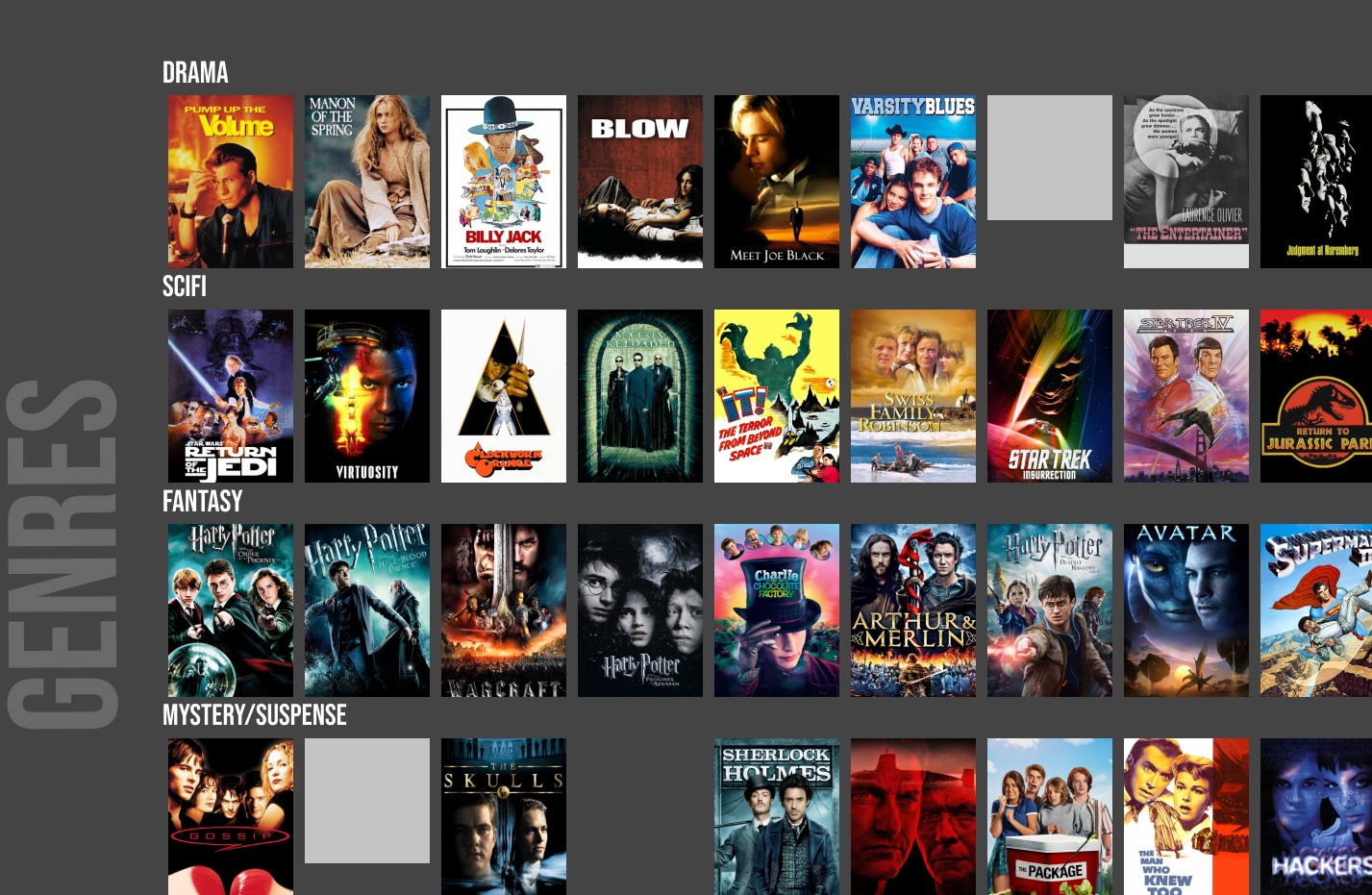
THE MISSSPRINTS

SPRINT 1



TESTING DOCUMENT

RAPID APP DEVELOPMENT

|  |  |  |
| --- | --- | --- |
| Catherine Burns | Benjamin Royans | Jai Ananda |

Contents

[Document Revision History 3](#_Toc42525681)

[Introduction 4](#_Toc42525682)

[Rapid Application Development (RAD) and Workflow Diagram 4](#_Toc42525683)

[Scrum Testing Strategy 5](#_Toc42525684)

[Scrum Attributes 5](#_Toc42525685)

