<Offbeat Escape>

Software Quality Assurance Plan 1.2

<Date :April 2021>

Document Modification Guidelines

This document is a template for a Software Quality Assurance Plan intended for use by Quality management team

There are three general types of text format: Text in Black, Text in Blue with<>, Textin Blue with [ ] and Text underlined inblue.

Text in Black

- This category marks the Software Quality Assurance Plan features without modification. All document section headings are in the same category.

Text in Blue with<>

- This category, <blue>, denotes text which are to be updated like project specific information such as the project name, team members name, designation title, date, revision #, Document Control Number and the like

Textin Blue with [ ]

- Text in this style, [blue], indicated additional modification on text in any specific section. It is to be deleted before the document is finalized.

Text underlined in Blue

- Text in this style, blue, is used to indicate active links. These links are usually to project or software assurance websites.

All the components included in the **table of contents** will be addressed, but the level of information provided is left to the Quality Assurance Engineer. The Software Quality Assurance plan should align with the scope and complexity of the project

Section headers can be included when necessity arises,however the **existing headers should be left unaltered.** If a particular section is invalid details are to be provided in the description section.

Once the SQA is finalised, this section **Document Modification Guidelines** will be delete

Signature Page

PREPARED BY	Quality Assurance Engineer/Scrum Master	Margaret Arulmalar Rebeka Nesaraj	110026527	03.02.2021
REVIEWED BY	Software Engineer	Srishti Jain	110026562	04.02.2021
APPROVED BY	Product Owner /Client	Professor Dr Aznam Yacoub		06.02.2021

SQA Revisions Records

Revision Number	Date	Description	Page Number	Modified by	Approvals
1	07.02.2021	Updating the review comments for Sprint duration and,SDLC	10	Margaret Arulmalar Rebeka Nesaraj	Dr Aznam Yacoub
2	13.04.2021	Roles and responsibilities update	3,4	Margaret Arulmalar Rebeka Nesaraj	Dr Aznam Yacoub

Table of Contents

< 1

1. Introduction 7

1.1Purpose 7

1.2Reference Documents 7

1.3Checkpoints 7

2.0 SQA Description 7

2.1 Scope and Objectives 7

2.2 Required Skillset 8

2.3 SQA Resources 8

2.4 SQA Acronyms and Abbreviations 9

3.0 Management 9

3.1 Management Team 10

3.1.2 Roles and Responsibility 10

3.2 Tasks 11

Planning tasks: 11

Management: 11

3.3 Software Quality Personnel 11

3.4 Reviews and Audit 11

3.4.1 Peer Review 12

3.4.2 Formal Technical Review 12

4.0 Documentation 12

5.0 Standard Practices and Conventions 12

5.1 Communication 12

5.1.1Client Communication 12

5.1.2 Internal Communication 13

5.2 Voting System: 13

6.0 Verification and Validation 13

Verification: 13

Validation:..... 13

7.0 Problem Reporting and Corrective Action 14

8.0 Tools, Techniques and Methodologies.....14

8.1 Offbeat Escape Tools 14

8.2 Software Quality Tools 14

9.0 Record Collections, Maintenance, and Retention 15

10.0 Training.....15

10.1 Self pace training: 15

10.2 Cross platform peer training: 15

11.0 Risk Management 15

12.0 Scrum Agile methodology 16

12.1.0 Process to follow 16

12.1.1 Software Development Life Cycle model 16

Requirement Gathering 16

Specifications..... 17

Design..... 17

Development 17

Testing 17

Deployment 17

12.1.2 Events in Scrum 18

12.1.3 Planning in Scrum..... 19

12.1.4 Scrum Artefacts 19

13.0 Defect reporting and Handling.....19

14.0 SQAP Change Procedure and History 22

1. Introduction

This Software Quality Assurance Plan (SQAP) sets forth the standard practices, process and procedures for the project <Offbeat Escape>to ensure “the right things are done in the right way. This document aids to deliver high quality within the given timeline.

