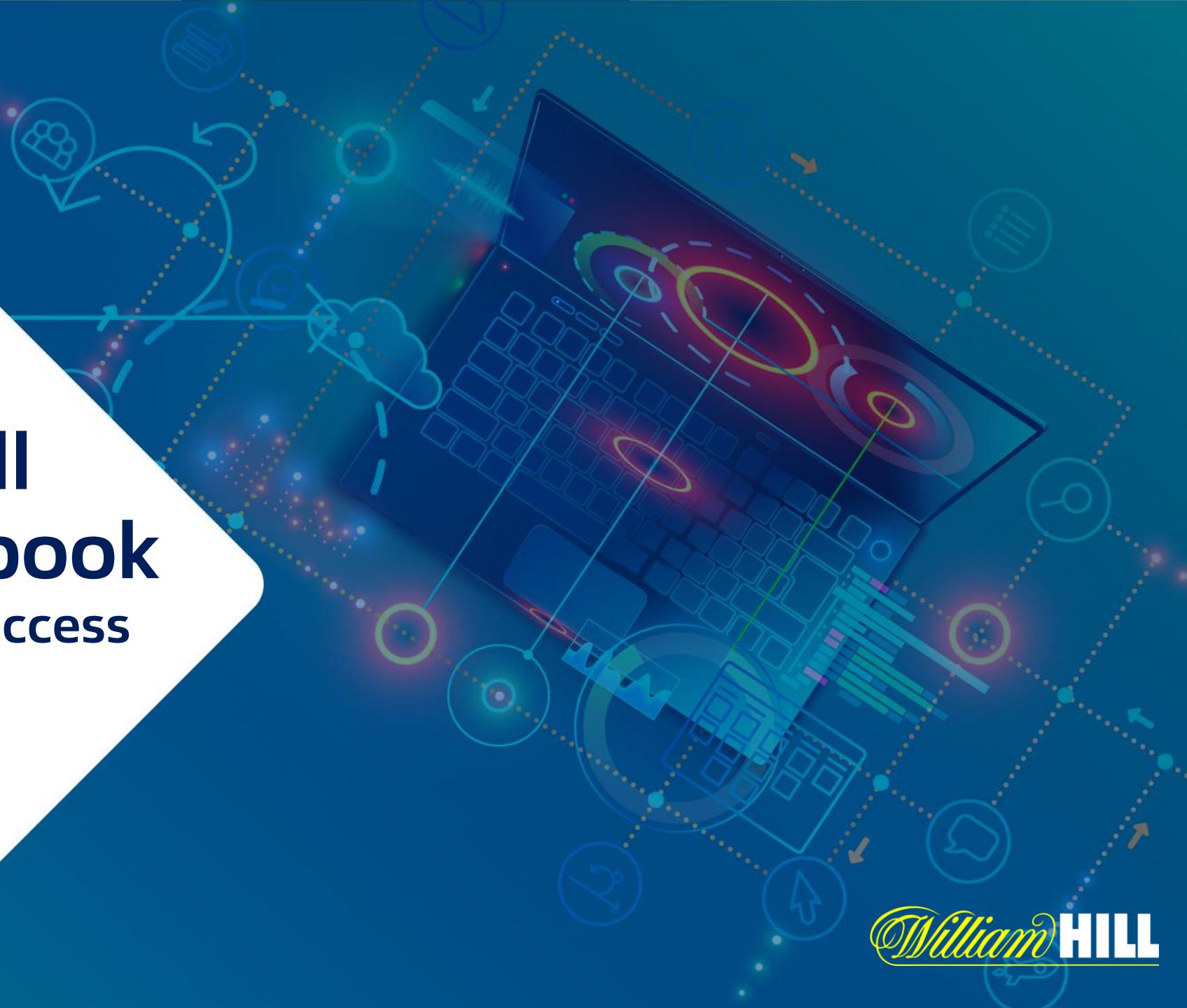


William Hill Agile Playbook

Setting up for success

v2.0.0

February 2019



William **HILL**

Contents

Chapters

Introduction, Vision & Objectives

Chapter 1- Product Dev Team Roles & Responsibilities

Chapter 2 - Team Norms

Chapter 3 - Scrum Events

Chapter 4 - Communities of Practice

Chapter 5 - Product Ownership

Chapter 6 - Measuring for continuous improvement

Chapter 7 - Governance

Intended Audience

Everyone

Everyone

Everyone

Teams

Everyone

Teams

Scrum Master, PM, Delivery Lead, PO

Scrum Master, PM, Delivery Lead, PO

Introduction to the William Hill Agile Playbook

Here at William Hill we aspire to have the best software product development around to give our customers the best sports betting and gaming experiences in the market. To do this "at scale" we need to be passionate about self-improvement. Doing more of what works and challenging anything that can and should be improved.

This is a **foundational document** and **toolkit** for any new starter at William Hill. Also a reference point for teams if they need to do any course corrections or refresh their understanding of how we should work.

Amongst other things it highlights things that we are **passionate about** and that are **important** to us. Its these key areas where it's vital to achieve consistency, as these are what will allow us to scale our agile ways of working. It is also meant as the "North Star" in terms of "what good looks like".

Finally, **It is ours.**

It was refined based on our feedback, our experiences, successes and most importantly failures.....

This is how we grow and learn.

It is a living document which is meant to support us during the Agile journey and it will be inspected and adapted whenever it is needed. **Because it is ours.**

**“Empowered, collaborative teams
putting customers first”**

William Hill's Agile@Scale Vision

Objectives

To drive WH product development to improve (quality and speed to market) across the digital portfolio.

To operate with **transparency** in and around software product development to surface issues and fix them.

Accelerate our cultural transformation with **continuous improvement being the aim**.

To build productive and engaged teams that **want to commit and deliver** exciting products for William Hill customers.

To be a **centre of excellence for agile software product development** that other organisations aspire to.

To change the culture and the way we work. If something can be improved **its on us to improve it**.

To deliver a scalable Product Development model that's based on the agile methodology with **productive** and happy teams that **we are proud of**.

To be **passionate about key areas of consistency that will allow us to scale**.

To embrace the use of Agile and apply them with **skill and knowledge to the correct situations**.

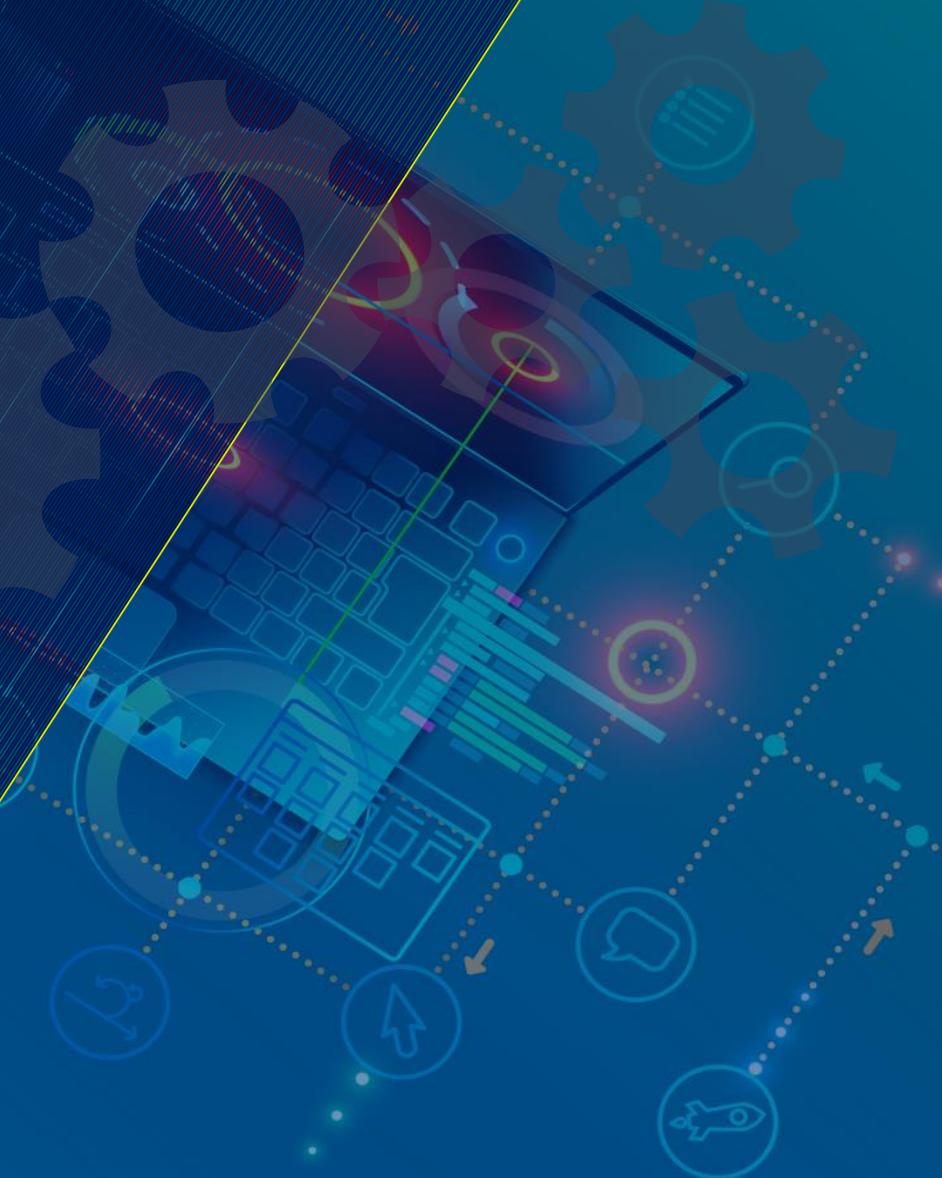
This we hope is useful as a guide, a reference and a toolkit for all members of the William Hill team across locations and divisions. It is focused on Product Development, but we embrace agile ways of working wherever they can add value. It should also present some great opportunities for those that embrace positive change.

