agility

- The highest priority is customer satisfaction by early and continuous delivery of valuable software.
- Agile process allows changes in products at every stage. Even in late stage of development changes by customers should be entertained by the development teams.
- Delivery of working software is of utmost important; the frequency could be a few weeks or months, but the shorter time period is preferred.
- Business representative and developers should work together as much as possible to ensure a product that fulfils customers' requirements.
- Projects should be developed by enthusiastic individuals who are motivated, and they should be provide with proper environment to harness their best.
- The most efficient way of communication between teams is through face to face interactions.
- Progress is measure by deliveries of working software
- Agile processes depend upon sustainable development, therefore the stakeholders involved in the project should be able to maintain a constant pace.
- Attention of good programming practices and good architecture enhances agility
- Agile process promotes simplicity to maximize the amount of work not to be done.
- The best architectures, designs and requirements are developed by self-organizing teams
- At subjected intervals, the development team reflect on the efficiency of development to become more effective accordingly

3.5 Software Testing

Software testing can be called as an affair which comprises several distinct activities or jobs. Of which execution is only one of these jobs

Software testing does the following:

- gauges the quality of the software
- risk of software failure in operation is reduced

Software Testing ≠ Test Execution

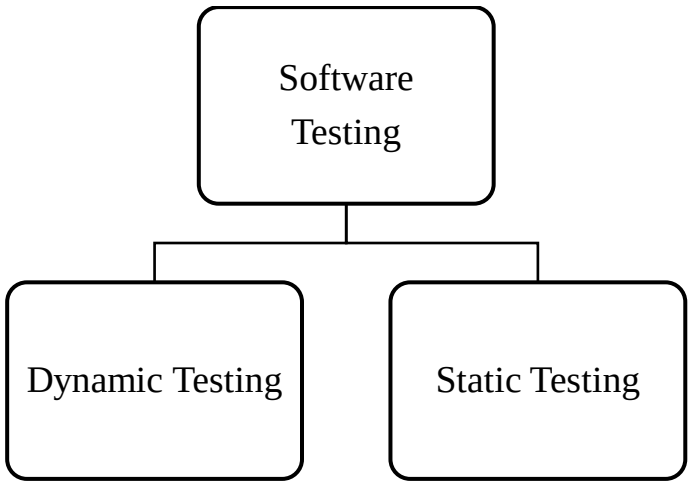


Figure 1 Types of Software Testing

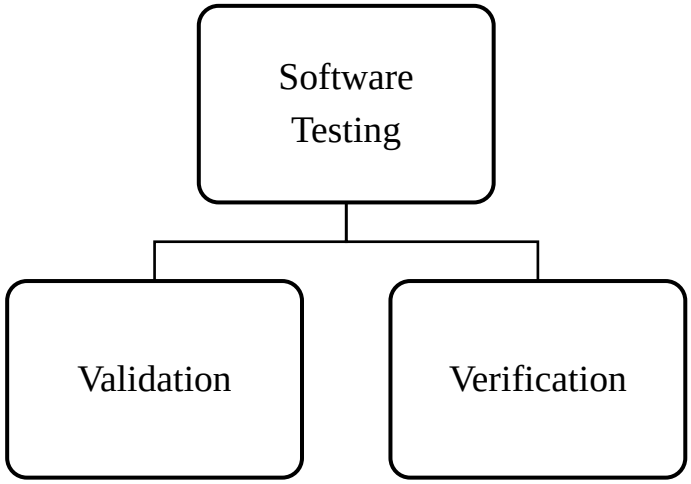


Figure 2Steps in Software Testing

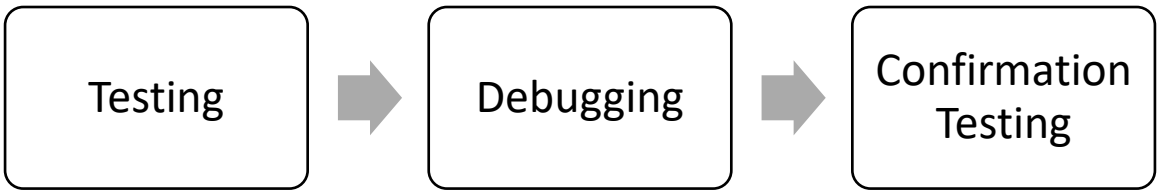


Figure 3Relationship between Testing and debugging

3.6 Testing Processes

There is nobody all-inclusive programming test measure, yet there are basic arrangements of test exercises without which testing will be more averse to accomplish its set up destinations.

Figure 4.11 Test Process

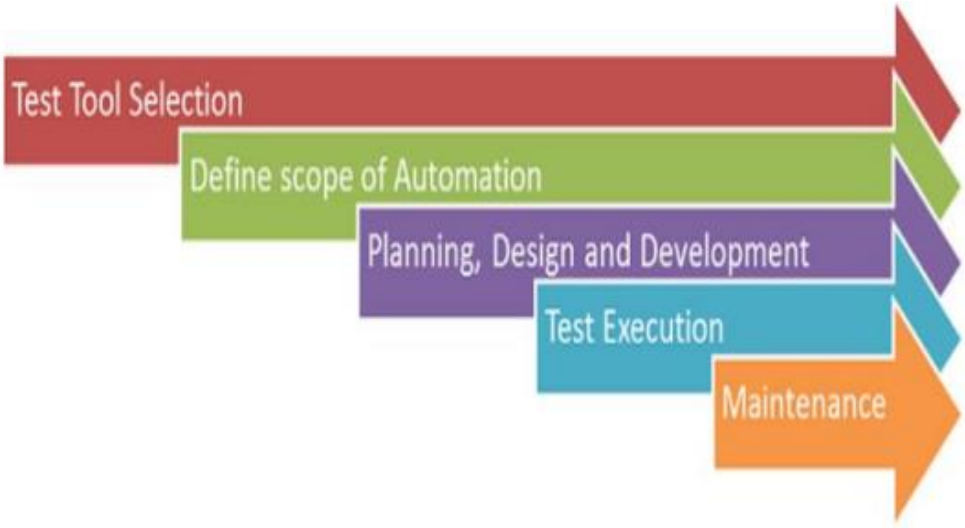


Figure 5.13 Test Process

3.7 Test Levels

Test levels are gatherings of test exercises that are coordinated and overseen together. Each test level is an occasion of the test interaction. Test levels are identified with different exercises inside the product improvement lifecycle.

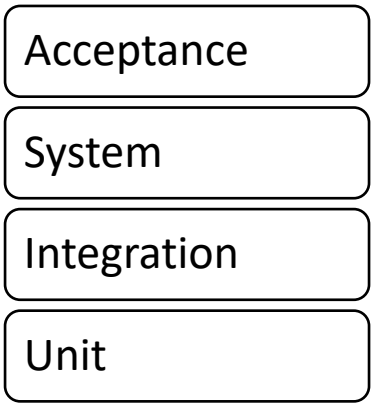


Figure 6Test Levels

3.8 Test case writing

A bunch of preconditions, inputs, activities (where relevant), anticipated outcomes and post conditions, created dependent on test conditions.

It involves:

- Test Case Title
- Pre-Conditions
- Test Steps
- Expected Results
- Test Suite
- Test Environment
- Actual Results
- Status
- Defect Report

3.9 Defect Report

Defect Report - Documentation of the event, nature, and status of a deformity. Wherein a blemish or inadequacy in a work item where it doesn't meet its prerequisites or particulars is a deformity.

Chapter IV

SYSTEM DEVELOPMENT

4.1 Programming Languages

4.1.1 C# or C Sharp

C# is articulated "C-Sharp".

It is an item arranged programming language made by Microsoft that sudden spikes in demand for the .NET Framework.

C# has roots from the C family, and the language is near other mainstream dialects like C++ and Java.

The principal rendition was delivered in year 2002. The most recent form, C# 8, was delivered in September 2019. C# is utilized for:

- Portable applications
- Work area applications
- Web applications
- Web administrations
- Sites
- Games
- VR
- Data set applications
- Also, a whole lot more!

Why is C # used?

It is one of the most widely used languages in the world. It is simple to learn and use. It has a lot of love from the media. C # is a target language for systems that offers transparency and encourages code reuse, lowering implementation costs. Since C # is more closely related to C, C ++, and Java, it is simpler for the framework to move from C to C # and vice versa.

4.2.1 .NET Framework

It is a software framework that was designed and developed by the company. The first version of the.Net platform, version 1.0, was released in 2002. In simple terms, it is a 20virtual machine for compiling and running programmes written in various languages such as C#, VB.Net, and others.

Form-based software, Web-based applications, and Web services are all built with it. On the, you can choose from a number of programming languages. The most common versions of the Net framework are VB.Net and C-sharp. It's used in building apps, among other platforms. It has a variety of features and even adheres to industry norms.

Basic Component Stack and Architecture of .NET

The fundamental architecture of is made up of the first three components from the bottom. After that, Microsoft added more modules to the Net platform, which was released in 2005. The following is the Net Framework:

Kinds of Applications: Mainly the applications which are underlying .Net system is partitioned into the accompanying three classifications:

❏ WinForms: Form – Based applications are considered under this classification. In straightforward terms, we can say user-based applications goes under this WinForms class.

❏ ASP.NET: ASP.Net is a structure for web and it gives the amazing coordination of HTML, CSS and JavaScript which makes it valuable to foster the web applications, sites and web administrations.

❏ ADO .NET: It includes applications that operate with data like MS SQL Server, Oracle and so forth comes. It predominantly comprises of classes that can be utilized to interface, recover, embed and erase information.