1.1Purpose

SQAP serves as the road map for instituting the best practices and SQA activitiesfor the project. The main purpose of SQAP is to ensure production of high quality end software product according to the specific requirements stated. The results of the reviews and audits conducted in the Software Quality Assurance plan would be provided to the product owner to track and assess the progress being made on the project.

1.2Reference Documents

- IEEE standard for Software Quality Assurance Planning
- IEEE guide for Software Quality Assurance Planning
- ANSI/IEEE Standard for Software Verification and Validation Plans

1.3Checkpoints

Each stage of development will have at least one formal checkpoint called a stage exit. Successful exit of a stage is marked by the deliverables completed by due date, all outstanding issues should have acceptable action plans. All affected functional teams participate and provide inputs to the exit stage

2.0 SQA Description

2.1 Scope and Objectives

The objectives of SQA are:

- Effective quality management approach using Software engineering methodology and tools
- Formal technical reviews that are applied throughout the software process

- Testing strategy to ensure maximum coverage
- Control of software documentation and the changes made to it ·
- A procedure to assure compliance with software development standards when applicable
- Measurement and reporting mechanisms

2.2 Required Skillset

The Quality assurance personal is desired to have the below mentioned skillset

- Good communication skills
- An understanding of software QA methodologies, tools, and processes
- Ability to analyse the requirement and come up with test design strategies
- Ability to perform gap analysis as and when required
- Knowledge of SQL and scripting
- Time Management - Managing one's own time and the time of others
- Quality Control Analysis - Conducting tests and inspections of products, services, or processes to evaluate quality or performance.
- Hands-on experience with Test Management and test Automation tool(s)
- Experience with Web-Based Testing Applications
- This position will work independently to ensure objective audits of the work products as they are being developed and objective reviews of project management processes and stage exits.
- Integrity - Job requires being honest and ethical.
- Cooperation - Job requires being pleasant with others on the job and displaying a good-natured, cooperative attitude.
- Innovation - Job requires creativity and alternative thinking to develop new ideas for and answers to work-related problems.

2.3 SQA Resources

No external SQA resources are defined for this project. The internal resources includes Software engineer, Test management tools, Defect trackers and Microsoft office software’s

2.4 SQA Acronyms and Abbreviations

The following acronyms will be used the

Acronyms	Abbreviations
QA	Quality Assurance
QC	Quality Control
RTM	Requirement Traceability Matrix
SDLC	Software Development Life Cycle
STLC	Software Test Life Cycle
SRS	System Requirement Specification
SCM	Software Configuration Management
SQAP	Software Quality Assurance Plan
SQA	Software Quality Assurance
FRS	Functional Requirement Specification
SB	Sprint Backloag
PB	Product Backloag

3.0 Management

This team includes Project Director, Developers, Technical Analyst and Quality Engineer

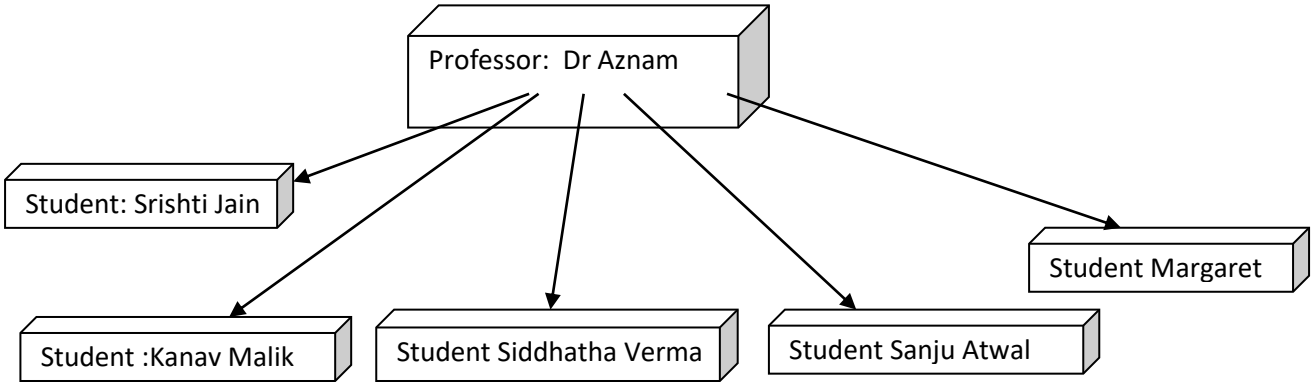
Management :