Chapter 1

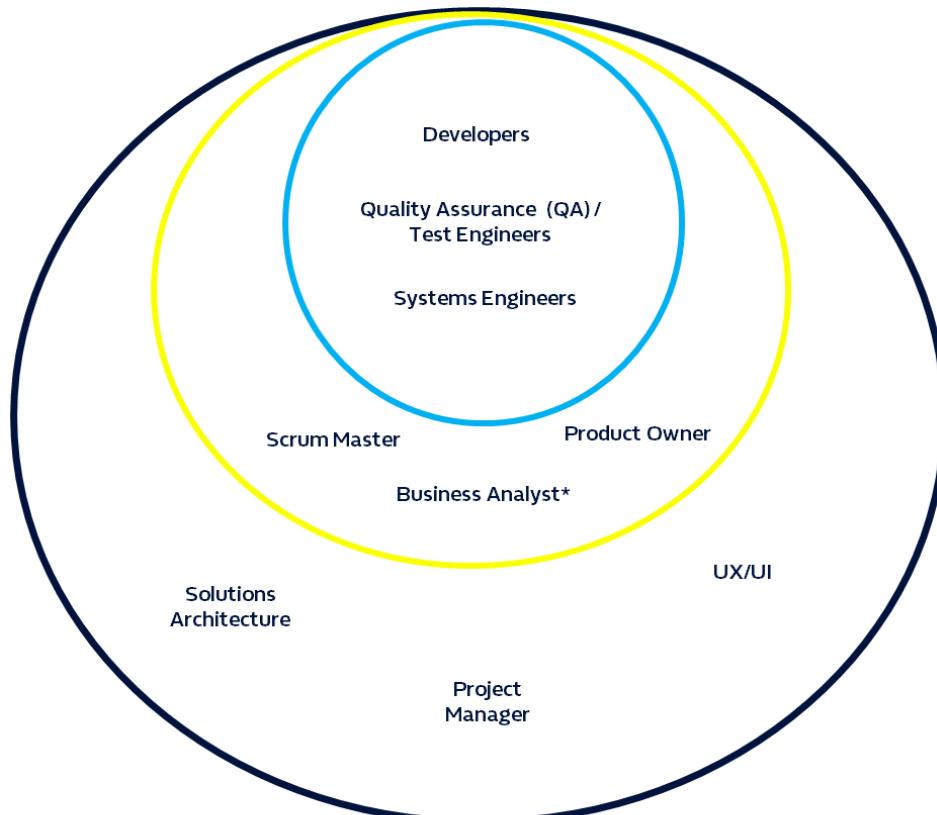
Product Dev Team Roles &

Responsibilities

Everyone



William Hill Product Development Team



*Digital don't have Business Analysts

Development Team

Development teams put quality at the heart of everything they do. They focus on built in quality during the entire process of software development. They do not consider testing and development as separate activities. They think of testing as an activity to prevent defects opposed to finding them.

They understand the SDLC (Software Development Life Cycle) and should endeavour to have complete control of their software delivery, including deployments and production support.

Scrum Team

The Scrum Team is a self-organising unit that is responsible realising value that meets the business objectives and goals.

They work from a product backlog to identify priorities, refine detail and a sprint backlog to manage the production of quality products. Using sprint goals to drive towards a common goal as a collaborative team.

Product Development Team

The Product team include all organisational roles required to make the Scrum Team successful. The scrum team know when to engage these team members as part of every software product delivery. Solutions Architecture, Project Management and UX/UI.

Whereas the Scrum Master focuses on the Scrum team the Project Manager concentrates on organisational roles and dependencies.

Some scrum and product roles are shared across multiple teams This can be 1-to-2 or 1-to-many (<4) teams depending sub-channel setup

Role	Level	Summary regarding the Product Team (Not a definitive Job Description)
Product Owner	Scrum	Owns Product, empowered to make decisions. Communicates vision, roadmap and progress. Manages the users expectations and decisions/communications on go-live dates. Is accountable for the Epics, User Stories & Acceptance criteria and orders and prioritises the product backlog. Using data and customer research to drive decisions.
UX / UI	Product	Designs and documents innovative user experiences and detailed user interfaces for developers to work from. A service to the team or part of the team depending on the nature of the product development.
*Business Analyst	Scrum	Makes sure stories have the right level of detail. Analyses business impact of product features. Responsible for generating and expanding User Stories, from the Product Roadmap, the UX and Architecture documents. Responsible for detailed Acceptance Criteria (for further elaboration by QA). Communicates functional and non-functional requirements.
Project Manager	Product	Co-ordinates the relationship with external teams and mitigates risks, assumptions, issues and dependencies (RAID). Forms the key support for PO and technical leaders in the teams ensuring all mandatory deliverables are identified, met and achievable. Makes sure release dependencies are taken into account. Helps Scrum Masters unblock impediments they are unable to resolve alone.
Solution Architect	Product	Collaborates with the Scrum Team to agree high level design, ensuring compliance with Architectural Guardrails and target architecture. Has non-functional and impact assessment accountability of the solution.

*Digital don't have Business Analysts

Continued....

Role	Level	Summary regarding the Product Team (Not a definitive Job Description)
Scrum Master	Scrum	Helps the team to unblock things they cannot themselves. Owns the agile process and coaches the team in self-organization and cross-functionality; facilitates communication, process efficiency and continuous improvement. Works with Development to ensure the SDLC is followed and agile ways of working are optimised. Ensures all scrum events and key areas of consistency for Agile@Scale are followed.
Systems Engineer	Development	Builds technology solutions to operational challenges (e.g. CI/CD Pipeline), manages Cloud Infrastructure to support team (e.g. creating environments)
Developer	Development	The developer creates solutions to appropriate quality and functional standards and writes unit tests. They work with QA from the outset, ensuring that all test scenarios and cases are considered whilst building the solution. Developers try to unblock issues that prevent the team from meeting sprint goals. They work with Scrum Masters on impediments they cannot resolve.
Quality Assurance (QA / Software Test Engineer)	Development	<p>QA are instrumental in the early design & sizing of solutions. They ensure solutions are robust and that the acceptance criteria is complete, testable and comprehensive – this includes both Functional and Non-functional Requirements.</p> <p>Their early involvement and understanding ensures fringe cases and exception handling is catered for from the outset of software builds. This allows automated test scripts to be created in parallel with the other development activities.</p> <p>There should be no handoffs between Software Engineers and Quality Engineers they should work together and in parallel as all engineers are responsible for quality.</p> <p>QA try to unblock issues that prevent the team from meeting sprint goals. They work with scrum masters on impediments they cannot resolve.</p>

Project Manager

- Works with the Product Owner to radiate information
- Supports delivery
 - Uses Agile methodologies to deliver
 - Collaborate with the Product Owner to manage stakeholders
 - Help resolve external dependencies
 - Supports planning of the roadmap
 - Team profiling, capability and priority alignment

**Efficient team
producing value
in delivery**

Scrum Master

- Encourages self-organisation, focuses on the team
- Oversees the Scrum process
 - They Coach the team
 - Focuses on efficiency and continuous improvement
 - Facilitates communication
 - Helps resolve internal blockers & dependencies

Responsibilities

Project Manager

Finances	3rd Party Vendor Management	Time
<ul style="list-style-type: none"> Tracks delivery budget Manages internal and external costs 	<ul style="list-style-type: none"> Attends Service Reviews Liaison on future Tech (Roadmaps) 	<ul style="list-style-type: none"> Delivery Timeline Logs and measures the team's time
External Blockers	Dependencies	Team
<ul style="list-style-type: none"> Helps resolve Manages Escalates 	<ul style="list-style-type: none"> Cross team/channel management 	<ul style="list-style-type: none"> Escalation of future need for additional people Holiday planning
Pre-Planning (support PO)	Stakeholders	Raid Log
<ul style="list-style-type: none"> Delivery requirements High Level planning / roadmap 	In collaboration with the PO: <ul style="list-style-type: none"> Updates / Status Escalations Wider org comms Manages expectations Manages 3rd parties 	<ul style="list-style-type: none"> Mitigation planning Capturing future risks
Agile Delivery		
<ul style="list-style-type: none"> Takes an Agile first approach Embracing the Agile mindset 		

