SummitAI - SDLC Process Document

Published On: 20th July 2020

SummitAI Documentation Team

Confidential Information

This document contains confidential and proprietary data and unauthorized use and disclosure of such information may result in damage or considerable loss to Symphony SUMMIT, Inc. The term “Confidential Information” denotes any and all technical and business information disclosed in any manner or form including, but not limited to, business strategies, methodologies, trade secrets, pricing, software programs, and relationships with third parties, client lists and related information, information pertaining to vendors, employees and affiliates. The Confidential Information shall be held in confidence and shall not be used other than for the purposes intended, and as specifically stated in the proposal. Further, the Confidential Information may only be released to employees and persons on a need to know basis, who shall be obliged to maintain the information confidential and the Confidential Information shall not be released or disclosed to any other third party without the prior written consent of SUMMIT Software, Inc.

© 2020 SUMMIT Software Inc. All Rights Reserved.

Version History

Document Version	Published Date
V01	04TH April, 2017
V02	20th December, 2018
V03	16th May, 2019
V04	10th May 2020
V05	20th July 2020

Document Owner – Angkur Sarma

Contents

Introduction

Overview

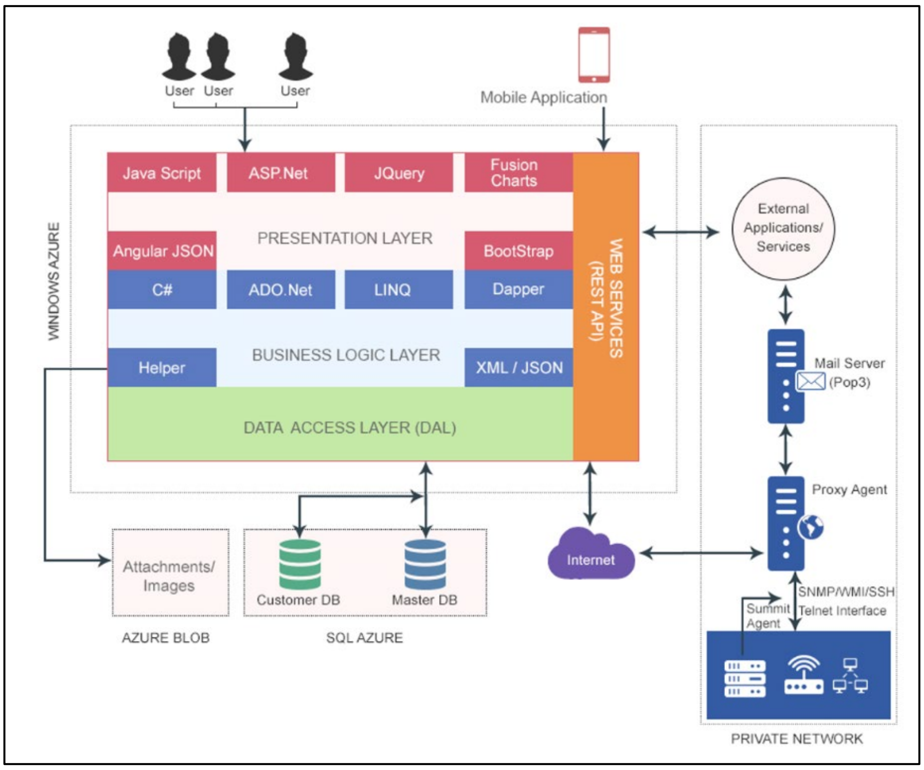
This Document covers the SDLC Process adopted by Symphony SummitAI for Feature and new Developments. Symphony SummitAI follows Scrum and Kanban Flavors of Agile Software Development methodology. There are six modules in total and two of them are following Kanban flavors of Agile Methodology. We will cover more about these topics in the upcoming sections.

Scope

The Scope of this document is to run through the Scrum Process and deliverables at each stage of the SDLC and to understand the layers involved in it.

Architecture Diagram

Please find below the Architecture Diagram of Symphony Summit AI for On Cloud.



Project Management

Team

The Team in Agile is generally referred to as the **development Team** which comprises of Developers, Testers, Documentation Members, Module Leads. Team also comprises of a **Product Owner and a Scrum Master**.
Furthermore, there is a **Product Management Team** comprising of a Chief Product Officer, a Senior Director and a Product Specialist Engineer. We will talk about these roles in the below section.

Responsibilities

Product Management Team

The responsibility of the Product Management Team is to oversee the deliverables in align with the Business-critical requirements for a Particular Release. All the requirements are prioritized in align with the Business Goal and market propositions. The prioritized requirements are detailed down in a document, which is called as a Product Requirement Document (PRD).
A single Main Release spans over a period of 3 months with 3 Iterations of 4 weeks each.

