



**BJIT Ltd.**

# **DEVELOPMENT PROCESS MANUAL**

**Abridged Version – 1.7**

**2021-08-02**

**YOUR NEXT DESTINATION  
OF SOFTWARE OUTSOURCING**

## Document Classification

*This document has been classified as **RESTRICTED** document as per BJIT's Information Security Management System and can only be shared with the following members / teams:*

- **BJIT employees**

# CONTENTS

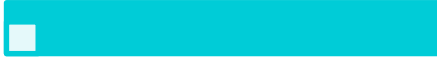
01	Process Framework	4
02	Annexure	34

# DEVELOPMENT PROCESS FRAMEWORK

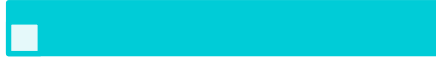
Abridged Version

# Process framework

## Project Initiation

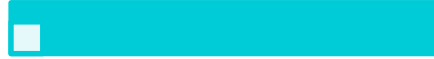


## Project Management



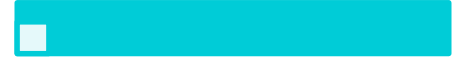
- ☐ Integrated project management
- ☐ Project planning
- ☐ Project monitoring & control
- ☐ Requirement management
- ☐ Risk management

## Engineering process



- ☐ Design
- ☐ CUT
- ☐ Product Integration Process
- ☐ Testing
- ☐ Review Process

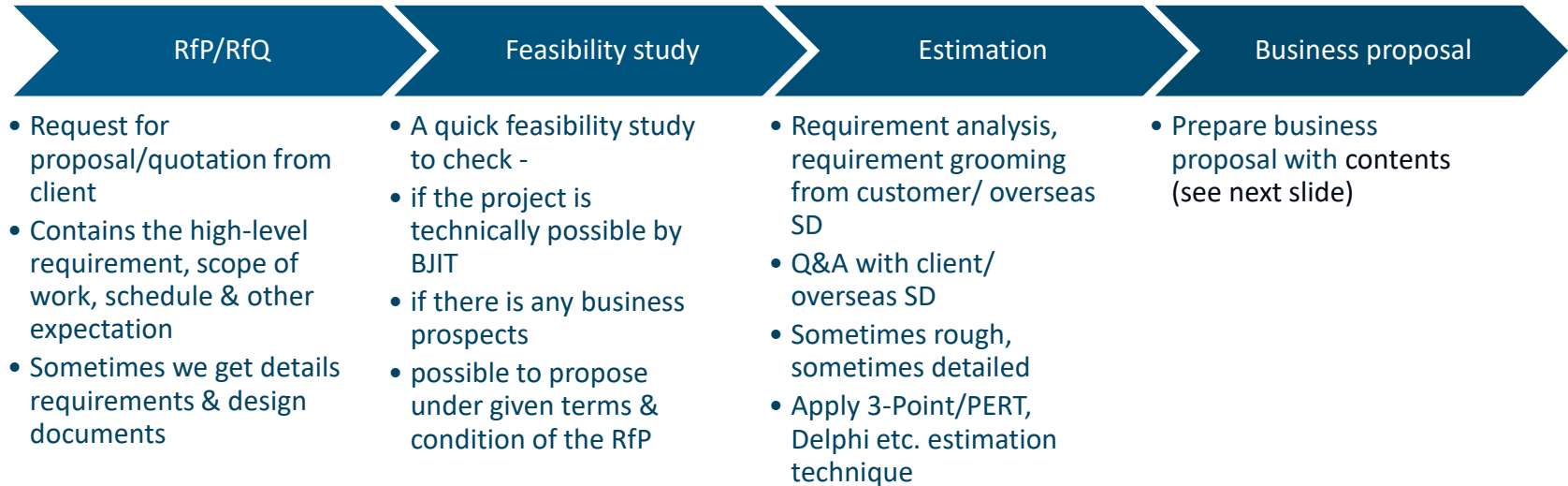
## Support Process



- ☐ Configuration Management
- ☐ Process & Product Quality Assurance
- ☐ Decision Analysis & Resolution
- ☐ Measurement Analysis
- ☐ Organizational Audit (PPQA,SEPG,OT)
- ☐ Release & Acceptance

# 1. Project initiation (1/2)

- Typically, a project starts with this activity. Sometimes called discovery phase.
- BJIT sales receive project leads & collect RfP/RfQ from client and SD prepares estimation & proposal
- The SD PIC of this phase is generally department or division head
- Sometimes, project team supports the PIC with the estimation & proposal



# 1. Project initiation (2/2)

## 1. Overview

- The background of the RfP/RfQ. The contents of the proposal

## 2. In/Out scope of work

- What are the activities we have to do and we don't e.g. SRS, UX, DD, Coding, Unit/Integration/Functional/Security/Performance testing, Deployment on prod

## 3. High level architecture

- Optional. If added, contains very high-level system image in block diagram with communication

## 4. Effort & Cost

- Estimated effort breakdown as per RfP request & cost of those breakdown. Cost is calculated by sales from RAP prepared by SD

## 8. Deliverables

- Deliverables for each activities within the scope

## 7. Milestones

- Detail milestones with scope, due date
- Sometimes finalized later

## 6. Dev methodology

- Agile or waterfall
- Sometimes finalized later

## 5. Project schedule

- From high to detail level depending on RfP. Need to mention milestones/releases
- Sometimes finalize later

## 9. Tools & technologies

- Tools & technologies will be used in the project including project management tools
- Sometimes finalize later

## 10. Communication mgt

- Different meetings schedule, participant, agenda. Reporting schedule etc.
- Sometimes finalize later

## 11. Team structure

- Team organization chart
- Sometimes finalize later

## 12. Assumptions & pre-conditions

- Assumptions related to scope, dependencies, tools & technologies etc.

## 16. Company profile

- Mandatory if RfP ask. BJIT strengths, services, technology & business vertical penetration, resource profile etc.

## 15. Relevant project exp

- Mandatory if RfP ask. Project reference relevant to this project. Basic info about the project e.g. tile, domain, client, size, tech

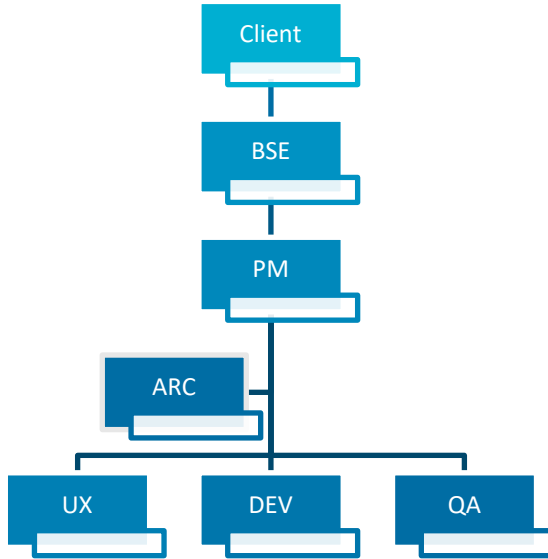
## 14. Roles & responsibilities

- Responsibilities of each project member
- Sometimes finalize later

## 13. Risks

- Risks related to any initial decisions

## 2. Project management [IPM, PP, PMC] (1/6)



- Div/Dept head form the team based on RAP
- Div/Dept head HO/KT the project details to team learned during estimation & proposal preparation
- Div/Dept head arrange project kick/off and introduce project team with different stakeholders

- Confirm different processes like requirement, change & risk management, coding, testing etc. with the client
- Ensure any process tailoring with client consent
- Finalize project tools & technologies, development methodologies

- Almost all the items of the project plan suppose to be covered in business proposal e.g. scope, schedule, quality metrics etc. Define all in this phase
- Create redmine, gerrit, Alfresco/FTP project. Client will share client tool access
- Ensure required H/W & S/W for team & project

- Create product, development, project & quality management in redmine or confluence
- Create product backlog, project roadmap, WBS, sprint planning in redmine/JIRA based on selected development methodology

- Project manager will ensure above activities with the support of BSE & Div/Dept head
- Depending on client, contract (LABO vs project) and the project itself, above activities may vary. For LABO contract, customer will define most of the items for the offshore
- Most of the items will be discussed & finalized with client during project initiation, some of the items will be finalized during project planning
- All the processes details will be covered separately in this document
- Please check the annexure very carefully for PPQA checklist, sample contents of project documents in redmine or confluence



## 2. Project management [IPM, PP, PMC] (2/6)



BJIT follows scrum in almost all projects. There are some variations between teams as per client



Few cases like migration, traditional project BJIT follows Waterfall. Even then, a variation of scrum is followed within a waterfall phase

### Project management (BJIT)

#### Remine



Product, development, project, task & quality management

#### Gerrit



Source code and their review management

#### Alfresco



Document management system

### Project management (Client)

#### JIRA



Task & quality management system

#### Confluence



Product, development, project, task & quality management

#### Bitbucket



Source code and their review management

### UX Design

#### Adobe XD



BJIT does UX design with this. BJIT also receives UX from client in XD

#### Photoshop



Traditional UX design tool. In some cases, BJIT still uses this

#### Figma



Online based UX design tool. Both BJIT & its client use this tool

### Other

#### MS Excel



BJIT use excel to design test matrix, test case

#### Astah



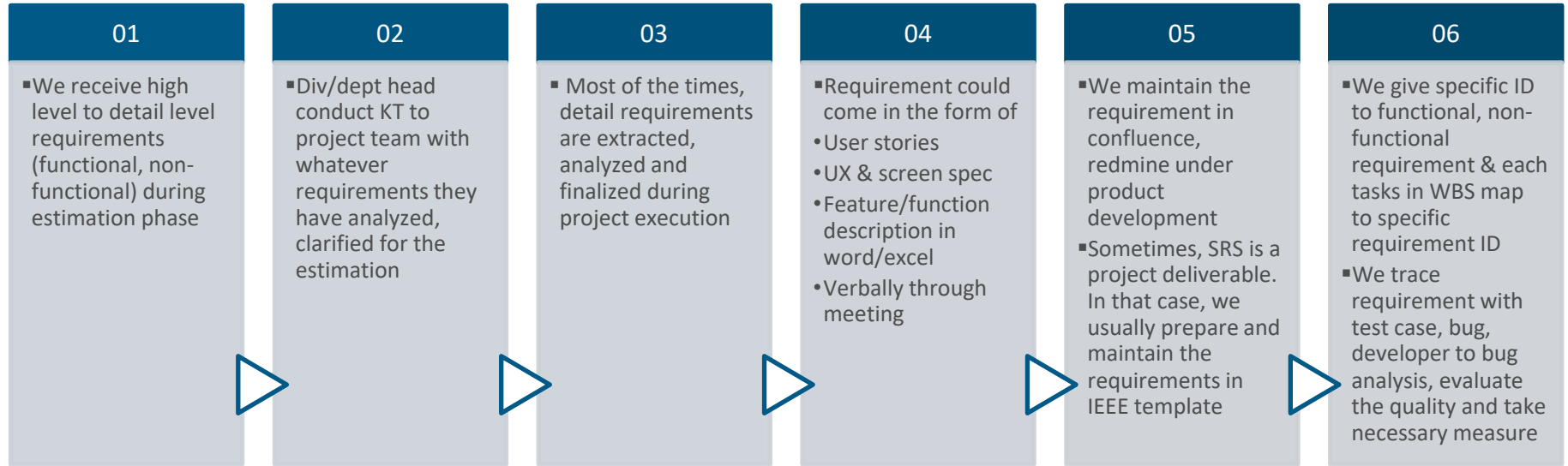
BJIT use Astah to design UMLs like SD, CD, activity etc. diagram