Scrum Master

Inspect and Adapt	Team
<ul style="list-style-type: none"> Coaches the team to identify areas of improvements Drives a continuous improvement approach 	<ul style="list-style-type: none"> Supports the team members Builds team relationships and trust Protects team from external interference
PO Support	Metrics
<ul style="list-style-type: none"> Story management Facilitating refinement Facilitating reviews 	<ul style="list-style-type: none"> Consistent usage for KPI reporting Sprint goals are captured and reviewed
Internal Blockers & Dependencies	Agile Mindset
<ul style="list-style-type: none"> Helps Resolve Escalates 	<ul style="list-style-type: none"> Coaches Agile principles and values to team members and leaders Coaches Agile practices Aims for self organisation & team empowerment Supporting DevOps culture
Scrum Events	
<ul style="list-style-type: none"> Books/arranges Facilitates Allows team to own 	

Other sub-channel roles*

Role	Level	Summary regarding the Product Team (Not a definitive Job Description)
Head of Product	Sub-Channel	Responsible for the sub-channel product roadmap and ultimately WHAT the teams deliver.
Head of Technology	Sub-Channel	Responsible for all tech members of the sub-channel team and essentially HOW the teams deliver. E.g. SDLC and team process.
Head of Delivery	Sub-Channel	Responsible for the portfolio management of the sub-channel (the programme). E.g. Finance, resourcing and overall RAID.
Release Analyst	Sub-Channel	Responsible for coaching the teams on release management best practice. Supporting release planning where appropriate. Leading retrospectives for product issues following problematic releases. Responsible for approving (amongst others) production releases that impact other products and services.
Infrastructure Architect	Sub-Channel	Works with the solutions architect and the team to ensure the infrastructure platform required for the product delivery is designed correctly. Responsible for deployment views and interfacing with other infrastructure teams as part of the solutions delivery.
Service Support Analyst	Sub-Channel	A bridge between operational support and product development. Helps raise understanding in the teams of production issues and operational problems. Feeds issues into the product development teams for delivery and provides a service interface for the operational business.
Security Architect	Sub-Channel	The predominant objective of the team is to increasingly integrate security into the fabric of the digital business as early in the design process as possible. The security architecture team is a "Centre of Know" rather than a "Centre of No." Works with the whole product delivery team during the design phase to ensure the discipline and associated process of planning and designing organizational, conceptual, logical, and physical components that interact in a coherent fashion, aligned with business requirements, in order to achieve and maintain a state of managed security-related risk.
Product Managers	Sub-Channel	

* not a definitive list

"we are all part of a delivery team"

Chapter 2

Team Norms

Everyone

William **HILL**



Each team should aspire to pragmatically implement their own team norms with the support of their team's Agile champions*

	Team Charter	Definition of Ready	Definition of Done
Definition	The Team Charter captures the behaviours and ways of working to which all members of the team aspire. Can range from how meetings should be run to how to accommodate individuals' working preferences and non-work commitments. Agreed when team is formed and checked regularly, particularly when any new members join	The Definition of 'Ready' should be agreed by each team but MUST exist, and sets the standard that all requirements should meet before Stories can be included in a Sprint , so that Stories that are included can be completed within the Sprint. Ensures all stakeholders have had a chance to input into the requirement Should follow INVEST principles (Independent, Negotiable, Valuable, Estimable, Small, Testable)	The Definition of 'Done' should be agreed by each team but MUST exist, and represents the criteria that a Story should meet before it can be considered finished at the end of the Sprint. It ensures the team are consistently estimating for all tasks to take a Story to completion, and that Product Owner and the team are aligned on what it means for a Story or task to be complete
Example	<ul style="list-style-type: none"> Our meetings will start and finish on time We eat lunch together once per week During team calls, one person talks at a time – we all listen We're not afraid to take risks. There's a no blame culture If I cannot resolve a problem in 30 min then I will ask for help <p>Real Teams examples</p>	<ul style="list-style-type: none"> Acceptance Criteria Set UI designs finalised Story reviewed by both Dev & Test representatives Estimates done by consensus Story reviewed by Analytics where appropriate (e.g. registration optimisation) External dependencies identified & mitigation put in place to reduce their likelihood of impeding the Sprint <p>Real Teams Examples</p>	<ul style="list-style-type: none"> Code developed Unit tests developed Code has been peer-reviewed Acceptance Criteria tests passed (as defined in the User Story) or exceptions and risks approved by PO End-to-end front-end tests passed Any additionally identified tests are passed (e.g. performance, security)
Minimum Viable Definitions	<ul style="list-style-type: none"> On time to meetings Respect and listen to each other Shared working agreement 	<ul style="list-style-type: none"> Acceptance Criteria Set UI designs finalised User Story reviewed by team Estimated (sized) External dependencies known Solutions Architecture agreed 	<ul style="list-style-type: none"> Code has been peer-reviewed Deployed into an environment (ideally to Production) Tested (acceptance criteria met) Non-functional's verified (if required) Regression test passed Product and UX approve

Technology standards (e.g. test coverage, infrastructure needs) to which the team aspire and which are necessary for the team to begin flying.
Examples are available tools/applications, test automation coverage, environment/infrastructure expectations and continuous integration/deployment expectations
(e.g. Unit test coverage at a team acceptance level, new development environments can be setup within 1 day)

Chapter 3

Scrum Events

Teams

William **HILL**



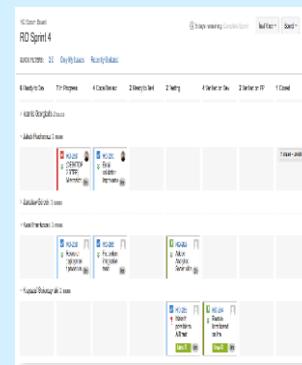
Scrum Events - Overview

Backlog Refinement
(Shared understanding of vision and Product Backlog Items)

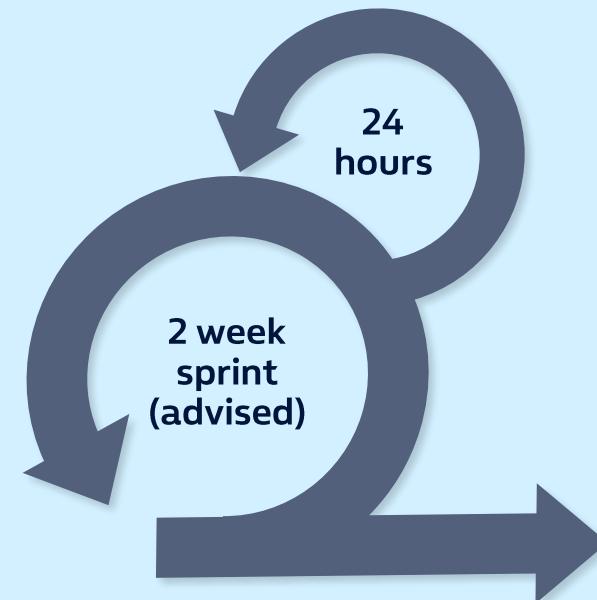


Definition of Ready

Sprint Planning
(Do we have a solid goal?)