Product Owner

- The product owner is the sole responsible to maintain the product Backlog.
- He is responsible to order the Items in the Product Back log to best achieve goals and Mission of organization as whole.
- Optimizing the Value of work which is being produced by Dev Team.
- Ensuring the Product Backlog is Clear and Visible to all. The Product Backlog is maintained in TFS.
- Detailing to the Product Backlog, Following the DEEP concept in agile. I,e Detailed , Emergent, Estimated and Prioritized. Estimation at this level is in T-Shrt sizing.

Scrum Master

Service to the Dev Team

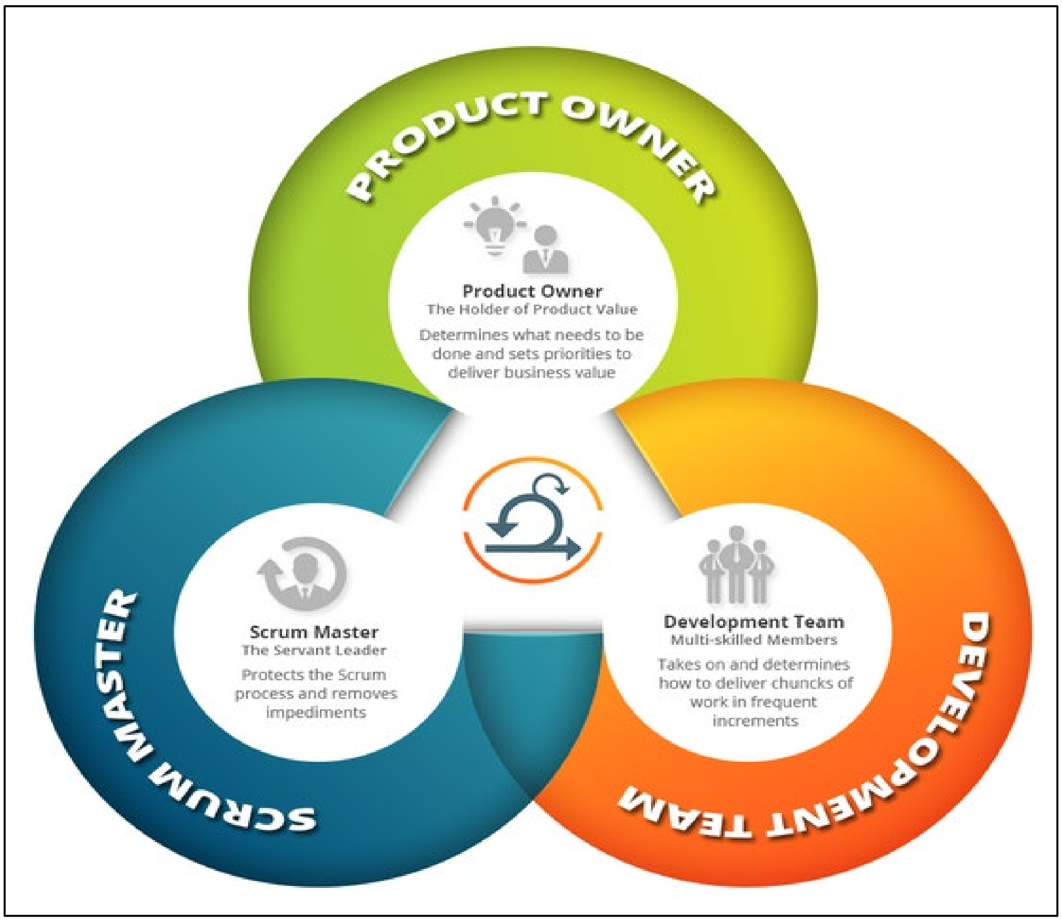
- Coaching the dev team in Self Organization and Cross Functionality
- Help Dev Team to Create High Value products by removing impediments.
- Facilitating Scrum events and Meetings.

Service to Product Owner

- Finding ways to effectively manager Product Backlog
- Helping understand for the need for clear and concise Product Backlog Items.

