AGILE WAY OF WORKING Q2/2025
20 MAY 2025

Internal Use Only

AGENDA

TOPIC	SPEAKER
• WHAT IS AGILE?	K. JUTARAT
• PROJECT LIFECYCLE IN RELATIONSHIP WITH CHANGE MANAGEMENT PROCESS	K. KLINCHABA
• AGILE ORGANIZATION AND KEY ROLES	K. JUTARAT
• AGILE CEREMONIES	K. WARANGKANA
• RISK MANAGEMENT IN AGILE	K. SORRAPONG
• AGILE REPORTING	K. JUTARAT
• AGILE DELIVERABLES	K. KLINCHABA

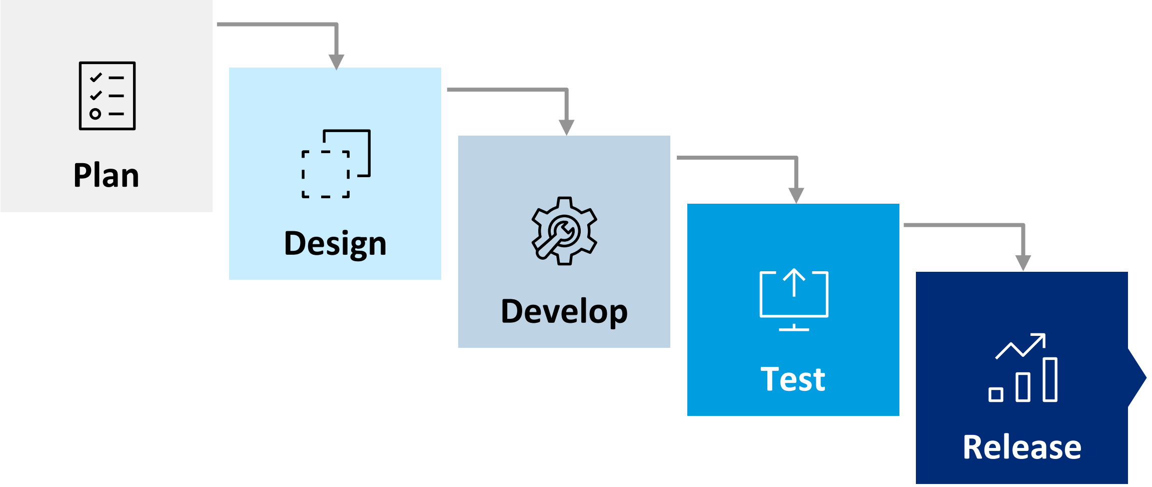
WHAT IS AGILE?

PMM 2.0 WATERFALL VS AGILE

Agile is different due to its iterative approach, allowing for earlier and frequent releases focusing on the highest priorities

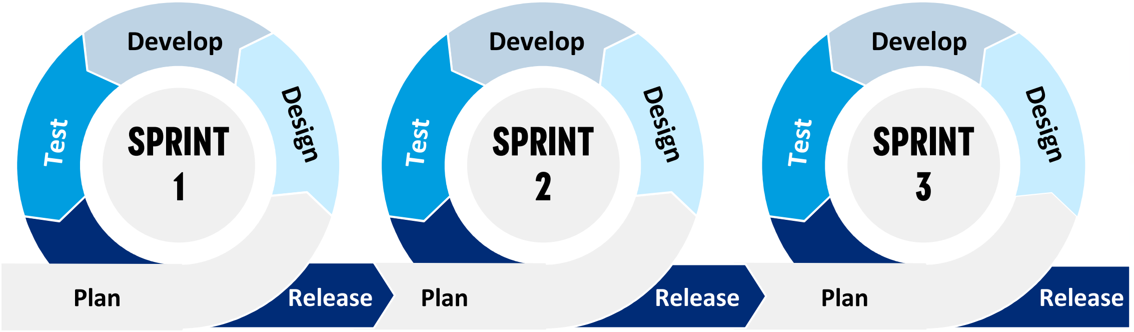
Waterfall is a series of phases separated with stage gates

SDLC[1] stages



Agile consists of iterative development cycles, allowing customers to provide input as quickly as possible

SDLC stages



- PMM 2.0 Waterfall is a linear, sequential approach where each phase must be completed before moving to the next, while agile is iterative and flexible, allowing for continuous feedback, adaptation, and incremental delivery.
- PMM 2.0 Waterfall emphasizes planning and predictability, whereas agile focuses on adaptability and collaboration.

THE AGILE MINDSET

Agile way of working challenges traditional mindset and organizational structure in favor of productivity outcomes

Traditional Structures		Agile Organizations	
Getting things started	≠	Getting things done	• Speed and customer centricity are seen as key competitive advantage • Focus on fewer in-flight activities
Being on a committee	≠	Being the decision-maker	• Competencies and authorities for decisions are attached to individual roles, rather than management functions
Having a plan with due dates and dollars	≠	Having certainty	• “Fail early, fail often” philosophy; Errors seen as investments for continuous improvements • Estimates seen as guesses that may change
Pooled, shared resources	≠	Maximum productivity	• Agile team members are 100% dedicated to a single effort to avoid “context switching” costs
Comprehensive documentation	≠	Shared understanding	• Agile relies on in-person dialogue and joint working sessions to replace written documentation

KEY AGILE CONCEPTS

with Agile ceremonies, Agile organization, Agile tools/Work items, Agile Enablers and others.

SCRUM FRAMEWORK

Agile Ceremonies

Sprint Planning
Meeting where the team selects and commits to work from Product Backlog that will be completed during Sprint

Daily Standup
A short, daily meeting where team members discuss progress, next steps and key blockers

Backlog Refinement
Meeting for adjusting the Backlog to ensure it contains the detailed and prioritized stories for future Sprints

Sprint Review
Meeting where the team demonstrates the work they have completed for review and acceptance

Sprint Retro
Meeting where the team reflects on what went well, what didn't, and how they can improve in the next Sprint

Scrum of Scrums
Meeting to co-ordinate work across dependencies or multiple scrum teams

Agile Organisation

Product Owner
Responsible for defining the features of the product and prioritizing the Backlog. They represent the stakeholders and the customer

Scrum Master
Responsible for ensuring the team follows Agile practices and removes any impediments that may hinder the team's progress

Agile Champion
Group who understand both the agile practices in the organization and continue to advocate to adopting and scaling Agile

Agile Coach
Responsible for guiding organization through the adoption and implementation of Agile; and improving and scaling Agile processes

Agile Tools / Work Items

Product Backlog
Prioritized list of work items for the Delivery team derived from the roadmap and user requirements

Epic
A large body of work that captures significant functionalities that are too big to be completed within a single increment

Feature
A usable, valuable piece of functionality that will allow a user to complete an activity in the real world using the software

User Story
The smallest item in the Product Backlog that delivers usable functionality to users which can be completed within a single Sprint

Agile Enablers

DevOps
A set of practices that combines software development (Dev) and IT operations (Ops) to shorten the system development life cycle and provide continuous delivery

CI/CD Pipeline
Automated sequence of steps to build, test, and deploy software. Continuous Integration (CI) ensures that code changes are automatically tested and merged, while Continuous Delivery (CD) automates the deployment of code to production environments, ensuring rapid and reliable software releases

Others

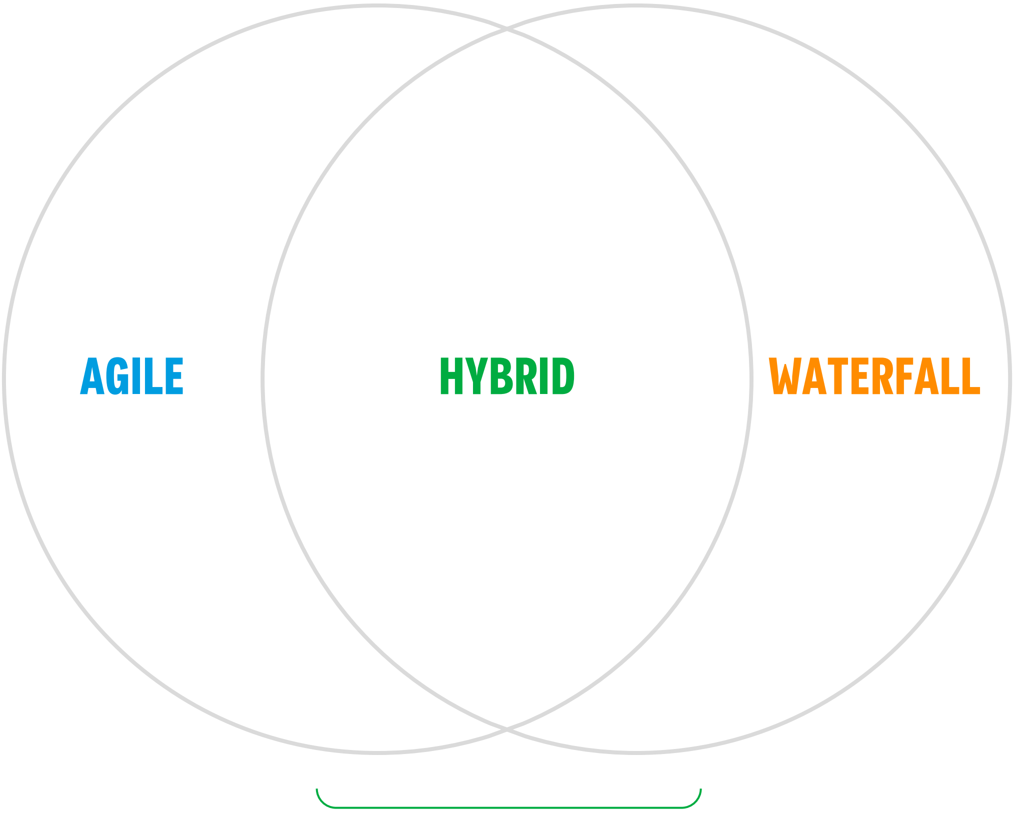
Guard Rails
Guidelines that help teams stay on track and make decisions that align with overall goals of the org., as opposed to, rigid checkpoints or approval gateways

PROJECT LIFECYCLE IN RELATIONSHIP WITH CHANGE MANAGEMENT PROCESS

AGILE FIRST PHILOSOPHY

Project characteristics *most* suitable for Agile

- Client/user feedback is constant and highly valued
- The nature of the features to be delivered allow for incremental release
- Flexibility of timeline and features for prioritization while requirements are continuously discovered and refined
- Delivering value to market as early as possible is crucial
- Experimental approach needed in product development



AGILE

HYBRID

WATERFALL

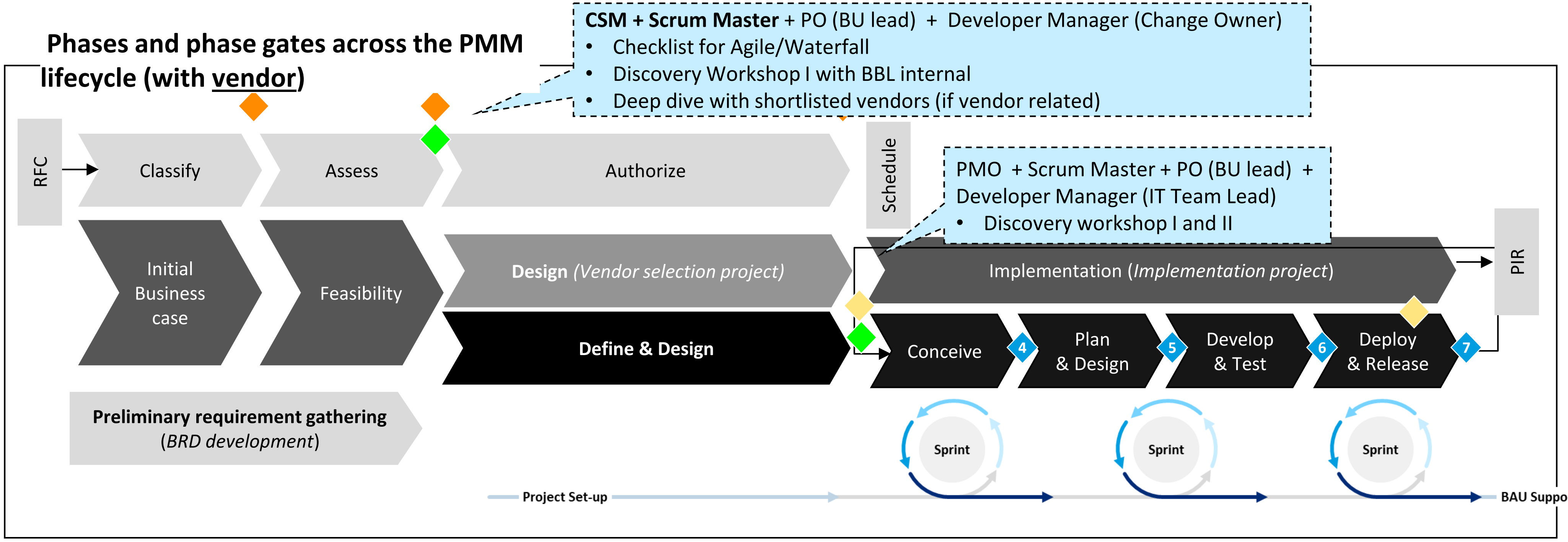
Projects can also be done part Agile, part Waterfall (i.e. Hybrid)

