|  |  |  |  |
| --- | --- | --- | --- |
| Authoring Site | Document Type - Subtype | | Document Unit |
| MM-France-BPS Operations | Procedural Document - Standard Operating Procedure | | N/A |
| Referenced Documents | | Special Notes | |
| *SDP Mango number* | | N/A | |

Contents

[**1.0** SUBJECT - **PURPOSE** – SCOPE 2](#_Toc487212627)

[SCOPE 2](#_Toc487212628)

[**2.1.1** **Username** 5](#_Toc487212629)

[**2.1.2** **Program Project Name** 5](#_Toc487212630)

[**2.1.3** **Project Key** 5](#_Toc487212631)

[**2.1.4** **Iteration number** 5](#_Toc487212632)

[**2.1.5** **Version Release to Customer** 6](#_Toc487212633)

[**2.1.6** **PI Version** 6](#_Toc487212634)

[**2.1.7** **Baseline** 6](#_Toc487212635)

[**2.1.8** **Review** 7](#_Toc487212636)

[**2.1.9** **External QMS Artefact** 7](#_Toc487212637)

[**2.1.10** **Risk Item** 7](#_Toc487212638)

[**2.1.11** **Documentation Item (Requirement, Specification, Test cases)** 8](#_Toc487212639)

[**2.1.12** **Filters** 8](#_Toc487212640)

[**2.1.13** **Export templates** 8](#_Toc487212641)

[**2.1.14** **Tag** 9](#_Toc487212642)

[**2.1.15** **Test Cycle** 9](#_Toc487212643)

[**2.1.16** **Test Group** 9](#_Toc487212644)

[**2.1.17** **Test Plan** 10](#_Toc487212645)

[**2.2.1** **Observation** 10](#_Toc487212646)

1. SUBJECT - **PURPOSE** – SCOPE
   1. SUBJECT
      1. Describe the naming convention used in JIRA and JAMA software according to the Software Development Plan (SDP) which is the main reference document.
   2. PURPOSE
      1. Provide a guidance for creation of new files, base lines, reviews or any item in the JAMA and JIRA software.

SCOPE

* + 1. This procedure applies to all Scrum teams in all sites of Merck Life Science [Millipore Sigma in North America] (LS) in sites below:
       - Molsheim
       - Bangalore
       - Bedford
       - St. Louis
  1. RESPONSIBILITIES
     1. The Quality Assurance is the owner of the procedure. He is responsible for maintaining and enforcing this procedure.
     2. Lead Tester, Testers, Lead Developer, Developers, Lead Architect and Product Owner are all responsible for the application of the rules and principles defined in this present SOP in the frame of functional and design specifications.
     3. The Project Quality Manager and the Quality Assurance are responsible for Quality documentation integrity assurance, for verification that traceability is duly maintained across specifications and for verification that reported results and conclusions match the requirements expressed in the present SOP and in the product specifications.
  2. DEFINITIONS

This document has been built with respect to the IEEE Standard 610.12-1990 « Standard Glossary of Software Engineering Terminology”. Any definition that would not be present here after might be found in this IEEE publication or in SDP Chapter 7.1 Acronyms & Definitions.

* 1. **SDP**: Software Development Plan is a corporate wide Quality policy document with Merck’s Quality and Validation approach for the development of Merck IOT/Automation Platform
  2. **Project Quality Manager (PQM)**: Person who is responsible of all quality aspects within the Scrum Team but is independent of the Quality Assurance Service
  3. **Quality Assurance (QA)**: Person who approves the validation approach and signs off on the Validation Master Plan, the Validation Plan and the Validation
  4. **Product Owner (PO)**: Person who defines the User Stories and prioritize them according to the market value. They can accept or reject the work done at the end of the Iteration.
  5. **Lead Tester (LT)**: Person who represents the Testers and executes tests throughout the software development process.
  6. **Tester (T)**: Person who executes tests throughout the software development process.
  7. **Lead Developer (LD)**: Person who represents the Developers and authors the code.
  8. **Developer (D)**: Person who authors the code.
  9. **Lead Architect (LA)**: Person who is responsible for defining and maintaining the structure of the solution, and ensuring that it will meet the requirements.
  10. **Design**: (IEEE) the process of defining the architecture, components, interfaces, and other characteristics of a system or component
  11. **FS**: (ISPE GAMP5) acronym standing for Functional Specifications
  12. **DT**: acronym standing for Design Tests
  13. **FT**: (ISPE GAMP5) acronym standing for Functional Tests
  14. **GAMP®**: a registered ISPE acronym standing for “Good Automated Manufacturing Practices”. “A system for producing quality equipment using the concept of prospective validation following a life cycle model. Specifically designed to aid suppliers and users in the pharmaceutical industry. Currently in revision 5
  15. **SOP**: Standard Operating Procedure
  16. **DS**: Design Specification
  17. **US**: User Stories
  18. **JAMA**: Software for Requirements, Specifications and Tests management.
  19. **JIRA**: Software for Tasks and work scheduling and following.