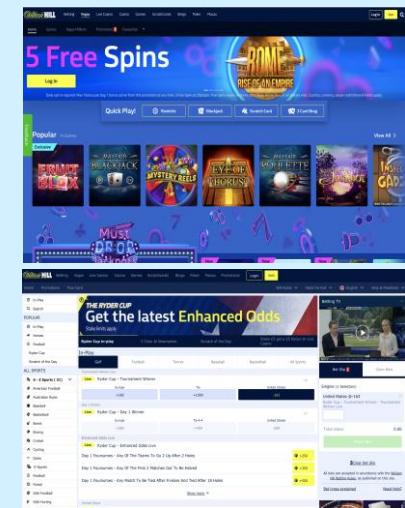


Daily Scrum

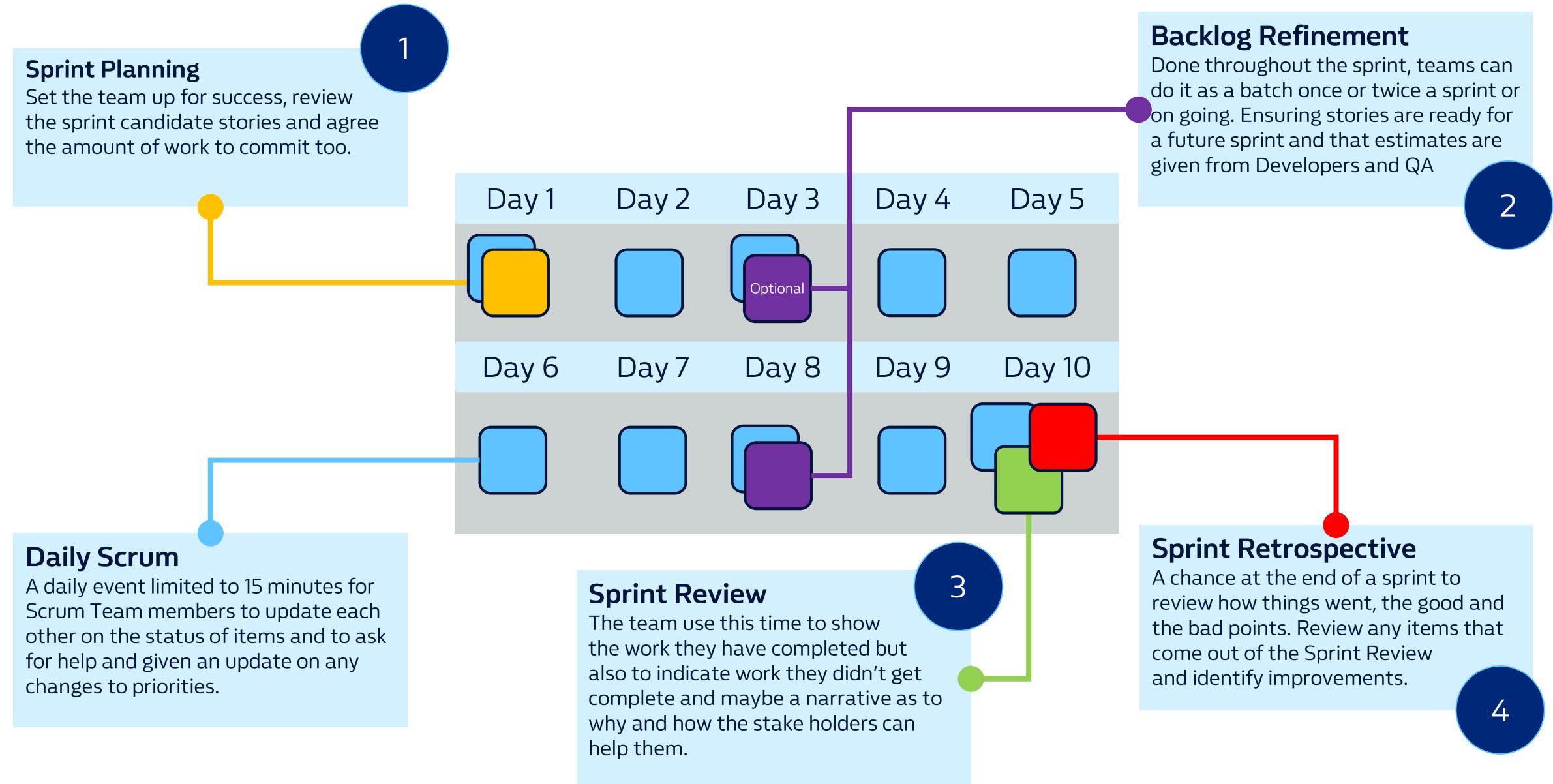


Definition of Done

Sprint Review
(Did we deliver value?)



Sprint Retrospective
(What can we improve?)



1

Sprint planning

The Sprint Planning meeting is a collaborative session to plan the work the team can do in the sprint. It is first event in every sprint.

Who are the attendees?

- The Product Team*
- SME's as required (Subject Matter Expert)

Guidelines

- Held at the beginning of each Sprint – guide is maximum of 4 hours for a 2 week Sprint
- Product backlog is used to facilitate the discussion
- Candidate stories for the Sprint meet the Definition of Ready
 - Has acceptance criteria and is sized
- Sprint Goal describes the objective for the Sprint
 - Selected stories deliver a coherent Sprint Goal or Goals
 - Goals can include anything that encourages the Development Team to work together
 - Provides flexibility for the team to adapt functionality as necessary to deliver the Sprint Goal
- Only the Development Team can assess what is achievable in the Sprint

Inputs

- Product Backlog
- Latest Product Increment from the previous Sprint
- Actions from the previous Sprint Retrospective
- Team Capacity and Velocity

Process

- What can we do
 - The Product Owner discusses the objective of the Sprint and the stories that would contribute to achieving it
 - A Sprint Goal is collaboratively created to provide guidance to the development team during the Sprint
- How will we do it
 - The development team select which stories can be "Done" during this Sprint
 - Each story is discussed in terms of dependencies, design, development and testing
 - When the Development Team reach their capacity no more stories are planned
 - The Development Team and Product Owner collaboratively review and make trade-offs
 - The Development Team confirm and describe how they will accomplish the Sprint Goal

Outputs

- Sprint Goal
- Estimated Tasks
- Sprint Plan
- Sprint Backlog

* William Hill Product Development Team

Backlog Refinement

Backlog refinement meeting is used to improve the quality of the backlog: understand and further define Stories and acceptance criteria, estimate work together with Devs and QA's and generally prepare the sprint backlog to ensure effective sprint planning

Who are the attendees?

- Product Owner
- Scrum Master
- Scrum Team* (not mandatory for all Scrum team members to attend)

Guidelines

- Product owner prepares and runs the meeting
- Suggested the team is involved to ensure understanding of the future work
- Usually 1–2 times during the Sprint, but can happen anytime
- Estimates are based on a team consistent effort story pointing.
- Estimates come from a combined involvement of all disciplines in the Development Team

Inputs

- Product Backlog
- Feedback from customers, users, and stakeholders
- Updated Stories or UX/UI designs

Process

- Team understands new Stories & information
- Team discusses development & testing approach and identifies technical spikes
- Team identifies need for further analysis or definition
- Initial or revised estimates done

Outputs

- Refined Product Backlog & actions to be ready for effective Sprint Planning
- Improved backlog quality (better % estimation or more 'Ready' Stories)
- High level idea of Sprints+1 and Sprints +2

3

Sprint Review

The Sprint Review meeting is used to demonstrate what was accomplished during the Sprint and review next things that could be done in the following one to maximize value. The main purpose of the session is to gather the feedback about current version of the Product from stakeholders and revise if the current direction of travel is correct.

Who are the attendees?

- Product Development Team*
- Stake Holders
- Sponsors

Guidelines

- Held at the end of each Sprint
- 1 hour for a 2 week Sprint
- Very informal, preferably with minimal to no slides
- Better in person where possible
- Held at the same time & day every 2nd week
- The Sprint Review should provide valuable input to subsequent Sprint Planning
- Consider recording the session and sharing it with the whole channel
- Important to celebrate success – the team worked hard to meet the goal!

Inputs

- Software that meets the Definition of 'Done' (Increment)
- Sprint Goal/Backlog
- Product Roadmap

Process

- Scrum Team present against the sprint goal (was it achieved?)
- Product Owner present Stories committed to and completed
- Dev Team demonstrates new features (Stories completed in the Sprint)
- Scrum Team seeks feedback from stakeholders about what was built during the Sprint
- Scrum Team review the current direction of travel with stakeholders

Outputs

- Updated Priorities (Product Backlog)
- Updated Scrum board
- Updated list of blockers for resolution

Sprint Retrospective

The Sprint Retrospective is a formal opportunity for the Scrum Team to "self identify" how to continuously improve. It is the last event in a Sprint and is held after Sprint Review and before the next Sprint Planning.

Who are the attendees?

- Scrum Team* only

Guidelines

- Held at the end of each Sprint after Sprint Review - 1-2 hours for a 2 week Sprint
- Scrum Master ensures the meeting is positive and productive
- Retrospective needs to be a safe place for the Scrum Team to give honest feedback
- It is important that everyone's voice is heard
- Identify and prioritise the major items that went well and potential improvements
- Consider using the Sprint Backlog to add your improvement actions
- Ensure the changes are considered in the next Sprint Planning
- Anything the team cannot resolve should be raised as an Organisational Impediment

Inputs

- Sprint Metrics
- Sprint Burn down
- Sprint Timeline

Process

- Set the stage
 - Welcome everyone to the retrospective meeting
 - Establish the rules of engagement
- Gather data
 - What went well? Start positive!
 - What could be better?
- Generate Insights
 - How could we make things better?
- Decide what to do
 - Agree which things we will do in the next sprint
- Close the retrospective
 - Thanks everyone and confirm the actions and owner

Outputs

- At least 1 actionable outcomes to improve the next Sprint (less than 3)
- Actions and owners
- Refreshed KPI's
- Potential changes to Definition of Done or/and Definition of Ready
- Organisational Impediments raised:
<https://jira.willhillatlas.com/browse/ORGIMP>

Daily Scrum

The Daily stand-up is a 15 minute time boxed event for the Development team. The Development team uses the Daily stand-up to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. At it, the Development Team plans work for the next 24 hours. Every day, the Development Team should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment by the end of the Sprint. This is a key inspect and adapt meeting.

Who are the attendees?