#### Jenkins

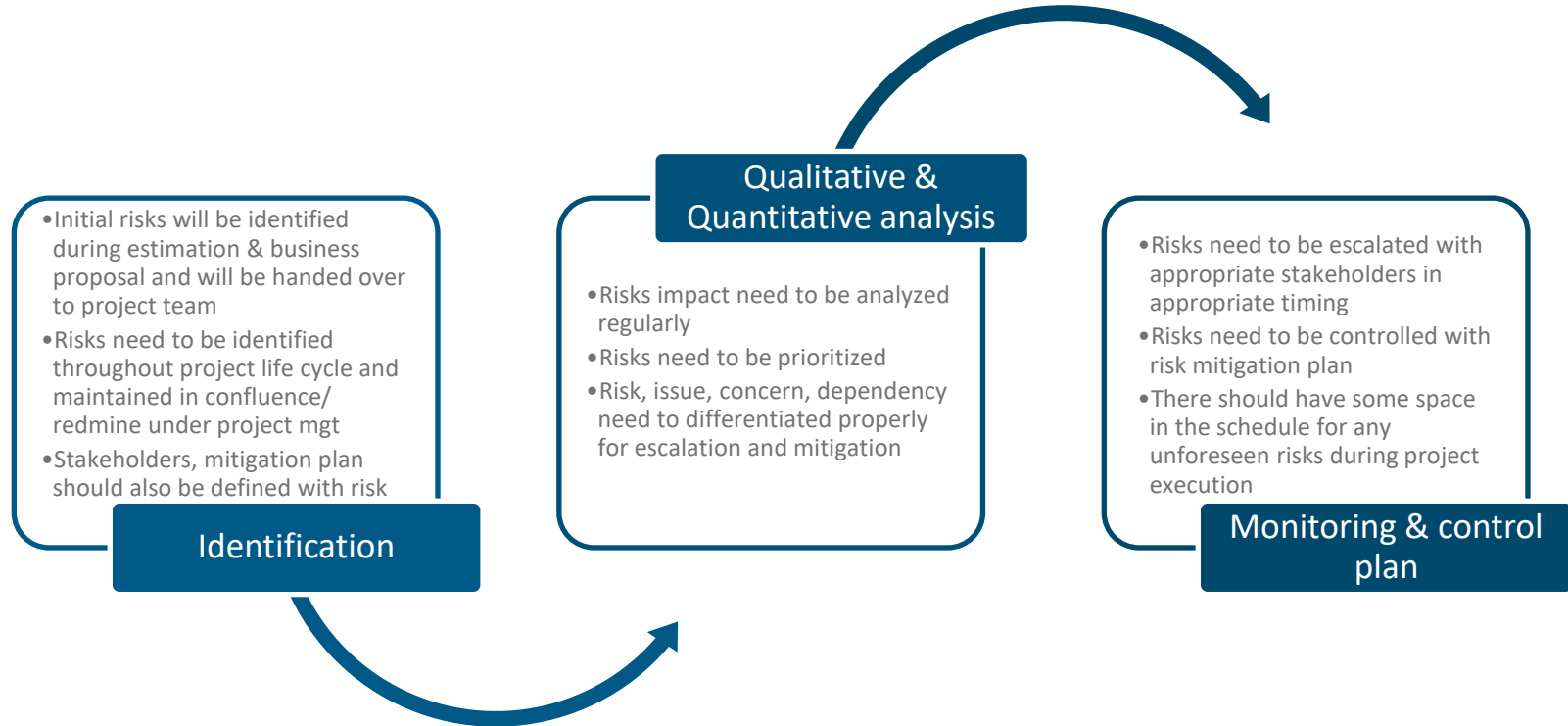


BJIT use Jenkins for CI/CD

## 2. Project management [REQM] (3/6)

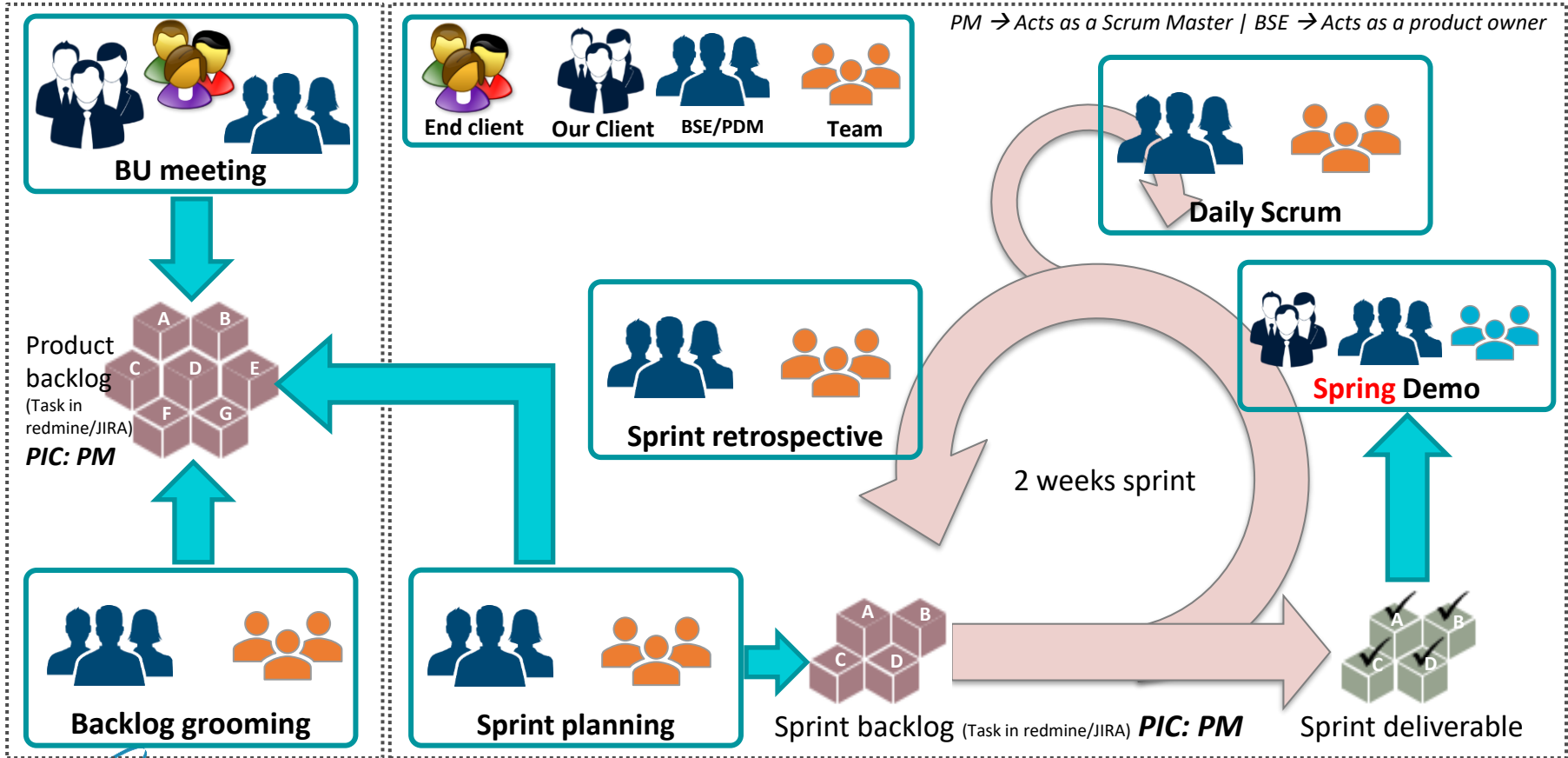


## 2. Project management [RSKM] (4/6)

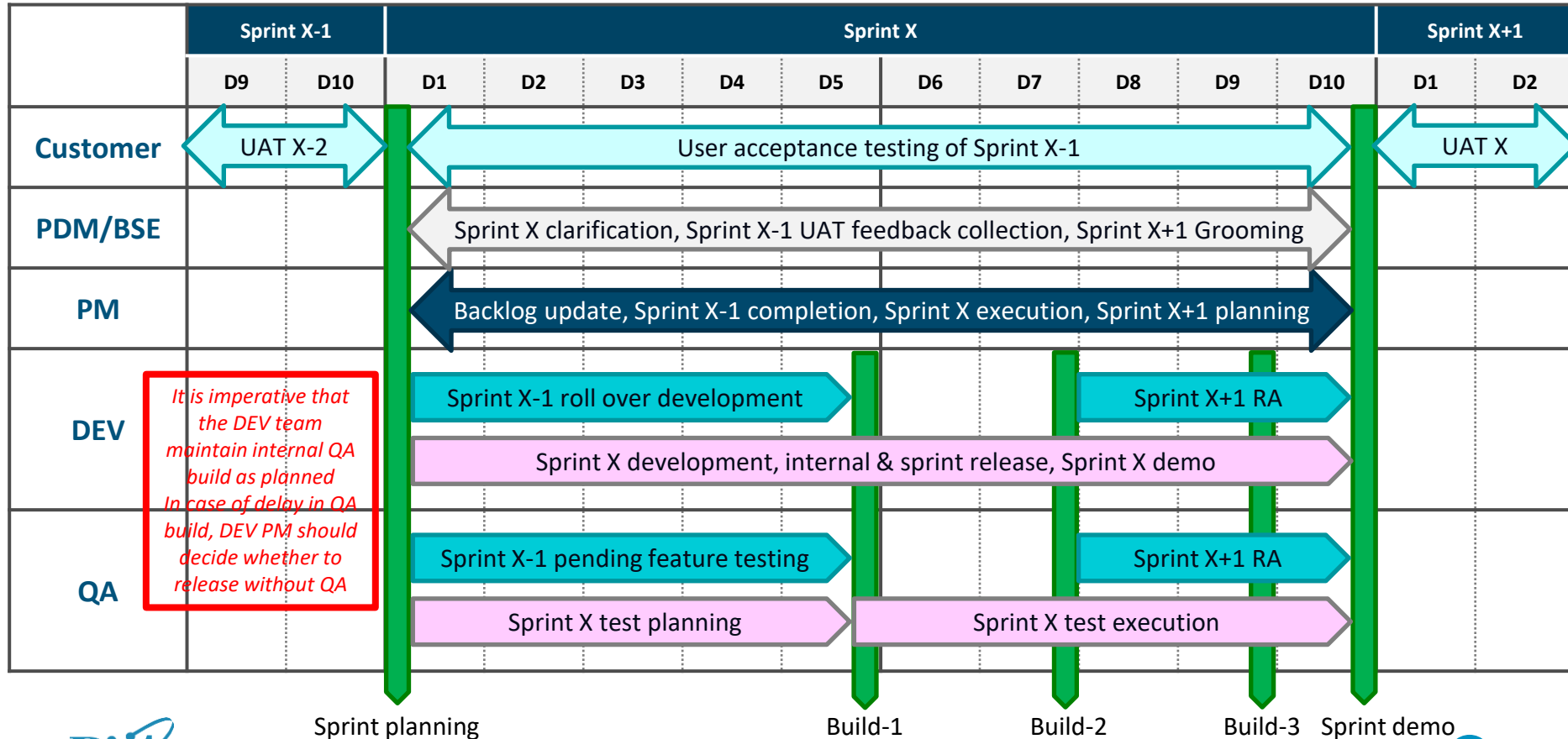


Anything that can impact on project cost, schedule, quality, productivity, future opportunity e.g. change of scope, resource skillset/turn over, testing device, dependencies, technical blocker etc. will be considered as risk factors. Check annexure for a risk/issue register sample

