

**Software Development Process**

**Version 1.7**

**Document Version History**

| **Version Number** | **Revision Date** | **Summary of Changes** | **Modifier** | **Approved by** | **Approval Date** |
| --- | --- | --- | --- | --- | --- |
| 1.0 | 05/03/2019 | Initial release | Jake Vose | Claude Shacklett | 05/07/2019 |
| 1.1 | 03/26/2020 | Updated SASS Guidelines | Shreyasi Rudra | Claude Shacklett | 03/26/2020 |
| 1.2 | 08/21/2020 | Edited document; added Confidentiality Statement and Legal Disclaimer | Dayo Akinwande | Larissa Khon | 09/29/2020 |
| 1.3 | 02/25/2021 | Annual Review 2021 | Claude Shacklett | CPG | 02/26/2021 |
| 1.4 | 05/27/2021 | Updated QMS Logo | Shreyasi Rudra | CPG |  |
| 1.5 | 04/14/2022 | Annual Review 2022 | Claude Shacklett | CPG | 04/14/2022 |
| 1.6 | 02/06/2023 | Annual Review 2023 | Juan Alvarado | CPG | 02/06/2023 |
| 1.7 | 05/06/2024 | Annual Review 2024 | Claude Shacklett | CPG | 05/06/2024 |

**Table of Contents**

[**CONFIDENTIALITY STATEMENT AND LEGAL DISCLAIMER**](#_4k8hhgjefndy) **4**

[**1. Objective**](#_r0nq8sqs7rmj) **5**

[**2. Scope**](#_1fob9te) **5**

[**3. Acronyms and Definitions**](#_2et92p0) **5**

[**4. Entry Criteria**](#_zhf7945k78n2) **6**

[**5. Inputs**](#_3dy6vkm) **6**

[**6. Process Description**](#_4d34og8) **6**

[**7. Permitted Tailoring**](#_32s7nhsu5t92) **10**

[**8. Measures**](#_3rdcrjn) **10**

[**9. Output**](#_ya9uucjm8jef) **11**

[**10. Exit Criteria**](#_lnxbz9) **11**

[**11. References**](#_35nkun2) **11**

# 

# CONFIDENTIALITY STATEMENT AND LEGAL DISCLAIMER

The information contained in this document is confidential, privileged, and only for the information of the intended recipient and may not be used, published, or redistributed without the prior written consent from CTAC. If you are not the intended recipient, you may not disclose or use the information in this document in any way. Any review, retransmission, dissemination, or other use of, or taking any action in reliance upon this information by persons or entities other than the intended recipient is prohibited. If you got access or received this document in error, please contact the owner and delete the document from all electronic devices.

# 1. Objective

The objectives of the Software Development process are to support the full software development life cycle and include a detailed plan for how to develop, alter, maintain, and replace a software system. It involves several distinct stages, including planning, design, building, testing, and deployment. It also relates to choosing a specific SDLC model, which includes the Waterfall, Spiral, and Agile methodologies.

# 2. Scope

The Process is applicable to all CTAC software development projects. The Scope of Software Development Process includes:

* Requirements Development & Management
* Technical Solution (Design; Code; Test)
* Product Integration (Build; Deploy)
* Peer Reviews of artifacts and deliverables
* Verification & Validation (Unit & System Testing)

# 3. Acronyms and Definitions

| **Acronym/Definition** | **Description** |
| --- | --- |
| AT | Architect |
| PO | Product Owner |
| CC | Configuration Controller |
| DEV | Developer |
| TL | Tech Lead |
| PM | Project Manager / Scrum Master |
| BA | Business Analyst |
| PQAA | Process Quality Assurance Analyst |
| IPA | Internal Process Auditor |
| SA | System Administrator |
| TE | Test Engineer |
| TW | Technical Writer |
| CPG | CTAC Process Group |
| SDP | Software Development Process |
| SDLC | Software Development Life Cycle |
| UXD | User Experience Designer |
| SASS | Syntactically Awesome Style Sheet |

# 4. Entry Criteria

* Agreement with Customer
* Scope of Work in the Proposal
* Kick-Off Meeting
* Potential Requirements and Domain Knowledge
* Application Documents (if any)
* Software Technical Documents and Standards (if any)