- Scrum Team* members
- Stakeholders are invited (Only Development team members share updates)

Guidelines

- Held at the same time and place each day to reduce complexity
- Time-boxed to 15 minutes
- Focus on how the Development Team is progressing towards the Sprint Goal
- The Daily Scrum is an internal meeting for the Development Team. If others are present, the Scrum Master ensures that they do not disrupt the meeting
- Not for problem solving - The Development Team or team members often meet immediately after the Daily Scrum for detailed discussions, or to adapt, or re-plan, the rest of the Sprint's work
- Daily Scrums improve communications, eliminate other meetings, identify impediments to development for removal, highlight and promote quick decision-making, and improve the Development Team's level of knowledge.

Inputs

- Sprint Goal
- Scrum board
- Sprint Burn-down

Process

- The structure of the meeting is set by the Development Team and can be conducted in different ways if it focuses on progress toward the Sprint Goal. Some examples of what questions we may ask:
 - What did I do yesterday to meet the Sprint Goal?
 - What will I do today to meet the Sprint Goal?
 - Do I see any impediment that prevents us from meeting the Sprint Goal?
- Or simply
 - "Will we achieve our Sprint Goal?"
 - "What might block us?"
 - "What must we do to achieve it"

Outputs

- Updated Scrum board/new tasks
- Updated list of impediments for resolution
- Updated Sprint Burn-down

Tools supporting Agile@Scale

Role	Tool(s)
Roadmap / initiatives	A-Ha
Product feature requirements	Confluence A-Ha
User stories / task tracking	JIRA
Collaboration	Slack Blue Jeans
Knowledge sharing	Confluence SharePoint A-Ha
Source control	Gitlab
CI/CD pipeline	Jenkins Gitlab CI

Chapter 4

Communities of Practice

Everyone

William **HILL**



"Communities of practice" (CoPs) are groups of individuals with common skills. They sit across teams and facilitate knowledge sharing and good practice



Communities of Practice (COP's) are a logical grouping of individuals with common functional skill sets i.e., Scrum Masters, UX designers, etc.

They...



Develop and maintain forums in which each functional role can share knowledge and good practices



Hold events to promote interaction across function



Setup support structures for 1:1 mentoring, buddying and coaching



Refine William Hill Playbook based on observations and experiences



Ensure guidelines are set and maintained across the CoP



Intended Benefits of Communities of Practice (COP's)

Enable Channels to become autonomous, but remain aligned

Upskill employees faster

Promote agreement on good practice

Facilitate rapid responses to problems

Initiation of new topics and innovation

Provide access to a knowledge hub

Provide training and coaching

Communities of Practice (CoP's) are led by a community leader who is responsible for running the community but may not have any formal line management responsibilities

Communities use a range of tools and practices to ensure members have the information and connections they need



The community itself should agree the format and frequency of meetings, and the tools they want to use to facilitate collaborative working

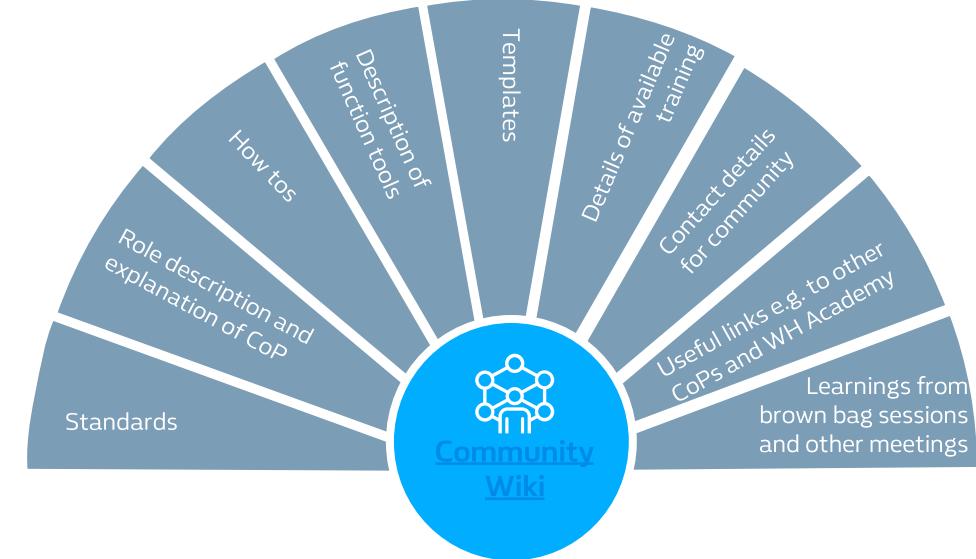


The community leader is responsible for ensuring that the agreed meetings and communications take place, although they may delegate responsibility for individual elements to willing members of the community



The function leader is responsible for standard setting, but should gather ideas from the CoP and communicate updates via the CoP channels

The community Wiki is the repository of community knowledge, with other channels used for sharing information



Slack, JIRA, and the community meetings are three other tools for sharing information



JIRA can be used to raise ideas and flag new innovations. If these lead to new standards or "how-tos" they can be added to the Wiki



Slack is an easy, quick and informal way to ask questions of the rest of the community



The **community meetings** provide a regular update on key changes and innovations across the practice

CoPs are built “bottom up” for and by its members - everyone is expected to contribute

Members of a community of practice should be expected to



Attend



Be engaged



Champion the community



Share ideas and good practice and contribute content



Support and mentor other team members

Review market trends and changes in technology outside of WH.
Bring new ideas back into the community so we're at the forefront

Lead and support the implementation of agreed good practice into the teams

Use other disciplines where required for advice and expertise e.g. dev and test working more closely on tech solutions

Innovate: shape the CoP to best serve the common good (e.g. badge framework in QA CoP)

Where can I find information about CoP's and how to get involved?

CoP Wiki: <https://conf.willhillatlas.com/display/COP>

To get involved you can also search for the CoP Slack channels that all start with #cop

Chapter 5

Product Ownership

Role of the Product Owner



To be successful, Product Owners require a holistic set of characteristics across multiple dimensions



The PO does

- Have ultimate responsibility for the success or failure of a Product
- Have the final say on deliverables
- Understand business needs, aims and objectives and translate them into Product deliverables
- Understand customer needs, aims and objectives and translate them into Product deliverables
- Use data (internal, market, customer, competitor) to drive decision-making and priorities
- Sign-off Stories as 'Done' on behalf of the user against agreed acceptance criteria
- Accountable for the team having 'Ready' work items to take into a Sprint
- Have to balance delivering business value whilst overseeing bug triage and ensuring technical teams priorities are incorporated (architectural frameworks, reducing technical debt, infrastructure etc.)
- Act as an enabler, unblocking issues where appropriate
- Communicate vision, roadmaps, releases, success stories and business outcomes
- Answer questions and make decisions immediately to facilitate team productivity
- Attends the Scrum Events to participant with the teams and be available for them and lead on some of the sessions such as Backlog refinement
- Leads customer or industry research with the support of other roles and areas of the business



The PO doesn't

- Have to deliver every suggestion that has been made, no matter how passionate, talented or senior the person making the suggestion
- Manage the team



Characteristics of a good Product Owner

- Lives the business and its customers
- Focused on delivering customer / business value
- Decisive: knows when to say "no"
- Excellent working relationship with the stakeholders
- Empowered by leadership to make decisions
- Able to coordinate across the Product Owner community to ensure roll-out of delivered functionality