Project characteristics *less* suitable for Agile

- Fixed project timeline, e.g. hard target deadline
- Highly-controlled budget and scope (with certainty on requirements)
- Projects with targeted and specific outcomes, e.g.
  - Regulatory-related projects with clear “asks”
  - Remediation type projects
  - Re-platforming for better efficiency (no change in operating model or function)

While most projects can be delivered using either methodology, BBL’s goal (*once enough trained staff*) should be to adopt an **AGILE FIRST** approach; certain elements can still be run in waterfall if required (e.g. one consolidated release instead of iterative releases)

PROJECT LIFECYCLE IN RELATIONSHIP WITH CHANGE MANAGEMENT PROCESS - AGILE



Phases and phase gates across the PMM lifecycle (with vendor)

CSM + Scrum Master + PO (BU lead) + Developer Manager (Change Owner)
- Checklist for Agile/Waterfall
- Discovery Workshop I with BBL internal
- Deep dive with shortlisted vendors (if vendor related)

RFC → Classify → Assess → Authorize → Schedule

Initial Business case → Feasibility → **Design** *(Vendor selection project)* → Implementation (*Implementation project*) → PIR

Define & Design → Conceive → 4 → Plan & Design → 5 → Develop & Test → 6 → Deploy & Release → 7

PMO + Scrum Master + PO (BU lead) + Developer Manager (IT Team Lead)
- Discovery workshop I and II

Preliminary requirement gathering (*BRD development*)

Sprint

Sprint

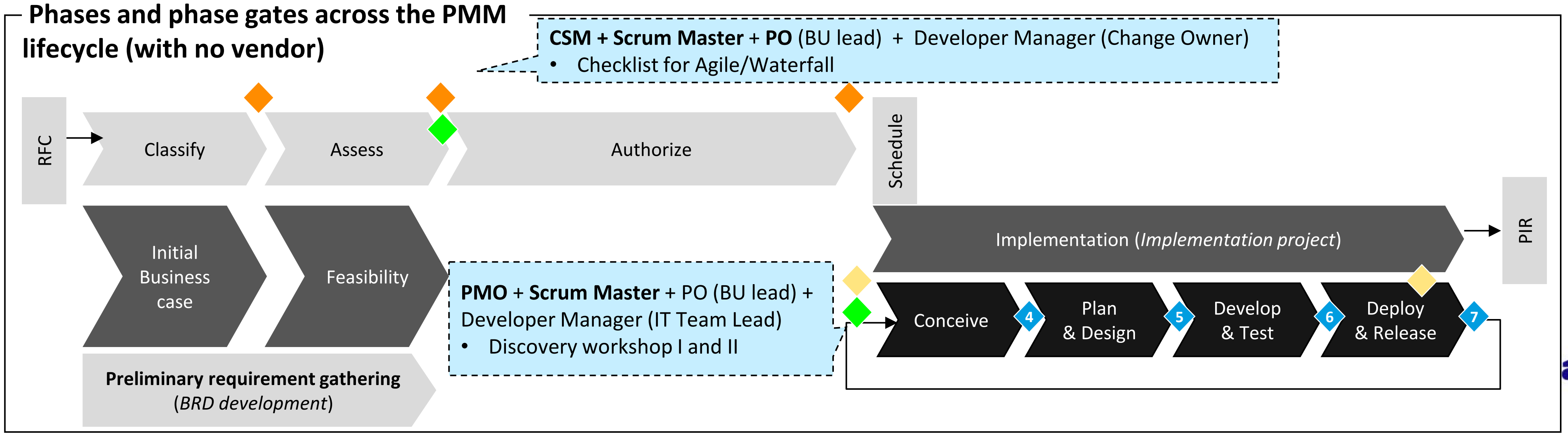
Sprint

Project Set-up

BAU Support

Agile Stage Gates are managed within Agile (scrum team), typically by Product Owner

Legend:
- ▷ CMP Phase
- ▶ ITSC Stages
- ▶ Agile Phases
- 🔶 ITSC Stage Gate
- 🔷 Agile Stage Gates
- ◆ CAB Approval Gate
- 🟩 Agile Checklist



Phases and phase gates across the PMM lifecycle (with no vendor)

CSM + Scrum Master + **PO** (BU lead) + Developer Manager (Change Owner)
- Checklist for Agile/Waterfall

RFC → Classify → Assess → Authorize → Schedule

Initial Business case → Feasibility → Implementation (*Implementation project*) → PIR

PMO + **Scrum Master** + PO (BU lead) + Developer Manager (IT Team Lead)
- Discovery workshop I and II

Conceive → 4 → Plan & Design → 5 → Develop & Test → 6 → Deploy & Release → 7

Preliminary requirement gathering (*BRD development*)

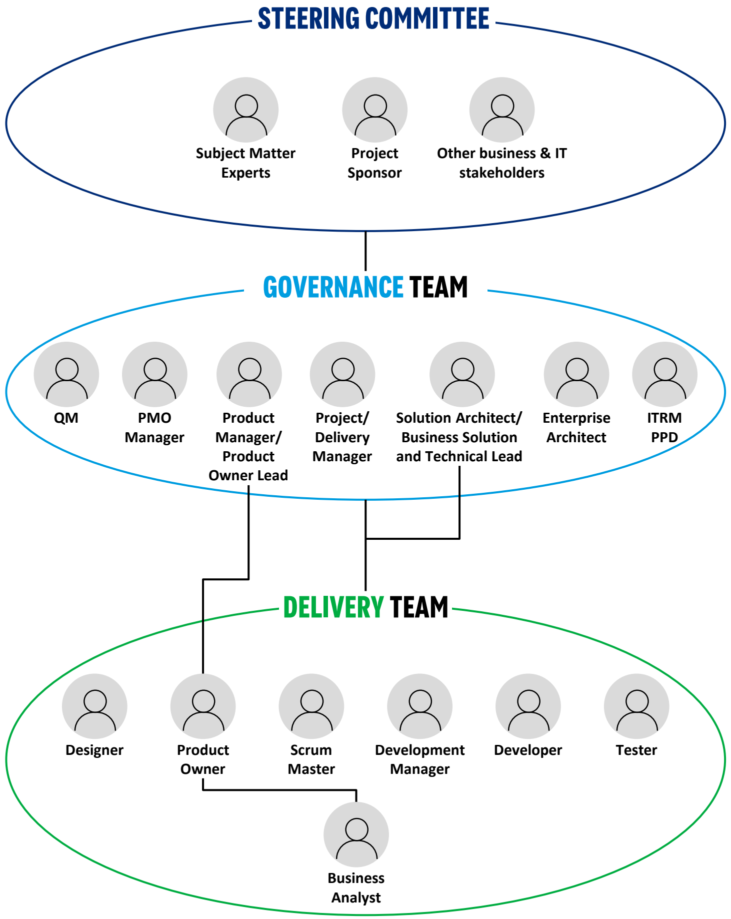
AGILE ORGANIZATION AND KEY ROLES

AGILE ORGANISATION – GUIDANCE

Agile Organisation[1]

ENABLEMENT

- Infra
- Legal/ Compliance
- CSM
- Implement-ation
- End User Training



STEERING COMMITTEE

Subject Matter Experts · Project Sponsor · Other business & IT stakeholders

GOVERNANCE TEAM

QM · PMO Manager · Product Manager/ Product Owner Lead · Project/ Delivery Manager · Solution Architect/ Business Solution and Technical Lead · Enterprise Architect · ITRM PPD

DELIVERY TEAM

Designer · Product Owner · Scrum Master · Development Manager · Developer · Tester · Business Analyst

High-Level Roles and Responsibilities

- Set high-level direction of the project and monitor strategic progress
- Review project deliverables and make toll-gate decisions
- Resolve major risks, and issues scope, timeline, and budget that the Governance Team cannot resolve

- Manage scope of work, timeline and budget to ensure the constraints are controlled
- Resolve risks and issues concerning the project constraints
- Ensure that functional and non-functional requirements are met

- Develop the software within specified timeline and budget
- Identify and mitigate risks and issues concerning project constraints

- Provide build requirements e.g., laws to be complied
- Review that project deliverables meet requirements
- Work with Delivery team to provide resources e.g., environments

- **Cross-functional** delivery teams, committed **full-time** to project

1. **Self-organizing, empowered** teams

2. Supported by **strong Product Owner**

[1] Please refer to the Agile Way of Working document for the detailed list of Agile roles and responsibilities

[2] For example, a value proposition Scrum team would have more business than IT members (incl. roles like customer insights specialist) than a value delivery team which would likely have more IT members than business

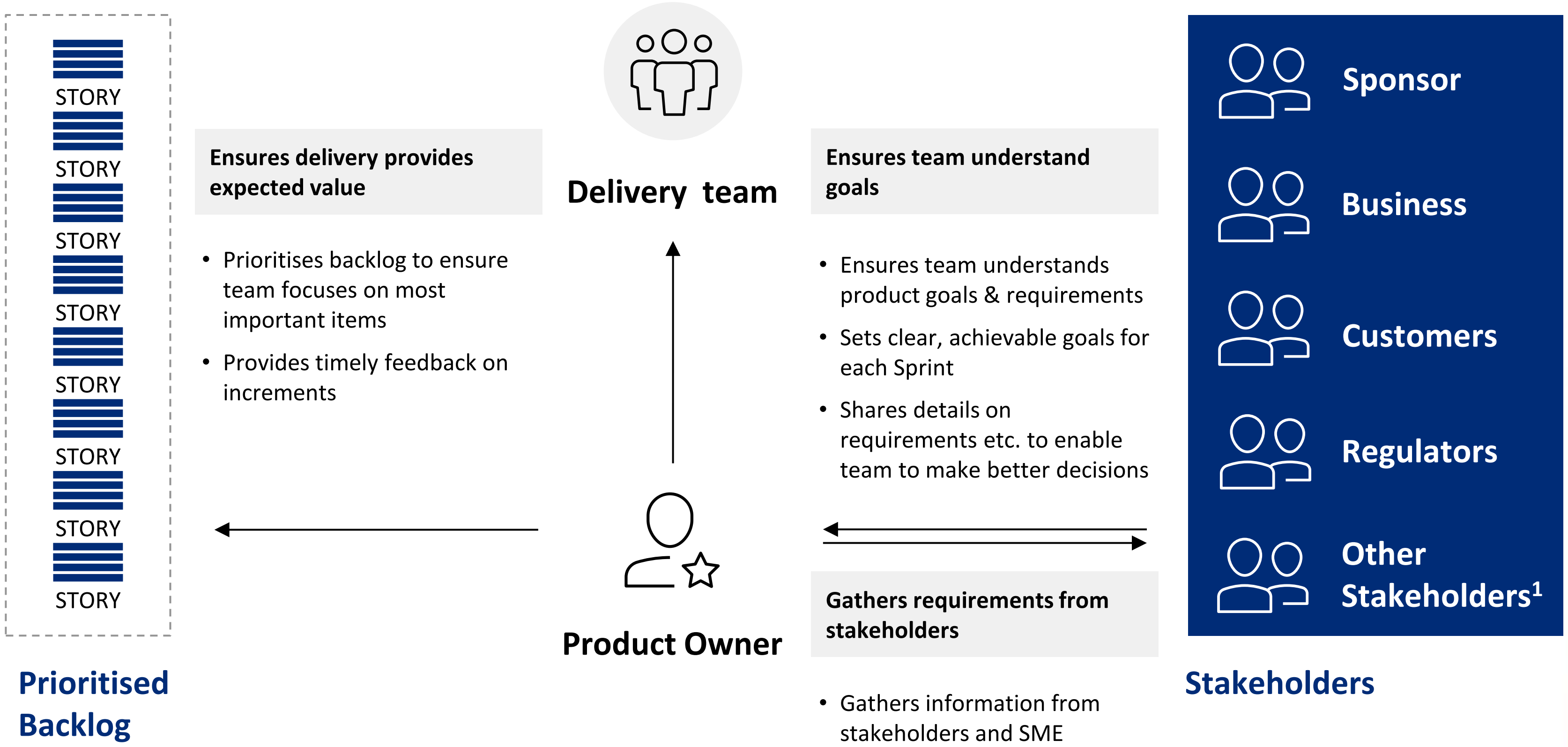
1. PRODUCT OWNER

Product Owner plays a crucial role in bridging the gap between the Delivery team, and other stakeholders, thereby empowering the team to deliver quickly