[Sprint Requirements 5](#_Toc42525686)

[Scope 6](#_Toc42525687)

[In Scope 6](#_Toc42525688)

[Out of Scope 6](#_Toc42525689)

[Quality Objective 7](#_Toc42525690)

[Roles and responsibilities of each team member 7](#_Toc42525691)

[Scrum Master: Benjamin Royans 7](#_Toc42525692)

[Dvelopement Team : Catherine Burns, Jai Ananda 7](#_Toc42525693)

[Test Methodology 8](#_Toc42525694)

[SCRUM 8](#_Toc42525695)

[What is SCRUM Methodology? 8](#_Toc42525696)

[Why SCRUM? 8](#_Toc42525697)

[Test Plan 8](#_Toc42525698)

[Test Levels 8](#_Toc42525699)

[Test Levels define the Types of Testing to be executed on the Application Under Test (AUT). 8](#_Toc42525700)

[Constraints: 9](#_Toc42525701)

[Test Environment 9](#_Toc42525702)

[INTEGRATION TESTING 1](#_Toc42525703)

[CONSISTENCY VALIDATION REPORT 1](#_Toc42525704)

[REQUIREMENT TRACEABILITY MATRIX 1](#_Toc42525705)

[TEST METRICS 3](#_Toc42525706)

[BUG TRIAGE 4](#_Toc42525707)

[Suspension Criteria and Resumption Requirements 1](#_Toc42525708)

[Test Completeness Check 1](#_Toc42525709)

[Resource & Environment Needs 1](#_Toc42525710)

[Testing Tools 1](#_Toc42525711)

[Terms/Acronyms 1](#_Toc42525712)

# **Document Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Date** | **Change History** | **Version** | **Associate** |
| 1 | 5/06/2020 | Created the document | 1.0 | Jai Ananda (30007873) |
| 2 | 8/06/2020 | Updated Document for Sprint 2 | 1.1 | Ben Royans |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

### Rapid Application Development (RAD) and Workflow Diagram

**RAD** is an effective software development standard. Under circumstances where the initial requirements are not clear or where requiements changes frequently takes place during the development, RAD provides a systematic software system that has the ability to be automated.

Sufficient sotware testing is required when producing software application of high quality. Classical testing methods are not ideal for iterative development like RAD. Hence, there is a need for tailored testing method for this development standard.

**Assumptions:** To adopt a testing stratgey we shall assume that RAD is

* Iterative
  + includes Sprints
* Is evolutionary
  + continuous development of the application with improvements
* Contains the familiar technical RAD language
* Library facilities and data retrieval capabilities
* Implemented code is from a high level language
* Support environment is sophisticated

### Scrum Testing Strategy

The following tests are performed on the software to ensure the application stands up to the QA standards of CITE :

* Complexity check
* Quality Testing
* Performace Testing
* Usability Testing
* Unit Testing

### Scrum Attributes

**Project Goals** includes basic idea and the requirement of the application with a focus on whether the application satisfies c;iets basic requirement or not. It involves all planning and requirement analysis.

**Product Backlog**  is a collection of user experiences observed. The product owner prepares a backlog and maintains it. It is owned by the product owner and anyone can add to the product backlog with product owners permission.

**Sprint Backlog** is a collection of user experineces to be completed in a sprint. Work is not asssigned here. This backlog is owned an dmanaged by the team and remaining work is updated regularly(ideally daily). This collection is to be performed in the sprint

### Sprint Requirements

#### Sprint 1

The team is advised to be working to fulfill the following requirements

1. Souce Control
2. Project Management Plan
3. Software Development Testing Plan
   1. This plan shall include validation plan used during software development
4. Analysis Documentation
   1. CITE business rules for software development,
   2. CITE Managed Services Quality Assurance,
   3. Acme Entertainment Pty Ltd development requirements.

#### Sprint 2

## Scope

### In Scope

CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. CITE has sought to employ The MisSprints developers to implement a Movie Database for Acme Entertainment Pty Ltd. The directive is to adopt sprint workflow to develop each set of user requirements.

This project will be overlooked by Mr Will E Coyote, Product Owner and liaison Consultant for CITE Managed Services.

The developer team has an established Scrum Master who shall allocate work the team memembers that are wihtin the scope of the first sprint.