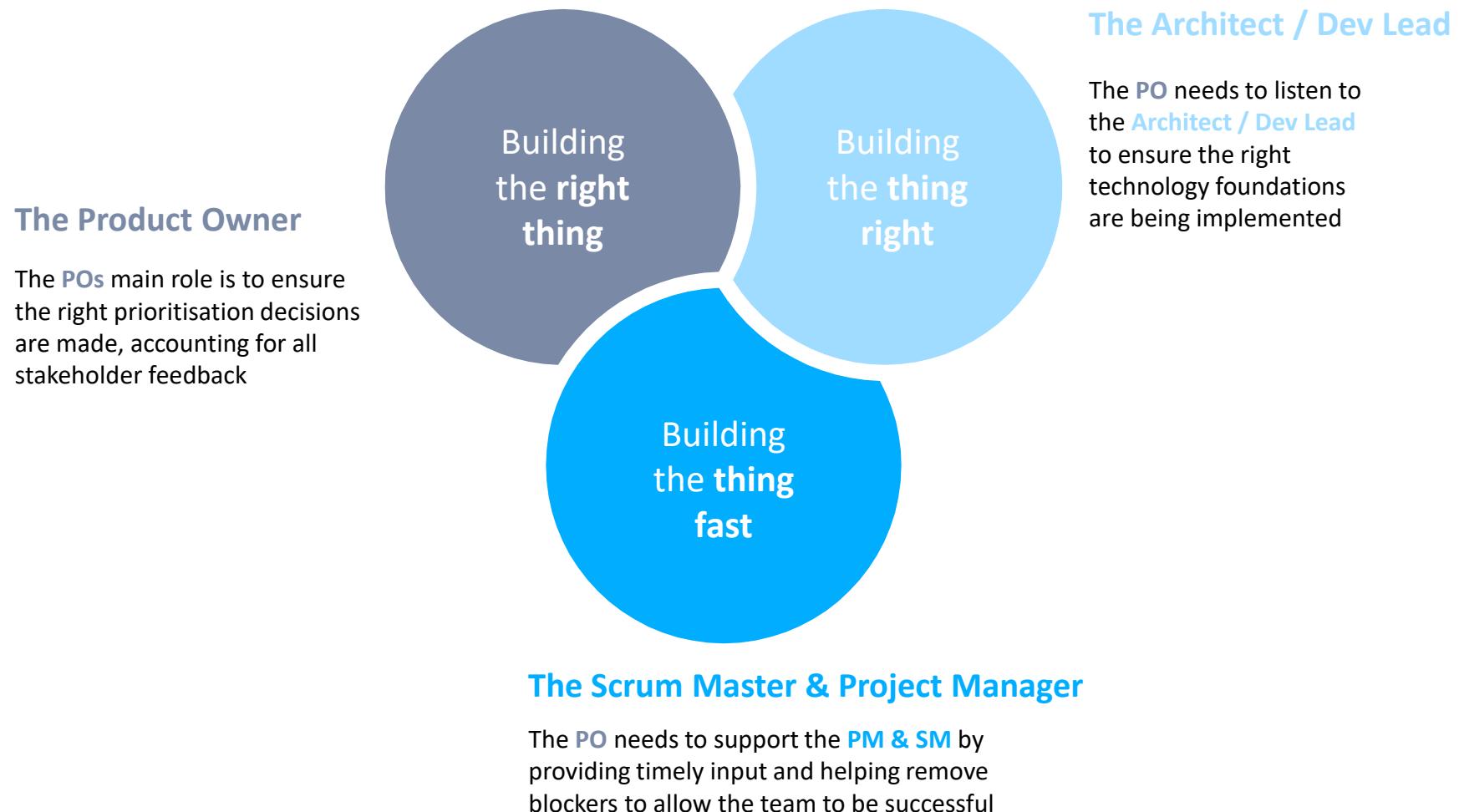


Product Ownership not working well

- Not available to team
- Defines the solution instead of the needs
- Does not understand the business / customer
- Does not prioritise based upon value to customer / business
- Micromanages the team
- PO defines the "how" instead of the what and why

Find more out more. Product Info [Link Required]

The Product Owner is jointly accountable for delivery, but is singularly responsible deciding what is delivered and in what order



To be successful, Product Owners require a holistic set of characteristics across multiple dimensions (1/2)

Turns the product vision into a tangible roadmap and aligns the team and stakeholders around the roadmap and vision

The PO demonstrates:

- Understanding of William Hill and product context
- Deep insight into business and product features & components
- Strong awareness of complimentary and dependent product roadmaps
- Ability to shape and communicate product roadmap
- Ability to navigate organisation and manage senior stakeholders
- Good use of data to drive product decisions

Directs the team play by play

The PO demonstrates:

Structured thought leadership

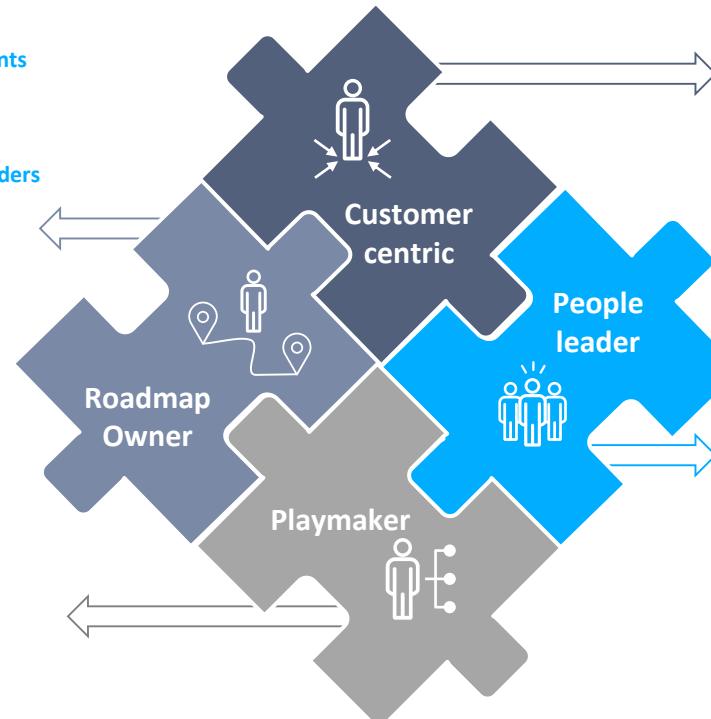
- Ability to drive prioritization decisions and lead decision making
- Structured problem solving
- Clear communication

Rigour in execution and attention to detail

- Ability to translate vision into specific deliverables
- Ability to manage demand versus supply
- Ability to plan and track delivery
- Ability to resolve issues
- Ability to adjust planning based upon feedback
- Good track record of successful delivery

Command of digitisation methodology

- Understands technology and can communicate with technical teams
- Understands need for strong technology foundations and prioritises with Tech Leads advice
- Embraces Agile way of working



Ensures engaged customers and happy stakeholders

The PO demonstrates:

- Obsession with solving customer & stakeholder needs
- Passion and commitment for optimising end-to-end customer experience
- Desire and ability to stay close to, and collect feedback from multiple customer and stakeholder groups
- Use of service and customer experience design

Inspires and leads others

The PO demonstrates:

- Exceptional communication skills
- Ability to influence people and strong negotiation skills
- Goal orientation, pro-activeness, and passion
- Openness, collaboration, and caring

Entrepreneurship and innovation

Collaborative and empathetic persona

Chapter 6

Measuring for continuous improvement

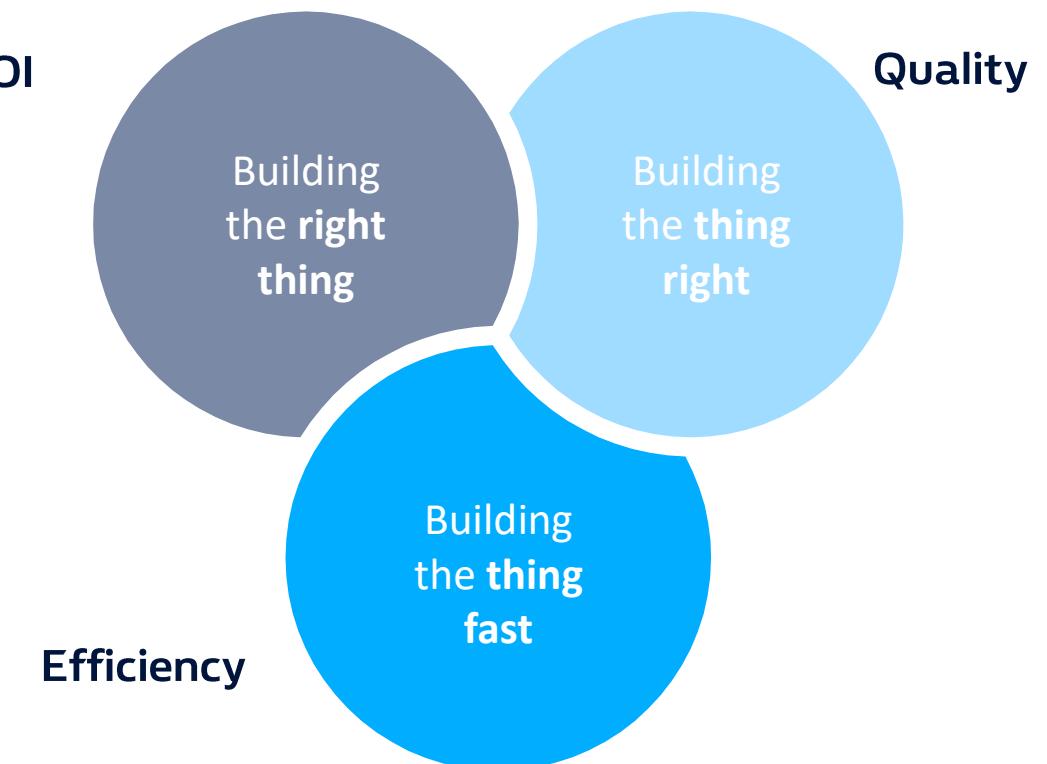
Teams



Using metrics is aimed for helping us improve our productivity, to support the continuous improvement process and help teams succeed in their mission of putting the customer first.

Measuring and reporting **Key Performance Indicators** (KPIs) is indispensable to support decision making at different levels in the organization and to assure that we are delivering:

- the most valuable products/ features from the business point of view ("**Building the right thing**") These metrics will be owed by Product Teams.
- in the most efficient way ("**Building fast enough**") These metrics will be owed by the Delivery Teams.
- stable products , of the highest quality with minimum cutting of corners to ensure we have sustainable products ("**Building the right way**"). These metrics will be owed by both Product & Delivery Teams



"If you can't measure it, you can't improve it"

KPI Metrics Categories – what we measure?