- Dr Aznam

Software Engineers:

- Srishti Jain
- Kanav Malik
- Siddhartha Verma
- Sanju Atwal
- Margaret Arulmalar Rebeka Nesaraj

3.1 Management Team



3.1.2 Roles and Responsibility

The role of each team member will multidimensional since we have a small team.

Dr.Aznam Yacoub – Product Director

The product director shall approve all important decisions.

Srishti –Software Engineer
Role – Software Engineer
Responsibility - planning, organizing, and directing the completion of Offbeat Escape project while ensuring the project is on time, on budget, and within scope. Design & develop backend APIs.

Siddharth –Software Engineer
Role – Software Development Engineer (Frontend)
Responsibility – Design & Develop user interfaces and work on deployment.

Kanav –SoftwareEngineer (Frontend & backend Developer)
Role – Software Development Engineer (Backend)

Responsibility - Design & develop backend APIsand work on deployment

Sanju –Software Engineer
Role – Software Development Engineer (Frontend & Backend)

Responsibility - Develop user interfaces and APIs.

Margaret –Software Engineer
Role - Scrum Master &Quality Assurance Engineer/Backup resource development.
Responsibility - Monitor the assigned tasks Coordinate the activities of the team

Schedule the scrum meeting coordinating the activities of the team, involved in requirement analysis,test design , test execution ,defect tracking and maintenance

3.2 Tasks

The head of the SQA unit is responsible for all the quality assurance tasks performed namely Planning tasks, Management, SQA professional activities

Planning tasks:

- Project Planning
- Task assignment

Management:

- Monitoring implementation of the SQA activity program

SQA professional activities:

- Preparation of Requirements traceability matrix, test reports
- Requirement Analysis
- Prepare the RTM
- Risk Management
- Software Problem Reporting and Corrective Action
- Test Management (Verification & Validation)

3.3 Software Quality Personnel

SQP have the responsibility and authority of verifying compliance to processes, standards, procedures and plans referenced in project specific Plans .Carry out Software Quality Assurance activities as. The SQP shall prepare the following: (i)Software Quality Assurance Plan (ii) Software Quality Assurance Audit Records. The SQP shall review various documents generated in the Development Life Cycle. Carry out Phase Transition Checks and Software Conformity Reviews. The SQP shall participate in various reviews conducted for the projects.

3.4 Reviews and Audit

Reviews are conducted weekly to ensure the software design is in compliance with requirements and to uncover the potential bugs at the early stages of development. A biweekly audit shall be operational to ensure the records are documented from time to time.

3.4.1 Peer Review

Peer reviews are conducted as and when the need arises to ensures we are on track

3.4.2 Formal Technical Review

Formal technical reviews shall be conducted with the Technical analyst frequently to achieve software development uniformity and to manage the project efficiently

4.0 Documentation

All the documents are to be uploaded in SharePoint folder for the project.The documents should be accessible by all the project members. The main headers shall be in Arial Font, Black color and size 14 while text content shall be in size 12 of the former. Spell checks are recommended before uploading the documents and the line spacing , numbering and indents should be taken care of

5.0 Standard Practices and Conventions

All the internal communications as well as the communication with the client shall we documented .The communications shall be in the form of email or Team Meeting .The standard format for official communications shall be in Arial font size 12 and black color. Make sure to be available for the meeting ahead of time. Unavailability or meeting cancellation or postponement should be intimated at least 1 hour prior.

5.1 Communication

5.1.1Client Communication

The review meetings, presentations and clarifications sessions with the client shall be conducted in Team Meetings .The agenda is attached to the meeting notes .The meetings shall be conducted based on the clients availability

5.1.2 Internal Communication

Microsoft Teams meeting or teams chat or Uwin emails are used for internal communication and should be in compliance with the standard format (refer section 5.0)

5.2 Voting System:

Due to the relatively small size of the team voting system works out fine for when we make some decision for the change of the project, we vote for it. Each team member either supports it or opposes it.