# 5. Inputs

* Requirements Provided by Customer
* Statement of Work/Contract

# 6. Process Description

| SN | | | | | | Description | | | Record | | | | Responsibility | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements Feasibility Study** | | | | | | | | | | | | | | |
| 1. | | | | | | Identify high-level project goals and critical requirements (Functional and Non-Functional) for the system | | | Task Tracker – Product Backlog, Requirements Elicitation Meeting Minutes | | | | PM, BA, and TL | |
| 2. | | | | | | Work with different stakeholders to find out the application domain and services that the system should provide and system operational constraints and dependencies | | | Task Tracker – Product Backlog; Requirements Elicitation Meeting Minutes | | | | PM, BA, and TL | |
| 3. | | | | | | Product Brainstorming sessions for domain analysis | | | Task Tracker – Product Backlog, Requirements Elicitation Meeting Minutes | | | | PM, BA, and TL | |
| 4. | | | | | | Study the process through manuals & documents (if exist) | | | Task Tracker – Product Backlog | | | | PM, BA, and TL | |
| 5. | | | | | | Raise questions and seek clarifications | | | Task Tracker – Comments; Requirements Elicitation Meeting Minutes; Requirements Clarification Log | | | | Scrum Team | |
| 6. | | | | | | Develop user stories for potential functional requirements | | | Functional Requirements Document, Task Tracker – Product Backlog, Use Cases | | | | Scrum Team | |
| 7. | | | | | | Identify and document non-functional requirements (e.g., Performance, Security, Statutory and Regulatory Requirements, Interface, etc.) | | | Non-functional/Technical Requirements Document, Task Tracker – Product Backlog | | | | Scrum Team | |
| 8. | | | | | | Identify the acceptance criteria | | | Task Tracker – Product Backlog | | | | Scrum Team | |
| 9. | | | | | | * Analyze requirements to ensure that they are clear, complete, and unambiguous; * Update the design document and close defects | | | Task Tracker – Product Backlog, Requirements Review Checklist, Review Defect Analysis | | | | Scrum Team | |
| 10. | | | | | | Prioritize the requirements based on the business value | | | Task Tracker – Product Backlog | | | | Scrum Team | |
| 11. | | | | | | Conduct a user story workshop to define a set of minimum releasable features for the sprint cycle | | | Task Tracker – Product Backlog, Workshop Meeting Minutes | | | | SM/Scrum Team | |
| **Prototyping (Optional)** | | | | | | | | | | | | | | |
| 12. | | | | | | Identify requirements for prototype | | | Requirements Document, Task Tracker – Product Backlog | | | | Scrum Team | |
| 13. | | | | | | Develop a prototype and evaluate | | | UI Mockups/Wireframes | | | | Scrum Team | |
| 14. | | | | | | Identify potential functional requirements | | | Customer Feedback, Functional Requirements Document, Task Tracker – Product Backlog | | | | Scrum Team | |
| 15. | | | | | | Develop use cases/User stories for potential functional requirements | | | Task Tracker – Product Backlog | | | | Scrum Team | |
| **Technical Design** | | | | | | | | | | | | | |
| 16. | | Identify the design alternatives | | | | | DAR Form | | | Scrum Team – Designer | | | |
| 17. | | Prepare the design specifications | | | | | Technical Design Document | | | Scrum Team – Designer | | | |
| 18. | | Identify the reusable components | | | | | Technical Design Document | | | Scrum Team – Designer | | | |
| 19. | | Review the design document and record defects | | | | | Design Review Checklist | | | Scrum Team – Designer | | | |
| 20. | | Update the design document and close defects | | | | | Technical Design Document, Design Review Checklist, Review Defect Analysis | | | Scrum Team – Designer | | | |
| 21. | | Baseline Design document | | | | | Approved Technical Design Document | | | Scrum Team – Designer | | | |
| **Test Case Design** | | | | | | | | | | | | | |
| 22. | | | Study requirements and technical design documentation and seek clarifications | | | | Task Tracker – Comments | | | Scrum Team – Tester | | | |
| 23. | | | Review and update the Test Plan based on Technical Specifications | | | | Verification & Validation Plan; Use Cases and Test Cases | | | Scrum Team – Tester | | | |
| 24. | | | Map Requirements to Test Cases | | | | Task Tracker – Epic/User story – Parent/Child Dependency or RTM | | | Scrum Team – Tester | | | |
| 25. | | | Prepare testing strategies and Test Cases (Integration & System) based on each user story | | | | Test Cases | | | Scrum Team – Tester | | | |
| 26. | | | Review & Update Test Cases and close defects | | | | Test Case Review Checklist | | | Scrum Team – Tester | | | |
| 27. | | | Approve Test Cases | | | | Test Cases Approval Mail/Note | | | Scrum Team – Tester | | | |
| 28. | | | Review and update the Test Plan based on Test Scripts | | | | Verification & Validation Plan; Use Cases and Test Cases | | | Scrum Team – Tester | | | |
| 29. | | | Update Requirements Traceability Matrix with Test Script information | | | | RTM or Task Tracker – Epic/User story – Parent/Child Dependency | | | Scrum Team – Tester | | | |
