How to run a Software Company.?

By Ammar Ameerdeen

a.b.ameerdeen@gmail.com

Disclaimer

The information provided in this book, "How to Run a Software Company," is intended to offer general guidance and insights based on personal experiences and research. While every effort has been made to ensure the accuracy and relevance of the content, the dynamic nature of the software industry means that practices and strategies may evolve over time.

Readers are advised to exercise their own judgment and discretion when implementing any advice or strategies outlined in this book. The author and publisher cannot guarantee specific outcomes or results, as success in running a software company depends on various factors beyond the scope of this publication.

Moreover, the book does not serve as a replacement for professional advice, legal counsel, or specific industry consultation. Readers should seek appropriate professional assistance tailored to their individual circumstances before making business decisions or taking actions that could impact their software company.

The author and publisher disclaim any liability, loss, or risk incurred as a consequence, directly or indirectly, of the use and application of any information presented in this book.

By using this book as a resource, readers acknowledge and agree to the terms of this disclaimer.

Feel free to adjust or customize it to better fit your specific needs or legal requirements!

Contents

Development

This is the central hub or the repository for the overall engineering quality related information and the engineering team should always refer to this for any engineering quality related queries or updates. And hence the engineering team should always make sure that the information shared is correct and up to date and any updates or changes should be communicated and shared in a timely manner within the organization. The information shared here should be kept as confidential and must not share with any external parties or publish in any public forum.

Guidelines

All the defined guidelines and the strategies are listed here.

Versioning Strategy

All the software projects must adhere to these guidelines and must refer to this document for any queries regarding the software versioning.

The version management must be done using the Semantic Version Specification.

Normal version number must take the form of X.Y.Z .

X, Y & Z are non-negative integers and must not contain leading zeros.

X – Major version. Must increment when incompatible / breaking changes are introduced.

Y – Minor Version. Must increment when adding backward compatible features.

Z – Patch version. Must increment when making backward compatible bugfixes.

- Major version zero (0.Y.Z) is for initial development releases. Hence the initial development release must be started with 0.1.0 and the minor (Y) and the patch (Z) versions must be incremented accordingly for all the subsequent development releases prior to the first production release.

- Version 1.0.0 must be used as the initial public release.
- Patch version Z must be incremented with the backward compatible bug fixes.

- Minor version (Y) must be incremented with the backward compatible functionality or feature additions or implementations of change requests (CRs). Further when batch of bugfixes are incorporated and if the change is not trivial, then the minor (Y) version must be incremented instead of patch version. When incrementing major version (Y), the patch version (Z) must be reset to zero.

- Major version (X) must be incremented when backward incompatible or breaking changes are introduced with the new release. Here both the minor (Y) and the patch (Z) versions must be reset to zero.

- A pre-release version may be denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version. Identifiers must comprise only ASCII alphanumeric and hyphens [0-9A-Za-z-]. Identifiers must not be empty. Numeric identifiers must not include leading zeroes. Examples: 1.0.0-alpha.1, 1.0.0-0.3.7

- Build metadata may be denoted by appending a plus sign and a series of dot separated identifiers immediately following the patch or pre-release version. Identifiers must comprise only ASCII alphanumeric and hyphens [0-9A-Za-z-]. Identifiers must not be empty. Build metadata must be ignored when determining version precedence. Examples: 1.0.0-alpha+001, 1.0.0-beta+exp.sha.5114f85

Versioning Product Suites

When versioning product suites which includes one or more subcomponents or products, the main or the product suites must have a separate version number adhering to this version guideline and rest of the subcomponents can have their own version numbers and they can be evolved independently. But should not share or mentioned them in the product suite production releases. Only the suite's version information must be included instead.

Examples: Let's consider a real-world example from one of Company ongoing client project here. The MGT project, we have a product suite which includes four subcomponents.

MGT Product Suite

- MGT Connector API
- MGT Heartbeat Application
- MGT Floating Icon Application
- MGT Connector Host Controller Application

In this case initially we will release the initial production release of the product suite as 1.0.0. And the rest of the subcomponents can have their own production versions. But we will not share the subcomponent versions to the public, instead should use that information internally when tracking source code branches, bugs or features and internal releases.

Mobile Application Display Version

The mobile application releases must have a separate display version number which is not compatible with the Semantic versioning schema. But it must be used only for the display purpose only and always the releases must have the version numbers compatible with Semantic versioning schema. In mobile application releases, this is common practice or guideline used, when the CodePush approach is used to update the application without introducing a new release. This approach must be used when CodePush is possible for any hot fixe for a production release. In this case, the release version remains the same, but since the users get the new update via CodePush that must be reflected in the display version.

Examples: Release version 1.0.0, Display version 1.0.0.00, once the hot fix is published via CodePush the to the users' devices, release version remains as 1.0.0, but the display version is changed to version like 1.0.0.001 / 1.0.0.002. Here users are able to see that they are having small updates on top of a given production release by looking at the display version.

Versioning Feature Flags

When using feature flags, to define the version of a given feature, it is not a must to consider Sematic version schema as it is used internally. Hence version format like explained in the display version section shall be implemented.

Examples: Feature A might have multiple versions like 1.0.0.005, 1.0.0.006, here version 1.0.0.005 can be targeted for user segment S1 while version 1.0.0.006 is targeting the user segment S2.

Source Code Branching Strategy

There are two types of branch categories.

- *Main or code flow branches* - These are the branches that follow the code flow of the code changes from the development to production. They are permanently available on the repository and have infinite lifetime.
- *Supporting or temporary branches* - These branches are created with a specific goal and have a limited lifetime. Hence, they are destroyed when they are no longer needed.

Main or Code Flow Branches

As mentioned above following four main or code flow branches should be permanently available throughout the lifetime of the repository and further, each branch directory mapped to a specific deployment environment as they follow the code flow from initial development to the production release. Direct commits to these branches are not allowed and all the work should be done by creating separate supporting branch. They should be incorporated or merged to the given code flow branch via a pull request (PR) or merge request (MR). Hence appropriate branch protection rules should be applied to each code flow branches. For test codes, only the develop and the main code flow branches should be created, and all the branch specific guideline and rules should be considered unless explicitly specified any exceptions here.

- *develop* - All the development work should be branched off and brough into this branch. All the new features developments, bug fixes and any other work not included to the next planned release should be branched off from this and create a separate supporting branch accordingly.

- *qa* - This branch contains all the codes ready for quality assurance (qa) testing. For any qa reported issues, new supporting branch should be created by branching off from the *qa* branch and merged once the fix is completed. Finally, these fixes should be merged to *develop* branch.

- *staging* - This branch contains all the tested features and bugfixes ready for demo, user acceptance testing (UAT) or the approval for stakeholders prior to the next planned production release. For any issues reported at this stage, new supporting branch should be created by branching off the *staging* branch and merged once the fix is completed. Finally, these fixes should be merged to *develop* branch.

- *main / master* - This branch is for the production and hence this should only contain the latest production code. For any issues reported from the production, new supporting branch (hotfix branch) should be created from specific version or the tag of the production branch and merged once it is ready to release. Finally, these fixes should be merged to *develop* branch. Further, merging to *qa* or *staging* branch should be done if there is an active release is in progress. This branch should be protected by locking (read-only) and should only be allowed to update when ready for a production release.

Supporting or Temporary Branches

These branches should be created by branching off from one of the code flow branches (develop, qa, staging & main) mentioned above. Only the specific set of supporting or temporary branches are allowed in certain code flow branches.

Naming Guideline

All the supporting branches except build, release and merge supporting branches should be created with their standard prefixes and the format given below.

[prefix]/[ticket system prefix]-[ticket id]_[work item description]

The hyphen character should be used to separate words in the work item description. The work item description should be short and self-explanatory. Note that the ticket information and the work item description must be separated by the underscore character. And should use lower case letters for words.

For build and merging supporting branches, following format should be used.

[prefix]/[work item description]

For release supporting branches, following format should be used.

[prefix]/[version number]-[prerelease suffix][pre release number with leading zero]

The version number should be minor, major and the patch component defined in the versioning strategy guideline. Allowed prerelease suffixes are alpha (for alpha testing), beta (for beta testing) and rc (for release candidate testing) while the prerelease version number should be versioned sequentially with leading zero.

Examples: feature/jira-123_login-module, bugfix/jira-001_home-page-layout-fix, merge/combine-user-login-backend-frontend, build/code-coverage-test-run, release/1.2.0, release/1.3.0-alpha01, release/2.3.0-beta02, release/5.0.0-rc01

Feature Branches

Any new feature, module or development should be done on a new feature branch and as the development is based on *develop* code flow branch, new feature branch should be branched off from the it. The prefix 'feature' should be used when creating a feature branch and they can only be created by branching off the *develop* code flow branch. They are not allowed to create with other code flow branches (*qa, staging & main / master*). Hence branch naming rules should be enforced to make sure that correct naming standards are followed by the teams.

Examples: feature/jira-500_user_management, feature/jira-400_user-profile-module

Bugfix Branches

Any issues or bugs reported should be fixed in separate supporting branch by branching off from a given code flow branch. The prefix 'bugfix' should be used when creating a bugfix supporting branch and can only be branch off from one of the code flow branches except main or master (*develop*, *qa*, *staging)* or any release branch targeting for a release. Further, any bugfixes done during a test or UAT stage should be merged into the *develop* branch.

Examples: bugfix/jira-100_bulk-update-performance-fix, bugfix/jira-002_login-error-typo-fix

Hotfix Branches

When the production issues reported, hot fix branches are created by branching off the specific version or the tag. The prefix ‘hotfix’ should be used when creating this type of supporting branch and can only be branch offed from the *main / master* (the production) code flow branch. The hot fixes should be merged into the *develop* branch and then into qa or staging branches if any release is in progress.

Examples: hotfix/jira-888_broken-user-profile-fix, hotfix/jira-700_invalid-configuration-fix

Experimental Branches

Any research & development (R&D) work or proof of concept (POC) which are not part of a release should be created as an experimental supporting branch. The prefix ‘experimental’ should be used when creating this type of supporting branch and can only be branch offed from the *develop* code flow branch.

Examples: experimental/jira-005_web-socket-poc, experimental/jira-009_reaseach-on-open-source-img-processing-libs

Build Branches

These are the supporting branches for creating specific build artifacts or for doing code coverage runs. The prefix ‘build’ should be used when creating this type supporting branches.

Examples: build/demo-configuration, build/code-coverage-test-run

Release Branches

These types of branches are created for specific release of component or system. The prefix 'release' should be used when creating this type of supporting branches and can only be branch offed from the *develop* code flow branch.

Examples: release/1.2.0, release/1.0.0, release/1.3.0-alpha01, release/2.3.0-beta02, release/5.0.0-rc01

Merging Branches

The merging supporting branches are used for resolving merge conflicts. The prefix ‘merge’ should be used when creating this type of supporting branches and can only be branch offed from the *develop* code flow branch.

Examples: merge/combine-user-login-backend-frontend, merge/combine-user-management-sub-modules

Code Review & Merge Process

Any work that is targeting for a release should always be reviewed and approved by the one or more members who are either component or code owners before releasing. For any test or test automation related work also should govern under these same policies and hence they should be reviewed and approved always even though they are not deployed to the production systems. It is mandatory to implement branch protection rules in each source code repository to enforce these policies and guidelines.

New feature or component development should provide a feature design document details along with the code review request so that the reviewers can validate the implementation with the design. For performance and security features or issue fixes, authors should provide performance test execution reports and the security test execution reports information to the reviewers when submitting the code review requests. If any of those mentioned above is not available with code review requests, the reviewers should request the required information from the authors and should not approve the changes until the enough information is available before approving. Further, it is a must to review and approve any work regardless of the size and the complexity of the work. The lead of the development team is responsible for code reviews and he or she can add any other senior members within the team or outside the team to the code review team.

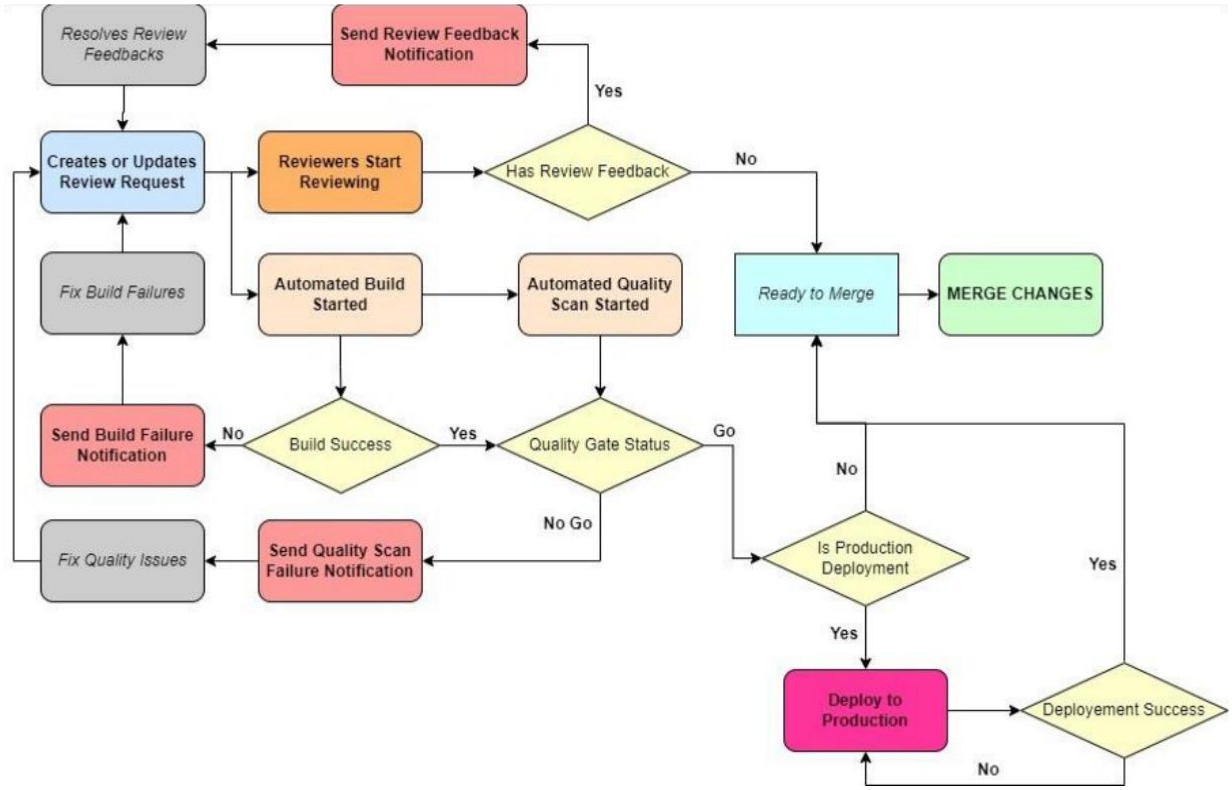
All the teams should collect or keep code review statistics as they are using to evaluate the quality aspect of the deliverables of each individual. Hence it is a must to have at least once a month code review meeting for entire team to discuss the review findings of members of the team, possible improvements and plan trainings if required. Hence team leads or technical leads should spend some quality time on these tasks and should consider as high priority lead task.

Apart from above mentioned manual checks or verification it is a must following automated checks, green build, quality gate pass, test execution and the code coverage under the continuous integration pipeline execution while successful deployment is a must when merging release branch to the production branch (main / master). Based on the verification results, merging the pull request should be automatically allowed or blocked.

Note: For more information about release management and the code quality refer to the release management guideline and code quality guideline respectively.

Process Flow

Following diagram illustrates the code review process flow. The branch naming validation checks are not included to the diagram for simplicity.

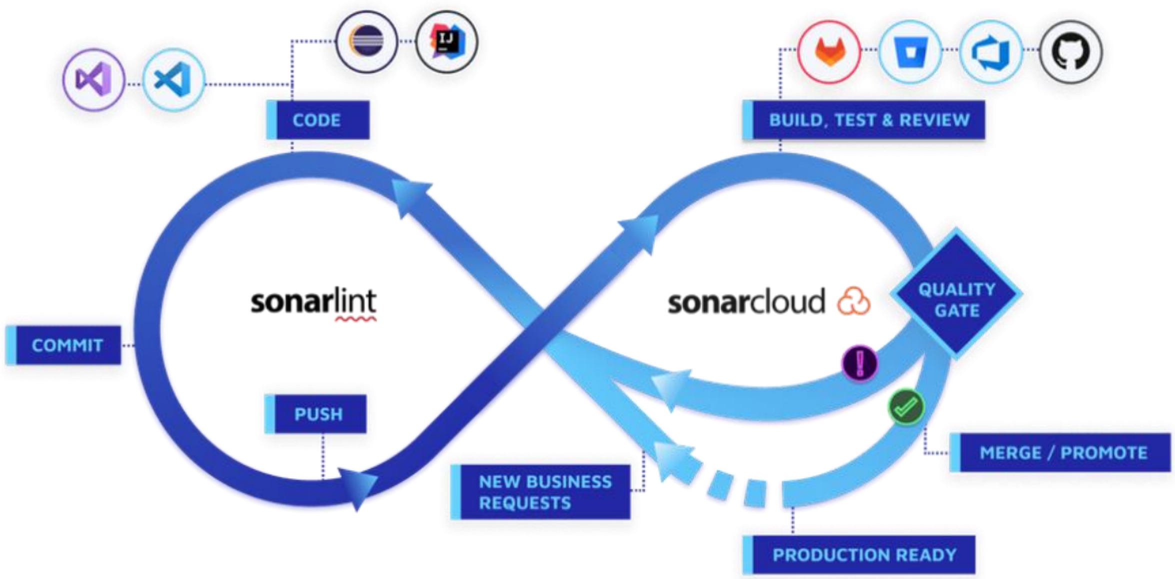

Resolves Review Feedbacks
Send Review Feedback Notification
Yes
Creates or Updates Review Request
Reviewers Start Reviewing
Has Review Feedback
No
Fix Build Failures
Automated Build Started
Automated Quality Scan Started
Ready to Merge
MERGE CHANGES
Send Build Failure Notification
No
Build Success
Yes
Quality Gate Status
Go
No
No Go
Fix Quality Issues
Send Quality Scan Failure Notification
Is Production Deployment
Yes
Yes
Deploy to Production
Deployement Success
No

Code Quality Guideline

Guidelines

The SonarLint static code analysis tool extension for the integrated development environments (IDEs) or editors should be installed and configure so that the first line of defense will resolve most of the issues or problems. Further the connected mode should be configured if supported for the IDE or editor where the rules and the analysis settings are automatically, and real time synchronized from SonarQube or SonarCloud to SonarLint which will make sure that the development teams have the exact rules and settings in their local development setup.

All the code repositories should be integrated with the SonarCloud solution, and the quality scan results should be used to implement the quality gates for go or no-go decision for code review requests (pull requests or merge requests). Hence continuous integration pipelines should integrate with the SonarCloud so that every change is scanned and going through the quality gate regardless of the size or the complexity of the change.



Sonar Ecosystem in Code Quality Process

For more details on SonarCloud refer to the official documentation.

The code coverage should be greater than or equal to 80% and for any projects not meeting this should request special request to change the threshold for a certain time period with the plan on how to increase the code coverage of the projects. No one in the team except the lead or technical lead should allow to or given permission to disable or off the quality scan or the quality gate checks. Hence all the code review requests (pull requests or merge requests) and the builds should be validated make sure that the quality of the releases or deliverables are up to the expected level and consistent. Hence these checks should be implemented and enforced in the continuous integration pipelines.

Finally, the single source of truth for the code quality will be the SonarCloud dashboard projects summary view and hence the integration from SonarLint to SonarCloud is a must for all the ongoing and future projects.

Develpment teams should have at least once a month code review meeting for their teams and should discuss the code review findings (manual code review and static code analysis details) for each individual and have a plan on improving the code quality of the team members as quality of the deliverables is one of the key performance indicators (KPIs) of each individual as well as the team. The lead of the team is responsible for improving the code quality of the team and hence should spend some considerable lead time on above mentioned tasks.

Release Management Guideline

Guidelines

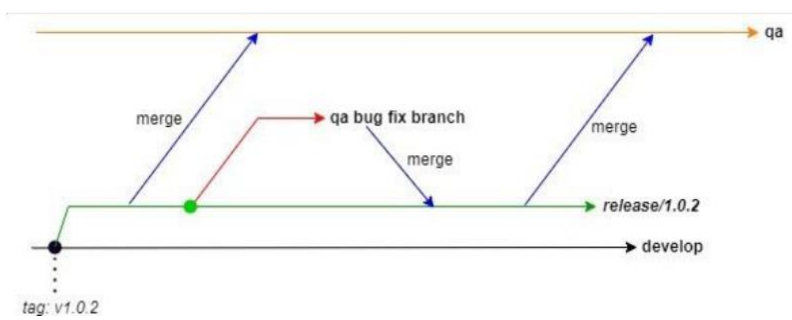
General

This guideline explains how a given release of component or subsystem or system should be managed and guides with the relevant instructions and standards that should be followed by the teams.

As mentioned in the branching strategy guideline a new release should be created from the *develop* code flow branch. when creating a particular release, this checkpoint must be tagged in the develop branch and the tag format should be [v]X.Y.Z-[prerelease suffix][prerelease number with leading zero] and the X, Y, & Z represents the minor, major and patch components mentioned in the versioning strategy guideline. For prerelease suffixes and number refer to the branching strategy guideline. Examples: v1.0.0, v1.2.0-alpha01, v5.0.0-rc02

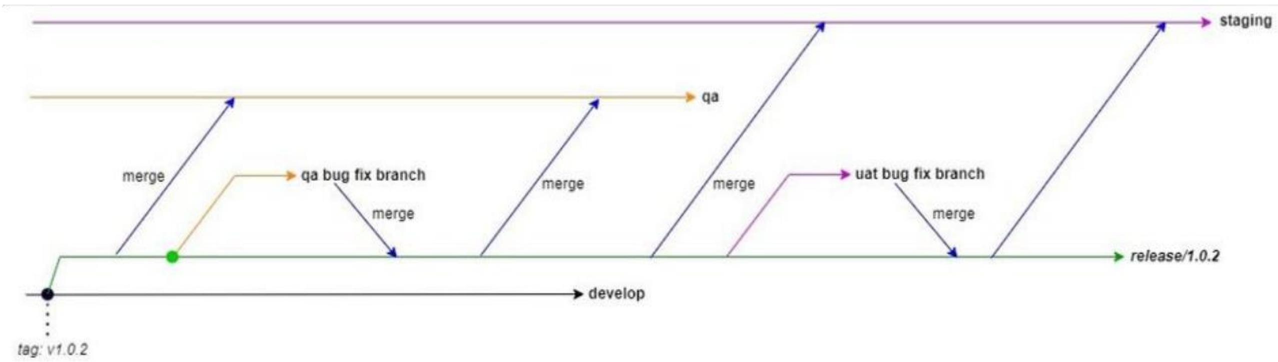
Release Branch Creation

Once a given release is created, then that release branch should be merged into the *qa* code flow branch once it is ready for quality assurance testing. The merge operation should still be adhered to the code review and merge process guidelines. Then if any qa bugs reported, then again, they should be fixed by creating separate bugfix branches from the release branch. Once the issues are fixed, they should be merged to the same release branch and redeploy the fixes for the further testing.



Fixing QA Bugs

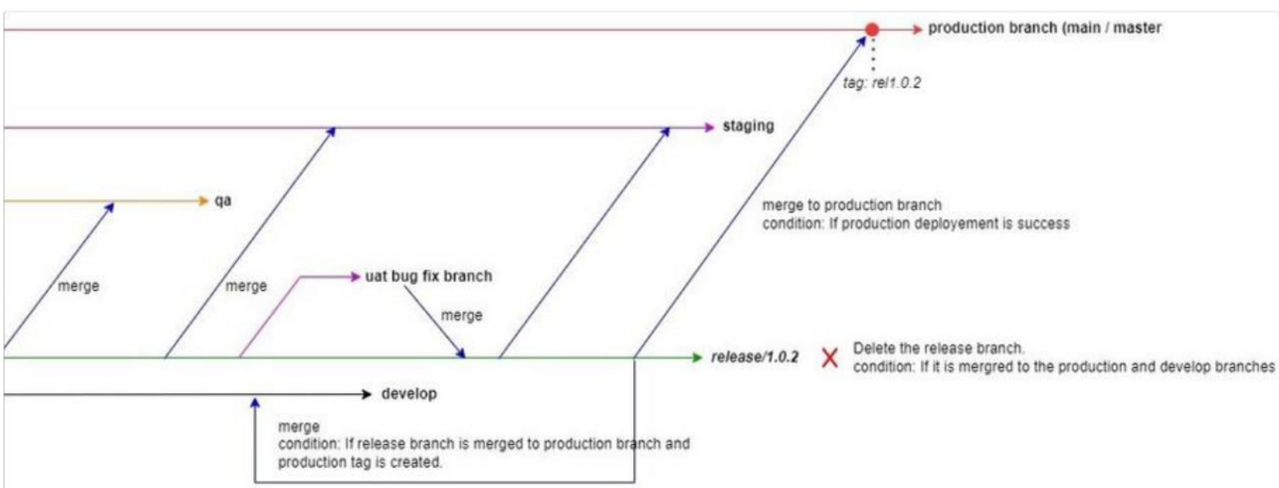
Once the quality assurance test is completed and passed, the same release branch should be merged to the staging code flow branch for user acceptance testing (UAT). If any UAT issues reported, again those bugs should be fixed in separate bug fix branches by branching off from the same release branch and redeploy them to the staging environment for further testing.



Fixing UAT Bugs

Once the UAT testing is completed and approved for the production release by the stakeholders, first the release branch should be deployed to the production environment and verify it to make sure that the new production deployment is working as expected. Now the release branch should be merged into the production branch (master or main) and the new production release should be tagged in the production branch. The tagging format is same as for tagging, but with no prerelease information and the release prefix should be 'rel'. Examples: rel1.0.2, rel1.0.0, rel3.0.1

Note: Now the release branch can be deleted as it no longer needed.



Production Release

Hotfix Process

The hotfix releases are not preplanned are made to address the critical bugs or issues in the current production release. A hotfix is a quick fix for a significant bug that affects the majority of the users of the system for which there is no reasonable workaround. Hence should consider followings questions or criteria when deciding to go for a hotfix.

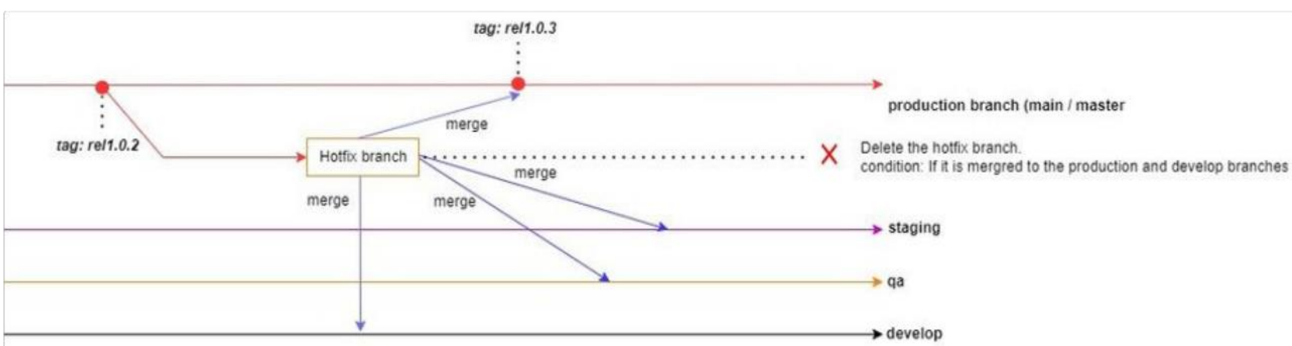
- Is the bug significant?
  - Does it produce incorrect results?
  - Is existing feature broken?
- Is the bug affecting to the majority of users?
  - Is it a common or major functionality failure?

  - Is it an edge or corner case?
  - Will the average user encounter the bug during normal usage?
- Is there a reasonable workaround?
  - Can a potential workaround be communicated effectively to the users?
  - Will the average user understand the workaround?
  - How many steps the workaround takes?
  - When the bug is fixed, will the workaround continue to work?
- Is it possible to include this fix to next release?
  - Is the next release within a few days?
  - Does the bug affect the entire user experience?

Checks the above criteria when considering a hotfix and the fix for these types of bugs should not be the proper fix as it is critical to fix them immediately, hence the developers should do a quick fix with minimal code changes within a limited time period. But still should update the relevant documentation and run unit tests. Further if the fix is not the proper fix, separate bug should be created to track it.

It is not mandatory to go through the standard testing cycles (QA and UAT) here as the it is critical to deploy the fix the production as quickly as possible. The hotfix branch should be branch offed from the correct version or the production release tag and once the fix is completed and verified the fix can be deployed to production and the hotfix branch should be merge to the production branch (*main or master*) and the *develop* branches. The production branch should be tagged with the new release version information and finally the hotfix branch can be deleted as it is no longer required.

Normally the patch component of the previous or issue reported production version is incremented for a hot fix release. But it is not a must even the minor version can be incremented if the change is significant. For more information about versioning refer to the versioning strategy guideline.



Hotfix Release

Secure Development

Templates

All the versioned templates under different areas are listed here. All new documents should be created with the latest version of the templates while existing documents shall continue with the older versions. It is recommended to migrate the documents to the latest version of the template, and it is a must when the older versions are discontinued.

List of Automations to do

1. All story points in stories should be summed up in their respective epics.
2. All time estimations in stories should be summed up in their respective epics.
3. When the task is moving to “Dev in-progress” it should be checked against all DOR checklist items.
    a. Estimation has to be done.
        i. Story points.
        ii. Time estimations.
    b. Assigned
    c. Attached the Wireframes or API swagger definitions.
    d. Description has the acceptance criteria.
4. Add a comment to the issue when it changes status.
5. Prevent epic closure until all its linked stories and sub-tasks are completed.
6. If issue has not been worked on in 7 days send a notification to the assignee
7. Get status updates every week

When moved to Dev in progress from Code review -> Assign it back to the original developer	DEV

Code-reviewer guide

A few general guidelines.

- A PR has to be resolved within 24 hrs. If it takes more than that, it should be reported to EM.

code-review flutter

References :

1. Performance best practices | Flutter
2. 13 Best Practices to Simplify Flutter App Development [2023] (intelivita.com)
3. flutter best practices - Google Search

Code review workflow

Things to consider.

code review process

- capture proper followup
- capture reviewer feedbacks in writing

comment structure - Company.atlassian.net/wiki/spaces/T/pages/112721968/Code+review+comment+structure

Evaluation Framework

Development - Evaluation critireas

Program quality metrics - Checklist

This software code quality guideline is a modified version of ISO 5055 best practices catering to our Agile Software Development Practices. For more information on what is ISO 5055 - please read here - What Is ISO 5055? | Perforce. Please get in touch with you Lead / EM for clarifications on this.

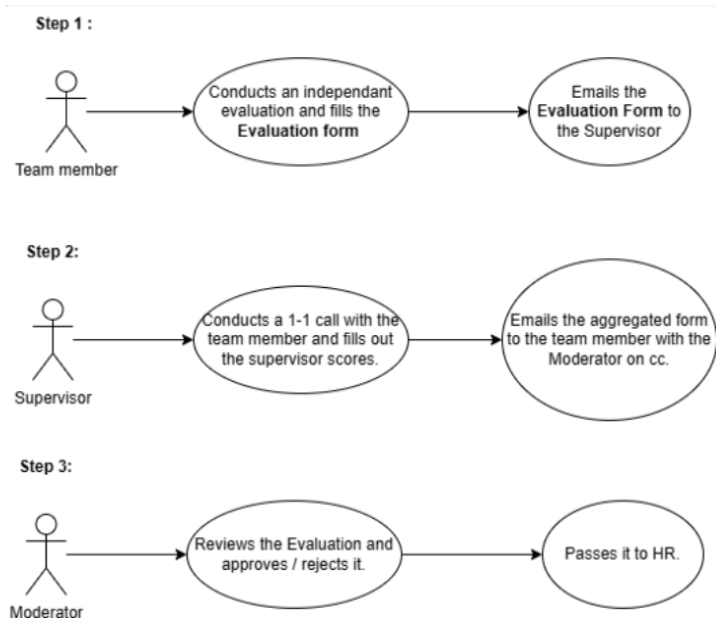
SOFTWARE QUALITY CHARACTERISTIC	CODING PRACTICES UNIT LEVEL	ARCHITECTURAL PRACTICES SYSTEM LEVEL
RELIABILITY (4)	☐ Protecting state in multi-threaded environments ☐ Safe use of inheritance and polymorphism ☐ Resource bounds management, Complex code ☐ Managing allocated resources, Timeouts	☐ Adhere to Multi-layer design compliance. ☐ Manage data integrity and consistency ☐ Handle Exception through transactions ☐ Adhere to Class architecture compliance
PERFORMANCE EFFICIENCY (5)	☐ Compliance with Object-Oriented best practices ☐ Compliance with SQL best practices ☐ Handling expensive computations in loops ☐ Handling static connections versus connection pools ☐ Compliance with garbage collection best practices	☐ Appropriate level of interactions with expensive or remote resources ☐ Handle Data access performance and data management ☐ Handle Memory, network and disk space management ☐ Centralized handling of client requests ☐ Use of middle tier components vs. procedures/DB functions
SECURITY (6 - 5)	☐ Avoid use of hard-coded credentials ☐ Handle Buffer overflows ☐ Avoid missing initialization issues. ☐ Handle validation of array index ☐ Avoid improper locking. ☐ Avoid uncontrolled format strings	☐ Do Input validation ☐ Secure against SQL injection ☐ Secure against Cross-site scripting ☐ Use vetted libraries or frameworks. ☐ Adhere to Secure architecture design compliance
MAINTAINABILITY (6 - 4)	☐ Avoid unstructured and duplicated code	☐ Avoid Duplicated business logic

	Handle high cyclomatic complexity. Handle controlled level of dynamic coding Avoid Over-parameterization of methods	Adhere to Strict hierarchy of calling between architectural layers Avoid Excessive horizontal layers

	Avoid hard coding of literals Avoid Excessive component sizes	Avoid Excessive multi-tier fan-in/fan-out

Evaluation Process Guide

Following is a step-by-step process how an individual member's performance evaluation should progress.