Efficiency

When we're looking at how efficient we are, we need to see where **we are improving** and where **our bottlenecks** have moved to. This will help us ensure that we are getting the **most from our teams**, with as little waste as possible in the flow.

When issues do happen in production, we need to know how efficient we are at getting the **service back for our customers** so they can start placing bets again.

For improving Efficiency we measure:

- Lead / Cycle / Flow Times
- Deployment Frequency
- MTTR (Mean Time To Restore)

Cycle Times
& Lead Time

Release Rates

MTTR

Where can I find more information about definitions and results for specific metrics?

[Agile@Scale KPI Metrics explained](#)

[AGILE KPIs and Dashboards](#)

KPI Metrics Categories – what we measure?

Quality

If we optimised only efficiency without ensuring **quality** we would be doing a disservice to our **stakeholders and customers**.

Every time we push out an update to our **products** we should be ensuring we're **improving the quality**.

When releases happen we want them to **happen more often**, and for this to work we need to ensure that are releases are **small and often** with minimal **disruptions**.

- Number of Bugs
- Number of Technical Debt
- CFR (Change Failure Rate)

Technical Debt

Number of bugs

CFR

Where can I find more information about definitions and results for specific metrics?

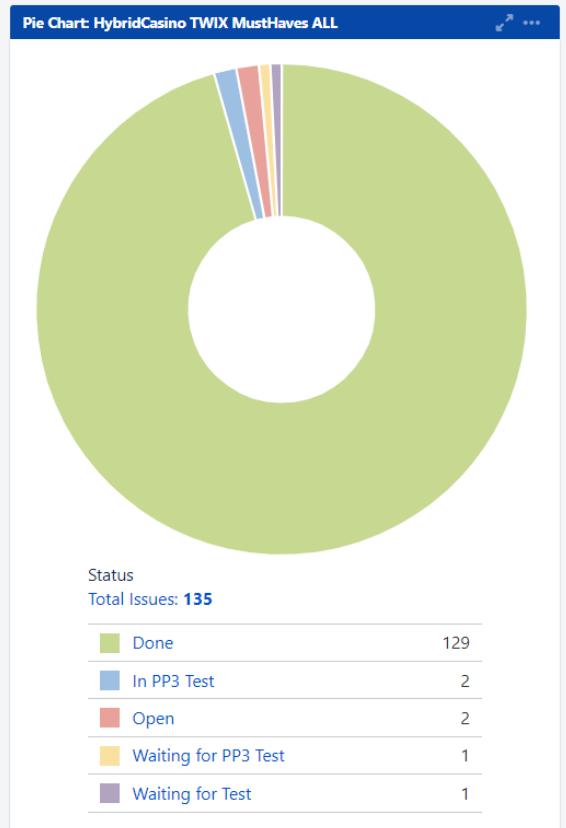
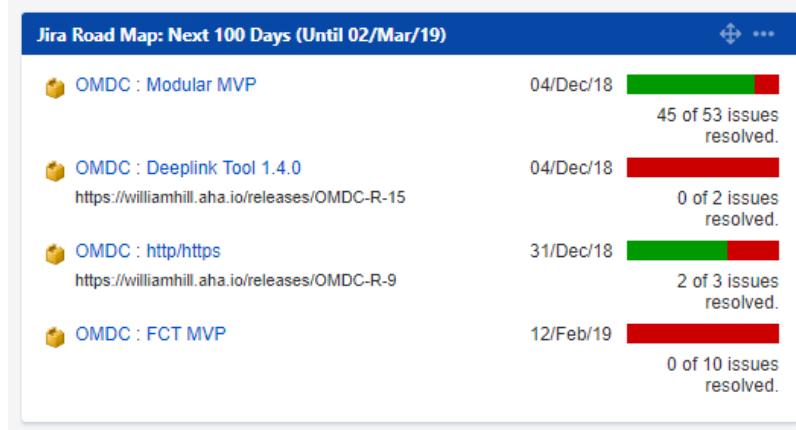
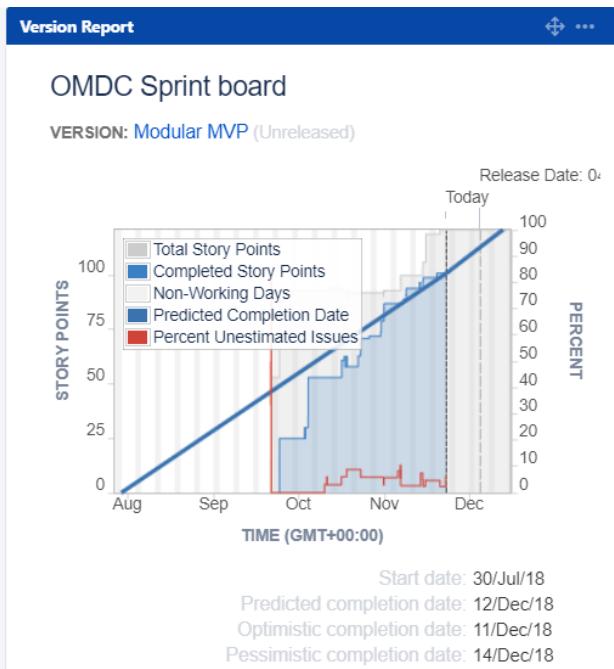
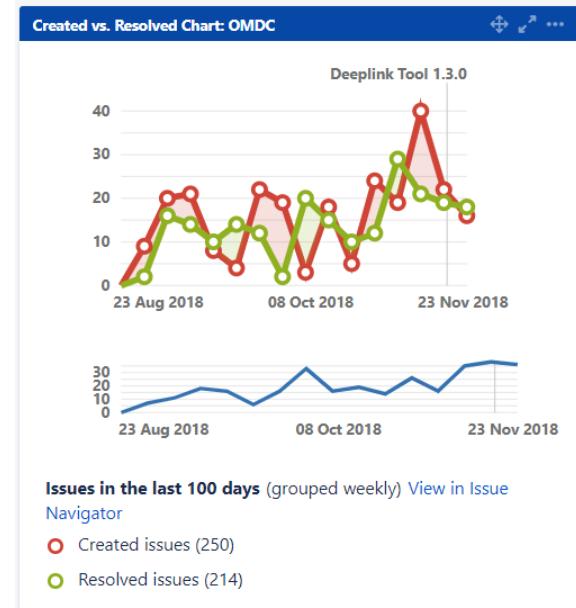
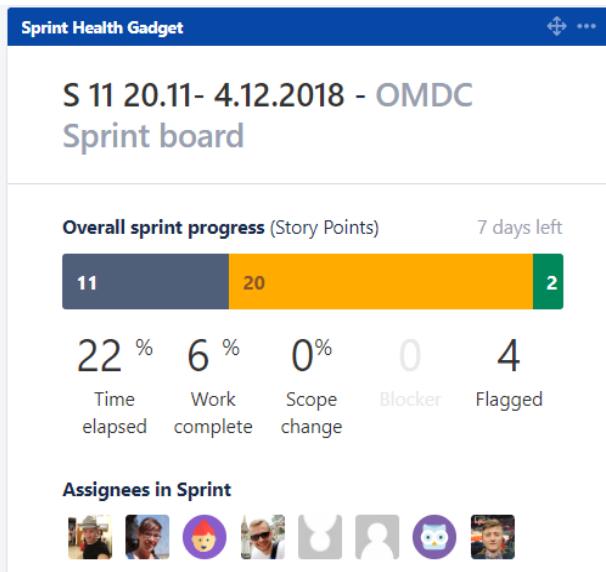
[Agile@Scale KPI Metrics explained](#)

[AGILE KPIs and Dashboards](#)

Teams specific metrics and references

Apart from main KPI metrics categories, Teams may use internal metrics which will help them in continuous improvement process by raising predictability and visibility.

More about those here: [Jira for Scrum Teams Health Check](#)



Chapter 7

Governance

William **HILL**



Introduction: Technology Governance

Our governance processes are meant to provide a structure for aligning technology strategy with business strategy.

As a PLC we also have a responsibility to show appropriate control over capital investment.

By having a formal governance framework in which we operate, we can measure, monitor and report on how we perform.

However to support Agility in our Teams we move to a less sequential controls model which allows freedom within this framework for our Teams. At William Hill we do this through our 'Guardrails' and Investment processes. With that approach business divisions have greater autonomy on the deployment of IT resource and budget. Therefore tech governance needs to ensure that the IT strategy in the divisions is pursued in line with, and cognisant of, the Group Technology Strategy.

**With 'Guardrails' approach,
Agile teams are empowered to
'continue' unless a guardrail is
'triggered' at which point the
engagement of appropriate SME
and Architecture, Security or
Operational process/resources
will be triggered.**

Governance aligned to Agile

We are implementing 'Guardrails' mechanisms to empower teams and embed sound design principles in projects from their inception, reducing the need for formal and sequential controls

From...