Role of Product Owner

- **Trusted ambassador** of the project sponsor, typically appointed from the business unit that benefits most from the project
- **Empowers the Delivery team** by bridging the gap between the team and other stakeholders incl.
  - Gathers requirements from all stakeholders
  - Ensures Delivery team understands these requirements
  - Ensures delivery provides expected value
- Thereby, Product Owner should have the skills of **collaboration, & autonomous decision making**

Empowerment of Delivery Team by Product Owner



Ensures delivery provides expected value

- Prioritises backlog to ensure team focuses on most important items
- Provides timely feedback on increments

Ensures team understand goals

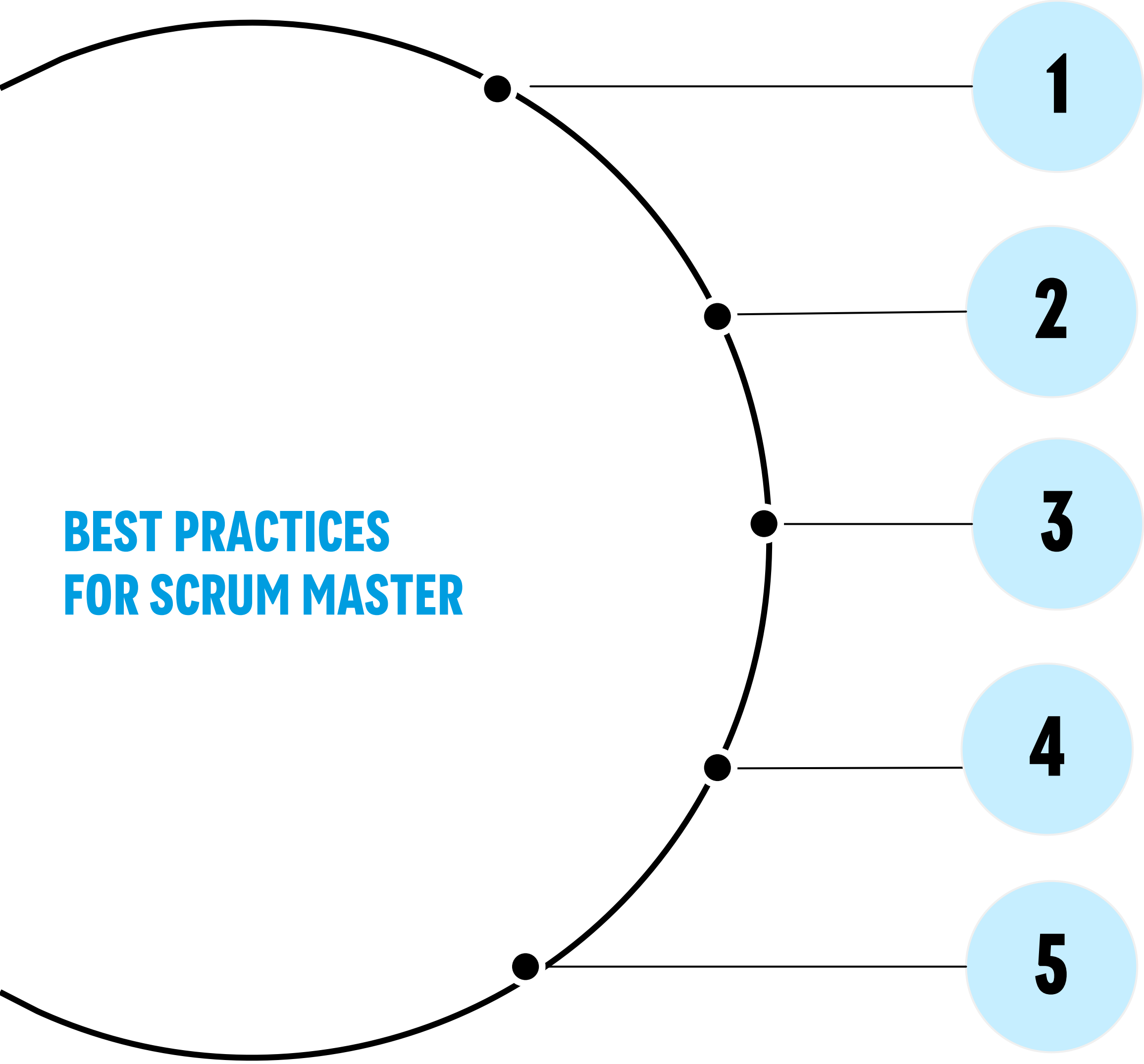
- Ensures team understands product goals & requirements
- Sets clear, achievable goals for each Sprint
- Shares details on requirements etc. to enable team to make better decisions

Gathers requirements from stakeholders

- Gathers information from stakeholders and SME

Bangkok Bank

2. SCRUM MASTER SETS SCRUM EVENTS FOR THE TEAM TO ENSURE A SUCCESSFUL SPRINT CYCLE WITH COLLABORATIVE, WELL-DOCUMENTED AND FOCUSES ON REALISTIC OUTCOMES



BEST PRACTICES FOR SCRUM MASTER

1

Good Collaboration

Encourage open communication and collaboration by the squad; Sprint goals are determined with inputs from and aligned with the squad; Agile ceremonies have been conducted regularly.

2

Focus on realistic commitments

Make realistic work commitments for each sprint; Only include stories that the build team confirms as achievable based on effort estimates.

3

Ensuring continuous improvement

Feedback is a critical tool for continuous improvement and problem-solving. This can be conducted within Scrum team during Sprint Retro.

4

Document outcomes

Document the outcomes of sprint planning (incl. user stories, story points, owners) in the sprint backlog as well as the sprint board so that it is visualized to the team.

5

Always celebrate

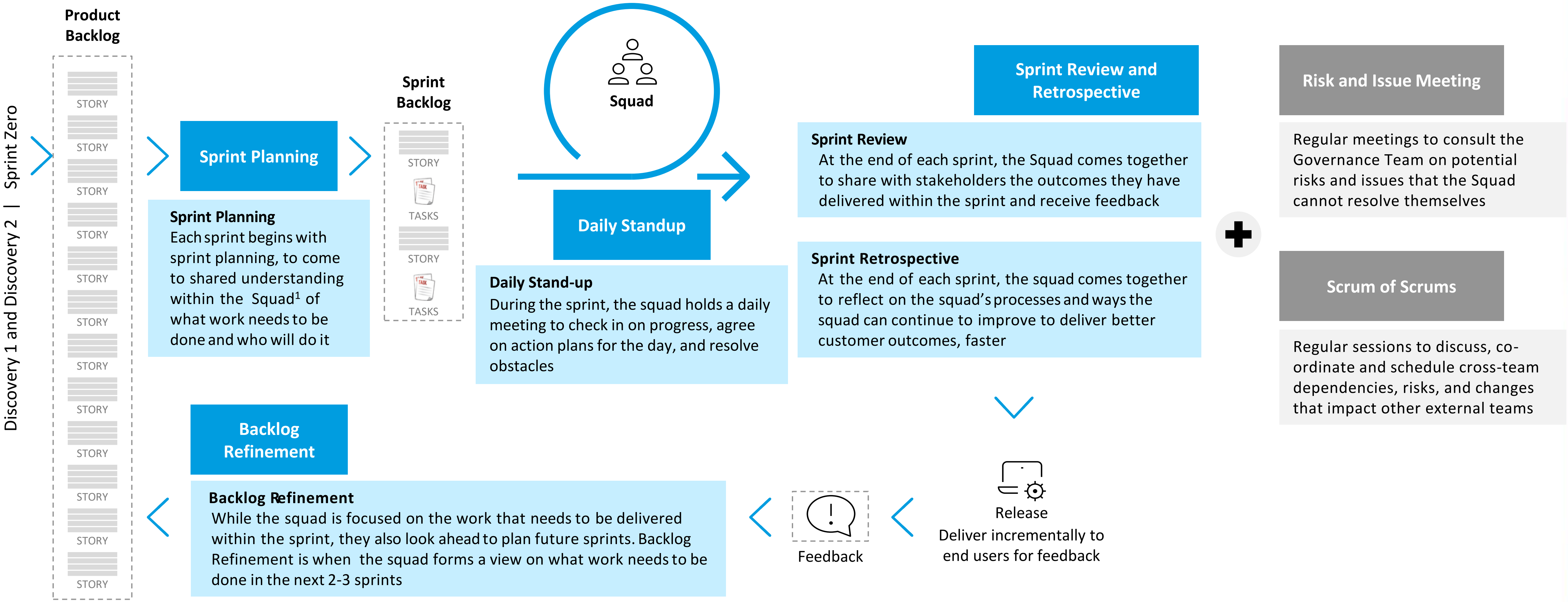
To recognize and reward the achievements of Scrum teams to motivate the good vibes for the teams to start the next sprints.

AGILE CEREMONIES

AGILE CEREMONIES ARE STRUCTURED MEETINGS THAT FACILITATE COLLABORATION, PLANNING, AND FEEDBACK, HELPING DELIVER WORK FROM THE BACKLOG ITERATIVELY

Agile Ceremonies

Other Ceremonies



Sprint Planning

Sprint Planning
Each sprint begins with sprint planning, to come to shared understanding within the Squad[1] of what work needs to be done and who will do it

Daily Standup

Daily Stand-up
During the sprint, the squad holds a daily meeting to check in on progress, agree on action plans for the day, and resolve obstacles

Sprint Review and Retrospective

Sprint Review
At the end of each sprint, the Squad comes together to share with stakeholders the outcomes they have delivered within the sprint and receive feedback

Sprint Retrospective
At the end of each sprint, the squad comes together to reflect on the squad's processes and ways the squad can continue to improve to deliver better customer outcomes, faster

Release
Deliver incrementally to end users for feedback

Backlog Refinement

Backlog Refinement
While the squad is focused on the work that needs to be delivered within the sprint, they also look ahead to plan future sprints. Backlog Refinement is when the squad forms a view on what work needs to be done in the next 2-3 sprints

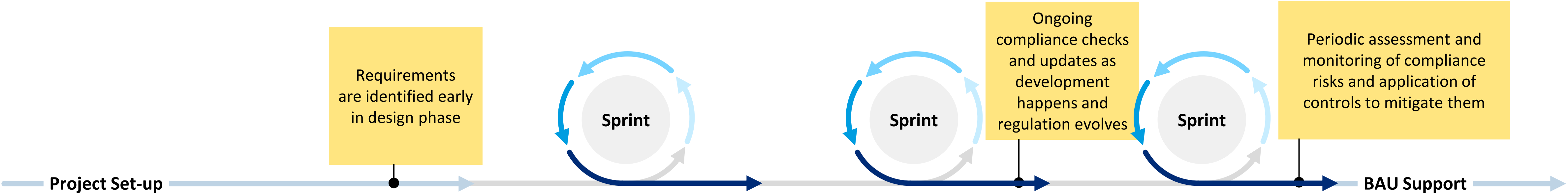
Risk and Issue Meeting

Regular meetings to consult the Governance Team on potential risks and issues that the Squad cannot resolve themselves

Scrum of Scrums

Regular sessions to discuss, co-ordinate and schedule cross-team dependencies, risks, and changes that impact other external teams

1. A Squad is a cross-functional, autonomous team which includes the Product Owner, build team (developers, designers, analysts, QA), and Scrum Master. This term can be used interchangeably with Scrum team or Delivery team.
2. Please note, these are additional Ceremonies specific to Agile projects. Any other meetings required for senior mgt. reporting and updates e.g., sponsor update, SteerCo. Should be carried out as per usual BBL cadence and agenda.