| 30. | | | Baseline the Test Cases | | | | Approved Test cases | | | Scrum Team – Tester | | | |
| **Coding** | | | | | | | | | | | | | |
| 31. | | | | Develop/modify code to confirm to design and coding standards | | | Source Code; Coding Standards | | | Scrum Team – Developer | | | |
| 32. | | | | Verify code for correct execution | | | Debug/Unit Testing | | | Scrum Team – Developer | | | |
| 33. | | | | Review code and log defects  Business Logic  Database Design & Access  Modularization & Reusability  Code Optimization  Performance Tuning  User Experience  Hard Coding  Inline Comments  Inline Documentation | | | Code review log pulled from BitBucket/GitHub,  Code Review Checklist, Review Defect Analysis, Issue Tracker | | | Scrum Team – Developer/BA/PM | | | |
| 34. | | | | Update code to close defects | | | Source Code, Review Defect Analysis, Issue Tracker | | | Scrum Team – Developer | | | |
| 35. | | | | Frontend Development SASS Guidelines | | | Source Code/Directory Structure | | | Scrum Team – Developer | | | |
| **Stage Build Test/Validation** | | | | | | | | | | | | | |
| 36. | | | | | Deploy application code to stage | | Source Code | | | Scrum Team – Developer | | | |
| 37. | | | | | Perform system and integration testing | | Test Results/Report | | | Scrum Team – Tester | | | |
| 38. | | | | | Record test results and log defects | | Task Tracker – Defect Log | | | Scrum Team – Tester | | | |
| 39. | | | | | Fix the defects | | Task Tracker – Defect Log, Test Results/Report | | | Scrum Team – Tester | | | |
| 40. | | | | | Obtain approval to deploy code to production | | Approval Mail/Note | | | Scrum Team – Tester | | | |
| 41. | | | | | Perform defect analysis | | Task Tracker – Defect Report, Test Defect Analysis | | | Scrum Team – Tester | | | |
| **Production Build Test/Validation** | | | | | | | | | | | | | |
| 42. | | | | | Evaluate the build | | Pre-build Checklist, Configuration Audit Checklist, | | | Scrum Team – Developer | | | |
| 43. | | | | | Package and deploy production build | | Release Notes, Post-build Checklist | | | Scrum Team – Developer | | | |
| 44. | | | | | Conduct smoke tests for verifying build stability | | Smoke Test results | | | Scrum Team – Tester | | | |
| 45. | | | | | If unstable, conduct a rollback plan | | Release Notes - Rollback Plan section | | | Scrum Team – Developer/SA | | | |
| **User Documentation** | | | | | | | | | | | | | |
| 46. | | | | | Study the requirements, technical design, and the software system and seek clarification | | Task Tracker – Comments | | | | | Scrum Team | |
| 47. | | | | | Prepare/modify user manual | | User Manual – Baseline Doc | | | | | Scrum Team | |
| 48. | | | | | Review User Documentation | | User Manual review comments | | | | | Scrum Team | |
| 49. | | | | | Update traceability matrix with documentation information | | RTM or TaskTracker – Parent/Child | | | | | Scrum Team | |
| 50. | | | | | Obtain approval | | Approval Mail/Note | | | | | Scrum Team | |
| 51. | | | | | Baseline documentation | | Configuration Status Accounting Register | | | | | Scrum Team | |
| 52. | | | | | Conduct a PQA review of coding, unit testing, and documentation activities and report quality issues | | PQA Review Checklist/Report | | | | | PQA | |
| **Managing Requirements Creep and Changes** | | | | | | | | | | | | | |
| 53. | | Identify and track the open and evolving requirements | | | | | | Change Request Form (CR Form), Change Request Log (CR Log) | | | | PM | |
| 54. | | Record the open and evolving requirements issues and seek clarifications | | | | | | CR Form, CR Log | | | | PM | |
| 55. | | Study and understand requests for adding a new requirement or changing the existing ones (of previous iterations). Analyze their impact | | | | | | CR Impact Analysis section in CR Form or in Issue Tracker | | | | PM | |
| 56. | | Assess the impact on application scope, effort, and schedule | | | | | | CR Impact Analysis in CR Form or in Issue Tracker, Project Schedule | | | | PM | |
| 57. | | Obtain approval from CCB | | | | | | CR Log | | | | CCB | |
| 58. | | Review updated technical specifications and record defects | | | | | | Review Records: revision table, Design Review Checklist, Review Defect Analysis | | | | PM | |
| 59. | | Update technical specification and close defects | | | | | | Technical Specification/ Design Document, Review Defect Analysis | | | | PM | |
| 60. | | Approve updated technical specification | | | | | | Approved Technical Specification/ Design Document | | | | PM | |
| 61. | | Re-baseline the revised requirement specification document | | | | | | Configuration Status Accounting Register | | | | PM | |
| **PQA Review** | | | | | | | | | | | | | |
| 62. | | Conduct a PQA review of requirements phase activities and report quality issues | | | | | | PQA Audit Checklist/Report | | | | PQA | |
| 63. | | Analyze and resolve quality issues | | | | | | Non-conformances fixes, Follow-up PQA Audit | | | | Scrum Team, PQA | |