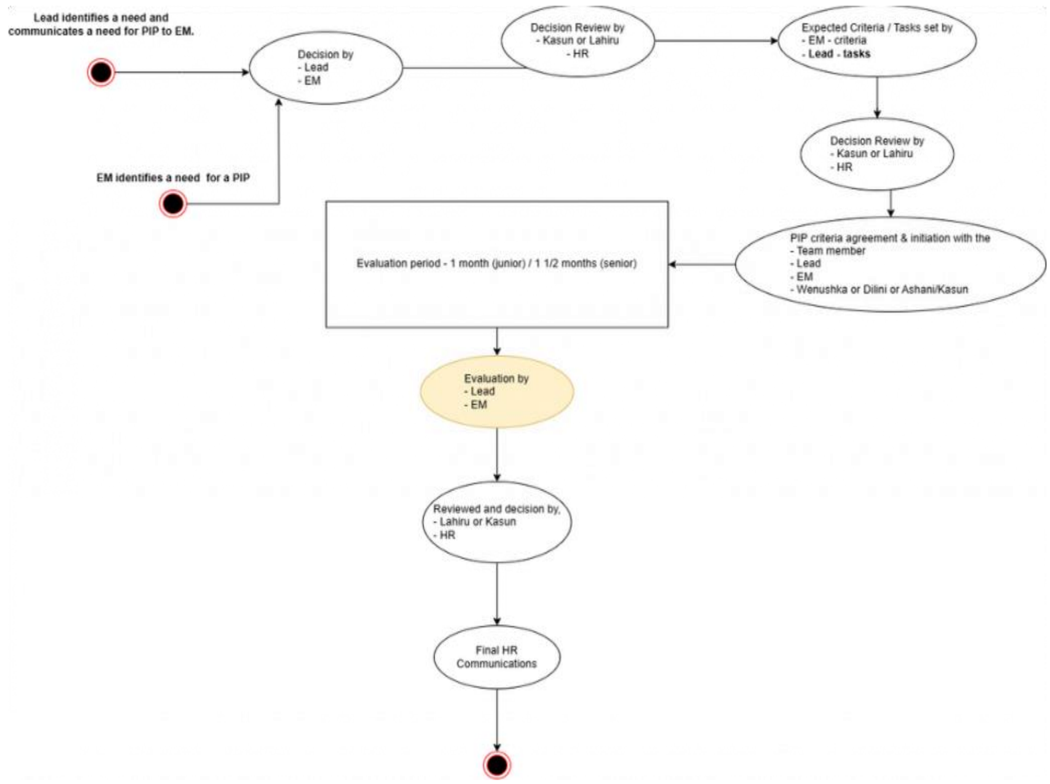


Evaluation Form - Developer

Form link - Evaluation form - Developer

Performance Improvement Plan

Here is the process.



Here is the evaluation form - Performance Evaluation Form

BA - Evaluation criteria

If any KPI output has a % value, the said value will be associated with the evaluation score ranging from 1-5

Percentage value	Score	Description
0-20%	1	Never meets expectations.
21-40%	2	Sometimes meets expectations.
41%-60%	3	Meets expectations.
61-80%	4	Sometimes exceeds expectations.
81-100%	5	Always exceeds expectations.

-

BA will be evaluated in 4 segments namely technical skills, Teamwork, Communication, and Delivery. The KPI(s) under each segment will assist the RO to evaluate the BA.

-

Technical Skills

KPI 1: Writing both functional and non-functional user stories.

Well defined user story:

- *User story format clearly followed (User story table/ As a, So that, I can/ etc.).*
- *Clear acceptance criteria with equal to or less than 8 lines per user story.*
- *Story point estimation per user story should be equal to or less than 8.*
- *Including diagrams/ wireframes or any other additional artifacts to support development.*

Substandard user story:

· *User stories which are not covered under the 'Well defined user story' will be considered as substandard user story.*

KPI 2: Use of elicitation or requirement gathering techniques.

When eliciting the requirement, the BA must follow at least 5 elicitation techniques altogether in all projects.

Examples

- Brainstorming.
- Document Analysis
- Focus Group
- Interface Analysis
- Interviews
- Observation
- Prototyping
- Workshops

Score	Description
1	Use of 1 technique
2	Use of 2 techniques
3	Use of 3 techniques
4	Use of 4 techniques
5	Use of 5 techniques

KPI 3: Requirement Analysis and Solution Design.

Score	Description
1	No participation in solution design discussions
2	Occasional participation in solution design discussions
3	Attending of all solution design discussions and providing inputs
4	· Actively participating and providing inputs in solution design discussions Providing solution diagrams/wireframes to support design discussion
5	· Inclusion of points discussed under score 4 and the below. · Solution proposed has, BA drafted wireframes (With all functions covered) CX (Customer journey, UI component placement, Policies etc) is clearly explained and defined. Followed with cross-functional teams for clarifications and clearance

Note: *Score 1-3* assesses the BA's attendance to solution design discussions.

Score 4 assesses the BA's attendance along with proactive action where the BA drafts a high-level solution design before attending the design discussion.

Score 5 assesses the BA's attendance along with proactive action where the BA drafts a complete solution design which consists of the entire user journey inclusive of wireframes/ flows/ policies etc.

KPI 4: Documentation Skills.

Score	Description
1	BA is not aware of available documentation guidelines and templates
2	BA is aware of available documentation guidelines and templates but not applied within the project(s)
3	· Has basic knowledge of word/ PowerPoint/ excel. Creation of documents Creation of presentations Creation of spreadsheets (e.g., For cost analysis) · BA is aware and has followed the predefined templates within the project. User story format User guide format Hand over document format etc. · BA documentation guidelines compliance is at least 80%
4	· Inclusion of points discussed under score 3 and the below, · BA documentation guidelines compliance is at least 90% · Conducted trainings/ awareness on documentation tools and techniques used to any member within Company
5	· Inclusion of points discussed under score 4 and the below. · BA documentation guidelines compliance is more 90% · Conducted trainings/ awareness on documentation tools and techniques used to any member within Company

	· Has suggested improvements to be incorporated into the documentation guidelines & templates which gets implemented in BA practice

KPI 5: Adherence to the BA Practices.

Score	Description
1	BA is not aware of available BA practices
2	BA is aware of available BA practices but does not apply within the project(s)
3	· Sharing daily work progress update [At least thrice a week] at the daily standup's and updating BA sync up board. · Participating in weekly BA sync up/trainings. · Completing assigned training materials · Conducting sessions for BA sync-ups [At least 2 per quarter]
4	· Inclusion of points discussed under score 3 and the below. ◆ Proposing new practices/ methods/ techniques which get approved for implementation [Idea generation and management approval. Idea to be submitted as a document/ via email]
5	· Inclusion of points discussed under score 4 and the below. · Implementing the new proposed practices/ methods/ techniques [Total solution implementation inclusive of Process flow diagrams/ Confluence page for process definition/ Trainings to peers etc]

<u>Teamwork</u>

KPI 6: Participation for sprint related ceremonies and activities (%).

<u>Communication</u>

KPI 7: Clear and concise client communication.

We will be sharing a questionnaire with the client as well as with the team lead/ architect to evaluate the communication skill of the BA who is assigned to the project.

Questionnaire attached under appendix 1.

Score	Description

1	Avg score of client feedback and team lead/architect feedback on BA communication on quarterly basis is 1
2	Avg score of client feedback and team lead/architect feedback on BA communication on quarterly basis is 2
3	Avg score of client feedback and team lead/architect feedback on BA communication on quarterly basis is 3
4	Avg score of client feedback and team lead/architect feedback on BA communication on quarterly basis is 4
5	Avg score of client feedback and team lead/architect feedback on BA communication on quarterly basis is 5

We will be only considering the floor value of the avg feedback score.

Example: If the total score is 54 and there are 12 questions. Then avg score = (54/12) = 4.5

Then the final score would be considered as 4.

Delivery

KPI 8: Percentage of on-time delivery of pre-sales activities (project scoping etc.).

KPI 9: Percentage of on-time delivery of project documentation (Handover document/ User manual etc.).

Appendix 1: Communication Feedback form

BA Name:		
Project:		
Filled by:		
Date:		
Questionnaire		
No	**Question**	**Responses**
1	On a scale of 1 to 5 (1 being lowest, 5 being high), how would you rate the Business Analyst's ability to articulate complex ideas clearly and concisely?	1 Poor 2 Fair 3 Good 4 Very Good 5 Excellent
2	How effectively does the Business Analyst tailor their communication style to suit the audience (e.g., technical teams, non-technical stakeholders, executives)?	1 Not Effective 2 Somewhat Effective 3 Moderately Effective 4 Very Effective

		5 Extremely Effective
3	To what extent does the Business Analyst demonstrate active listening skills during meetings and discussions?	1 Rarely 2 Occasionally 3 Sometimes 4 Often 5 Always
4	Rate the Business Analyst's proficiency in writing clear and comprehensive documentation (e.g., requirements, user stories).	1 Poor 2 Fair 3 Good 4 Very Good 5 Excellent
5	How well does the Business Analyst use visual aids (charts, graphs, diagrams) to enhance written communication?	1 Not at all 2 Occasionally 3 Sometimes 4 Often 5 Always
6	Does the Business Analyst effectively communicate complex technical concepts to non-technical stakeholders in written documents?	1 Rarely 2 Occasionally 3 Sometimes 4 Often 5 Always
7	How well does the Business Analyst facilitate and lead discussions in meetings, ensuring all voices are heard?	1 Poorly 2 Fairly 3 Well 4 Very Well 5 Exceptionally Well
8	Rate the Business Analyst's ability to handle questions and challenges during presentations or discussions.	1 Weak 2 Average 3 Above Average 4 Strong 5 Outstanding
9	To what extent does the Business Analyst encourage collaboration and open communication within the project team?	1 Rarely 2 Occasionally 3 Sometimes 4 Often

		5 Always
10	How well does the Business Analyst establish and maintain effective communication channels with stakeholders?	1 Ineffective 2 Somewhat Effective 3 Moderately Effective 4 Very Effective 5 Extremely Effective
11	Rate the Business Analyst's ability to manage and resolve conflicts that arise during stakeholder interactions.	1 Poor 2 Fair 3 Good 4 Very Good 5 Excellent
12	Does the Business Analyst provide timely and clear updates to stakeholders regarding project progress and changes?	1 Rarely 2 Occasionally 3 Sometimes 4 Often 5 Always

BMS Team KPIs for 23/24

1. **Preparation:**
   - **Team Goals:**
   - **Review Previous KPIs:**.
2. **Purpose:**
   - is to enhance analysis efficiency, accuracy, and overall contribution to the brave BMS objectives.

Context:

- **Brave Goals:**.
- **Team Goals:** Discuss a specific project, such as optimizing a data, and how it ties into broader team goals.

Interactive Discussion: ◦ Common KPIs:

- Metric 1:
- Metric 2:

◦ Individual KPIs:

- Metric 1:
- Metric 2:

Prioritization and Consensus: ◦ Voting or Ranking:

- **Consensus Building:** Encourage team members to discuss the importance of each metric and work towards a shared understanding.

SMART Criteria:

- **Specific, Measurable, Achievable, Relevant, Time-Bound:**
  - Specific: "Increase data accuracy by 10%."
  - Measurable: "Complete 95% of projects within the given timeline."
  - Achievable: Ensure that the team has the necessary resources to meet the targets.
  - Relevant: Align the KPIs with the overall goal of improving data analysis efficiency.
  - Time-Bound: "Achieve the accuracy target within the next quarter."

Documentation:

- **Document Decisions:** Create a document summarizing the chosen KPIs, including the rationale and any specific targets (HR/Lead).
- **Share Documentation:** Distribute this document among team members (HR).

Feedback and Adjustments:

- **Open Door for Feedback:** Encourage team members to provide feedback on the selected KPIs and be open to making adjustments.
- **Iterative Process:** If a KPI is not proving effective, be willing to revisit and revise it.

3. **Communication of KPIs:** Send a team-wide email detailing the finalized KPIs, explaining their significance, and reiterating how they contribute to the team and organizational goals.

Regular Review:

- **Scheduled Check-Ins:** Plan monthly or quarterly check-ins to review progress on each KPI, discuss challenges, and make any necessary adjustments.

Scrum Ceremonies

Daily Standup

The daily stand-up is a short, daily meeting to discuss progress and identify blockers. The reason it's called a "stand-up" is because if attendees participate while standing, the meeting should be kept short.

Duration: 15 minutes (for a team of 6 engineers + BA)

Here is a standard structure for your daily update.

- What did I work on yesterday?
  - What's the current status of it - briefly.
  - Declare if you have to change the due date for the task.
- What am I working on today?
  - Declare when you are expecting to complete it.
- What issues are blocking me?

Post meeting outcomes

Backlog Review

Backlog review (grooming / refinement) is the process of reviewing, ranking, and editing the product backlog. This helps the development team to prioritize and build the features and functionalities that the product owner wants and the business needs.

Backlog review vs sprint planning

Sprint planning and backlog review rely on each other. Backlog review ensures that the backlog is reflective of lessons learned, customer insight, and business value. In sprint planning, the development team pulls work from the backlog and commits to deliver that work in the upcoming sprint. Sprint planning is reliant on a refined backlog and backlog review gets the highest-priority work into the next sprint. Product owners should refine the product backlog before each sprint planning meeting.

Participants

- PO / BA
- SM
- Dev & QA team

Things to be done on the backlog review day.

- 100% physical attendance from everyone in the team is a must.
- Book the meeting room.
- Do dependency review for all not-estimated tasks in “Not Ready”.
- Move all estimated stories to “To Do”.
- Create a dependency summery for the reviewed items.
- Create a tentative sprint backlog for the next sprint.

Things to be added to the backlog review process.

- Release plan review should be done as the last step of the review process.

Sprint planning

What is a sprint planning?

Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.

Participants

- PO / BA
- SM
- Dev & QA team

Things developers should concern.

- Keep the dev cycle less than 5 days.

Things QAs should concern.

- Keep the QA cycles less than 3 days.

- Each team member should at least have 2 items assigned to them.
- Each team member should be connected to [adjusted velocity * 3 / number of individuals]

Outcomes

- Sprint plans share among all stakeholders including PO.
- Candidate task list.

Release Planning

A release plan in Scrum reflects the features in the backlog to be worked on in the upcoming sprints and provides an approximate date for the release. The plan should include the responsibilities, resources, and activities needed for each release and their potential priority.

The release plan is broken down into multiple sprints and what is to be accomplished in each sprint is recorded. The release date is estimated based on the number of sprints needed - multiplied by the velocity of the team. A release plan can help to show the feature-driven development and also remind everyone of the fixed-scope of each sprint.

Participants

- PO
- BA
- SM
- Sometimes members of Dev & QA team

Benefits of release plan

- The team gets a shared understanding and a shared vision of what needs to be done.
- Can be shown at the end of the Sprint Review to show the upcoming major releases and the progress made.
- The product owner gets guidance for prioritizing stories and tasks.
- Team members don't get off topic as quickly because they don't do unplanned work.

Sprint Review

What is a sprint review?

A sprint review is a collaborative meeting that is held at the end of every sprint (10th day of the sprint). This is when the team runs through work items they completed during the sprint or iteration. A sprint review ensures key stakeholders are up to date, and it enables them to provide feedback. At the end of each sprint review, the BA uses this feedback to create necessary work items in the product backlog.

Participants

- PO -
- SM -
- Dev team -
- QA team -

Agenda

Agenda item	Responsible
Sprint plan progress overview against the sprint plan in Confluence.	SM -
Demo items in the done lane by QAs.	QAs -
Meeting notetaking	BA -

<u>Post meeting work</u>

Work	Responsible	Participants
Meeting-notes review session.	SM - SM	Dev Lead - Developer BA - BA
Send the reviewed meeting note (with placeholders for tasks) to PO.	BA - BA	
Convert action items in the meeting note to Jira tasks.	BA - BA	

Sprint Retrospective

Key reports to discuss at the meeting.

- QA reports
- Velocity report.
- Burndown chart.
- Devops metrics.
    - Deployment frequency.
- Identify devops metrics to evaluate and add it to sprint retro practice.

Release plan review session

- Do story level allocations.
- Ensure tsizing completion.
- Change epic epic assignments for epics where stories are completed.
- Send the updated release plan to the team and all stakeholders.
-

Best practices

Development Best Practices

- Use pair programming when neccesary
- Do TDD.
    - Explain how a regular TDD flow goes.
- 100% resources must be on cloud. Provisioned and deployable.
- Use IAC.
- Write as much as reusable modules.

QA best practices

1. Keep the test cycles shorter than 3 days. Need to consider this during planning.
2. Write and run NFRs.
    a. Automate them. On a story level or on an application level.
3. Write at least the happy path test cases before development starts.
4. Test with production-like data.

Devops Best Practices

- Can you deploy at the click of a button.
- Can you roll back at the click of a button.
- Can you provision and release infrastructure at the click of a button.?
- Do you use IAC and version control for your infrastructure.?

BA Best practices

- An epic should be requirements cleared when taking into a sprint.

Logging & Monitoring

Code review comment structure

Comment title - [cycle name]

Commend body:

Changes done:

[Explain the changes in a pointed format]

eg:-

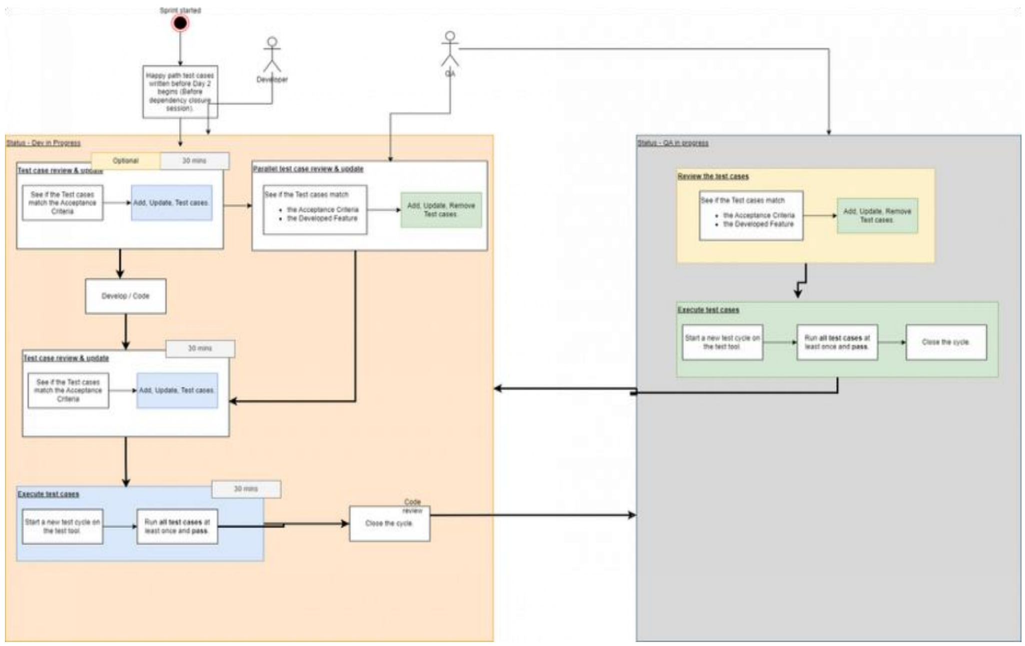
dev-test cycle 2

changes done:

- Added field x to the database to store data y.
- Changed button color red to gray.

Test case driven development

- Set outlook alerts for Jira AIO assignee emails.
- Following is a flowchart explaining the process.
- Do dev testing in the local environment.
  - Code review comment structure
  - To be added to the process - add a comment in the above structure when doing a release.
  - Or give a training for QAs to check github PR comments.
- The cycle name should be [Dev/QA] - [sequence number 0 - n] - [some other information about the cycle]



Triages

- Do triages. Introduce a process around this concept. This can be used to make mid-sprint scope adjustment decisions.

Default due dates per transition

These due dates will be set automatically by default. The assignee can then change the date manually to reflect the actual date of next transition.

- QA IN-PROGRESS → DONE / QA IN-PROGRESS → DEV IN-PROGRESS (Under QA responsibility)
  If status changed to QA IN-PROGRESS before 10:00
    due by EOD
  If status changed to QA IN-PROGRESS after 10:00
    due by next day EOD.

- DEV IN-PROGRESS → DEV CODE-REVIEW (Under lead responsibility)
  If status changed to DEV CODE-REVIEW before 10:00
    due by EOD
  If status changed to DEV CODE-REVIEW after 10:00
    due by next day EOD.

- DEV TODO → DEV IN-PROGRESS (Under dev responsibility)
  - if SP = 1
    - If status changed to DEV IN-PROGRESS before 10:00

      due by EOD
    - If status changed to DEV IN-PROGRESS after 10:00

      due by next day EOD.
  - if SP = 3
    - If status changed to DEV IN-PROGRESS before 10:00

      due by +1 day EOD.
    - If status changed to DEV IN-PROGRESS after 10:00

      due by +2 days EOD.
  - if SP = 5
    - If status changed to DEV IN-PROGRESS before 10:00

      due by +1 day EOD.
    - If status changed to DEV IN-PROGRESS after 10:00

      due by +2 days EOD.
  - if SP = 8
    - If status changed to DEV IN-PROGRESS before 10:00

      due by +2 day EOD.
    - If status changed to DEV IN-PROGRESS after 10:00

due by +3 days EOD.

- if SP = 13
  - If status changed to DEV IN-PROGRESS before 10:00

    due by +4 day EOD.
  - If status changed to DEV IN-PROGRESS after 10:00

due by +5 days EOD.

- Any → ON HOLD (Under TPM responsibility)
    - If status changed to DEV CODE-REVIEW before 10:00
        - due by EOD
    - If status changed to DEV CODE-REVIEW after 10:00

      due by next day EOD.

Assumptions

- We are currently not setting due dates for Dev todo & QA todo statuses.
- Automated due date defaults are a rough estimation. The actual due date could be largely different from the automated setting. Team members can update the due date after automation sets due dates to reflect the actual date.

QA

General Checklists

The checklists includes the check points which contains when QA needs to consider when proceeding API, Web, Mobile Testing.

Test Case writing - General guideline

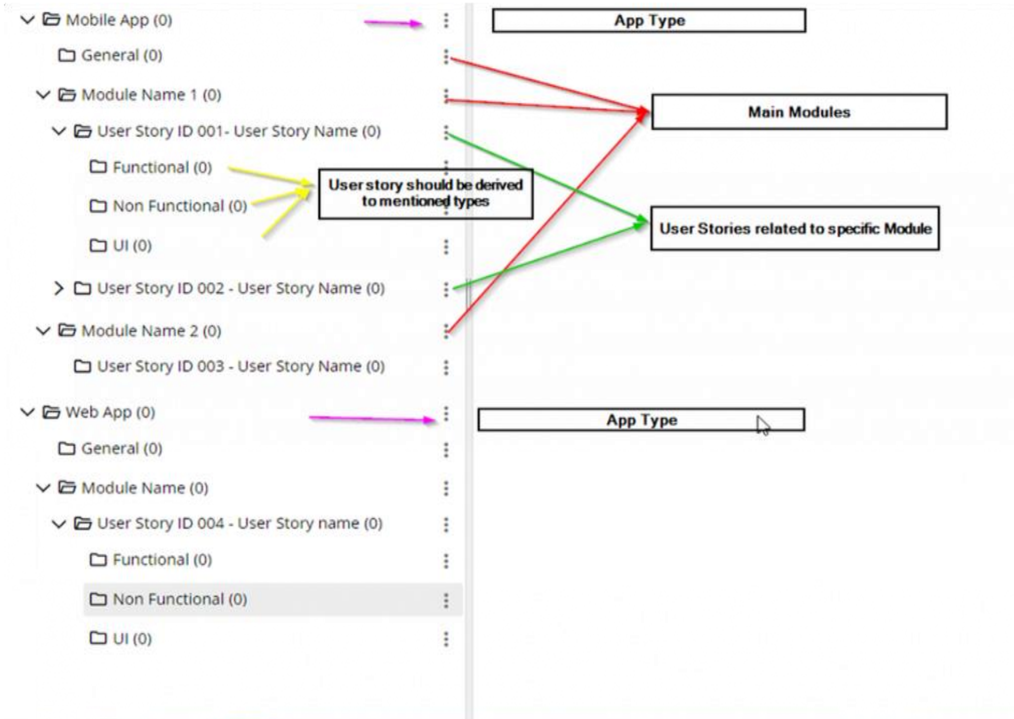
Focus Criteria	Expected coverage
Understand Requirements	• Thoroughly understand the requirements and functionality of the software.
	• Clear understanding of what needs to be tested and the expected outcomes.
	• Get the Clear Idea of acceptance criteria and points described
	• Analyze and Clarify the requirement in grooming sessions
Test Scenario Planning	• Plan the Test scenarios by reading acceptance criteria
	• Do not copy acceptance criteria as it is for the test case
	• Focus on positive and Negative functionalities
	Include test cases for boundary and edge conditions.
	• Test the limits of the system's capabilities to identify potential issues.
	• Include negative test cases to validate error handling and system responses to invalid inputs.
	• Check how the system behaves under unexpected or erroneous conditions.
Use a Standard Template	• Use AIO for writing test cases
	• Follow the AIO format
Test Case Title	• Use descriptive, short and unique titles for test cases.(Do not write lengthy titles)
	• Titles should indicate what functionality or scenario is being tested.
	• Title should start with "Verify the" or "Verify whether" (Rarely we can drop these as per the test scenario covering)

Test Case structuring	• if it is a UI related test then start the test with the tag [UI] then module name [Module Name] then verification
	Sample - [UI][Search Field]- Verify whether the Search option displays next to search Bar field
	• If it is functional test case then [Module Name] [Sub Module Name]- Verification (functional tag is not adding to functional test cases since most of the test cases are to cover functional)
	Sample - [Search Field]- Verify the search result by entering a search term with special characters
	• If it is Non functional then [NonFunctional] [Module Name] then verification
	Sample - [Non Functional][Login]- Verify the application behavior when login user count >100
	• Should follow the AIO Folder structure
Test Preconditions	• Clearly state any prerequisites or preconditions necessary for executing the test case.
	• Ensure the system is in a specific state before the test case is executed.
	• If it is highlighted Pre Req the better to add it as the first step
	Sample - Pre Req- User should be logged in as a admin to view the "Specific feature name"
	• If it is general pre requisite then better to add it in pre req field in AIO
	• Do not cover main steps in pre requisites
Test Steps	• Write detailed step-by-step instructions to reproduce the test scenario.
	• Include specific input data, actions to be performed, and the expected outcomes.
	• Do not write more and more steps to cover the navigation steps-either cover it by pre req or maximum 1 or 2 steps
	• Always write the test steps by thinking the executor has no knowledge in Domain

Test Data	• Always try to add test data or SQL query to preparation test data
	• Always try to validate data with DB
	• Handle data dependencies and ensure data consistency across test cases.
Expected Result	• Clearly define what the expected results should be after executing each step.
	• Be specific about the outcomes, including any messages, errors, or visual cues.
End Verification	• Always last step should be a verification step and it should matches the title
Reusability	• Write test cases that are modular and can be reused for regression testing.
	• Avoid hard-coding values whenever possible.
Review and Revision	• Have test cases reviewed by peers or stakeholders to ensure accuracy and completeness.
	• Revise and update test cases as the application evolves and requirements change.
	• Focus on previous given comments/feedback when writing test cases
	• Do not keep test case just moving to under review. Contact or communicate to someone responsible or share a report with leads
Communication	• Communicate effectively with the product owner/ Dev leads to clarify doubts and resolve issues.
	• Foster a collaborative environment between QA and development teams

AIO Folder structure

AIO Confluence page Ref- AIO Tests - All-In-One Test Management for Jira



Folder type	Description	
Module Name	Main Module where user stories related to	
User story Id	Jira User Story ID	
User Story Name	User story title or covering feature name	
Functional	All the Functional Test Cases Shoud be added in this folder	
Non Functional	All the Functional Test Cases Shoud be added in this folder	
UI	All the UI Test Cases Shoud be added in this folder	
Note	If there are no specific module then create a folder call General and add test cases in to that	If the application has both mobile app and web app and API coverage then create different folders to maintain test cases

API Testing Checklist

Focus Area	Test scenario	
Functional Testing	**Endpoint Testing**	Verify that each API endpoint functions correctly and returns the expected responses.
	HTTP Methods	Test all supported HTTP methods (GET, POST, PUT, DELETE, etc.) for each endpoint.
	Request Parameters	Test various combinations of request parameters, including required, optional, and edge cases.
	Authentication and Authorization:	Ensure that authentication and authorization mechanisms work as intended.
	Status Codes	Verify that the API returns appropriate HTTP status codes for different scenarios (e.g., 200 OK, 400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found, 500 internal server error).
Data Validation	**Input Validation**	Check how the API handles invalid input data, such as missing or incorrect parameters.
	Data Integrity	Verify that data sent and received remains intact and unaltered during transmission.
	Data Formats	Ensure that data is in the expected format (e.g., JSON, XML, etc.) and adheres to defined schemas.
Error Handling	**Error Responses**	est error responses to ensure they provide useful and meaningful error messages.
	Exception Handling	Check how the API handles unexpected errors and exceptions.

Security Testing	**Injection Attacks**	Test for common security vulnerabilities like SQL injection and Cross-Site Scripting (XSS).
	Authentication and Authorization	Verify that only authorized users can access certain endpoints and data.
	Rate Limiting	Test if rate limiting and throttling mechanisms are in place to prevent abuse.
Integration Testing		Test how the API interacts with other components or third-party services it relies on.
		Ensure proper data flow between integrated systems.
Regression Testing		Continuously run tests to detect any regressions or issues introduced by code changes.
Version Compatibility		Ensure backward and forward compatibility with different versions of the API
Documentation Testing		Ensure that the API documentation is accurate, up-to-date, and aligns with the actual API behavior.
Environmental Factors		Test the API in different environments (e.g., development, staging, production) to catch environment-specific issues.

UI/UX Testing Checklist

UI checklist has been divided into the categories UI Testing, Ux Testing, Web responsive test and mobile responsive test

UI Checklist

Area	Conditions
Validation messages	Mandatory fields validation
	Characters validation
	Special characters validation
	Number validations
	Specific messages for difference scenarios
	Validate that error messages are displayed clearly and help users understand how to resolve issues.
Spellings	Check for correct capitalization
	The spellings are accurate and according to relevant language in each UI Component
Consistency	Color codes are correct
	Font style is according to the requirement
	Font size is according to the requirement
	Labels are according to the requirement
	Line spacing should be consistent
	Alignments are consistent
	Positioning is correct
	Date format is according to the requirement
Images	Image quality is sufficient
	Support for the different formats
	Cropping option are available or not
Browsers	Check if the UIs are according to the Figma in different browsers
Overlapping	Overlapping should not happen among UI component

Icons and Logos	Check for the icons and logos placement
Header	Header should be according to the requirement
Footer	Footer should be common for all the pages
Disabled/Enabled	Check for the disabled/enabled behavior of every UI component
Tooltip	Verify tool tip is available when hovering to the UI component
Resolution	Verify UI behavior under different resolution
Zoom	Entire UI should not change after zoom in or zoom out
	Verify the Screen appearance in Full screen mode
Navigations	Forward navigations should be able to to based on permissions
	Backward navigations should be able to do based on permissions
Scrollbars	Both horizontal and vertical scroll bars should appear based on page sizes
Mouse & Keyboard	Should be able to operate using Keyboards
	Should be able to operate using mouse

UX Checklist

Area	Condition
Usability	Evaluate the overall ease of use and user-friendliness of the UI.
	Test common user tasks to identify any bottlenecks or confusing elements.
Loading and Performance	Check loading times for various UI components.
	Test performance under different levels of user load.
Feedback and Progress	Ensure that users receive appropriate feedback when actions are performed (e.g., success messages, loading spinners).
	Test long-running processes and confirm that users are informed of progress.
Localization	Test the UI with different language and regional settings to ensure text translations and formatting are accurate.
Accessibility	Check if the UI follows accessibility guidelines (WCAG) for users with disabilities.
	Test with screen readers and keyboard navigation.
Consistency	Verify that design elements, language, and behaviors are consistent throughout the application.