Requirements are identified early in design phase

Ongoing compliance checks and updates as development happens and regulation evolves

Periodic assessment and monitoring of compliance risks and application of controls to mitigate them

Project Set-up

Sprint

Sprint

Sprint

BAU Support

	Discovery 1 and Discovery 2	Sprint Zero	Sprint Planning*	Backlog Refinement*	Sprint Review*	Risk and Issue*	BAU Support
Delivery team activities	• [SM|PM] Share project scope and objectives with everyone and compliance before discovery so the right policy experts can be identified and invited to the Discovery 1 • [**SA**|PO] [DM] *Ensure requirements are included in the high-level solution design created for* Discovery 2	• [PO] Define Team mission and norms • [PO] Ensure requirements are added and prioritized in the Product Backlog • [SA] [DM] Refinement of Sequence Diagram, Architecture Diagram • [PO][SM]Backlog creation / Refinement	• [PO] *introduces business objectives* to be achieved for that sprint • [PO] (with inputs from the squad) sets sprint goals • *Squad selects stories* (from Product Backlog) *relevant to sprint goals and achievable based on effort,* and adds them to the sprint backlog	• [PO][SM] *Product backlog prioritization* • [PO][SM] [DM}Refining User Stories and conduct estimation • Answers any questions that remaining	• [Scrum team] *Demonstrate completed work*, highlighting how requirements have been met, and gather feedback • [PO] Add and *prioritize feedback as user stories* into the product backlog • *Review of completed work* to ensure requirements are met • Provide feedback on any compliance-related issue	• Identify and assess risks • [PM] Escalate and consult compliance relevant risks and issues to Governance team (incl. SME like Compliance) • Provide consultation and monitor compliance related risk and issues	• [Scrum team] Support as BAU • [Scrum team] Ensure timely resolution of these issues
Outcome / Deliverable	• *Functional requirements* • *Non-functional requirements* • Project constraints • *Risks, Issues*, Concerns of this project • Measurable success statement/definition • Scoped requirements – what's in scope, what's not in scope • *Agreed ways of working* (incl. Agile ceremonies and attendees) • Initial Incremental Release approach (what would be order of release for the core functionalities at the first few releases)	• *Team charter* • Wireframe Sequence Diagram (updated) • Architecture Diagram (updated) • *Detailed User Stories/Work Tasks* (Tech) for 1-2 Sprints • Sprint Definition of Done, Release DoD • *Environments ready* (e.g., SIT, UAT) • *Tools setup* (e.g., DevOps, CI/CD, etc.)	• *Sprint backlog* (committed user stories) • *Subtasks to complete User Stories*	• Product backlog (updated) • Refined User Stories (incl. estimation of story points)	• *Working Software* • *Sprint Report*	• *RAID log* • Risks to be escalated (if any)	• Liaise with scrum team for remediation of incidents e.g., data loss as part of the incident mgmt. process

Notes:
* Full Ceremony, outlined in the PMM0003 - Way of Working document and demonstrated during the walkthrough on 26 Nov.

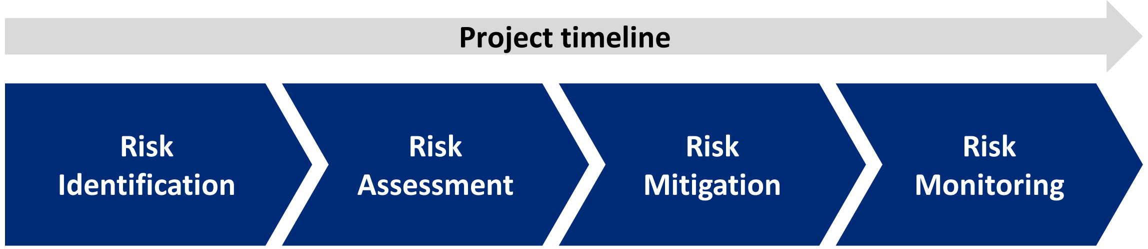
➡ Plan ➡ Design ➡ Develop ➡ Test ➡ Release

SM – Scrum Master **PM –** Project Manager **PO –** Product Owner **SA –** Solution Architect **DM –** Development Manager

RISK MANAGEMENT IN AGILE

BOTH WATERFALL AND AGILE METHODOLOGY MUST HAVE SOLID RISK MANAGEMENT PROCESS REGARDLESS OF THE DIFFERENT APPROACH

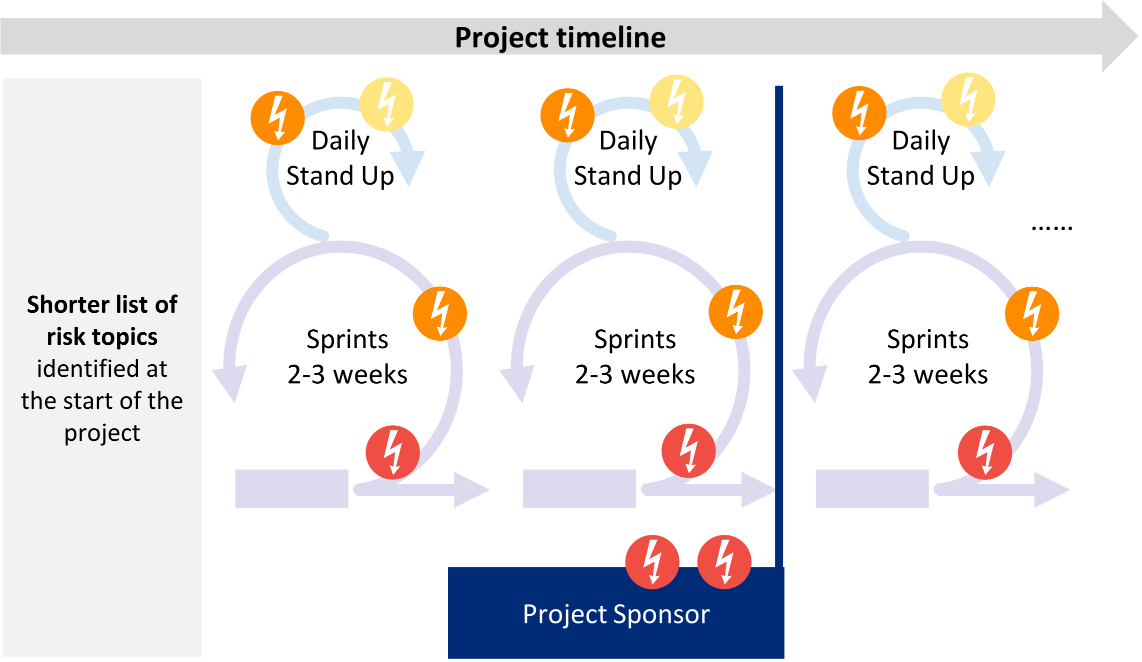
Risk management on a Waterfall project



Long list of risks identified at the start of the project

- Conduct risk assessment based on **rigorous root cause analysis**
- Require **identification** of all relevant risks **upfront**
- **Regular measurement and reporting**
- **Escalate risks** according to **fixed meeting cadence** to IT Leadership
- Develop **robust mitigation plan** with clear action items

Risk management process an agile project



Trivial risk events get resolved by scrum team throughout the day

Small risk events get reviewed and resolved by scrum team/ project at end of sprint

Only serious risk events get escalated to Project Sponsor and resolved by senior stakeholders

TO ADDRESS COMMON RISK MANAGEMENT CHALLENGES, WE CAN LEVERAGE FRAMEWORKS AND TOOLS, INCLUDING RISK AND ISSUE / RAID LOGGING WITH ROAM TECHNIQUE

Frameworks and tools in Agile risk management			Common challenges addressed
FRAMEWORKS	RAID	A simple project management practice that organises a project by its key risks, assumptions, issues and dependencies	• **Comprehensive tracking** through collaborative and frequent risk identification • **Proactive identification of potential risks** in addition to existing issues
	ROAM	A collaborative, lightweight framework used to manage risks and issues once they have been identified	• **Ensures** all RAID outputs have a mgmt. plan and nothing slips through the cracks • Effective risk management through **collaborative decision making**
TOOLS	RISK AND ISSUE / RAID LOG	A centralised live database of all risks and issues adversely affecting or likely to affect the project	• **High visibility** as the log acts as single source of truth in tracking all RAID items and mitigation plans • **Key input to product backlog** (for planning)

THE RAID FRAMEWORK FACILITATES TRACKING OF POTENTIAL ROADBLOCKS TO PROJECT SUCCESS AS RISKS, ASSUMPTIONS, ISSUES AND DEPENDENCIES

RAID components	Description	Examples
RISKS	An event that might occur, and if it does, will impact delivery or pose a threat to the Bank	*Team is using an untested framework for the project which may have unforeseen bugs or limitations that could delay development or require significant rework*
ASSUMPTIONS	Belief that things should happen in a particular way but there is a possibility that they might not. In the latter case, there will be an impact on delivery	*Team assumes that all end-users will have reliable high-speed internet access, which is critical for the performance of the web application being developed*
ISSUES	A problem that has already happened or will definitely occur	*The project is experiencing frequent build failures in CI/CD pipeline, causing delays in testing and deployment from the Sprint plan*
DEPENDENCIES	An item that a team are reliant on for completing a task	*Team cannot proceed with certain integration tasks, as they depend on an updated version of third-party API*

THE ROAM TECHNIQUE ENSURES THAT EACH RAID ITEM IS MANAGED EFFECTIVELY BY FACILITATING ASSIGNMENT AND TRACKING OF RELEVANT NEXT STEPS

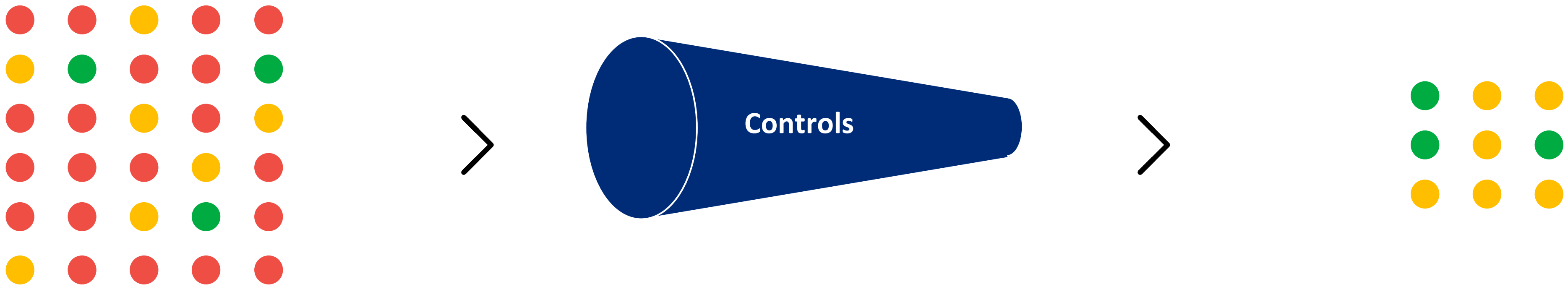
Risk - Team is required to use a new tech stack which may lead to delays in project timeline due to ramp-up period

ROAM model	Description	Examples
RESOLVED	Identified risk or issue is not a threat anymore. Either the team has taken mitigation actions to eliminate it, or the level of risk or issue is negligible	*The team has completed a comprehensive training program on the new technology stack, and they have successfully implemented it in a pilot project. The risk is no longer a threat because the team is now proficient with the technology*
OWNED	An owner has been assigned and has accepted responsibility to manage the risk or issue. This could mean that the owner accepts responsibility for identifying and/or implementing mitigation plan(s)	*A senior developer with experience in the new technology stack has been assigned to the team. This developer is responsible for guiding the team, providing training, and ensuring that the technology is implemented correctly. The risk is managed by this owner*
ACCEPTED	Risk / issue is known, and impact is considered acceptable by relevant stakeholders. This may happen when the cost of mitigation is greater than impact of letting the risk transpire or when no mitigation actions are available	*After evaluating the potential impact and the cost of mitigation, the stakeholders decide that the risk of the team being unfamiliar with the new technology stack is acceptable. They believe the team will learn on the job and potential delays are manageable within the project timeline*
MITIGATED	Mitigation actions have been taken; however, the severity of the risk and issue is only partially mitigated, and the risk and issue is not eliminated entirely	*The team has attended several training sessions. They have started using the new technology stack in a controlled environment. While the risk is reduced because the team is gaining familiarity, it is not eliminated as the team is still not proficient in the technology, which may still lead to delays*

CONTROLS REDUCE THE INHERENT LEVELS OF (PROJECT DELIVERY) RISKS

Overview of controls being used to reduce risk exposure of the project

Inherent risks | Controls | Residual risks



Simply by existing, banks are exposed to risk events with varying **impact and likelihood (inherent risk)** based on the project’s characteristics

These inherent risks must be managed and **reduced by applying controls** (measures prescribed to protect the bank against risks on deliver projects e.g. processes, tools, testing etc.)