Service to Organization

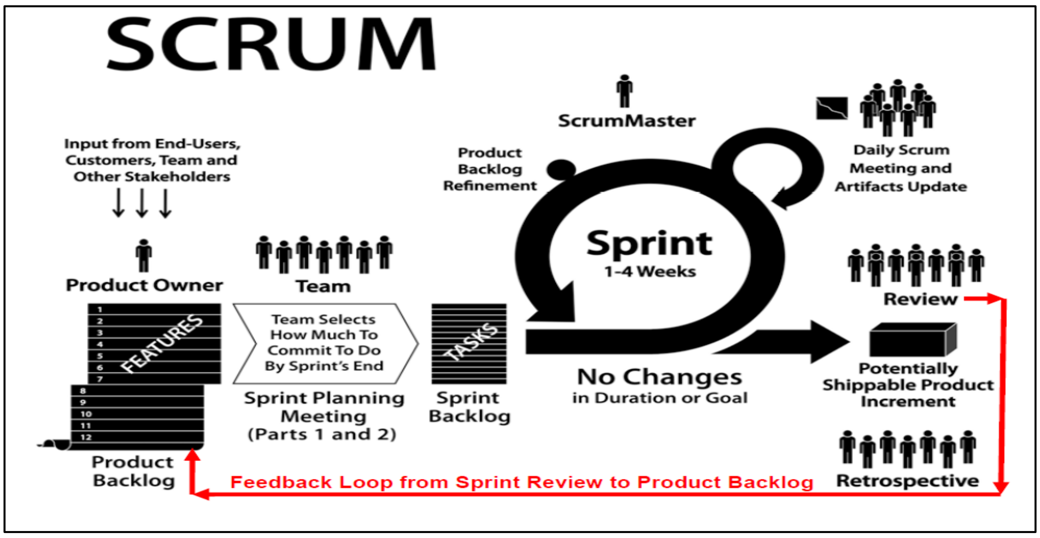
- Leading and Coaching in understanding, Adopting Scrum
- Planning Scrum Implementation within the Organization
- Act as a Change Agent.


PRODUCT OWNER
Product Owner
The Holder of Product Value
Determines what needs to be done and sets priorities to deliver business value
SCRUM MASTER
Scrum Master
The Servant Leader
Protects the Scrum process and removes impediments
DEVELOPMENT TEAM
Development Team
Multi-skilled Members
Takes on and determines how to deliver chuncks of work in frequent increments

Software Lifecycle Model

Scrum

In Symphony SummitAI, Scrum || Kanban is followed for New Feature Developments. Please refer to the below Diagram.


SCRUM
ScrumMaster
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Sprint
1-4 Weeks
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
Sprint Planning Meeting (Parts 1 and 2)
Sprint Backlog
No Changes in Duration or Goal
Potentially Shippable Product Increment
Product Backlog
Feedback Loop from Sprint Review to Product Backlog
Retrospective

As a part of the Scrum Process,

- The Requirements which are prioritized and in align with the Business Goals are pushed Down to the Product Owner from the Product Management Team. The PRD contains the Technical Details of a requirement i.e. Architectural changes, Common Code changes and the key Use cases which will be taken for further development in a phased manner.

- The Features listed down in PRD document are then inserted into the Product Back log according to its priority by the Product Owner. The items are discussed internally in terms of its coverage feasibility in a release and effort estimation. Further Details are sought from the Product Management team should there be a need.

- The next level of planning includes the Sprint Planning meeting with all Modules. In Symphony SummitAI, we follow one Iteration of four weeks. Hence, the Sprint Planning meeting spans over 4-6 hours depending on the level of Discussions required for the Targeted Items for a single Iteration. The First Half of the session is pre-dominantly used to describe the User Stories in hand to all Team members. The second part of the session is used by the team to estimate the User Stories by Playing Planning poker and using the Fibonacci series to give a Story point to a User Story. This will ascertain which of the User Stories for a Feature can be taken up for a Particular Iteration. The Capacity and Velocity of the Team are calculated beforehand, so that it will be easier for the Team to commit. Once, User Stories are listed down for an Iteration and taken up by Developer, those are being fed in Microsoft TFS for Tracking purpose. The User Stories also contain Acceptance Criteria, which are received From the PO to make sure there is no deviation from the original requirement VS the end outcome result. The Dashboard of TFS is used to provide metrics to PO and Management on Weekly Basis. TFS is updated on daily basis by Team members in terms of Hours Spent and Remaining for a Task.

- Once the Sprint Planning is done, Daily Stand ups are conducted every day for 15mins, where the key focus is on what has been completed, what is being planned for the day and Are there any Impediments which need immediate attention. Team also goes through the Burn Down Chart to make sure committed timelines are adhered to and there is no deviation from it.

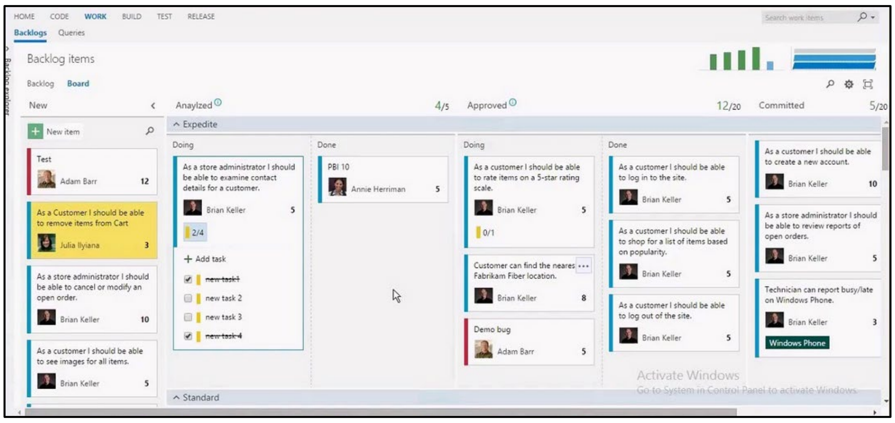
- Scrum of Scrums is conducted every week to Discuss items which are dependent on to other modules. The PO is a must attendee for this meeting.

- At the end of each Iteration, Review meetings are scheduled with PO to make sure, Definition of Done is met and is a potentially Shippable item. This is checked against the Acceptance criteria which were fed in an User Story.

- Lastly, Retrospective sessions are conducted at the end of the release.

Kanban

There are 2 modules in Symphony SummitAI which Follows Kanban Flavors of Agile due to the fact, the teams are more into Customer Issue addressal, Research and Development for Upcoming Features. Hence, first In, first out way of working is best suited for these types of scenario.



Test — Adam Barr 12

As a Customer I should be able to remove items from Cart — Julia Ilyiana 3

As a store administrator I should be able to cancel or modify an open order. — Brian Keller 10

As a customer I should be able to see images for all items. — Brian Keller 5

As a store administrator I should be able to examine contact details for a customer. — Brian Keller 5 — 2/4

~~new task1~~

new task 2

new task 3

~~new task 4~~

PBI 10 — Annie Herriman 5

As a customer I should be able to rate items on a 5-star rating scale. — Brian Keller 5 — 0/1

Customer can find the neares Fabrikam Fiber location. — Brian Keller 8

Demo bug — Adam Barr 5

As a customer I should be able to log in to the site. — Brian Keller 5

As a customer I should be able to shop for a list of items based on popularity. — Brian Keller 5

As a customer I should be able to log out of the site. — Brian Keller 5

As a customer I should be able to create a new account. — Brian Keller 10

As a store administrator I should be able to review reports of open orders. — Brian Keller 5

Technician can report busy/late on Windows Phone. — Brian Keller 3

Reviews with Product Owner

There are periodic Reviews with PO

Weekly Reviews

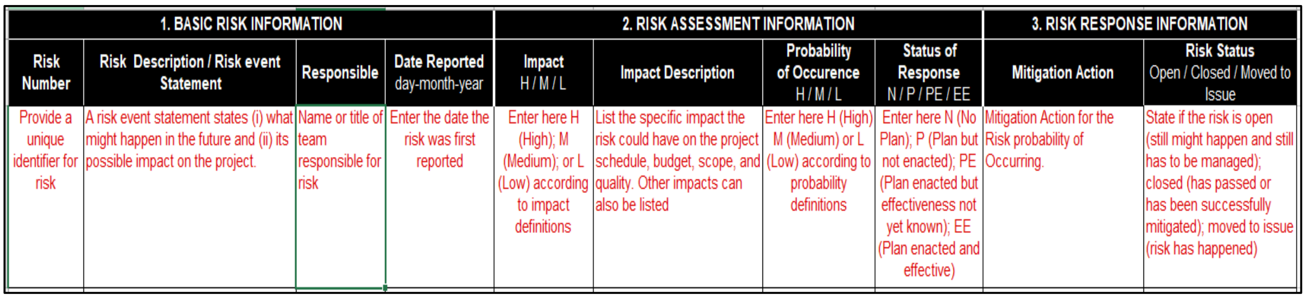
User Stories for which the Development is completed, and the System Testing is In-Progress. This is to ensure, that the Development is in align with the Acceptance Criteria.

Iteration Review

Once an Iteration is complete, review of All scoped User Stories is done, so that if any critical scenarios are missed, can be tapped. If required, the review points are taken up in the next Iteration.
Once all the reviews are done along with QA and Documentation, it sits in the Build Repository with the Build and Release Management team.

Risk Management

Risk Management in Symphony SummitAI is done by maintaining a Risk Register for a Release. Risk Register is circulated before the Start of a Release to all the Teams, so that probable Risks are listed down beforehand. It is updated every week, to make sure none of the risks have become an Issue OR in a critical Path to become an issue. This document is maintained by the Scrum Master. He is responsible to communicate, should he see any Listed items are in the verge of becoming an Issue.



1. BASIC RISK INFORMATION				2. RISK ASSESSMENT INFORMATION				3. RISK RESPONSE INFORMATION	
Risk Number	**Risk Description / Risk event Statement**	**Responsible**	**Date Reported** day-month-year	**Impact** H / M / L	**Impact Description**	**Probability of Occurence** H / M / L	**Status of Response** N / P / PE / EE	**Mitigation Action**	**Risk Status** Open / Closed / Moved to Issue
Provide a unique identifier for risk	A risk event statement states (i) what might happen in the future and (ii) its possible impact on the project.	Name or title of team responsible for risk	Enter the date the risk was first reported	Enter here H (High); M (Medium); or L (Low) according to impact definitions	List the specific impact the risk could have on the project schedule, budget, scope, and quality. Other impacts can also be listed	Enter here H (High) M (Medium) or L (Low) according to probability definitions	Enter here N (No Plan); P (Plan but not enacted); PE (Plan enacted but effectiveness not yet known); EE (Plan enacted and effective)	Mitigation Action for the Risk probability of Occurring.	State if the risk is open (still might happen and still has to be managed); closed (has passed or has been successfully mitigated); moved to issue (risk has happened)

The Document is segregated in three Sections namely Basic Risk Information, Risk Assessment Information and Risk mitigation information. Please find the screen shot.
Furthermore, once a Risk is identified as an Issue, discussion is taken up on priority to make sure the Impact of the risk is ascertained and contained at the initial level.

Calling out the Risks and Mitigation Process.

The same document is used to Track High-Level risks about a Release or Product. Based on the probability of the risk occurrence, the risks in the Risk register are prioritized and can be deleted if the risk does not occur or decreased.
The mitigation steps include to keep a tight vigil on the Risks and to ensure that the Risks are not turned into Issues. This is done by going through the risks every two weeks timeframe. Enter the details of the mitigation steps in the Risk Response Information column.
This document is uploaded in the SharePoint or TFS so that everyone in the organization is aware of the risks that might occur in due course of time.

Following are the few risks that are identified at the organizational level:

1. **Build Manager**
   Officially, there is one build manager for all types of builds, whether it is for HF or Main release. There is a back-up build Manager who has the skills and Knowledge to perform build activities in the absence of the primary build engineer. Head of Engineering VP must approve the work that is performed by the secondary build engineer. Hence, there is no impact.

2. **TFS server**
   The TFS server is used for the internal development work. It is not a production server where customers get affected when it is down. The IT team of Summit works swiftly to resolve any issues to the TFS servers.

3. **Team Member Absentees**
   In case of any emergency leave by any team member, there are enough skills in the teams to take over the work. However, there may be a small impact of a speculated delay in the deliverables.

4. **Scrum Master**
   In the absence of the Chief Scrum Master, all the Module Leads can perform the duties of a Scrum Master. Enough Training is provided to the leads on the Scrum Framework to perform all the ceremonies, tasks, etc. daily. In addition to that, there is already a plan in place for a Second Scrum Master to be on-board soon. Hence, there is no impact.

Communication and Collaboration

Symphony Summit AI follows the Agile Methodology, hence so, there specific Meetings or Ceremonies as we call it for Iterations and for the Release as a whole.
The communication Management starts, the moment, Development team receives the PRD from Product Management Team. The Release Planning meeting happens with the PO and Team leads to ascertain and confirm which of the Scoped Items from PRD can be taken based on Business Criticality and Priority. This meeting is named as the **Release Planning Meeting**.
Once the release Planning meeting is concluded with the outcome of the Scoped items for Teams from PRD, **Sprint Planning meeting** is conducted. The Items are estimated and aligned IN for a Particular Iteration.

Then follows the **Daily Stand Up** where the updates are received from Team members, any blockers and committed items for the day.
Finally, **Iteration review** of Done things are conducted with PO and review comments are received. Once the Release is done, the **Retrospective session** is conducted.
Furthermore, we also have a weekly slot for **Scrum of Scrums,** to discuss and clarify dependencies among different Teams.

Change Management or Requirement change

Although, once the Iteration starts, any changes to the requirements are limited or NIL, there is always a chance of a change. Any change coming in between of an iteration, are discussed with PO on its feasibility to take it up in the running iteration. If the change is taken up in the running Iteration, then negotiation happens on what present items will be put in Back Log. If the change is not a high priority for the current Iteration, the change is put in back Log and is prioritized for the upcoming Iteration. These changes are taken up only when PO nods ok for the same.
Hence, the steps are as below:

- The change Request comes from sources outside of the team mainly from Stakeholders.
- The change is discussed with PO in terms of its complexity and sizing.
- If PO agrees to take the change in the current iteration, Negotiation happens which other item will be pushed back to the back log and the Stake holder is kept informed of the decision.
- If the change is not taken up in the current Iteration, it finds its way to the product Back Log and Stakeholder is kept informed of the same.
- Out of these two scenarios, there are also chances of adhoc requests coming on to the development Team to cater to some customer issues or customer Support. To fulfill these types of expectation, the capacity for development team is always at >100% capacity. These support activities are tracked in TFS in a different team Area and communicated to PO on weekly basis.

Tools used in Development

Development Team uses Microsoft TFS to insert, Update and Track User Stories for a particular Iteration. Issues are logged against the User Stories while Testing for the same by QA Team members. TFS features like Dashboard, Burn Down Chart, Cumulative charts are being used to report Progress on a Particular Iteration to the PO and Management.

Build Management System

Symphony SummitAI Follows twice a week Build process for Development. All the User Stories which are complete in terms of UT, Code review and Peer Review along with Issue fixes are checked-In the relevant branch. The build engineer takes the build and process it to the QA. At this point the Build and Upgrade system is a Manual one and looking into feasibility to migrate to some automated system.

QA Exeuction and Verification

The QA Team uses Microsoft TFS tool to insert, Update and Track the tasks. TFS is also used to Log issues for the completed User Stories. The Traceability is hereby maintained on the scenarios covered for a particular User Story. Furthermore, once an issue is logged, it will be in New State and will be assigned to the respective Developer, who worked on that User story. Once the issue is taken up by the Developer, the State will be changed to Active and will be changed to Resolved once it is fixed. It will be assigned back to the QA Team member and once it is verified by QA, the state will be changed to Closed.
TFS is used to keep a Track of all issues reported for a Particular Release.

Bug Priorities and Impacts

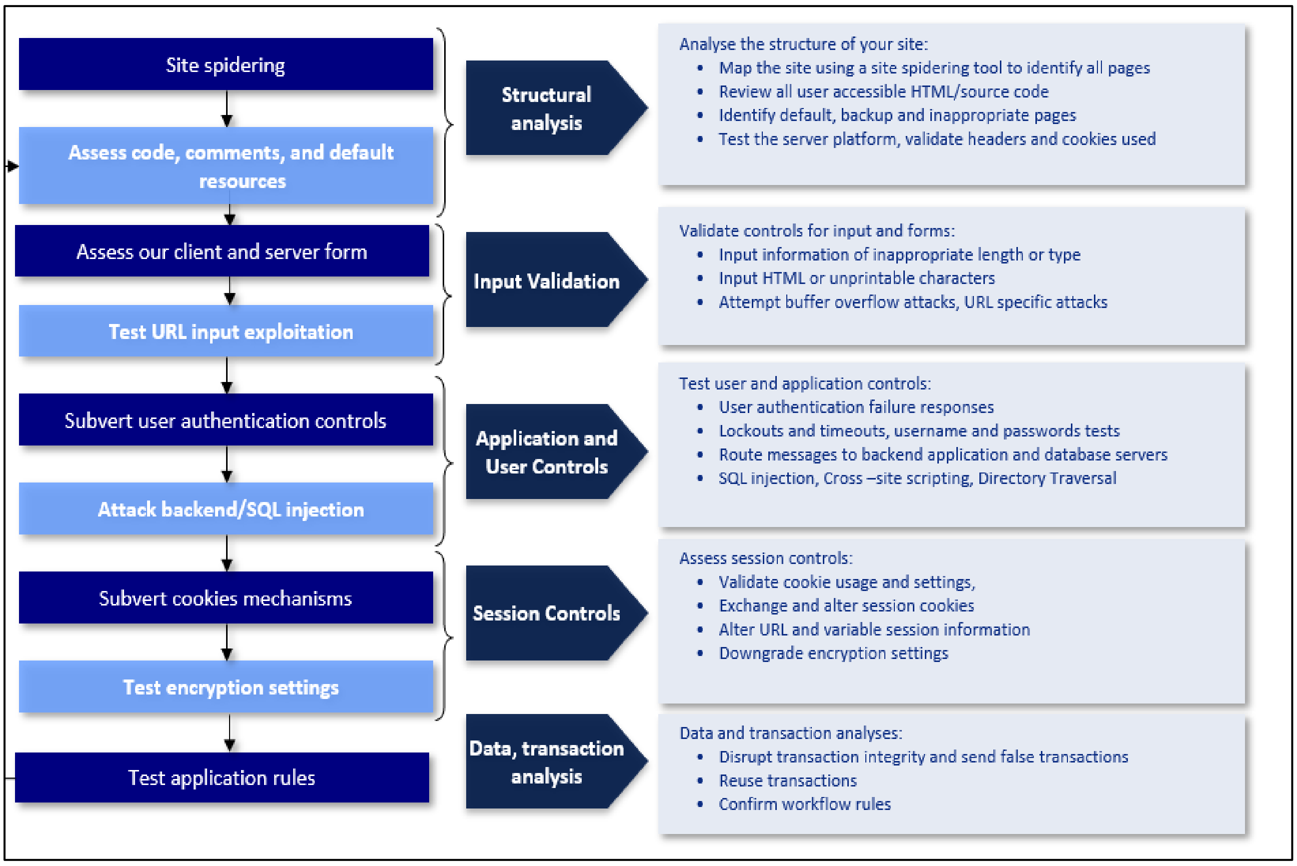
The QA team classifies a Bug into the following four categories:

- **Critical:** Critical defect affects functionality or critical data. It does not have a workaround and impacts the total system. The Development team must fix the issue on priority so that the QA team can continue the testing.
- **High:** High defect affects major functionality or major data. It has a workaround but is not obvious and difficult. It impacts most of the functionalities.
- **Medium:** Medium defect affects minor functionality or non-critical data. It has a workaround and impact may be on large areas of the system but there is no obvious breakage or issue with the system.
- **Low:** Low defect does not affect any functionality or data. It does not require a workaround and does not impact productivity or efficiency. It is hardly an inconvenience.

Penetration Test Process

Summit Application is also tested for Security Vulnerability by a Leading Security Consultant. The Process for the same is depicted below.

WEB APPLICATION TESTING



Site spidering
Assess code, comments, and default resources
Assess our client and server form
Test URL input exploitation
Subvert user authentication controls
Attack backend/SQL injection
Subvert cookies mechanisms
Test encryption settings
Test application rules
Structural analysis
Input Validation
Application and User Controls
Session Controls
Data, transaction analysis

Analyse the structure of your site:

- Map the site using a site spidering tool to identify all pages
- Review all user accessible HTML/source code
- Identify default, backup and inappropriate pages
- Test the server platform, validate headers and cookies used

Validate controls for input and forms:

- Input information of inappropriate length or type
- Input HTML or unprintable characters
- Attempt buffer overflow attacks, URL specific attacks

Test user and application controls:

- User authentication failure responses
- Lockouts and timeouts, username and passwords tests
- Route messages to backend application and database servers
- SQL injection, Cross –site scripting, Directory Traversal

Assess session controls:

- Validate cookie usage and settings,
- Exchange and alter session cookies
- Alter URL and variable session information
- Downgrade encryption settings

Data and transaction analyses:

- Disrupt transaction integrity and send false transactions
- Reuse transactions
- Confirm workflow rules

Mobile APPLICATION TESTING

Server-side Security

The server-side security testing is carried out using one of the approaches described in the application security assessment methodology.

Client-side Security

The client application is tested either using a platform emulator typically provided together with SDK and/or actual hardware device.

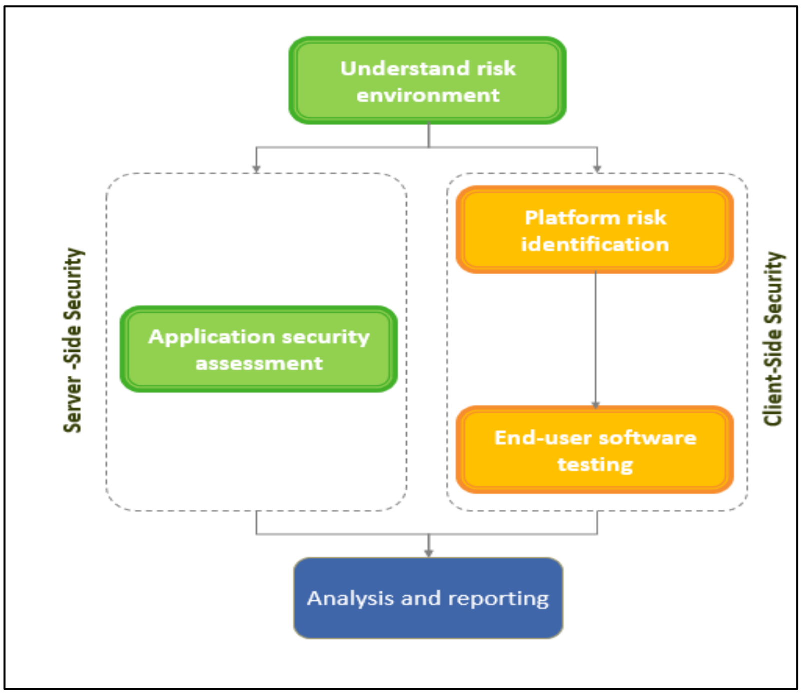
- **Platform Risk Identification**

  Functionality of the client application is thoroughly analysed to identify assumptions about platforms of executions that may not be always true, for example:

  - An application relies on GPS data being accurate, then such data may be spoofed if the application is executed on an emulator;
    - Storage and exchange of cryptographic keys or shared secrets between application and a security device such as SIM card cannot be intercepted by other applications;

- **End-user Software Testing**

  The data exchange between client-side application and server-side application is intercepted using various tools and the client-side application is being supplied with invalid responses to trigger erroneous behaviour. Fuzzing tools are used where possible to cover the maximum attack surface followed by manual investigation of suspicious behaviour.


Understand risk environment
Server -Side Security
Application security assessment
Platform risk identification
End-user software testing
Client-Side Security
Analysis and reporting

Documentation Management

The Documentation Development Life Cycle begins with a Planning Meeting. In the Planning Meeting the Tech Writers get to know about the features that are being planned for the release. Based on the features that are planned for the release, Tech Doc tasks, related to the User Stories, are created in the Team Foundation Server (TFS). The Knowledge Transfer (KT) sessions, for features planned in each Iteration, are conducted by the Development Teams. For the current Iteration, the Tech Writers document the features based on the KT received from the Development Teams. The draft version of the content created by the Tech Writers are shared to the Development and Quality Assurance teams for Technical Review. After the content is approved by the Development and QA teams, the technically reviewed content is shared for Peer Review amongst the writers in the Technical Documentation team. After the Peer Review, the content is finally shared for the Editorial Review. After the Editorial Review, the content is published, and the respective Tech Doc tasks are updated in the TFS. After updating the tasks, the writers close their respective Tech Doc tasks on completion of the documentation activities. The above process is then repeated in each iteration.

Release Management

The release Management starts from the point where all the Features scoped for a release are reviewed, Tested and Documented. The release document is reviewed before publishing it to the released version of the application. The Build Engineer builds the final tested version of the application with Installation and Proxy guides. The same information collated with release Document is then sent out as a communication to all interested parties and Stake Holders. The version stands released.

HF Process and Straetegy

SummitAI has a periodic Hotfixes Releases for the customer reported issues, after the release of a Main Version and/or upgradation to the Staging environment. There is a workflow for the same and the below points are critical to adherence to the HF process and Strategy for smooth collaboration between HF process and Core Development Process:

- *Support and Implementation team* at LEVEL 1 handles all the customer reported issues. This team analyzes the issue and understands the configurations made in the customer environment to ensure the system is configured as it should be. After all the analysis if the issue still exists, a work order is raised in the portal and pushed to the Quality Team to verify and re-confirm.
- *Quality team* tries to reproduce the issue. If the issue is reproduced then the Quality team raises a workorder in the portal and is aligned, according to the priority, for the next upcoming Hot fix Release and follows the Kanban method of going through the whole implementation process.

Now, below are the key Points which are modified based on the inputs from different teams and to ensure, the Hotfix process flows seamlessly with the ongoing Sprint Activities for a particular release.

- It was discussed and Confirmed that, the list of issues is pulled from the portal, every 20th of a particular month, and those issues are aligned to be released in subsequent Month end.
- The list is shared with the interested parties (Here mainly the Support and Implementation Team), to make sure, the Issues which are to be taken up are in align with the expectation in

terms of what should be taken and what should not. This step is very important to ensure, that no P1 and P2 are missed after Development team starts working on it.

- The issues which are taken up, will be fixed in the latest branch of the Customer version and the latest branch. For example:
  - **Issues from v5.5 and below, v57sp+ will be fixed in v57sp5hf04, HF05, HF06 and So On.**
  - **V56+ will be fixed in v56 sp2 hf14, HF15 and so on.**
  - **V5.8 ALPS+ will be fixed in ALPS Sp1 HF01, HF02 and So on.**
- It is suggested to create one Work Order for a reported issue instead of multiple Work Orders. This suffices issues related to updating and closure of multiple Work Orders by Multiple teams in the portal.
- In terms of the communication channel for the teams to be on Same page, make use of a common Platform. The said platform is in Discussion Phase. OR have
- Post starting on the agreed upon issues, if there are any P1 or P2 (Decision need to be taken after discussing with Management). The decision is based on the Criticality of the issue to the OPERATIONAL factor for the customer. This decision will also take counter of the type of Customer.

Post Development and QA cycle the released items can be in TFS for a particular release.