EDISON DESIGN SYSTEM

Software Development Plan (SDP)

Table of Contents

[1 Introduction 4](#_Toc52557130)

[2 References 4](#_Toc52557131)

[3 Definitions, Acronyms, Abbreviations 5](#_Toc52557132)

[4 Organization Chart 5](#_Toc52557133)

[4.1 Core Team 5](#_Toc52557134)

[4.2 Extended Team 6](#_Toc52557135)

[4.3 Leadership Team 6](#_Toc52557136)

[5 Control and Operating Mechanisms 6](#_Toc52557137)

[5.1 Meetings 6](#_Toc52557138)

[5.2 Communication 7](#_Toc52557139)

[5.3 Execution 7](#_Toc52557140)

[5.4 Change Control 7](#_Toc52557141)

[6 Software Development Processes 9](#_Toc52557142)

[7 SDLC Work Products 9](#_Toc52557143)

[8 Traceability 10](#_Toc52557144)

[9 Software Configuration Management 10](#_Toc52557145)

[9.1 Internal Document CIs 10](#_Toc52557146)

[9.2 Customer Deliverable Software, Hardware, and Document CI 10](#_Toc52557147)

[9.3 Supporting Software Items, Tools 11](#_Toc52557148)

[10 Defect Management 11](#_Toc52557149)

[11 Design Transfer 12](#_Toc52557150)

[11.1 EDS Finished Design Outputs 12](#_Toc52557151)

[11.2 Release Notes 12](#_Toc52557152)

[11.3 User and Service Manuals 12](#_Toc52557153)

[11.4 Technical Support Strategy 12](#_Toc52557154)

[11.5 Communication Mechanism 13](#_Toc52557155)

[12 Regulatory Compliance 13](#_Toc52557156)

[13 Usability Engineering Plan 13](#_Toc52557157)

[14 Risk Management 13](#_Toc52557158)

[14.1 Software Classification 13](#_Toc52557159)

[14.2 Safety & Cyber security Risk Management Plan 14](#_Toc52557160)

[14.3 Personnel Qualifications 14](#_Toc52557161)

[14.4 FMEA 14](#_Toc52557162)

[15 Software of Unknown Pedigree (SOUP) 14](#_Toc52557163)

[15.1 SOUP Identification 14](#_Toc52557164)

[15.2 SOUP Activities 14](#_Toc52557165)

[15.2.1 Evaluation 14](#_Toc52557166)

[15.2.2 Development 14](#_Toc52557167)

[15.2.3 Integration 15](#_Toc52557168)

[16 Design Verification Plan 15](#_Toc52557169)

[16.1 Product Specifications 15](#_Toc52557170)

[16.2 Entrance Criteria 15](#_Toc52557171)

[16.3 Verification of Design Outputs 15](#_Toc52557172)

[16.4 Verification Methods 15](#_Toc52557173)

[16.5 Test Cycles 16](#_Toc52557174)

[16.6 Resources 16](#_Toc52557175)

[16.7 Equipment and Environment 17](#_Toc52557176)

[16.8 Acceptance Criteria 17](#_Toc52557177)

[16.9 Software Unit Acceptance Criteria 17](#_Toc52557178)

[16.10 Integrating Software Items 18](#_Toc52557179)

[17 Design Validation 18](#_Toc52557180)

[18 Design Transfer 18](#_Toc52557181)

[19 External Evaluation 18](#_Toc52557182)

[20 Technical Review Schedule 19](#_Toc52557183)

[21 Review and Approval Requirements 19](#_Toc52557184)

[21.1 Design Planning & Design Inputs 19](#_Toc52557185)

[21.2 Regulatory Documents 19](#_Toc52557186)

[21.3 Design Outputs 19](#_Toc52557187)

[21.4 Design Reviews 20](#_Toc52557188)

[22 Revision History 20](#_Toc52557189)

# Introduction

This Software Development Plan (SDP) outlines deliverables, responsibilities, and activities necessary to develop, verify, release, and maintain the software product. This plan applies to EDISON DESIGN SYSTEM.