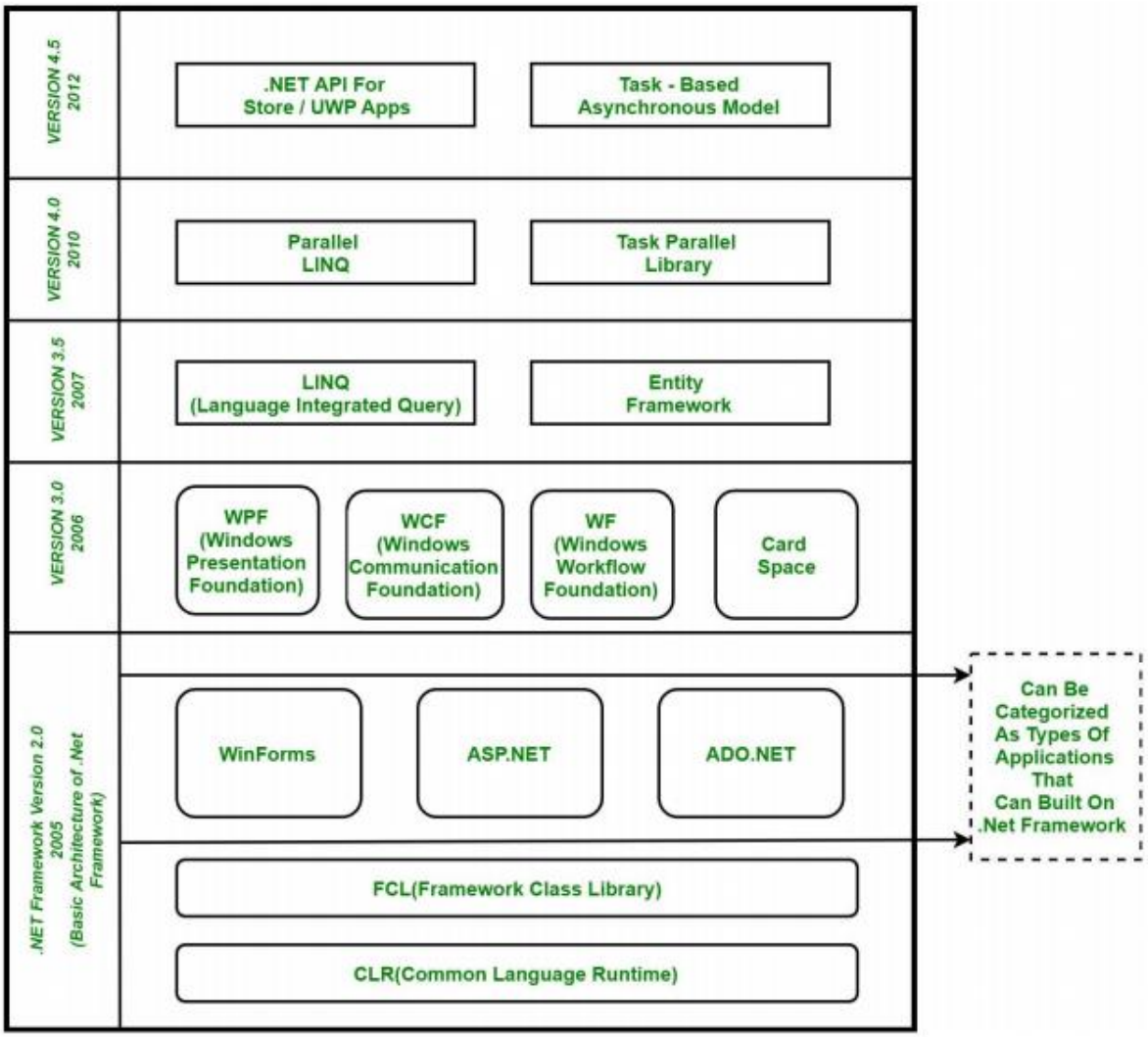


Fig 3.6 Architecture of .NET

Chapter V

UI AUTOMATION

UI automation testing process is like manual testing process, but the user doesn't have to click the automation does it for him, through the application, and visually verify the data. Each test case can be scripted in XML script and executed to generate a test result.

Selenium

Selenium is a free (open source) automated testing framework used to validate web

applications across different browsers and platforms. You can use multiple programming

languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done utilizing the Selenium testing device is generally alluded to as Selenium Testing.

Selenium Software isn't only a solitary instrument yet a set-up of programming, each piece obliging distinctive Selenium QA testing needs of an association. Here is the rundown of apparatuses

• Selenium Integrated Development Environment (IDE)

• Selenium Remote Control (RC)

• WebDriver

• Selenium Grid

XPath: XML path is a way to identify a WebElement in a HTML page. XPath provides various expression to matching expressions

Relative XPath: This involves searching the complete document using specific attributes, text or using relationship to possibly search unique text.

Absolute XPath: In this method the complete tags are provided from root to descendants with precision and without missing any of them.

XPath Functions:

1. Contains
2. Starts-with ()
3. Text ()
4. AND Operator
5. OR Operator

Example of an XPath in a Web Page:

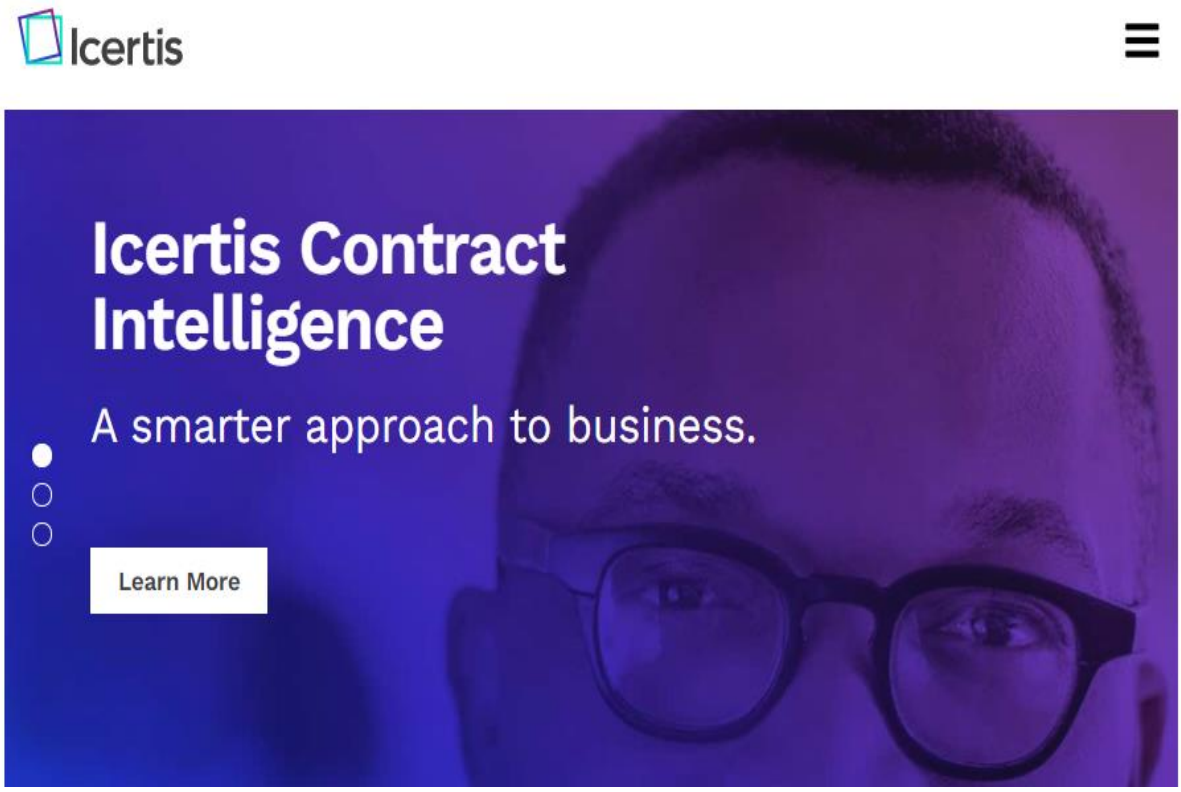
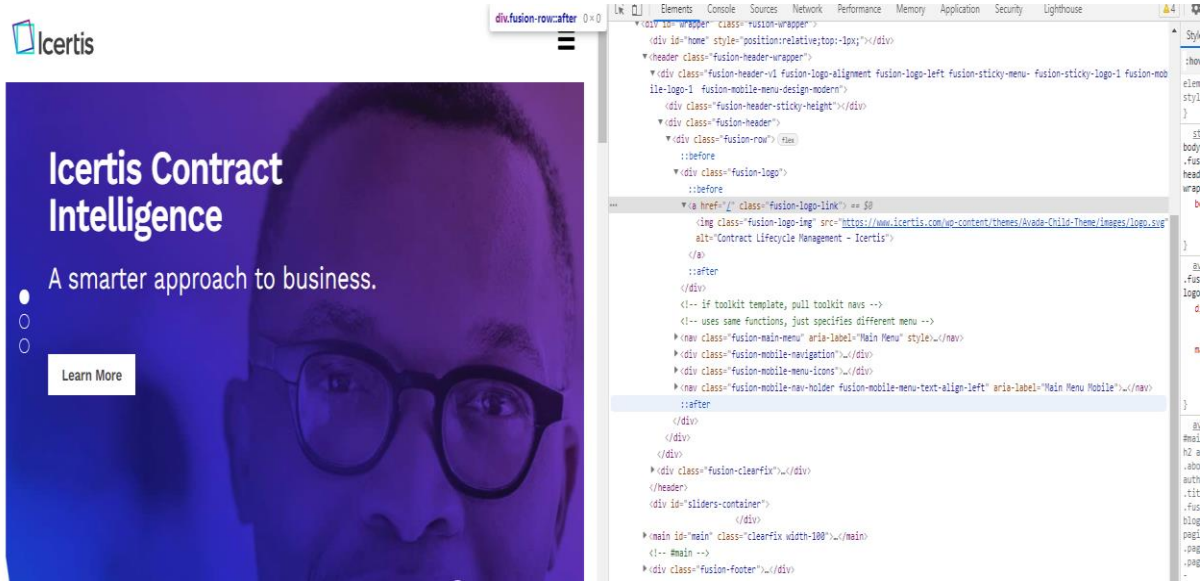