Web Responsive checklist

Responsive design, in the context of web apps, refers to the approach of designing and developing applications in a way that allows them to adapt and provide an optimal user experience across various devices and screen sizes. This includes desktop computers, laptops, tablets, and smartphones. The primary goal of responsive design is to ensure that the web app's layout, content, and functionality remain usable and visually appealing regardless of the device being used to access it.

Context	Description
Fluid Grid Layouts	This ensures that elements on the web app's interface can adjust their size and position based on the screen width.uses percentage-based widths instead of fixed widths,
Flexible Images and Media	Images and media elements are scaled proportionally to fit different screen sizes. This prevents images from being too large or too small on different devices.
Media Queries:	Media queries are CSS rules that allow you to apply styles based on various characteristics of the device, such as screen width, height, and orientation. These queries enable you to customize the appearance of your web app for different devices.
Viewport Meta Tag	The viewport meta tag allows you to control how your web app is displayed on different devices. It lets you set the initial scale, maximum scale, and other viewport-related properties.
Content Prioritization	In responsive design, you often need to prioritize certain content on smaller screens due to limited space. This might involve reorganizing elements, hiding less important content, or collapsing menus into a mobile-friendly format.
Touch-Friendly Interfaces	Since mobile devices rely heavily on touch interactions, responsive design often includes optimizing UI elements for touch gestures, like swiping and tapping.
Testing Across Devices	Responsive design involves thorough testing on various devices, screen sizes, and orientations to ensure a consistent and pleasant experience for all users.
Progressive Enhancement	This design philosophy involves starting with a basic version of the web app that works on all devices, and then progressively adding more advanced features and styling for larger screens.

Ref- Responsive Web Design: What It Is And How To Use It — Smashing Magazine .

Mobile Responsive checklist

Responsive design in the context of mobile apps refers to the approach of designing and developing mobile applications in a way that allows them to adapt and function well across a variety of devices and screen sizes. The goal of responsive design is to provide a seamless user experience regardless of whether the app is being accessed on a smartphone, tablet, or other mobile devices.

Context	Description
Flexible Layouts:	Designing layouts that can adjust to different screen sizes and orientations. This involves using fluid grids and relative units for elements like fonts and spacing, so they scale proportionally.
Adaptive User Interface (UI):	Modifying the user interface based on the available screen space. This might involve rearranging or reorganizing elements, changing navigation patterns, and adapting the content hierarchy.
Media Queries:	Utilizing CSS media queries to apply different styles and layouts based on the device's characteristics, such as screen width, height, and resolution. This allows developers to create specific designs for various breakpoints.
Touch-Friendly Interactions:	Designing interactive elements and controls that are easy to tap, swipe, and interact with using touch gestures. This includes ensuring that buttons are appropriately sized and spaced to accommodate fingers of different sizes.
Optimized Images and Assets:	Delivering images and media at appropriate resolutions and formats based on the device's capabilities. This helps to balance visual quality with loading times.
Progressive Enhancement:	Building the app's functionality in layers, so that even on devices with limited capabilities, the core functionality is still accessible and usable.
Performance Optimization:	Paying attention to performance considerations, such as minimizing unnecessary animations, reducing server requests, and optimizing code for faster loading times.
Cross-Device Testing:	Thoroughly testing the app on a variety of devices, screen sizes, and orientations to ensure consistent performance and usability across the board.

Mobile Testing Checklist

This has been divided into three main categories to make things more clear. The main categories are UI, Functional and General

UI - Checklist

Verifications	Category
• Check whether all the mandatory fields are indicated as per requirement	**Validation**
• Check whether app validate mandatory field when left blank	
• Check whether validation message shown (If applicable)	
• Check whether app validate input data types (If applicable)	
• If message is given : must be compatible with given text	**Must check all the messages (Success, error, warning etc.) text**
• If message is not specified: Suitable generic message list must be prepared and verify accordingly	
Check spellings of the text	**Spellings**
If UK Project-Spelling must be in UK English	
If US Project - Spelling must be in US English	
• Check consistency of the UI components	**Consistency**
01. Font and font sizes	
02. Button label, colour, sizes	
03. Field label, colour, sizes	
• Date format (country wise if not specified)	
01. Date must be in the specified format	
02. IF Date format not specified, date should be formatted according to the project country	
• Check ability to use and access mobile feature (camera, gallery etc) when required (When devise permission granted)	**Mobile Permission**
• Check restrict access mobile features (camera, gallery etc) (When devise permission not granted)	

• Upload different types of images	**Images**
• check Image quality (Pixelate, resolution)	
Check whether are there any auto re size, cropper or fit to screen functionality available	
• If cropper functionality available, must be able to adjust	
• Must check with different size images	
• Must check image quality after crop	
Must check text compatibility with design	**Text**
• Font size	
• Font colour	
• Font family	
• Text decoration	
Browser compatibility	**Browse**
• Inner browser UI compatibility and responsive	
• Email functionalities	

Functional Checklist

Verification	Category
Check whether app demand unnecessary permission at the begging of installation	**Device permission**
(Location permission, subscription etc)	
If product has requirement that user must be able to use app offline(work app without internet)	**offline behaviors**
• Must identify what are the features must be able to use offline	
• Must test app functionality and behaviours without internet	
Check data loading time	**Data Loading**
• Check average time take to perform a search (Data loading speed/time)	
• Check impact on memory	
• Check maximum data load (up to crash)	
• Loading time	
• crash/time out point	
• Check lazy loading behaviours	
• Check whether data loading happens simultaneously when scroll up/down	
• Check whether relevant data loading happens as expected (text, images)	
• Check whether data loading hinder user experience (smoothness)	
Note: Use script developed by devs	
Speed	**Check Uploading Time**
• app crash point	

Must clarify are there any exception before start testing	**Exception**
• If there are exception, must test all those scenarios	
Check application loading times	**Performance**
This should check device wise	**Gesture functionality must check**
Must check smoothness when touch	
• Check whether notification shown for both app types (iOS and Android)	**Notification behaviours must check**
• Check whether notification shown for both app types at the same time (iOS and Android)	
• Check whether notification shown upon WIFI/mobile data on for both app types (iOS and Android) - when offline	
• Check whether notification shown when app open	
• Check whether notification shown when app close	
• Check whether notification shown when app run in background	
• Check whether Opening app through notification	
• Check whether redirect app feature through notification (if applicable)	
Block screen (when update release) behaviors must check	
Check whether app allow to send email in invisible mode	**iOS Specific**

General Checklist

Verification	Category
Test the installation and uninstallation process of the app	**Installation and Updates**
Verify that updates or patches do not break existing functionality.	
Should get the app from test environment for testing	**Test Environment**
Tester must get the app from below locations for testing	
iOS - Test flight	
Android - google Play Console	
Note: Should not test app from dev build. Tester must get the testing app from the respective test site	
Check that the app functions on older devices and operating system versions.	**Back Ward Compatibility**
Must clarify whether app should allow either dark or light mode or both	**Dark /Light Mode**
IF So,	
• Check whether user is able to change the mode	
• Check text colour changes	
• Check text design changes	
• Check component colour changes	
Must clarify are there any mode must be blocked	
IF So,	
• Check whether app allow to change mode	
Check whether are there any requirement to block either Portrait or Landscape mode	**Portrait and Landscape**
• If so,	
• must identify applicable devices types (smart phone, tab, etc)	

• must check this has been implemented	
• Check UI in Portrait mode and landscape mode	
IF Landscaped mode blocked	
• Check whether the device allow landscape mode (app must not auto re size the screen in to landscape(	
• App should not depend on device configuration and must not allow landscape	
Email password or OTP Login	**Check whether app allow to sign in with following methods (if appliable)**
Social login (google, fb, apple etc)	
finger print	
facial recognition	
Check social logo, icon and text when use social sign in / sign up	**Check apple logo and text when use apple sign in / sign up**
Follow all the relevant guideline specified by the org.	
Android	**Must test on multiple devices**
• Smartphone	
• Tablet	
iOS	
• iPhone	
• iPad	
Standard screen sizes for testing	
Android	
• Version : API Level 21 / Android 5 (Lollipop) or later version	
iOS	
• Version : iPhone 11 or later version	
Note: mention the tested and agreed devices and	

tested scenarios	
01. We agreed app will support iPhone 11 or later version	
02. We use all the devices what we have and others check simulators and emulators in oder to cover maximum device types and sizes	
Must check following screen sizes (according to device availability)	**Screen Sizes**
ex- iOS	
• 13	
• 13 Mini	
• 13 Pro	
• 13 Pro max	
ex- Android devices	
• Privacy policy	**Check Policies and other statements (if applicable)**
• Terms and condition	
• Other	
Version of the application (Release management)	**App Version**
• Should check version number of the app (Version number must be mentioned in the app)	
01. Check the place where version mentioned in the app	
02. Check the app store	
• Check whether the version number is defined according to the release (Build, enhancement or bug fix)	
Tested scope and tested device types and sizes must be mentioned in the release note	**Release note**
Release note must include following information	
• Devices used for testing (Resolution and screen size)	
• Test scope	

Automation

NFRs

Automated NFRs testing planning and technologies to be defined and setup.

Defect handling

- Pre-done bugs.
  - create failing test cases.
- UAT bugs.
  - create a bug ticket.
  - tag the component field.
- Production bugs.
  - create a bug ticket.
  - tag the component field.
- Pair testing
  - e2e bugs.
    - create a story.
    - set ceiling value to limit the scope.
    - add bug tickets as test cases.
    - #define a workflow
  - Pre-done stories.
- Once UAT and Production bug received

  a. Proper investigation should be done and needs to share a postmortem report to the team

QA Deliverables

Within the Sprint

- Test Cases written for specific user stories (AIO should up to date)
- Test Execution summary for each sprint (AIO should up to date with Cycles)
- Sprint Review session (end of the sprint)
- Sprint Summary report/ Sprint Release note
- Automation Progress report (If project implemented the automation)

Major release to a client

This release note should be shared when a major release / After CR implemented or After client bug fixes

- QA Release Note

After UAT/Production Feedback

- Postmortem for the UAT Bugs
- Postmortem for the production bugs

Deliverable Formats

The sample format should be customized as per the project

Sample Sprint Summary Report/ Internal Release note

Sprint - Relase … te.docx
09 Nov 2023, 02:56 AM

Sample Client Release Note

Release Note.docx
08 Nov 2023, 07:55 AM

Postmortem update for UAT/Production bugs

Postmortem Up…te.docx
08 Nov 2023, 08:03 AM

Resource planning

Resource	Count	Personnel name	Status
Senior Flutter developer	1	-	PENDING
Junior Flutter developer	1	Developer	OK
Backend Developer (NestJS)	2	Developer Developer2	OK
QA Engineer	2	QA QA2	OK
QA lead	1	-	PENDING
Business Analyst	1	BA	OK
Total			

QA Team and Allocation

This page is to maintain the QA team and their allocation

Internal QA Team

Team Member	Project Allocation	Working Areas
Dilini Dassanayake	All	
Dharaka Mahanthage	CDM PAL	• Engage with Test case writing, execution and Review • API Automation
Akmal Ahamed	CDM PAL	• Engage with Test case writing, execution and Review • API Automation
Chithma Anusaki	CDM PAL	• Engage with Test case writing, execution and Review • API Automation
Fathima Azka	CDM PAL	• Engage with Test case writing, execution and Review • API Automation
Priyadarshi	Transpomate	• Engage with Test case writing, execution and Review • Mostly focus of Mobile test cases
QA	Transpomate	• Engage with Test case writing, execution and Review • Mostly focus of web app test cases
Buwani Anupama	CADS	• Engage with Test case writing, execution and Review
Amna Haneef	CADs	• Engage with Test case writing, execution and Review
Nowshin Tabassum	New Joiner	Training materials
Niyumi Nawalaella	New joiner	Training materials
Mahamudul Islam	New Joiner	20th Nov

QA Team Engage with Clients

Team Member	Client	Working Areas
Yashika Gunasekara	Brave	

Sajitha Jithmi	Brave	

Piyumi Wanigasekara	Pencs	
Vindya Samaranayake	Pencs	
Sheikh	Pencs	Automation

Definitions

Definition of Ready

A definition of ready (DoR) is used to determine whether work on a item is ready to be started. It is mandatory that any item has to fulfill DoR criteria by the time it's moved to the Todo status.

Criteria to Qualify

1. An item can be nothing but a story or a production bug.
   a. Tasks has to be converted into a story.
   b. All bugs have to be reproducible in a production environment.
   c. All non-production bugs (eg:- Development, QA, Staging, etc.) has to be a part of a story that is WIP (Work In Progress).
2. Product owner must've signed-off the item. This should ideally be done in the backlog grooming session but its also possible to sign-off in any ceremonial/non-ceremonial meeting as well.
3. The item has to be estimated in story points. Estimation has to be finalized in the backlog grooming session.
4. The story point weight of the task has to be below 8 (Assuming Fibonacci estimation being used)
5. A development cycle for the story should be equal to or less than 5 days (excluding R&D)
6. A QA cycle for the story should be equal to or less than 3 days.
7. The task has to be clearly understood by the team (indicated by the story points).
   a. QA owners should have a clear idea of how to test the task. Test cases are not mandatory at this point.
   b. Developers should have a clear idea of how to develop the story.
8. The acceptance criteria have to be finalized and approved by the product owner the team has to have a clear idea of how to evaluate it during sprint review.
9. The story or bug has to have a business value signed-off by the PO.
10. All dependencies have to be identified and signed-off by the team.
    a. Development and QA leads has to sign-off the dependencies by development leads in a pre-planning session prior to backlog grooming session. The item should be signed off from the architecture lead (Either the Solutions Architect or Dev lead) and that should reflect in the relevant field in the item. The same goes with the QA lead sign-off.
11. There can't be more than 3-point story point allocations per story for R & D. If it is more than 3, it should either be split into separate tasks or created as a separate story with clear goals and acceptance criteria.
    a. In such cases, BAs can act as QA to verify the success of the R&D. SUGGESTION
12. NFRs must've been defined. Non Functional Requirements (NFRs)

What happens after.?

- When a story qualifies against the DoR, the task has to be moved to the "Todo" status from "Not Ready".
- The task is ready to be taken into a release planning and then to sprint planning.
- If there is a "Hold reason", it has to be cleared.

What happens when a story doesn't qualify against the DoR.?

- If the story doesn't qualify due to a blocker that isn't possible to clear during the backlog grooming session, a "Hold reason" has to be set. For, tasks outside the sprint, "On hold" status shouldn't be set.
- The team should work toward clearing the DoR disqualification reasons.

Dependancies

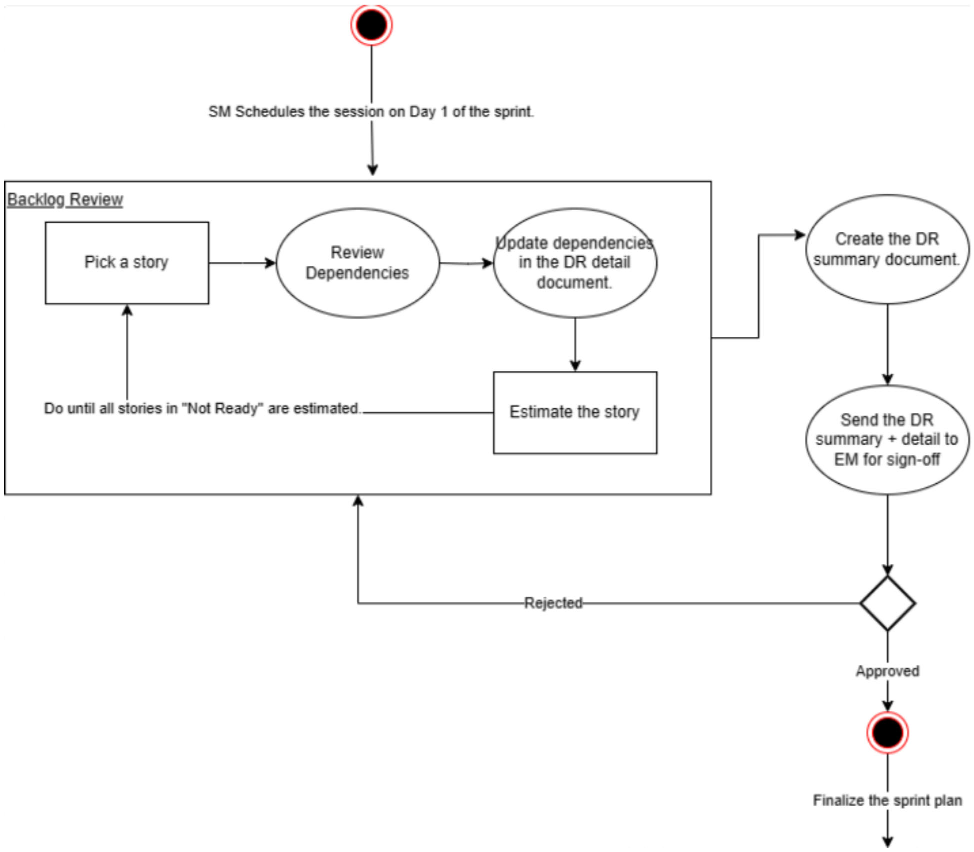
Following is a set of dependencies that has to be cleared before at different stages.

Dependency	Item	When to Review	Respon sible	Reviewer	Done.?
Requirements	User story BA diagrams Acceptance criteria	BEFORE GROOMING	BA	SM	Ok
UI	User interface - Figma	BEFORE GROOMING	BA	SM	Ok
Resources	Cloud resource provisioning & configuration (Dev environment) Cloud resource provisioning & configuration (QA environment) Devices Software Licenses. Subscriptions	BEFORE SPRINT ST...		Architect1	Ok
Skills	Checklist all necessary skills required to complete a task.		EM	pending - Architect1	Ok
Architecture	Architecture diagram (optional) Technical debt declaration. Sign-off	BEFORE GROOMING	EM	EM	Ok
Development	Database diagram Sequence diagram (optional)	BEFORE GROOMING	DEV TEAM	SM	Ok
DevOps	Deployment diagram	BEFORE GROOMING		SM	Ok
Quality	Happy path test cases	BEFORE PLANNING	QA TEAM	SM	Ok

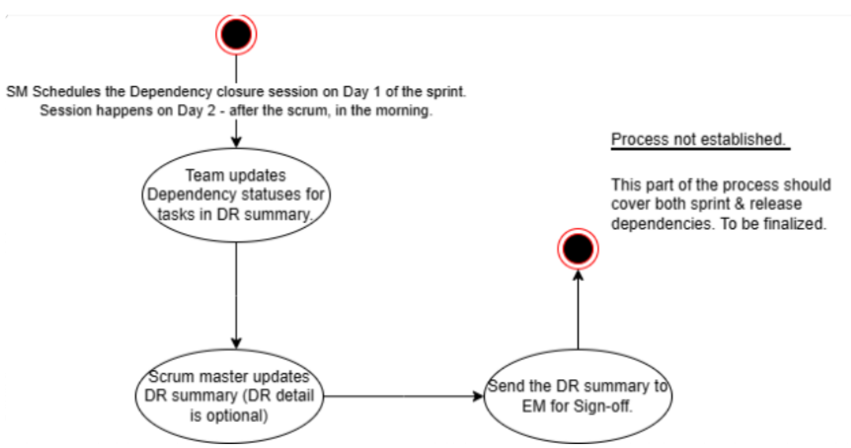
Pending items to be added:

- External API specs - potentially under requirements or development.

Dependency review Process



Dependency closure : Day 1 of sprint



Definition of Done

The “Definition of Done” (DoD) describes the quality standards for the Increment to be considered “done” and in a state that it can be effectively inspected. It creates transparency by providing everyone a shared understanding of what work was completed and what standards were met as part of the Increment. Without this crystal clarity, it’s not possible for stakeholders to provide informed and reliable feedback.

Criteria to qualify.

All test cases are automated. Every merge to the QA branch runs these test cases.	OPTIONAL	
Developer Test case runs.		
QA owner test case runs.		
Code review.		
Test cases run with high quality production-like data.		

Definition of Dev done

- At least one cycle of test cycle done.
- Added missing test cases.
- Code review completed and passed.

Definition of QA done.

- At least one cycle of test done.
- Add missing test cases from the previous run.

What happens when a story doesn’t qualify against the DoD?

If the story doesn't qualify against the definition of `Dev Done` and `QA Done`, the story shouldn’t be presented at the Sprint Review.

Story points

Story points are units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item or any other piece of work. The best definition of story points is that they represent the **effort** to develop a user story or product backlog item. Effort is a question of time: how long it will take to finish something. Many factors go into determining effort, including

- The amount of work to do
- The complexity of the work
- Any risk or uncertainty in doing the work

When estimating with story points, many things come into play: complexity, effort, risk, and volume. But ultimately, story points are an estimate of effort.

When estimating with story points, a point value will be assigned to each item. Some teams use a **modified Fibonacci sequence** (1, 2, 3, 5, 8, 13, 20, 40…); others use a **doubling sequence** (1, 2, 4, 8, 16…). The raw value assigned to story points is not important, what matters are the *relative values*. A user story that is assigned two story points should be twice as much effort as a one-point story. It should also be two-thirds the effort of a story that is estimated as three story points.

One of the main reasons story points are so valuable is that they allow team members with different skill levels to communicate about and agree on an estimate. Instead of arguing about how long it might take each team member personally to do something, teams instead can quickly say that this user story is about twice or three times as much effort as that user story. With story points, it's all relative.

Examples

Story points	Example stories
1	• TN-612: [CR] As a Transport Coordinator, I need to view all scheduled voyages, So that I can manage and coordinate scheduled voyages effectively DONE
3	• TN-619: [CR] As a Driver, I need to see the pending voyage, So that I can plan and prepare for the voyage DONE • TN-616: [CR] As a Driver, I need to take a picture of the odometer reading, So that I could send it to the Transport Coordinator DEV - TODO
5	• TN-598: As an admin, I need to search for users, So that I can assign the searched user with a role (assigning the role isn't included) DONE • TN-499: As a user, I need to log into the TMS system, So that I can access the system and its features/functions. DONE • TN-53: As a Driver, I want to view all the completed voyages, so that I can review the details of the past voyages DEV - TODO
8	• TN-718: As a Manager, I need to view trip requests in a way that distinguishes between my own trip requests and those of my subordinates DEV - TODO • TN-602: As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app DEVELOPMENT IN PROGRESS
13	• TN-591: As a Transport Coordinator, I need to receive an automated email, So that I know that the voyage has started DONE

Learn from past estimates.

Retrospectives are a time for the team to incorporate insights from past iterations–including the accuracy of their estimates. Many agile tools track story points, which makes reflecting on, and re-calibrating estimates a lot easier. Try, for example, pulling up the last 5 user stories the team delivered with the story point value 8. Discuss whether each of those work items had a similar level of effort. If not, discuss why. Use that insight in future estimation discussions.

Remember the Definition of Done

A story point estimate must include everything involved in getting a product backlog item all the way to done. If a team's definition of done includes creating automated tests to validate the story (and that would be a good idea), the effort to create those tests should be included in the story point estimate.

Further readings:

- What are story points in Agile and how do you estimate them?
- What Are Story Points and Why Do We Use Them?

10 Types of Technical debts

Technical debt is a concept that plays a significant role in software development. It represents the compromises made in the software's design, code quality, or architecture to meet short-term goals or deadlines. While taking on some technical debt is often necessary, accumulating too much can lead to long-term problems. In this article, we'll explore the various types of technical debt that could exist in a software project, helping you identify and manage them effectively.

Code Debt:

This is perhaps the most common form of technical debt. Code debt includes poorly structured or undocumented code, inconsistent naming conventions, and shortcuts taken to meet tight deadlines. Over time, code debt can make the software difficult to maintain and extend.

Design / Architecture Debt

Design debt refers to compromises made in the software's architecture or overall design. It might involve taking shortcuts to expedite development without considering long-term consequences. This can lead to rigid or inflexible architecture that's challenging to modify as requirements evolve.

Testing Debt

Testing is a critical aspect of software development. Testing debt occurs when there's insufficient or inadequate testing due to time constraints or a lack of proper test cases. This can result in undetected bugs and issues that become more challenging to address as the project progresses.

Documentation Debt

Incomplete or outdated documentation is a common form of technical debt. It can hinder new team members from understanding the software's functionality and structure. Documentation debt often arises from neglecting to update documentation when changes are made to the software.

Dependency Debt

Many software projects rely on third-party libraries, frameworks, or components. Dependency debt arises when these external dependencies are not properly managed. This can lead to compatibility issues, security vulnerabilities, and difficulties when upgrading or replacing components.

Performance Debt

Performance debt occurs when software performance optimization is postponed. This could involve inefficient algorithms, resource-intensive operations, or slow database queries. Accumulated performance debt can result in a sluggish application that frustrates users.

Security Debt

Neglecting security considerations can lead to security debt. This includes not applying the latest security patches, not following secure coding practices, or failing to conduct thorough security audits. Security debt can expose the software to vulnerabilities and pose a significant risk.

Infrastructure Debt

Infrastructure debt pertains to the underlying infrastructure and tools used in the software project. Outdated server configurations, unsupported software versions, or suboptimal hosting choices can result in increased operational issues and maintenance overhead.

Process Debt

Process debt is related to the development process itself. It includes issues like a lack of automated testing, inconsistent code reviews, or poor communication within the team. These process-related challenges can impede productivity and quality.

Tooling Debt

The tools and development environments used in a project can accumulate technical debt if they are outdated or not effectively integrated. Outdated development tools, inefficient build pipelines, and unsupported IDEs can hinder development speed and efficiency.

Recognizing these various types of technical debt is essential for managing and mitigating their impact. While some technical debt is inevitable, it's crucial to strike a balance between short-term expediency and long-term sustainability. Regularly assessing your software project for technical debt and addressing it as part of ongoing development efforts is key to maintaining a healthy and efficient software ecosystem. By managing technical debt proactively, you can ensure your software remains adaptable, maintainable, and secure in the long run.

Job Descriptions & KPIs

TPM

JD - TPM

Following is the detailed job description for the above designation.

convert this into a TPM JD.

Agile practice

- Proficient in Scrum methodology.
- Has Good knowledge in other Agile frameworks (Crystal, Kanban, XP, SAFe, Agile Games etc.) and Lean will be added advantage.
- Has good knowledge in Software development techniques such as ATDD, TDD, BDD, CI/CD, Continuous Testing, Pairing, Automated Testing.
- Understanding of software development life cycle models (SDLCs) as well as in-depth knowledge of traditional project management principles (e.g.: - waterfall) and practices.
- Ensure process compliance across all stages of Solution Delivery evident through reports.
- Excellent Knowledge in user stories, Definition of done, backlog refinement, story splitting, estimation, velocity, retrospection, and other Scrum techniques.
- Project facilitator who guides cross-functional teams in executing the principles of the Scrum framework.
- Conducting of all Scrum ceremonies including, effective planning sessions, daily stand-ups, and retrospectives and promoting team ownership.
- Assessing the Scrum Maturity of the team and organization and coaching the team to higher levels of maturity, at a pace that is sustainable and comfortable for the team and organization
- Guiding the team on self-organizing to fill in the intentional gaps left in the Agile/Scrum frameworks
- Knowledge of numerous well documented patterns and techniques for filling in the intentional gaps left in the Scrum approach (example: numerous Burndown techniques, numerous Retrospective formats, handling bugs, etc.)

Project execution

- Has a solid understanding of all 9 areas of project management.
- Reviews the progress, identifies risks/issues, and works with product management and stakeholders to resolve them to achieve sprint commitments.
- Resolve conflicts/impediments and remove obstacles that occur.
- Provides insights for the Product Manager and team in the ongoing management of the product backlog.
- Provides insight into the priorities, through an understanding of technical issues, and identifying dependencies on other teams and finding ways to close gaps in execution.
- Removing impediments that prevent a team from completing the work it has committed for a given sprint.
- Coaching the new joiners to the team to bring them up to speed on the agile processes practiced by the team.
- Identifying gaps in skills in the team and working with the product management to address them effectively.
- Build and maintain information radiators that reflect project performance data (KPIs) and analyzing it enable informed decision making

.

- Manages virtual, distributed teams remotely comprising members from different backgrounds and cultures.
- Ensures sprint commitment and completion and owns sprint delivery.

- Measure performance using appropriate project management tools and techniques.
- Report and escalate to management as needed.
- Track project performance, specifically to analyze the successful completion of short and long-term goals.
- Makes sure due dates are updated within the first day of starting the sprint and ensures it reflects the reality throughout the sprint.

Tools

- Strong understanding of agile Project Management Tools such as Atlassian JIRA, Confluence.
- Skilled in using Agile/Scrum reports to manage and improve team performance.

Leadership

- Displays a servant leadership personality and skills.
- Leadership coaching, mentoring and motivation skills.
- Balancing the right ingredients (structure, simplicity, etc.) for each project to boost the team's engagement and performance.
- Help build a productive environment where team members 'own' the product and enjoy working on it.
- Building a trusting and safe environment where problems can be raised without fear of blame, retribution, or being judged, with an emphasis of healing and problem solving.
- Facilitating getting the work done without coercion, assigning, or dictating the work.

-

Communication

- Excellent communication and mentoring skills.
- End-to-end process coordination across multiple stages – from consuming the use cases and the business value to converting them to a robust, scalable, and high-performance solution.
- Quickly ramp up on new delivery methodologies and delivery processes
- Manage and coordinate across multiple stake holders both technical and business.
- Serve as the main point of contact for internal/external stakeholders.
- Maintaining the highest levels of transparency among all stakeholders for regarding sprint status, risks, and risk mitigation.
- Facilitate internal communication and effective collaboration for the purposes of discussion, decision making, and conflict resolution.
- Persuasive communication fluency in English.
- Excellent email communication skills.
- Problem-solving and conflict-resolution ability.
- Develop comprehensive project plans to be shared with the product management.

Technology

- Sufficient understanding of technologies in use.

Other

- Perform ad-hoc tasks assigned by the product management.

- Work with product owners to handle backlogs and new requests
- Bringing in suggestions to continuously improve the productivity of the team.
- Supporting the Product Owner, especially with respect to grooming and maintaining the product backlog.
- Solid organizational and multi-tasking skills.
- Has attention to detail.

KPIs - TPM

- Agile and scrum
- Project execution
- Leadership
- Communication
- Tools
- Technology
- Professional rapport
- Mentoring & Coaching

Agile and scrum

- Conducts all scrum ceremonies adhering to given formats. Scrum Ceremonies
- Understands and applies lean principles and contributes to improve processes. Lean Agile Process Improvement
- Coaches the team on estimating work items and planning with above 80% accuracy [(completion/commitment) * 100]. Uses the sprint planning & release planning sessions for this.
- Implements recommended processes as instructed in the Engineering space.
- Provides reports to the EM / HPMO on-time as instructed in TPM reports.
- Ensures an upward trend on team velocity.
- Ensures quality in below areas and reports to EM / HPMO,
  - Quality of processes - uses sprint retrospectives effectively for this.
  - Quality of estimation - uses backlog reviews effectively for this.
- Ensures DoD criteria being met for stories moving to Done.
- Coaches the team on scrum framework.
- Coaches the team on agile thinking guided by lean principles.
- Conducts daily scrum standups.
- Promotes and ensures teamwork and team ownership on deliverables.
- Assesses the scrum maturity of the team regularly using reports Scrum reports and coaches the team achieve higher levels of maturity at a sustainable and comfortable pace for the team and organization.
- Assesses project delivery of the team regularly using reports Scrum reports and coaches the team achieve higher levels of maturity at a sustainable and comfortable pace for the team and organization.
- Guides the team on self-organizing to fill in the intentional gaps left in the Scrum framework.