# References

The following table lists other documents associated with the program:

| Document Name | Document ID |
| --- | --- |
| Design History File | DOC2347322 |

# Definitions, Acronyms, Abbreviations

The following are the acronyms and abbreviations used in this document:

| Term | Definition |
| --- | --- |
| EDS | EDISON DESGIN SYSTEM |
| FDR | Formal Design Review |
| FMEA | Failure Mode Effects Analysis |
| IDP | Integrated Development Plan |
| MWS | MyWorkshop document management system. |
| PI | Program Increment: fixed length time boxed development |
| QA | Quality Assurance |
| QMS | Quality Management System |
| SDLC | Software Development Life Cycle |
| SDP | Software Development Plan |
| SOUP | Software of Unknown Pedigree |
| SPR | System Problem Report |
| TBD | To Be Done/Defined |
| TDR | Technical Design Review |
| WI | Work Instruction |
| V&V | Verification and Validation |
| SW | Software |
| GEHC | GE Healthcare |
| DHF | Design History File |
| UX | User Experience |
| NC | Non-Conformance |
| UA | Unacceptable |
| RMR | Risk Mitigation Required |

# Organization Chart

## Core Team

The Program Core Team is established by the Business Leadership Team (see later section) at the earliest stage of the product program and is staffed as a cross-functional, decision-making team with the expertise and skills to make the cross-functional program decisions. After commercial release of a New Product Introduction (NPI), the Program Core Team is responsible for authorizing the initiation and release of product design changes.

| Role | Name | Notes |
| --- | --- | --- |
| Program Manager | Buddi, Sandhya | N/A |
| Engineering Manager | Aluru, Prasad | N/A |
| Lead System Designer | Aluru, Prasad | N/A |
| Product Manager | Yuda, Kevin | N/A |
| Product Owner | Buddi, Sandhya | N/A |
| V&V Engineers | Kulkarni, Geeta  Somasila, Sasanka | N/A |
| UX/UI Developers | Murrell, Brian  Yoneyama, Mitsuko | N/A |
| QA Lead Engineer | Lala, Achint | N/A |

## Extended Team

| Role | Name | Notes |
| --- | --- | --- |
| UX Designers | Dufour, Lindsey  Janicki, Christopher | N/A |
| Product RA Executive | Manarik, John | N/A |

## Leadership Team

The Business Leadership Team is defined by the business functions required to sponsor the determination, definition, and execution of the product program. The functional leaders below are part of this team and participate in the various ‘M’ milestones and/or Formal Design Reviews (FDR) associated with a product NPI.

| Role | Name | Notes |
| --- | --- | --- |
| Business Leader / VP of UX | Kutscher, Birgit | N/A |
| Product QA Executive | Fox, Meghan | N/A |

# Control and Operating Mechanisms

## Meetings

Following are the Planned Team meetings as part of the EDS development.

|  |  |  |
| --- | --- | --- |
| **Meeting Type** | **Participants** | **Schedule** |
| Program Increment Release Train Planning | Development Teams | Once in 3 months or as agreed by the Team |
| Sprint Planning | Development Teams | Bi-weekly |
| Sprint Demo | Development Teams | Bi-weekly |
| Requirements/Design handoff meetings | UX team/ UI Team/Product Manager/Owner | Weekly |
| Release Meetings | All stakeholders | Quarterly |

## 

## Communication

All internal team communications will be utilizing effective tools like MSTeams etc.

Formal handoff of design specs from UX team to Developers will be provided by uploading them to Source Control (GitHub) and established automated process will deploy them to EDS-DOCS website in development environment for review.

## Execution

Development is performed in several consecutive Program Increments (PI). The EDS release development cycles and development team will use Scrum, and task-based execution approach to produce working increments of component software that can be integrated and tested reliably on a periodic basis.

Scrum cycles are used during development, integration build, and verification to define incremental implementations of software to meet the content goals of the release cycle.

## Change Control

We follow semantic versioning for our releases. Any changes requested will be categorized into three categories like Major/Minor/Fixes.

Major change is a breaking change that needs a major release. Changes like this will be incorporated in planned quarterly release unless it is a critical requirement for user.