Figure 7: Icertis Homepage

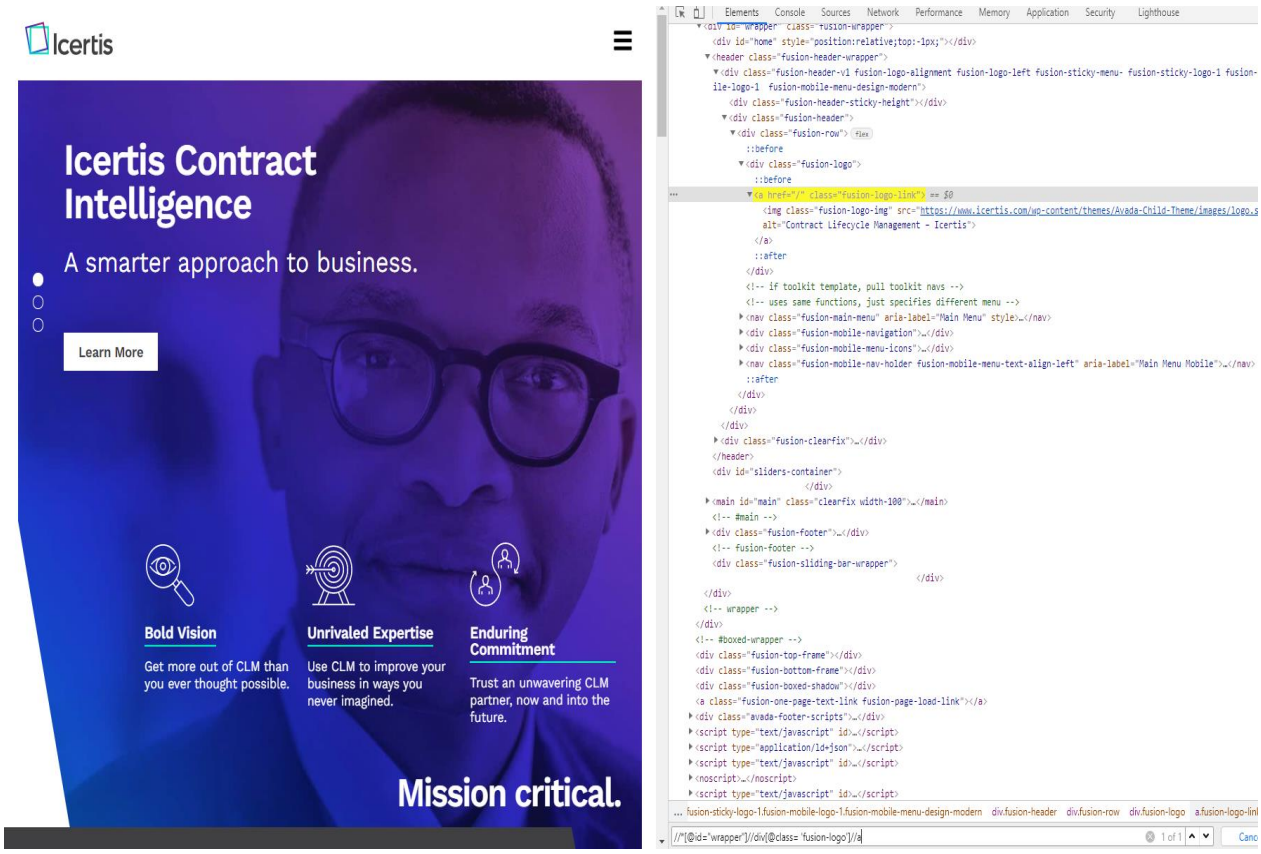
Here we try to identify the XPath of the Icertis Logo:

1. Use dev tools to look at the HTML layout
2. Write an XPath to identify the desired Web Element



XPath: //*[@id="wrapper"]//div[@class= 'fusion-logo']//a

Check if the XPath identifies the logo correctly:



The XPath identifies the Icertis Logo correctly, so this indicates how we can use XPath to identify the desired Web Elements and Automate the Test Case.

Selenium Components:

Figure 8

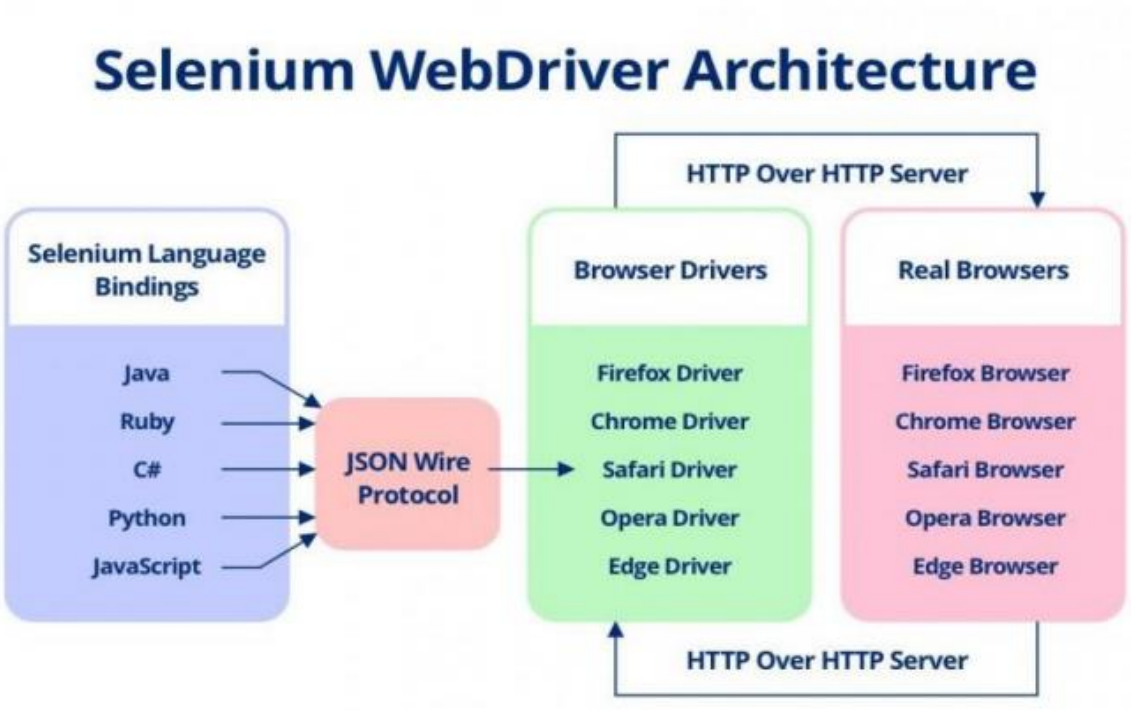


Figure 9: selenium web driver architecture

Steps of Selenium Test:

1. A WebDriver instance must be created
2. Navigate to a webpage & locate the WebElement on the WebPage
3. Action should be performed on the WebElement
4. Compare the browser response to the expected response
5. Record the test

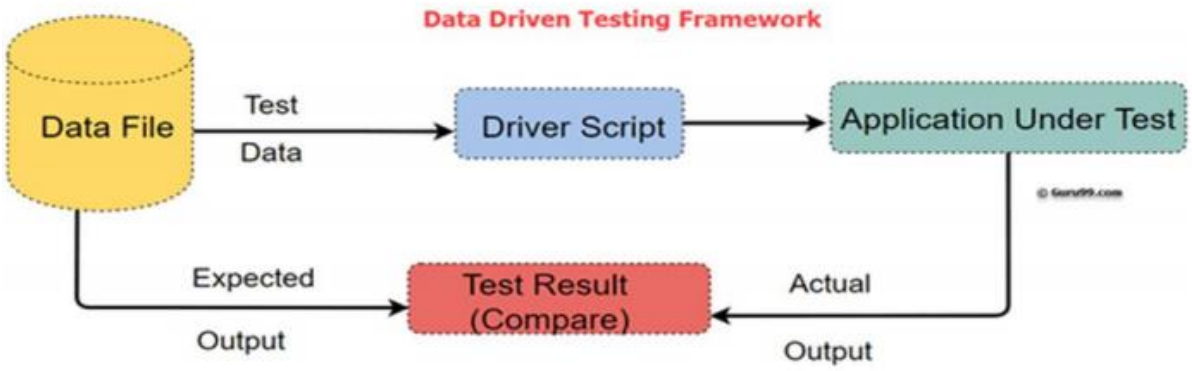


Figure 10 data driven test framework

BUG LIFECYCLE

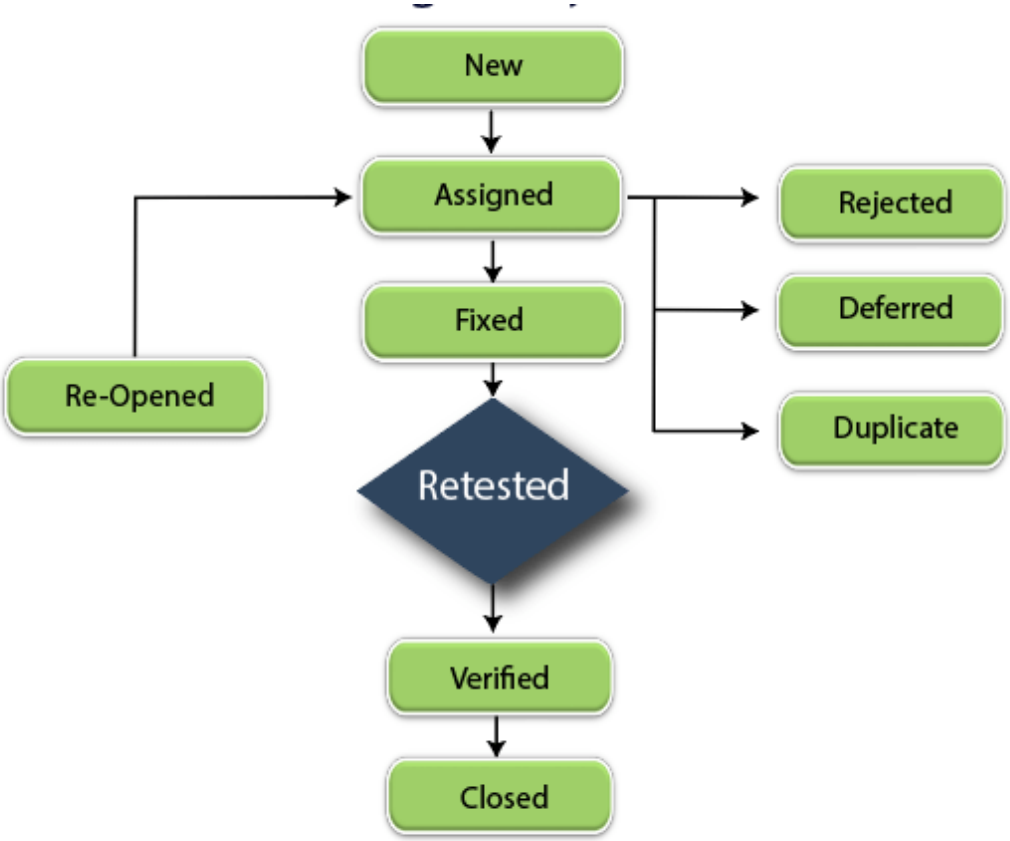


Figure 11 Bug lifecycle

New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.

• Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team

• Open: The developer starts analysing and works on the defect fix

• Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."

• Pending retest: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is "pending retest."

• Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test."

• Verified: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."

• Reopen: If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.

• Closed: If the bug is no longer exists then tester assigns the status "Closed."

• Duplicate: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."

• Rejected: If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."

• Deferred: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs

• Not a bug: If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug"

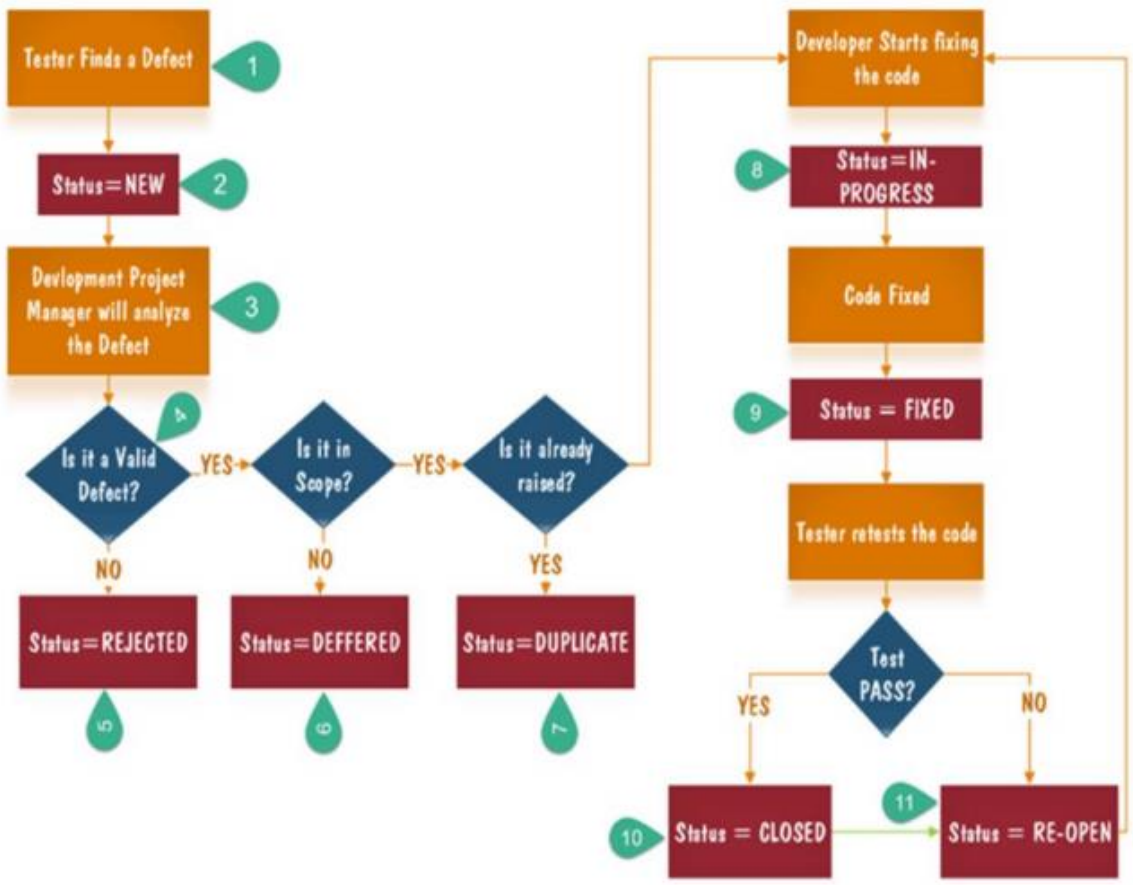


Figure 12 Bug Lifecycle -2

Chapter VI

CONCLUSION AND FUTURE WORK

It can be concluded that the internship experience as it is was an enriching learning experience which helped me gain experience in the industry. Working on a live project for the team responsible for a major upgrade of the ICI platform was an immense pleasure.

The automation tile work done in this internship will be further used the different teams across the platform to write multiple test cases it made the product for seamless experience for the end user. the automation framework would be further modified by a different team to accommodate API testing as well.

Selenium is considered the first choice for automation for web-based applications. Ever since the beginning of selling him it has been a primary choice for automation testing because of the pack popular languages it supports and being open source it has a huge developer community which makes it even better every day selenium is easy to use and makes a robust automation framework.

Throughout this internship the learning process was immense, and the aid provided was sufficient for development and automation testing. technologies like selenium, C# and .NET but used to do the required tasks. Overall, the learning experience was immensely yielding, and the learning gathered from the internship was extremely beneficial.

REFERENCES

[1]https://www.eduramp.in/

[2] https://www.udemy.com/

[3] https://www.geeksforgeeks.com

[4] https://www.icertis.com/

InternshipReportIcertis.pdf

by

Submission date: 24-May-2021 06:10PM (UTC+0530)
Submission ID: 1593111276
File name: InternshipReportIcertis.pdf (1.23M)
Word count: 3162
Character count: 16429

END TERM MAJOR PROJECT REPORT
(1st February 2021 – 2nd July 2021)
On

Internship experience as a Platform Engineer at Icertis

Under the guidance of

Sachin Jere, Associate Manager

Icertis Solutions

Submitted by:

Devesh Rattan Sharma

171369

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT
DISTT. SOLAN, HIMACHAL PRADESH 173234
(2017-2021)

DECLARATION

I hereby declare that the work reported in this report entitled **"Internship experience as a Platform Engineer at Icertis"** is submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science and Engineering**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work

Icertis does not allow the project material to be used beyond the said guidelines, I Devesh Rattan Sharma am aware of the fact and appropriately created the project report without violating any compliance.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Devesh Rattan Sharma
Enrolment No.: 171380
B. Tech (CSE)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Sachin Jere
Associate Manager
Icertis

ACKNOWLEDGEMENT

It was a privilege for me to work as a full-time intern at **"ICERTIS SOLUTIONS"** under the supervision of Mr. Sachin Jere.

This report describes the training that underwent, for the duration of 28 February 2021 - 25th June 2021 at intern at "**ICERTIS SOLUTIONS**" a world leader in providing a Contract Lifecycle Management (SaaS) Solution.

I would like to express our sincere gratitude of the all the people who have helped and supported me throughout. I am deeply indebted to Mr Sachin Jere (Associate Manager), Swapnil Mali (Senior Software Engineer) and other fellow colleagues at **ICERTIS SOLUTIONS** for organizing an effortless internship program, efficiently and providing me valuable resources and for their cooperation and willingness to share their expertise and knowledge and to devote their precious time to discuss related topics.

The help and co-operation extended by the staff at **ICERTIS SOLUTIONS** is fully acknowledged. I thoroughly enjoyed my entire internship program and would like to thank everyone at **ICERTIS SOLUTIONS** for their guidance and support.

TABLE OF CONTENTS

ABSTRACT

This report describes the internship that I did with "**ICERTIS SOLUTIONS**", Pune Office, during the period of February to June 2021. The project assigned to me was "**Internship experience as a Platform Engineer at Icertis**". The report itself sheds light on the various process and concepts I have learned as a full-time intern.

The Work from home internship started on 1st February 2021. I was assigned with the Platform Engineering department. Initially the PE interns were given a basic training around the workflow and security awareness. The second training that the interns received was a basic overview of the ICI platform, where we got a hands-on experience to use the platform, we created contracts, agreements, etc, and got a sense of the Platform.

Then to sharpen our technical skills we were given a C# and .NET training by an external company **EduRamp** which is specialised to give this training to B2B clients. The training helped us understand the basics of C# and .NET, this made us ready for our next endeavour working for a functional team at Icertis.

Working with a functional team was the real experience that we as interns were excited about. I was assigned with a team responsible to upgrade the ICI platform and introduce new features in the platform, this was a major upgrade as the UI was to be updated. These new changes meant that the UI automation framework also needed an update, so that automated test cases could be written for the new platform. I was assigned a part of this task to update the framework and automate few test cases that our team was working on. It was a great learning experience as I could use my newly acquired skills in the real world.

Chapter I

INTRODUCTION

1.1 About Icertis

Icertis is World leader in Contract Lifecycle Management (CLM) with its ICM platform, Icertis platform manages more than 10 million contracts, in 90 different countries and 40+ supported languages. Icertis has offices in 12 countries, with Pune being the biggest hub, catering more than a 1000 Icertians. With a strong business outlook, Icertis is an employee centric company with strong foundations of FORTE values instilled in the very roots of the organization.

1.2 Internship Program

The learning journey contains two stages, first stage consisted of a basic company and product overview, and in the second stage the interns were put in a functional team to work alongside experienced engineers to get a real hands-on experience.

- Company and product overview
- C# and .NET basics
- Selenium and OpenXML

STAGE 2 – Working with a Functional team

- Worked with a team responsible for the upgrade of the ICI Platform

1.3 Software Testing and Development

In the information age, all organizations depend intensely on data for everyday use, except data itself, needs to investigate to be followed up on by human entertainers. This is the place where programming comes in software helps in everything from ticket booking to fragile operation, virtual products have given clients to do things effectively and save time.