Project execution

- Has a solid understanding of all 9 areas of project management. ----
- Risk management - Coaches the team to identify risks during scrum ceremonies and works with stakeholders on contingencies and solutions to achieve sprint & release commitments.
- Resolve various types of conflicts/impediments that could arise in a project.
  - Scope conflicts
  - Communication conflicts
  - Time conflicts
  - Cost conflicts
  - Quality conflicts
  - Resource conflicts
  - Risk conflicts

- Procurement conflicts

- Keeps the EM / HPMO in the loop communications with the team and maintains transparency.
- Owns the “Ready to Groom” backlog.
- Owns the release plan. Allocation & release management
- Owns resource allocations. Allocation & release management / (eg:- Allocation plan for transpomate.xlsx (sharepoint.com)
- Owns onboarding of new team members and team members who take up new roles. Onboarding checklists
- Provides insight into the priorities, through an understanding of technical issues, and identifying dependencies on other teams and finding ways to close gaps in execution. Effectively uses Scrum Ceremonies for this.
- Makes sure due dates are updated within the first day of starting the sprint and ensures it reflects the reality throughout the sprint. (See - JIRA - How to & Default due dates per transition)
- Removing impediments/blockers that prevent your team from completing the work it has committed for a given sprint.
- Identifies new tools and techniques to manage and measure project performance.
- Acknowledges production bugs within the same working day or as soon as the next working day starts. Defect handling

Leadership

- Displays a servant leadership personality and skills.
- Leadership coaching, mentoring and motivation skills.
- Balancing the right ingredients (structure, simplicity, etc.) for each project to boost the team's engagement and performance.
- Help build a productive environment where team members own the product and enjoy working on it.
- Building a trusting and safe environment where problems can be raised without fear of blame, retribution, or being judged, with an emphasis of healing and problem solving.

Communication

- Persuasive communication fluency in English.
- Identify team members needing mentoring and mentor them. Mentor team members when needed or highlighted by HPMO / EM.
- Provides end-to-end coordination across all stages of the project / story when necessary. At the same time actively reduces communication delays by implementing lean principles in communication management.
- Manage and coordinate across multiple stake holders both technical (Devs, QAs, Dev/QA Leads) and business (BAs, PO).
- Maintaining the highest levels of transparency possible among internal and external stakeholders for regarding sprint status, risks, and risk mitigation.
- Displays email communication skills.
- Actively resolves people conflicts that could arise.
- Maintains Scrum reports and any instruction given to the team in confluence.
- Keep the Ops & Leadership sync board upto-date.
- Do escalations to the HPMO / EM efficiently and on-time.
- Build and maintain information radiators that reflect project performance data (KPIs) and analyzing it enable informed decision making.
- Displays organizing skills.

Tools

- Strong understanding of agile Project Management Tools such as Atlassian JIRA, Confluence.

Technology

- Displays sufficient understanding of technologies in use for the project.

Professional rapport

- Perform diligently or do effective communication with regard to any project/team tasks assigned by the customer-side PO / HMPO / EM.
- Maintain a professional relationship with your team members.
- Respect different cultures and maintains a welcoming environment for all.

Mentoring & Coaching

- Identifying gaps in skills in the team and working with the EM to address them effectively.
- Bringing in suggestions to continuously improve the productivity of the team.

BA

BA - KPIs

Documentation

- Creates accurate user stories and develops accurate acceptance criteria that passed DoR checklist in one-go. Definition of Ready
- Writes accurate Business Requirement Documents (BRDs). Business Requirement Document (BRD)
- Include both functional and non-functional requirements in acceptance criteria.
- Uses Diagrams to convey complex ideas efficiently and effectively.
- Prepare mandatory BA documents when necessary. (Documents)

Domain expertise

- Proficient in processes, standards, entities, artifacts and related expertise to the product domain you are working on.
- Understand your product market.
- Anticipates changes in micro and macro environments.
- Understand compliance requirements relating to your product.
- Assist sales team in client facing teams with necessary information needed for sales activities.
- Be proactive - identify gaps in requirements and drive discussions with internal team and external stakeholders (PO, Customer) toward solutions.
- Go beyond what your PO / Customer says identify/suggest new ways of solving problems.
- Uncover areas of improvement and solutions.

Communication

- Persuasive communication fluency in English.
- Conducts effective and efficient in-person communications with,
  - Internal stakeholder - (Devs, QAs, Architect, EM, TPM, HPMO)
  - External stakeholders - (PO, SMEs)
- Takes meeting notes, converts them into action items and shares them with stakeholders on-time.
- Clarify requirements to team members on-time and effectively when necessary.
- Focus on building and maintaining strong relationships with customers and POs.
- Uses requirement elicitation techniques (Requirement elicitation) effectively.
- Sharing updates in daily standups.

Agile and scrum

- Attends all scrum ceremonies adhering to given formats. Scrum Ceremonies

- Display understanding in the scrum framework and agile / lean thinking.

Tools

- Maintains the BA & Lead board (BA & Lead board).
- Uses Wireframes and Diagramming tools efficiently. Diagrams
- Proficiency in designated tools related to BA activities. BA - Tools

Professional rapport

- Perform diligently or do effective communication with regard to any project/team tasks assigned by the customer-side PO / HMPO / EM.
- Maintain a professional relationship with your team members.
- Respect different cultures and maintains a welcoming environment for all.

Technology

- Displays sufficient understanding of technologies in use for the project.

Training

- Participating in weekly BA sync up/trainings.
- Completing any assigned training materials
- Conducting sessions for BA sync-ups [At least 2 per quarter]

Day-to-day best practices

- In the event where you work in multiple projects, manage time in between projects efficiently while setting and maintaining expectations with all parties in a transparent manner.
- Keep the BA WIP within healthy limits (Min 5 - Max 10 stories) while maintaining at least 2 sprints worth of work Ready to Groom for the team.
- Uses lean management principles to make work more efficient. Contributes suggestions in TS - Sprint Retrospectives .

JD - BA

Following is the detailed job description for the above designation.

- Requirement gathering and documentation.
- Analytics.
- Communication
- Tools usage
- Agile and scrum
- Project management
- Teamwork.
- Time management
- Technology
- Domain expertise
- Post-release activities
- Pre-sales activities
- Attitude and culture-fit
- Other

Requirement gathering and documentation.

- Proficient in interpreting complex information.
- Maintains high-quality product backlogs against stakeholder expectations.
- Identifies functional/non-functional requirements.
- Gathers critical information from meetings /discussions with internal & external stakeholders in the form of **meeting notes** and **follows up** on action items addressing them systematically.
- Ensures meetings are productive and efficient.
- Conducts special stakeholder / SME interviews.
- Creates accurate / high-quality user stories.
- Create and maintain comprehensive project documentation.

Analytics.

- Identify suitable service delivery models/process models.
- Uncover areas of improvement and solutions.
- Analyze complex business requirements and create user stories.
- Stakeholder analysis.
- Costs & Benefits analysis.
- Process modeling.
- Conducting market and competitor analysis
- Plan, consult and perform BPR (Business Process Reengineering).
- Variance analysis.
- Problem solving.
- Strategic thinking.

- Attention to detail and proactive.
- Backward compatibility of product changes.
- Analysis frameworks
  - PESTLE
  - SWOT
- Evaluates ROI to the organization for items in product backlog.

Communication

- Interacts effectively with product owner / product-team / other internal-external stakeholders.
- Provides monitoring reports to management on-time.
- Liaises with vendors (Third-party companies).
- Conduct Product status meetings with the product management.
- Negotiate specifications and priorities with external stakeholders and product management.
- Possesses team problem solving skills.
- Possesses key consulting skills.
- Explains complex issues in understandable ways both in written and verbal forms.
- Facilitates internal/external meetings.
- Conducts user trainings.
- Focus on building and maintaining strong relationships with project stakeholders.

Tools usage

- Proficient in JIRA.
- Proficient in spreadsheets and excel based analytics and visualization.
- Proficient in MS Word and Confluence.
- Possesses basic Power BI skills.
- Possesses basic SQL and SharePoint skills.
- Proficient in using UI mockups (wireframes) to visually communicate information.
- Uses diagrams for analysis and communications effectively.

o Flow charts.

o DFDs (Data flow diagram)

o PFDs (Process flow diagram)

o Use case diagram

o Entity relationship diagram.

o Activity diagram.

o Business Domain Models.

Agile and scrum

- Proficient in Scrum / Agile methodologies.

- Participates in sprint planning and backlog review meetings.
- Hosts sprint planning, backlog review and sprint review meetings in the absence of scrum master.

Project management

- Makes decisions within the provided parameters.
- Takes an active part in sprint planning.
- Create and own roadmaps.
- Manages change in close collaboration with PM to manage the triple constraints (scope, time, cost) and monitor and report changes in all 9 product management areas to the management.
- Anticipates changes in micro and macro environments.
- Finish the project on-time / budget as per requirements.
- Conducts Budgeting and forecasting.
- Conducts product pricing.
- Delegate project tasks based on junior staff members' individual strengths, skill sets, and experience levels.
- Overseeing the aspects of the development life cycle including budget, quality, risks, hand over to operations, procurement, vendor relations, etc. to ensure a smooth flow.

Teamwork.

- Supports the team on sprint completion by clarifying requirements on-time effectively.

Time management

- Manages time efficiently to meet tight-deadlines.
- Communicates feature roadmap timeline changes in advance (1 sprint ahead).
- Works in multiple projects and manages time in-between.

-

Technology

- Possesses basic understanding of technologies used in the projects.
- Possesses a good understanding on software development lifecycle (SDLC)

Domain expertise

- Excellent Business domain knowledge.
- Process compliance related work. (ISO 9001 (Quality)/ 27001 (Security) / 27701(Privacy)/HIPAA)

Post-release activities

- Conducts UAT and records them in JIRA.
- Ensures proper adoption post-release.
- Keeps track of KPIs and perform course corrections.

Pre-sales activities

- Prepare pre-sales documents.
    - SOW
    - Contracts
    - Proposals
    - PoCs
    - RFP
- Assist sales team in client facing teams with necessary information needed for sales activities.

Attitude and culture-fit

- Client focused attitude.
- Work independently.
- Understand company and product vision and reflects in work.

Other

- Handles ad-hoc tasks assigned by the product management.

Functional Lead

JD - Tech Lead

Job title – Associate Tech Lead

Following is the detailed job description for the above designation.

Technology

Leadership

Communication

Dev Lead - KPI

41. Adoption and Implementation of DevOps Practices

KPI – Dev

Project execution

Code reviews

1. Fully addresses Code Review Comments. Code review
2. Maintains code-review turnaround time within healthy limits (30 mins - 1 working day). Default due dates per transition

Development

2. Adheres to Development best practices. Development Best Practices
3. Writes high-quality code. Program quality metrics - Checklist Code Quality Guideline
4. Commits code to the repository at least once per day for development stories or at least at a reasonable frequency.
5. Writes your own code following coding standards and clean code principles.
6. Refactors codes other developers (when required) has written adhering to Coding standards while applying clean Code Principles.
7. Develops high-performing software.
8. Complies to Security Standards. Security Standards
9. Comments code sufficiently. Code comments
10. Utilizes state of the art Design Patterns and Architectural Principles. Design Patterns and Architectural Principles
11. Successfully Implements all applicable areas of UX Specifications. selected for the task UX Checklist
12. Understands database architecture of the project you are working on in-depth by the time you transition a relevant task to the QA.
13. Efficiency in Identifying and Fixing Regression Issues.
14. Innovation and Creativity in Problem-Solving in Software Development.

Delivery

2. Adheres to the sprint commitment.
3. Spends a sufficient percentage of your working hours producing deliverable high-quality increments to the product.
4. Produce sufficient amount (measured in Story points) of deliverable work within a sprint with a maximum of 10% spilling over to the next sprint (for valid reasons).
5. Updates reasonable due dates and delivers tasks to QAs on-time as agreed. JIRA - How to
6. Delivers high-quality releases to QAs frequently. (set a frequency here.?)
7. Engages and contributes to developing realistic sprint plans. Sprint planning
8. Acceptable cycle time for Feature Development.
9. Raises Bug Triages for bugs that has a chance of impacting the sprint delivery within 24 hours of the development cycle.

Version controlling

1. Accurately resolving and Merge Conflicts in GitHub. Code Review & Merge Process
2. Follows the git branching strategy (Source Code Branching Strategy) and version controlling strategy (Versioning Strategy) effectively.

Test case driven development

2. Reviews, updates and removes test cases written against a story by other team members within the allocated time box your cycle time. Test case driven development
3. Adds test cases against a story within the allocated time box of your cycle. Test case driven development
4. Fully covers the acceptance criteria and test cases relevant to stories and bugs.
5. Collaboration with QA for Test Case Refinement
6. Sufficiently tests the software before releasing to QAs. Maintains a high-level of unit and integration test coverage.

Agile, Scrum & Lean

6. Contributes to sprint retrospections in the forms of feedback and suggestions in ways that helps reduce wastages. Sprint Retrospectives 7 Types of Lean Waste in software
7. Contributes to identifying dependencies during backlog reviews and clears off dependencies assigned to self during backlog review session within the first day of the sprint. Backlog Review Dependency management Sprint Schedule

Collaboration & Communication:

29. Collaboration with Cross-functional Teams.
30. Ability to Handle and Resolve Team Conflicts
31. Proactiveness in Cross-Functional Collaboration
32. Effectiveness in Communicating Technical Concepts to Non-Technical Stakeholders
33. Timeliness and Clarity in Status Updates.
34. Contribution to Team Knowledge Sharing Sessions
35. Active Participation in Code Pairing Sessions
36. Responsiveness to Team Queries and Requests
37. Recognition and Acknowledgment from Peers and Team Leads
38. Level of Contribution in Team Innovation Initiatives
39. Shares knowledge on technologies and updates with fellow team members on a regular basis and on KT sessions. Shares research & development outcomes in Research & Development section.
40. Contributes to product architectural discussions.
41. Does effective communication with regard to bug fixes reported from the QA end.
42. Demonstrates Understanding of Project ~~Domain during backlog review~~ calls, other sessions and shares them.
43. Proactively identifies Technical Debts (10 Types of Technical debts) and shares in all-team meetings. Meetings
44. Mentorship and Coaching Contribution to Junior Developers.

Continuous Learning & Self-Improvement:

81. Participation in Continuing Learning Programs/Courses related to projects.

82. Incorporation of Team Feedback into Development Process.

83. Learns and adopts new Technologies and Frameworks within a reasonable time (maximum is a per individual effort of an average sprint).

QA - KPI & JD

Associate QA Engineer, QA Engineer, Senior QA Engineer, Associate QA Lead, QA Lead, QA Manager

Automation QA Engineer. Senior Automation QA Engineer, Automation Lead, QA Architect

Have a metric over the capacity of stories that can be completed by a QA in a sprint.

QA - KPIs

Project execution

Test case creation

- Writes accurate test cases using AIO reflecting the acceptance criteria while fully covering relevant QA checklists focusing on critical items related to the particular feature and project.
- Analyzes technical documents (eg:- API spec) when required for test case creation.
- Write test cases covering areas highlighted in UX Checklist in all the projects.
- When testing APIs, ensure that you create test cases for scenarios in the API Testing Checklist .
- When testing UIs, ensure that you create functional and non-functional test cases for all scenarios in the UI Checklist. . NFRs
  - When testing Web broweser UIs, ensure that you create test cases for areas mentioned in the Web Responsive checklist.
  - When testing Mobile app UIs, ensure that you create test cases for areas mentioned in the Mobile Responsive checklist and UI - Checklist.
  - When testing Mobile app, ensure that you create test cases for functional aspects of the app. Functional Checklist
  - When testing a Mobile app, ensure that you create test cases for general aspects of the app. General Checklists.

Test case execution

- Executes test cycles as soon as the task moves to QA in-progress and executes all test cases within a working day and closes the cycle before moving the status. Test cycles & Case creation & execution
- Executes test cases following the process and expectation. Test cycles & Case creation & execution
- Analyzes data, databases and logs when required for test case execution.
- Uses production-like test data for test case execution.

General

- In the event where you work in multiple projects, manage time in between projects efficiently while setting and maintaining expectations with all parties in a transparent manner.
- Keep the QA In-progress and QA-todo within healthy limits.
- Uses lean management principles to make work more efficient. Contributes suggestions in TS - Sprint Retrospectives
- Handles UAT & Production defects following the right process. Defect handling

Domain expertise

- Proficient in processes, standards, entities, artifacts and related expertise to the product domain you are working on and reflect them in your test-cases.
-

Understand compliance requirements relating to your product.

- Understand NFRs specific to your product.
- Uncover areas of improvement and solutions in the product. Provide these insights during Meetings.
- Display full understanding of the BRD and provides BAs on the acceptance criteria when there are misalignments during the ready to groom stage and other phases of the story.

Agile and scrum

- Attends all scrum ceremonies adhering to given formats. Scrum Ceremonies
- Display understanding in the scrum framework and agile / lean thinking.

Documentation

- Submit below reports on-time. Deliverable Formats
  - Sprint Summary Report/ Internal Release note
  - Client Release Note
  - Postmortem update for UAT/Production bugs.

Communication

- Effective communication fluency in English.
- Participates and clarifies different test case scenarios from fellow team members during scrum ceremonies and other meetings. Mee tings
- Clarify test cases team members on-time and effectively when necessary.
- Conduct product demos during sprint reviews.
- Be proactive in communications - identify gaps in requirements and drive discussions with the internal team toward solutions.
- Conducts effective and efficient in-person communications with Internal stakeholders - (Devs, QAs, Architect, EM, TPM, HPMO)
- Takes meeting notes when you are a designated passive member of a discussion. Meeting note-taking
- Sharing updates in daily standups.

Training & awareness

- Participating in weekly QA related trainings.
- Stays upto date on documents related to QAs and general.
  - QA - Labelled content - Company Engineering - Confluence (atlassian.net)
  - General - Labelled content - Company Engineering - Confluence

(atlassian.net) • Completing any assigned training materials

- Conduct trainings and awareness sessions for fellow QA team members when allocated.

Professional rapport

- Perform diligently or do effective communication with regard to any project/team tasks assigned by the customer-side PO / HMPO / EM.
- Maintain a professional relationship with your team members.
- Respect different cultures and maintains a welcoming environment for all.

Tools

- Maintains QA-todo, QA In-progress columns of the Scrum Board.
- Uses AIO effectively and efficiently. Diagrams
- Proficiency in designated tools related to QA activities. QA - Tools

- Uses Github PR comments before executing test cycles (from the second cycle onwards) Code review comment structure

Technology

- Displays sufficient understanding of technologies in use for the project.

Test Automation

- Technical skills
- Soft skills
- Tools usage
- Agile and Scrum
- Project Management
- Teamwork

Domain knowledge

- Possess comprehensive knowledge on all RCM processes.

Technical skills

- Review and analyze the User Stories and requirements specifications and convert them into comprehensive and well-structured test cases without supervision.
- Identify all test scenarios on the first test round itself.
- Carry out sole manual testing if necessary and only if instructed to do so by supervisor/PM.
- Execute test cases, review, and report back to the team within the day.
- Report both functional/non-functional issues.
- Update your supervisor daily on progress.
- Participate Daily Scrums and Provide feedback and Suggestions when required.
- Should be able to plan and prioritize tasks based on sprint priorities.
- Setup and perform automated regression testing on the staging environment within set intervals on a regular basis.
- Identify the Automation and Non-Automation Test Cases.
- Develop test automation plans.
- Capable of building a test automation framework from the scratch and deploying it in dev/test/staging and production environments.

Soft skills

- Demonstrate analytical skills.
- Demonstrate Problem-Solving skills.
- Motivated to self-learn and upskill at a quick pace and stays upto date with the state-of-the-art.
- Flexibility and adaptability in carrying out duties.
- Maintains a smooth and professional relationship with team members and management.

Tools usage

- Proficient in Java and Selenium.
- Proficient in TestNg and building dashboards out of it.
- Generate custom test automation reports.
- Working knowledge on Jenkins and CI/CD.
- GIT (Version Controlling)
- Proficient in querying databases with MySQL within the test automation framework and workbench.

- Skilled in postman-based API testing.
- Proficiency in load testing with JMeter.
- Good understanding in HTML/XML and Json.
- Skilled in working in JIRA.

Agile and Scrum

- Proficient in Scrum / Agile methodologies applies Agile/scrum methodologies on daily basis.
- Actively takes part in story point estimation.
- Participates in sprint planning and backlog review meetings and provides at least one input per session.

Project Management

- Proficient in creating test planning and monitoring.
- Understands the role and importance of Quality and Automation within the overall project.

Teamwork

- Support the team after the release in identifying bugs and issues on production.

JD - TA

Domain knowledge

- Possess comprehensive knowledge on all processes.

Technical skills

- Review and analyze the User Stories and requirements specifications and convert them into comprehensive and well-structured test cases without supervision.
- Identify all test scenarios on the first test round itself.
- Carry out sole manual testing if necessary and only if instructed to do so by supervisor/PM.
- Execute test cases, review, and report back to the team within the day.
- Report both functional/non-functional issues.
- Update your supervisor daily on progress.
- Participate Daily Scrums and Provide feedback and Suggestions when required.
- Should be able to plan and prioritize tasks based on sprint priorities.
- Setup and perform automated regression testing on the staging environment within set intervals on a regular basis.
- Identify the Automation and Non-Automation Test Cases.
- Develop test automation plans.
- Capable of building a test automation framework from the scratch and deploying it in dev/test/staging and production environments.

Soft skills

- Demonstrate analytical skills.
- Demonstrate Problem-Solving skills.
- Motivated to self-learn and upskill at a quick pace and stays upto date with the state-of-the-art.
- Flexibility and adaptability in carrying out duties.
- Maintains a smooth and professional relationship with team members and management.

Tools usage

- Proficient in Java and Selenium.
- Proficient in TestNg and building dashboards out of it.
- Generate custom test automation reports.
- Working knowledge on Jenkins and CI/CD.
- GIT (Version Controlling)

- Proficient in querying databases with MySQL within the test automation framework and workbench.
- Skilled in postman-based API testing.
- Proficiency in load testing with JMeter.
- Good understanding in HTML/XML and Json.
- Skilled in working in JIRA.

Agile and Scrum

- Proficient in Scrum / Agile methodologies applies Agile/scrum methodologies on daily basis.
- Actively takes part in story point estimation.
- Participates in sprint planning and backlog review meetings and provides at least one input per session.

Project Management

- Proficient in creating test planning and monitoring.
- Understands the role and importance of Quality and Automation within the overall project.

Teamwork

- Support the team after the release in identifying bugs and issues on production.

TA - KPIs

Scrum Team Description & KPI

KPI Guidelines

Characteristics of a good KPI

- Should fulfill SMART criteria.

Training & awareness

Devops Training content

https://kodekloud.com/learning-path/devops-engineer/

Business Analysis Training

Objective	Task
BA Basic Training	BA Basic Training.docx

AIO Training plan

Summary & Goal	The goal of this training program is to upskill Company employees in using AIO test tool on JIRA. The training is targeted at both QAs and Developers The training will include, • Training videos (• a physical/remote training session by the coach. • a process to evaluate participation.
Coach	Frontend Developer
Training content	React - The Complete Guide 2023 (incl. React Router & Redux) | Udemy
No of Lectures / hours	586 / 55.5 total hours
Training repo	Companysl/training-react (github.com)
Workshop included.	Yes - on Oct 23, 2023 and Oct 24, 2023
Board	Training Board - Agile board - COMPANY (atlassian.net)
Next batch start (Quarterly)	Jan 1, 2024

Steps

1	Setup your environment.	due - Oct 23, 2023	Start a New React Project – React
2	Training session.	due - Oct 23, 2023	
3	• Mark training task as started. • Follow the Udemy course. • Create a Github branch. • Commit to GitHub daily. • Mark training task as done.	mark your own start - end date of the program in the Jira task assigned to you.	• React - The Complete Guide 2023 (incl. React Router & Redux) | Udemy • Companysl/training-react (github.com)

Udemy course username and password. Make sure not to use the platform when someone else is using it. Drop a message in the training batch group before starting.

Udemy Account 1

Username:
Companyudemy@gmail.com
Password: UdemyCompany@123

Udemy Account 2

Username:
Companyudemy2@gmail.com
Password : UdemyCompany@456

Statuses of training

Summary	Created	Updated	Due	Assignee	Status

React training - Malshan	23/Oct/23 12:52 PM	27/Oct/23 6:38 AM	Malshan Gunawardana	IN PROGRESS
React training - Yashoba	23/Oct/23 12:52 PM	27/Oct/23 8:38 AM	Yashoba Wickramage	IN PROGRESS
React training - Dilini Tharaka	23/Oct/23 12:52 PM	27/Oct/23 9:11 AM	Dilinee Tharaka	IN PROGRESS
React training - Sandali	19/Oct/23 12:03 PM	27/Oct/23 9:20 AM	Sandali Yasodhara	IN PROGRESS
React training - Sakunthala	19/Oct/23 10:58 AM	27/Oct/23 9:13 AM	Sakunthala Jayasundara	IN PROGRESS

5 issues

React Training plan

Summary & Goal	The goal of this training program is to upskill Company employees in ReactJS. The training will include, • training content provided by udemy. • a physical/remote training session by the coach. • a process to evaluate participation.
Coach	Frontend Developer
Training content	React - The Complete Guide 2023 (incl. React Router & Redux) | Udemy
No of Lectures / hours	586 / 55.5 total hours
Training repo	Companysl/training-react (github.com)
Workshop included.	Yes - on Oct 23, 2023 and Oct 24, 2023
Board	Training Board - Agile board - COMPANY (atlassian.net)
Next batch start (Quarterly)	Jan 1, 2024

Steps

1	Setup your environment.	due - Oct 23, 2023	Start a New React Project – React
2	Training session.	due - Oct 23, 2023	
3	Provide feedback		Training program feedback (office.com)
4	• Mark training task as started. • Follow the Udemy course. • Create a Github branch. • Commit to GitHub daily. • Mark training task as done.	mark your own start - end date of the program in the Jira task assigned to you.	• React - The Complete Guide 2023 (incl. React Router & Redux) | Udemy • Companysl/training-react (github.com)

Udemy course username and password. Make sure not to use the platform when someone else is using it. Drop a message in the training batch group before starting.

Udemy Account 1

Username:
Companyudemy@gmail.com
Password: UdemyCompany@123

Udemy Account 2

Username:
Companyudemy2@gmail.com
Password : UdemyCompany@456

Statuses of training

Summary	Created	Updated	Due	Assignee	Status

React training - Malshan	23/Oct/23 12:52 PM	27/Oct/23 6:38 AM	Malshan Gunawardana	IN PROGRESS
React training - Yashoba	23/Oct/23 12:52 PM	27/Oct/23 8:38 AM	Yashoba Wickramage	IN PROGRESS
React training - Dilini Tharaka	23/Oct/23 12:52 PM	27/Oct/23 9:11 AM	Dilinee Tharaka	IN PROGRESS
React training - Sandali	19/Oct/23 12:03 PM	27/Oct/23 9:20 AM	Sandali Yasodhara	IN PROGRESS
React training - Sakunthala	19/Oct/23 10:58 AM	27/Oct/23 9:13 AM	Sakunthala Jayasundara	IN PROGRESS

5 issues

SQL Training plan

Scrum Master training

Architecture training plan

Awareness sessions

- Story points.
- Agile

Project Leadership training

Sprint Schedule

Meeting time	**1 day + 4.5hrs / 12.5hrs (15.6%)**	(15 mins * 10) + 30 mins + 1hr + 4hrs + 2hrs + 30mins + 2hrs
Focus time	**8 days 3.5hrs** / **67.5 hrs (84.4%)**	

Day 1	• Clear out all dependency review action items. DEV TEAM QA TEAM • Schedule all the ceremonial calls TPM	
Day 2	• DEPENDENCY CLOSURE SESSION (15 - 30 mins) in the morning. - Dependancies DEV TEAM QA TEAM • DEPENDENCY CLOSURE SIGNOFF TPM	
Day 3	• RELEASE PLANNING & REVIEW (1 hr) PO TPM DEV TEAM QA TEAM	
Day 4		
Day 5	• BA and lead signs off new stories added through the previous sprint and informs the team. NEW BA LEAD	
Day 6		
Day 7	• BACKLOG REVIEW SESSION - 4 hrs. - Backlog Review (Re-schedulable between 6th - 8th day) PO	
Day 8		
Day 9	• DEPENDENCY REVIEW - SIGNOFF • SELF-REVIEW DEV TEAM QA TEAM	
Day 10	• SPRINT PLANNING SESSION - 2 hrs (in the morning) • PROJECT ALLOCATION PLANNING - 1 hr (in the morning) • SPRINT RETROSPECTIVE - 30 mins (in the morning) • SPRINT REVIEW SESSION - 2 hrs. (after lunch) - Sprint Review (sample from Transpomate) • SPRINT PLAN - SIGNOFF	

Checklist - Sprint 0

- KTs and Training TPM
- Jira TPM
- Communication channels TPM
- Test Automation OPTIONAL QA TEAM
- GitHub DEV LEAD
- Deployment DEVOPS
- Systems monitoring OPTIONAL DEVOPS
- Project scope

KTs and Training TPM

- Make sure the team has enough knowledge around the scrum and agile process and the specific configuration of it we are following. A few key areas would be (but not limited to), TPM
  - Workflows - Board workflows
  - Scrum ceremonies - Scrum Ceremonies
  - Best practices. - Best practices NEGOTIABLE
  - Story points - Story points
  - Definitions - Definitions
  - Sprint schedule - Sprint Schedule
- If a particular member of the team is new to their role, make sure they go through the onboarding checklist for their role. Onboarding checklists NEGOTIABLE . TPM has to have 1-1s with each team member if necessary and assign this.

Jira TPM

- Create a company managed Jira project sharing the Template project settings.
  - Create Boards by copying and editing the template project boards (both Kanban and Scrum).
  - Import all the automations into the project and edit specific details such as individual assignees in automations.
  - If Technologies you are using in your project aren't available in the "Technologies" dropdown field, escalate it to EM.
  - Ensure your project adheres to the configurations mentioned here - JIRA Configurations

Communication channels TPM

Setup following channels for these users → TPM, Lead Dev, Devs, QAs, EM, HPMO

- Internal Teams group.
  - Name format - [Internal] - (Project name as in JIRA)
- Email group. (Via IT). Invite this group to all scrum ceremonial calls.
  - Name format - Team(Project name as in JIRA)example.com

Test Automation OPTIONAL QA

Every commit/merge to the QA branch triggers an automated tests case run.

1. Configure automated code quality checks.
2. Configure automated test-case based quality checks.
3. QA CICD should fail if any of the automated test cases fail.

4. Automation should test against basic NFRs.

GitHub DEV LEAD

- Create all project repositories.

Deployment DEVOPS

- Assign a Devops Team member. Facilitate training if required. Devops Training content
- Write scripts and configure CICD to all the repositories.
- Select a deployment strategy.
  - Recreate: Version A is terminated then version B is rolled out.
  - Ramped (also known as rolling-update or incremental): Version B is slowly rolled out and replacing version A.
  - Blue/Green: Version B is released alongside version A, then the traffic is switched to version B.
  - Canary: Version B is released to a subset of users, then proceed to a full rollout.
  - A/B testing: Version B is released to a subset of users under specific condition.
  - Shadow: Version B receives real-world traffic alongside version A and doesn't impact the response.
- Configure deployment rollback in either of the above scenarios. OPTIONAL
- Authorize a lead to do 1 click deployment to the staging and production environments. Provide KTs if required. OPTIONAL
- Authorize all developers to do 1 click deployment to QA. Provide KTs if required.
- Write IaC (Infrastructure as Code) scripts to setup QA / Dev / Staging environments. OPTIONAL

Systems monitoring OPTIONAL DEVOPS

Configure centralized monitoring system(s)

- To monitor applications.
- To monitor resources.
- To monitor environments.
- To monitor costs.
- Setup alerts based on set parameters.
- Setup alerts to notify the team on production issues.
- Logging system for developers across environments with ability to filter and obtain logs within few clicks and key strokes.

Project scope

- Do T-sizing for all stories. DEV LEAD
- All stories in Jira and assigned to epic. BA
- All epics created in JIRA. BA

Sprint Day 1 checklist

Current Sprint:

Responsible person	Work item
SM • SM •	Schedule Backlog Review session. Schedule Sprint Review session. Schedule Sprint Retrospective session. Schedule Sprint Planning session. Schedule Dependency Closure Session
DEV TEAM QA TEAM	Close pending dependencies.
EM • EM	Schedule Dependency Review Summary discussion with SM. Schedule Sprint Plan Review.

Release Closure Sprint

- Make sure the there is qa/prod parity.

Support processes

Bug tracing

- It should be possible to trace an issue when it appears in the production environment to where it originally had to be addressed.

Issue reporting

Use JSM

Standard form for PIS

- Every process improvement form has to be named in a way when you read the title itself you should be able to get some clarity on the changes proposed in that form.
- Following is a standard form all Process Improvement Suggestion forms should include.