6.0 Verification and Validation

The following IEEE definitions apply in this SQAP:

Verification: The process of determining whether or not the products of a given stage of the software development cycle fulfil the requirements established during the previous stage.

Validation: The process of evaluating software at the end of the software development process (acceptance testing activity in the Testing stage) to ensure compliance with software requirements. The term requirements encompasses the areas of hardware, user interface, operator, software interface, functionality, performance, communications, security, access, and backup and recovery.

Verification:

During the verification stage the criticality of requirements are assessed to identify key performance or critical areas of software

Validation:

The following activities will be performed as part of requirements validation:

• Plan acceptance testing, including criteria for:

•Compliance with all requirements

• Adequacy of user documentation

• Performance at boundaries and under stress conditions.

•Plan documentation of test tasks and results.

• Execute the Acceptance Test Plan.

• Document acceptance test results.

7.0 Problem Reporting and Corrective Action

On encountering a bug it is duly reported in Jira and assigned to the development team .Once the bug is accepted and fixed, it is retested. If the bug is still open then it is reassigned , else it is closed if it is in conformance with the requirements .The Bug Report (BR) shall be audited to assure that problem reports are completely filled out, corrective action is documented, change review and change control are performed, and they are properly closed.

8.0 Tools, Techniques and Methodologies

The <Offbeat Escape>tools will be used by the Development team meanwhile the testing team uses the Software quality tools as listed below

8.1 Offbeat Escape Tools

- Tech stack:
- frontend: angularV11 Stable
- backend: Express JS V12.16.3 LTS
- VersionControl: Git
- CI/CD: Travis-ci
- Hosting Service: Amazon Web Services
- Images Hosting Service: AWS S3 bucket
- Load balancing: NGINX
- frontend /Backend deployment: Docker
- Chatbot: Python, NodeJS, Google,Heroku

8.2 Software Quality Tools

- Defect tracking : Jira
- Snipping tool

- Test management : Microsoft excel
- Screenshot repository :SharePoint, Word Document
- Confluence :Clarification tracker

9.0 Record Collections, Maintenance, and Retention

A soft copy of the design documents and QAP are to be documented in Jira . In addition a local copy is maintained for backup by the concerned person

10.0 Training

As the team is relatively small in size the resources are expected to update their skills and contribute to the development area to meet the goals on time .Self paced training and Cross platform peer trainings are encouraged.

10.1 Self pace training:

The individuals scale up to the project needs by learning new technologies through internet resources and academic books.

10.2 Cross platform peer training:

The individuals learn by observing the project mates work and make notes of the same. Once they have acquired the necessary skills they start working on sub tasks and move to independent contributors

11.0 Risk Management

SQ personnel will assess the project's risk management process against the <Offbeat Escape>Risk Management Plan.SQA personnel participates regularly in risk management meetings and reports any software risks to the SAM and the project Manager.

12.0 Scrum Agile methodology

12.1.0 Process to follow

Considering the time constraints and the limited availability of the resources in <Offbeat Escape>project we are adopting the Agile Scrum Methodology in which we plan to test the features parallel to development cycle. If n features are planned for sprint 1, the features are shared with QA. Even when the development of the 1st feature is in progress the QA analyses the requirements and finalise the test strategies and design the test cases and prepare the test data for both constructive and destructive testing .Once a feature is developed QA will proceed even when the development of the next feature in the same sprint is under development. Every sprint will have a 2-3 day buffer to cover up for bugs and QA sign off for the sprint. Also not every sprint is expected to deliver a build.The duration of the Sprint is 2 weeks

The development team shall commit the code to the git repository frequently even when the feature development is partially complete to ensure smooth delivery in case of emergency or unavailability of a resource, in addition this will act as a primary source of code backup and prevent mishaps resulting in code loss.