Minor changes are fixes in the system that has less impact on consuming applications

Fixes are defects identified or reported by consumers will be triaged and assigned priority/severity.

Each release will have release notes associated with all details. Any breaking changes will be highlighted with change details.

Releases will follow GitFlow for approval process.

Our Git Flow is described in below diagram



Have git tagging approved by approval authority (Program Manager, PO & Architect) and document the approvers details in GitHub tag description for each release.

Major releases will be reviewed by QA Representative for final approval.

# Software Development Processes

EDS software will be developed in accordance with GEHC\_GQP\_10.01.016 Software Development Lifecycle Work Instruction: DOC1617896

Note that EDS does not produce a commercialized product, so catching programs integrating EDS into their systems are ultimately responsible for performing requirements flow down, verification of the intended use of the component in the product system design, validation of the overall system and all related DHF documentation for their product.

Development is performed in several consecutive Program Increments (PI). The EDS team will use Scrum, and task-based execution approach to produce working increments of component software that can be integrated and tested reliably on a periodic basis.

Scrum cycles are used during development, build and verification to define incremental implementations of software to meet the content goals of the release cycle.

When the content for the release cycle is achieved, the system formally transitions into a verification/Testing phase with all DHF requirements met according to the Verification/Testing Plan.

In term of Scrum Roles, the EDS Team have identified Scrum Masters and Product Owners within their teams to facilitate this development method. The content of each sprint will be determined by the backlog each team is working on for this or other releases. This method will be used for the entire development of this release.

Note: PIs are labeled with <i> - <yy> where <yy> represents the last 2 digits of the year and <i> represents the physical quarter of that year, this being incremented after each PI.

For example:

Q1 - 2020 – is the 1st increment completed during Q1 of 2020

Q2 - 2020 – is the 2nd increment completed during Q2 of 2020

# SDLC Work Products

The following list of work products are delivered as part of SDLC activities:

* UX Design specifications / Feature Requirements
* EDS Component Bundle
* Verification Summary Report
* EDS Release Notes
* Technical Design Review (if any)
* SOUP List

# Traceability

Design inputs for EDS are in the form of UX specifications/ feature requirements. And design outputs are the components artifacts that matches given UX specifications/ feature requirements by tracing them to test cases.

UX specifications/ feature requirements (Design Inputs) will be traced to the coded elements/widgets (Design Outputs) and to their testing.

Traceability will be documented in Verification Report.

# Software Configuration Management

## Internal Document CIs

The following identifies the internal, proprietary document CIs that are placed under configuration management:

| Document CI | Acquisition Point |
| --- | --- |
| Design History File (DHF) Index: Provides reference to each CI contained in the DHF\*.  EDS DHF Index | GitHub:  <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-dhf-index.xlsx> |

\* The items are under configuration control once they are released in the location identified in the DHF Index.

## Customer Deliverable Software, Hardware, and Document CI

The following identifies the CIs that are delivered to the customer:

| Customer Deliverable CIs | Acquisition Point |
| --- | --- |
| EDS Bundle | Artifactory:  <https://hc-us-east-aws-artifactory.cloud.health.ge.com/artifactory/api/npm/npm-gehc-ux-all> |
| EDS Documentation Website | Website:  <https://edisondesignsystem.com> |
| Release Notes | GitHub:  <https://edisondesignsystem.com/eds-release-notes>  <https://github.build.ge.com/gehc-ux/eds-design/blob/master/eds-release-notes/release-notes.md> |
| UX specifications/ Feature requirements | GitHub:  <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-feature-requirements.xlsx> |
| Verification Summary Report | GitHub:  <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-verification-summary-report.xlsx> |
| SOUP List | GitHub:  <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-soup-list.xlsx> |
| Product Integration Matrix | <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/product-integration-matrix.xlsx> |