Suggested by	your name
Peer-reviewed by	get your suggestion reviewed by at least 1 person as a prerequisite for approval.
Reviewed and approved by	This should be one of the following, • CTO ( ) • COO () • Engineering Manager (EM)

Problem	Explain the problem as clearly as possible.
Suggested solution	Explain your solution as clearly as possible.
Justifications (optional)	This is completely optional. You can add more justifications into
Does this problem impact WIP.?	YES / NO
If answer to the above question is yes, which project it is.?	

Decisions

Problem : QAs should know the impacted areas of the development when they recieve a bug-fixed version of. Such as database changes, API changes, etc. These are mostly technical stuff.

Solution:

- Add this information as a comment to the story. The comment must have a defined structure covering all the information requirements (action item for EM).

- Additionally , the component field should be used to identify buggy components of the project in reports. Have to include this part into the process and take it up in the next iteration of QA / Dev / BA sync this week (action item for EM)

- **Problem** : How to report and keep track of pre-UAT bugs (bugs in QA env).?
- Solution : Report it exclusively with AIO. (action item for EM - Need to train the teams on AIO)

- **Problem** : How to report and keep track of UAT bugs (bugs in staging)
- Solution : Create a bug ticket in JIRA as the story is marked as done at this point. For UAT CRs, follow the standard story process.
- **Problem** : How to report and keep track of Production bugs, hot fixes.?
- Solution: Report this in a different board. This part of the process is yet to be defined (action item for EM)

Problem: It should be possible to trace an issue when it appears in the production environment to where it originally had to be addressed. (We need to define this part - action item for EM). A tentative solution should be something like below.

- Anything reported in the Production starts with the Bug reporting board. Attach the ticket in JSM to the respective epic and you can track it down to where it has gone wrong.
- If it is in UAT or pre-UAT, the bug is attached to a story. The story must've gone through multiple dev/test cycles. We can do the tracing using AIO.

Problem: There are situations where we have to decide to move things out of sprint. But there has to be some control and monitoring over this.

Solution: Do triages. Introduce a process around this concept (action item for EM)

-
-

QA - 10/30/2023

Problem : QAs should know the impacted areas of the development when they recieve a bug-fixed version of. Such as database changes, API changes, etc. These are mostly technical stuff.

Solution :

- Add this information as a comment to the story. The comment must have a defined structure covering all the information requirements (action item for EM).

- Additionally , the component field should be used to identify buggy components of the project in reports. Have to include this part into the process and take it up in the next iteration of QA / Dev / BA sync this week (action item for EM)

- **Problem** : How to report and keep track of pre-UAT bugs (bugs in QA env).?
- Solution : Report it exclusively with AIO. (action item for EM - Need to train the teams on AIO)

- **Problem** : How to report and keep track of UAT bugs (bugs in staging)
- Solution : Create a bug ticket in JIRA as the story is marked as done at this point. For UAT CRs, follow the standard story process.
- **Problem** : How to report and keep track of Production bugs, hot fixes.?
- Solution: Report this in a different board. This part of the process is yet to be defined (action item for EM)

Problem: It should be possible to trace an issue when it appears in the production environment to where it originally had to be addressed. (We need to define this part - action item for EM). A tentative solution should be something like below.

- Anything reported in the Production starts with the Bug reporting board. Attach the ticket in JSM to the respective epic and you can track it down to where it has gone wrong.
- If it is in UAT or pre-UAT, the bug is attached to a story. The story must've gone through multiple dev/test cycles. We can do the tracing using AIO.

Problem: There are situations where we have to decide to move things out of sprint. But there has to be some control and monitoring over this.

Solution: Do triages. Introduce a process around this concept (action item for EM)

QA/BA - 11/03/2023

Participants : PM, QA Lead, EM

Problem : Need to identify on which sprint an API went to Clounote.

Solution :

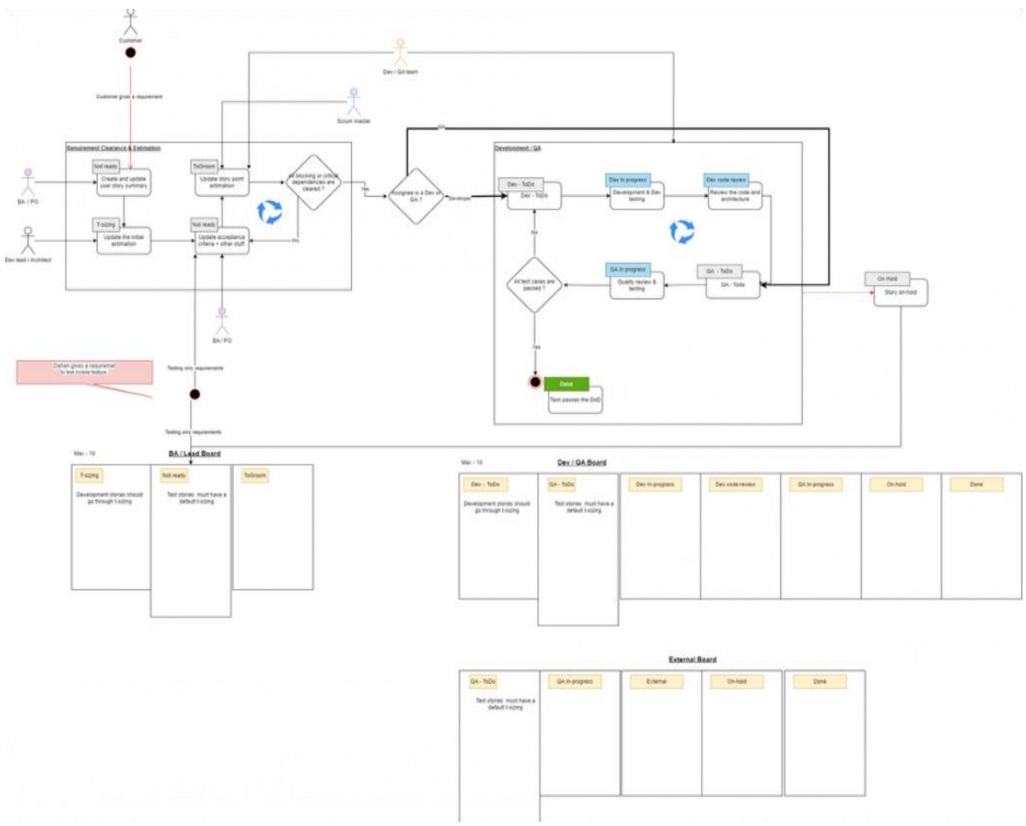
- Custom field to set the API delivered sprint

Problem : Unable to create a board to manage clounote stuff.

Solution:

Migrate the project to company managed project.

Agreed on the below workflow.



Definition of a good change

1. Should not introduce or increase 7 types of lean wastes defined here. (this is highly important). Should decrease or eliminate one or more of these wastes from the process.
   a. Refer - 7 Types of Lean Waste in software
2. If possible, should improve.
   a. Test coverage in a measurable manner.
3. The processes and artifacts should be monitorable through auto generated reports. These reports include
   a. Test coverage
   b. reports to measure post release bugs, etc.
4. The change should not break data related to reports currently in use.
5. Should not break the agile/scrum process related configurations. If does, the remedy should be included in your solution.

Process Improvement Session - Agenda & Best Practices

- Engineering manager goes through the evaluation criteria mentioned in the **DoGC** and the purpose of the session.
- Each member gets 15 minutes to present their idea. You should go through **DoGC** criteria here → Definition of a good change and justify your proposal to the forum.
  - You should explain how your suggestions help minimizing 7 types of lean waste introduced in a process. 7 Types of Lean Waste in software
  - This ideally be an uninterrupted slot because we shouldn't get to judgements without knowing the full idea well. If you have a question, you can raise the hand and the presenter gets to decide when to give time to answer your question.
  - The Engineering Manager takes notes.
- After each member of the discussion presents their proposal a collab session is held for group discussions.
- The Engineering manager goes through each suggestion and presents the evaluations against the criteria to the group. We still haven't made a decision. This part of the session is there to drive the discussion toward an end and see if everyone is onboard.
- The group takes a final decision. At this point it's a group decision and should be carried out that way. This is always and us vs our objectives discussion and decision.

7 Types of Lean Waste in software

7 types of wastes in software projects.

In lean world	In the lean agile world	+ / - / =
1. Waste of inventory	partially done work (or work in process)	
2. Waste of over production	delivering extra or unneeded features	
3. Waste of extra processing	relearning	
4. Waste of transportation	handoffs	
5. Waste of waiting	delays	
6. Waste of motion	context switching	
7. Waste of defects	defects	

Tips to avoid waste

Over production

1. validate assumptions in shorter cycles when it comes to all types of work.
2. do shorter cycles of quality assurance against development.

Inventory (WIP)

1. Sprint plan is the scrum team's WIP. Keep it within achievable limits.

Extra processing

1. Practice good code documentation.
2. Document best practices.

Handoffs

1. Encourage cross functional learning and activities. Such as dev-testing, test automation.
2. Keep your teams as self-sufficient as possible and eliminate repeating dependencies.

Waiting Delays

1. Eliminate dependencies before taking a story into a sprint.
2. Remove bottlenecks in handoffs. (eg:- Dev → QA handoffs, QA → Dev handoffs)

Motion (context switching)

1. Prioritize work really well in backlog reviews.
2. Avoid taking unplanned work in sprints.

Defects

1. Work on quality up-front - Meaning, plan and write test cases in advance.
2. Keep Dev / QA cycles as short as possible by removing bottlenecks.

Samples

Sample Project Decisions

Moved all Automation epic tasks to not-selected status.

Sprint 13

Date - 9/12/2023

Decision made by: EM

Decision: Removed 16.5 story points worth of stories from the sprint.

Reason: overcommitment during planning.

CC: Developer , SM

Date - 9/14/2023

Decision made by:

Decision: DevOps and deployment architecture implementation and Reports functionalities has to be a part of phase 1.

CC: COO

Sprint 14

Date - 9/26/2023

Decision made by: EM

Decision:

→ Before starting a sprint, frontend developer, backend developer and tester should be assigned to each story.

→ Decided to clear out all the spilled over stories and setup sprint goals for that.

Date - 10/05/2023

Decision made by: EM

Decision:

→ There should be a way to review the completed work against the sprint goals during the middle of the sprint. (Preferably send an email to the EM)

Sprint plan sample (From Transpomate)

Sprint goals:

- [ ] Complete spilled over mobile + web stories (SC - 4 / SP - 13)- Developer , Developer ,
- [ ] Achieve 46 SPs - (13 SP (4) spillovers + 36SP (8) new items)
- [ ] Complete R&D task on SMS service comparison and provide a summary of the findings - Developer2 (Story points - 3)

Sprint summary:

total spillover stories points / count	13 SP / 4 - Developer , Developer2
total new stories points/count	33 SP / 8 - Developer , Developer2
current velocity	20.2 SP
commitment	46 SP (commitment is 33 + spillovers)
dependency status	• TN-602 ◦ Acceptance criteria needs to be updated. BA PENDING ◦ UI design - BA PENDING ▪ `Error page` should be designed to show if the login feature is crashed. ▪ `Not access page` should be designed to show if the user has the access to login or not. ◦ Sequence diagram - Developer TN-592 • ◦ R&D on how to create a secure URL to upload files - Developer PENDING ◦ R&D on how to clear the files stored in the server side. Developer PENDING More information → Issue Dependencies - Sprint 15

Spill overs:

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution
No issues found										

New tasks:

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Lab
TN-804	[UI[Status] - Toggle alignment should be center.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-793	[Trip History] - The added trip request is not visible on the trip history page.	QA	QA					DONE			
TN-792	[User][Status] - The user can login to the system successfully after being disabled.	QA	QA					DONE			
TN-791	R&D on SMS services	Thambara Supun Basnayake	BA	Thambara Supun Basnayake				DONE			
TN-787	[UI] - The ID column should be removed.	QA2	QA					DONE			
TN-782	[Add Vehicle] - Validation messages should display in particular fields when saving with invalid data.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-781	[Edit Vehicle] - Vehicles should not be saved with zeros in the "no seat" and "cost per km" fields.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-780	[UI][All Modules] - Validation messages are not in the correct capitalization.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-777	[UI][Admin][All Modules]-The gap between all the fields of add employees are not same as figma	QA2	QA2	Sasani Jayathilake		QA2		DONE			
TN-773	[Suggestion][UI][All Modules] - Lack of separation between title and fields.	QA	QA					DONE			
TN-692	[UI][Vehicle] - The name of the 7th column is not as figma.	QA2	QA					DONE			
TN-674	[All Modules]- The latest data should be visible at the top of the list.	QA	QA					DONE			

TN-673	As an Admin, I need to access only the functions assigned to me, So that I can do the specific assigned functions within the system.	QA	BA	dileepa.mabulage	dileepa.mabulage	QA	8.0	DONE		
TN-671	[UI][Edit Owner] - The error message is not in correct capitalization.	QA	QA					DONE		
TN-669	[UI][View Vehicle]- The column names are not in the correct capitalization.	QA	QA					DONE		
TN-666	[Email] - The email does not contain the accurate timezone	QA	QA					DONE		
TN-627	As a Manager, I need to receive an email, So that I am notified that an employee has requested a trip.	QA	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	3.0	DONE		
TN-612	[CR] As a Transport Coordinator, I need to view all scheduled voyages, So that I can manage and coordinate scheduled voyages effectively	QA	BA	Sasani Jayathilake		QA	1.0	DONE		
TN-602	As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app	Rishimithun Muralidharan	Thanusan irran	Rishimithun Muralidharan	dileepa.mabulage	QA2	8.0	DEVELOPMENT IN PROGRESS	30/Nov/23	
TN-592	As a mobile tester, I need a backend endpoint to return a unique URL to upload files, So that the files can be uploaded securely.	QA2	BA		dileepa.mabulage	QA2	5.0	DONE		
TN-574	[LocalApk][Mobile] [Passenger List]-Check box functions independent of the passenger cards	QA2	QA2					DONE		LocalA
TN-427	As an admin, I want to view the users registered to the	QA	Thanusan irran	Sasani Jayathilake	Thambara Supun Basnayake	QA	3.0	DONE		

	system, So that I can view user details.							
TN-225	As an admin, I want to enable and disable user, So that they are inactive in the system once disabled and active once enabled	QA	Thanusan irran	Abdul Qadir	QA	2.0	DONE	

23 issues

Sample Issue Dependency Summary Document (From Transpomate)

Overall status	URGENT
Total Impacted story count/points.	~~9 / 44.5~~
Blocked story count / points	~~1 / 18~~
Current velocity in story points	16.5
Dependency clearance due date	Sep 22, 2023 CLEARED
Dependency Score	to-be-decided

Issues Impacted

Key	Summary	Assignee	Reporter	Status	Frontend Dev.	Backend Dev.	Tester	Story Points
TN-598	As an admin, I need to search for users, So that I can assign the searched user with a role (assigning the role isn't included)	QA	BA	DONE	Sasani Jayathilake	dileepa.mabulage	QA	5.0
TN-591	As a Transport Coordinator, I need to receive an automated email, So that I know that the voyage has started	QA	BA	DONE	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	13.0
TN-561	As a Transport Coordinator, I need to create new starting and ending points for a voyage, So that I can change the voyage route.	QA	BA	DONE	Abdul Qadir	Thambara Supun Basnayake	QA	5.0
TN-513	As a Driver, I need to upload the selected image, So that the TC could use the image to compare with the odometer text the driver entered.	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	5.0
TN-461	As a Driver, I need to verify the attendance of the passengers and mark the checkbox if present, So that I can see the pins on the map for the present passengers	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0
TN-346	As a Driver, I want to enter the odometer reading, so that I can start the voyage.	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan	dileepa.mabulage	QA2	0.5
TN-345	As a Driver, I want to take a picture of the odometer, so that I can keep track of the odometer beginning reading before starting the voyage.	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan		QA2	5.0
TN-209	As a Driver, I want to drop trips for employees, So that the mobile app will be notified with each drop and help in preparing the report.	QA2	Thanusan irran	DONE	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	5.0
TN-207	As a Driver, I want to view a list of the passengers, So that I know which passengers have been allocated to the voyage and mark their attendance.	QA2	Thanusan irran	DONE	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0

9 issues

Dependencies

Issue Key	Dependancy	Status
TN-598	Sequence Diagram – Developer	NON-CRITICAL
TN-346	• API keys to be shared with Rishi - Developer - DONE • Architecture diagram - Architect1 - DONE	DONE
TN-513	• User story / Acceptance Criteria / BA Diagram - BA - DONE • Blob storage for the QA environment. Setup QA environment - Developer - DONE • Architecture Diagram - Architect1 - DONE • Device for QA testing - SM - DONE • Sequence Diagram - Developer2 DONE • Deployment Diagram - Developer / Architect1 to confirm which deployment method to use. (candidate choices: GitHub actions, azure Devops)	DONE
TN-345	QA environment resource provisioning - Developer - DONE	DONE
TN-209	• Architecture Diagram - Architect1 - DONE • Sequence Diagram - Developer	NON-CRITICAL
TN-461	• Database Diagram & Changes - Developer2 - DONE • Sequence Diagram - Developer2	NON-CRITICAL
TN-207	• Sequence Diagram - Developer	NON-CRITICAL
TN-561	• Sequence Diagram - Frontend Developer	DONE
TN-591	• Create required cloud resources in Dev environment. Pending due to naming convention issue. - Architect1 - DONE • Create required cloud resources in QA environment. Pending due to naming	NON-CRITICAL

	convention issue. - Architect1 - DONE • Sequence Diagram - Developer2 DONE • Deployment Diagram - Developer - REMOVED • Happy path test cases - QA2	

Sample Sprint Review (14/09/2023) meeting notes & action items

(New story - CR linked to TN-119: As a Transport Coordinator, I want to view all voyages (scheduled, completed, ongoing, and abandoned), So that I can see and manage all voyages effectively.
DONE **119)** / Manage Voyages page **[** TN-612: [CR] As a Transport Coordinator, I need to view all scheduled voyages, So that I can manage and coordinate scheduled voyages effectively DONE **]**

- Change the distance to units KMs.
- Add starting odometer reading instead of ending odometer reading.
- below UI changes to be done in voyage information
  - address --> drop location
  - name --> full name
  - date --> remove pipe or keep a space
  - vehicle --> introduce a format to vehicle ids
- Fix UI alignment issues.

Mobile app

New story - CR / Mobile Landing page / **[** TN-613: [CR] As a Driver I need to view the landing page of the mobile app, So that I can log in and view the pending voyages DONE **]**

- Change the logo to the newly designed Transpomate logo. (contact marketing) Button
- names & UI should match Figma.

Pending Voyage - CR / [TN-619: [CR] As a Driver, I need to see the pending voyage, So that I can plan and prepare for the voyage DONE **]**

add below items as Acceptance Criteria to TN-54: As a Mobile User, I want to view all the pending voyages, so that I can stay informed about the scheduled voyages for the day DONE **)**

- Need to add a loader if there is a delay when loading the voyage on the screen.
- In the pending voyages page, change the name to "pending voyage."

Setup a UI review session and explore (Responsible: BA)

- Identify gaps between Figma and the App make changes to the plan to close them. (eg:- Passenger list - alignment issue for the call icon - Developer)

New story - CR / Voyage preview page / [TN-614: [CR] As a Driver, I need to view a preview of the voyage before starting the voyage, So that I can view the route and be prepared DEV - TODO **]**

- In the preview page, we should show the *exact location.*
- Remove voyage ID.
- Need to add the re-arrangement in the voyage preview page as well.
- should show the start and end locations in the route preview but there should not be drops (red pins) in those points.

New story - CR / start voyage confirmation. / [TN-615: [CR] As a Driver, I need to see a confirmation message when starting a voyage, So that I could either start the voyage or mark the missed passengers DONE **]**

- In the confirmation (after starting voyage) change the `Back` to a `Close`.

New story - CR / Odometer capture / [TN-616: [CR] As a Driver, I need to take a picture of the odometer reading, So that I could send it to the Transport Coordinator DEV - TODO **]**

- Merge these 2 functions → Take picture and select picture.
  - What should happen is when you take a picture, it should be shown as a preview instead of selecting.

(follow-up with PO if we should remove the existing feature - BA)

New story - CR / Odometer ending / [TN-617: [CR] As a Driver, I need to enter the odometer reading, So that I could send it to the Transport Coordinator for him to compare with the image TOGROOM **]**

- In the odometer ending page, remove the restriction of 5 decimals and need to add decimal values as well.

New story - CR / Change the status columns of the tables to an icon / [TN-618: [CR] As an Admin, I need to see clear icons on all the view pages, So that I can quickly know the status DONE **]**

- In the view pages, we need to show an icon/ toggle switch rather than having a label saying `Enable` or `Disable`, to show the status of the records so that it's easier for the users (admin / manager / TC) to see the status.

Sample Sprint Retrospective - Sprint 15 (Day 10)

A sprint retrospective is a type of meeting within the Agile framework, when teams reflect on what went well and what could be improved for their next sprint. Retrospectives are essential to continuously improve your sprint process and ensure key learnings are incorporated for next time.

Things that went well

EM

- Spilled-over story points counts have been reduced.

BA

- Completed most of the bugs.

QA3

- None of the stories were put to on-hold state.

Things that didn’t go well

Developer2

- Some stories were delayed during code review state.
- Need a way to track R&D efforts during the sprint.

TN - Transpomate

Project Decisions

Moved all Automation epic tasks to not-selected status.

Sprint 13

Date - 9/12/2023

Decision made by: EM

Decision: Removed 16.5 story points worth of stories from the sprint.

Reason: overcommitment during planning.

CC: Developer , Developer

Date - 9/14/2023

Decision made by: PO

Decision: DevOps and deployment architecture implementation and Reports functionalities has to be a part of phase 1.

CC: COO

Sprint 14

Date - 9/26/2023

Decision made by: EM

Decision:

→ Before starting a sprint, frontend developer, backend developer and tester should be assigned to each story.

→ Decided to clear out all the spilled over stories and setup sprint goals for that.

Date - 10/05/2023

Decision made by: EM

Decision:

→ There should be a way to review the completed work against the sprint goals during the middle of the sprint. (Preferably send an email to the EM)

Research & Development

Power BI for reports

What is Power BI?

Power BI is a business intelligence (BI) and data visualization tool developed by Microsoft. It allows users to connect to a wide range of data sources, including on-premises and cloud-based databases, spreadsheets, and web services. Once the data is connected, users can create interactive reports and dashboards that can be shared with others.

Power BI is a self-service BI tool, meaning that it is designed for users of all skill levels, including those with no prior BI experience. It is easy to use and provides a wide range of features for creating and sharing data insights.

Some of the key features of Power BI include:

- **Data connectivity:** Power BI can connect to a wide range of data sources, including on-premises and cloud-based databases, spreadsheets, and web services.
- **Data visualization:** Power BI provides a wide range of data visualization options, including charts, graphs, maps, and tables.
- **Interactive reports and dashboards:** Power BI users can create interactive reports and dashboards that can be shared with others.
- **Natural language querying:** Power BI supports natural language querying, which allows users to ask questions about their data in plain language.
- **Collaboration:** Power BI makes it easy to collaborate with others on reports and dashboards.

How is it beneficial for our Product?

- **Improved collaboration:** Power BI makes it easy to share reports and dashboards with others, both inside and outside of the organization. This can help to improve collaboration between different teams and departments, and with partners and stakeholders.
- **Safety and security:** Power BI can be used to analyze safety and security data. This information can be used to identify and address potential safety and security risks.
- **Reduced costs:** Power BI can help to reduce the costs associated with data analytics and reporting. Power BI is a cloud-based service, so there is no need to invest in expensive hardware or software. Additionally, Power BI is relatively easy to use, so businesses do not need to invest in extensive training for their employees.

How to use Power BI with Azure

Power BI can be integrated with Azure in a number of ways. For example, you can use Azure data sources to connect to your data in Power BI. You can also use Azure services to host and deploy your Power BI reports and dashboards.

Here are some examples of how you can use Power BI with Azure:

- **Connect to Azure data sources:** Power BI can connect to a wide range of Azure data sources, including Azure SQL Database, Azure Synapse Analytics, and Azure Data Lake Storage. This allows you to bring your Azure data into Power BI for analysis and visualization.
- **Host and deploy Power BI reports and dashboards:** You can use Azure services, such as Azure App Service and Azure Blob Storage, to host and deploy your Power BI reports and dashboards. This allows you to share your reports and dashboards with others and make them available on the web and mobile devices.
- **Use Azure services to enhance your Power BI reports:** You can use Azure services, such as Azure Machine Learning and Azure Cognitive Services, to enhance your Power BI reports with artificial intelligence (AI) capabilities. For example, you can use Azure Machine Learning to create predictive models that can be used to forecast future trends in your data.

How to integrate Power BI in our own app

Power BI Embedded allows developers to embed Power BI reports, dashboards, and tiles into their own applications. This allows developers to provide their users with interactive data visualizations and insights without having to build their own BI solution.

To implement Power BI in a web app, developers need to:

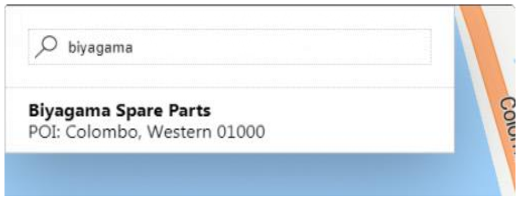
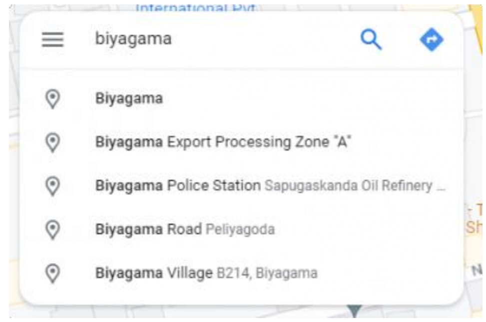
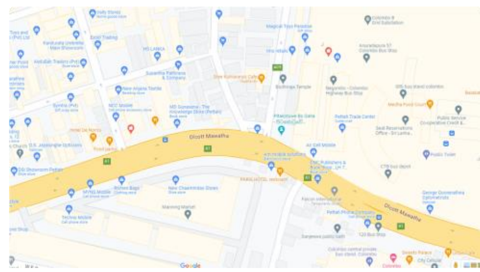
1. Create a Power BI workspace and publish the reports and dashboards that they want to embed in their web app.
2. Register an Azure AD application and obtain the client ID and client secret.
3. Embed the Power BI reports and dashboards in their web app using the Power BI Embedded API.

For more detailed steps, Microsoft provides a number of resources to help developers implement Power BI in their web apps, including documentation, tutorials, and sample code.

Useful Articles and Tutorials on Power BI

- **Articles:**
  - Microsoft Power BI on Azure
  - Power BI + Azure | Microsoft Power Platform
  - Azure and Power BI - Microsoft Learn
- **Videos:**
  - Power BI Integration with Azure | Power BI Tutorial 2023 | Power BI for Beginners | K21Academy
  - Real-time Dashboard Using Azure Cloud & Power BI | Intellipaat
  - PowerBI Azure Tutorial For Beginners | Power BI Integration with Azure | Edureka | Azure Live - 3
  - How to start building Data & Analytics solutions using Power BI & Azure?
  - Azure DevOps Integration with Power BI | Azure DevOps Analytics to Power BI Dashboards

Comparison

	Azure Maps	Google Maps
Extensive search functionality (Which requires for users to select their drop locations by search their nearest landmarks)		
Richness and depth of data of the map		
Timeliness and reliability		
Multi-Language Support	Limited	Comprehensive

Conclusion

Extensive Search Functionality:

- Users easily select drop locations by searching for landmarks or addresses.
- Real-time autocomplete suggestions and a vast database of points of interest enhance convenience.

Richness and Depth of Map Data:

- Google Maps offers detailed, up-to-date maps with continuous updates.
- Satellite imagery and Street View provide visual insights and navigation assistance.

Timeliness and Reliability:

- Real-time data updates, including traffic information, ensure accurate navigation.
- Google Maps is known for its consistent and reliable map data, making it a trusted choice for users.

In summary, Google Maps excels in offering extensive search capabilities, rich map data, and timely, dependable information, making it the preferred choice for a wide range of navigation and location-based services.

Google map localization on mobile

This document will attempt to capture the process of localizing google maps within the flutter framework for android devices.

We first start off by adding the google maps plugin to our flutter project, or specifically the ***pubspec.yaml*** file:

```
google_maps_flutter:
```

Then, we add the Google Maps API key in the **AndroidManifest.xml** file:

```
<meta-data android:name= "com.google.android.geo.API_KEY"
    android:value="YOUR API KEY HERE"/>
```

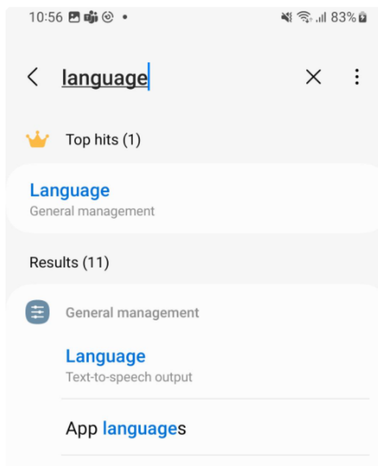
Following this, we shall display the map, by initializing the GoogleMap widget in flutter:

```
GoogleMap(
  zoomControlsEnabled: false,
  compassEnabled: false,
  tiltGesturesEnabled: false,
  polylines: Set<Polyline>.of(polylines.values),
  mapType: MapType.normal,
  initialCameraPosition: CameraPosition(
    target: LatLng(
        widget.previewDriverEndLat, widget.previewDriverEndLong), // LatLng
    zoom: 10.0,
  ), // CameraPosition
  markers: {
    Marker(
      markerId: MarkerId("Driver Start Location"),
      infoWindow: InfoWindow(
        //tooltip info
        title: '${widget.previewDriverStartPointName}',
        snippet: 'Voyage Start Point',
      ), // InfoWindow
```

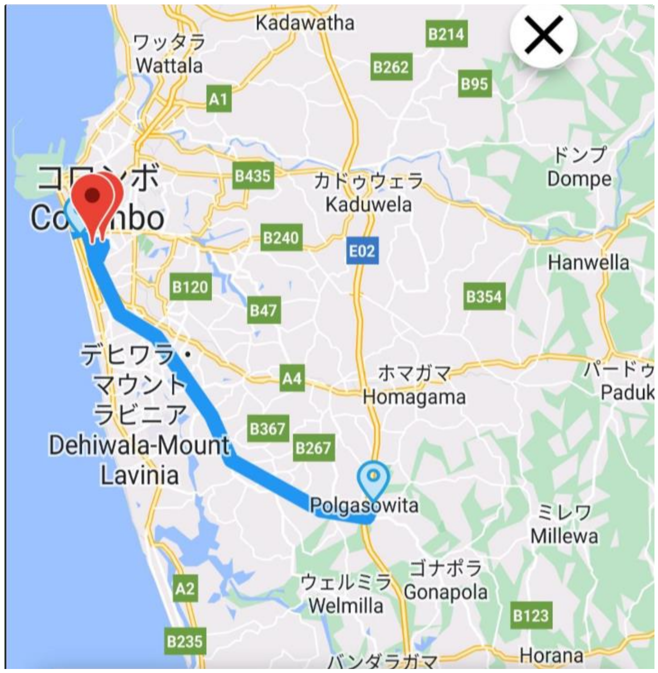
```
      icon: markerIcon,
      position: LatLng(widget.previewDriverStartLat,
          widget.previewDriverStartLong), // LatLng
    ), // Marker
    ...generateDropLocationMarkers(),
    Marker(
      markerId: MarkerId("Driver End Location"),
      infoWindow: InfoWindow(
        title: '${widget.previewDriverEndPointName}',
        snippet: 'Voyage End Point',
      ), // InfoWindow
      icon: markerIcon,
      position: LatLng(
          widget.previewDriverEndLat, widget.previewDriverEndLong),
    ), // Marker
  },
  onMapCreated: (mapController) {
    _controller.complete(mapController);
  },
), // GoogleMap
```

The result is the following view, when you hot reload the application:

Now, let’s assume that as the viewer, you cannot understand the language the map is in. Instead, you are more familiar with **Japanese**. To ensure that this is reflected on the map, you simply have to change the **device language** by accessing the device settings:



The map view will look like this:



Currently, however there is no property to change the locale, within the google maps flutter package. This plugin simply wraps itself around the Android SDK, so we cannot control the language settings apart from the standard system localization demonstrated above.

External Readings:

- Adding Google Maps to Flutter. This article will show you step-by-step… | by Kenzie Davisson | Flutter | Medium

- [google_maps_flutter]: Localization in google maps · Issue #100118 · flutter/flutter (github.com)

SMS Service Comparison

This document compares three alternative SMS service providers to Azure Communication Services SMS. Azure Communication Services is a reliable service, but it has limited regional availability. This prevents us from creating an Azure (ACS SMS) resource within our Azure tenant.