Functional requirement's can be clarified with the Product owner and the technical analysts shall clarify the technical queries. New resources are given knowledge transfer and trained parallel and involved in all the sprint activities as well

12.1.1 Software Development Life Cycle model

Scrum is an agile development methodology will be used in the development of Offbeat Escape project .The primary objective of Scrum is to satisfy the customer's need through an environment of transparency in communication, collective responsibility and continuous progress. The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority (PB) that the owner of the product wants to obtain.

Requirement Gathering

At the beginning of each sprint, a client call was arranged to discuss last sprint's analysis and confirm requirements for next sprint features/tasks.
Requirement Analysis

Client's feedback from requirement gathering phase was analysed and in case of any discrepancies, another meeting was scheduled to close the open points.

Specifications

Specifications for all the features was shard with other team members for their inputs, suggestions, and improvements.

Design

Design was created once the specifications were approved by the team. Design and research for task to be implemented went hand in hand given knowledge constraints and inexperience with tech being employed.

Development

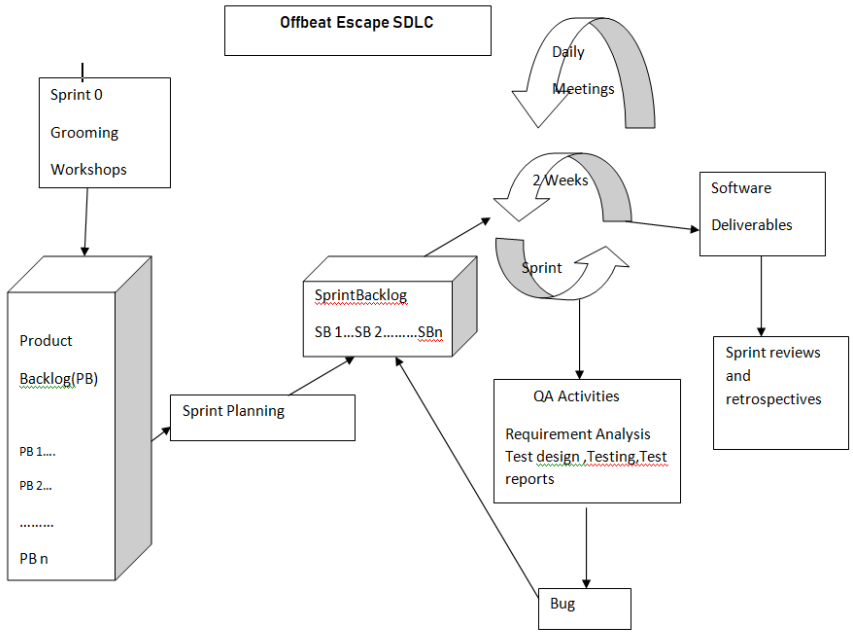
Development was done based on specifications and design. Code Reviews and walkthroughs were done to ensure quality adherence. Documentation was added parallelly to help a new member or a co-member to get the overview and pick up with development quickly when needed.

Testing

With requirement analysis, testing started in parallel to analyse the requirement, plan the test suite, cases, and design.
All the features developed in one sprint were put to testing in next and the bugs were reported over Jira with added details for steps and screenshots for
clear understanding of dev team.

Deployment

Starting from Sprint-2, deployment was researched, and setup was being progressively performed and achieved by the end of project.



12.1.2 Events in Scrum

Sprint: Sprint is the basic unit of work for a Scrum team

Sprint Planning: The goal of the Sprint Planning is to define what is going to be done in the Sprint and how it is going to be done. This meeting is held at the beginning of each Sprint and is defined how it will approach the project coming from the Product Backlog stages and deadlines. Each Sprint is composed of different features.

Daily Scrum: The objective of the Daily Scrum is to evaluate the progress and trend until the end of the Sprint, synchronizing the activities and creating a plan for the next 24 hours. It is a brief meeting that takes place daily during the Sprint period. Three questions are answered individually: What did I do yesterday? What am I going to do today? What help do I need? The Scrum Master should try to solve problems or obstacles that arise.

Sprint Review: The goal of the sprint review is to show what work has been completed with regards to the product backlog for future deliveries. The finished sprint is reviewed, and there should already be a clear and tangible advancement in the product to present to the client.