After mitigating inherent risk levels by applying controls, the project must **evaluate and accept the residual risks** and monitor it on an ongoing basis

🔴 High 🟡 Medium 🟢 Low

DELIVERY TEAMS LEVERAGE BOTH PERIODIC AS WELL AS ITERATIVE (SPRINT) PLANNING TO MANAGE RISKS

1

Milestone (e.g., release) Planning

Project / Delivery manager creates milestone plans for the delivery team based on the prioritised epics, user stories and tasks in the product backlog

2

Risk Identification

Delivery team collaboratively brainstorms potential challenges in the successful completion of the periodic plan using RAID

3

Risk Mitigation

Delivery teams collaboratively determine risk mgmt. actions and action owners using ROAM

4

Risk Logging and Reporting

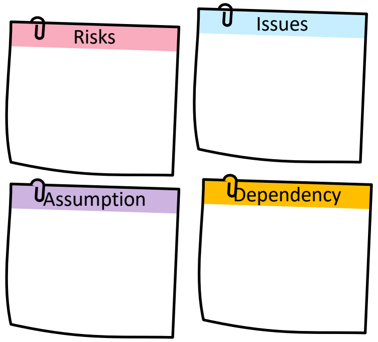
Risks are added to the Risk and Issue / RAID Log and backlog (if mitigation actions are assigned)

Risks that cannot be resolved by the delivery team are escalated

5

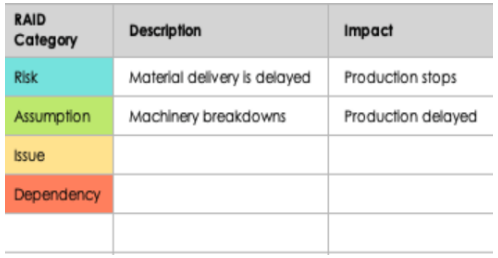
Iterative Risk Planning in Sprints

Any new risks identified during sprint executions (that are not resolved during that cycle) are added to the Log and reviewed frequently (ideally every sprint cycle)



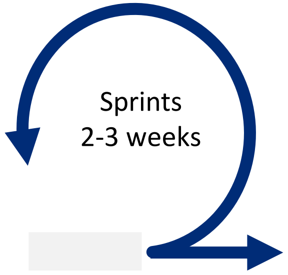
RAID Framework

ROAM Framework



RAID Category	Description	Impact
Risk	Material delivery is delayed	Production stops
Assumption	Machinery breakdowns	Production delayed
Issue		
Dependency		

Risk and Issue / RAID Log



Iterative Risk Identification and Monitoring

AGILE BEST PRACTICES IN THE RISK ESCALATION PROCESS

A

What risks should be escalated?

B

When should risks be escalated?

C

How should risks be escalated?

Best Practice Recommendations (Deep-dives ahead)

A

- High level of risk mgmt. delegation; Delivery team independently resolves trivial risks and escalates small risks, if required
- Material and urgent risks should be directly escalated to Project Sponsor out of regular meeting cadence

B

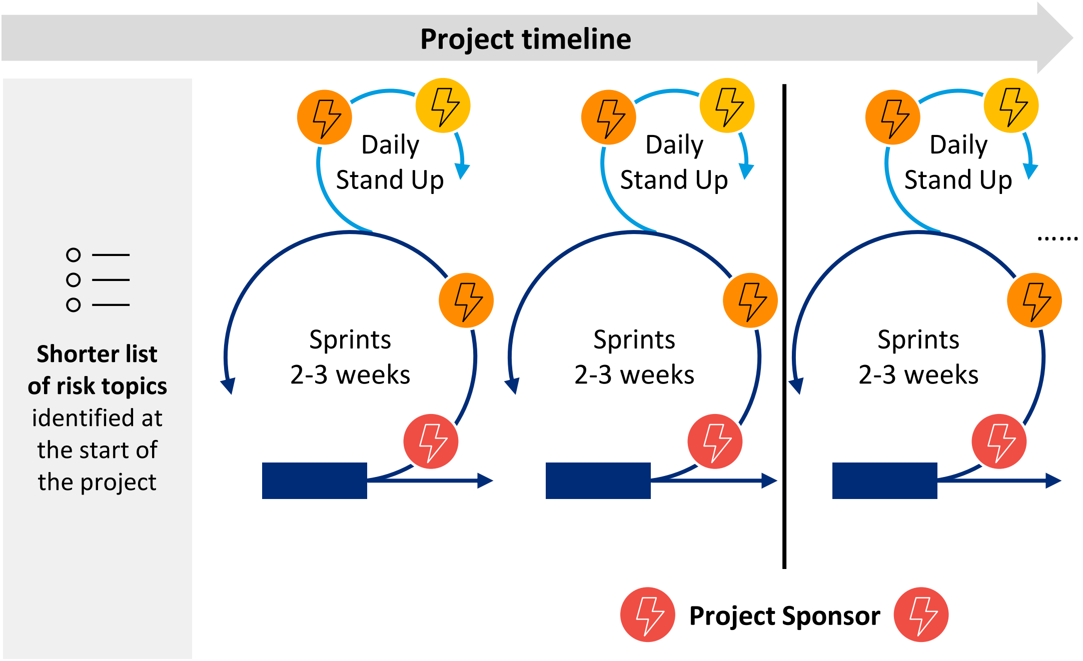
- Escalate with sufficient lead time to allow senior mgmt. to take mitigation actions and before major impact to project health
- Factor in likely contingencies when determining timing of escalation

C

- Risk reporting should be light-weight and catered to meet the desired outcomes of each escalation
- Senior management reporting should be limited to the most critical open risks, that require their attention

A DELIVERY TEAMS AUTONOMOUSLY RESOLVE TRIVIAL RISKS THROUGHOUT THE SPRINT; OTHER RISKS ARE ESCALATED FOR CONSULTATION BASED ON NATURE AND SEVERITY

Risk Management in Agile



Trivial risk events are resolved by Delivery team through the day

E.g., Incompatible technology for implementing unit tests for batch process

Small risk events that cannot be resolved by Delivery team and are escalated to Governance team and further to Project Sponsor, If required

E.g., Delay in availability of SIT environment

Serious risk events get escalated directly to Project Sponsor and resolved by senior stakeholders

Escalation Mechanisms for Common Risks and Issues

Initiator	Nature of Risk/Issue to be Escalated	1st Escalation	2nd Escalation
Project/ delivery manager	Exceed project budget	**Project Sponsor**	N/A[1]
	Exceed project timeline		
	Material quality impact		
	Significant scope change		
	Milestone release not met	**Risks & issues**	**Project Sponsor**
	Agile way of working standard not followed		
	Risks and issues not raised by scrum teams		
	Vendor progress delayed		
Product manager/ product owner lead	Material quality impact		
	Product goals not met		
Enterprise architect	Enterprise architect standards not met		
Scrum master	Scrum framework not followed		
	Definition of done not met		
	Definition of ready not met		
	Test plan not followed		
	Release plan not followed		
Product owner	Prioritization of requirements not followed		
	Sprint goals not met		
Scrum team	Release timeline not met		
	Disruption to scrum process		

[1] 2nd escalation possible to senior mgmt. on major and critical projects

B RISK OWNER SHOULD ESCALATE TO AVOID ADVERSE IMPACT – SUFFICIENT LEAD TIME SHOULD BE BUILT IN FOR CONTINGENCIES, AND ACTION BY SENIOR MGMT.

Good Practices for Timely Risk Escalation

Timely risk management is an **art rather than science**; good practices are

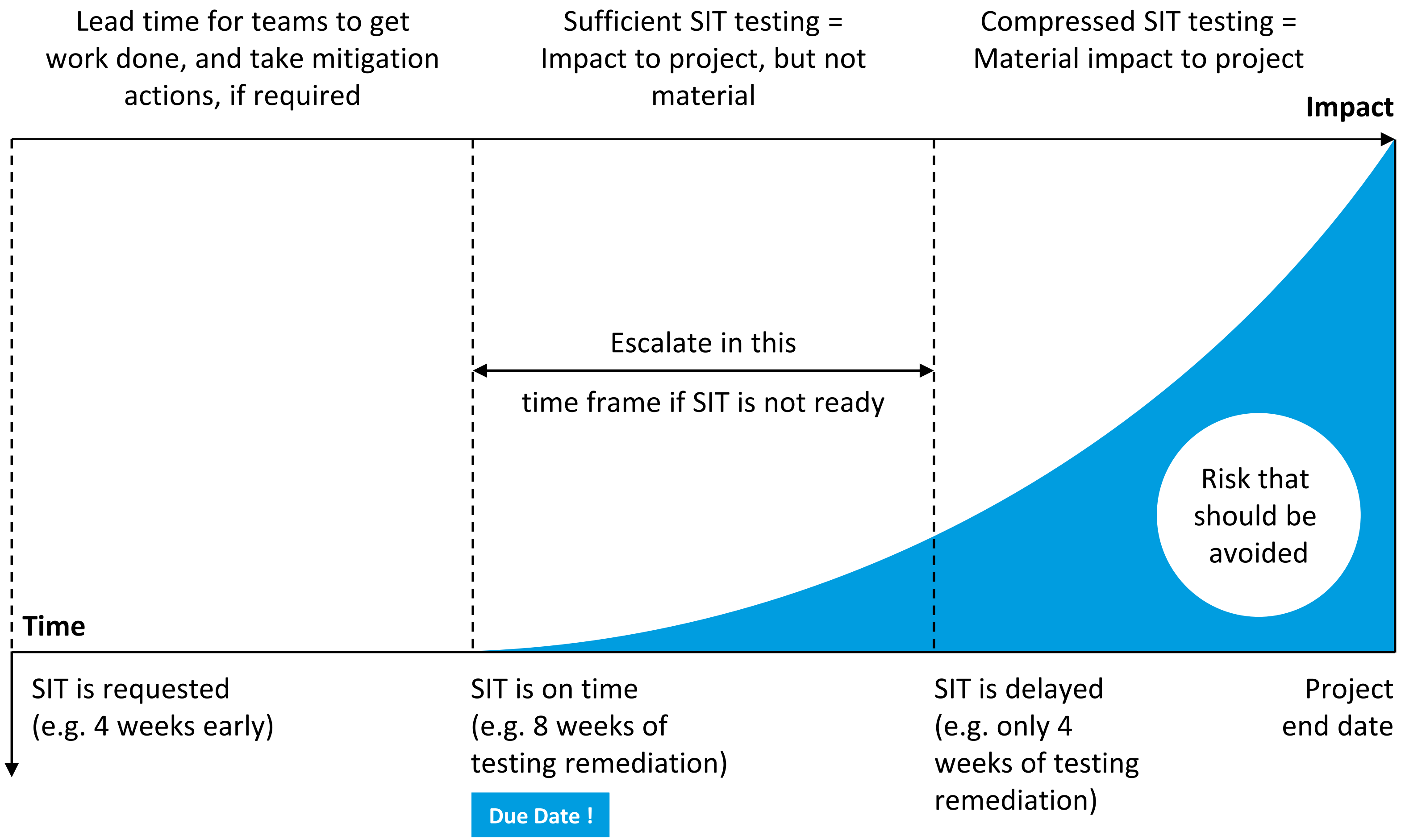
- Escalate with sufficient time (after factoring likely contingencies) for resolution before major impact to project e.g., delay in project timeline

- Escalate when team has identified mitigation actions, but is facing blockers

- Where dependencies are involved, manage and track closely, and escalate with sufficient lead time to allow senior mgmt. to influence work schedule[1]

Such escalations can be before the due date

Illustrative example

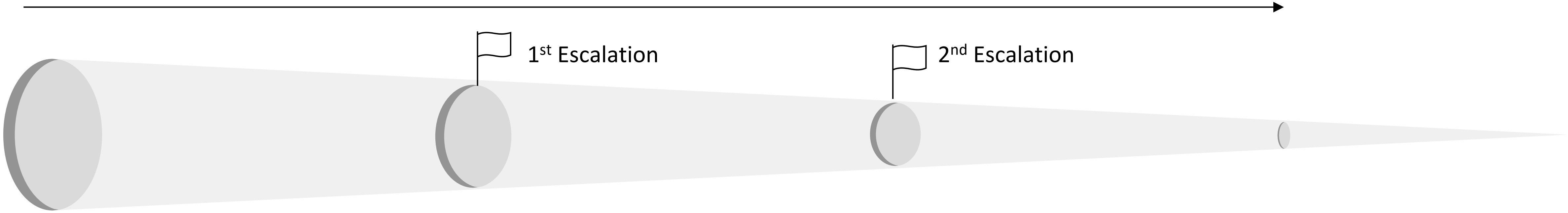


Lead time for teams to get work done, and take mitigation actions, if required
Sufficient SIT testing = Impact to project, but not material
Compressed SIT testing = Material impact to project
Impact
Escalate in this
time frame if SIT is not ready
Risk that should be avoided
Time
SIT is requested (e.g. 4 weeks early)
SIT is on time (e.g. 8 weeks of testing remediation)
Due Date !
SIT is delayed (e.g. only 4 weeks of testing remediation)
Project end date