## Supporting Software Items, Tools

The following identifies the non-product software and hardware CIs that are placed under configuration management:

| Non-Product CIs | Validation Status |
| --- | --- |
| **Microsoft Visual Studio Code:** Development tool. | Status: Validation not required per GxP  Reference: DOC2230713 |
| **MyWorkshop:** Document library. | Status: Validated  Reference: BOK11527 |
| **GitHub:** Software configuration management. | Status: Validated  Reference: DOC2346360 |
| **Rally:** Engineering Agile framework management. | Status: Validation not required per GxP  Reference: DOC0949694 |
| **Artifactory:** Artifact repository | Status: Validation not required per GxP  Reference: DOC1122719 |
| **Jenkins:** Build Tool | Status: Validation not required per GxP  Reference: DOC1699948 |

# Defect Management

During the development phase any defects discovered will be reviewed and corrected within the work product. Rally is the tool of choice during the development to manage defects. Teams will use a defect management system (Rally) to capture and schedule the found issues. Defects will be prioritized as part of the backlog and fixed during a PI.

During the Verification any defect encountered and not being fixed as part of the release, a known issue list will be created and added to release notes.

When a defect is closed by without taking any action to correct any element of the design a justification shall be added to the defect notes section in Rally.

Examples: typographical, editorial, or record-keeping errors, irrelevance of problems due to already-closed defects, and defects documenting product enhancements or improvement opportunities*.*

# Design Transfer

## EDS Finished Design Outputs

The following table lists the design outputs supported in this transition plan:

| Media/Document Name | Repository |
| --- | --- |
| EDS Bundle | <https://hc-us-east-aws-artifactory.cloud.health.ge.com/artifactory/api/npm/npm-gehc-ux-all> |
| Documentation Website | <https://edisondesignsystem.com> |
| Source Code | <https://github.build.ge.com/gehc-ux> |
| QMS Documentation | <https://github.build.ge.com/gehc-ux/eds/blob/master/qms> |

## 

## Release Notes

A detailed release notes of each version is available at: <https://edisondesignsystem.com/eds-release-notes>

<https://github.build.ge.com/gehc-ux/eds-design/blob/master/eds-release-notes/release-notes.md>

## User and Service Manuals

Usage documentation of Edison design system is available in developer documentation website: <https://edisondesignsystem.com>

## Technical Support Strategy

Detailed developer documentation including sample code snippets for each element/widget and reference examples are available at our documentation website: <https://edisondesignsystem.com>

For technical issues user should log issues in GitHub <https://github.build.ge.com/gehc-ux/eds> or you can also use our online form https:// <https://edisondesignsystem.com/submit-issue>

For all other questions visit our contact details page <https://edisondesignsystem.com/contact-us>

## Communication Mechanism

Each release will be communicated via email to the distribution list @HEALTH Global UI Engineering and also a release meeting will be conducted to share the release updates.

Following channels are used to communicate with incorporating programs:

| Channel | Link | Notes |
| --- | --- | --- |
| Website | <https://edisondesignsystem.com> | Developer documents with example code snippets |
| GitHub | <https://github.build.ge.com/gehc-ux> | Source code access and option to log issues |
| Yammer | <https://www.yammer.com/ge.com/#/threads/inGroup?type=in_group&feedId=10509957> | Used to share release announcements and other technical discussions |

# Regulatory Compliance

This program does not produce a commercialized product. So regulatory compliance is not applicable for this program. Catching programs are responsibility to address the regulatory requirements.

# Usability Engineering Plan

This program deliverables are UX/UI building blocks for catching programs to use in building their software applications. It is catching program responsibility to perform and complete usability engineering activities.

# Risk Management

## Software Classification

EDS is not a medical device however it is developed with Class A rigor.

## Safety & Cyber security Risk Management Plan

This software delivers UX elements/widgets to be reused in catching programs. There is no safety & cyber security risk directly associated with this software. It is the responsibility of catching programs of EDS to perform Safety & Cybersecurity risk management planning and assessment for their products.

## Personnel Qualifications

All personnel assigned to participate in the risk management actives are required to be trained in Risk Management. Personnel performing risk management activities have knowledge and experience appropriate to the tasks assigned.

Appropriate personnel qualification records are maintained per procedure identified in the applicable site quality plan.