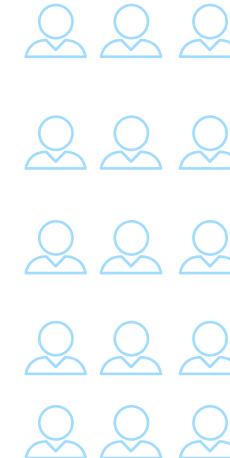


Process illustration



SME involvement

- Idea generation
- Business case
- Case approval
- Detailed design
- Technical Sign off

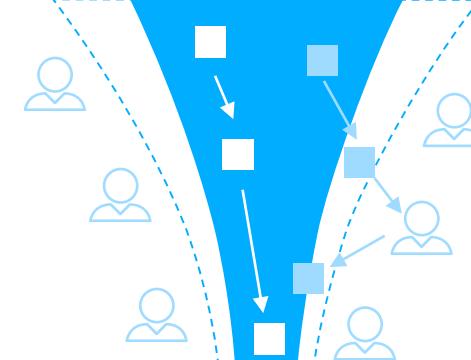


To...



Process illustration

Idea generation



Release

SME involvement

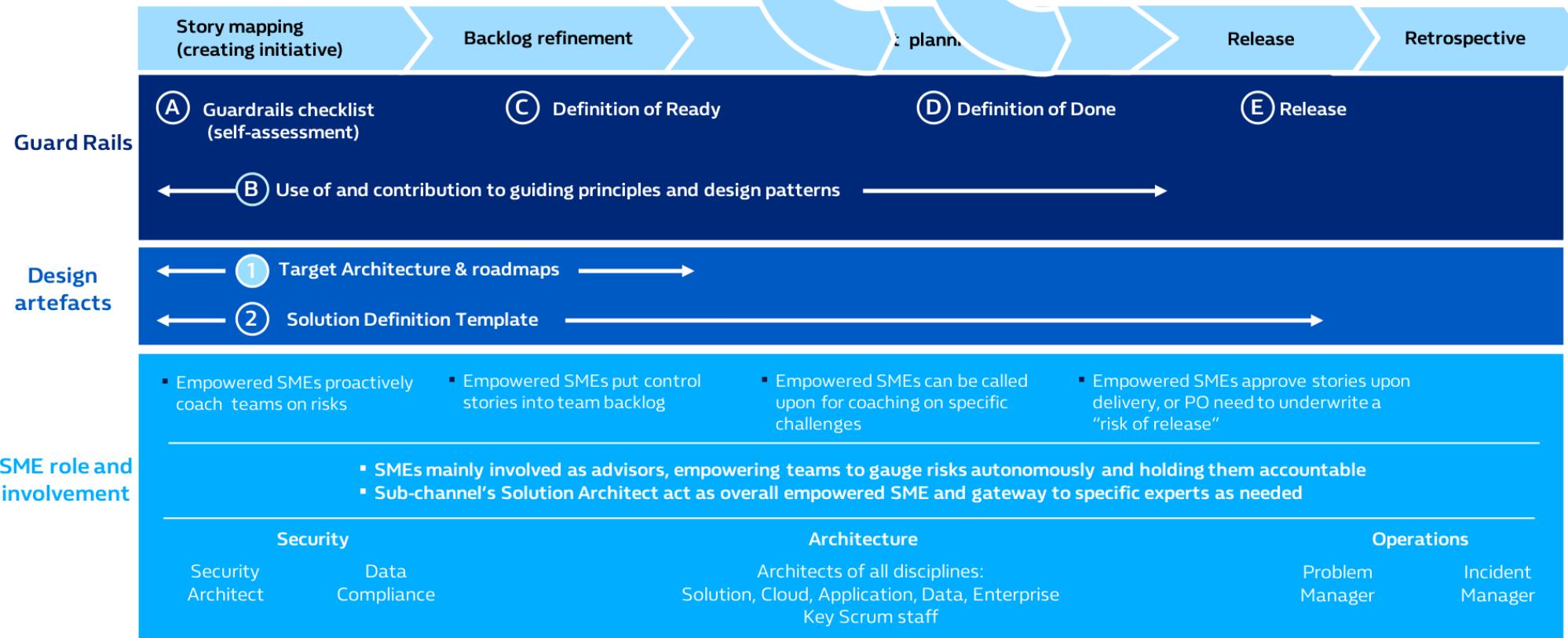
- Guard rails replace gates; allowing teams to move quickly
- SMEs are involved throughout; collaborating early on design challenges

- **Long and costly** process
- **False precision and significant waste** from upfront detailed design and approval, for features that we need to test and adapt
- **Limited adaptability** as design is front-loaded and followed by straight execution

- **Self-assessment** through easily available documentation on guiding principles
- **Early and ongoing involvement of SMEs** (e.g., Architect, Security) to minimize re-work and delays in later phases and collaborate at the right time
- **Flexibility** to deliver low-risk features quickly, and to easily identify when SME insights required

Governance aligned to Agile

The intent of the Guardrails is to reduce friction and increase business 'velocity'.



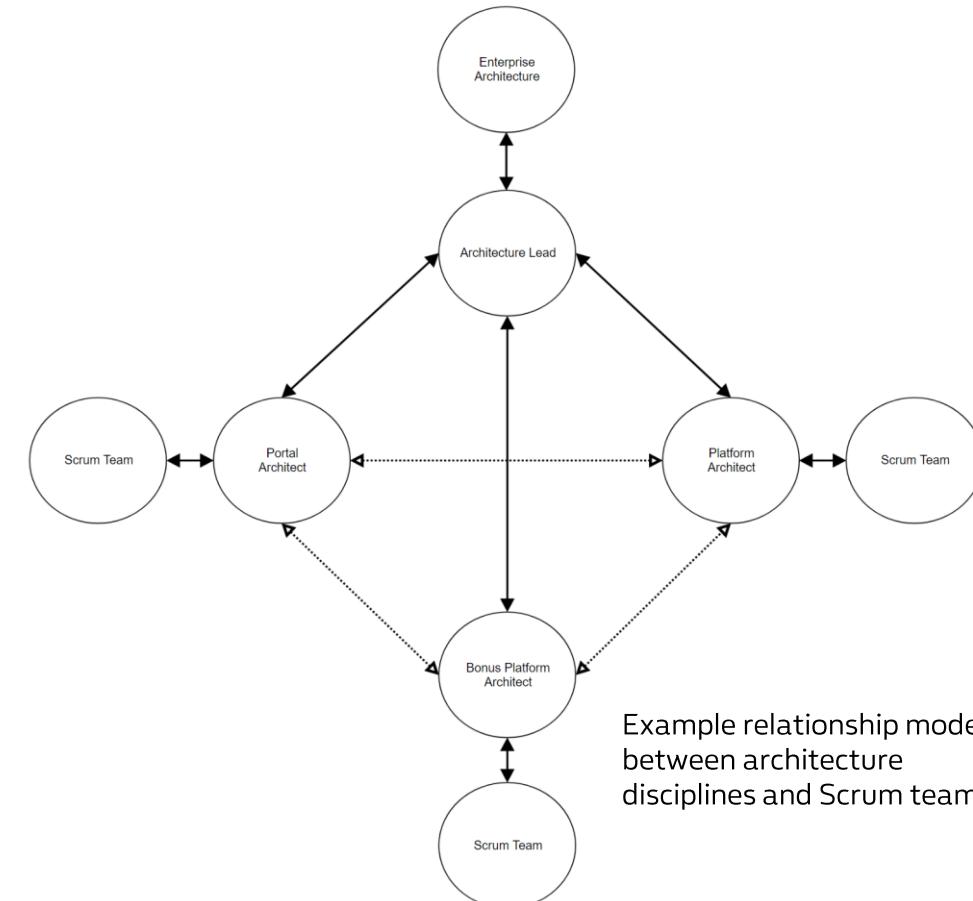
The Guardrails are aligned to the SDLC with a simple set of check list questions.

The Guardrails are aligned to the SDLC definitions of Ready and Done.

The full Tech Guardrails pack is located [here](#).

Guardrails – Who should be using them

- The Guardrails are intended for use by all architecture disciplines and Scrum teams.
- Delivery Channels will typically have an SA / Lead SA; Larger channels will have domain/segment (application/software architects) who are embedded or work closely with Scrum teams.
- The SA will align the channel architecture to enterprise architecture; The domain/segment architect will work with the channel SA to understand the channel architecture and then conceptualising how this would fit into the segment architecture, identify any opportunities for reuse, and produce an initial architecture assessment.
- The Scrum team and the domain/segment architects will collaborate to agree the high level design that meets the target architecture.
- The low level design created by the Scrum team will be reviewed against the emergent architecture through retrospectives, any divergence from the architecture will be discussed with the domain/segment architect and the Lead Architect to reach a compromise.
- All architects and Scrum teams should have a good understanding of and have input to develop the Guardrails.



Example relationship model
between architecture
disciplines and Scrum teams.

Governance-friendly strategies for a delivery team