[1] Factor in additional time as dependent teams may not be working in Agile methodology and / or may have other urgent or high priority work items that may not be known to Delivery teams

C IN AGILE, RISK REPORTING IS KEPT LIGHT-WEIGHT AND CATERED TO THE RELEVANT AUDIENCE; RAID LOG AND RISK AND ISSUE MATRIX ARE THE PRIMARY REPORTING TOOLS

Risk Escalation



Delivery team
Identify, manage and monitor risk throughout the Sprint; Escalate to Governance team or Project Sponsor, as required

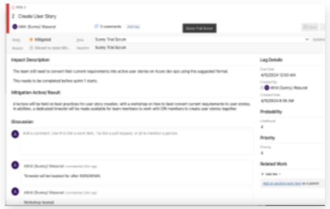
Governance team
Provide consultation and monitors RAID items escalated by the delivery team; Escalate to Project Sponsor, as required

Project sponsor
Provide consultation and monitor RAID items escalated by Delivery or Governance team

IT SteerCo.
Receive regular updates and monitor status of critical RAID items

Risk Reporting Tooling *(Deep-dives ahead)*

RAID Log



2 Create User Story

Impact Description

The team still need to convert their current requirements into active user stories on Azure dev ops using the suggested format.

This needs to be completed before sprint 1 starts.

Mitigation Action/ Result

A lecture will be held on best practices for user story creation, with a workshop on how to best convert current requirements to user stories. In addition, a dedicated timeslot will be made available for team members to work with [illegible] members to create user stories together.

Discussion

Workshop hosted

Log Details

Probability

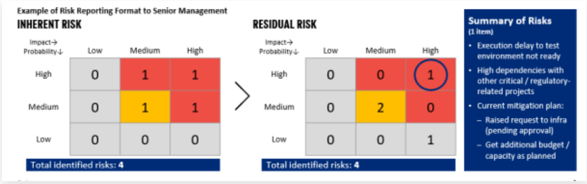
Priority

Related Work

Primary log for all RAID items; Captures best practice components for day-to-day monitoring and escalation decisions making by Delivery & Governance team

Deep – dive ahead

Risk and Issue Matrix



Example of Risk Reporting Format to Senior Management

INHERENT RISK

Impact→ Probability↓	Low	Medium	High
High	0	1	1
Medium	0	1	1
Low	0	0	0

Total identified risks: 4

RESIDUAL RISK

Impact→ Probability↓	Low	Medium	High
High	0	0	1
Medium	0	2	0
Low	0	0	1

Total identified risks: 4

Summary of Risks
(1 item)

- Execution delay to test environment not ready
- High dependencies with other critical / regulatory-related projects
- Current mitigation plan:
  - Raised request to infra (pending approval)
  - Get additional budget / capacity as planned

Summary view of the most critical RAID items are escalated to Project Sponsor

Deep – dive ahead

AGILE REPORTING

WE PROPOSE BEST PRACTICE AGILE TOOLS & REPORTS, SET-UP ON INDUSTRY STANDARD SOFTWARE FOR SEAMLESS MONITORING & REPORTING OF AGILE PERFORMANCE

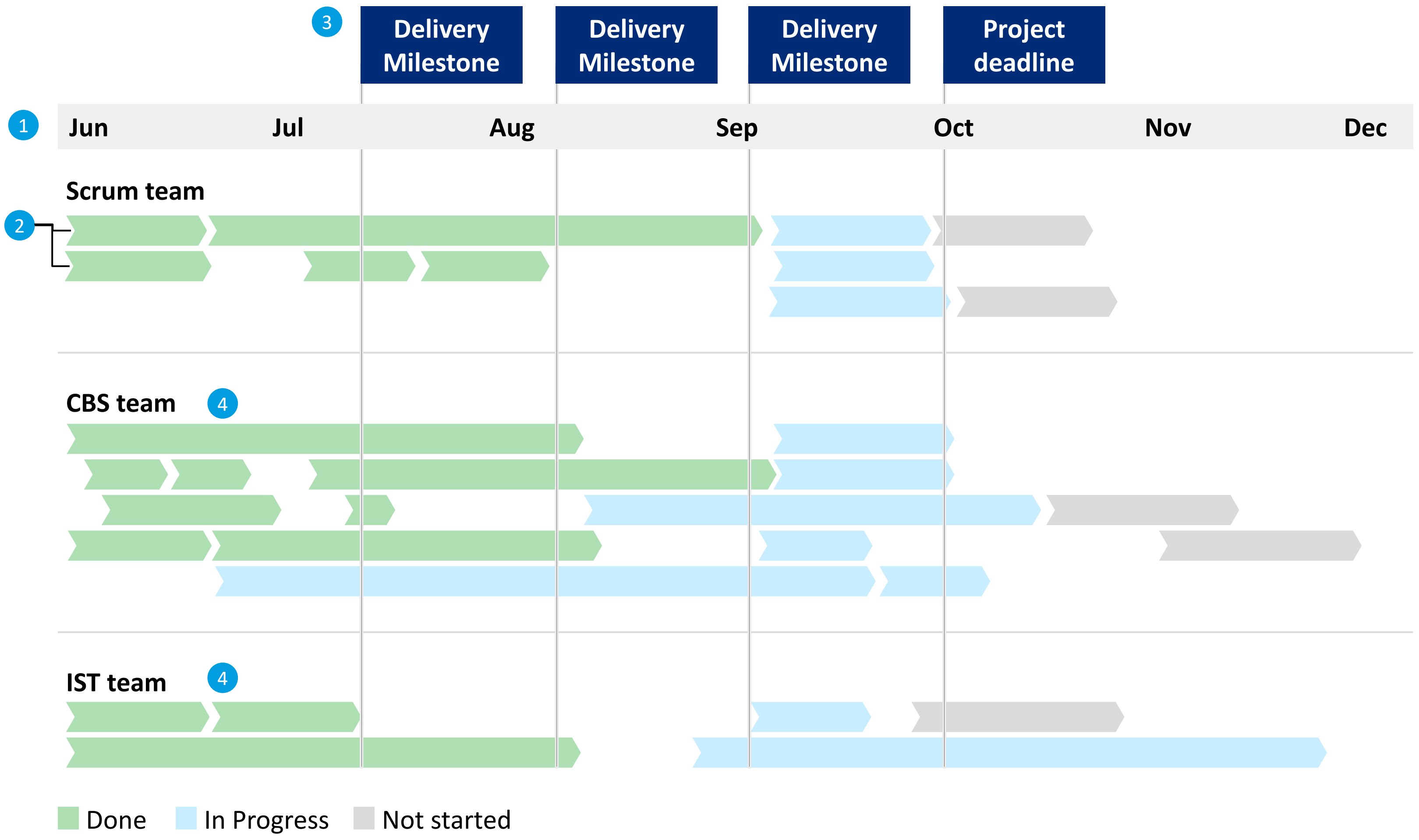
		Name	Description	Owner[1]	Monitoring & Reporting Uses	Software
Agile Tools	1	Project Tracker	High level timeline of workstreams at scrum and dependent teams in reference to delivery milestones	Project Manager | Scrum Master	• Regular monitoring by PM / Scrum Master • Periodic reporting to Project Sponsor (based on cadence of Project Sponsor update)	MS PowerPoint
	2	Project Burnup Chart	Visual representation of work completed over the project timeline vis-à-vis the project scope		• Regular monitoring by PM / Scrum Master • Periodic reporting to Project Sponsor (based on cadence of Project Sponsor update)	PowerBI
	3	Sprint Burndown Chart	Visual representation of remaining work vis-à-vis sprint plan at any given point during a sprint cycle		• Daily monitoring by PM / Scrum Master	PowerBI
	4	Velocity Tracker	Visual representation of rate of work accepted vs completed over consecutive sprints		• Regular monitoring by PM / Scrum Master • Included in Sprint Report for reporting purposes	PowerBI
Agile Reports	5	Sprint Report	Summary of the progress and outcome of a sprint cycle	Project Manager | Scrum Master	• Primary record keeping tool for sprints • Periodic reporting to Project Sponsor (based on cadence of Project Sponsor update)	Power BI

For guidance on set-up of Agile Tools on PowerBI, refer to **Agile Reports Set-up on PowerBI** document.

[1] Responsible for creation and update of the tool / report

1 PROJECT TRACKER: A HIGH-LEVEL OVERVIEW OF PROGRESS ACROSS TEAMS FOR SENIOR MGMT. TO MAKE QUICK DECISIONS ON PRIORITY OF WORK AND UNBLOCKING TEAMS

Project Tracker



Commentary

Provides overview of workstreams at scrum and dependent teams for progress tracking towards achievement of milestones

1 **Timeline**: 6-month forward looking timeline

2 **Workstreams**: High-level description of work at feature, MVP level

– Different states of work i.e., “Done”, “In Progress”, and “Completed” represented by different colours for progress tracking

3 **Delivery Milestones**: Significant milestone e.g., release

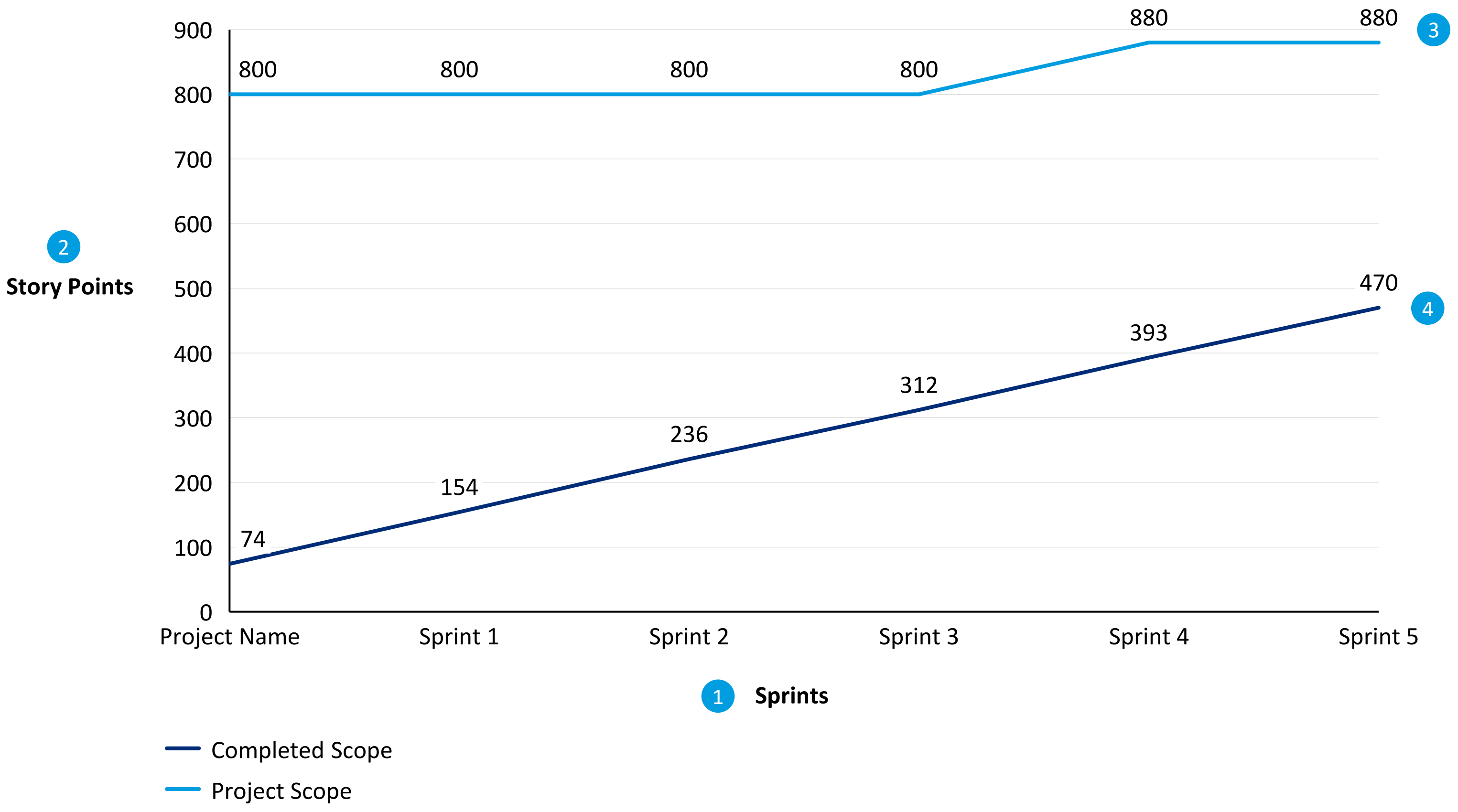
4 **Dependencies**: Workstreams at teams where dependencies need to be managed