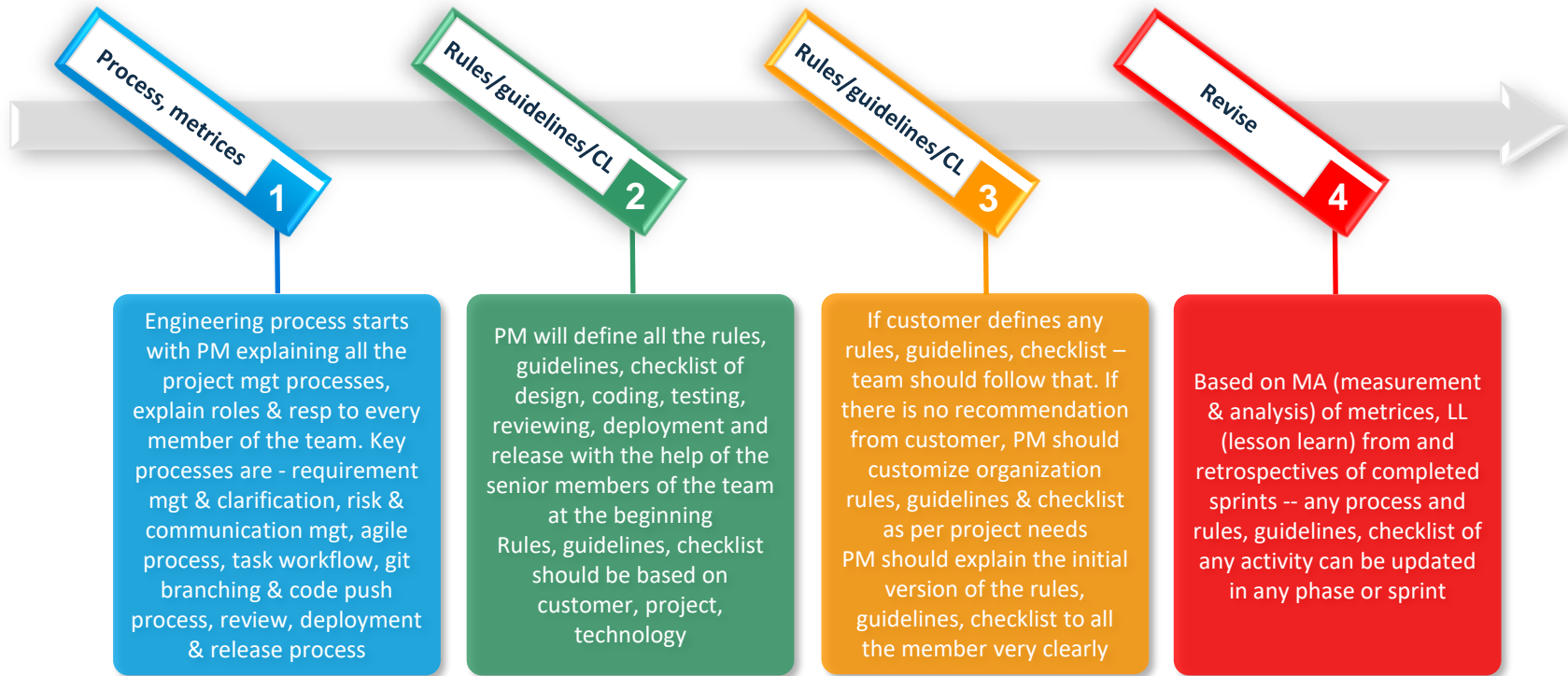
## 2. Project management [SCRUM] (5/6)



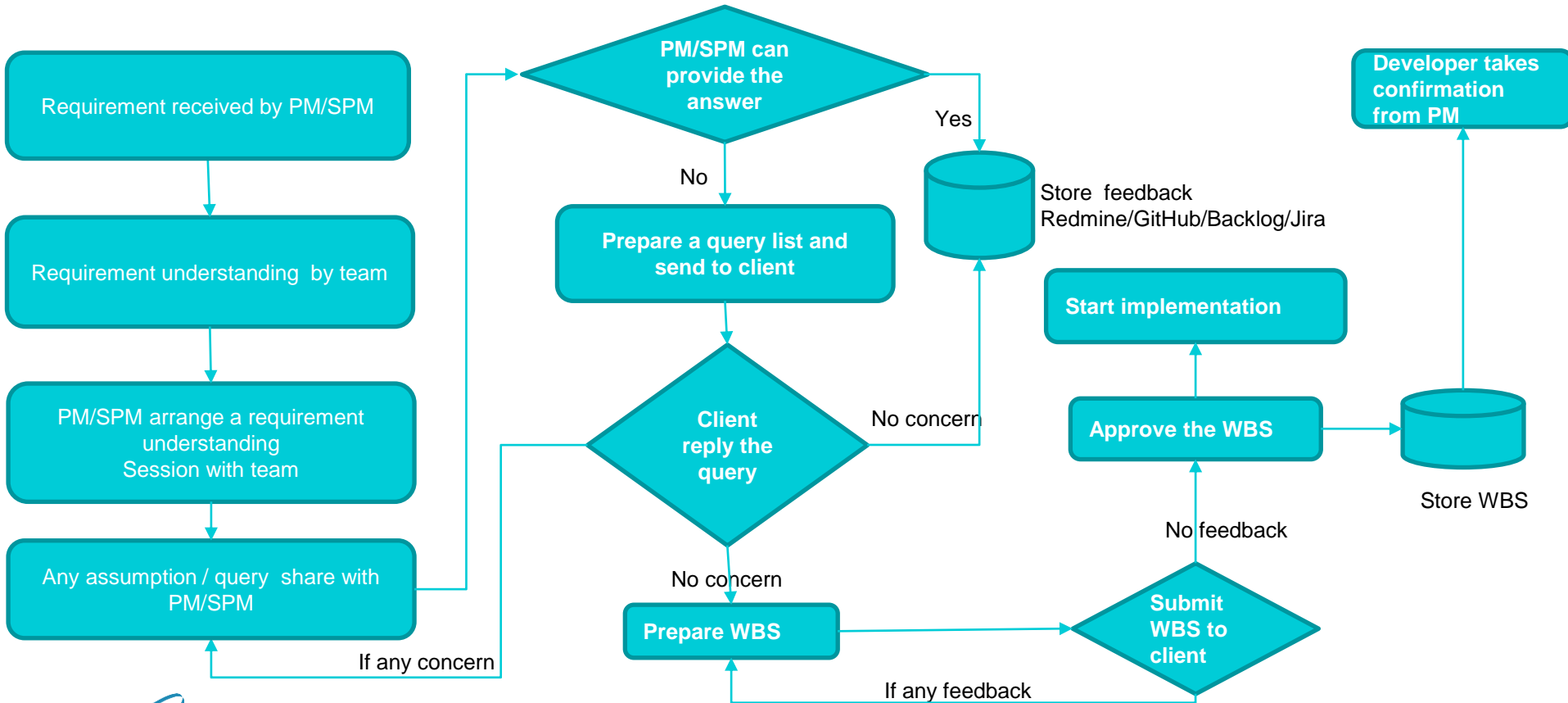
## 2. Project management [SCRUM] (6/6)



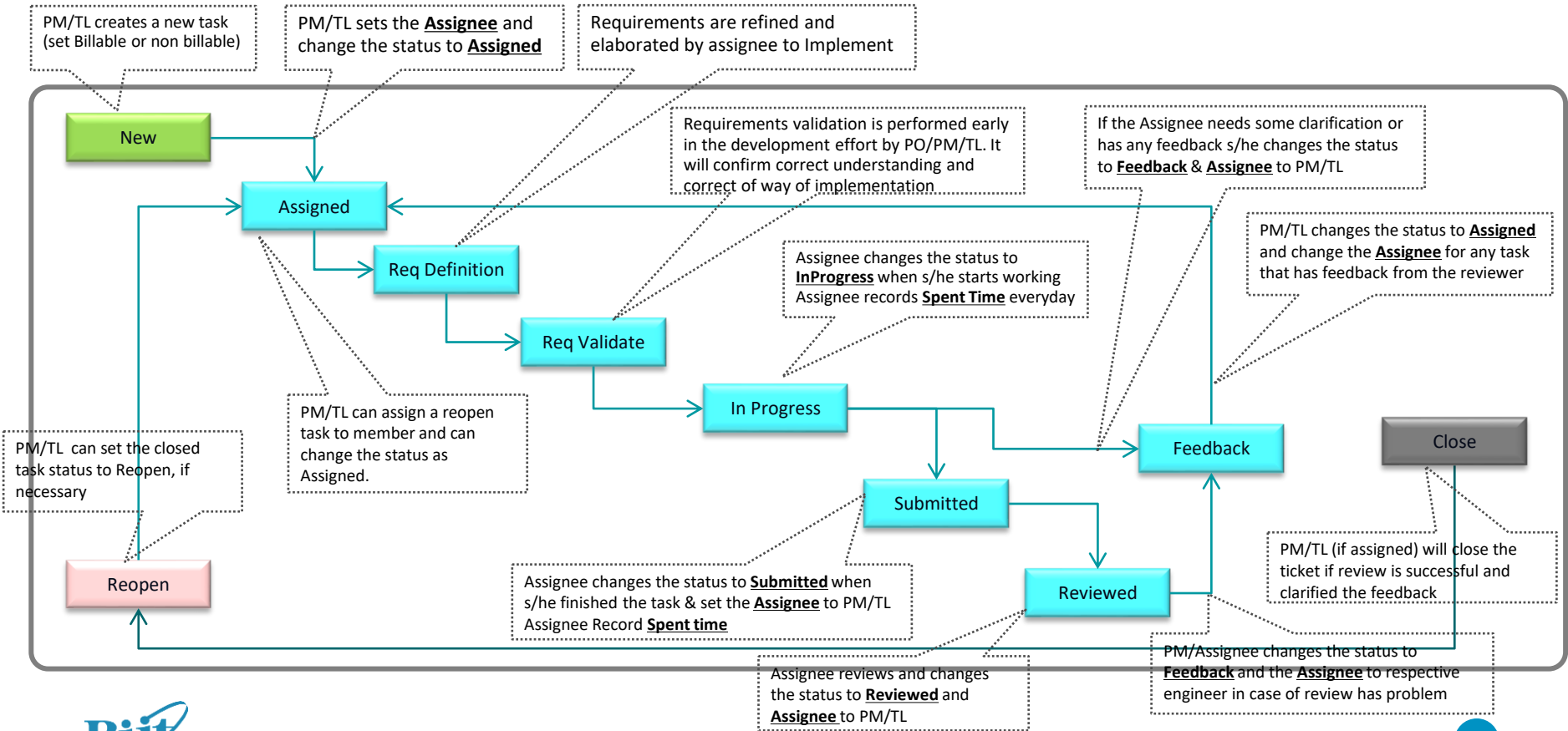
### 3. Engineering process [Pre-requisite] (1/8)