1. **PROCEDURE**
   1. **NAMING CONVENTION FOR DEVELOPMENT PROJECT**

Be careful: No space in Jama item name, use CamelCase (example "TestPlan") instead, this forces us to remove spaces from the attached file too.

* + 1. **Username**

Target:

Username registered in Jama and Jira:

Rules:

* For Sigma collaborators: Sigma Network Login ID
* For Merck collaborators: XMUID  
  + 1. **Program Project Name**

Target:

Program Jama Project name:

Rules:

* “Value Stream-project-Release”

Examples:

* BIOP-Bioreactor-Current
* Next Connect-Next connect-R1.00.00.00
* Each project starts with a “-Current” where the main flow is built.
  + 1. **Project Key**

Target:

Project Jama Key. It is the project unique ID that will be used to prefix items of the project in JAMA

Rules:

* Project Key (16 characters maximum) Caps should be used
* “Value Stream(XX)-project(XXXX)-Release(RXXXXXXX)”
* “Value Stream(XX)-project(XXXX)-CURRENT”

Examples:

* BP-BIOR-CURRENT
* NC-NECO-R1000000
  + 1. **Iteration number**

The number of the iteration is an increment. If we begin a new Program Increment, we have to define the new iteration number by taking the number of the last iteration and increment it by 1. Example, the PI01-Iteration4 is followed by the PI02-Iteration5 and the PI02-Iteration8 is followed by the PI03-Iteration9.

* + 1. **Version Release to Customer**

Target:

Version Release number. This is the release number displayed when the software is deployed to Customer in Jama and Jira:

Rules:

* “RA.BB.CC.DD”:
* RA is main release
* BB is Functional release
* CC is Custom release
* DD is minor correction release

Example:

* R1.00.00.01
  + 1. **PI Version**

Target:

PI Version in Jama and Jira:

Rules:

* PI Version NOT released to customer
* “RA.BB.CC.DD-PIXX”

Example:

* R1.00.00.01-PI01
  + 1. **Baseline**

Target:

Baseline in Jama:

Rules:

* “Version-PI increment-Iteration increment-Step”
* Version: version number “RX.XX.XX.XX”
* PI increment: Number of the PI “PIXX”
* Iteration Increment: Iteration increment in the PI “IterationX”
* Step: Summing up of the related step (e.g. “US approval” “FS approval”)

Example:

* R1.00.00.00-PI01-Iteration1-FT approval

* + 1. **Review**

Target:

For each review a baseline shall be created. The name of the review will have the same name as the baseline.

Rules:

* Version-PI increment-Iteration increment-Step”
* Version: version number “RX.XX.XX.XX”
* PI increment: Number of the PI “PIXX”
* Iteration Increment: Iteration increment in the PI “IterationX”
* Step: Summing up of the related step (e.g. “US approval” “FS approval”)

Example:

* R1.00.00.00-PI01-Iteration1-FT approval

* + 1. **External QMS Artefact**

Target:

All the documents (Word, Excel, Power point…) that need to be stored in JAMA.

The Jama item will have the same name as the attached file.

Rules:

* “Version-PI increment-Iteration increment-Content\_Rev”
* Version: Version number “RX.XX.XX.XX”
* PI increment: Number of the PI “PIXX” if applicable
* Iteration increment: Iteration increment in the PI “IterationX” if applicable
* Content: Summing up of the content (e.g. “USApproval” “TestPlan” “CodeReview”)
* Rev: Revision of the document “RevXX"

Examples:

* R1.00.00.00-PI01-Iteration01-CodeReview\_Rev01
* R1.00.00.00-PI01-Iteration01-TraceMatrix\_Rev01

* + 1. **Risk Item**

Target:

The risk item naming follows the same rules as all other items in Jama. No automated naming is possible.