Insights

- **Progress of project**, used by Governance team to identify risks and issues
- High-level work breakdown structure for senior mgmt. **decision making on work priority, and dependencies**
- **Team capacity** for program level planning (Tool enables shift in mindset to long-standing teams)

2 PROJECT BURNUP CHART: A VISUAL FOR MONITORING THE TEAM'S PROGRESS AGAINST PLANNED SCOPE, AND SUPPORTING PRIORITY DISCUSSIONS ON SCOPE CHANGES

Project Burnup Chart



Commentary

A burnup chart shows the amount of work completed over time, vis-vis the total project scope

1 **X-Axis**: Sprints cycles across the project timeline

2 **Y-axis:** Number of work items or Story Points

3 **Project Scope:** Represents the total story points to be completed across all work or features required to deliver the project

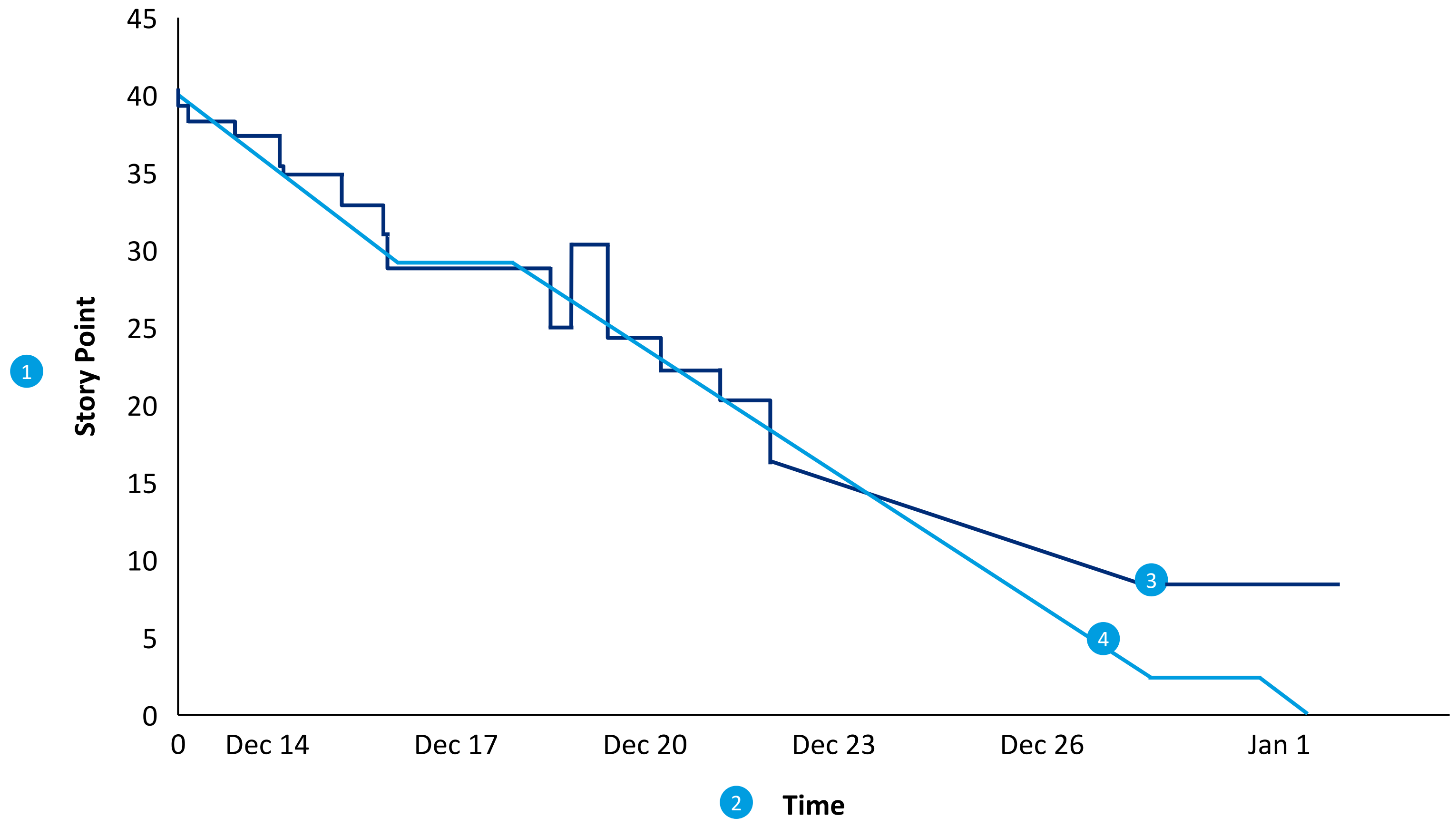
4 **Completed scope**: Accumulation of all story points completed (not planned or accepted) across sprints over time

Insights

- **Project progress vis-à-vis estimated timelines**, to identify blockers and take corrective actions
- **Changes in project scope over time**, to make decisions regarding team priorities (e.g., extend timeline vs de-prioritize some scope)
- **Work rate of the project team**, to assess if the team is on track to meet scope and plan capacity

3 SPRINT BURNDOWN CHART: QUICK PULSE CHECKS ON PROGRESS IN NEAR REAL-TIME DURING SPRINTS ENABLES TEAMS TO IDENTIFY AND SOLVE BLOCKERS IN A TIMELY MANNER

Sprint Burndown Chart



Commentary

A burndown chart shows the amount of remaining work vis-à-vis sprint plan during a sprint cycle

1 **Y-Axis:** Represents total estimation of all work items in the sprint measured in either story points or hours

2 **X-Axis:** Represents the sprint timeline

3 **Remaining values:** Total amount of work in sprint scope that is not completed

- Fluctuates based on the rate of actual work completion and new user stories added o sprint

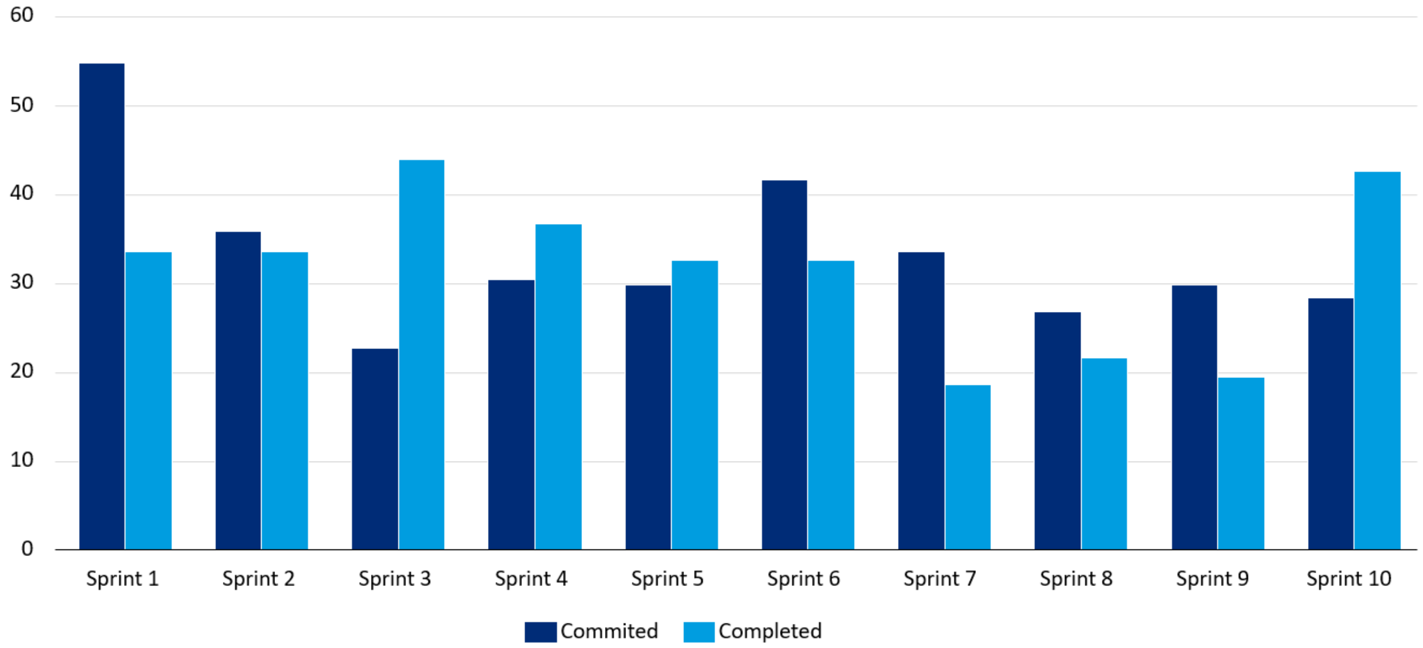
4 **Ideal Work Rate:** Approximate estimation of amount of work remaining at each point of the sprint cycle, assuming linear progress

Insights

- **Check on sprint progress**, allowing for forecasting whether the team is on track to meet Sprint plan
- **Early warning signs for blockers to achievement of sprint goals** i.e., when remaining work is above ideal work rate marker, for corrective actions
- **Trends on team velocity / productivity** (e.g., flat slope may indicate productivity issues)

4 VELOCITY TRACKER: VISUAL ON TRENDS FOR PLANNED & ACTUAL TEAM VELOCITY OVER TIME CAN HELP DETECT ANOMALIES IN PERFORMANCE & TAKE TIMELY CORRECTIVE ACTIONS

Velocity Tracker



Commentary

Graphical representation of the velocity of a scrum team across multiple Sprints

1. **X-axis:** Timeline across multiple historical sprints
2. **Y-axis:** Number of work items or story points
3. **Committed:** Represents work committed to be completed by the scrum team in each sprint
4. **Completed:** Represents work actually completed by the scrum team in each Sprint i.e., actual velocity of the team