### 3. Engineering process [Requirement Understanding Workflow] (2/8)

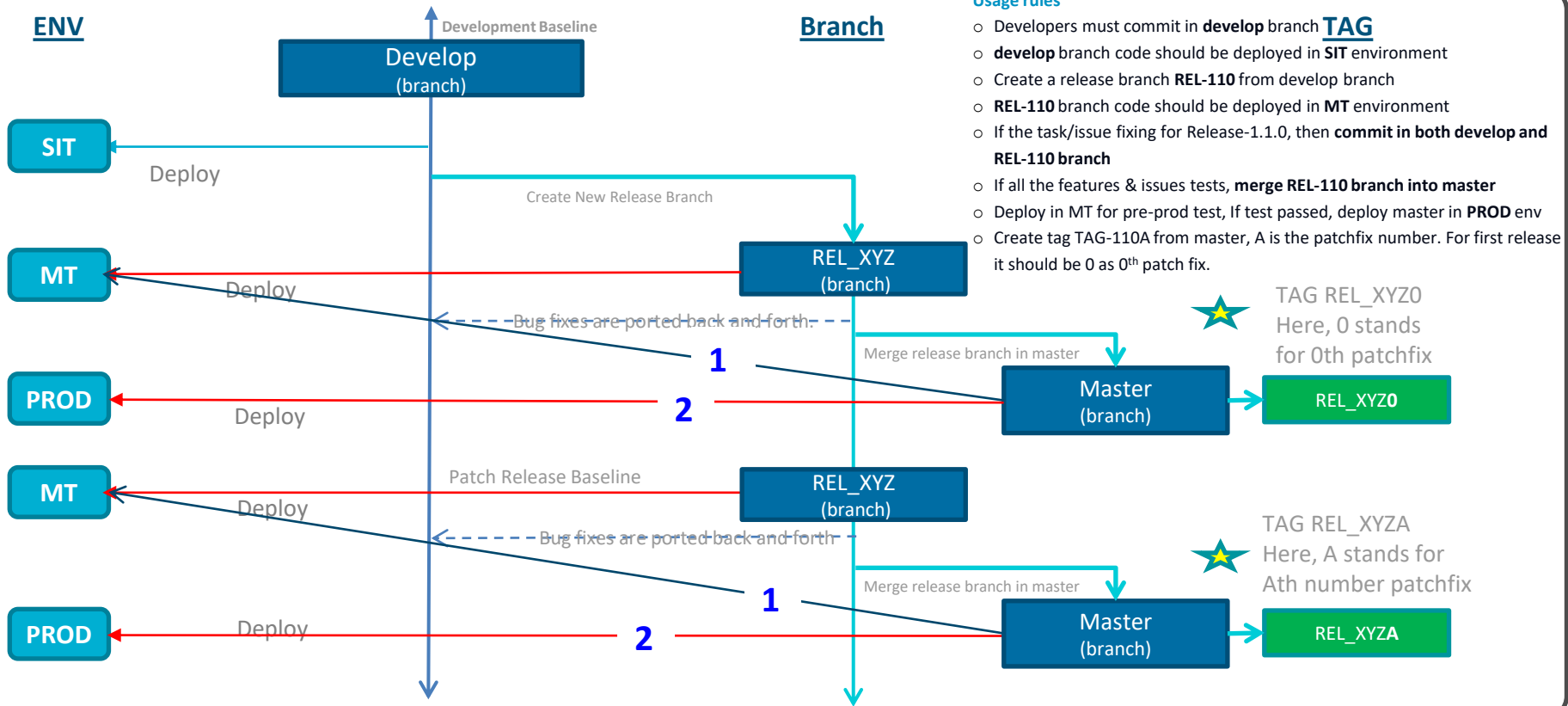


### 3. Engineering process [Redmine Ticket Workflow] (3/8)

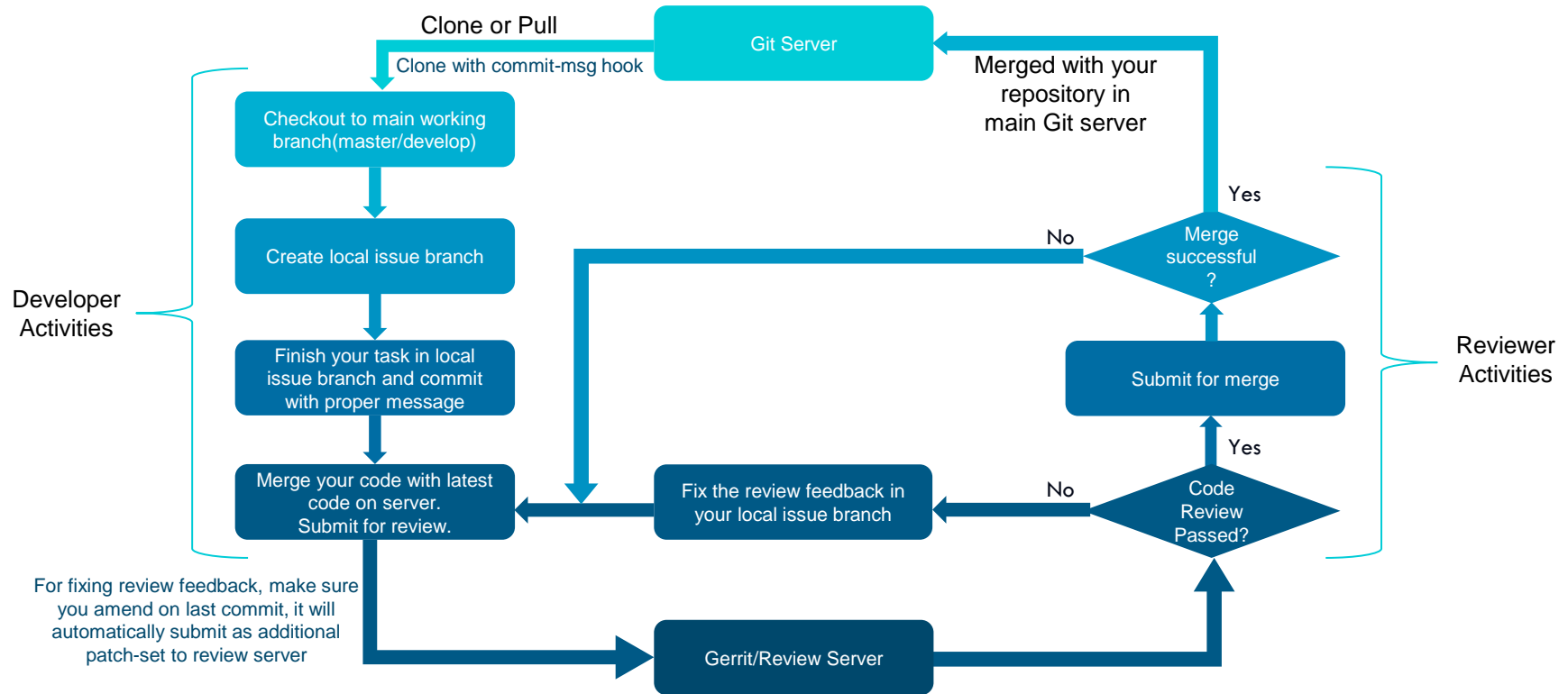




### 3. Engineering process [Git Branching] (4/8)



### 3. Engineering process [Git Workflow] (5/8)



Note: Git-Gerrit tutorial content is available at [URL](#)

### 3. Engineering process [Review] (6/8)

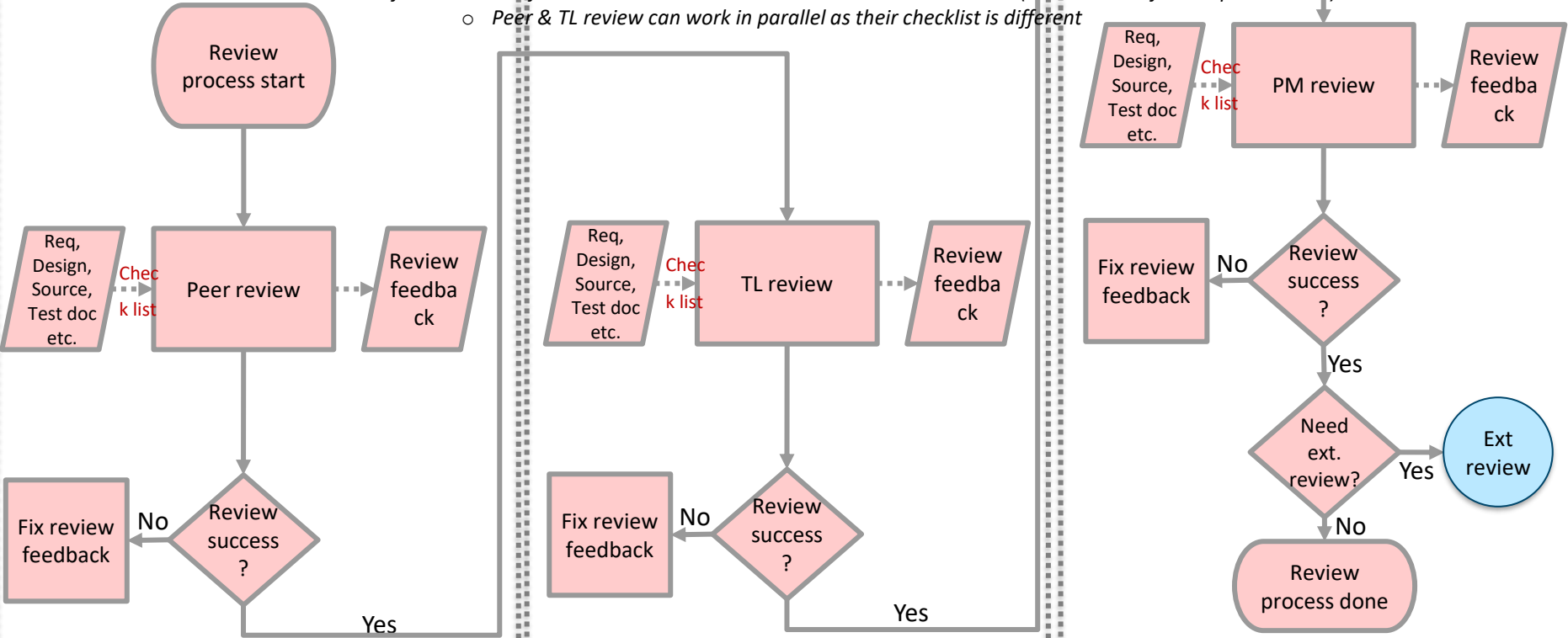
#### 1<sup>st</sup> Level Review (Peer)

- There is checklist for each level of review. The checklist should be attached as evidence (see annexure for sample checklist)