Rules:

* The best practice is to give an understandable short name.
* This name will be a resume of the failure mode

Example:

* Pump overheat

* + 1. **Documentation Item (Requirement, Specification, Test cases)**

Target:

The documentation item naming has to be created with the following rules to be organised/sorted by the item object name. No automated naming is possible.

Rules:

* DocumentStructureItemName: Summing up of what represent the document item in the project whole documentation (e.g. “Overview”, “ObjectParent”, “ObjectChildren”, “Property”)
* “DocumentStructureItemName\_Rev Version-PIIncrement-IterationIncrement-Team ”
* Version: Version number “RX.XX.XX.XX”
* Rev: Revision of the document “RevXX"
* PIIncrement: Number of the PI “PIXX” if applicable
* IterationIncrement: Iteration increment in the PI “IterationX” if applicable

Examples:

* AnalogSensor-ModelBase\_R1.00.00.00-PI02-Sprint5
* AnalogSensor-Definition\_R1.00.00.00-PI02-Sprint5
  + 1. **Filters**

Target:

All public filters shall have a unique name to avoid use of wrong informations.

Rules:

* "XX.XX Target - Container - Criteria"
* XX.XX: Chapter number to organize the filters
* Target: Item presented as result
* Container: Location of the items
* Criteria: Nature of the filter

Examples:

* 02.01 Features - Current release - In Backlog
* 12.02 OBS - Current release - Resolved

* + 1. **Export templates**

Target:

Each document produced with JAMA shall have a unique name.

Rules:

* "Documentation system reference - Document content - Rev"
* Documentation system reference: System where the template is qualified
* Document content: What can be found in the document
* Rev: official reference of the document (Template Document Number)

Example:

* Mango - FS - 20180735,2.0

* + 1. **Tag**

Target:

Tags are used to help users to group items.

Rules:

* Must be very simple, maximum 4 words.

Be careful: No space in Jama attribute's name that can be use in filter, use CamelCase (example "RecipeEditor") instead.

Examples:

* Login
* Security
* RecipeEditor

* + 1. **Test Cycle**

Target:

The test cycle is a group of test to be played.

Rules:

* "Test Plan version number-campaign(Campaign number)-Environment-Test Group name"
* Test Plan version number: Increment each time the test plan is approved after edition.
* Campaign number: Increment for each update of the test environment.
* Environment: The environment type.
* Test Group name: The name of the Test Group run in this Test Cycle.

Be careful: No space in Jama attribute's name that can be use in filter, use CamelCase (example "FunctionalTest") instead.

Examples:

* 1.0-campaign1-Bench-FunctionalTest
* 2.0-campaign3-VM-DesignTest-EnvironmentQualification
  + 1. **Test Group**

Target:

Rules:

* "Test type-Environment-Purpose(if needed)"
* Test type: The type of test in the Test Group
* Environment: The environment type, if a specific environment is required to execute the tests in the Test Group
* Purpose: If the Test Group have a specific goal

Be careful: No space in Jama attribute's name that can be use in filter, use CamelCase (example "FunctionalTestCase") instead.

Examples:

* "FunctionalTestCase-Bench"
* "DesignTestCase-VM-EnvironmentQualification"

* + 1. **Test Plan**

Target:

Rules:

* “Version-PI increment-Iteration\_incrementStep”
* Version: version number “RX.XX.XX.XX”
* PI increment: Number of the PI “PIXX”
* Iteration Increment: Iteration increment in the PI “IterationX”
* Step: Summing up of the related step "\_TestPlan"

Be carefull: No space in Jama item name, use CamelCase (example "TestPlan") instead.

Example:

* R1.00.00.00-PI01-Iteration1\_TestPlan
  1. **NAMING CONVENTION FOR ENABLER PROJECT**
     1. **Observation**

Target:

Rules:

* “EnablerReference\_ObservationDescription”
* EnablerReference: the reference name of the enabler
* Description: Short description of the observation without space "\_Description"

Be carefull: No space in Jama item name, use CamelCase (example "TestPlan") instead.

Example:

* TestEnvironment\_MissingSoftware

**REVISION SHEET**

| **Date** | **Version** | **Nature of Changes** | **Changed By** |
| --- | --- | --- | --- |
| 26-Apr-17 | 1.0 | Initial Release |  |
|  |  |  |  |
|  |  |  |  |

**ANNEXE 1 / APPENDIX 1 DOC N°**