## FMEA

EDS deliverables are UX/UI building blocks for catching programs to use in building their software applications. It contains styles, themes and basic elements and not a complete application. Therefore, it doesn’t introduce any failure modes. FMEA is not applicable for EDS.

# Software of Unknown Pedigree (SOUP)

## SOUP Identification

Within the context of this appendix, SOUP includes all tools, scripts, libraries, and software units that are acquired by EDS team from vendors outside of GEHC, which is part of the codebase that is shipped with a product.

Complete list of SOUP items can be found in EDS SOUP document available in QMS folder of GitHub.

## SOUP Activities

### Evaluation

Initial evaluation of the SOUP will be performed before the decision is made to take the SOUP into the program and start development and integration activities. Evaluation includes software procurement, license review, and informal functional/Integration testing.

### Development

Development activities include authoring and placing SOUP under revision control.

### Integration

Integration activities include formally adding the SOUP to the product software build and performing informal SW integration testing.

# Design Verification Plan

The design verification plan identifies the activities to confirm that the system design outputs meet the design input requirements.

This plan outlines the verification effort for the scope of work for the Edison Design System (EDS) program which is a UI component library built as web components

In addition, in-process verification, software unit acceptance criteria, and software item integration are conducted as an integral part of the software development process, as described below.

## Product Specifications

Verification test procedures are developed using the updated performance, functional, subsystem, component, and platform requirements as well as the individual defects identified in Section 2.

## Entrance Criteria

The following items will be completed prior to the start of verification testing:

* Test Processes are defined
* Specifications are defined

## Verification of Design Outputs

The following table identifies the Design Outputs and the method(s) in which each will be verified:

| Finished Design Output | Inspection | Test | Demonstration | Analysis |
| --- | --- | --- | --- | --- |
| [Final software - website] - <https://edisondesignsystem.com> |  |  |  |  |
| [Service and installation procedures] – Documented in <https://edisondesignsystem.com/consuming-edison-design-system> |  |  |  |  |
| [Release notes] <https://edisondesignsystem.com/eds-release-notes> |  |  |  |  |

## Verification Methods

The following identifies the verification methods that will be used:

|  |  |
| --- | --- |
| **Integration (Test)**  Evaluation of software integration to confirm that the Software units have been integrated into Software items. | |
| Procedure: | Automation and/or manual testing. |
| Evidence: | <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-verification-summary-report.xlsx> |

|  |  |
| --- | --- |
| **Exploratory Testing (Test)**  Unscripted testing performed by experts (e.g. users, designers, etc.) to challenge the design and expose anomalous behaviors not necessarily related to documented specifications. | |
| Procedure: | Manual |
| Evidence: | <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-verification-summary-report.xlsx> |

|  |  |
| --- | --- |
| **Manual Testing (Test)**  Verification of system requirements, architecture, interfaces, and detailed design by means of manual tests. | |
| Procedure: | Manual testing against functional specifications |
| Evidence: | <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-verification-summary-report.xlsx> |

## Test Cycles

During verification, a new test cycle will start each time a new build is introduced (two weeks sprints). When a new test cycle is required, an impact assessment will be conducted to determine what previous verification results are still valid. The impact assessment determines the level of testing required in the new test cycle based upon what has changed since the previous build.

## Resources

Verification will be performed by qualified personnel. Roles may include, but are not limited to the following:

* QA/UX Verification Engineer – manual and e2e automation testing

To ensure independence from design, software developers will not participate in Verification testing of their own code. Verification testing will be performed by qualified personnel. The actual test personnel (internal and/or external) will be documented within the test results.

To be eligible to perform verification testing, the training identified in the quality management system (QMS) is required. No additional training or qualification is required beyond the normal job function for this verification effort.

## Equipment and Environment

Non-product hardware and software tools that are required to support the verification process, along with evidence of their qualification/validation status, are identified in: The following table identifies the product configurations that will be verified based on the Product Configuration Master:

|  |  |  |  |
| --- | --- | --- | --- |
| Browser | Version | Platform | Comments |
| Chrome | 79.0.3945.79 | Mac OSX, Windows | Refer to verification summary report:  <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-verification-summary-report.xlsx> |
| Firefox | latest | Mac OSX | Planned for future releases |
| Edge | latest | Windows | Planned for future releases |
| IE11 | latest | Windows | Planned for future releases |

Based upon an assessment of end user configurations, the above representative configurations were selected.