Sprint Retrospective: The team reviews the completed goals of the finished sprint, write down the good and the bad, so as not to repeat the mistakes again. The goal of the sprint retrospective is to identify possible process improvements and generate a plan to implement them in the next Sprint.

Grooming : In this meeting the product backlog items are discussed and the next sprint planning is prepared

12.1.3 Planning in Scrum

The Sprint Planning Meeting is held at the beginning of each Sprint. There are 5 sprints identified for Offbeat Escape project starting from Sprint 0 to Sprint 4. All the members of the Team participate in the meeting, i.e., the Scrum Master and all the Development Team. The entire Scrum team must understand and define what objective should be obtained in that Sprint (Sprint Goal). From this point the development team must design a work plan to achieve the objective.

12.1.4 Scrum Artefacts

Scrum Artifacts are designed to guarantee the transparency of key information in decision making.

Product Backlog (PB): The product backlog is a list that collects everything the product needs to satisfy the potential customers. It is prepared by the project manager and the functions are prioritized according to what is more and less important for the business.

Sprint Backlog (SB): It is a subset of items of the product backlog, which are selected by the team to perform during the sprint on which they are going to work

User Stories: It is an end goal expressed from user's perspective .The requirements are gathered from the product owner and the user stories are documented in jira. User stories are written throughout the <Offbeat Escape>Project

13.0 Defect reporting and Handling

The QA reports any bug uncovered during the dev testing phase in Jira and assigns it to the dev team. The details steps of bug replication along with the timed screenshots and test data are to be provided with the bug .

Once the dev team acknowledges the bug validity the bug is opened and root cause analysis is performed by the developers. Frontend bugs are related to UI and the backend bugs are related to data organisation issues

Once the bug is fixed the developer provides resolution details and assigns it back to the developer. If the defect is fixed after retesting by the QA then it is closed after providing screenshot, else it is assigned back to the developer. Incase a bug cannot

be fixed in the current iteration it moves to the new sprint and tracked until closure. The bugs shall not be closed before QA signoff.

14.0 SQA Milestone

The following are the <Offbeat Escape>SQA project activities and work products that the QA Engineer will review and audit:

Sprint	Start Date	End Date	SQA Tasks	Person Incharge	Deliverable
0	17.01.2021	29.01.2021	Initial requirement analysis, Project proposal review	Margaret	Not applicable
1	30.01.2021	12.02.2021	i)Review and audit the SRS Provisional planning and development deliverables ii)Requirement analysis for Signup and login frontend /backened /api iii)Test designing and reports Sprint 1 iv)Test data generation	Margaret	SQAP version 1 Test plan Test Case Screenshots Defect reports Status reports Coding scripts Builds (not necessarily delivered)

2	22.02.2021	07.03.2021	ii)Requirement analysis for User post creation , edit , delete and branding, myposts iii)Test designing and reports Sprint 2 iv)Test data generation v)Technical training vi)Automating smoke test cases	Margaret	Test plan Test Case Screenshots Defect reports Status reports Coding scripts QA signoff
3	08.02.2021	21.03.2021	ii)Requirement analysis for post comments, google,facebook login , friend , search friends , friend requests and notification iii)Test designing and reports Sprint 3 iv)Test data generation and retesting closed defects v)Internal training	Margaret	SQAP version 3 Test plan Test Case Screenshots Defect reports Status reports Coding scripts QA signoff
4	22.03.2021	04.04.2021	ii)Requirement analysis for dashboard features , trending posts , inspirer post iii)Test designing and reports Sprint 4 iv)Test data generation and retesting closed defects v)Product demo	Margaret	SQAP version 1.1 Test plan Test Case Screenshots Defect reports Status reports Coding scripts QA signoff

14.0 SQAP Change Procedure and History

SQA personnel are responsible for the maintenance of this plan. It is expected that this plan will be updated throughout the life cycle to reflect any changes in support levels and SQA activities. Changes to this document require prior approval of the <Offbeat Escape> product owner.