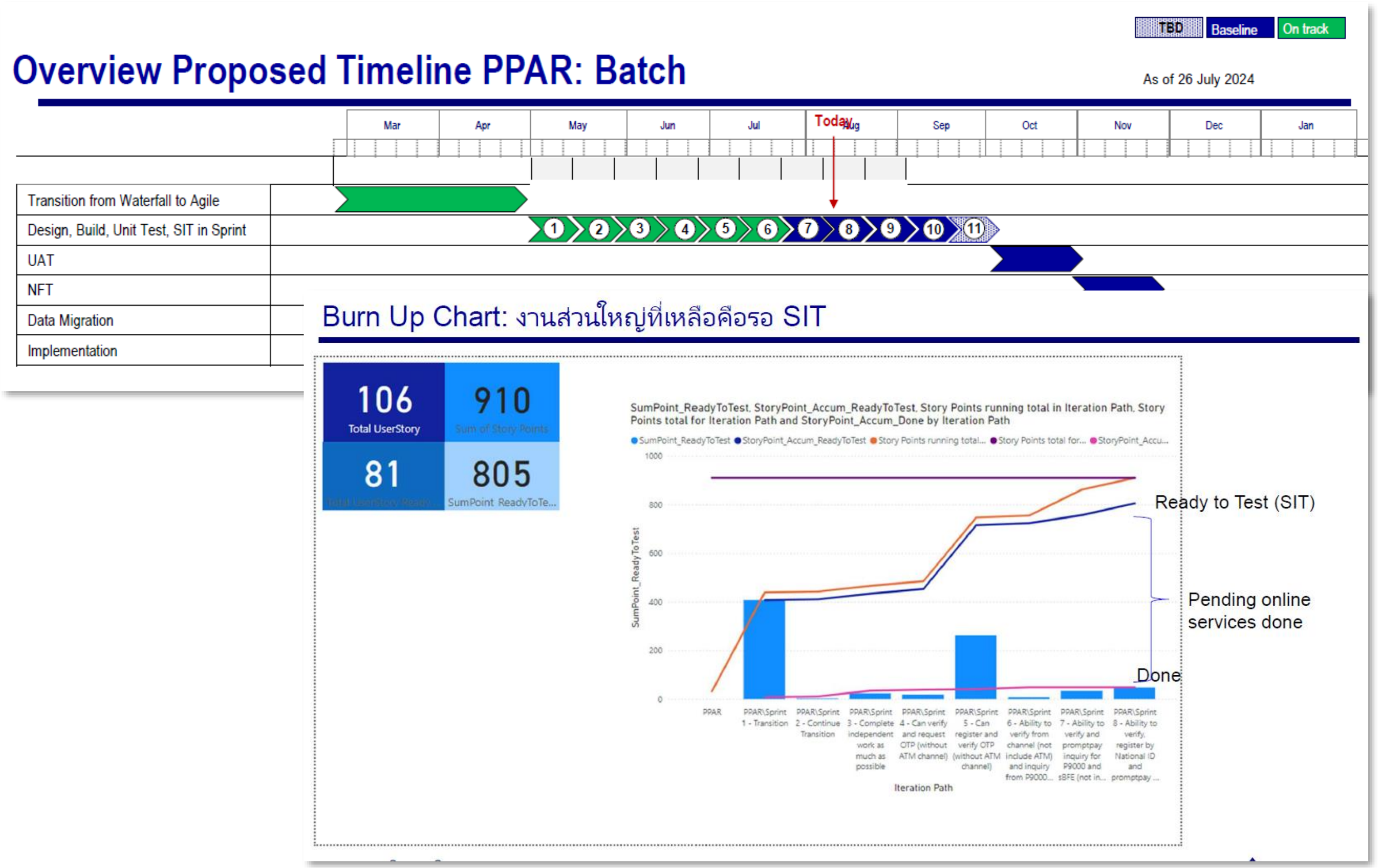
Insights

- **Team velocity over time** for more accuracy in **sprint planning**
- [For long standing teams] **Team velocity trends** for better **project and program level planning**
- **Anomalies in team performance**[1], which should be assessed to implement solutions to improve team performance

[1] Anomalies can include scenarios where Completed is much less than Committed or vice versa, and Completed in a sprint is much less than Completed in past sprints or vice versa

5 SPRINT REPORT: SERVES AS A LIGHTWEIGHT SUMMARY ON SPRINT GOALS, PERFORMANCE AND OUTCOMES FOR REGULAR REPORTING TO STAKEHOLDERS

Sprint Report



Overview Proposed Timeline PPAR: Batch

As of 26 July 2024

Burn Up Chart: งานส่วนใหญ่ที่เหลือคือรอ SIT

TBD
Baseline
On track
Mar
Apr
May
Jun
Jul
Today
Aug
Sep
Oct
Nov
Dec
Jan
Transition from Waterfall to Agile
Design, Build, Unit Test, SIT in Sprint
1
2
3
4
5
6
7
8
9
10
11
UAT
NFT
Data Migration
Implementation
106
Total UserStory
910
Sum of Story Points
81
Total UserStory Ready...
805
SumPoint_ReadyToTe...
SumPoint_ReadyToTest, StoryPoint_Accum_ReadyToTest, Story Points running total in Iteration Path, Story Points total for Iteration Path and StoryPoint_Accum_Done by Iteration Path
SumPoint_ReadyToTest
StoryPoint_Accum_ReadyToTest
Story Points running total...
Story Points total for...
StoryPoint_Accu...
1000
800
600
400
200
0
SumPoint_ReadyToTest
PPAR
PPAR\Sprint 1 - Transition
PPAR\Sprint 2 - Continue Transition
PPAR\Sprint 3 - Complete independent work as much as possible
PPAR\Sprint 4 - Can verify and request OTP (without ATM channel)
PPAR\Sprint 5 - Can register and verify OTP (without ATM channel)
PPAR\Sprint 6 - Ability to verify from channel (not include ATM) and inquiry from P9000...
PPAR\Sprint 7 - Ability to verify and promptpay inquiry for P9000 and sBFE (not in...
PPAR\Sprint 8 - Ability to verify, register by National ID and promptpay ...
Iteration Path
Ready to Test (SIT)
Pending online services done
Done

Commentary

Summary of goals, progress, and outcomes for each sprint

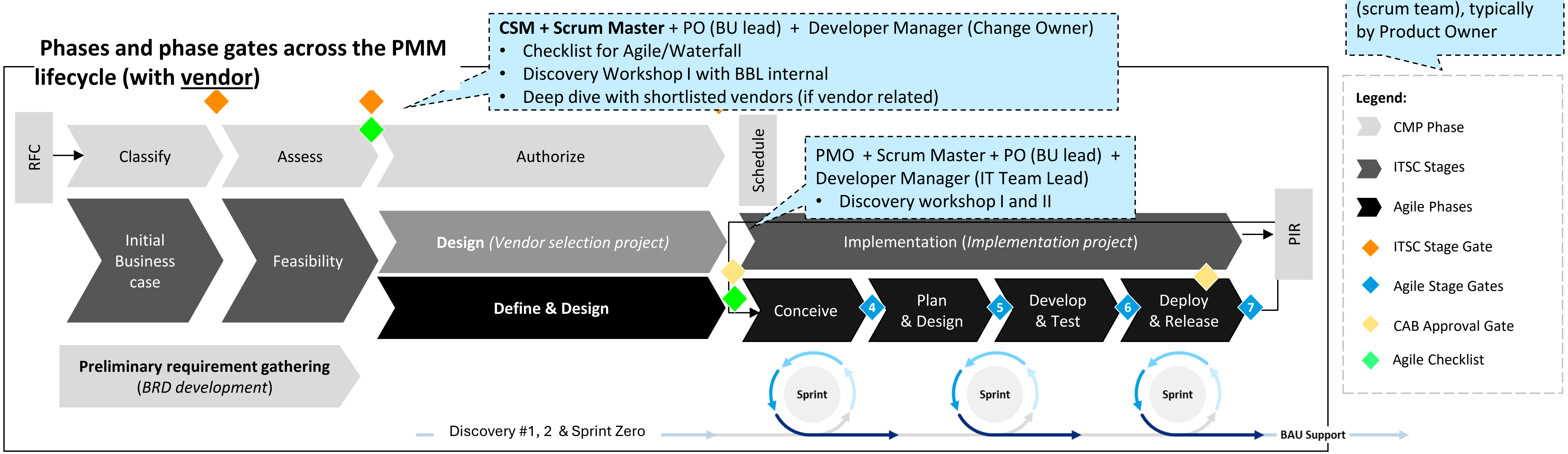
- **Sprint Summary:** Brief summary of the sprint cycle, including the duration, start and end dates
- **Sprint Goals:** Goals selected by team for sprint
- **Project Progress:** Overview of progress (incl. dependencies) against project scope as depicted by Project Tracker, and Project Burnup Chart
- **Work Completed:** User stories or tasks that were completed during the sprint, including story ID, title, short description, acceptance criteria, & story points
- **Team Performance:** Trend analysis on progress indicators for scrum team incl. velocity, number of user stories completed etc.
- **[OPT] Next Steps:** Introduction to next sprint's goals

Insights

- Primary **record keeping tool on all sprint related activities** incl. sprint goals, user stories executed etc. for future reference
- Analysis on **team performance trends** e.g., sprint velocity over time, for better capacity estimation and planning for future sprints

AGILE DELIVERABLES

Agile Deliverable – Outcome



Phases and phase gates across the PMM lifecycle (with vendor)

CSM + Scrum Master + PO (BU lead) + Developer Manager (Change Owner)

- Checklist for Agile/Waterfall
- Discovery Workshop I with BBL internal
- Deep dive with shortlisted vendors (if vendor related)

PMO + Scrum Master + PO (BU lead) + Developer Manager (IT Team Lead)

- Discovery workshop I and II

Agile Stage Gates are managed within Agile (scrum team), typically by Product Owner

Key Deliverables Project Mangagement

Tools (e.g on Miro, Board)

- Functional requirements
- Non-functional requirements
- Team Charter / Team Norm
- Functional Design & Technical Design, Conversion Design
- Architecture and Application Design

Paper – Due to regulatory

- Criticality Assessment (CA)
- ITTP Risk and Control Assessment (RAT)

Tools (e.g on ADO)

- *(Continues update)* Functional Design ,Technical Design, Conversion Design
- *(Continues update)* Architecture and Application Design
- Product backlog , Sprint backlog, Working software, Sprint Report
- Release notes
- RAID log,
- Test Result (All testing activities) , Dress Rehearsal Result
- End User Manual and Operation Manual

Paper – Due to regulatory

- Implementation Plan
- Implementation Readiness Checklist
- CAB Documentation
- IT Risk Analysis
- IT Risk Analysis Report for BOT
- Go-Live memorandum

Expected outcomes at each phase of Agile delivery (1/2)

Agile Sessions	Discussion	Deliverables/Outcomes	Format
Discovery #1	• High-level understanding of: – Customer needs, Business stakeholder needs, Management needs – Project constraints – Integrated system (dependencies) owner's needs • Review of any existing solutions that are similar	• Functional requirements • Non-functional requirements • Project constraints • Risks, Issues, Concerns of this project	• Requirements list • Constraints list • Risks list
	• Definition of Successful Outcome of the project • Agreed scope definition to satisfy successful outcome	• Measurable success statement/definition • Scoped requirements – what's in scope, what's not in scope	• Meeting minutes
	• Alignment of the approach and objective	• Agreed ways of working (incl. Agile ceremonies and attendees)	• Sprint Calendar
		• Initial Incremental Release approach (what would be order of release for the core functionalities at the first few releases)	• User Story Map
		• Action items and To-dos (especially to prepare for Discovery #2)	• Action items list
Discovery #2	• High-level solution design • Technical work tasks • Unknowns and technical risks to be addressed	• Non-functional requirements (infra and tech focused)	• NFR list
		• Updated/refined high-level solution design (including requirements from dependencies)	• Wireframe (UX, UI) • Sequence Diagram • Architecture Diagram
		• Approval for Infrastructure needs	• Meeting minutes
Sprint Zero	• Team mission and norms • Refinement of Wireframe • Refinement of Sequence Diagram • Refinement of Architecture Diagram • Backlog Refinement	• Team charter • Wireframe (updated) • Sequence Diagram (updated) • Architecture Diagram (updated) • Detailed User Stories/Work Tasks (Tech) for 1-2 Sprints • Sprint Definition of Done, Release DoD • Environments ready (e.g., SIT, UAT) • Tools setup (e.g., DevOps, CI/CD, etc.)	• Ready tools and environments • User Stories • RAID log • Work Tasks (Tech)

Expected outcomes at each phase of Agile delivery (2/2)

Agile Sessions	Discussion	Deliverables/Outcomes	Format
Sprint Planning	• Sprint backlog (committed user stories)	• Sprint backlog (committed user stories) • Subtasks to complete User Stories	• User Stories • Work Tasks (Tech)
Backlog Refinement	• Product backlog prioritization • Refining User Stories and conduct estimation • Answers any questions that remaining	• Product backlog (updated) • Refined User Stories (incl. estimation of story points)	• User Stories • Work Tasks (Tech)
Risk and Issue meeting (RAID)	• Identify and assess risks • Seek support for outstanding risks (if needed)	• RAID log • Risks to be escalated (if any)	• RAID items
Sprint Review	• Working software, changing needs, goals met (unmet) • Show proof of work for the past Sprint • Post Sprint review, to generate Sprint report	• Working software • Sprint Report	• Software • Azure DevOps • PPT
Sprint Retro	• Reflection on the past Sprint	• List of action items (if any)	• Collaboration tool (e.g. Miro)
Scrum of Scrums	• Discuss work items across dependencies	• Work to be delivered + committed time	• Scrum of Scrums board
SIT	• All dependencies complete tests related to new system integration	• Output of test result (automated output[1])	• Test logs (automated[1])
UAT	• Users or business representatives' complete tests related to UI/UX	• Output of test results (incl. screenshots)	• Test logs (automated[1])
Product Release	• Product Owner reviews and approves the release	• User stories (resolved) • Release notes • User training	• Azure DevOps (SharePoint/Wiki)

[1] Manual documentations are highly discouraged, but may be used on exception basis (e.g. no test automation, tools not supported)

THANK YOU