# 7. Permitted Tailoring

* Refer [Tailoring Guidelines](https://docs.google.com/document/d/1zgUhPRrt5T2k56NzlGZHoQc9fsef6ciBE_TPln0nVRI/edit?usp=sharing)

# 8. Measures

* Effort spent on requirements gathering
* Effort spent on design documentation
* Number of defects found in peer reviews (Requirements, design code, and testing)
* Effort spent on peer reviews (Requirements, design code, and testing)
* Defects found in testing
* Effort spent on testing

# 9. Outputs

* Requirements Elicitation Meeting Minutes
* Requirements Clarification Log
* Approved Functional Requirements Document
* [**Requirements Review Checklist**](https://drive.google.com/drive/folders/1P4oB2-1XB7WOQlHDk1JhE_Mzvt7GtuW-) (every sprint for DEV. projects; every quarter for O&M projects)
* Use Cases
* Non-functional/Technical Requirements Document
* Design Prototypes, UI Mockups/Wireframes (optional)
* Product Backlog (Epic >> User Stories in Task Tracker)
* Sprint Backlog (Epic >> User Stories in Task Tracker)
* Approved Technical Design Document
* [**Design Review Checklist**](https://drive.google.com/drive/folders/1P4oB2-1XB7WOQlHDk1JhE_Mzvt7GtuW-)(quarterly, as needed)
* Source Code
* [**Code Review Checklist**](https://drive.google.com/drive/folders/1P4oB2-1XB7WOQlHDk1JhE_Mzvt7GtuW-)(every sprint/cycle, as needed)
* Unit Test results - task tracker comments
* Requirements Traceability Matrix
* System Test Cases/Reports
* [**Test Case Review Checklist**](https://drive.google.com/drive/folders/1P4oB2-1XB7WOQlHDk1JhE_Mzvt7GtuW-) (every sprint/cycle for DEV. projects; every quarter for O&M projects)
* Defect Analysis Report
* [**Pre-build Checklist**](https://drive.google.com/drive/folders/1P4oB2-1XB7WOQlHDk1JhE_Mzvt7GtuW-) (every sprint/cycle/prod push, as needed)
* [**Configuration Audit Checklist**](https://drive.google.com/drive/folders/1B19DdYvlQrtucBppJv45Ef_Fv884oxox) (every sprint/cycle/release, as needed)
* Release Notes
* [**Post-build Checklist**](https://drive.google.com/drive/folders/1P4oB2-1XB7WOQlHDk1JhE_Mzvt7GtuW-)(every sprint/cycle/prod push, as needed)
* Post-deploy smoke testing
* Rollback Plan
* User Manual
* Configuration Status Accounting Register
* Change Request form with Impact Analysis
* Change Request Log
* Review Issue and Defect Analysis
* PQA Audit Report
* Updated Project Schedule (WBS)
* Process Improvement Suggestions to CPG

# 10. Exit Criteria

* Acceptance from the customer for production release

# 11. References

**CMMI V 2.0**

* Requirements Development & Management
* Technical Solution
* Product Integration
* Peer Reviews
* Verification & Validation