Sending SMS alerts can be identified as one of the core components of our system. It allows us to send SMS alerts to users regarding various event that occur within our system such as,

- Trip Request Approve / Decline.
- Voyage Allocations / Deallocations etc.

This comparison is solely to compare and select a SMS service for development and QA testing, as there are plans of integrating our application with the client's existing SMS gateway.

Introduction

For this comparison the following 3 services will be considered.

1. Twilio - (Communication APIs for SMS, Voice, Video & Authentication | Twilio)
2. Infobip - (Build meaningful customer relationships across any communication channel)
3. Notify.lk - (Notify - SMS Gateway, Campaign Manager and API Provider in Sri Lanka)

1. Twilio

Twilio is a cloud communications platform that enables developers to build, scale, and operate real-time communication applications. It provides a suite of APIs that allow developers to add voice, video, messaging, and authentication capabilities to their applications.

Twilio is used by businesses of all sizes, from startups to Fortune 500 companies, to build a wide range of communication applications.

Twilio is a popular choice for developers because it is easy to use, scalable, and reliable. It also offers a wide range of features and integrations, making it a versatile platform for building a variety of communication applications.

Here are some real-world businesses who uses Twilio:

- **Airbnb**
- **Netflix**
- **Uber**
- **Slack**

3. Infobip

Infobip is a global cloud communications platform that enables businesses to reach their customers across multiple channels, including SMS, email, voice, video, and social media.

Infobip is used by businesses of all sizes, from startups to Fortune 500 companies, to improve their customer engagement and communication. It is a popular choice because it is easy to use, scalable, and reliable. It also offers a wide range of features and integrations, making it a versatile platform for businesses of all types.

Here are some real-world businesses who uses Infobip:

- **Nike**
- **Spotify**
- **Vodafone**

3. Notify.lk

Notify.lk is a SMS gateway provider for Sri Lanka. It is a SMS gateway provider for Sri Lanka. It is developed and maintained by Ceylon Systems. Notify has the coverage to send SMS for any phone number of any network which has SMS receiving capability.

Here are some Sri Lankan businesses who use Notify:

- **Washapp**
- **Chenara Dodge**
- **Alaris**
- **Ganna.lk**

Comparison

	Twilio	Infobip	Notify.lk
Pricing Model	Region based Consumption Based Billing (Pay as you Go)	Region based Consumption Based Billing (Pay as you Go)	Pay as you go + Packages Available
Pricing Page	SMS Pricing in United States for Text Messaging | Twilio	Pricing - SMS - portal.infobip.com	Pricing - Notify.lk
Price per SMS (Sri Lanka)	0.2260 USD	0.3085 USD	Package Dependent **Pay as you go:** One-time fee: Rs.7,500/- SMS Cost: Rs.0.60/-* **Lite Package:** **Rs.690**/Month

			FREE SMS: 480 SMS Cost: Rs.1.10/-* - Free SMS count is the calculated number of SMS that you can send for the monthly fee you are paying.

			• Free SMS quota will be added as a balance in your account and this quota will be valid for 60 days from the offered date.
Global SMS Availability?	180+ Countries supported.	Not clearly mentioned (Likely supports 150+ countries)	Only Supports Sri Lanka
Restricted Countries if any	no info	no info	Only Supports Sri Lanka
Free Trial Available	Yes (Credit System). The Twilio free trial offers up to $15 worth of credits to be spent on their services. Since the cost is region dependent the amount of messages that can be sent may differ	Yes. The free trial offers up to 100 messages to be sent.	Yes. The Free Trial offers up to 10 messages to be sent.
Free Trial Limitations	https://support.twilio.com/hc/en-us/articles/360036052753-Twilio-Free-Trial-Limitations	• Can only send up to 100 messages. • Can only send messages to verified numbers.	• Can only send up to 10 messages.
Max SMS Length (chars)	Up to 160 characters, or up to 70 if the message contains one or more Unicode characters.	Up to 160 characters, or up to 70 if the message contains one or more Unicode characters.	Maximum character count for a single paged SMS is 160 (70 for Unicode content). You can send up to 621 characters in a SMS (268 for Unicode contents).
Bulk SMS Support	Yes Bulk SMS support is available. However, the messages aren't cc'd like emails. A message to each recipient is treated as 1 message, so we are required to loop over the number of recipients.	Yes. Bulk SMS support is available. However, the messages aren't cc'd like emails. Have to call the REST endpoint for per each recipient.	Yes. Bulk SMS support is available. However, the messages aren't cc'd like emails. Have to call the REST endpoint for per each recipient.
Analytics Dashboard	Yes	Yes.	Yes.

HTTP API (REST)	Yes	Yes	Yes
Developer Documentation (API)	API Reference - In-Depth Reference for all Twilio APIs and SDKs	Infobip Documentation Hub	Notify.lk API Documentation
NPM Packages	npm: twilio	npm: @infobip-api/sdk Note: The above package is a community developed package	not available for Node.js
SMS Testing though Web portal	DONE	DONE	DONE
Service Dev R & D	DONE	DONE	DONE
Developer Comments	• Integration with the provided NPM package is straightforward and easy.	• The NPM package works to a certain extent, but since it's a community developed package, there are concerns regarding LTS. • Tested the REST endpoint with Postman.	• Tested the REST endpoint with Postman.

My Comments

All the above three services provide a limited free trial account for us to test out their features. However, the trial versions may not be sufficient for development and thorough Dev and QA testing. Considering the use cases in our system the trial SMS / Credit limit will be expended mid testing.

If we are to consider testing the system end-to-end for different use-cases, I believe subscribing to an appropriate service is the best option, as it's difficult to estimate the number of messages required for development and testing.

Technical Debts / Pending Tasks for Ansell release

Item	Sub items	Epic	Jira ID	Estimated days.
Multi-location within single organization support	Tenant model of the application needs to be configured according to the requested organization hierarchy. (Should allow only one level of hierarchy such as location or department etc.)	~~TN-742: Multi-Location Support~~ ~~NOT READY~~	~~TN-709: As a Product Admin, I need to configure our application's tenant model to support multiple locations within a single organization~~ ~~NOT READY~~	Dev: 20 Test: 3
Tenant onboarding / tenant enrollment	• Information about the organization should be configured. • Manual tenant onboarding	TN-730: Tenant Onboarding NOT READY	TN-710: As a product Admin, I need to capture information on the new tenant, So that I can manually onboard new tenants to the application. BA IN-PROGRESS	Dev: 5 Test: 2
	The system should create an admin role and add the user details to that role and give particular permissions for the role.	TN-730: Tenant Onboarding NOT READY	TN-711: As a Product Admin, I want the system to create an "Admin" role and add specific user details to that role. BA IN-PROGRESS	
	Tenant provisioning - once the organization information received and configured, a subscription needs to be assigned to the tenant.	TN-730: Tenant Onboarding NOT READY	TN-712: As a Product Admin, I need to provision a tenant's account once the organization's information has been received and configured BA IN-PROGRESS	Dev: 10 Test: 2
Database transactions	-	TN-621: Technical Debts - Release 1 DEVELOPMENT IN PROGRESS	TN-593: As a User, I want my data to be consistently stored in a database while maintaining integrity. (using db transactions) DEV - TODO	Dev: 6 Test: 2
Mobile Code refactoring	• Migration of project structure to clean code architecture. • Migrate code base to provider pattern. • QA Regression	TN-621: Technical Debts - Release 1 DEVELOPMENT IN PROGRESS	TN-661: Code Refactor Mobile - Migrate to Provider Pattern NOT READY	Dev: 10 Test: 1
DevOps	IaaC for Azure resource provisioning (BE, FE)	TN-741: DevOps NOT READY	TN-722: Create Infrastructure as Code (IaaC) for Azure Resourc	Dev: 12

			e Provisioning (Backend and Frontend) NOT READY	
	Automation test for CI pipeline for all 3 components (BE, FE, Mobile)	TN-741: DevOps NOT READY	TN-723: Run Automation Tests inside Continuous Integration (CI) Pipeline for All Three Components (Backend, Frontend, Mobile) NOT READY	Dev: 6
	Azure resource provisioning part should be included in CD pipeline (GitHub) for all 3 components (BE, FE)	TN-741: DevOps NOT READY	TN-724: Create Azure Resource Provisioning in Continuous Deployment (CD) Pipeline for All Three Components (Backend, Frontend) - Development Env. DONE	Dev: 5
	Mobile quality scan	TN-741: DevOps NOT READY	TN-725: Mobile code quality scan TOGROOM	Dev: 5
	Build environment specific (QA, staging, production) CI/CD pipelines for all components.	TN-741: DevOps NOT READY	TN-726: Build Environment-Specific CI/CD Pipelines for All Components (QA, & Staging) TOGROOM	Dev: 9
User login (Web)	Azure tenant ID should not be passed via requests, a unique string ID should be generated for a tenant and that should be used in the requests.	TN-599: Login & Authorization DEVELOPMENT IN PROGRESS	TN-739: A unique string ID should be generated for a tenant instead of the Azure tenant ID NOT READY	Dev: 3 Test: 2
User Profile (Web)	• User should be able to create an employee account if he does not have one. When login, a pre-filled form should be shown with the data we can gather from the azure authentication and the user should be able to fill out the other required data and create the account.	TN-740: User Profile NOT READY	TN-728: As a user, I need to create an employee account if I don't have one already. DELETED	Dev: 6 Test: 2
	User should be able to edit his employee details from the edit profile page.	TN-740: User Profile NOT READY	TN-727: As a logged in User, I need to edit my employee details DELETED	

User should be able to view his profile	TN-622: As a logged in User, I need to view my user profile, So that

			I can see and manage my personal information TOGROOM	
Change notification service email sending domain.	Need to change the `DoNotReply` domain to Ansell domain.	https://Company.atlassian. net/browse/TN-640 Can't find link	TN-713: As a Product Admin, I need to update the domain used for outgoing "Do Not Reply" emails to the official Ansell domain. TOGROOM	Dev: 1
	Complete SMS sending tasks.	https://Company.atlassian. net/browse/TN-640 Can't find link	• TN-630: As an Employee, I need to receive an SMS when my trip has been allocated to a voyage DONE • TN-628: As an Employee, I need to receive an SMS when a manager approved my request, So that I am alerted about my trips status DEV - TODO	Dev: 6 Test: 2
	Complete Email sending tasks.		• TN-635: As a Manager, I should receive an email when a voyage is abandoned, So that I'm alerted if my employee's voyages have been abandoned NOT READY • TN-634: As a Transport Coordinator, I should receive an email when a voyage is abandoned, So that I'm alerted when a voyage is abandoned NOT READY • TN-633: As a Transport Coordinator, I should receive an email when the driver ends the voyage, So that I'm alerted when the voyage ends TOGROOM	Dev: 8 Test: 3

			• TN-632: As a Manager, I should receive an email when the Transport Coordinator deletes a scheduled voyage, So that I can stay informed on allocations of my employees. TOGROOM • TN-631: As a Manager, I need to receive an email when my employee's requests have been allocated to a voyage DEV-CODE REVIEW • TN-629: As a Transport Coordinator, I should receive an email when a manager approves an employees request, So that I can allocate them to a voyage. DEVELOPMENT IN PROGRESS • TN-627: As a Manager, I need to receive an email, So that I am notified that an employee has requested a trip. DONE	
Functional testing	Unit testing for all the backend business cases. (should have the 80% of coverage)	TN-744: Unit Test coverage completion (Technical debt) DELETED	TN-743: Unit Testing for Backend (Business Case) DELETED	Dev: 10
	Test automation for FE and Mobile.	TN-500: Frontend Automation DEV - TODO TN-745: Mobile Automation NOT READY		Test: 25 Might need a senior automation resource.
Non-functional testing and implementation	• Need to test for both mobile platforms (Android, IOS) - Will only be testing on one latest device configuration of both the platform. • Effort will be differed according to the number of devices, supporting OS versions and etc.	TN-751: IOS Implementation NOT READY https://Company.atlassian.net/browse/TN-750 Can't find link	TN-746: IOS implementation (build) and UI responsiveness - Mobile NOT READY TN-747: As a Mobile User, I need the app screens to adapt seamle	Dev: 7 Test: 10

	• Need to fix the UI responsiveness in both platforms.		ssly to different screen sizes, So that I can have a consistent and user-friendly experience. TOGROOM	
	• Need to test web application in different browsers (at least Chrome, Edge, and Firefox). • UI responsiveness should be there for the selected screen sizes.		TN-748: As a user, I want to ensure that Transpomate functions seamlessly across different browsers, So that I can ensure a consistent and reliable experience. TOGROOM ~~TN-675: As a logged-in user, I need to view the User's page in different screen sizes.~~ QA - TODO ~~TN-676: As a logged-in user, I need to view the Department page in different screen sizes.~~ DEV - TODO ~~TN-677: As a logged-in user, I need to view the Cost Center page in different screen sizes.~~ NOT READY ~~TN-678: As a logged-in user, I need to view the Owner page in different screen sizes.~~ NOT READY ~~TN-679: As a logged-in user, I need to view the Vehicles page in different screen sizes.~~ DEVELOPMENT IN PROGRESS ~~TN-680: As a logged-in user, I need to view the Trip Request page in different screen sizes.~~ NOT READY ~~TN-681: As a logged-in user, I need to view the Roles page in different screen sizes.~~ NOT READY	Dev - 3 & Testing: 12

			TN-682: As a logged-in user, I need to view the Manage Trips page in different screen sizes. NOT READY TN-683: As a logged-in user, I need to view the Voyages page in different screen sizes. NOT READY TN-595: As a logged-in user, I need to view the Employee page in different screen sizes. DONE	
	Do a load testing for all 3 components (FE, BE, Mobile)	TN-767: Load Testing DEV - TODO	TN-731: Conduct Load Testing for Frontend NOT READY TN-752: Conduct Load Testing for Backend NOT READY TN-753: Conduct Load Testing for Mobile NOT READY	Test: 20
	• Need to do a security testing for all 3 components (FE, BE, Mobile) ◦ API keys and blob storage keys has to be stored in a key-vault in Mobile.	TN-768: Security Testing DEV - TODO	TN-732: Conduct Security Testing for Frontend NOT READY TN-754: Conduct Security Testing for Backend NOT READY TN-755: Conduct Security Testing for Mobile NOT READY	Test: 10
Bug fixes	Need to find active bug count for all 3 components.			Dev: 20 Test: 5
Mobile app error logs - Crashlytics	• Implement firebase Crashlytics to the mobile application. • Firebase subscription is required.	TN-756: Error Logs NOT READY	TN-729: Implement Firebase Crashlytics in the Mobile Application NOT READY	Dev: 3
System monitoring and observability.	Application monitoring for all the 3 components.	TN-757: Technical Debt - System Monitoring DELETED	TN-735: Implement Application Monitoring for All Three Components NOT READY	Dev: 12
	Infra monitoring - cloud resources.	TN-757: Technical Deb		

		DELETED	g NOT READY	
Application logs for web FE	Application logs should be done using application insights in FE.	TN-756: Error Logs NOT READY	TN-737: Implement Application Logging for Web NOT READY	Dev: 3
Bulk upload employees	• User should be able to upload the employee details to the system via a csv file. • User should be able to download a sample csv file and edit the relevant employee details there and upload that file to the system.	TN-226: Administration NOT READY	TN-714: As an Admin, I need to upload employee details to the system using a CSV file BA IN-PROGRESS	Dev: 10 Test: 3
Enable / disable filtration of the entities.	• Admin should be able to enable / disable an entity (employee, department, cost center, vehicle owner, vehicle, driver). • Disabled entities should not be accessible for any task done within the system.	TN-226: Administration NOT READY	TN-758: Enable and Disable Employees NOT READY TN-759: Enable and Disable Departments NOT READY TN-760: Enable and Disable Cost Centers NOT READY TN-761: Enable and Disable Vehicle owner NOT READY TN-762: Enable and Disable Vehicle NOT READY	Dev: 6 Test: 3
Abandon voyage	• TC should be prompted immediately when the driver abandoned the voyage. • Preferable to show the `Call TC` option mandatory and send an email.	TN-749: Notification Service V1 - Ansell DEVELOPMENT IN PROGRESS	TN-634: As a Transport Coordinator, I should receive an email when a voyage is abandoned, So that I'm alerted when a voyage is abandoned NOT READY	Dev: 15 Test: 3
	• Voyage summary should indicate the number of trip requests completed and remaining. • Cost calculation should be done up to the abandoned point of the scheduled voyage.	TN-763: Abandon Voyages DEV - TODO	TN-716: As a Driver, I need to view a clear voyage summary of an abandoned voyage NOT READY	
	• TC should be able to allocate the remaining employees to a separate voyage from that point. • TC should also have the opportunity to decide whether a new voyage should be allocated or not. (in the	TN-763: Abandon Voyages DEV - TODO	TN-717: As a Transport Coordinator, I need to allocate the remaining passengers to a new voyage immediately when/if a voyage is a	

bandoned. **NOT READY**

	cases like abandoning happen very closer to the destination of voyage)			
Intune integration for mobile	Intune integration should be done in the mobile application for the driver authentication.	TN-668: Driver - Login DEV - TODO	TN-734: Intune Integration for Mobile LEAD REVIEW	Dev: 5
Driver registration	Admin should be able to create driver entities.	TN-668: Driver - Login DEV - TODO	TN-765: As an Admin, I need to create driver entities within the system. DELETED	Dev: 5 Test: 2
	Admin should be able to link driver entity with vehicle entity and vehicle owner entity.	TN-668: Driver - Login DEV - TODO	TN-766: As an Admin, I need to link an Azure account to the vehicle entity, So that the driver can log into the Mobile App. DEV-CODE REVIEW	
Driver login	Driver should be able to login to the system and see the allocated voyage for the time being using mobile application.	TN-668: Driver - Login DEV - TODO	TN-602: As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app DEVELOPMENT IN PROGRESS	Dev: 6 Test: 2
Subordinate trip request	Manager should be able to see his own trip requests and subordinate trip requests separately.	TN-100: Manager view trips option DEVELOPMENT IN PROGRESS	TN-718: As a Manager, I need to view trip requests in a way that distinguishes between my own trip requests and those of my subordinates DEV - TODO	Dev & Test: 3
Voyage odometer image upload	• Re-uploaded image should replace the previously uploaded image/s. • There should be 2 endpoints; to upload and to delete. • Images should be scanned by a malware scanner.	TN-608: Odometer image upload DEVELOPMENT IN PROGRESS	TN-513: As a Driver, I need to upload the selected image, So that the TC could use the image to compare with the odometer text the driver entered. DONE	Dev: 9 Test: 1
	Need to create a scheduled job to clear out the images related to not started voyages. (preferably run daily)	TN-608: Odometer image upload DEVELOPMENT IN PROGRESS	TN-738: Create a scheduled Job for clearing odometer images TOGROOM	Dev: 1 Test: 0.5
Voyage UI change	Need an icon to indicate voyage		s	t

tuses (in Progress / completed / abandoned) in both mobile and web.	TN-662: Manage voyages DEV - TODO	TN-719: As a Transport Coordinator, I need to view a clear and visually distinct icons for	Dev: 3 Test: 1

			all voyage statuses (Scheduled, In-progress. Completed, Abandoned) NOT READY	
Permission management	Admin permission management	TN-684: Permission guards DONE	TN-673: As an Admin, I need to access only the functions assigned to me, So that I can do the specific assigned functions within the system. DONE	Dev: 6 Test: 3
	Manager permission management	TN-684: Permission guards DONE	TN-647: As a Manager, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system. DONE	
	Transport Coordinator permission management	TN-684: Permission guards DONE	TN-654: As a Transport Coordinator, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system. DONE	
	Generic User permission management	TN-684: Permission guards DONE	TN-653: As a Generic User, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system. DONE	
	HR permission management	TN-684: Permission guards DONE	TN-720: As an HR associate, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system. BA IN-PROGRESS	
	Finance permission management	TN-684: Permission guards DONE	TN-721: As a Finance associate, I need to access only the functions assigned to me by the admin, So that I can do the specific assigne	

			d functions within the system. BA IN-PROGRESS	
Reporting	Power BI integration	TN-620: Reports DELETED	TN-764: Power BI Integration DELETED	Dev: 20 Test: 3
	Reports need as an HR associate. • Trip request analysis by employee.	TN-806: Power BI Reports DEV - TODO	TN-798: As an HR associate, I need to view the Trip Summary of employees on a Power BI integrated webpage, So that I can search and view a summary of trip requests for the searched employee. ON-HOLD	
	Reports need as a Finance associate. • Cost calculation by vehicle.		TN-800: As a Finance associate, I need to view the Payment Summary of vehicles on a Power BI integrated webpage, So that I can search and view a summary of payments for the searched owner and/or vehicle. TOGROOM	
Mobile Application Multi Language Support		TN-769: Multi-language support DEV - TODO	TN-733: Implement Localization in the Mobile Application NOT READY	Dev - 15 Test - 10

Dependency management

Issue Dependencies - Sprint 13

Overall status	URGENT
Total Impacted story count/points.	~~9 / 44.5~~
Blocked story count / points	~~1 / 18~~
Current velocity in story points	16.5
Dependency clearance due date	Sep 22, 2023 CLEARED
Dependency Score	to-be-decided

Issues Impacted

Key	Summary	Assignee	Reporter	Status	Frontend Dev.	Backend Dev.	Tester	Story Points
TN-598	As an admin, I need to search for users, So that I can assign the searched user with a role (assigning the role isn't included)	QA	BA	DONE	Sasani Jayathilake	dileepa.mabulage	QA	5.0
TN-591	As a Transport Coordinator, I need to receive an automated email, So that I know that the voyage has started	QA	BA	DONE	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	13.0
TN-561	As a Transport Coordinator, I need to create new starting and ending points for a voyage, So that I can change the voyage route.	QA	BA	DONE	Abdul Qadir	Thambara Supun Basnayake	QA	5.0
TN-513	As a Driver, I need to upload the selected image, So that the TC could use the image to compare with the odometer text the driver entered.	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	5.0
TN-461	As a Driver, I need to verify the attendance of the passengers and mark the checkbox if present, So that I can see the pins on the map for the present passengers	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0
TN-346	As a Driver, I want to enter the odometer reading, so that I can start the voyage.	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan	dileepa.mabulage	QA2	0.5
TN-345	As a Driver, I want to take a picture of the odometer, so that I can keep track of the odometer beginning reading before starting the voyage.	QA2	Lithira.Senath	DONE	Rishimithun Muralidharan		QA2	5.0
TN-209	As a Driver, I want to drop trips for employees, So that the mobile app will be notified with each drop and help in preparing the report.	QA2	Thanusan irran	DONE	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	5.0
TN-207	As a Driver, I want to view a list of the passengers, So that I know which passengers have been allocated to the voyage and mark their attendance.	QA2	Thanusan irran	DONE	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0

9 issues

Dependencies

Issue Key	Dependancy	Status
TN-598	Sequence Diagram – Developer	NON-CRITICAL
TN-346	• API keys to be shared with Developer. - Developer - DONE • Architecture diagram - Architect1 - DONE	DONE
TN-513	• User story / Acceptance Criteria / BA Diagram - BA - DONE • Blob storage for the QA environment. Setup QA environment - Developer - DONE • Architecture Diagram - Architect1 - DONE • Device for QA testing - SM - DONE • Sequence Diagram - Developer2 DONE • Deployment Diagram - Developer / Architect1 to confirm which deployment method to use. (candidate choices: GitHub actions, azure Devops)	DONE
TN-345	QA environment resource provisioning - Developer - DONE	DONE
TN-209	• Architecture Diagram - Architect1 - DONE • Sequence Diagram - Developer	NON-CRITICAL
TN-461	• Database Diagram & Changes - Developer2 - DONE • Sequence Diagram - Developer2	NON-CRITICAL
TN-207	• Sequence Diagram - Developer	NON-CRITICAL
TN-561	• Sequence Diagram - Frontend Developer	DONE
TN-591	• Create required cloud resources in Dev environment. Pending due to naming convention issue. - Architect1 - DONE • Create required cloud resources in QA environment. Pending due to naming	NON-CRITICAL

	convention issue. - Architect1 - DONE - Sequence Diagram - Developer2 DONE - Deployment Diagram - Developer - REMOVED - Happy path test cases - QA2	

Issue Dependencies - Sprint 14

Dependency summary

Overall status	URGENT
Total impacted estimated story count/points.	~~5 / 17~~
Total impacted not-estimated story count.	3
Blocked story count / points	3
Current velocity in story points	26.4
Dependency clearance due date	Oct 12, 2023 CLEARED
Dependency Score	to-be-decided

Backlog summary

Unplanned (Backlog) story count / points	4 / 18
Planned story count / points.	15 / 56
Tentative next sprint story count / points.	10 / 24
Not ready story count	47

Issues Impacted

Key	Summary	Assignee	Reporter	Status	Frontend Dev.	Backend Dev.	Tester	Story Points
	and view the pending voyages							
TN-602	As a Driver, I need to validate my email	BA	dileepa.mabulage	QA2	5.0			

QA2

8 issues

Dependencies

Issue Key	Dependency	Status
TN-663	• R&D on deriving time zone from the offset sent from client side - Developer2 DONE • Database diagram - Developer2 PENDING	NON-CRITICAL
TN-661	• Need to discuss with Architect1 how the code refactoring should be done. DONE	DONE
TN-616	• Sequence diagram - Developer DONE	DONE

TN-602	- User story needs to be updated. BA PENDING - Architect1 suggested to maintain a separate entity called `Driver` in the app. Need to make stories related to that entity and need to get approval from PO - BA DONE - UI design - Error page should be designed to show if the login feature is crashed. - BA NEED REVIEW - Cloud resource for Dev & QA - need to create separate user accounts for Drivers and their accounts should be password less accounts in Azure. - Developer PENDING - Skills - Need to do R&D on Microsoft Authenticator app integration. - Developer PENDING - Architecture diagram - Architect1 PENDING - Database diagram, sequence diagram, deployment diagram -	BLOCKED

	Developer	
TN-592	• Acceptance criteria - BA DONE • Resource provisioning Dev & QA - Developer PENDING • R&D on how to create a secure URL to upload files - Developer PENDING • R&D on how to clear the files stored in the server side. Developer PENDING • Database diagram, sequence diagram - Developer DONE	CRITICAL

Issue Dependencies - Sprint 15

Dependency summary

Overall status	NOT-CRITICAL
Total impacted estimated story count/points.	6 / 32
Total impacted not-estimated story count.	1
Blocked story count / points	0
Current velocity in story points	26.4
Dependency clearance due date	Oct 20, 2023
Dependency Score	to-be-decided

Backlog summary

Unplanned (Backlog) story count / points	26 / 107
Current sprint story count / points.	14 / 42
Tentative next sprint story count / points.	8 / 30
Not ready story count.	99
Bugs count.	45

Issues Impacted

Key	Summary	Assignee	Reporter	Status	Frontend Dev.	Backend Dev.	Tester	Story Points
9 issues								

Dependencies

Issue Key	Dependency	Status
TN-766: As an Admin, I need to link an Azure account to the vehicle entity, So that the driver can log into the Mobile App. DEV-CODE REVIEW	• User story - BA DONE Acceptance • criteria - BA DONE • UI - BA DONE ◦ Add vehicle page, Edit vehicle page, View vehicles page.	DONE
TN-728: As a user, I need to create an employee account if I don't have one already. DELETED	DELETED	DELETED
TN-727: As a logged in User, I need to edit my employee details DELETED	DELETED	DELETED

TN-718: As a Manager, I need to view trip requests in a way that distinguishes between my own trip requests and those of my subordinates DEV - TODO	• Acceptance criteria - BA DONE • UI - BA DONE • Sequence diagram - Frontend Developer PENDING	NOT-CRITICAL
TN-630: As an Employee, I ~~need to receive an SMS when my trip has been allocated to a voyage~~ DONE	• Cloud resource for Dev & QA / Subscription - SMS service in azure is not provisioned for the organization subscription. need to check with PO , IT and Architect1 - Developer DONE • Skills - R&D on writing function app in NestJs - Developer2 PENDING Database diagram - Developer2 DONE •	NOT-CRITICAL

TN-627: As a Manager, I need to receive an email, So that I am notified that an employee has requested a trip. DONE	-	DONE

TN-602: As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app DEVELOPMENT IN PROGRESS	• Acceptance criteria needs to be updated. BA DONE
	• UI design - BA DONE
	  ◦ `Error page` should be designed to show if the login feature is crashed.
	  ◦ `Not access page` should be designed to show if the user has the access to login or not.
	• Skills - Developer PENDING
	  ◦ R&D on Microsoft Authenticator app integration.
	  ◦ R&D on flutter package for login with Microsoft.
	• Sequence diagram - Developer DONE
		NOT-CRITICAL

TN-592: As a mobile tester, I need a backend endpoint to return a unique URL to upload files, So that the files can be uploaded securely. DONE	• R&D on how to create a secure URL to upload files - Developer DONE • R&D on how to clear the files stored in the server side. Developer PENDING	NOT-CRITICAL
TN-433: As an admin, I want to edit users, So that I can manage their accounts and roles effectively. DEV - TODO	• Acceptance criteria - color change in role name tags BA DONE • UI - BA DONE	DONE

Issue Dependencies - Sprint 16

Dependency summary

Overall status	
Total impacted estimated story count/points.	7 / 50
Total impacted not-estimated story count.	3
Blocked story count / points	1
Current velocity in story points	20.2
Dependency clearance due date	Nov 9, 2023
Dependency Score	to-be-decided

Backlog summary

Unplanned (Backlog) story count / points	16 / 73
Current sprint story count / points.	11 / 43
Tentative next sprint story count / points.	10 / 40
`BA IN-PROGRESS` story count.	15
Bugs count.	16
`ToGroom` story count	23

Issues Impacted

Key	Summary	Assignee	Reporter	Status	Frontend Dev.	Backend Dev.	Tester	Story Points
TN-820	As a Driver, I need to re-arrange passengers before starting the voyage, So that I can view the re-routed map route.	Rishimithun Muralidharan	BA	DEV - TODO	Rishimithun Muralidharan		QA2	8.0
TN-798	As an HR associate, I need to view the Trip Summary of employees on a Power BI integrated webpage, So that I can search and view a summary of trip requests for the searched employee.	dileepa.mabulage	BA	ON-HOLD	dileepa.mabulage	dileepa.mabulage	QA	8.0
TN-774	[CR] As a Transport Coordinator, I need to click on the "scheduled" tab on the Manage Voyages page, So that I can view all the scheduled voyages	SM	BA	DEV - TODO	SM	Thambara Supun Basnayake	QA	8.0
TN-726	Build Environment-Specific CI/CD Pipelines for All Components (QA, & Staging)	BA	BA	TOGROOM		dileepa.mabulage		
TN-724	Create Azure Resource Provisioning in Continuous Deployment (CD) Pipeline for All Three Components (Backend, Frontend) - Development Env.	Architect	BA	DONE	dileepa.mabulage	dileepa.mabulage	dileepa.mabulage	
TN-714	As an Admin, I need to upload employee details to the system using a CSV file	BA	BA	BA IN-PROGRESS	SM	dileepa.mabulage	QA	

TN-685	As an admin, I need to unassign users from a role	SM	BA	DEV - TODO	SM		QA	5.0
TN-632	As a Manager, I should receive an email when the Transport Coordinator deletes a scheduled voyage, So that I can stay informed on allocations of my employees.	BA	BA	TOGROOM	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	
TN-631	As a Manager, I need to receive an email when my employee's requests have been allocated to a voyage	SM	BA	DEV-CODE REVIEW	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	5.0
TN-630	As an Employee, I need to receive an SMS when my trip has been allocated to a voyage	QA	BA	DONE	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	8.0

10 issues

Dependencies

Issue Key	Dependency	Status
TN-820: As a Driver, I need to re-arrange passengers before starting the voyage, So that I can view the re-routed map route. DEV - TODO		DONE
TN-798: As an HR associate, I need to view the Trip Summary of employees on a Power BI integrated webpage, So that I can search and view a summary of trip requests for the searched employee. ON-HOLD	• Cloud resource provisioning for Power BI integration in Dev & QA environment - Developer PENDING • Architecture diagram - Developer PENDING • Sequence diagram - Developer PENDING • Happy path test cases - Done	NOT-CRITICAL
TN-774: [CR] As a Transport Coordinator, I need to click on the "scheduled" tab on the Manage Voyages page, So that I can view all the scheduled voyages DEV - TODO	• Technical debt declaration - need to get the confirmation on whether the voyage delete feature needs a soft delete. DONE ◦ **Do the soft delete.** • Happy path test cases - Done	NOT-CRITICAL
TN-726: Build Environment-Specific CI/CD Pipelines for All Components (QA, & Staging) TOGROOM	• Technical debt declaration - need to get the confirmation on whether to do the deployments using GitHub actions or Azure pipelines or both - DONE ◦ **Use GitHub actions.**	TO BE REVIEWD
TN-724: Create Azure Resource Provisioning in Continuous Deployment (CD) Pipeline for All Three Components (Backend, Frontend) - Development Env. DONE	• Skills - skills on bicep templates - Developer PENDING • Architecture diagram - DONE • Deployment diagram - DONE	NOT-CRITICAL
TN-714: As an Admin, I need to upload employee details to the system using a CSV file BA IN-PROGRESS	• Acceptance criteria - BA PENDING • UI - BA PENDING • Technical debt clarification - DONE ◦ Check whether generating a csv file for the given fields from AD is feasible.	BLOCKED

	◦ Decide how to save departments and cost centers for the all employees as the employee - department relation is based on department IDs. ▪ **Use department ID and cost center ID to validate those columns. DB should be updated chunk by chunk. There should be a UI to show failed updates and the failed reason. Departments and Cost centers should be updated prior to the employee bulk update.** • Sequence diagram - **blocks with the acceptance criteria.** PENDING • Happy path test cases - QA **blocks with the acceptance criteria.** PENDING	
TN-685: As an admin, I need to unassign users from a role DEV - TODO	• UI - remove save button - BA PENDING • Sequence diagram - PENDING • Happy path test cases - Done	NOT-CRITICAL
TN-632: As a Manager, I should receive an email when the Transport Coordinator deletes a scheduled voyage, So that I can stay informed on allocations of my employees. TOGROOM	• Technical debt clarification - blocks with audit logs - need to update the `createdBy` column in the voyage table for the requirement. - DONE ◦ **Audit logs and audit columns are separate things, implement audit columns for the feature.**	TO BE REVIEWD
TN-631: As a Manager, I need to receive an email when my employee's requests have been allocated to a voyage DEV-CODE REVIEW	• Technical debt clarification - blocks with audit logs - need to update the `approver` column in the trip request table for the requirement. - DONE ◦ **Audit logs and audit columns are separate things, implement audit columns for the feature.**	DONE
TN-630: As an Employee, I need to receive an SMS when my trip has been allocated to a voyage DONE		DONE Note: **Since there is no subscription bought for SMS service, the SMS's will be tested via emails.**

Issue Dependencies - Sprint 17

Dependency summary

Overall status	NOT-CRITICAL
Total impacted estimated story count/points.	24 / 6
Total impacted not-estimated story count.	3
Blocked story count / points	2
Current velocity in story points	26.4
Dependency clearance due date	Nov 23, 2023
Dependency Score	to-be-decided

Backlog summary

Unplanned (Backlog) story count / points	24 / 97
Current sprint story count / points.	10 / 40
Tentative next sprint story count / points.	9 / 48
`BA IN-PROGRESS` story count.	11
Bugs count.	9
`ToGroom` story count	22

Issues Impacted

Key	Summary	Assignee	Reporter	Status	Frontend Dev.	Backend Dev.	Tester	Story Points
TN-821	[CR] As a logged-in User, I need to view the selected drop location from the map, So that I can identify the location that I have chosen on the location field.	QA	BA	DONE	SM		QA	5.0
TN-815	[CR] As a Transport Coordinator, I need to only see the available vehicles from the vehicles dropdown, So that I can allocate an available vehicle to a new voyage.	BA	BA	DEV - TODO	SM	Thambara Supun Basnayake	QA	5.0
TN-814	[CR] As an Admin, I need to click on the i icon under the actions column, So that I can view more information of the selected role.	SM	BA	DEV - TODO	SM		QA	5.0
TN-726	Build Environment-Specific CI/CD Pipelines for All Components (QA, & Staging)	BA	BA	TOGROOM		dileepa.mabulage		
TN-633	As a Transport Coordinator, I should receive an email when the driver ends the voyage, So that I'm alerted when the voyage ends	BA	BA	TOGROOM	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	
TN-629	As a Transport Coordinator, I should receive an email when a manager approves an	Thambara Supun Basnayake	BA	DEVELOPMENT IN PROGRESS	Thambara Supun Basnayake	Thambara Supun Basnayake	QA2	3.0

	employees request, So that I can allocate them to a voyage.							
TN-628	As an Employee, I need to receive an SMS when a manager approved my request, So that I am alerted about my trips status	Thambara Supun Basnayake	BA	DEV - TODO	Thambara Supun Basnayake	Thambara Supun Basnayake	QA2	3.0
TN-617	[CR] As a Driver, I need to enter the odometer reading, So that I could send it to the Transport Coordinator for him to compare with the image	BA	BA	TOGROOM	Rishimithun Muralidharan		QA2	3.0
TN-600	As a Driver, I need to re-arrange passengers after starting voyage, So that I can view the re-routed map route.	BA	BA	TOGROOM	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	

9 issues

Dependencies

Issue Key	Dependency	Status
TN-821: [CR] As a logged-in User, I need to view the selected drop location from the map, So that I can identify the location that I have chosen on the location field. DONE	• Happy path test cases - Done	NOT-CRITICAL
TN-815: [CR] As a Transport Coordinator, I need to only see the available vehicles from the vehicles dropdown, So that I can allocate an available vehicle to a new voyage. DEV - TODO	• Acceptance criteria - DONE ◦ Vehicle should have a status which indicate the availability of the vehicle. ◦ This status should change when voyage is created and completed. ◦ Vehicle get all endpoint should have a filter to get the vehicles by the status. • Happy path test cases - Done	NOT-CRITICAL
TN-814: [CR] As an Admin, I need to click on the i icon under the actions column, So that I can view more information of the selected role. DEV - TODO	• Acceptance criteria - DONE ◦ Generic user role information popup need not to show all the users. Role description and permissions are enough for the generic user role. ◦ Scroll bar should apply for the permissions section as well. • Figma design - DONE • Happy path test cases - Done	NOT-CRITICAL
TN-726: Build Environment-Specific CI/CD Pipelines for All Components (QA, & Staging) TOGROOM	• Needs approval for cloud resource provisioning for staging environment from PO - BA - PENDING	BLOCKED
TN-633: As a Transport Coordinator, I should receive an email when the driver ends the voyage, So that I'm alerted when the voyage ends TOGROOM	• Acceptance criteria - DONE ◦ Abandoned voyage section should be removed from the scope as that is not implemented in this release.	BLOCKED

	• Depends on the `passenger drop time`. As the mentioned dropped time field is not updated when the passenger is dropped; a separate story is required for that task. - DONE ◦ Jira issue for the new task (TN-826: [CR] As a Driver, I need the passenger details card to be moved to the bottom of the list after dropping them off, So that I can easily identify and prioritize passengers who haven't been dropped off yet. BA IN-PROGRESS)	
TN-629: As a Transport Coordinator, I should receive an email when a manager approves an employees request, So that I can allocate them to a voyage. DEVELOPMENT IN PROGRESS	• Happy path test cases - Done	NOT-CRITICAL
TN-628: As an Employee, I need to receive an SMS when a manager approved my request, So that I am alerted about my trips status DEV - TODO	• Happy path test cases - Done	NOT-CRITICAL
TN-617: [CR] As a Driver, I need to enter the odometer reading, So that I could send it to the Transport Coordinator for him to compare with the image TOGROOM	• Acceptance criteria - DONE ◦ There should be number validation for the odometer reading and the continue button should be disabled. • UI - DONE ◦ Invalid number input design. • Happy path test cases - Done	NOT-CRITICAL
TN-600: As a Driver, I need to re-arrange passengers after starting voyage, So that I can view the re-routed map route. TOGROOM	• Acceptance criteria - DONE ◦ Refresh icon for the update map will not focus once the list rearranged. Propose to add a button popping up in the screen when re-arranged the passenger list. ◦ dropped passengers' map points should not count for the map re-routing. • UI - DONE ◦ Update button design. • Happy path test cases - Done	CRITICAL

Sprint Plans

Sprint 14

Sprint goals:

- [x] Clear 75% of WIP - Developer
- [x] Clear 50% of Mobile stories (based on story points) - Developer / QA
- [x] Fix all Bootstrap styling issues. - Dev / QA
- [x] Complete email notifications system. - Developer2 /
- [x] mplement Routeguard for Manager user. - Developer /
- [ ] Complete 75% of Login & Authorization epic. Developer , /

Sprint summary:

total spillover stories points / count	42.5 sp / 10 - Developer , Developer2
total new stories points/count	13 sp /2 - Developer , Frontend Developer
current velocity	17.5 sp
commitment	55.5 sp (commitment is 13 + spillovers)
dependency status	☑ 1 architecture input dependency. Architect1 ☑ Pending test case reviews. There are 121 test cases pending review and no reviewer assigned yet for web test cases. (@Dilini Dassanayake) ☑ Mobile device purchase (PO forwarded to IT) - PO More information → Sample Issue Dependency Summary Document (From Transpomate)

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Labels
TN-647	As a Manager, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system.	QA	BA	dileepa.mabulage	dileepa.mabulage	QA	8.0	DONE			
TN-638	[CR] As an Admin, I need to check the format of the vehicle license plate (Register Number) in the view, edit and add vehicles pages	QA	BA				2.0	DONE			

TN-626	Bootstrap styles are not being applied properly and causing visibility issues.	QA	QA	Abdul Qadir		QA	5.0	DONE
TN-598	As an admin, I need to search for users, So that I can assign the searched user with a role (assigning the role isn't included)	QA	BA	Sasani Jayathilake	dileepa.mabulage	QA	5.0	DONE
TN-596	As an admin, I need to view all roles, So that I can manage roles and their permissions effectively.	QA	BA	Abdul Qadir	Thambara Supun Basnayake	QA	5.0	DONE
TN-591	As a Transport Coordinator, I need to receive an automated email, So that I know that the voyage has started	QA	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	13.0	DONE
TN-516	As a Driver, I need to navigate back to the passenger list page, So that I can mark the attendance of passengers that I missed.	QA2	BA				0.5	DONE
TN-514	As a Driver, I need to view a description of the voyage in the preview page, So that I can have a clear understanding of the voyage details.	Rishimithun Muralidharan	BA	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	3.0	DONE
TN-464	As an admin, I need to assign roles to individual users, So that the user will be using role-specific features and functionalities of the system.	QA	Lithira.Senath	Sasani Jayathilake	Thambara Supun Basnayake	QA	3.0	DONE
TN-461	As a Driver, I need to verify the attendance of the passengers and mark the checkbox if present, So that I can see the pins on the map for the present passengers	QA2	Lithira.Senath	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0	DONE
TN-346	As a Driver, I want to enter the odometer	QA2	Lithira.Senath	Rishimithun Muralidharan	dileepa.mabulage	QA2	0.5	DONE

	reading, so that I can start the voyage.							
TN-345	As a Driver, I want to take a picture of the odometer, so that I can keep track of the odometer beginning reading before starting the voyage.	QA2	Lithira.Senath	Rishimithun Muralidharan		QA2	5.0	DONE
TN-209	As a Driver, I want to drop trips for employees, So that the mobile app will be notified with each drop and help in preparing the report.	QA2	Thanusan irran	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	5.0	DONE
TN-207	As a Driver, I want to view a list of the passengers, So that I know which passengers have been allocated to the voyage and mark their attendance.	QA2	Thanusan irran	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0	DONE
TN-51	As a Driver I need to view a preview of the voyage route after marking the passengers, So I can get an idea of the route.	QA2	Thanusan irran				8.0	DONE
TN-50	As a Driver, I need to start the trip after marking passenger list	QA2	Thanusan irran	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	8.0	DONE

16 issues

Sprint 15

Sprint goals:

- [x] Complete spilled over mobile + web stories (SC - 4 / SP - 19)- Developer , , Developer
- [x] Achieve 43 SPs - (19 SP (10) spillovers + 24SP (4) new items).

  Completed 33 SPs

  13 SPs out of 19 SPs spilled over completed→ 68% from target

  17 SPs out of 24 SPs new stories completed → 70% from target

  3 SP sprint scope change - 6.9% from the commitment

- [x] Complete R&D task on Power BI and provide a concise documentation about the research on confluence -
- [x] Complete R&D task on Azure Maps and provide a summary of the findings - Developer

Sprint summary:

total spillover stories points / count	19 SP / 4 - Developer, Developer
total new stories points/count	24 SP / 10 - Developer , Frontend Developer , Developer2, Developer
current velocity	26.4 SP
commitment	43 SP (commitment is 24 + spillovers)
dependency status	More information → Issue Dependencies - Sprint 14
burndown	Transpomate New

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Labels
TN-789	[Admin][Assign Role] - The success toast message should not be displayed without assigning users.	QA	QA					DONE			
TN-771	R&D on Google map localization	Rishimithun Muralidharan	SM					DONE			
TN-708	[Permission Guard]- The "Manage Trips" module should not be visible when login as a TC.	QA	QA					DONE			
TN-706	[UI][Allocate Voyage] - The placeholder should be changed in vehicle and time place holders.	QA	QA					DONE			

TN-705	[Allocate Voyage]-The voyage start time is not displaying the accurate time.	QA	QA	DONE
TN-704	[Mobile][Drop employees]-The first passenger drop button is disabled without confirming the drop	QA2	QA2	DONE
TN-703	[Trip History]-The "Trip start date and time" and ""Trip end date and time" should not be the same.	QA	QA	DONE
TN-701	[Mobile][Drop employees]-The next passenger drop button is enabled without confirming the first passenger drop.	QA2	QA2	DONE
TN-700	[Mobile][Drop employees]-The app is not navigating to the drop employees page by clicking yes in the confirmation popup	QA2	QA2	DONE
TN-695	[UI][Manager][Subordinate trip request]-The system is not showing an error message without adding the location	QA2	QA2	DONE
TN-694	[UI][View roles]-The tool tip should be added for all "i" icons and possible icons	QA2	QA2	DONE
TN-693	[UI][All modules]-The tool tip should be added for all "+" icons	QA2	QA2	DONE
TN-690	[UI][Transport coordinator][Scheduled voyages]-The route icon alignment is not proper in the scheduled voyages table	QA2	QA2	DONE
TN-689	[UI][Edit Employee] - The employee is created successfully with an empty cost center field.	QA2	QA	DONE

TN-687	[UI][Transport coordinator][Allocate Voyages]-The alignment of the date field is not proper	QA	QA2					DONE
TN-665	[Transport Coordinator] [Allocate Voyages]-The trip trip requests duplicated while allocating voyage	QA2	QA2					DONE
TN-664	[Transport Coordinator] [Allocate Voyages]- System is not allowing to allocate voyages for more than 8 trip requests	QA2	QA2					DONE
TN-663	[CR] As a Transport Coordinator, I need to receive a list of the absent passengers, So that I know the number of absent passengers when the driver starts the voyage	QA	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	5.0	DONE
TN-659	R&D to convert google API's to Azure Maps	dileepa.mabulage	BA					DONE
TN-657	R&D on Power BI	Abdul Qadir	BA					DONE
TN-654	As a Transport Coordinator, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system.	QA	BA	dileepa.mabulage	dileepa.mabulage	QA	5.0	DONE
TN-653	As a Generic User, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system.	QA	BA				3.0	DONE
TN-652	[Owners][Edit] - An error message is displayed after editing the status.	QA2	QA		Thambara Supun Basnayake	QA	1.0	DONE
TN-649	[Mobile][Drop Employees] - The "Drop" button is enabled in every passenger.	QA2	QA					DONE

TN-647	As a Manager, I need to access only the functions assigned to me by the admin, So that I can do the specific assigned functions within the system.	QA	BA	dileepa.mabulage	dileepa.mabulage	QA	8.0	DONE
TN-619	[CR] As a Driver, I need to see the pending voyage, So that I can plan and prepare for the voyage	QA2	BA	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	3.0	DONE
TN-618	[CR] As an Admin, I need to see clear icons on all the view pages, So that I can quickly know the status	QA	BA	Abdul Qadir		QA	2.0	DONE
TN-613	[CR] As a Driver I need to view the landing page of the mobile app, So that I can log in and view the pending voyages	QA2	BA	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	2.0	DONE
TN-595	As a logged-in user, I need to view the Employee page in different screen sizes.	QA	BA	Sasani Jayathilake		QA	5.0	DONE
TN-464	As an admin, I need to assign roles to individual users, So that the user will be using role-specific features and functionalities of the system.	QA	Lithira.Senath	Sasani Jayathilake	Thambara Supun Basnayake	QA	3.0	DONE
TN-443	API Performance Issue	QA2	Thambara Supun Basnayake			QA2	1.0	DONE
TN-209	As a Driver, I want to drop trips for employees, So that the mobile app will be notified with each drop and help in preparing the report.	QA2	Thanusan irran	Rishimithun Muralidharan	Rishimithun Muralidharan	QA2	5.0	DONE
TN-207	As a Driver, I want to view a list of the passengers, So that I know which passengers have been allocated to the voyage and mark their attendance.	QA2	Thanusan irran	Rishimithun Muralidharan	Thambara Supun Basnayake	QA2	3.0	DONE

33 issues

Sprint 16

Sprint goals:

- [x] Complete spilled over mobile + web stories (SC - 4 / SP - 13)- Developer ,
- [x] Achieve 49 SPs - (13 SP (4) spillovers + 36 SP (8) new items)
  Completed 38 SPs → 78% of total SPs.
  13 SPs out of 13 SPs spilled over completed→ 100% from target.
  25 SPs out of 36 SPs new stories completed → 70% from target.
- [x] Complete R&D task on SMS service comparison and provide a summary of the findings - Developer2 (Story points - 3)

Sprint summary:

total spillover stories points / count	13 SP / 4 - Developer , , Frontend Developer
total new stories points/count	36 SP / 8 - Developer , Developer2 , Developer3 , Developer
current velocity	20.2 SP
commitment	49 SP (commitment is 36 + spillovers)
dependency status	• TN-602 ◦ Acceptance criteria needs to be updated. BA DONE ◦ UI design - BA DONE ▪ `Error page` should be designed to show if the login feature is crashed. ▪ `Not access page` should be designed to show if the user has the access to login or not. ◦ Sequence diagram - Developer DONE • TN-592 ◦ R&D on how to create a secure URL to upload files - Developer DONE ◦ R&D on how to clear the files stored in the server side. Developer PENDING More information → Issue Dependencies - Sprint 15

Spill overs:

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Resolution

No issues found

New tasks:

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Lab
TN-804	[UI][Status] - Toggle alignment should be center.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-793	[Trip History] - The added trip request is not visible on the trip history page.	QA	QA					DONE			
TN-792	[User][Status] - The user can login to the system successfully after being disabled.	QA	QA					DONE			
TN-791	R&D on SMS services	Thambara Supun Basnayake	BA	Thambara Supun Basnayake				DONE			
TN-787	[UI] - The ID column should be removed.	QA2	QA					DONE			
TN-782	[Add Vehicle] - Validation messages should display in particular fields when saving with invalid data.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-781	[Edit Vehicle] - Vehicles should not be saved with zeros in the "no seat" and "cost per km" fields.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-780	[UI][All Modules] - Validation messages are not in the correct capitalization.	QA	QA	Sasani Jayathilake		QA		DONE			
TN-777	[UI][Admin][All Modules]-The gap between all the fields of add employees are not same as figma	QA2	QA2	Sasani Jayathilake		QA2		DONE			
TN-773	[Suggestion][UI][All Modules] - Lack of separation between title and fields.	QA	QA								
TN-692	[UI][Vehicle] - The name of the 7th			c o	lumn is not as figma.						

QA2	Q A	DONE	

TN-674	[All Modules]- The latest data should be visible at the top of the list.	QA	QA					DONE			
TN-673	As an Admin, I need to access only the functions assigned to me, So that I can do the specific assigned functions within the system.	QA	BA	dileepa.mabulage	dileepa.mabulage	QA	8.0	DONE			
TN-671	[UI][Edit Owner] - The error message is not in correct capitalization.	QA	QA					DONE			
TN-669	[UI][View Vehicle]- The column names are not in the correct capitalization.	QA	QA					DONE			
TN-666	[Email] - The email does not contain the accurate timezone	QA	QA					DONE			
TN-627	As a Manager, I need to receive an email, So that I am notified that an employee has requested a trip.	QA	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	3.0	DONE			
TN-612	[CR] As a Transport Coordinator, I need to view all scheduled voyages, So that I can manage and coordinate scheduled voyages effectively	QA	BA	Sasani Jayathilake		QA	1.0	DONE			
TN-602	As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app	Rishimithun Muralidharan	Thanusan irran	Rishimithun Muralidharan	dileepa.mabulage	QA2	8.0	DEVELOPMENT IN PROGRESS		30/Nov/23	
TN-592	As a mobile tester, I need a backend endpoint to return a unique URL to upload files, So that the files can be uploaded securely.	QA2	BA		dileepa.mabulage	QA2	5.0	DONE			
TN-574	[LocalApk][Mobile] [Passenger List]-Check box functions independent of the passenger cards	QA2	QA2					DONE			LocalA

TN-427	As an admin, I want to view the users registered to the system, So that I can view user details.	QA	Thanusan irran	Sasani Jayathilake	Thambara Supun Basnayake	QA	3.0	DONE	
TN-225	As an admin, I want to enable and disable user, So that they are inactive in the system once disabled and active once enabled	QA	Thanusan irran	Abdul Qadir		QA	2.0	DONE	

23 issues

Sprint 17

Sprint goals:

- [ ] Complete spilled over mobile + web stories (SC - 2 / SP - 11)- Developer , Achieve 46 Developer2

SPs - (11 SP (2) spillover + 35 SP (7) new items) - Developer Developer

Supun Basnayake

Completed 4 SPs → 9% of total SPs.

3 SPs out of 11 SPs spilled over completed→ 27% from target. 1 SPs out of 35 SPs new stories completed → 3% from target.

- [x] Identify all the defects in E2E testing stories (SC - 2 / SP - 10).

Sprint summary:

total spillover stories points / count	11 SP / 2 - Developer
total new stories points/count	35 SP / 7 - Developer , Developer2 , Developer
current velocity	20.2 SP
commitment	52 SP (commitment is 36 + spillovers)
dependency status	• TN-724 ◦ Skills on bicep templates - Developer DONE More information → Issue Dependencies - Sprint 16

Spill overs:

Key	Summary	T	Created	Updated	Due	Assignee	Reporter	P	Status	Re
TN-602	As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app		14/Sep/23 10:55 AM	Today 8:10 AM	30/Nov/23	Rishimithun Muralidharan	Thanusan irran		DEVELOPMENT IN PROGRESS	Unre

1 issue

New tasks:

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Lab

TN-811	End to end testing for all modules.	SM	SM	SM	Thambara Supun Basnayake	QA	5.0	DEV - TODO
TN-794	End to end testing for "Voyages" module.	dileepa.mabulage	QA2	SM	Thambara Supun Basnayake	QA	5.0	DEVELOPMENT IN PROGRESS
TN-786	[Manager][Pending Trip Request] - The trip request should not be visible under "Pending" tab when it's added by the manager.	QA2	QA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA		DONE
TN-779	[UI][Add subordinate trip Request]-The Date and time fields are not in the same line as figma	QA2	QA2	SM		QA2		DONE
TN-778	[UI][Add trip Request]-The date and time field sizes are not same as figma	QA2	QA2	SM		QA2		DONE
TN-724	Create Azure Resource Provisioning in Continuous Deployment (CD) Pipeline for All Three Components (Backend, Frontend) - Development Env.	Architect	BA	dileepa.mabulage	dileepa.mabulage	dileepa.mabulage		DONE
TN-702	[Pending Trip Request]-The date is not getting cleared when clicking on the "Clear" button.	QA	QA			QA		DONE
TN-698	[UI][Trip History]-The "Cancel Request" pop-up is not same as the figma.	QA2	QA	Sasani Jayathilake		QA		DONE
TN-696	[UI][Manager][manage trip requests]-The confirmation popup is not as figma	QA2	QA2	Sasani Jayathilake		QA2		DONE
TN-691	[UI][All Modules]-The "Save" and "Reset" buttons are not as Figma.	QA2	QA	Sasani Jayathilake		QA		DONE
TN-688	[All Modules] - The search result is not coming for the employees on the second page.	QA2	QA	SM		QA		DONE
TN-686	Deployment script for all function apps	dileepa.mabulage	SM	dileepa.mabulage	dileepa.mabulage	dileepa.mabulage		ON-HOLD

TN-631	As a Manager, I need to receive an email when my employee's requests have been allocated to a voyage	SM	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	5.0	DEV-CODE REVIEW		30/Nov/23
TN-630	As an Employee, I need to receive an SMS when my trip has been allocated to a voyage	QA	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	8.0	DONE		
TN-615	[CR] As a Driver, I need to see a confirmation message when starting a voyage, So that I could either start the voyage or mark the missed passengers	QA2	BA	Developer		QA2	1.0	DONE		
TN-513	As a Driver, I need to upload the selected image, So that the TC could use the image to compare with the odometer text the driver entered.	QA2	Lithira.Senath	Developer	Developer	QA2	5.0	DONE		

16 issues

Sprint 18

Sprint goals:

- [ ] Complete spilled over mobile + web stories (SC - 7 / SP - 44)- Developer , Developer2 , Developer
- [ ] Achieve 76 SPs - (44 SP (7) spillover + 32 SP (6) new items) - Supun Basnayake ,

Sprint summary:

total spillover stories points / count	44 SP / 7 - Developer , Developer2 , Developer
total new stories points/count	32 SP / 6 - Developer , Developer2 , Developer , SM
current velocity	27.2 SP
commitment	76 SP (commitment is 32 + spillovers)
dependency status	• TN-686 ◦ Depend on TN-630: As an Employee, I need to receive an SMS when my trip has been allocated to a voyage DONE . More information → Issue Dependencies - Sprint 17

Spill overs:

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Lab
TN-794	End to end testing for "Voyages" module.	dileepa.mabulage	QA2	SM	Thambara Supun Basnayake	QA	5.0	DEVELOPMENT IN PROGRESS			
TN-686	Deployment script for all function apps	dileepa.mabulage	SM	dileepa.mabulage	dileepa.mabulage	dileepa.mabulage		ON-HOLD			
TN-631	As a Manager, I need to receive an email when my employee's requests have been allocated to a voyage	SM	BA	Thambara Supun Basnayake	Thambara Supun Basnayake	QA	5.0	DEV-CODE REVIEW		30/Nov/23	
TN-602	As a Driver, I need to validate my email using both Azure authenticator (MS authentication app) and system validator, So that I can log into the Mobile app	Rishimithun Muralidharan	Thanusan irran	Rishimithun Muralidharan	dileepa.mabulage	QA2	8.0	DEVELOPMENT IN PROGRESS		30/Nov/23	

4 issues

New tasks:

Key	Summary	Assignee	Reporter	Frontend Dev.	Backend Dev.	Tester	Story Points	Status	T	Due	Lab

approved my request,

So that I am alerted

about my trips status

Meeting notes & Action items

All backlog grooming notes and documents come here.

Sprint reviews

Sprint Review (14/09/2023) meeting notes & action items

(New story - CR linked to TN-119: As a Transport Coordinator, I want to view all voyages (scheduled, completed, ongoing, and abandoned), So that I can see and manage all voyages effectively. DONE **119)** / Manage Voyages page **[** TN-612: [CR] As a Transport Coordinator, I need to view all scheduled voyages, So that I can manage and coordinate scheduled voyages effectively DONE **]**

- Change the distance to units KMs.
- Add starting odometer reading instead of ending odometer reading.
- below UI changes to be done in voyage information
  - address --> drop location
  - name --> full name
  - date --> remove pipe or keep a space
  - vehicle --> introduce a format to vehicle ids
- Fix UI alignment issues.

<u>Mobile app</u>

New story - CR / Mobile Landing page / **[** TN-613: [CR] As a Driver I need to view the landing page of the mobile app, So that I can log in and view the pending voyages DONE **]**

- Change the logo to the newly designed Transpomate logo. (contact Ajenth from marketing)
- Button names & UI should match Figma.

Pending Voyage - CR / [TN-619: [CR] As a Driver, I need to see the pending voyage, So that I can plan and prepare for the voyage DONE **]**

add below items as Acceptance Criteria to TN-54: As a Mobile User, I want to view all the pending voyages, so that I can stay informed about the scheduled voyages for the day DONE **)**

- Need to add a loader if there is a delay when loading the voyage on the screen.
- In the pending voyages page, change the name to "pending voyage."

Setup a UI review session and explore (Responsible: BA)

- Identify gaps between Figma and the App make changes to the plan to close them. (eg:- Passenger list - alignment issue for the call icon - @Surani Withanage)

New story - CR / Voyage preview page / [TN-614: [CR] As a Driver, I need to view a preview of the voyage before starting the voyage, So that I can view the route and be prepared DEV - TODO **]**

- In the preview page, we should show the *exact location.*
- Remove voyage ID.
- Need to add the re-arrangement in the voyage preview page as well.
- should show the start and end locations in the route preview but there should not be drops (red pins) in those points.

New story - CR / start voyage confirmation. / [TN-615: [CR] As a Driver, I need to see a confirmation message when starting a voyage, So that I could either start the voyage or mark the missed passengers DONE **]**

- In the confirmation (after starting voyage) change the `Back` to a `Close`.

New story - CR / Odometer capture / [TN-616: [CR] As a Driver, I need to take a picture of the odometer reading, So that I could send it to the Transport Coordinator DEV - TODO **]**

- Merge these 2 functions → Take picture and select picture.
  - What should happen is when you take a picture, it should be shown as a preview instead of selecting.

(follow-up with PO if we should remove the existing feature - BA)

New story - CR / Odometer ending / [TN-617: [CR] As a Driver, I need to enter the odometer reading, So that I could send it to the Transport Coordinator for him to compare with the image TOGROOM **]**

- In the odometer ending page, remove the restriction of 5 decimals and need to add decimal values as well.

New story - CR / Change the status columns of the tables to an icon / [TN-618: [CR] As an Admin, I need to see clear icons on all the view pages, So that I can quickly know the status DONE **]**

- In the view pages, we need to show an icon/ toggle switch rather than having a label saying `Enable` or `Disable`, to show the status of the records so that it's easier for the users (admin / manager / TC) to see the status.

Sprint Review (06/10/2023) meeting notes & action items

Web Notes

1. Assign Role pop-up: TN-598: As an admin, I need to search for users, So that I can assign the searched user with a role (assigning the role isn't included) DONE

- Need to add all the users in the pop-up and make it scrollable.
- There should be another button in line with the assign button called apply or save.
- These pop-ups should be the same as other pop-ups in the system (need to make them consistent)

1. Allocate Voyages: TN-560: As a Transport Coordinator, I need to select the start and drop point for the voyage, So that I can plan and view the route of the voyage. DONE

- The date and time should be close together. The order should be Date, Time, Vehicle, Start point, and End point.
- In the voyage allocation/view map pop-up, we must change the origin and destination to the start point and end point. Same as the names given in the fields.
- In all location pop-ups (Except for the pop-up when requesting a trip), "suggested address" should be renamed to just "address".
- There are some UI misalignments. Make it look professional/consistent.

1. Manage Voyages and Manage Trip Requests :

- The color of the tabs should be one color (color as in the stroke color). The selected tab should be shown as highlighted.
- The tabs/ buttons don't look clickable; therefore, we need to change the UI. Need to do a UI revamp and change it to a tab kind of page.
- Search and date fields are not aligned. Need to align them.

1. Email received by TC: TN-591: As a Transport Coordinator, I need to receive an automated email, So that I know that the voyage has started DONE

- The date shown is wrong. The solution given by PO is to keep UTC and the AD should be able to pick the time zone of the recipient and display that time on the email. (This should be done later- possibly phase 2)
- For now, we can show the time the driver started the voyage.
- Everywhere you update a date (request date or voyage date), there must be a time zone stored of the user who did that action.
- When deploying, we need to change the "DoNotReply" part to "DoNotReplyTMSexample.com". There is a task for arm templates - within this task, we need to do this.
- Same thing for our product. Change “http://Client.com " with Project1.
- Asked to check this offline.

Mobile Notes

1. Passenger List: TN-461: As a Driver, I need to verify the attendance of the passengers and mark the checkbox if present, So that I can see the pins on the map for the present passengers DONE

- Make the checkbox a bit small in the mobile app
- Need to change the list to a table format

1. Voyage preview: TN-514: As a Driver, I need to view a description of the voyage in the preview page, So that I can have a clear understanding of the voyage details. DONE

- Need to make the outer box a bit compact and it should be translucent (could reduce the opacity)

- Check with Rishi if there is a scrollbar already implemented. If so, then no need.

Sprint Review (24/10/2023) meeting notes & action items

Web Notes

1. The sign-out button is a bit too big (**Jira Link -** TN-813: As a logged-in user, I need to see the transpomate logo, my logged-in username, and a user profile icon in the header, So that I can access my user profile options with convenience. TOGROOM)
   a. Need to have the sign-out button under a dropdown. This dropdown should be placed on the header.

2. User Profile Page (**Jira Link -** TN-622: As a logged in User, I need to view my user profile, So that I can see and manage my personal information TOGROOM)
   a. Required Fields:
      i. Email
      ii. First and Last Name
      iii. Mobile Number
      iv. Role/s assigned.
      v. Status
   b. All these fields should be non-editable.

3. View User page (**Jira Link -** TN-427: As an admin, I want to view the users registered to the system, So that I can view user details. DONE)
   a. The roles should be shown comma-separated.

4. If a user does not have an employee account, the user should see an “Access Denied” page and the user should contact the admin (Admin should add that employee, and then the employee should log in). A “Create employee account” form is not needed.

Sprint Review (07/11/2023) meeting notes & action items

Web Notes

1. When assigning users in “Roles” page from assign users popup, keep one search option selected. Preferably “First Name” option. (**Jira Link -** TN-822: [CR] As an Admin, I need to have a default search option selected on the "Assign Users" pop-up, So that I can effectively search for and assign users based on the default selected option DELETED)

2. There should be a separate UI to show access denied page like the other status code pages (404 / 500). (**Jira Link -** TN-823: [CR] As a user, I need to view a "Access Denied" page with a user-friendly interface, So that I know that my access is denied. DEV - TODO)
   a. Need a button to redirect to login page.

3. Voyage information page - Future enhancement
   a. Distance metric unit should be configurable as part of the internationalization (mi / km)
   b. Drop location should be fetched from the selected location for the trip request. Currently it’s showing employee address.

4. Create trip request page (**Jira Link -** TN-821: [CR] As a logged-in User, I need to view the selected drop location from the map, So that I can identify the location that I have chosen on the location field. DONE)
   a. The selected location should be visible after closing the map modal. (UI is made. Jira ticket is pending).

Sprint Review (21/11/2023) meeting notes & action items

Web Notes

1. All Manager Mails - Future Enhancements
   a. Need to add a link in every mail to auto approve or reject (a button) (**Jira Link** - To be added)
   b. Need to add a link to navigate to the approve/decline page. (This is for the mail that the manager receives when the employee makes a new trip request) (**Jira Link** - To be added)

2. All emails - (**Jira Link** - To be added)
   a. The time zone should either be added or removed from all the emails.

3. Add Trip Request page - (**Jira Link** - TN-821: [CR] As a logged-in User, I need to view the selected drop location from the map, So that I can identify the location that I have chosen on the location field. DONE)
   a. Location field should be mandatory (The star is missing from the field)

Mobile Notes

1. Driver Login - (**Jira Link** - To be added)
   a. Turn off multifactor authentication. The driver should only log in using the authenticator app.

2. Prepare a document for Ansell for Azure setup. (**Jira Link** - To be added)

Sprint Retrospectives

Sprint Retrospective - Sprint 14 (Day 10)

A sprint retrospective is a type of meeting within the Agile framework, when teams reflect on what went well and what could be improved for their next sprint. Retrospectives are essential to continuously improve your sprint process and ensure key learnings are incorporated for next time.

Things that went well

EM

- The team was more delivery focused. This has led to completing 58 story points within the sprint.
- Acknowledging documents.

Developer

- Workflow between Devs and Qas was good.

Developer

- Appreciating Thambara's work on function apps. Especially given the task was completely on a new technology for the entire team.

QA

- Developers delivered things on time.

Things that didn’t go well

EM

- Not considering the Devops pipelines within the sprint. The unexpected pipeline work has led to spilling over of one task.

Developer2

- Time spent on learning and doing stuff that we weren’t doing before such as dependency clearance, updating due dates and a lot of calls.

Sprint Retrospective - Sprint 15 (Day 10)

A sprint retrospective is a type of meeting within the Agile framework, when teams reflect on what went well and what could be improved for their next sprint. Retrospectives are essential to continuously improve your sprint process and ensure key learnings are incorporated for next time.

Things that went well

EM

- Spilled-over story points counts have been reduced.

BA

- Completed most of the bugs.

QA

- None of the stories were put to on-hold state.

Things that didn’t go well

Developer2

- Some stories were delayed during code review state.
- Need a way to track R&D efforts during the sprint.

Sprint Retrospective - Sprint 16 (Day 10)

A sprint retrospective is a type of meeting within the Agile framework, when teams reflect on what went well and what could be improved for their next sprint. Retrospectives are essential to continuously improve your sprint process and ensure key learnings are incorporated for next time.

Things that didn't go well

Dev Lead

- Code review time delays

EM

- There should be place where best practices for code reviewing and developments has documented.
- There should be a way to track front-end and back-end development in progress.

Development environment

Frontend Web

Backend

Mobile

Framework:

- Flutter (Version: 3.10.6)

Language:

- Dart (Version: 3.0.6)

IDEs used:

- Visual Studio Code (Version: 1.84.0)
- Android Studio (Version: Flamingo 2022.2.1)

Purpose:

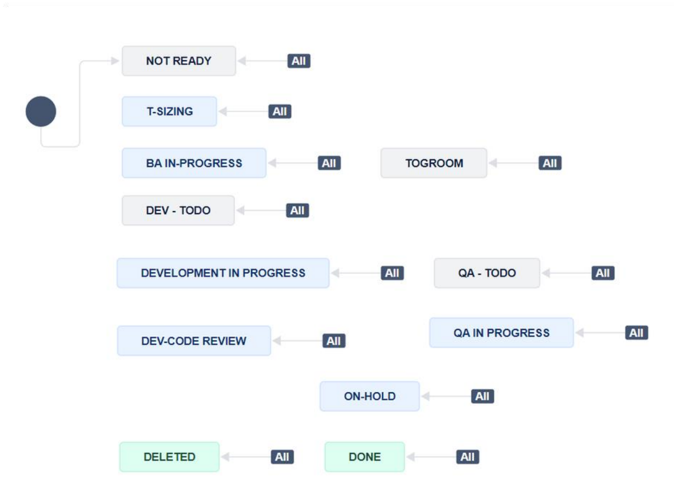
- Visual Studio Code - Lightweight and enables fast application build.
- Android Studio - Can carry out extensive debugging and enables **APK** build for QA release.

Set Up:

- How To Set Up: A Guide

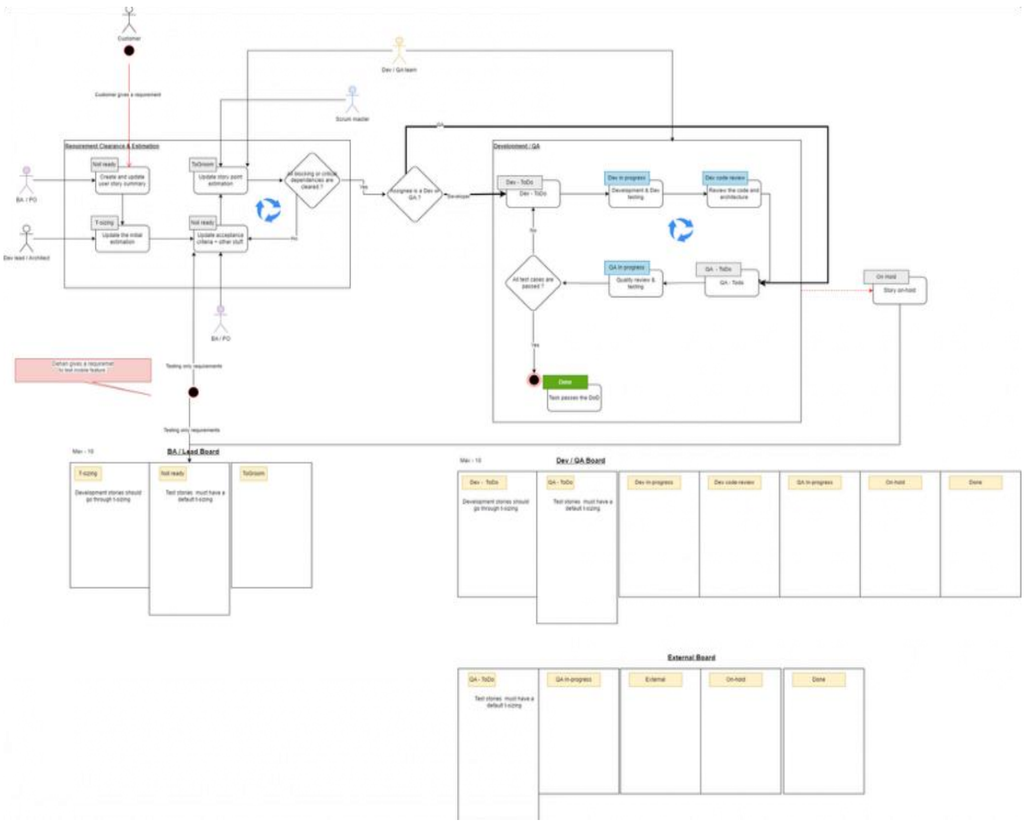
JIRA Configurations

Story / Task / Bug statuses



Board workflows

- We are not restricting transitions on board workflows. Any status can be transitioned to any status. This is to give enough autonomy to the team to get work done. At the same time, status changes are monitored through slack, regular calls and spot checks. These learnings are then taken into account, discussed and addressed in retros.
- There are three types of workflows.
  - BA & Lead workflow (Kanban)- all the work that happens external to the sprint.
  - Scrum workflow (Scrum) - all the work that happens inside a sprint.\
  - External workflows. (Scrum / Kanban) - optional - workflows that happen with third-party companies.



Custom fields

- Frontend Dev - A user selection field
- Backend Dev - A user selection field
- QA owner - A user selection field
- Hold/Delete reason - Selectable options
  - Options available
    - Incomplete Requirements
    - Requirement Change
    - Dependancies
    - Require more R&D
    - Regulatory & Compliance
    - External Disruptions
    - Procurement Delays
    - Personnel unavailability
- Held date - A Date field
- Due date - A Date field
- Estimated Dev Hours - A Time field
- Estimated QA Hours - A Time field
- Errors - A number field

Custom filters

- Team member wise filters (Considering Frontend / Backend / QA owner and Assignee fields)
- User story and bugs

Automations

Import this file into your project:

https://Companysl-my.sharepoint.com/:u:/g/personal/ammar_ameerdeen_Companyglobal_com/EXmiDphIWqFEpWwXVU0mZcEBd2_IeHtIbKkBWam942rQ3w?e=0bKxoh Connect your OneDrive account

1	TP - Assign - Backend development to Thambara
2	TP - Assign - Devops development to Dileepa
3	TP - Assign - Mobile Frontend development to RIshi
4	TP - Assign - Web Frontend development to SM
5	TP - Daily - Due date & hold status check
6	TP - Daily - Management pulse checks
7	TP - Dev Review Assigning
8	TP - Epic fix version update applies to all stories under the epic
9	TP - Epic & Story default allocation is to BA
10	TP - Issue status change -> clear due date
11	TP - Mid-sprint acceptance criteria changes -> Send to Lead Review
12	TP - On-Hold Issue Transitions Rule
13	TP - On-hold status to other status transition
14	TP - QA Assigning
15	TP - Sprint started -> Sprint 0 checklist
16	TP - Status transition to Delete -> Story point clearing
17	TP - When all stories are completed → then close epic
18	TP - When initial estimation is updated -> Move it to BA WIP
19	TP - When issue is added to an epic -> Fetch and update the fix version of the issue from the epic
20	TP - When moved to Dev in progress from Code review -> Assign it back to the original developer
21	TP - When parent is QA in progress → move all sub-tasks to QA in progress

Backlog configs

Apply a filter to filter out stuff the development team shouldn't focus on while the sprint is running. These items are captured under the NOT READY status.

JIRA - How to

This document explains a few rules we've setup when it comes to using Jira in your day-to-day work.

Due dates

Why due dates.?

- Helps you set clear goals for yourself and achieve them.
- Helps other team members working on the same task to plan their day.
- Helps leads identify gaps in the flow of work and address it upfront.
- Helps us keep delivery timelines and project plan up-to date.

A few guidelines to follow on this.

- The due dates are assignee's responsibility. The due date you've mentioned in a work-item should reflect the reality. Not an overestimation or even an underestimation. Try to be as realistic as possible.
- The expectation is that you'll be able to complete the work from your side and transition it to another team member within EOD the date mentioned in the task.
    - If not, you should update the date by EOD. This change should go with your daily update in the DAILY SCRUM
- Dates are checked every morning automatically. SM should verify this in the DAILY SCRUM

Comments

A few guidelines to follow on this.

- Be polite and professional.
- keep it crystal clear and on point. Avoid being vague. (eg:- "working on this", "there are technical issues.."). We are a tech team - we speak the same lingo.

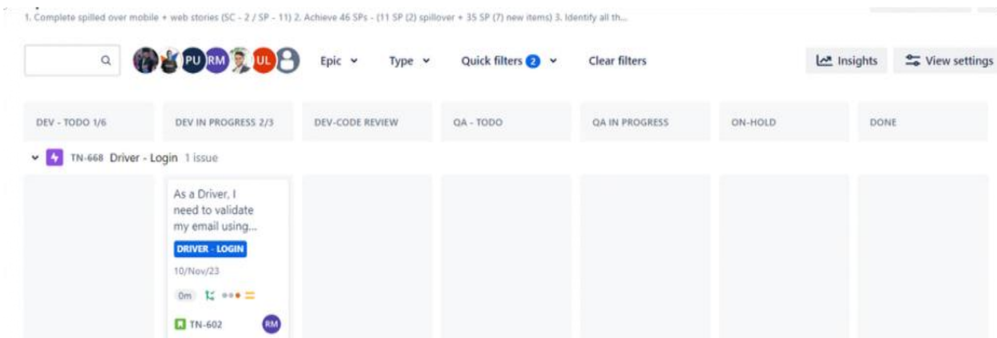
Boards

-
-

Scrum Board

Color codes

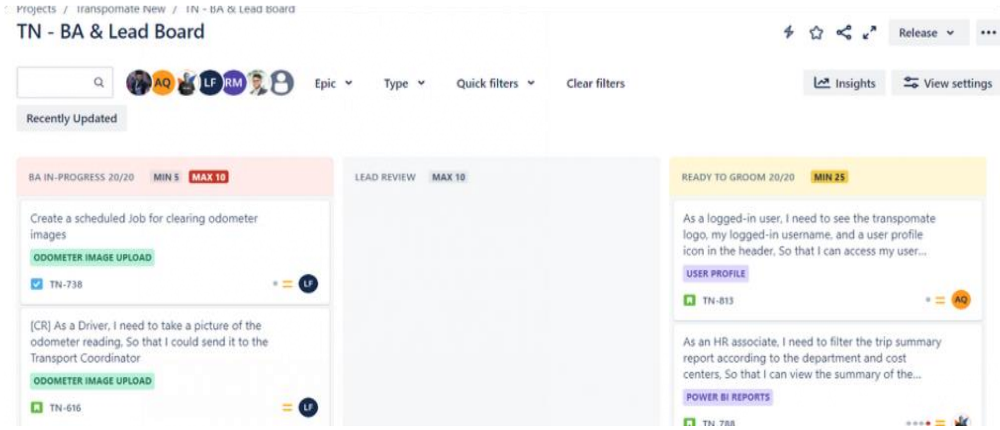
Quick filters



BA & Lead board

Color codes

Filters



Project Management

9 areas of project management

Integration

Scope

- Requirement collection - BA
- Define scope - BRD
- Create epics and stories. - BA
- Verify - BRD verification, Backlog Grooming for user story and epic verification.
- Control scope - Change management

Time

- Estimate stories - T-sizing, Backlog Grooming
- Sequencing - Release planning & review
- Resource requirement estimation - Allocation planning
- Develop a schedule - Release planning & review
- Control schedule - Release planning & review

Cost

- Estimate costs - Need to define
- Determine budget - Need to define
- Control costs - Need to define

Quality

- Need to define

Human Resources

- Resource plan - Allocation planning
- Hire - Need to define an interviewing flow
- Setup a project team - Allocation planning
- Manage the team - Agile & Scrum, Evaluation & Feedback Framework IN PROGRESS

Communication

- Identify stakeholders - Sprint 0 - Stakeholder documentation
- Plan communications - Sprint 0 - Need to define
- Distribute information - All comms happen on Jira and Confluence. Some on teams. Some happen on emails.
- Expectation management - Need to define how to set buffers.
- Report performance - Sprint Reviews

- Plan risk management - Biweekly Dependency Reviews
- Identify risks - Dependency reviews (Needs improvemets)
- Perform qualitative and quantitative risk analysis. - Dependency score.? RCI.? Need to define
- Plan risk response - Need to define
- Monitor and control risks - Dependency Review signoffs. Need improvements

Procurement

- Plan procurement - Define process to raise procurement requests.
- Conduct procurement - Delegated to IT. Need to coordinate.
- Administer procurement - Delegated to IT. Need to coordinate.
- Close procurement - Define a process.

Allocation & release management

Allocations and releases are managed in different stages of the project execution. The stages are,

1. Sprint 0 - LEAD BA TPM
2. BA & Lead cycle. - LEAD BA TPM
3. Self-review (30 mins)- DEV TEAM QA TEAM (Individually, offline)
4. Backlog-review & Dependency review (30 mins)- TPM DEV TEAM QA TEAM LEAD
5. Allocation planning session (30mins)- TPM DEV TEAM QA TEAM LEAD
6. Release planning session (1hr)- TPM LEAD
7. Sprint review (30 mins) - TPM DEV TEAM QA TEAM LEAD PO

These processes add (per sprint),

- 30 mins of overall overhead for the PO.
- 30mins of overall overhead for individual scrum team members for offline work.
- 1.5hrs of overall overhead for the entire team time spent in meetings.
- 2.5hrs of overall overhead for the lead and BA.

Sprint 0 and then in **BA & Lead cycle**

participants : LEAD BA TPM

- Estimations - Tsize estimations for non-estimated stories during the BA & Lead phase.
- Technology stack selected for all the stories during the BA & Lead phase.
  - Automated assignments happen when technology stacks are selected.
- Release timelines
  - Release is set as Jira fixed versions.
  - There has to be 2 fixed versions/release. Set during Checklist - Sprint 0 . This release strategy is bound to the project execution model. Depending on the execution model the decision to have a UAT release is made.
    - UAT release - has a soft deadline. OPTIONAL
      - This deadline is moved forward / backward at the end of every release planning session. UAT release
      - Allowed to move this deadline each sprint by x number of weeks. (x = contingency weeks / planned number of sprints, contingency weeks = total weeks communicated to the customer- planned weeks).
    - Production release - has a hard deadline. Production release
- Lead and BA adds new stories with default allocations to the release on the first day of the sprint.
- Epics
  - An epic is a chunk of scope that is immutable after moving it to a Todo status (DEV TODO or QA TODO).
  - A release must have all the epics.
  - An epic is considered final when its status changed to a Todo status. After that there shouldn't be new story additions to it. To edit the epic, it has to go through the epic workflow.

Self-review

participants : DEV TEAM QA TEAM

- Each team member goes through their allocation plan against the active release.

Backlog review & Dependency review

participants: TPM DEV TEAM QA TEAM LEAD

- Prioritized backlog from the backlog review.
- Critical task dependencies cleared during dependency closure.
- Estimations: - Story point estimates for estimated stories during BR.

time: 4hrs

Allocation Planning (second part of the Sprint planning)

participants: TPM DEV TEAM QA TEAM LEAD

- Start off with a clear release goal statement by the Delivery lead.
- Do pending allocations for the epic stories + redo the allocations if necessary (because the default allocation was done automatically).
- Clear all issues from the allocation plans highlighted in excel. - Team members

time: 1hr

Release planning session:

participants: TPM LEAD

- Start off with a clear release goal statement by the Delivery lead.
- When these sprint allocations happen on JIRA, the excel should get updated automatically.
- Move the release timeline according to the allocation plan if allowed (see Sprint 0 → Release timelines).
- Identify risks and risk mitigation strategies.
- Publish the outcomes in the form of a Release plan document in confluence and communicate it to the PO with the team in cc.

Sprint review

participants: TPM DEV TEAM QA TEAM LEAD PO

- Take the release plan changes up in the next sprint review discussion.
  - Discuss the release goal.
  - Discuss the release content (epics).
  - Discuss the timeline.
  - Discuss risks and risk mitigation strategies.

Mid sprint scope change workflow

- Automation done to change status to t-sizing. Later will rename to Lead review.
- Lead adds makes a decision.
- If the change changes the sprint delivery significantly. raise a triage.
- Or, change it back to where it was.

TPM reports

Report	Sample	Due
Sprint Plan	Sprint plan sample (From Transpomate)	Sprint Day 10 - EOD
Dependency Summary	Sample Issue Dependency Summary Document (From Transpomate)	The Day after BR (Typically falls between Day 7 - 9) - EOD
Allocation planning	Allocation plan for transpomate.xlsx (sharepoint.com)	Day 4 - EOD

Scrum reports

- Sprint Burndown
- Cumulative Flow Diagram (CFD)

Onboarding checklists

Onboarding - HPMO

Objective	Task
Access	☐ Email ☐ Jira - Software ☐ Jira - Confluence ☐ SharePoint - Technology projects ☐ Capacity planning sheet ☐ Allocation planning sheet

Project KTs	☐ Go through all the epics of all projects.
	☐ Demo of all products.
	  ☐ BA Demo
	  ☐ Developer & QA Demo

Process KTs	☐ Go through all the process documents.
	☐ KT - SharePoint/Confluence Structures
	☐ KT - Project Governance
	☐ KT- Invoicing
	☐ Project Profitability Metrics
	☐ Change Management.
	☐ Scrum practice.
	  ☐ The ceremonies.
	  ☐ Story points and velocities.
	☐ Lean/Agile transformation.

Stakeholder intros.	☐ Get introduced to all the stakeholders of projects.
	  ☐ Teams
	  ☐ POs
	  ☐ SMEs
	☐ Get access to all the channels of communication to stakeholders. (Teams. / WhatsApp / Email)

Technologies.	☐ The list of GitHub repositories and what they consist of. - Done in the developer demo
	☐ Brief KT on technologies in use.
	  ☐ Frontend
	  ☐ Backend
	  ☐ Cloud resources.

Team	☐ 1-1s with all the team members.

Onboarding - TPM

Objective	Task
Access	☐ Email ☐ Jira - Software ☐ Jira - Confluence ☐ Sharepoint - Technology projects
Understand the project.	☐ Go through all the remaining stories of the project. ☐ Go through the product. ☐ Demo the product.

Understand the processes.	☐ Go through all the process documents.
	☐ Understand the scrum practice.
	  ☐ The ceremonies.
	  ☐ Story points and velocity.
	☐ Understand lean/agile thinking.
	☐ Suggest a process improvement following lean/agile principles implementable with scrum.

Understand the stakeholders.	☐ Get introduced to all the stakeholders of the project.
	  ☐ Team
	  ☐ PO
	  ☐ SMEs
	☐ Get access to all the channels of communication to stakeholders. (Teams. / WhatsApp / Email)

Understand technologies.	☐ Get to know the list of GitHub repositories and what they consist of.
	☐ Have a brief understanding of the technologies in use.
	  ☐ Frontend
	  ☐ Backend
	  ☐ Cloud resources.

Understand the team	☐ Have 1-1s with all the team members.

Onboarding - QA

Objective	Task
Understand the project.	☐ Go through all the remaining stories of the project. ☐ Go through the product. ☐ Demo the product to the team.
Understand the processes. (General)	☐ Go through all the process documents. ☐ Understand the scrum practice. ☐ The ceremonies. ☐ Story points and velocity. ☐ Understand lean/agile thinking.
Understand the processes. (QA specific)	☐ Understand Test Case Driven Development (TCDD) ☐ Know how to work with AIO.

Deliverable understanding & KPIs	☐ Understand QA KPIs.
	☐ Understand QA checklists.
	  ☐ API Testing checklist
	  ☐ Web Testing checklist.
	  ☐ Mobile checklist
	  ☐ Security checklist - Pending to be added to confluence ()
	  ☐ UX checklist
	  ☐ Nonfunctional requirements checklist
	☐ Test Automation
	  ☐ Get skilled in Web Automation
	  ☐ Get skilled in API Automation.

Understand the stakeholders.	☐ Stakeholder intro. ☐ Team ☐ Get access to all the channels of communication to stakeholders. (Teams. / WhatsApp / Email)

Understand technologies.	☐ Get to know the list of GitHub repositories and what they consist of.
	☐ Have a brief understanding of the technologies in use.
	  ☐ Frontend
	  ☐ Backend
	  ☐ Cloud resources.

Onboarding - PO

Objective	Task

Understand the processes. (General)	- [ ] Process training materials focused on PO activities.
	  - [ ] Sprint reviews.
	  - [ ] Backlog reviews.
	  - [ ] Sprint planning.
	- [ ] Understand the scrum practice.
	  - [ ] What is a sprint.?
	  - [ ] What are story points and velocity.?
	- [ ] How do we prioritize work and plan releases.

How requirements work.	- [ ] Importance of having clear requirements.
	- [ ] Scope changes, timeline changes and causes.
	  - [ ] Technology choice changes.
	  - [ ] Scope creep.

Understand how QA works.	- [ ] What QA engineers do and what they don't.
Understand how BA works.	- [ ] What BAs do and what they don't.
Team introduction	- [ ] BA - [ ] TPM - [ ] your scrum Team.

Onboarding - BA

Onboarding BA to Company

Objective	Task
Deliverable understanding & KPIs	BA Basic Training Business Analysis Training
Understand the processes. (General)	Go through all the process documents. Understand the scrum practice. The ceremonies. Story points and velocity. Understand lean/agile thinking.
Understand the processes. (BA specific)	Understand BA & Lead board activities. Understand scrum board activities. Understand the BA workflow.

Onboarding BA to Project

Objective	Task
Understand the stakeholders.	Stakeholder intro. Team, TPM PO SMEs Get access to all the channels of communication to stakeholders. (Teams. / WhatsApp / Email)
Project Essentials	Get project folder access. Get Jira Access
Understand the project.	Go through BRD and UI's Go through all the remaining stories of the project. Go through the product. Demo the product to the team.
Understand technologies.	Have a brief understanding of the technologies in use. Frontend Backend Cloud resources.

Onboarding - Dev

Objective	Task
Understand the project.	☐ Go through all the remaining stories of the project. ☐ Go through the product. ☐ Product intro by a QA, BA, Dev and PM. (Architect1) ☐ Development environment setup. ☐ Demo the product to the team.
Understand the processes. (General)	☐ Go through all the process documents. ☐ Understand the scrum practice. ☐ The ceremonies. ☐ Story points and velocity. ☐ Understand lean/agile thinking.
Understand the processes. (Dev specific)	☐ Understand Test Case Driven Development (TCDD) ☐ Know how to work with AIO. ☐ Understand dev related board activities.
Deliverable understanding & KPIs	☐ Understand the git & git branching strategy. ☐ Understand Development Quality metrics. ☐ Understand the code-quality standards. ☐ Understand the versioning strategy. ☐ Understand code-review and merging process. ☐ Understand security standards and how to maintain them in development. ☐ Understand how to create and resolve a pull request. ☐ Understand Development KPIs.
Understand the stakeholders.	☐ Stakeholder intro. ☐ Team ☐ Get access to all the channels of communication to stakeholders. (Teams. / WhatsApp / Email)

Understand technologies.	☐ Get to know the list of GitHub repositories and what they consist of.
	☐ Have a brief understanding of the technologies in use.
	  ☐ Frontend
	  ☐ Backend
	  ☐ Cloud resources.

CI-CT-CD

Continuous Integration

Continous Testing

Continous Deployment

PMO & EM

The Project Management Office (PMO) and Engineering Manager play distinct but interrelated roles in an organization. Their collaboration is essential to ensure that projects are executed efficiently and effectively. Here's how they should work together:

1. Clear Communication:

- Regular and open communication is key. The PMO should keep the Engineering Manager informed about project statuses, timelines, and any changes in project priorities or scope.
- The Engineering Manager should provide feedback on the feasibility and resourcing of projects, helping the PMO understand the technical constraints and requirements.

2. Project Planning and Prioritization:

- The PMO is responsible for portfolio management and project prioritization. They should work closely with the Engineering Manager to understand the engineering team's capacity and constraints.
- The Engineering Manager should help prioritize projects based on technical complexity, resource availability, and business impact.

3. Resource Allocation:

- The Engineering Manager should work with the PMO to allocate the right resources for each project. This involves understanding the skills and availability of engineering team members.
- The PMO should consider this information when making resource allocation decisions and ensure that projects are adequately staffed.

4. Project Monitoring and Reporting:

- The PMO is responsible for tracking project progress, identifying issues, and providing regular project reports. The Engineering Manager should collaborate by giving technical insights and updates.
- The Engineering Manager should provide technical input and feedback on project status, ensuring that any technical challenges are addressed promptly.

5. Risk Management:

- The PMO and Engineering Manager should collaborate on identifying and mitigating project risks. The Engineering Manager's technical expertise is crucial in assessing technical risks.
- The PMO should help prioritize risk mitigation efforts and communicate them to the relevant stakeholders.

6. Change Management:

- When project scope changes or new requirements arise, the PMO and Engineering Manager need to work together to assess the impact on timelines and resources.
- They should jointly determine the feasibility of requested changes and communicate the implications to stakeholders.

7. Quality Assurance:

- The Engineering Manager is responsible for ensuring the quality of the engineering work. The PMO can support by defining quality standards and ensuring that they align with the project objectives.
- The PMO should also facilitate reviews and audits to ensure that quality is maintained.

8. Feedback Loop:

- The PMO and Engineering Manager should establish a feedback loop for continuous improvement. They can meet regularly to discuss what's working well and what can be improved in their collaboration and project management processes.

In summary, the PMO and Engineering Manager should work collaboratively to ensure that projects are planned, executed, and monitored effectively. This collaboration helps align technical and project management aspects, resulting in successful project delivery. Open and transparent communication is essential for their partnership to be successful.

Releases

UAT release

Production release

Epics

Self review checklist

- Allocations
- Dependencies

Tsizing

Lead participates in all BA/PO meetings. So it's possible for them to do a 1 minute estimation for a story.

Business Analysis (BA)

BA - Tools

- Proficient in spreadsheets and excel based analytics and visualization.
- Proficient in MS Word and Confluence.
- Possesses basic Power BI skills.
- Possesses basic SQL and SharePoint skills.
- Proficient in using UI mockups (wireframes) to visually communicate information.
-

Diagrams

- Uses diagrams for analysis and communications effectively.

o Flow charts.

o DFDs (Data flow diagram)

o PFDs (Process flow diagram)

o Use case diagram

o Entity relationship diagram.

o Activity diagram.

o Business Domain Models.

Documents

- pre-sales documents.
  - SOW
  - Contracts
  - Proposals
  - PoCs
  - RFP

Business Requirement Document (BRD)

Requirement elicitation

Techniques

Brainstorming.

Document Analysis

Focus Group

Interface Analysis

Interviews

Observation

Prototyping

Workshops

Acceptance Criteria

Functional Acceptance Criteria:

-

Non-functional Acceptance Criteria:

- Non Functional Requirements (NFRs)

eg:-

- Page loads within 1 second under normal network conditions on a device that has 4GB remaining RAM.
- Images load within 2 seconds given the image that load is compressed enough and the server responds fast enough.

Meetings

- Scrum Ceremonies
- Solutioning meetings.

Meeting note-taking

The meeting note taker should always be a passive-key attendant.

Test cycles & Case creation & execution

- Naming convention
- creating test cases
  - Writes “Happy path test cases” when the task is in ready groom status and assigned to you.
  - Write test cases to stories while the task is in sprint.
  - Review Test cases be written against the story by developers.
  - Happy path test cases should be sufficient for the first cycle of development.
- When executing test cases
  a. if there is any failed scenario then specific test step of relevant test case should be failed by attaching the actual result
  b. Once full cycle is done with execution then user story should be moved to Dev In Progress by assigning to relevant developer
  c. Once the fix done by developer then dev has run test cases and proceed the test
  d. Dev has add the PR to user story with view access to QA also add the comment by mentioning what’s fix done and impacted areas
  e. Once the user story move to ready to QA with the Fix
    i. QA has to view the PR and get an idea of any logic changes
    ii. QA has to check the fix mentioned in comment and the impacted areas
    iii. Run another cycle for bug fixes and attach the cycle to same story
- When anyone in the project opens the User story the all the test cases and executed cycles should be visible
- Proper testing coverage
- Follows the AIO folder structure standard. AIO Folder structure
-

QA - Tools

- AIO
- Test Automation Framework
- Load testing tools
- Security testing tools.

Code review

Code review turnaround time - 24 hrs. Default due dates per transition

- Missing business requirements
- Code quality improvements.

Code comments

Technical debt management

- Technical debt management.
- Technical debt review.