Features of the application:

* The SMT Movie Rental provides 2300 movies for renting to the customers
* The prototype is in a visual format with navigation hierarchy established
* The application can be navigated to search for a movie record using movie Title, Genre, Rating and/or year of release
* The application also gives information on Top 10 of most searched movies

### Out of Scope

1. Secure login
2. Authorization: In case of lost credentials, there is no secured way of accessing the customer file to reset the confidential password
3. Multimedia auto loading into the thumbnails
4. User reviews

## Quality Objective

Reference: ISO 9001:2005: International standard that specifies requirements for a quality management system (QMS)

* To Establish protocl between the application, application design and development \_ what are the expectations from the website
* Empower Quality Assurance to have the application development proritize between fixing old bugs and developing new features

## Roles and responsibilities of each team member

### Scrum Master: Benjamin Royans

### Dvelopement Team : Catherine Burns, Jai Ananda

**Scrum Master** is the facilitator of the Scrum- the lightweight framework of Agile framework. Scrum master act as coaches the team.

**Test Manager** is responsible for the test effort’s success by advocating, resource planning a management and resolution of the issues that impede testing. This encompasses the following:

* Negotiate purpose and deliverables of testing
* Planning and management of test resources
* Assessing testing progress
* Advocating the quality of testing by quantifying defects
* Advocate appropriate level of testability in the software development process
* **Developers** is responsible for the following
* Creating and implementing the source code for new applications
* Testing and debugging the source code
* Evaluate applications and update with suitable modifications
* Develop technical handbook to represent design code
* Create digital software interface that entices the user into a seamless interaction between the client and the application

# Test Methodology

## SCRUM

**Scrum** is an agile project management framework. In this methodology the decision making is entirely in the hand of the teams and this can be applied to project of any type. This is most suited for the projects that are rapidly changing/developing or has high emergent requirements.

## What is SCRUM Methodology?

SCRM begins with a brief planning, meeting and conclusion with a final review. This includes series of iterations called “Sprints” to create/develop a software application rapidly. Listed advantages being:

* Successful development does not require significant requirement analysis and documentation
* Progress is frequently updated results in visible project development
* Lightly controlled method
* Daily stand up ensures individual productivity is measured, favoring continuous improvemnet of the developers.

## Why SCRUM?

SCRUM development and testing has following advantages:

* Separation of components of the app development and distribution of expertise
* Use of Prints and product backlog increases transparency between the client supervisor, teams and management
* Software features are easier to plan
* Testing quality is high and is done much earlier in the process
* Testing criteria is an integral part of the project definition
* QA is a joint reponsibility of the whole team
* Synchronization between the team is significantly improved

# Test Plan

## Test Levels

### Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

1. Integration Testing
2. Testing by combining
3. Test for data flow
4. System Testing
5. Check for compliance as per requirements
6. Over interaction of components (Load, performance, reliability and security testing)
7. Optional check for non-functional requirements testing – on demand
8. Acceptance Testing
9. Check against requirement specification at deployment/delivery
10. Involve customer and other stockholders to run this testing process

## Constraints:

**Time constraint :** Delivery date for Sprint 1 presentation to the client scheduled for June 5th 2020

## Test Environment

It mentions the minimum **hardware** requirements that will be used to test the Application.

|  |  |
| --- | --- |
| Memory: 16GB | **Processor:** i7 10th Gen Intel 10950, 2.4GHz |

Following **software’s** are required in addition to client-specific software.