Software itself is an immense subject and software improvement and testing is at the core of each software organization. Software advancement alludes to the cycles and models that help in making of the software, these cycles and models give a through comprehension of what can anyone do, how it ought to be finished. Software testing is huge and complex field of testing software applications.

Automation of testing system helps the software advancement firms and business to reduce on expense and settle on successful choices and improvement is simpler and stronger. Web applications are the ideal possibility for robotization as they need to serve a wide range of sorts of programs just as various customers also. Consequently, Automated testing of sites is gigantic expense saving procedure for organizations.
An automated site sees the fascinating necessities of visitors and updates content for each watcher. The item helps with supporting help contacts even more beneficially. It is responsive (adaptable pleasing) out-of-the-case, totally joined with each exhibiting channel, and flexible to the changing necessities of leads and customers.

C# and .NET is characterized by a determination and comprises of a programming language, a compiler, center libraries and a runtime (CLR). .NET framework is used to compile C# code to make it machine independent. The C# code is compiled to intermediate language (IL) which is according to the specifications of the common language infrastructure (CLI). This code is run on the CLR provided by the .NET framework which makes the code truly machine independent.

Selenium is an open source instrument that robotizes internet browsers. Selenium is Open Source UI Automation Tool. It is utilized to Automate the Web-Application however not for Desk-Top

Base Application Automation. It was made in 2004 by a ThoughtWorks Engineer. Selenium Web-driver was presented later in 2006

Chapter II

LITERATURE SURVEY

2.1 Difference Between C# and .NET

2.1 C# and .NET

S.No	C#	.NET
1.	It is a programming language	It is a framework on which language is built.
2	It is also created by Microsoft after C and C++.	It is created by Microsoft and is Network Enabled Technology..
3.	It is flexible and very easy to use than Java and C++	It supports many programming languages and contains libraries of the languages that they will use.
4.	It's vast features allow the developers to create a good number of projects and applications.	It's RunTime Environment is CLR, which is Common Language Runtime.

In paper [1][2][3], Developers write test cases alongside standard code in unit testing, which is a

common procedure. Automation systems like JUnit for C# have popularised this approach by allowing unit test suites to be run frequently and automatically. Regardless of how unit testing is perceived in practise, software engineering researchers see room for innovation and are investigating innovative strategies such as automatic unit-test generation. We used a global online marketing research tool to perform a survey of 225 app developers, covering various programming languages and 29-countries, in order to match such research with the needs of practitioners. The results of the survey show that unit testing is an integral part of software production and that there is a need for further research into unit testing automation. The findings assist us in identifying areas of interest for which additional testing is needed (e.g., unit test maintenance), as well as providing insight into the suitability of online marketing research tools for software's engineering survey.

Chapter III

SOFTWARE DEVELOPMENT

3.1 Software Development Lifecycle

A Software Development Lifecycle Model tells us about the types of activities executed at every stage in a software development project, and how the ventures relate to one another aptly and chronologically.

- SDLC is the elision of Software Development Life Cycle.
- It is furthermore called as Software Development Process.
- SDLC is a complex set apart assignments executed at each parade in the product up gradation measure.
- ISO/IEC 12207 is a global typical for programming wheel of life measures. It is said to be the streamer that sets every one of the assignments needed for fabricating and looking after programming.

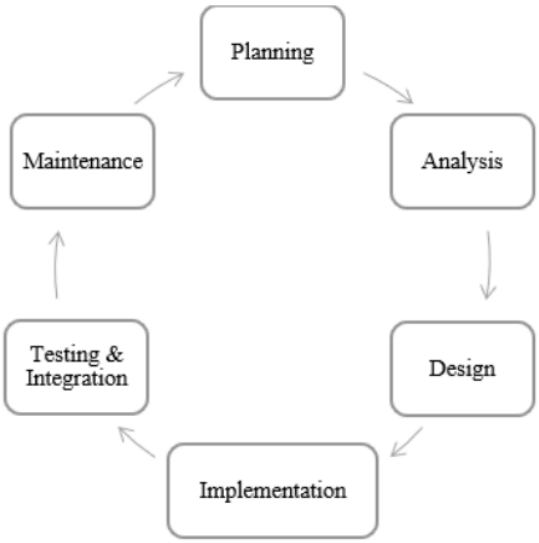


Fig 2.1 Graphical representation of the processes of SDLC

3.2 Software Development Models

Software Development is generally of two types:

1. Sequential
2. Iterative and Incremental

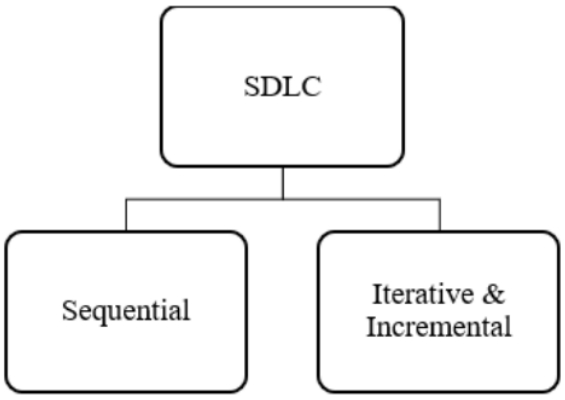


Figure 2.2 Types of SDLC

3.1.2 Sequential Software Development

A sequential development model expresses the software development process as a continuous, progressive flow of activities. This implies that any stage in the advancement, interaction should start when the past stage is finished. In principle, there is as such no cover of stages, yet by and by, it is useful to have early input from the accompanying stage.

Advantages

- Cost Effective
- Less time taking
- Suitable for different geographical positions
- It's linear
- Maximized customer satisfaction
- No pre knowledge required

Disadvantages

- Rigid
- Absence of central authority
- Lack of intuitiveness
- No centrality of the client

Waterfall Model

In the Waterfall model, the maturing ventures are completed one after another. In this model, test ventures only occur after all other maturing activities have finally been out righted.

Its diagrammatic portrayal looks like a course of cascades.

Figure 1Stages in Waterfall model

V-Model

Disparate the Waterfall model, the V-model mingles the test operations all-round the development process, executing the principle of before time testing.

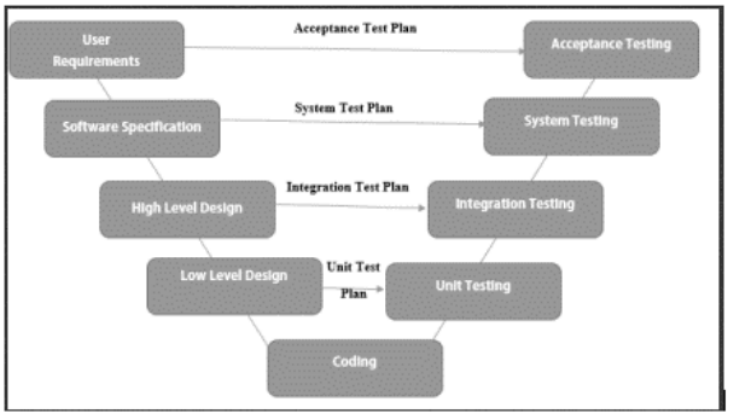


Fig 2.5 Steps Involved

Incremental and Iterative Development

Incremental Model is a cooperation of programming progression where essentials disengaged into various autonomous modules of the item improvement cycle. In this imitation, each and every module goes through the necessities, aim, execution, and preliminary stages. Each resulting coming of the section adds extent to the prior release. The cooperation gets moving until the whole skeleton achieved.

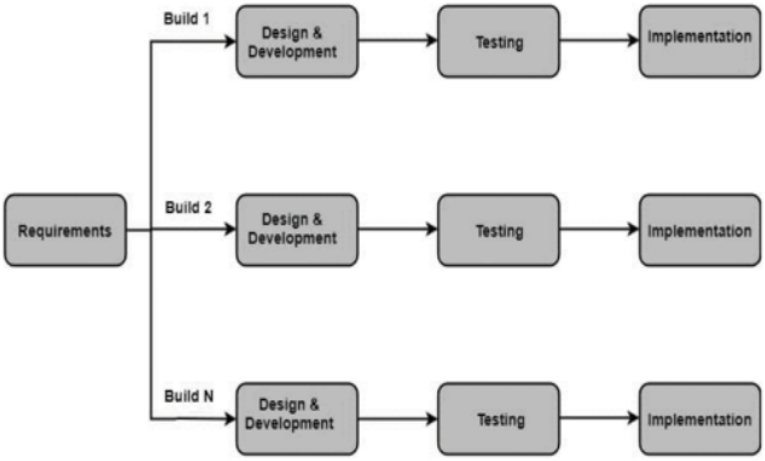


Fig 2.6 Incremental Model

In Iterative Model, you can begin with a portion of the product determinations and build up the main adaptation of the product. After the main form on the off chance that there is a need to change the product, another rendition of the product is made with another emphasis. Each arrival of the Iterative Model will result in a definite and fixed period that is called cycle.

This replica sanctions the coming by to antecedent stages, in which the diversities made discretely. The rearmost yield of the venture recharged towards the varnish of the Software Development Life Cycle measure.

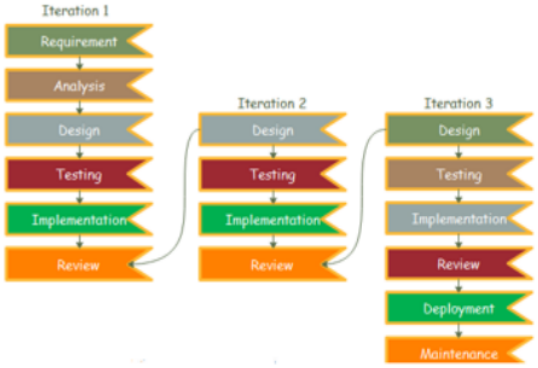


Fig 2.7 Iterative Model

3.3 Agile Methodology

Agile methodology for software development is a combination of incremental and iterative processes. With Agile the main objective of the developers is on adaptability and customer satisfaction by rapid delivery of working software products. These products are developed in rounds of weeks, which are then iterated over and over, keeping customer's priorities in mind.