## Acceptance Criteria

The acceptance criteria for the completion of Design Verification are as follows:

* Planned verification test procedures have been successfully executed or justified in the Design Verification Report.
* All Defects have been addressed per the GEHC Design Controls System Problems Report Work Instruction.

## Software Unit Acceptance Criteria

Software Unit Acceptance Criteria define the criteria and characteristics against which a unit is evaluated for integration readiness. Unit Acceptance Criteria are organized and described in the two tables that follow, along with the defined method(s) for verifying each.

|  |  |
| --- | --- |
| **Unit Acceptance Criteria** | **Verification Method** |
| Code conforms to non-functional requirements | Technical Design Review  Code Review  Automated Static Code Analysis  Automated Unit Testing  Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Code free of contradiction with interface design | Technical Design Review  Code Review  Automated Static Code Analysis  Automated Unit Testing  Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Code conforms to prescribed programming procedures & coding standards | Technical Design Review  Code Review  Automated Static Code Analysis  Automated Unit Testing  Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

## Integrating Software Items

EDS components are developed independently and integrated with the consuming product. As appropriate, with the technical team’s decision, Integration testing will be performed in the form of exploratory testing with mockup examples.

This activity does not replace integration verification into the Catching program product which is their responsibility**.**

# Design Validation

This program does not produce a commercialized product. Catching programs will be responsible for validation of all user requirements connected to applications and use of the elements/widgets consumed from EDS.

# Design Transfer

**EDS** will not be sold independently as a standalone product. It will be delivered to an end customer in conjunction with an application by a product team. As such, all design transfer and service activities of the applications and the **EDS** will be the responsibility of the catching program product teams. This includes but is not limited to all commercial activities, manufacturing design transfer, service validation, service instructions, and operator manuals delivered.

# External Evaluation

As part of this NPI we are continuously evaluating EDS elements/widgets in the application context with early adaptors doing proof of concepts for their new product initiatives. For these activities we have been releasing EDS as Alpha and Beta releases.

The result of these activities will be documented in EDS Verification Report: DOC2347363

# Technical Review Schedule

TDRs are conducted as needed. Evidence will be stored in GitHub, refer to DHF Index: <https://github.build.ge.com/gehc-ux/eds/blob/master/qms/eds-dhf-index.xlsx>

# Review and Approval Requirements

The following sub-sections define the owners (O) as well as the required reviews (R) and approvals (A) for each DHF artifact

## Design Planning & Design Inputs

|  | Program Manager | Engineering Manager | QA Representative | Product Manager |
| --- | --- | --- | --- | --- |
| DHF Index | O |  | A |  |
| Software Development Plan (SDP) | R | A/O | R |  |
| SOUP | R | A/O |  |  |
| Requirement Specifications | A |  | R | O |
| Requirements Traceability | O |  | A |  |

## Regulatory Documents

N/A for EDS

## Design Outputs

|  | Program Manager | Engineering Manager | Product Manager |
| --- | --- | --- | --- |
| EDS Bundle | A | A | O/R |
| Release Notes | A |  | O |

## Design Reviews

|  | Program Manager | Product Owner | Engineering Manager | QA Representative | RA Representative | Architect | Supplier Quality Engineer | Independent Reviewer | Business Leader | Engineering Leader | Sourcing Leader | Service Leader | Manufacturing Leader | Business QA/RA Executive |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Formal Design Reviews | O | R | R | R | R |  | R | R | R | R | R | R | R | A |
| Incremental Design Reviews | A |  | A | A |  |  |  |  |  |  |  |  |  |  |
| Technical Design Reviews |  |  | A |  |  |  |  |  |  |  |  |  |  |  |

# Revision History

| Revision | Reason For Change | Author |
| --- | --- | --- |
| 1 | Initial revision | Aluru, Prasad |