1. Windows 10, iOS
2. MS Office
3. VS Code
4. NotePad++
5. Google Chrome
6. MS Edge
7. Mozilla Firefox
8. MS Excel

# INTEGRATION TESTING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Use case | Steps | Test Results | Test Status |
|  | Create DB, Tables and data | 1. Navigate to the URL | DB, Table, Constraints and data created successfully |  |
|  | Dashboard | 1. Click on ‘Index’ references 2. Click on ‘Add’ references 3. Click on ‘Search’ references 4. Click on ‘Find All’ references 5. Click on ‘Most Searched’ references 6. Click on ‘Chart’ references 7. Click on ‘Update’ references 8. Click on ‘Delete’ references | 1. Navigates to index.php successfully 2. Navigates to add.php successfully 3. Navigates to search.php successfully 4. Navigates to findall.php successfully 5. Navigates to mostsearched.php successfully 6. Navigates to chart.php successfully 7. Navigates to update.php successfully 8. Navigates to delete.php successfully |  |
|  | Add | 1. Add a movie title 2. Add Studio 3. Add status 4. Add sound 5. Add version 6. Add rectecprice 7. Add rating 8. Add year 9. Add Genre 10. Add Aspect | Navigates to add.php successfully  A validation message is returned to the UI on successful addition. |  |
|  | Search | Search by:   1. Movie title 2. Rating 3. Year 4. Genre | Results are found based on logical AND of the attributes if they are patterns of their respective fields or properties.  The search results are inserted into the table movie\_searched if the corresponding movies are not in the table or else, the count or occurrences is incremented that correspond to the movies from the results.  A validation message is returned when no search criteria is entered. |  |
|  | Find All | Click on ‘Find All’ references | All the results from the table ‘movie\_dvd’ are retrieved. These results are paginated to show 20 results per page. |  |
|  | Most Searched | Click on ‘Most Searched’ references | Top 10 most searched movies are retrieved. |  |
|  | Chart | Click on ‘Chart’ references | Top 10 most searched movies are plotted as a histogram along with the movie names and their occurrences.  This histogram is scaled and normalized and hence, it adjusts itself depending on if the occurrences are means or extemes.  The Histogram adjusts depending on the scaling factor that user may decide to vary using the given feature named ‘Scaling factor’ and hits the ‘Submit’ button.  By default, the scale would be set to the middle of the range. |  |
|  | Update | Click on ‘Update’ references | A table displaying movie details is displayed which is paginated.  The table is also provisioned with an edit button which would let the user enter the values to update the movie details.  A success message is returned if the update is successful. |  |
|  | Delete | Click on ‘Delete’ references | A table displaying movie details is displayed which is paginated.  The table is also provisioned with a delete button which would let the user to delete the corresponding movie.  A success message is returned if the delete is successful. |  |
|  | Responsiveness Webs of Web design | Change the size of the browser | All the images on the web page are displayed properly on all the different devices and resolution.  Text and headings on the web page are properly aligned.  All Clickable links on the web page are readable and work as expected.  Scrolling of the web page works as expected.  Verify if there are input boxes and text areas to enter data then we need to make sure that the text entered is displayed properly on the web page and they are aligned as expected.  Image size, Font size and font type are consistent across all the web pages.  Verify if contents of the page are displayed consistent on all resolutions.  Verify the color changes after hover over the elements.  Verify the consistency of color combination on different resolutions.  Verify images, text, different controls are not going beyond the screen border.  Verify if there should not be any horizontal scrolling bar since everything should be fit according to the size of the screen.  Verify on rotating your mobile device, all the contents should be rotated and displayed as expected without any technical glitch.  Verify if the user able to click on clickable area.  Verify padding of elements on the edges.  Verify if enter text in input box are displayed as expected without any UI glitches. |  |

# 

# CONSISTENCY VALIDATION REPORT

|  |  |  |
| --- | --- | --- |
| PAGE | INPUT | RESPONSE |
| add.php | When all the inputs are empty | Inputs cannot be empty! |
| When one of the inputs is empty | Inputs cannot be empty! |
| When ID is duplicate | Add request denied! Duplicate value. |
| When invalid rectecprice is entered | Invalid rectec price. Please enter a valid rectec price.  The value must begin with numbers from 0 to 9.  Please enter a value that is accurate upto 2 decial places. |
| When invalid year is entered | Invalid year. Please enter a valid year. |
| When invalid ratings are entered | Invalid ratings entered. |
| When invalid status is entered | Invalid status entered. |
| When invalid genre is entered | Invalid genre entered. |
| search.php | When all the inputs are empty | Search criteria cannot be empty. |
| When one or more inputs are provided | Search results are based on logical “OR” operation between search criteria in the “WHERE” clause of the “SELECT” query |
| When partial search values are entered | Results returned would consist of the entered search criteria |
| When exact and full movie names are entered | The empty space values on the far ends of the search values are trimmed and the movie name if exists will be inserted into the table “movie\_searched” in the database |
| modifyMovie.php | When one or more input boxes are empty | Inputs cannot be empty. |
| When ID is attempted to update | The ID value is Read Only and hence cannot be overwritten |
| When invalid rectecprice is entered | Invalid rectec price. Please enter a valid rectec price.  The value must begin with numbers from 0 to 9.  Please enter a value that is accurate upto 2 decial places. |
| When invalid year is entered | Invalid year. Please enter a valid year. |
| When invalid ratings are entered | Invalid ratings entered. |
| When invalid status is entered | Invalid status entered. |
| When invalid genre is entered | Invalid genre entered. |
| topRated.php | When there are no search results | No results found. |
| When search criteria are entered | The entered values are case converted to LOWER with adjustments so that the searched values are matched with the values existing in the database to display results |
| findall.php | When no values are existing in the movie database | No results found. |