- Peer & TL review can work in parallel as their checklist is different

#### 2<sup>nd</sup> Level Review (TL)

#### 3<sup>rd</sup> Level & External Review (PM/BSE & ext)

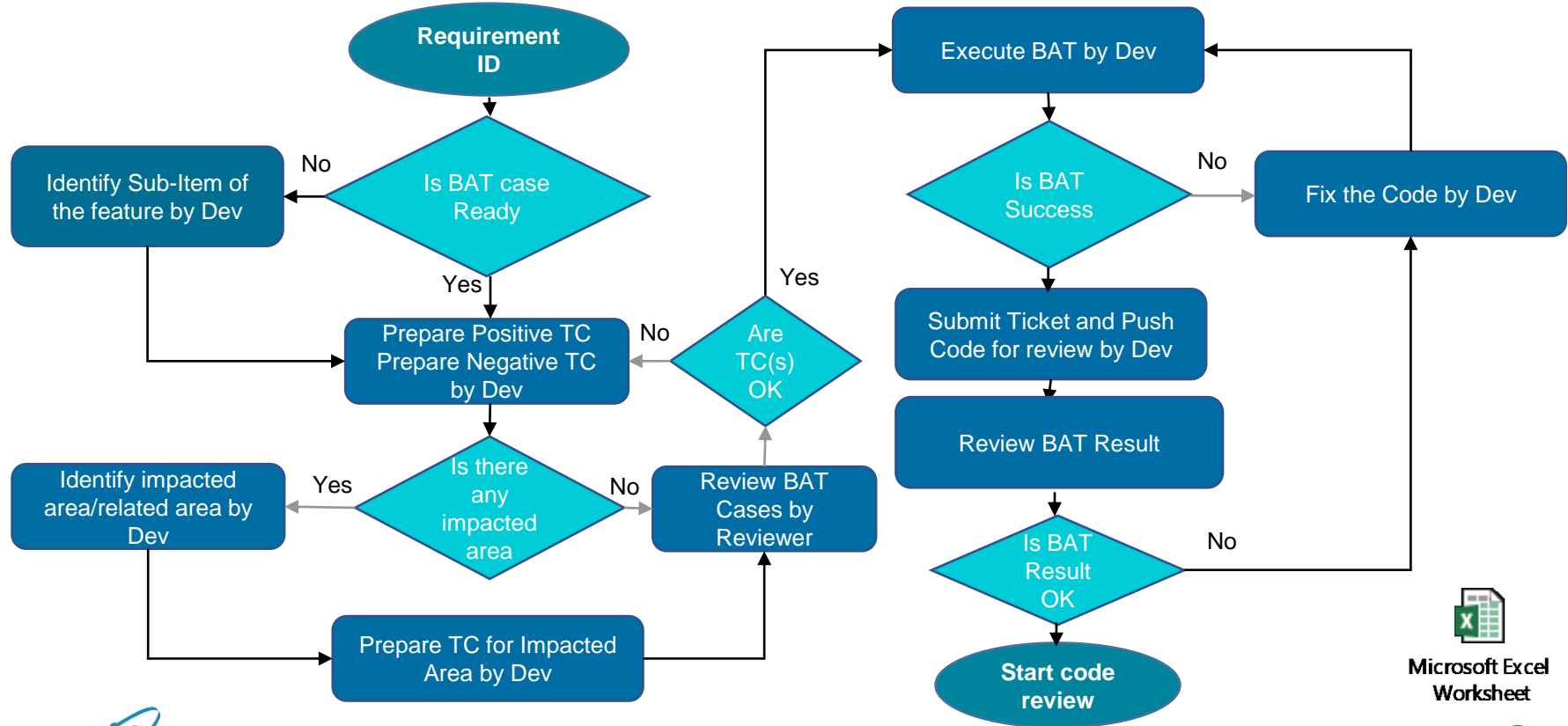


Focus is on compliance of guidelines & convention

Focus is on logics, integration & performance

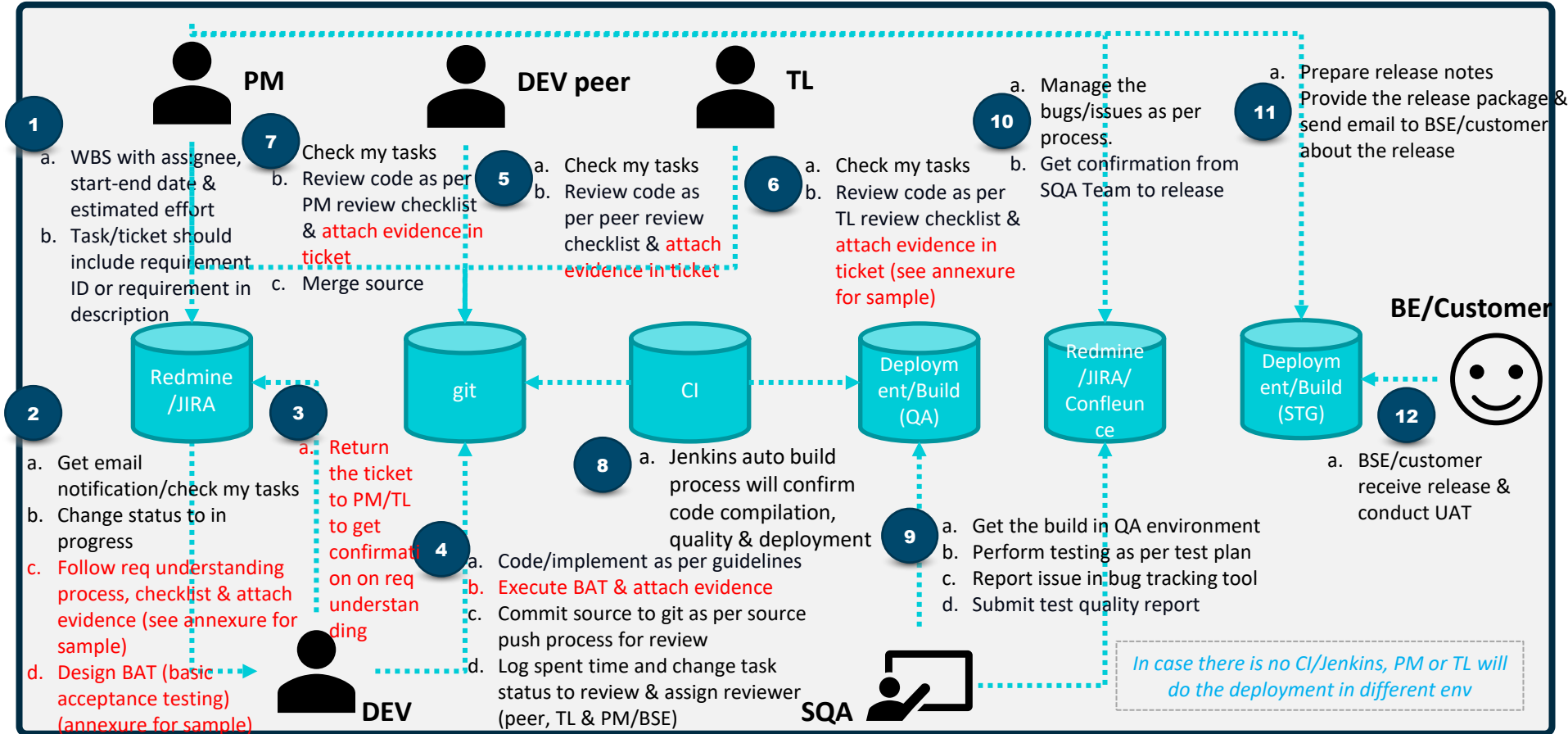
Focus is on overall quality

### 3. Engineering process [Developer BAT] (7/8)



Microsoft Excel  
Worksheet

### 3. Engineering process [Development] (8/8)



## 4. Support process [Release Management] (1/6)

- There should have release checklist for all kinds of releases. The checklist should contain -
  - All the internal releases are made on time and testing are done as per plan
  - Release scope has confirmation or sign-off
  - Release meets coding & testing quality metrics and sign-off from respective person
  - Release meets all acceptance criteria from client
  - All the release artifacts are tested and uploaded in appropriated places
  - Release notes and other required manuals or guidelines are sign-off from respective person
  - Source code, test matrix, test case, test result etc. are maintained release wise?
- Release judgement meeting is held, the checklist is ensured one by one and release decision was taken before the release
- Appropriate action plan has been taken and follow-up plan is prepared from each release judgement meeting for future reference

## 4. Support process [MA] (2/6)

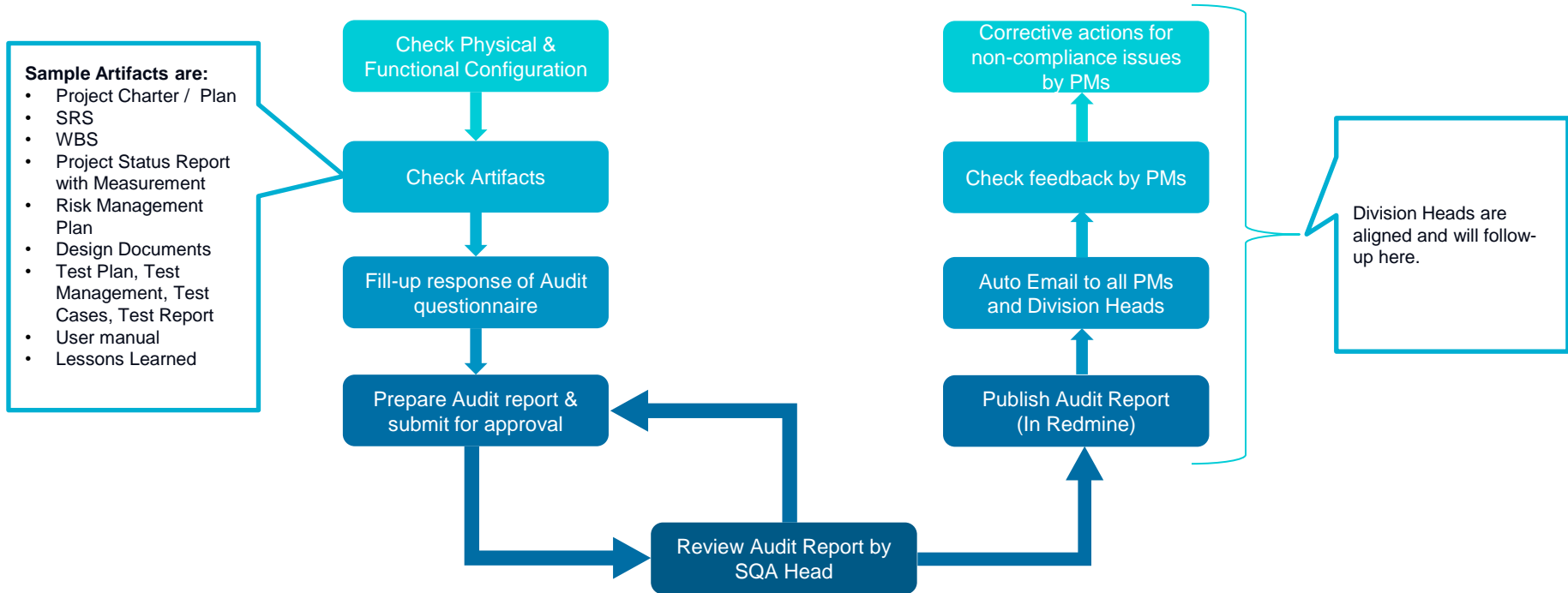
RAG Status		Overall	RED/AMBER/GREEN
Schedule	R	<b>Commitment Slippage = No of stories de-scoped (2) / No of Stories planned(8) * 100 = 25%</b> <G = less than or equal 10%> <A = less than or equal 15% & greater than 10%> <R = greater than 15%>	
Effort	G	<b>Effort Deviation = ((Actual Effort - Planned Effort) / Planned Effort) * 100</b> <G = less than or equal +/-10%> <A = less than or equal +/-15% & greater than +/-10%> <R = greater than +/-15%>	
Quality	A	<b>Code review effectiveness = <math>\frac{\text{\#of bug found in BJIT Dev review}}{(\text{\#of bug found in BJIT Dev review} + \text{\#of bug reported by BJIT QA})} \times 100</math></b> <G = greater than or equal 70%> <A = greater than or equal 50% & less than 70%> <R = less than 50%>  <b>Test efficiency = <math>\frac{\text{\#of bug reported by BJIT QA}}{(\text{\#of bug reported by BJIT QA} + \text{\#of bug reported from UAT, PROD})} \times 100</math></b> <G = equal to 100%> <A = greater than or equal 90% & less than 100%> <R = less than 90%> <b>&lt;Bug density = #of defects / Size (Story points or effort) * 100&gt;</b> <G = less than or equal 10%> <A = less than or equal 15% & greater than 10%> <R = greater than 15%> <b>&lt;Post Delivery Defect Rate = #of sprint defects / Size (Story points or effort) * 100 &gt;</b> <G = less than 5%> <A = less than 8% & greater than 4%> <R = greater than 7%>	
Customer compliance	R	<G = If customer is happy/give positivity feedback/appreciate deliverable etc. with deliverables> <A = If customer is concerned/not very happy/negative feedback etc. deliverable etc. with deliverables> <R = If customer is very unhappy/give very negative feedback/reject deliverable etc. with deliverables>	
Risk/issue	G	<G = If there is no risk> <A = If all the risks are minor (impact is not significant) & no possibility of becoming an issue> <R = If the possibility of the risks to become issue is HIGH & the impact is HUGE>	

## 4. Support process [PPQA] (3/6)

- ❑ PPQA stands for **P**rocess & **P**roject **Q**uality **A**ssurance. This is one of the 22 process areas of CMMI.
- ❑ To ensure the defined process implementation and ensure quality in all projects, we have a team, PPQA team.
- ❑ This audit process is conducted bi-weekly basis. There is some specific questionnaire for the above areas.
- ❑ Status of each target area is represented by 3 colours: Red, Yellow or Green. If all areas are green then the project status will be GREEN.
- ❑ Target area of this audit are:
  - Project Initiation
  - Requirements Management
  - Design
  - Development(CUT)
  - Change Management
  - Testing
  - Project Management
  - Risk Management
  - Inspection and Review
  - Configuration Management
  - Decision Analysis and Resolution
  - Organizational Process, Tools and Templates



## 4. Support process [PPQA] (4/6)



Please Check **Annexure [PPQA Details and a Sample Report]**

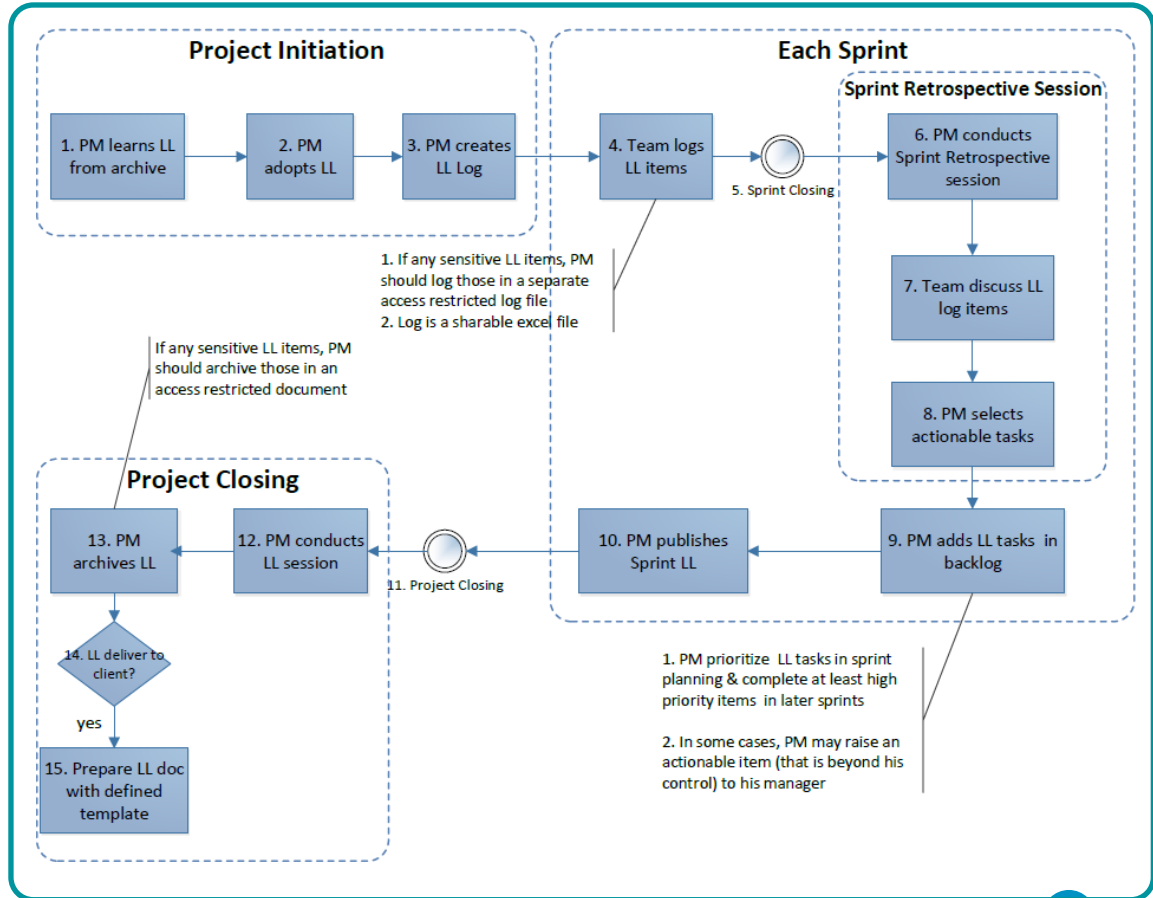
## 4. Support process [LL] (5/6)

The lessons learned session is typically a meeting at closing phase that includes:

1. Owner → Project Manager
2. When → Project closing phase
3. Participants →
  - Project team
  - Selected stakeholders including external of project, auditors, and/or QA
  - Project support staff
4. Output → Consolidate Lesson learn and Recommendation

The purpose of this document is to provide a quick reference checklist for use by the project manager to ensure that all appropriate activities to lesson learned have been addressed

1. Have the lessons learned been documented throughout the entire project life cycle?
2. Has a lessons learned been conducted towards the end of the project?
3. Have all lessons learned been documented?
4. Have documented lessons learned been distributed to other managers or PMO?
5. Have documented lessons learned been saved as part of the projects historical archive?



## 4. Support process [Project Closing] (6/6)

### EFFORT

#### MODULES & FUNCTIONS

- Estimated & actual effort of modules e.g. WEB client, API, CMS, Android & iOS
- Estimated & actual effort functions of modules

### LOC DATA

#### LANGUAGES

- Modules wise LOC of all the sources e.g. JS, SCSS, Java etc.
- Use Statistics tools to count them
- Input SLC, CLC, BLC separately

### SURVEY

#### CHECKLIST, SURVEY

- Ensure the predefined checklist upon closing. You can customize the checklist
- There are some surveys to be filled
- Record if there is any future opportunity

### EAF

#### ESTIMATION

- EAF stands for Estimation Adjustment Factor
- After finishing the project, we know a lot about the project, our resources & performance
- This data helps in similar future project estimation

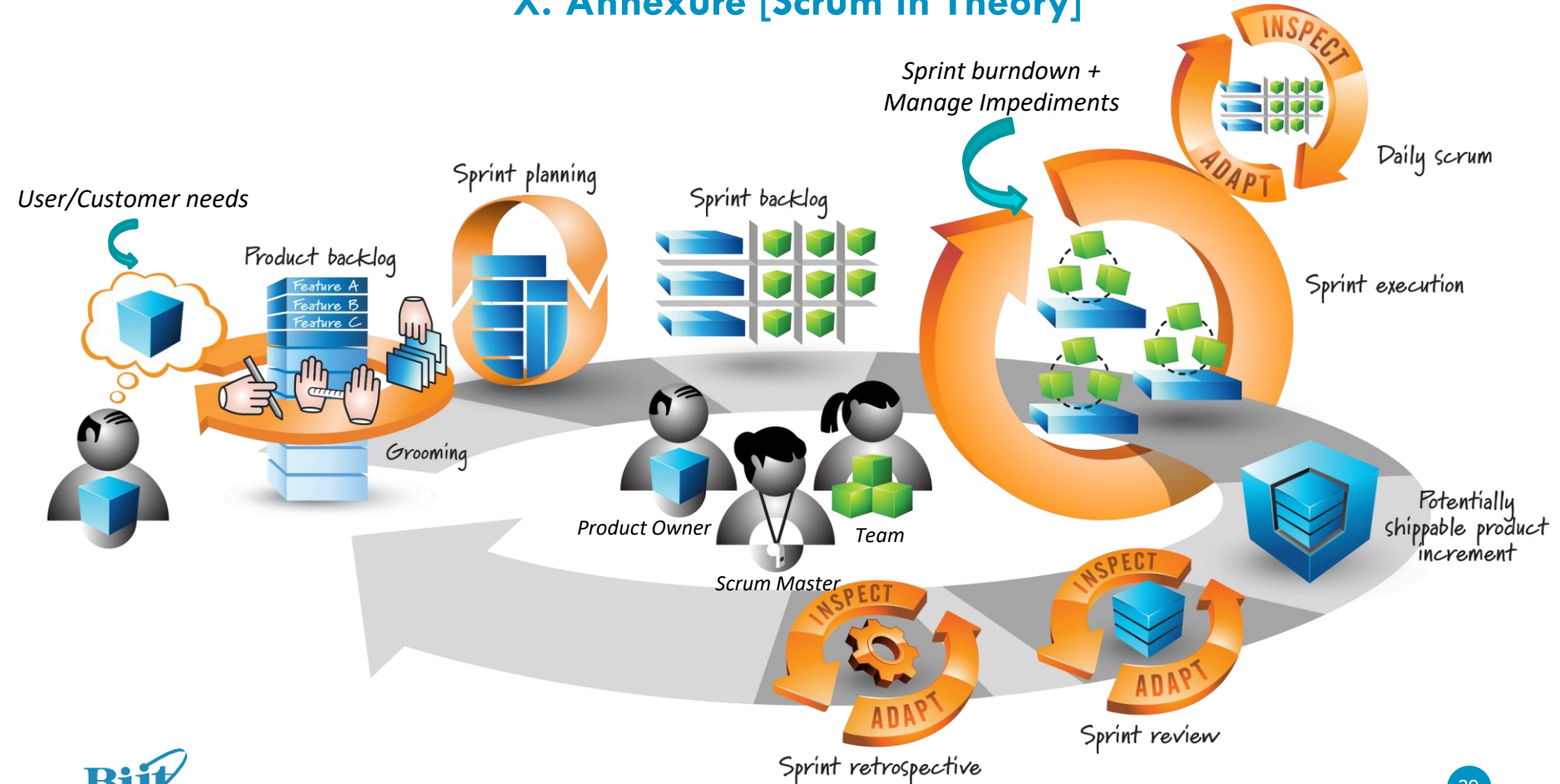
**Note:** We maintain all these data in ERP in Project module. Please check annexure for details  
PM & Dept/Div head is the PIC to manage project closing information

A white L-shaped line consisting of a horizontal segment and a vertical segment, forming the top-left corner of a frame.

# Annexure

A white L-shaped line consisting of a horizontal segment and a vertical segment, forming the bottom-right corner of a frame.

## X. Annexure [Scrum in Theory]



## X. Annexure [Sample Req. Understanding Evidence]

Feature Name: Easy creation with D&D

Requirement understanding Proof and Process:

Responsible Person to maintain this <xyz>

SN	Items to be confirmed	Value	Comment
1	Product to integrate this requirement	IkinariPDF V8	
2	WBS & Estimation	#1197 (comment), Deadline <2020/05/18>	
3	Developer BAT TC	#1472 (comment), Deadline <2020/6/20>	
4	Developer Code review checklist	#1465 (review)	
5	Developer impact analysis report	NA	
6	Release Date	6/26/2020	
7	Base source code to start development for this requirement	V8_BJIT_development	
8	Branch name to start working	V8_BJIT_CreationAppChange_Feature_Branch	
9	Final branch name to merge	V8_BJIT_development	
10	Test EXE based on product	NA	
11	Final release with test EXE or product		
12	Any assumption based on this has been confirmed	#1197 (comment), Done	
13	Make a requirement understanding session PM and DEV	Deadline<2020/05/11>, Done	
14	Share the understanding in the Ticket<Dev/Member> and get confirmation from PM	#1197 (comment), Done	
15	Whenever any change from client informs the team<Dev.QA> by PM/SPM	<a href="https://github.com/agendaPD/ikip/issues/1196#issuecomment-652251973">https://github.com/agendaPD/ikip/issues/1196#issuecomment-652251973</a>	
16	Update all relevant items as soon as get any change from client <...> inform SPM about the update	<a href="https://github.com/agendaPD/ikip/issues/1196#issuecomment-652251973">https://github.com/agendaPD/ikip/issues/1196#issuecomment-652251973</a>	
17	QA Team is aware about latest modification/implementation of this requirement	Yes	
18	All dependencies has been confirmed with client<Image, Icon, Text, Query, Content...etc.>	No dependency	
19	SRS link	#1197 (comment)	

## X. Annexure [Sample DEV BAT Sample]

GitHub ID/Feature		1196_Easy_Drag_And_Drop	BAT_Test Cases for <Requirement.ID> Date:YYY-MM-DD, Frequency- 0, Execution Time:0 hour				
TC ID		Sub Item	Test Type	Test Items	Result (Dev)	Result (Reviewer)	Comments
1	Feature Test	Shortcut File	Positive	Shortcut file name should be "いきなりPDFかんたん作成モード"	NE	NE	Need installer to test it. Depend on Agenda san
2		Shortcut File	Positive	On starting (without drag and drop) Parameter should be ""{Install Folder}\\WizMakePdf.exe" /EASYMODE"	OK	OK	
3		Common Matter	Positive	RMS licensing should work similar to normal mode of WizMakePdf	OK	OK	Checked by code changing
4		Common Matter	Positive	App is terminated when it is an invalid license.	OK	OK	Checked by code changing
5		Settings condition	Positive	Setting dialouge of Easy mode will be shown if double click is applied on the shortcut	OK	OK	
6	Impact Area Test	Normal Mode	Positive	Normal mode can be started from PDFEdit app and WizMainMenu App	OK	OK	
7		Normal Mode	Positive	WizMake normal shortcut file opens normal mode in double click	OK	OK	
8		Easy Mode	Negative	Password protected zip file dragged and password in canceled by user. App should be closed	OK	OK	
9		Easy Mode	Negative	Drag and drop corrupted pdf, exe, invalid file. Application should behave like as V7 of normal mode while processing	OK	OK	
10	White Box Test	Easy Mode	Negative	Cancel the creation process. Application should exit properly	OK	OK	
11		Normal Mode	Negative	Drag a folder named EASYMODE in its name to normal shortcut. EasyMode shouldn't start.	OK	OK	
12		Easy Mode	Negative	Change the shortcut files /EASYMODE to /EASYMODESSS , easy mode shouldn't start with.	OK	OK	
13		Easy Mode	Negative	Close the detail list for failure scenario . Created temporary files should be deleted.	OK	OK	

## X. Annexure [Sample Risk/Issue Register]

#	Description	Impact Description	Action/Answer Description (incl. Judge Description)	Priority	Type (Risk/Issue/Q&A)	Project	Category	Impact Level	Status	Probability	Handling Strategy/Response Type	Raised by	Raised Date	Assigned to (PIC)	Target Resolution Date	Closed Date	Original Source
1	As the project scope has not defined yet and we have prepared estimation based on some assumption. It will impact on project deadline.	At the time of the estimation & schedule, some scopes or requirements were unclear like the depth of SRS, the depth of sequence diagram, implementation of UT, additional UT, IT test case design to improve coverage. From Rakuten feedback/comment/clarification, if the assumption gets clarified, the estimation & schedule needs to be revised.	As of Feb/06, Implementation/test of Used API completed. This item is obsolete and should be closed.	Low	Risk	OSB2 Mule	Estimation	High	CLOSE D	High			2016/12/22	Milon		2017/02/07	Item API Risk Register as of 20170105
2	Rakuten has provided Japanese version of test specification which was translated by Yamamoto-san. If English version is not reviewed and done necessary correction, then there is chance that team can misunderstand the test specification.	Rakuten might need to review the translated document.	QA is out of BJIT scope	Middle	Risk	OSB2 Mule	Test(QA)	High	CLOSE D	High	Mitigate		2017/12/13	Zia Uddin		2017/02/07	Item API Risk Register as of 20170105
3	Mule is new technology to the team. Team is planning to learn mule in sprint-1. If team can not finish learning mule by sprint-1 then there is a high chance that sprint-2 delivery date will be delayed	Mule knowledge is necessary for this project as the team needs to design sequence diagram, refactor code & them. Lack of Mule knowledge might impact the delivery of those tasks	Team has got knowledge of mule technology using documentation and got training about sequence diagram preparation	Low	Risk	OSB2 Mule	Mule	High	CLOSE D	High			2016/12/16	Zia Uddin		2017/01/16	Item API Risk Register as of 20170105



## X. Annexure [PPQA Details]

Process Area	Item
Project Initiation	Project plan/business proposal/project charter/project scope is available and the contents are as per the template either in ppt, word, mail, confluence or any other tools?
	Project approval from client, overseas sales or SD is available?
	Depending on contract type whether project estimation is done?
	Inscope, out of scope for both development & testing are defined?
	Process, tools etc. are defined?
	Process tailoring is done & approved from client or DIV head?
	Project resources are defined?
	Roles and responsibilities are defined?
	Quality or productivity measurement metrics are defined?
	All the relevant assumptions, pre-condition, dependencies, risks are listed during business proposal preparation?
	Project is created in management tools (Redmine, Alfresco, gerrit or in clients defined tools)
	Project PO (purchase order) is received from sales or get formal commitment over mail about proceeding execution without PO?
	Project kick off meeting conducted & action points are tracked?
	Requirements are documented either as formal SRS or user stories, UX & screen spec or even as spec grooming sessions
Requirement development & management	Requirement grooming sessions or meetings minutes and record are stored properly?
	Div/dept head conduct KT to project team with whatever requirements they have analyzed, clarified for the estimation?
	Are functional (including UI as screen flow, UI display actions behavior etc.) requirements well defined and documented?
	Are non functional requirements well defined and documented?
	Is each requirement correctly and uniquely identified?
	CR management process are defined & following carefully?
Risk Management	Requirements are validated by customer or product owner or BSE
	Risk identified during project planning are transferred to project team?
	Risk register is properly maintained with appropriate categorization, priority?
	Risks are periodically reviewed and prioritized?
	Risks mitigation plan are defined & risks are regularly raised, escalated & mitigated?
	Risk and action point life cycle are maintained in Redmine or Jira?

## X. Annexure [PPQA Details]

Process Area	Item
Project Management	All the unconfirmed/finalized items during project initiation are confirmed in this phase? E.g. processes, process tailoring, development & testing scope, project tools, quality & productivity metrics etc.
	Product, development, project & quality management in redmine or confluence etc. are available
	Did the PM explained all the project mgt processes, explain roles & resp to every member of the team and kept the evidences? Key processes are - requirement mgt & clarification, risk & communication mgt, agile process, task workflow, git branching & code push process, review, deployment & release process
	Product backlog, project roadmap, WBS, sprint planning in redmine/JIRA based on selected development methodology are available
	Ensured required H/W & S/W for team & project
	Different meetings & reporting schedule, agenda, templates are defined?
	MoM of important meetings are properly maintained and action points are being defined & tracked?
	Daily, Weekly or other important reports are properly maintained?
	- Daily scrum meeting
	- Sprint planning meeting
Metrics & Analysis	- Sprint demo meeting
	- Sprint retrospective meeting
	- Weekly status meeting
Design, Development & Review	Training plan & OJT plan is prepared & tracked?
	Effort log is maintained in Redmine or Jira with both estimated and actual effort?
	PMP is reviewed and revision history is maintained?
	Both dev & QA metrics are measured & stored periodically as defined during project planning?
	Results of the metrics are analyzed and compared with target values and in case of any variation, appropriate action points are defined?
	Action point to correct metrics value are being tracked & executed?
	Metrics value are included in periodical report and shared with every stakeholders?
	Design guidelines are defined and updating regularly?
	Coding guidelines, standards, convention etc. are prepared and updating regularly as needed?
	Team is following redmine or client task management tools workflow properly?
	Team is following git or client SCM tools branching & commit, PR workflow properly?
	Requirement understanding workflow is followed and evidence is shared regularly?
	Developer BAT are prepared, reviewed and published?
	Developer BAT are executed and execution result provided?
	Redmine or JIRA or any other task management tool contains comments and spent hours are properly logged?
	Review guidelines, checklist is prepared for every level of reviewers and all kinds of deliverables e.g. SRS, design document, code, unit test etc.
	Review guidelines & checklist are updating regularly?
	All levels of review is done and evidence is kept?

## X. Annexure [PPQA Details]

Process Area	Item
Testing	Test scope (different types of testing), device, browser, OS compatibilities are defined?
	Test plan is prepared including test environment, resources, schedule?
	Test plan describe test strategy, test level and test types?
	Test matrix, test case design & test execution, issues management guidelines, standard, convention are prepared & being followed?
	Test matrix, test case design & test execution and issue management review guidelines, checklist are prepared & being followed?
	Test matrix is prepared if within the scope?
	Test cases are prepared?
	Test cases are executed and test results & reports are shared with every stakeholders and kept in project management tools?
	Test matrix, test case, test results are reviewed and evidences are kept as per process?
	Issues or defects life-cycle is maintained?
Lesson Learned	Bug analysis are done periodically & reported?
	LL log created
	Maintaining LL log during sprint
	Publish LL in Alfresco after end of sprint
	LL are shared in Sprint retrospective meetings?
	Sprint LL are published?
	LL action points or tasks are backlog?
	LL session conducted at project closing
	LL document archived at project closing in Alfresco/Confluence
	LL document accessible in Alfresco
Release Management	Sensitive LL items accessible only by authorized person
	Project Closing data are store in ERP according to the process?
	Internal sprint release plan are properly maintained?
	Release checklist is prepared, checked?
	Release judgement meeting is done and release meet release judgement checklist/acceptance criteria? Release judgement meeting results are stored in redmine/confluence?
	Release note is prepared and maintained properly?
	Source code, test matrix, test case, test result etc. are maintained release wise?

## X. Annexure [Sample Peer Review Checklist]

#	Category	Check item	Check	Comments if you didn't cover
1	Quality	PR contains all the information based on PR template (Quick Explanation, All Specs Link, All dependent PR List, external PR, sonarr Report Link, sampleCurl)	√	
2	Quality	No major/critical issue/minor issues in the sonar report in new code and status is green	√	
3	Quality	Followed the coding guidelines (please check all items in "https://confluence.rakuten-it.com/confluence/display/ECSG/15.11+Java+Coding+Standards" sheet)	√	
4	Quality	Code is properly formatted with defined code style and no unexpected code diff in PR	√	
5	Quality	Commit message is as per defined format (should include JIRA ticket?)	√	
6	Quality	Gitpub branching is properly maintained	√	
7	Quality	All variables/method properly defined with meaningful, consistent, and clear names?	√	
8	BUG	Check if property files have been updated as per environment profiles ie proxy, port, stg and prod url, cache properties etc	√	
9	Quality	Check the properties names are meaningfully defined in the property's configuration file	√	
10	BUG	all variables have proper type consistency or defensive casting	√	
11	BUG	Invalid parameter values are handled properly early in methods (Fast Fail)	√	
12	BUG	Check if input validation are properly implemented in right places (API JSON, Validator, Interface-Preprocessor)	√	
13	Quality	Check if classes, functions, constants are placed in appropriate module i.e. architecture followed correctly	√	
14	BUG	check whether unreachable or dead code not exists	√	
15	Quality	Code handles the exception properly and as per exception handling design of the project	√	
16	BUG	Code handles specially NullPointerException/ any expectation that a lib function may throw	√	
17	BUG	Check the proper exception message ie not contains any sensitive information in the errorMessage like sid, authkey etc	√	
18	Quality	Check if error messaging convention & properties are updated properly	√	
19	Quality	Check if proper logging (Ensure all exception, not just throwing) has been applied and as per architecture	√	
20	Quality	Unit test cases exists for all conditions as per "UT guidelines"	√	
21	Quality	Application of proper mock data are as per current implementation	√	
22	Quality	Check if test cases are verifying requirements and only one requirement	√	
23	BUG	Check if test cases have appropriate assertions	√	
24	Quality	Check if duplicated test cases can be replaced with Parameterized UT	√	
25	Quality	Check if all the newly implemented & modified UT cases are passing	√	
26	Quality	All of the given review feedback incorporated and confirmed & placed	√	

## X. Annexure [Sample TL Review Checklist]

#	Category	Check item	Check	Comments
1	BUG	Check if input validation are properly implemented in right places (API JSON, Validator, Interface-Preprocessor)	√	
2	BUG/Quality	Cross check the business logic with restful application if migration api		
3	BUG	Check whether the code implemented the requirement properly (First review the code with the spec and then conduct some manual I/F call using POSTMAN/JMETER or whatever REST client you use)	√	
4	BUG	Check if there are any performance issues/BUG with the code	√	
5	BUG	check if any backend compatibility issue (Modification of the existing schema/json/method used by another feature/api)	√	
6	BUG	check whether unreachable or dead code not exists	√	
7	BUG	Code handles the exception properly and as per exception handling design of the project	√	
8	BUG	Code handles specially NullPointerException/ any exception that a lib function may throw	√	
9	BUG	Check the proper exception message ie not contains any sensitive information in the errorMessage like sid, authkey etc	√	
10	Quality	Check if proper logging (Ensure all exception, not just throwing) has been applied and as per architecture	√	
11	BUG	Check if property files have been updated as per environment profiles ie proxy, port, stg and prod url, cache properties etc	√	
12	Quality	Check if caching has been considered for applicable scenarios	√	
13	Quality	Unit test implementation is as per guidelines	√	
14	BUG	check the mock data's are created dynamically and no side effect on time-zone and current datetime	√	
15	Quality	Check if test cases are verifying requirements and only one requirement	√	
16	BUG	Check if test cases have appropriate assertions	√	
17	Quality	Check if duplicated test cases can be replaced with Parameterized UT	√	
18	BUG	Check if all the newly implemented & modified UT cases covers all the business logic for all branching	√	
19	BUG	Check of all the queries in query resolution page is clarified and resolved and reflected in code	√	
20	Quality	All of the given review feedback incorporated and confirmed & placed	√	

## X. Annexure [Sample PM Review Checklist]

#	Check item	Check	Comments
1	PR contains all the information based on PR template (Quick Explanation, All Specs Link, All dependent PR List, external PR, sonar Report Link, sampleCurl,reviewerName, coverage, link)	√	
2	No major/critical issue in the sonar report and status is green		
3	Check how many review bug founds in review		
4	All of the given review feedback incorporated and confirmed & placed		
5	Check of all the queries in query resolution page/discussion channel is clarified and resolved and reflected in code		
6	Check BAT, Peer and TL Review evidence and coverage whether it is satisfactory		
7	if Peer and TL Review not satisfactory, do the review as required to ensure the quality		
8	if the review bug is zero, do the review for couple of check of type BUG		
9	Check whether the code implemented the requirement properly (First review the code with the spec and then conduct some manual I/F call using POSTMAN/JMETER or whatever REST client you use)		
10	Check the review matrix confluence is updated by Reviewer		
11	Check all the clarification evidences in jira, review evidence in appropriate place are maintain properly		

## X. Annexure [PPQA Sample Audit Report]

WK-2037												
	Project Initiation	Requirement Definition	Requirement Management	Project Management	Risk Management	Metrics and Analysis	Design and Development	Testing	Lesson Learned	Release Management	Project Compliant	Comments
Project1	100%	78%	100%	88%	100%	100%	93%	100%	50%	100%	91%	LL is not maintained periodically.
Project2	100%	78%	100%	88%	100%	100%	93%	100%	50%	100%	91%	LL is not maintained periodically.
Project3	100%	89%	100%	88%	100%	100%	93%	100%	100%	100%	96%	
Project4	100%	67%	75%	100%	50%	100%	60%	70%	67%	100%	77%	Risk & LL is not maintained periodically.

## X. Annexure [Sequence Diagram Design Sample Guidelines]

Title Lv1	Lv2	Lv3	has comment	can trace input to output	Remarks
Input Output					
	Input				
		From mongo db's data	Yes	Yes	should have mongo db's schema.
		From request headers	Yes	Yes	
		From request body	Yes	Yes	
		From request path	Yes	Yes	Need discussion for the host part
		From properties file	Yes	Yes	
		From xml file.	Yes	Yes	
	Output				
		inner path	Yes	Yes	Need discussion for the host part
		query	Yes	Yes	
		request body	Yes	Yes	
	Exception Output				
		Error Response	Yes	Yes	
		Alert Mail	Yes	Yes	Need to add more comments
		Log data	Yes	Yes	
Data Flow discontinuity					
	catch exception		Yes		should have variable name. Need discussion regarding variable name
Validation and business logic					
	Validation				
		XSD	Yes	Yes	
		value check	Yes	Yes	
	Resolve shopid or shopurl				
		From ESA header	Yes	Yes	ShopId is resolved from ESA with isAuth flow as "externalProviderId"
		From shopmngt API			Not applicable for ItemInsert. This innerAPI is used in ItemInsert to check whether this shop has medicine/drug store or not.
		Insert shopid or shopurl	Yes	Yes	



# X. Annexure [Java BE Sample Coding Guidelines]

Sl	Module	Type0	Type1	Define Task
1	All	Class	Class	Write javadoc comments for CLASS explaining what the class is about.
2	All	Function	Function	Write javadoc comments for every new FUNCTION explaining what the function intends to do along with parameter and parameter description.
3	All	Properties	Variable, Constant	Try to write small explanatory remarks about the VARIABLES & CONSTANTS if restricted due to variable naming length rule.
4	All	Class	Class, Package	Class & Package names should not be Plural.
5	All	Class	Class	Class name should not represent more than one responsibility.
6	All	Function	Function, Unused	Do not leave unused functions in the code
7	All	Properties	Variable, Unused	Do not leave unused variables in the code.
8	All	Properties	Variable	Avoid unnecessarily creating local variables.
9	All	Convention	General, Unnecessary	Do not copy unnecessary code from existing application source, reuse only what is necessary.
10	All	Convention	String, Variable, Constant	There should not be any hard-coded string on java files. Those values must come from constants.
11	All	Function	Function	Make a separate class for common method.
12	All	Convention	Unused	No unused Imports should be there.
13	All	Convention	Todo, Comments	Add //TODO comments for the code which you are planning to modify in near future.
14	All	Comments	Comments	Add //FIXME comments if you think code needs optimization in near future.
15	All	Function	Function	Do not write long functions; maintain modular approach with functions having least number of lines.
16	All	Convention	Warning	Please follow eclipse warning and remove them.
17	All	Exception	Function, finally, return, Exception	A finally block should not contain a return statement and should not throw any exceptions that are not caught within the finally block.
18	All	Exception	Exception	Use try-with-resource statement wherever possible with JDK7.
19	All	Exception	Exception	There should be a valid reason to drop or ignore Exceptions. Either exceptions should be caught and handled, or they should be thrown to the outermost level.
20	All	Exception	Log, Exception	Don't log and throw exception in same function.
21	All	Convention	Comment, Git	Always add justified comment for SVN/Git commit.
22	All	Function	Function	Naming convention of methods should be related to functionality or feature.
23	All	Function	Function, Comment	Justify any logic, if it is complex and add the justification in the method comment.
24	All	Convention	Comment, Unused	Do not leave commented out codes.
25	All	Function	Function, Variable	One blank line should always be used between methods, local variables and between logical sections inside a method to improve readability
26	All	Properties	Variable	The names of variables declared class constants should be all uppercase with words separated by underscores (" _ ")
27	All	Function	Function, Variable	Extra spacing between functions/variable declaration and definitions should not be there
28	All	Exception	Exception	Throw all the checked exceptions till the highest level.

## X. Annexure [Angular FE Sample Coding Guidelines]

SI	Type1	Define Task
1	Data Item	Are the names of the variables meaningful?
2	Data Item	If the program language allows mixed case names, are the variables names with confusing use of lower-case letters and capital letters?
3	Data Item	Are the variables initialized?
4	Data Item	Are there similar sounding names?(For unintended errors)
5	Data Item	Are all the common structures, constants and flags to be used defined in a header file rather than in each file separately?
6	Data usage	Are values of right data types being assigned to the variables?
7	Data usage	Are bounds to array subscription and pointers properly checked?
8	Data usage	Has the usage of similar looking operators checked?
9	Control Flow	Are all the conditional paths reachable?
10	Control Flow	Are all the individual conditions in a complex conditions separately evaluated?
11	Control Flow	If there is a nested IF statement, are the THEN and ELSE parts appropriately delimited?
12	Control Flow	In the case of a multi-way branch like SWITCH/CASE statement, is a default clause provided? Are the breaks after each CASE appropriates?
13	Control Flow	Is there any part of code that is unreachable?
14	Control Flow	Are there any loops that will never execute?
15	Control Flow	Are there any loops where the final condition will never be met and hence cause the program to go into an infinite loop?
16	Control Flow	What is the level of nesting of the conditional statements? Can the code be simplified to reduce complexity?
17	Style	Are unhealthy programming constructs being used in the program?
18	Style	Is sufficient attention being paid to readability issues like indentation of code?
19	Miscellaneous	Have you checked for memory leaks?
20	Document	Is the code adequately documented, especially where the logic is complex or the section of code is critical for product functioning?
21	Document	Are the interfaces and the parameters thereof properly documented?
22	General, Data Item	Are all vars used somewhere?
23	General, Control Flow	No unnecessarily duplicated logic
24	General	Code intent is clear upon initial reading
25	General	Any code that isn't clear, but is needed, has an effective comment
26	General	Are method calls or attribute lookups every called on entities that could be undefined/null?
27	General	The functions can appropriately handle unexpected inputs.
28	General, Comment	Commented code has been removed (comments themselves are fine).
29	Test	There are tests for the proposed functionality (if not, there's a good reason)
30	Angular	Need to use ng-bind instead of {{Lable or variable}}


## X. Annexure [Sample UT Guidelines]


OK	✓	Not OK	X	NA	—	
Category 1	Category 2	check item	Status	Test Item Checklist		
T e s t  I t e m	F u n c t i o n a l i t y	Normal case	✓	Cover all normal business patterns		
			✓	Test case written for checking that the return value is correct		
			✓	Contents of exceptional object at exception occurrence is correct		
			—	Check the point of business rule considering operational condition and maximum transaction number		
		Validation check	—	Input data format is checked, like whether the format of edy number is as per the spec or not		
			—	In case of number, the digits are checked as per the spec		
			✓	In case of string empty input is checked		
			—	Date time input format and the data is checked		
			—	Boundary value greater than the upper limit is checked		
			—	Boundary value less than the lower limit is checked		
			—	Array out of bound case is handled properly		
			—	Negative case for input parameter values are checked • 0, null, null list, null array		
			✓	Co-relation validity is checked: When there are some values which is dependent on a specific value. For example: When A_flag=false, A_value must be empty		
		Confirm I/F input	—	Common part of API header and response is checked: HTTP-Header has been set correctly or not		
			—	Common part of API header and response is checked: I/F type has been set correctly or not		
			—	Common part of API header and response is checked: UID has been set correctly or not		
		DB edit value	—	Test case written for those case when update DB value (If applicable)		
		Confirm conversion value	—	When convert from string to int, int to string, date to other format etc. are checked		
		Abnormal case	—	In case of mock object method call, exception or error code returns		
			✓	Cover all abnormal business patterns		
		Coverage	✓	Statement coverage: 100% C0		
			✓	Loop coverage for all possible iteration (1, 2, n)		
			✓	Condition coverage: 100% C2		
		Special case	—	Performance check done		
			—	Security check done		


## X. Annexure [Project Closure: Effort of Modules]


# Test project

Confidential Project ☐

 Allocation

 0 Documents

 0 Tasks

 Active

[Settings](#) [Project Timeline](#) [Project Technologies](#) [Project Members](#) [Project Metrics](#) [Project Lesson Learn](#)

[Project Modules](#)

[Project Closure](#)

[Project EAF](#)

Module	Estimated Effort	Actual Effort
WEB Client	0.00	0.00
API	0.00	0.00
CMS	0.00	0.00
Android	0.00	0.00
iOS	0.00	0.00

## X. Annexure [Project Closure: LOC data of Modules]

Open: Project Modules ×

**Module**

WEB Client

**Estimated Effort**

0.00

**Actual Effort**

0.00

Technologies

Functions

Technology/Language	Source Code Line	Source Code Line %	Comment Line	Comment Line %	Blank Line	Blank Line %	Total Line of Code
JS	10,000.00	95.24	300.00	2.86	200.00	1.90	10,500.00
SCSS	2,000.00	90.91	100.00	4.55	100.00	4.55	2,200.00

Add an item

## X. Annexure [Project Closure: Effort of Functions of Module]

Open: Project Modules

Module

Estimated Effort

Actual Effort

WEB Client

2.50

2.60

Technologies

Functions

Function	Estimated Effort	Actual Effort
User registration	0.50	0.60
Role management	1.00	1.10
Customer management	1.00	0.90

Add an item

## X. Annexure [Project Closure: Checklist, Survey & Future benefits info]

Test project

Allocation

Documents 0

Tasks 0

Active

Name of the tasks :

Confidential Project ☐

Settings

Project Timeline

Project Technologies

Project Members

Project Metrics

Project Lesson Learn

Project Modules

Project Closure

Project EAF

Checklist


Survey


Future Benefits Realization


Checklist	Yes/No	Comments
All deliverables completed to the required quality	<input checked="" type="checkbox"/>	
All invoices & cross charges paid (if applicable)	<input checked="" type="checkbox"/>	
Project documentation completed	<input checked="" type="checkbox"/>	
Risks and Issues closed or resolved	<input checked="" type="checkbox"/>	
Handover to organisation complete	<input type="checkbox"/>	
<a href="#">Add an item</a>		


## X. Annexure [Project Closure: Estimation Adjustment Factors]

# Test project

 Allocation

 0 Documents

 0 Tasks

 Active

Name of the tasks :

Confidential Project ☐

[Settings](#) [Project Timeline](#) [Project Technologies](#) [Project Members](#) [Project Metrics](#) [Project Lesson Learn](#)

[Project Modules](#) [Project Closure](#) [Project EAF](#)

Attributes	Value
Product Attributes	1.00
Hardware Attributes	1.00
Personnel Attributes	1.00
Project Attributes	1.00



## X. Annexure [Project Closure: Estimation Adjustment Factors]

Open: Project EAF



### Attributes

Personnel Attributes

Attribute Criteria	Very Low	Low	Normal	high	Very High	Extra High	Rating
Analyst capability	1.46	1.19	1.00	0.86	0.71	0.00	High
Applications experience	1.29	1.13	1.00	0.91	0.82	0.00	High
Software engineer capability	1.42	1.17	1.00	0.86	0.70	0.00	Very High
Virtual machine experience	1.21	1.10	1.00	0.90	0.00	0.00	Normal
Programming language experience	1.14	1.07	1.00	0.95	0.00	0.00	Normal

「Thank you」