Every agile iteration involves, cross functional teams working simultaneously on different areas like:

- Planning
- Requirement Analysis
- Design
- Coding/Development
- Unit Testing
- Acceptance testing

Agile model relies on the belief that every project has different requirements and has to be handled differently, therefore, the methods have to suit the project requirements.

In the earlier days of software development, there were different methodologies that were synonymous to agile, but there were few differences because each of these methodologies were developed by different groups with varying project requirements.

Popular methods like Extreme Programming (XP), Scrum, feature driven development, etc, are now collectively known as Agile Methodologies, after the agile manifesto came into existence in 2001.

Agile Values are:

- **Working Software:** A working prototype of the software is considered to be the best means of communication between the developers and customers as it allows customers to see the product and request features, while developers can observe the different interactions
- **Responsive:** The product has to responsive to changing customer needs, and teams should be able to quickly develop the feature requested.
- **Customer Interaction:** As the requirements cannot be gathered completely during the initialisation of the project and due to requirement changes and feature requests it is considered that customer should be involved in the product to get proper requirements and criteria for the product.
- **Individuals and Interaction:** The agile methodologies tells that interactions between individuals is a good way to develop software, which allows increased motivation, self-organization and tools like pair programming helps in developing a quality product.

3.4 Agile Principles

Agile principles are the guiding rules/practices that helps in implementation and execution with agility

- The highest priority is customer satisfaction by early and continuous delivery of valuable software.
- Agile process allows changes in products at every stage. Even in late stage of development changes by customers should be entertained by the development teams.
- Delivery of working software is of utmost important; the frequency could be a few weeks or months, but the shorter time period is preferred.
- Business representative and developers should work together as much as possible to ensure a product that fulfils customers' requirements.
- Projects should be developed by enthusiastic individuals who are motivated, and they should be provide with proper environment to harness their best.
- The most efficient way of communication between teams is through face to face interactions.
- Progress is measure by deliveries of working software
- Agile processes depend upon sustainable development, therefore the stakeholders involved in the project should be able to maintain a constant pace.
- Attention of good programming practices and good architecture enhances agility
- Agile process promotes simplicity to maximize the amount of work not to be done.
- The best architectures, designs and requirements are developed by self-organizing teams
- At subjected intervals, the development team reflect on the efficiency of development to become more effective accordingly

3.5 Software Testing

Software testing can be called as an affair which comprises several distinct activities or jobs. Of which execution is only one of these jobs

Software testing does the following:

- gauges the quality of the software
- risk of software failure in operation is reduced

Software Testing ≠ Test Execution

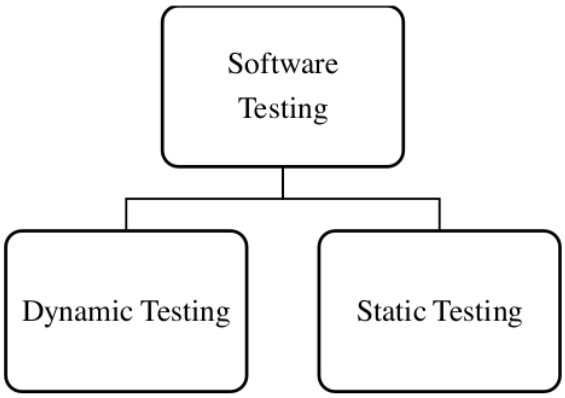


Figure 1 Types of Software Testing

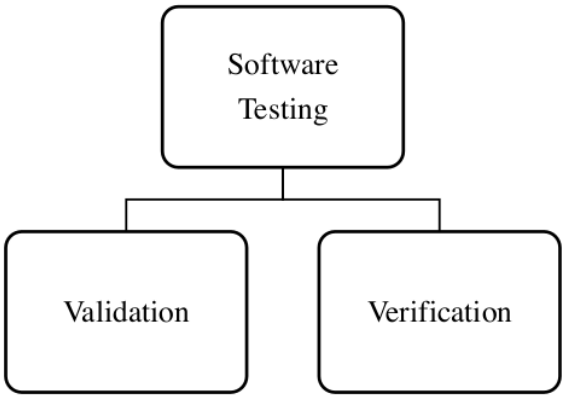


Figure 2Steps in Software Testing

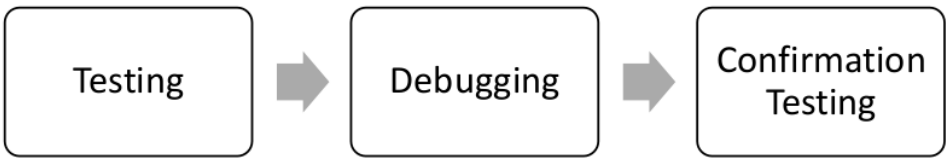


Figure 3Relationship between Testing and debugging

3.6 Testing Processes

There is nobody all-inclusive programming test measure, yet there are basic arrangements of test exercises without which testing will be more averse to accomplish its set up destinations.

Figure 4.11 Test Process

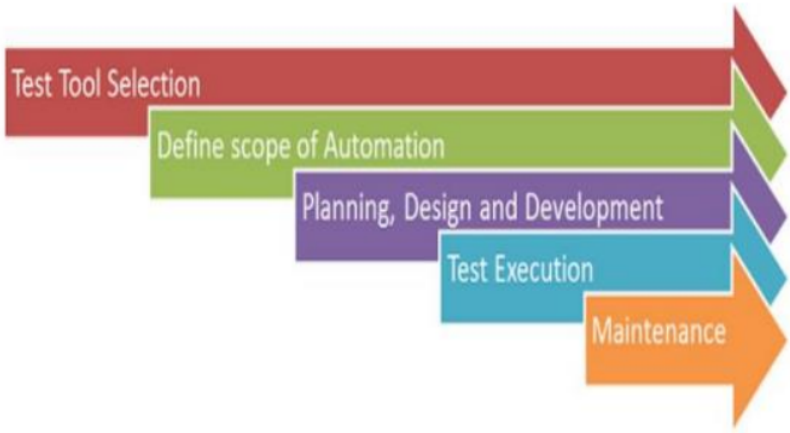


Figure 5.13 Test Process

3.7 Test Levels

Test levels are gatherings of test exercises that are coordinated and overseen together. Each test level is an occasion of the test interaction. Test levels are identified with different exercises inside the product improvement lifecycle.

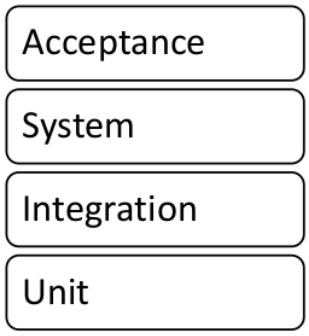


Figure 6Test Levels

3.8 Test case writing

A bunch of preconditions, inputs, activities (where relevant), anticipated outcomes and post conditions, created dependent on test conditions.

It involves:

- Test Case Title
- Pre-Conditions
- Test Steps
- Expected Results
- Test Suite
- Test Environment
- Actual Results
- Status
- Defect Report

3.9 Defect Report

Defect Report - Documentation of the event, nature, and status of a deformity. Wherein a blemish or inadequacy in a work item where it doesn't meet its prerequisites or particulars is a deformity.

Chapter IV

SYSTEM DEVELOPMENT

4.1 Programming Languages

4.1.1 C# or C Sharp

C# is articulated "C-Sharp".

It is an item arranged programming language made by Microsoft that sudden spikes in demand for the .NET Framework.

C# has roots from the C family, and the language is near other mainstream dialects like C++ and Java.

The principal rendition was delivered in year 2002. The most recent form, C# 8, was delivered in September 2019. C# is utilized for:

- Portable applications
- Work area applications
- Web applications
- Web administrations
- Sites
- Games
- VR
- Data set applications
- Also, a whole lot more!

Why is C # used?

It is one of the most widely used languages in the world. It is simple to learn and use. It has a lot of love from the media. C # is a target language for systems that offers transparency and encourages code reuse, lowering implementation costs. Since C # is more closely related to C, C ++, and Java, it is simpler for the framework to move from C to C # and vice versa.

4.2.1 .NET Framework

It is a software framework that was designed and developed by the company. The first version of the.Net platform, version 1.0, was released in 2002. In simple terms, it is a 20virtual machine for compiling and running programmes written in various languages such as C#, VB.Net, and others.

Form-based software, Web-based applications, and Web services are all built with it. On the, you can choose from a number of programming languages. The most common versions of the Net framework are VB.Net and C-sharp. It's used in building apps, among other platforms. It has a variety of features and even adheres to industry norms.

Basic Component Stack and Architecture of .NET

The fundamental architecture of is made up of the first three components from the bottom. After that, Microsoft added more modules to the Net platform, which was released in 2005. The following is the Net Framework:

Kinds of Applications: Mainly the applications which are underlying .Net system is partitioned into the accompanying three classifications:

❑ WinForms: Form – Based applications are considered under this classification. In straightforward terms, we can say user-based applications goes under this WinForms class.

❑ ASP.NET: ASP.Net is a structure for web and it gives the amazing coordination of HTML, CSS and JavaScript which makes it valuable to foster the web applications, sites and web administrations.

❑ ADO .NET: It includes applications that operate with data like MS SQL Server, Oracle and so forth comes. It predominantly comprises of classes that can be utilized to interface, recover, embed and erase information.

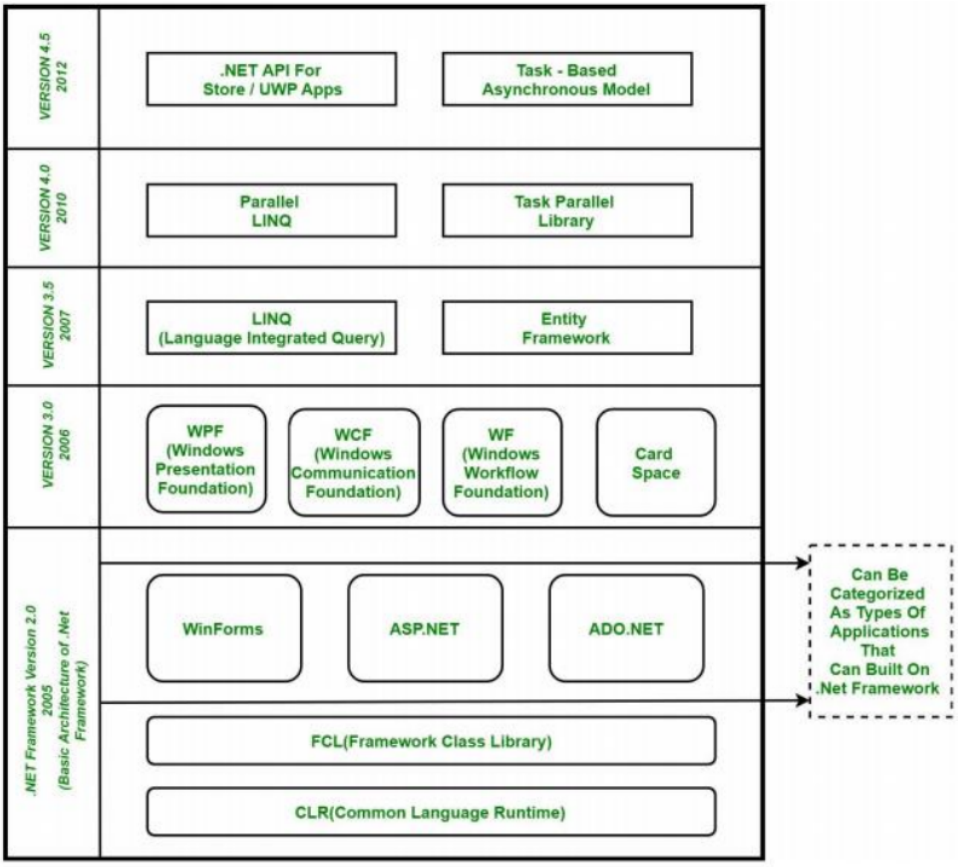


Fig 3.6 Architecture of .NET

Chapter V

UI AUTOMATION

UI automation testing process is like manual testing process, but the user doesn't have to click the automation does it for him, through the application, and visually verify the data. Each test case can be scripted in XML script and executed to generate a test result.

Selenium

Selenium is a free (open source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done utilizing the Selenium testing device is generally alluded to as Selenium Testing.

Selenium Software isn't only a solitary instrument yet a set-up of programming, each piece obliging distinctive Selenium QA testing needs of an association. Here is the rundown of apparatuses

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

XPath: XML path is a way to identify a WebElement in a HTML page. XPath provides various expression to matching expressions

Relative XPath: This involves searching the complete document using specific attributes, text or using relationship to possibly search unique text.

Absolute XPath: In this method the complete tags are provided from root to descendants with precision and without missing any of them.

XPath Functions:

1. Contains
2. Starts-with ()
3. Text ()
4. AND Operator
5. OR Operator

Example of an XPath in a Web Page:

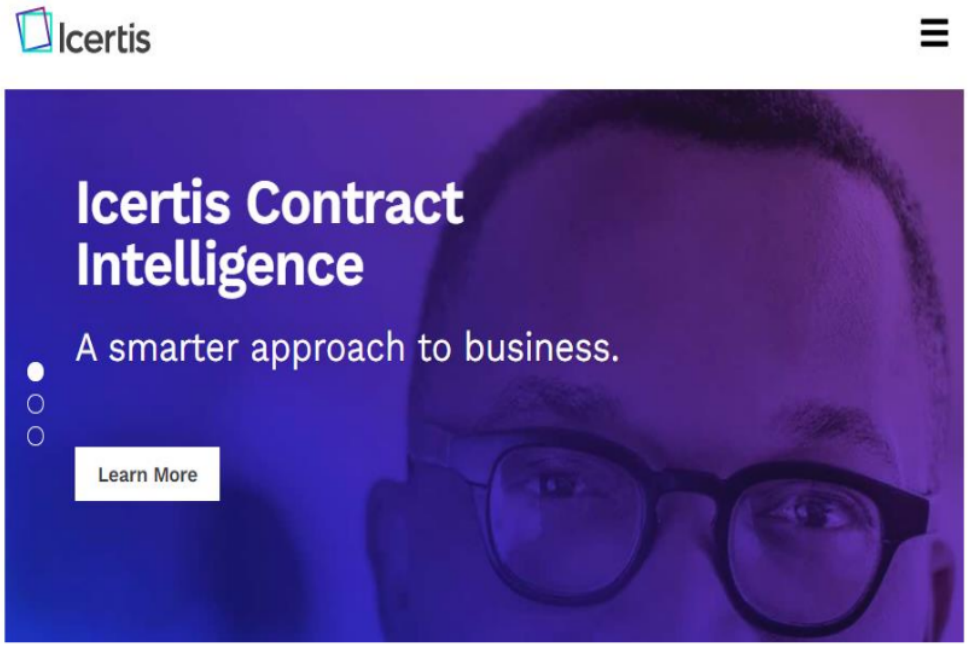
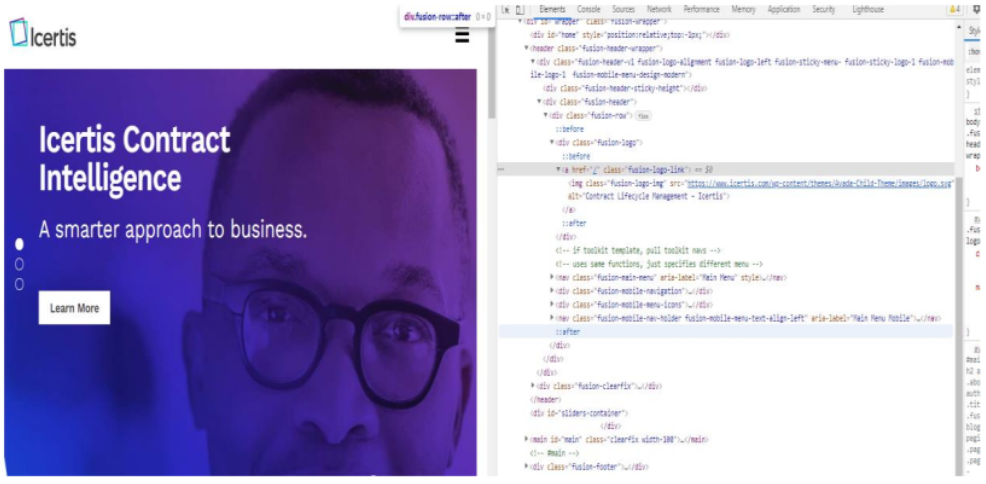


Figure 7: Icertis Homepage

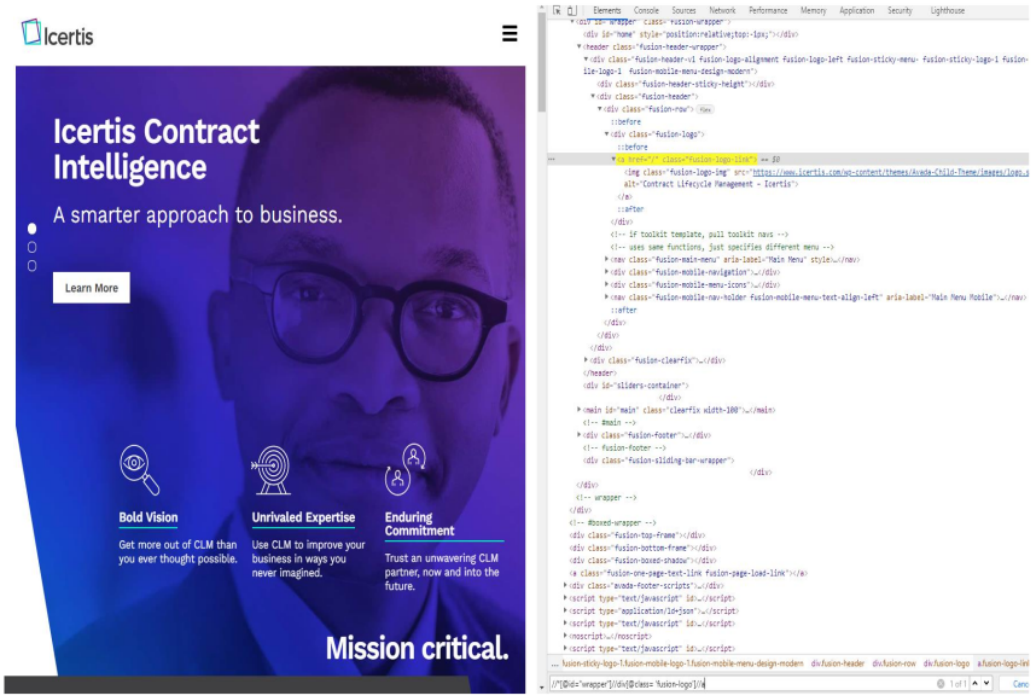
Here we try to identify the XPath of the Icertis Logo:

1. Use dev tools to look at the HTML layout
2. Write an XPath to identify the desired Web Element



XPath: //*[@id="wrapper"]//div[@class= 'fusion-logo']//a

Check if the XPath identifies the logo correctly:



The XPath identifies the Icertis Logo correctly, so this indicates how we can use XPath to identify the desired Web Elements and Automate the Test Case.

Selenium Components:

Figure 8

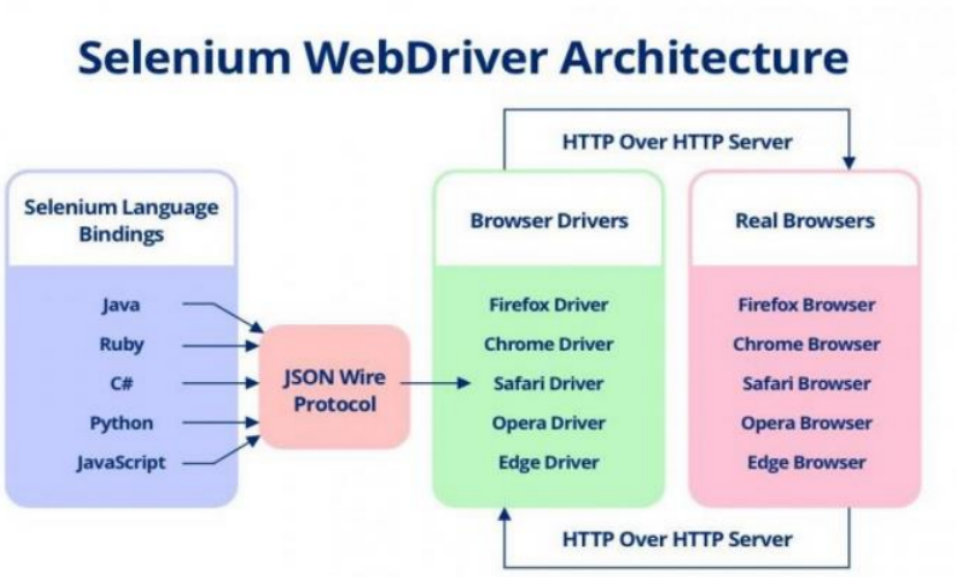


Figure 9: selenium web driver architecture

Steps of Selenium Test:

1. A WebDriver instance must be created
2. Navigate to a webpage & locate the WebElement on the WebPage
3. Action should be performed on the WebElement
4. Compare the browser response to the expected response
5. Record the test

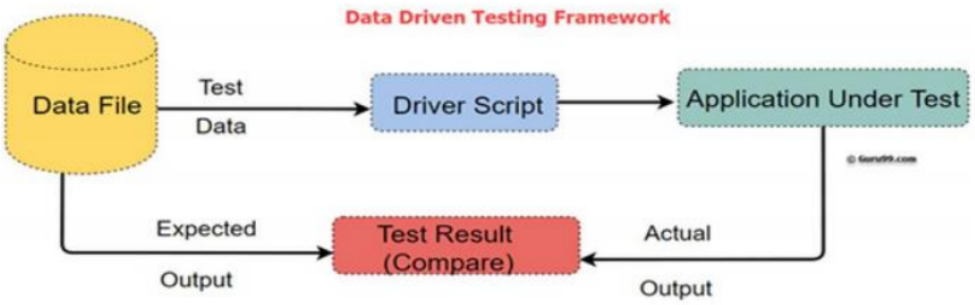


Figure 10 data driven test framework

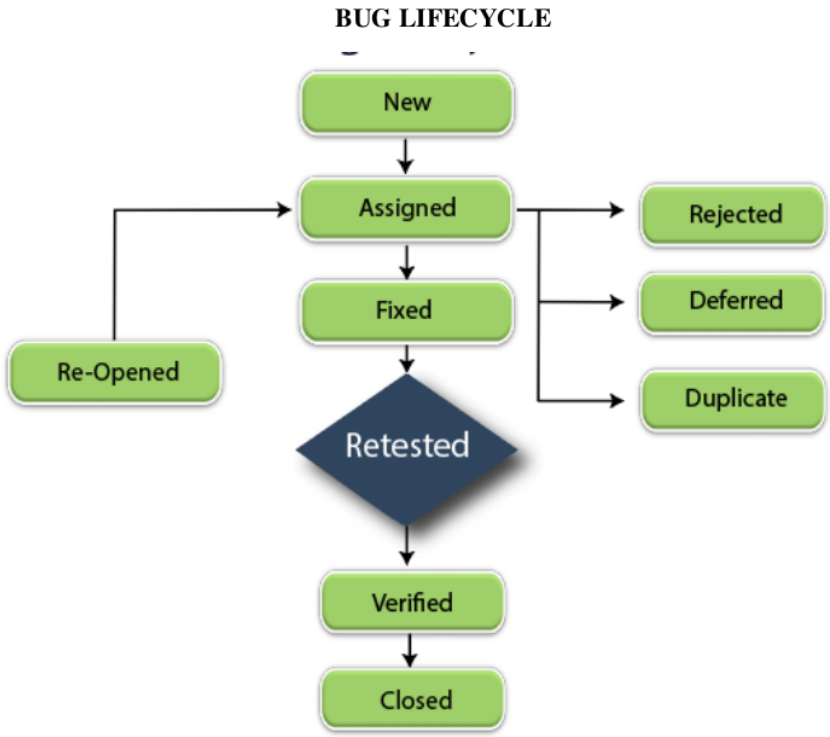


Figure 11 Bug lifecycle

New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.

• Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team

• Open: The developer starts analysing and works on the defect fix

• Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."

• Pending retest: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is "pending retest."

• Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test."

• Verified: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is "verified."

• Reopen: If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.

• Closed: If the bug is no longer exists then tester assigns the status "Closed."

• Duplicate: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."

• Rejected: If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."

• Deferred: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs

• Not a bug: If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug"

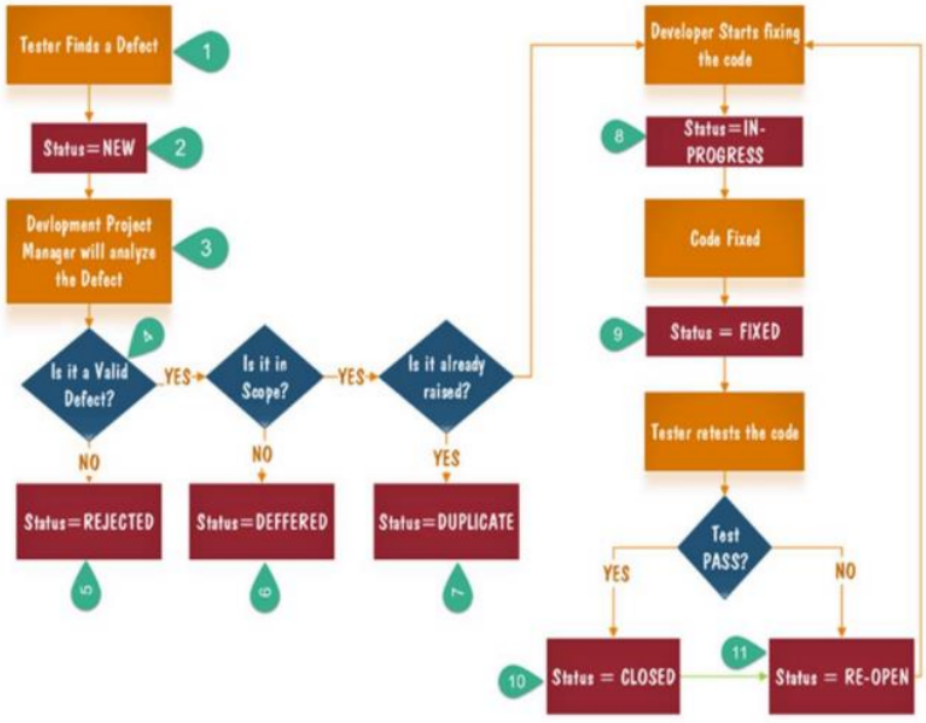


Figure 12 Bug Lifecycle -2

Chapter VI

CONCLUSION AND FUTURE WORK

It can be concluded that the internship experience as it is was an enriching learning experience which helped me gain experience in the industry. Working on a live project for the team responsible for a major upgrade of the ICI platform was an immense pleasure.

The automation tile work done in this internship will be further used the different teams across the platform to write multiple test cases it made the product for seamless experience for the end user. the automation framework would be further modified by a different team to accommodate API testing as well.

Selenium is considered the first choice for automation for web-based applications. Ever since the beginning of selling him it has been a primary choice for automation testing because of the pack popular languages it supports and being open source it has a huge developer community which makes it even better every day selenium is easy to use and makes a robust automation framework.

Throughout this internship the learning process was immense, and the aid provided was sufficient for development and automation testing. technologies like selenium, C# and .NET but used to do the required tasks. Overall, the learning experience was immensely yielding, and the learning gathered from the internship was extremely beneficial.

REFERENCES

[1]https://www.eduramp.in/

[2] https://www.udemy.com/

[3] https://www.geeksforgeeks.com

[4] https://www.icertis.com/

InternshipReportIcertis.pdf

ORIGINALITY REPORT

23%	20%	2%	20%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.guru99.com Internet Source	12%
2	www.coursehero.com Internet Source	2%
3	digitallibrary.ust.edu.ph Internet Source	2%
4	www.ir.juit.ac.in:8080 Internet Source	2%
5	Submitted to Malaviya National Institute of Technology Student Paper	1%
6	Submitted to northcap Student Paper	1%
7	Ermira Daka, Gordon Fraser. "A Survey on Unit Testing Practices and Problems", 2014 IEEE 25th International Symposium on Software Reliability Engineering, 2014 Publication	1%
8	Submitted to Harare Institute of Technology Student Paper	1%

9	Submitted to Segi University College Student Paper	1%
10	Submitted to auf Student Paper	1%
11	Submitted to University of Sheffield Student Paper	1%
12	medcraveonline.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 17/06/2021

Type of Document (Tick): ~~PhD Thesis~~ ~~M.Tech Dissertation/ Report~~ B.Tech Project Report ✓ ~~Paper~~

Name: Devesh Rattan Sharma Department: CSE Enrolment No 171369

Contact No. +91 82194 80488 E-mail. deveshrattansharma@gmail.com

Name of the Supervisor: Dr. Himanshu Jindal

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters):

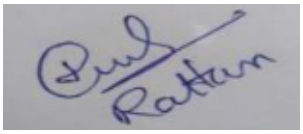
END TERM MAJOR PROJECT REPORT ON
INTERNSHIP EXPERIENCE AS A PLATFORM ENGINEER AT ICERTIS

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 32
- Total No. of Preliminary pages = 4
- Total No. of pages accommodate bibliography/references = 1



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ...23.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Himanshu Jindal

(Signature of Guide/Supervisor) Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String	23%	Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com