# 

# REQUIREMENT TRACEABILITY MATRIX

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| USE CASE | REQUESTOR | ACTOR | DESCRIPTION | TEST CASE ID | STATUS | ACTIVE  /INACTIVE  /CANCELLED |
| Creating Database | Client | Developer, Client | The developer will run the SQL to create  the database/table/records you used in your program |  |  |  |
| The UI access Database | Client | Developer, Client, Tester | The UI reads the Database to populate the UI with the movie list |  |  |  |
| Setting up server-side environment and database | Client | Developer | This case is executed when the developer creates user account and establishes the structure of the database |  |  |  |
| Produce prototype of UI in virtual format | Client | Developer | In this use case an interactive UI that is visually complimenting client business domain is incorporated |  |  |  |
| Implementing CSS and attach header and footer | Client | Developer | Styling is done in this use case that meets the client needs |  |  |  |
| Flexibility to view all the entries and ADD new record | Client | Client, Tester, Developer | A new movie entry should be added to the database |  |  |  |
| Search for a record | Client | Client, Tester, Developer | Searches the movie record and displays it based on title, genre, rating and displays full details |  |  |  |
| Generate chart of top searched movies | Client | Client, Tester, Developer | A bar chart is generated to display the search records and number of hits |  |  |  |
| Updating Record | Client | Client, Tester, Developer | This case is for manual updating record |  |  |  |
| Delete Record | Client | Client, Tester, Developer | This case is for manual deleting record |  |  |  |
| Check for browser compatibility | Client | Client, Tester, Developer | The website should work with the UI effectively on all browsers |  |  |  |
| Reactiveness of the Application | Client | Client, Tester, Developer | The UI should scale to the sizing on various display screen sizes |  |  |  |
| Adaptability Check | Client | Client, Tester, Developer | The Ui should adapt to be flexible to new requirements and inputs |  |  |  |

# TEST METRICS

|  |  |  |
| --- | --- | --- |
| Rework Effort Ratio | = (Actual rework efforts spent in that phase/ total actual efforts spent in that phase) X 100 |  |
| Requirement Creep | = (Total number of requirements added/No of initial requirements) X100 |  |
| Schedule Variance | = (Actual efforts – estimated efforts) / Estimated Efforts) X 100 |  |
| Cost of finding a defect in testing | = (Total effort spent on testing/ defects found in testing) |  |
| Schedule slippage | = (Actual end date – Estimated end date) / (Planned End Date – Planned Start Date) X 100 |  |
| Passed Test Cases Percentage | = (Number of Passed Tests/Total number of tests executed) X 100 |  |
| Failed Test Cases Percentage | = (Number of Failed Tests/Total number of tests executed) X 100 |  |
| Blocked Test Cases Percentage | = (Number of Blocked Tests/Total number of tests executed) X 100 |  |
| Fixed Defects Percentage | = (Defects Fixed/Defects Reported) X 100 |  |
| Accepted Defects Percentage | = (Defects Accepted as Valid by Dev Team /Total Defects Reported) X 100 |  |
| Defects Deferred Percentage | = (Defects deferred for future releases /Total Defects Reported) X 100 |  |
| Critical Defects Percentage | = (Critical Defects / Total Defects Reported) X 100 |  |
| Average time for a dev team to repair defects | = (Total time taken for bugfixes/Number of bugs) |  |
| Number of tests run per time period | = Number of tests run/Total time |  |
| Test design efficiency | = Number of tests designed /Total time |  |
| Test review efficiency | = Number of tests reviewed /Total time |  |
| Bug find rote or Number of defects per test hour | = Total number of defects/Total number of test hours |  |

# BUG TRIAGE

The goal of the triage is to

* To define the type of resolution for each bug
* To prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Defect ID | Scenario | Expected | Actual | URL | Status | Comments |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |

# 

# Suspension Criteria and Resumption Requirements

Suspension criteria define the criteria to be used to suspend all or part of the testing procedure while Resumption criteria determine when testing can resume after it has been suspended

## 

# Test Completeness Check

Here you define the criterias that will deem your testing complete.

For instance, a few criteria to check Test Completeness would be

* 100% test coverage – Check in alignment with the requirements (frontend+backend)
* 100% code quality coverage
* User Acceptance testing coverahe – 100%
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

# Resource & Environment Needs

## Testing Tools

Make a list of Tools like

* Requirements Tracking Tool
* Bug Tracking Tool
* Metrics testing tool

Required to test the project

# Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |