New Zealand Government

Assuring Digital Government Outcomes

Assurance Guidance for Agile Delivery

Version 1.1 October 2019

Better information, better conversations, better decisions

Document History

Version	Issue date	Description of changes
Version 1.0	May 2019	Initial version (published)
Version 1.1	October 2019	Minor updates to reflect website updates

"a positive declaration intended to give confidence"

Confidence

Informative

Certainty

"the goal of improving information or the context of information so that decision makers can make more informed, and presumably better, decisions"

Independence

"the comfort that can be derived from credible information"

"an independent and objective oversight of the likely future performance of major investments for those responsible for sanctioning, financing or insuring such undertaking"

AN OBJECTIVE EXAMINATION AND INDEPENDENT ASSESSMENT OF AN INVESTMENT INCLUDING RISKS, CONTROLS, PROCESSES, AND GOVERNANCE.

Assurance is the process of providing confidence to stakeholders that *an investment* will achieve their objectives, and realise their benefits.

Credibility

Table of Contents

1 Introduction 4

1.1 Purpose 4

1.2 Context 4

1.3 Benefits of Agile 4

1.4 When to use Agile 5

2 Agile roles 7

2.1 Scrum teams 7

2.2 Product Owner 7

2.3 Scrum Master 8

2.4 Development team 8

3 What’s different about Agile assurance? 9

3.1 First line of defence 9

3.2 Second line of defence 10

3.3 Third line of defence 12

4 Applying assurance thinking to Agile delivery 13

4.1 Understanding your context 13

4.2 Developing your assurance approach 14

4.3 Areas to probe with Agile assurance 15

5 Engaging with us 20

5.1 Lifting risk management and assurance capability 20

5.2 Guidance and templates 20

5.3 How to contact us 20

Glossary of terms and abbreviations 21

1 Introduction

1.1 Purpose

The primary purpose of this guidance is to support government organisations in applying good practice assurance to Agile delivery, typically as part of a digital transformation programme or project.

1.2 Context

This guidance sits under the All-of-Government Portfolio, Programme and Project Assurance Framework[1] and should be viewed as an extension to that framework, which outlines the principles of good assurance.

We recognise that many government organisations are adopting 'agile' ways of working beyond software development. We believe that the guidance in this document can be applied more broadly to these programmes and projects. For that reason, we have used the term 'Agile delivery' throughout the document to denote both IT and non-IT delivery. We have used the terms 'software' and 'product' interchangeably to denote the delivery of value.

This guidance is intended to be methodology agnostic. However, where necessary, we have used Scrum terminology to illustrate what's different about Agile delivery. Scrum is the most common Agile approach used by government organisations.

1.3 Benefits of Agile

Over the last two decades, linear or sequential approaches to software development (often described broadly as 'Waterfall' delivery) have been supplemented by more iterative and incremental approaches. These approaches are typically described as 'Agile', although this umbrella term covers a family of many diverse software development and delivery methods, including:

- Scrum and Scrum hybrids (Scrumban, Scrum/XP, etc.)
- Kanban
- Lean
- Scaled Agile (SAFe), Large-Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), Spotify Model, etc.

Agile approaches focus on creating and delivering value incrementally. For example, working software is frequently released to customers to realise benefits.

[1] https://www.digital.govt.nz/standards-and-guidance/governance/system-assurance/all-of-government-portfolio-programme-and-project-assurance-framework/

A key component of any Agile approach is an iterative, continuous approach to planning that defines and elaborates requirements just in time. This is a shift from the traditional, fully scoped approach that plans and defines requirements at the very start of the project. Moreover, Agile delivery is based on business priority or value, and no assumption is made at the start that all scope items will be delivered.

Agile teams have fast learning and adapt cycles to continually improve their performance and to closely engage with the actual users and customers of the product so that quality can be built in and tested during development.

The team should be continuously delivering ‘just enough’ to achieve the business need and to be able to respond quickly to change while still having delivered benefits to the customer.

According to the 12th annual State of Agile report (COLLABNET, 2018), the benefits of adopting Agile delivery include[2]:

- Ability to manage changing priorities
- Project visibility
- Business/IT alignment
- Delivery speed/time to market
- Increased team productivity.

1.4 When to use Agile

It’s important to recognise that Agile may not always be the most appropriate approach for delivering change. Agile methods are generally more suited to projects where the requirements and how to fulfil them using current knowledge and technology is unclear. This is illustrated in the diagram below.

[2] https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report

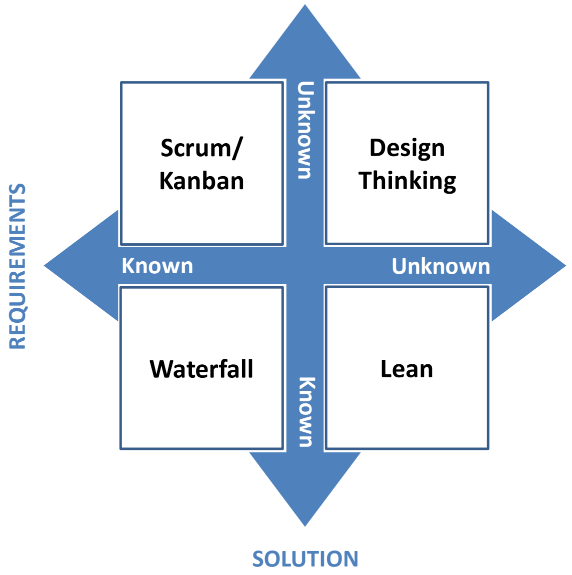


Figure 1: Selecting the right approach

For example, projects that have clear and stable requirements with a clear solution will generally have less uncertainty and may be best delivered using Waterfall, especially if your organisation does not have experience in Agile delivery.

Conversely, if both the requirements and solution are unknown, then Design Thinking can be used to develop a Lean start-up experiment which may help determine whether a Scrum/Kanban, Lean or Waterfall is the most appropriate delivery method.

Other factors that can contribute to project uncertainty include:

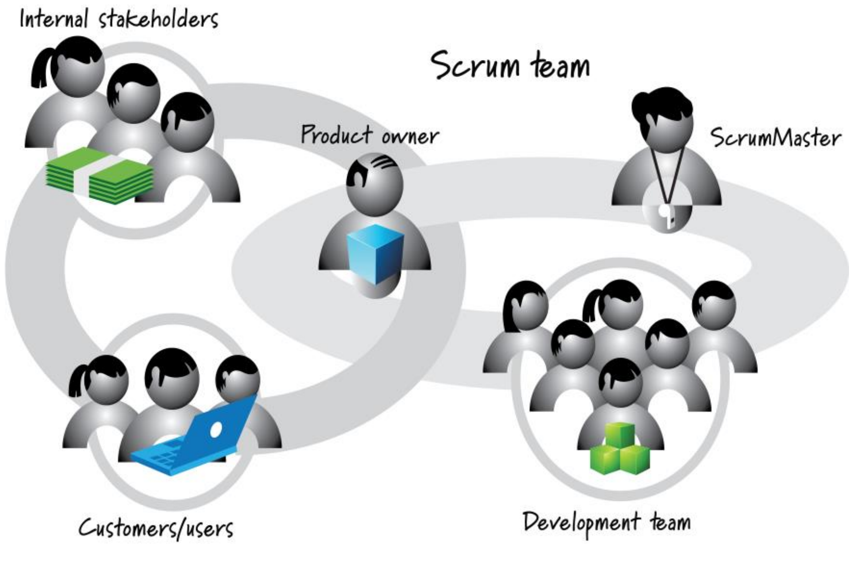
- High numbers of stakeholders with conflicting needs
- High levels of dependencies between parts of the project
- Complicated procurement models, multi-vendor engagements
- Larger, longer programmes or projects.

2 Agile roles

2.1 Scrum teams

Scrum is the most widely-used Agile approach and is suitable for most types of software development and delivery.

The Scrum team and its interaction with key stakeholders and customers/users is illustrated in the diagram below. There are three key roles in the Scrum team: Product Owner, Scrum Master and the Development team. Each role is described in more detail below.



Copyright © 2012, Kenneth S. Rubin and Innolution, LLC. All Rights Reserved.

Figure 2: Agile roles

2.2 Product Owner

The Product Owner is a key role in Agile delivery. They work with key stakeholders to agree and prioritise scope, create a vision and delivery roadmap for the product, and agree the defined scope of releases with the Development team.

The Product Owner role is also critical in managing complexity by making sure any ambiguity in scope or any conflicting stakeholder requirements are resolved in a way that lets the team focus on a small amount of well-defined scope at a time (typically called a 'feature' or 'story').

This role typically places much higher demands on an individual than the equivalent PRINCE2 role of Senior User on a Waterfall project. Teams often need frequent (ideally daily) contact with the Product Owner to clarify scope, make priority calls and review completed work. The Product owner sits at the centre of many Agile frameworks, acting as the interface between users/stakeholders and the Development team.

Appropriately tailoring this role for your project context is important, particularly the decision-making authority of this role compared with a Senior Responsible Owner (SRO) or Programme/Project Manager. A common approach is to use tolerances to delineate responsibility (i.e. the Project Manager may make decisions that impact a sprint or iteration, whereas the Product Owner or Programme Manager may have authority to make similar decisions within the context of an overall Release or Increment. Using this approach, any decisions that impact overall benefits need to be escalated to the SRO.

2.3 Scrum Master

The role of the Scrum Master is as a 'servant leader' and guide, helping the Development team, Product Owner and the wider organisation understand and implement Agile practices.

The responsibility of a Scrum Master is quite different to that of a Project Manager (but may sometimes be regarded as the 'Project/Team Lead'). The Scrum Master does not have responsibility for delivery and does not allocate work packages or tasks to Development team members (instead the Sprint or Iteration Planning process governs the agreement between Product Owner and the Development team for each sprint or iteration).

In the context of a plan-driven programme or project, the Scrum Master may help the Programme or Project Manager to integrate typical Agile processes such as iterative and incremental planning and delivery of scope using a programme or project level schedule and budget.

2.4 Development team

The Development team in Scrum ('Dev Team' in SAFe) is a generic name for the people working at the delivery level (e.g. analyst, designer, developer, tester), who often have cross-functional roles. The role description is deliberately abstract to reinforce the concept of group accountability and to encourage multi-skilled or 't-shaped' people.

The Development team is self-organising and any interaction outside the team is managed by the Product Owner who has regular interactions with all stakeholders and is typically regarded as voice of the customer.

In contrast with traditional approaches, the agreement to deliver a work package or task is directly between the Development team and the Product Owner (work is not allocated; rather it is taken on at regular intervals by the team, based on their assessment of their current capacity and capability).

Team size is generally kept at below 10 people and physical colocation is encouraged to maximise effective communication. Ideally, the Product Owner and user representatives are also colocated with the team

3 What's different about Agile assurance?

The outcomes for Agile assurance are no different to those for traditional assurance approaches, i.e. providing stakeholders with confidence that the expected investment outcomes and benefits will be achieved.

However, Agile is more self-assuring than Waterfall. This means that assurance is part of the delivery process, with risk management embedded into day-to-day operations and governance arrangements.

This section defines some of the key differences between Waterfall and Agile delivery using the 'three lines of defence' model. This model is used to clearly define roles and responsibilities for effective risk management across the organisation. In the world of delivering programmes and projects, we use the following descriptions for each line of defence:

- The first line of defence is the day-to-day project management processes and controls you have in place, including quality management
- The second line of defence is the governance and oversight arrangements that exist, including clear and signed off terms of reference for all governance bodies
- The third line of defence is the independent assurance you obtain from internal (e.g. Internal Audit) and third-party assurance providers.

It's important you give appropriate thought to tailoring your assurance approach to take into account the following key differences between Waterfall and Agile delivery across the three lines of defence.

3.1 First line of defence

- Planning documents focus on specifying outcomes and desired capabilities rather than specifying deliverables in detail. This focus enables the team to be agile.
- Day-to-day management of workflow shifts from being Project Manager led to being delegated to self-organising teams. Teams define and self-allocate the tasks required to meet agreed goals. This is usually done via a daily stand-up.
- Planning (e.g. 'Sprint Planning' or 'Programme Increment Planning') is conducted more frequently, occurring in waves as the project proceeds and details become clearer ('rolling wave' planning).
- The Definition of Done (DoD) is the key measure of quality, and the focus shifts to building in quality and testing during delivery rather than testing for quality at the end. A clearly defined DoD creates an environment of 'assurance by design' that is supported by:
  - regularly scheduled ceremonies throughout the programme or project
  - appropriate technical practices which are embedded into delivery (e.g. testing, continuous integration, and automated release).

- Teams have mechanisms to regularly review and improve how they are working together. For example, the ‘Sprint Review’ and ‘Sprint Retrospective’ in Scrum and ‘Inspect and Adapt’ process in SAFe.

- As Agile environments are more collaborative, relationships with vendors and the colocation of teams and business representatives (or lack of them) become more important indicators of likely outcomes.

- The key risk mitigation in Agile delivery is frequent releases of software to production. Failure to release frequently exposes the programme or project to high levels of risk. This is because Agile delivery typically doesn't have a planned 'test and fix’ period at the end of development, rather it is handled within the sprint before the ‘feature’ or ‘story’ is completed. A programme or project that isn’t releasing software frequently should act as a 'red flag' for governance bodies and independent reviewers.

- The Product Owner role is critical to being able to quickly clarify scope and set priorities for teams, and to make sure the outcomes of the project can be delivered. This means the Product Owner should be delegated the appropriate authority to make decisions about time, scope and cost, and have a strong relationship with the SRO as their day-to-day representative on the programme or project.

3.2 Second line of defence

- Clear and signed off terms of reference for governance bodies are tailored to allow appropriate delegation of decision-making to the Product Owner (or equivalent role).

- Governance bodies are prepared to meet more frequently, as required, particularly where limited authority is delegated to the Product Owner. (Agile delivery requires faster and more frequent decision-making).

- Governance bodies need to understand a new set of performance metrics and should consider appropriate executive coaching if Agile delivery is new to the organisation. When working well, Agile provides increased transparency with the availability of real-time measures of progress:

    - The primary measure of progress is seeing working product delivered into production (rather than milestones based on achieving phases of the delivery lifecycle).

    - Burn-up or burn-down charts at the project or release level show the likely project outcomes (rather than qualitative 'percentage-complete' or 'red-amber-green' reporting). In the chart below, the forecast is based on actual performance to date and indicates that the project is unlikely to deliver within the agreed tolerance range.

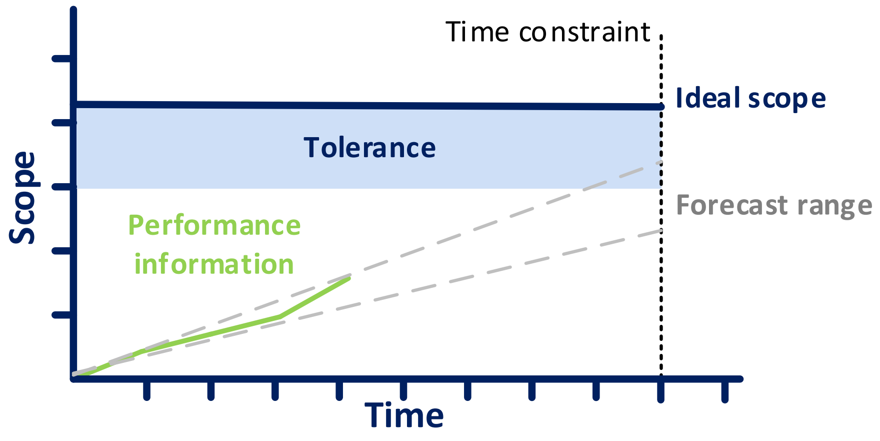


Figure 3: A burn-up chart showing actual progress over time

- A shift in focus from confirming to the plan, to one of appropriately responding to near-real time quantitative performance information (i.e. choosing to de-scope, adding more time or budget, or changing quality levels based on release level forecasting). This includes deciding to close a programme or project if enough benefits have been delivered (e.g. avoid gold-plating or continuing to invest if the business need has been met) or if the business justification has shifted.

- Prioritisation of scope acts as a key indicator of effective risk mitigation and governance engagement. For example, using MoSCoW (Must have, Should have, Could have, Won't have) to sequence the delivery of features. This might include an economic view such as Weighted Shortest Job First[3] or Cost of Delay Divided by Duration[4].

- Making sure Agile delivery teams are fully engaged with the users of the product they are developing and that feedback from users is appropriately actioned.

- Agile teams proactively engage with oversight functions such as Architecture Review Boards, Enterprise Portfolio/Programme Management Offices (EPMOs) and Security and Risk teams early in delivery and on an ongoing basis. They also make sure key non-functional assurance activities (such as performance testing, security certification and accreditation) are iterated throughout as the product is delivered.

- Release management and IT environment controls need to be appropriately tailored for Agile delivery (e.g. the Development team should have control of development and test environments, and processes governing releases to production should be appropriately streamlined – ideally automated – to enable fast and regular software delivery).

[3] https://www.scaledagileframework.com/wsjf/

[4] http://blackswanfarming.com/cost-of-delay-divided-by-duration/

3.3 Third line of defence

- A shift in the timing and length of assurance reviews, to shorter, more frequent reviews.
- The use of appropriately skilled independent reviewers with a background and experience in practical Agile delivery.
- A change in the format and formality of key programme or project artefacts used to evidence review findings:
    - Planning information is at a high level in formal documents with detailed scope information contained in a delivery management tool (e.g. JIRA or Visual Studio Team Services/Azure DevOps). Detailed progress information may also be held in a delivery management tool.
    - Typical 'management plans' may be written up informally in Wikis or other online sources.
    - Designs may evolve and be captured in a series of whiteboard photographs as a living document.
    - Quality control and review activities may simply be part of day-to-day delivery processes with a DoD covering quality standards and quality review.
    - Photographs of Kanban boards and other highly visible artefacts in team areas are key reference sources for independent reviewers.
- A shift to focus on behaviours and practices, including observing the four key Agile 'ceremonies' (planning meetings, stand-ups, reviews (or demonstrations), and retrospectives).
- A higher level of engagement between independent reviewers and the delivery team(s) and less engagement and/or one-on-one interviews with senior stakeholders.
- Advice to SROs on the suitability of Agile practices and how they have been tailored to the specific programme or project context.
- Review outputs will be less formal and may be provided as 'one pagers' or PowerPoint decks rather than formal reports. In all cases guidance contained in the Government Chief Digital Officer (GCDO) Assurance Services Panel pocket guide[5] should be followed (e.g. providing a clear Executive Summary, an assessment of delivery confidence, major findings, key decisions required, and a management comment from the SRO).
- The outputs from assurance reviews should be made available to the delivery team(s) and any recommendations should ideally be able to be actioned quickly (i.e. in the next sprint or iteration).

[5] https://www.digital.govt.nz/dmsdocument/90-gcdo-assurance-services-panel-pocket-guide

4 Applying assurance thinking to Agile delivery

4.1 Understanding your context

New Zealand’s government investment system and associated funding model is based on the Better Business Cases (BBC) process. This defines a standard business case lifecycle that all significant investments need to align to. This is illustrated in the diagram below which shows how Agile approaches may operate in the investment system[6].

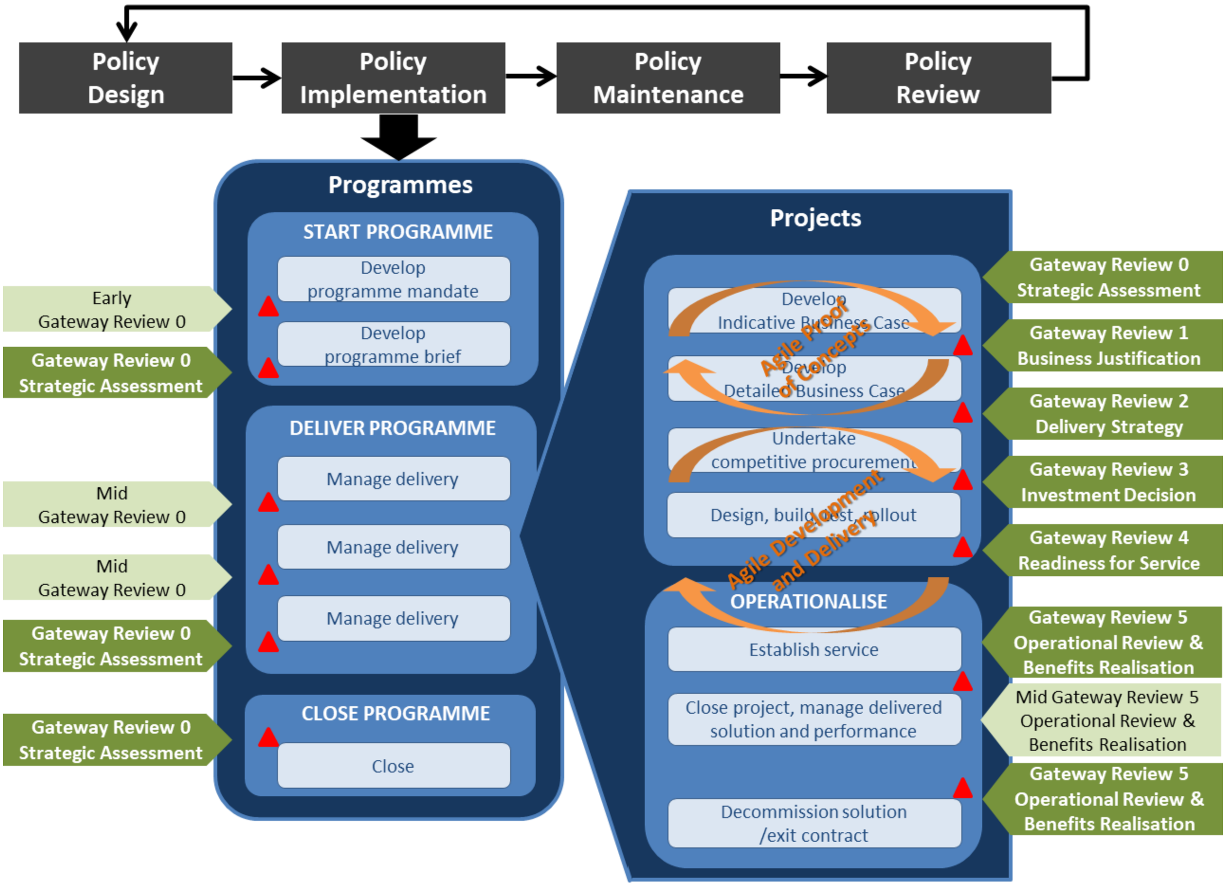

Policy Design
Policy Implementation
Policy Maintenance
Policy Review
Programmes
START PROGRAMME
Develop programme mandate
Develop programme brief
DELIVER PROGRAMME
Manage delivery
Manage delivery
Manage delivery
CLOSE PROGRAMME
Close
Early Gateway Review 0
Gateway Review 0 Strategic Assessment
Mid Gateway Review 0
Mid Gateway Review 0
Gateway Review 0 Strategic Assessment
Gateway Review 0 Strategic Assessment
Projects
Develop Indicative Business Case
Develop Detailed Business Case
Agile Proof of Concepts
Undertake competitive procurement
Design, build, test, rollout
Agile Development and Delivery
OPERATIONALISE
Establish service
Close project, manage delivered solution and performance
Decommission solution /exit contract
Gateway Review 0 Strategic Assessment
Gateway Review 1 Business Justification
Gateway Review 2 Delivery Strategy
Gateway Review 3 Investment Decision
Gateway Review 4 Readiness for Service
Gateway Review 5 Operational Review & Benefits Realisation
Mid Gateway Review 5 Operational Review & Benefits Realisation
Gateway Review 5 Operational Review & Benefits Realisation

Figure 4: Agile delivery within the context of the government investment system

This means that projects that use Agile delivery approaches will need to consider how they align assurance activities to the key decision points that support the BBC process. Key decision points are represented by the red triangles in the image above. For high risk investments, projects will be subject to monitoring and independent assurance oversight by central agencies and functional leads, including Gateway reviews.

[6] https://treasury.govt.nz/information-and-services/state-sector-leadership/investment-management/review-investment-reviews/gateway-reviews

Government organisations may also use Agile approaches to de-risk their project initiation – e.g. using proof of concepts to test out assumptions during Detailed Business Case development.

In addition, organisations may use Agile delivery approaches for any non-IT scope items that use more plan-driven, traditional approaches, such as policy development or organisational change initiatives.

When using multiple delivery methodologies, organisations should set clear criteria and provide guidance around when and how each approach will be used and how to effectively tailor approaches depending on the nature of the change. Specialist advice or coaching is recommended for organisations using Agile delivery approaches.

4.2 Developing your assurance approach

When starting to develop your assurance approach, it is important to remember that the principles of good assurance still apply.

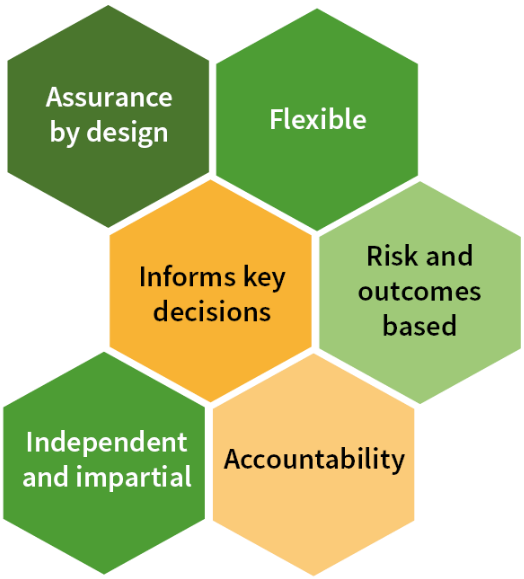


Figure 5: Principles of good assurance

The following key considerations, based on the principles of good assurance, will help inform early assurance thinking in Agile delivery:

- Plan early for Agile assurance and not just during delivery – e.g. advice to SROs about whether an Agile approach is suitable for their programme or project, and how well it has been tailored to the specific programme or project context.
- Tailor your assurance approach based on a clear understanding of the Agile delivery approach and the wider programme or project context in which the investment is funded and managed:

  - Consider the extent to which the DoD includes assurance related activities e.g. security and privacy assessments, control assessments, and other risk impact assessments that will deliver quality outputs, including confidentiality, integrity and availability of systems, processes and data.

  - Consider the extent to which shorter, more frequent assurance reviews are needed in order to provide ongoing assurance that governance, risk management and Agile delivery are operating effectively e.g. these reviews may be aligned to iterations/increment planning cycles.

  - Consider the extent to which assurance reviews need to support key decision points and/or key programme or project management activities, including business case reviews, Technical Quality Assurance (TQA), programme or project Independent Quality Assurance (IQA), Gateway, etc.

- Develop a 'tool box' of scope items that can be tailored to specific areas of Agile delivery to provide valuable insights that can be actioned quickly (see Areas to probe in next section).

- Build in assurance activities that observe ceremonies, including Sprint planning, daily stand-ups, Sprint reviews and retrospectives, backlog grooming, stakeholder engagement and management of impediments or other issues. The assurance plan should include a list of these activities, their purpose and frequency. Risk and Internal Audit can play a key role in observing the effectiveness of these ceremonies

- The assurance plan will need to be more actively managed in order to be responsive to iterative changes. It should give a detailed three-month view of assurance activity and only a high level view of 'indicative' activity after that. Make sure there are mechanisms built in to regularly review the assurance plan.

- Use an integrated assurance plan to ensure that all stakeholder needs are met and that assurance activities are coordinated.

- Ensure third-party assurance providers have practical Agile delivery experience.

4.3 Areas to probe with Agile assurance

Examples of areas to probe that are specific to Agile delivery are listed below. These should be integrated with any other planned assurance activities.

4.3.1 During programme or project set-up

Area to probe	Key questions
Selection of approach	• Is there a clear rationale for choosing Agile as the most appropriate delivery method for this programme or project? • Is Agile a good fit for the organisation's culture and management practices? • Is there an awareness of the likely impacts on resourcing, particularly if key people are shared across Agile projects and business-as-usual work? • Are there appropriate reporting systems in place to show progress clearly in a language the organisation understands?
Vision and strategy	• Does the programme or project have a vision statement? • Is the related strategy compelling and does the problem/value statement have enough detail for Agile teams to begin planning? • Is there a Product Roadmap which sets out target delivery dates for capabilities or features?
Benefits realisation and risk mitigation	• Has the programme or project been split into smaller chunks to deliver benefits early and to mitigate key risks? • Is the method through which features are prioritised for delivery based on expected value? • Are the highest risks mitigated or de-risked through proof of concepts or spikes (time-limited experiments to test out technology choices or otherwise learn about an aspect of the system being built)? • Is there alignment between the Product Roadmap, planned tranches (for programmes) and project plans? • Does the benefits realisation approach map to the delivery of features, as per the Product Roadmap?
Resource capability	• Do key programme or project roles have experience in practical Agile delivery? • Are the right resources with the required capability in using Agile delivery approaches available when needed? • Are Agile practices being used appropriately and is there evidence that the approach has been tailored to the programme or project's organisational context? • Are Agile coaches and other expert resources being used, where appropriate?
Governance	• Is governance appropriately calibrated to provide enough direction, while letting teams 'get on and deliver'? • Is the Product Owner(s) in place with the appropriate delegation from the SRO? • Have appropriate tolerances been assigned? • Is there agreement of what an *acceptable* on-time delivery looks like (a Minimal Viable Product or MVP)? • Does governance reporting include up-to-date performance information?

4.3.2 During programme or project delivery

Area to probe	Key questions
Benefits management	• Is the programme or project delivering working software regularly and is it enabling early realisation of benefits? • Are there processes are in place to capture benefits as they are delivered by the programme or project?
Scope management	• Is the programme or project using a system to manage scope items at varying levels of granularity (for example, a 'Backlog') • Is there evidence that the Backlog is updated and prioritised by key stakeholders on an ongoing basis?
Schedule management	• Does the programme or project have a Release Plan that shows how the capabilities or features in the Product Roadmap will be delivered? • Is the programme or project using actual performance information to forecast if Release Plans continue to be realistic and achievable (for example, Story Points are delivered per iteration or features are delivered over an agreed period)?
Budget and Cost management	• Is the funding model flexible and robust (e.g. allows lean budgeting, incremental draw down of funding, and an incremental method of measuring return on investment)? • Is there alignment between the budget planned for the Product Roadmap and the forecast costs of the Release Plan(s) • Is budget spend tracked and correlated with scope delivery (i.e. based on forecasting what is likely to be delivered within the current budget)?
Quality management	• Do all deliverables have an effective DoD that is well understood at all levels and sets a minimum-quality requirement? • Where the programme or project is using a layered approach to DoD (i.e. there's a lower threshold for a story or sprint than for a feature or increment), is there full alignment between 'done' for a feature or increment, and 'ready for production release'? • Are all deliverables (or components of deliverables) not reported as 'done' unless the DoD has been met? • Is the DoD reviewed and/or updated as delivery progresses?
Risk management	• Do all stakeholders agree on what the MVP looks like? • Is risk being managed through delivery of the MVP first, followed by enhancements? • Is risk management based on nimble or agile practices to fit the uncertainty and risk level of the project? • Are technical risks managed through the use of proof of concepts or 'spikes' (time-limited experiments to test out technology choices or otherwise learn about an aspect of the system being built)? • Are critical non-functional requirements progressively delivered over time (e.g. 'shift left' performance testing, security certification and accreditation processes)?

Area to probe	Key questions
	• Are processes in place to seek feedback early and often? For example, is Backlog prioritisation clearly influenced by user feedback and progress to date? • Are products released into production on demand or as per targeted release cycle? • If less frequent releases are planned, is there awareness that a longer release frequency substantially increases programme or project risk and that this risk has been accepted by the SRO?
Governance	• Are governance meetings aligned to programme or project delivery cycles? • Do governance group members understand the workings of Agile delivery and have they appropriately delegated decision-making downwards? • Is there close engagement between the SRO, Product Owner and the delivery team(s), evidenced by attendance at planning and review sessions? • Does governance have good visibility into the workings of the programme or project? And is actual performance information shared widely and displayed prominently (i.e. burn-up or burn-down charts are used in reporting and visible in the team area)? • Is working software/product used as the primary metric of progress?
Stakeholder management	• Are key stakeholders fully engaged in the definition of scope, prioritisation of features and review of outputs? • Do key stakeholders attend Agile events (e.g. planning and review meetings)? • Is there evidence that feedback from key stakeholders is being actioned (e.g. changes to Backlog content and/or prioritisation)?

4.3.3 Prior to release and handover to operations

Area to probe	Key questions
Release (expected multiple times)	• For each production release during programme or project delivery, has the DoD been met for: ○ Compliance work (e.g. certification and accreditation) ○ Business readiness, including user guides and training of end users ○ Operational support readiness (e.g. production support is in place and has appropriate documentation)? • Where the programme or project will be both supporting production and working on future release(s), is the resource plan and schedule for future releases realistic? • For the final production release, is the Backlog appropriately refined and ready to hand over to business-as-usual?

Annex A of the UK government's *'Assurance and approvals for agile delivery of digital services'*[7] also provides useful information for independent reviewers.

[7] https://www.gov.uk/government/publications/assurance-for-agile-delivery-of-digital-services

5 Engaging with us

The System Assurance team has an independent assurance oversight role over high risk digital investments. This includes programmes and projects using Agile delivery approaches.

The AoG Portfolio Programme and Project Assurance Framework provides guidance on how and when to engage with the System Assurance team.

https://www.digital.govt.nz/standards-and-guidance/governance/system-assurance/all-of-government-portfolio-programme-and-project-assurance-framework/.

5.1 Lifting risk management and assurance capability

The System Assurance team works collaboratively with government organisations to lift risk management and assurance capability. Further information on the role of the System Assurance team can be found on the GCDO's website: https://www.digital.govt.nz/standards-and-guidance/governance/system-assurance/role-of-the-system-assurance-team/

5.2 Guidance and templates

Further guidance and templates can be found on the GCDO's website: https://www.digital.govt.nz/standards-and-guidance/governance/system-assurance/all-of-government-portfolio-programme-and-project-assurance-framework/guidance-and-templates/.

5.3 How to contact us

Contact the System Assurance team for digital investment assurance advice.

Email: systemassurance@dia.govt.nz

Glossary of terms and abbreviations

Term	Definition
CI or Continuous Integration	A code-management process that integrates the code of multiple developers at a regular frequency, ideally at least daily. This practice relies on test automation, starting with unit tests written by developers as part of their normal coding practice.
DAD	Disciplined Agile Delivery. A form of scaling Agile to multi-team or whole of organisation. https://www.disciplinedagiledelivery.com/
DoD or Definition of Done	How quality is managed in Agile delivery. The Definition of Done defines acceptable quality (levels and practices). It can be applied to a work item (epic, feature or story) or to a process step (i.e. analysis, development or testing).
Epic	A very large work item. May be made up of multiple features and/or stories.
Feature	A mid-sized work item (more than one story, but smaller than an epic). May or may not be used by an Agile team.
Kanban	An iterationless approach with a focus on lowering Work In Progress (WIP) and improving flow. Makes use of tools such as task boards, work in progress limits, and continual process improvement through use of metrics. Rather than fixed planning at agreed intervals (sprint or iterations), Kanban teams replenish a short queue of work as required to ensure there is always at least one prioritised item in the queue.
Lean Start-up	A methodology which aims to shorten product development cycles and rapidly discover if a proposed business model is viable; this is achieved by adopting a combination of business-hypothesis-driven experimentation, iterative product releases, and validated learning.
LeSS	‘Large-scale Scrum’. A form of scaling Agile to multi-team or whole of organisation. https://less.works/
MSP	Management of Successful Programmes - the standard programme management framework within New Zealand government organisations. https://www.axelos.com/best-practice-solutions/msp
PMBoK	Project Management Body of Knowledge. A project management framework created by the Project Management Institute. https://www.pmi.org
PRINCE2	Projects In Controlled Environments - the de-facto standard project management framework within New Zealand government organisations. https://www.axelos.com/best-practice-solutions/prince2
Product Roadmap	A simple plan showing the capabilities or new features desired at specific time periods (i.e. ‘In Q3 of 2019 we need the ability to support login via Facebook’).
Release Plan	A plan that shows how one or more software releases are mapped to a Product Roadmap (upwards) and to the iterative delivery of working software through sprints or a continuous flow process (downwards). The Release Plan is the mid-term planning layer that sits between longer-term product planning (the Product Roadmap) and the short-term sprint/iteration planning or replenishment activities of a Scrum or Kanban team respectively.

Term	Definition
SAFe	The ‘Scaled Agile Framework’. A form of scaling Agile to multi-team or whole of organisation. https://www.scaledagileframework.com/
Scaling	Techniques to use Agile delivery for larger projects or programmes where multiple software delivery teams are required.
Scrum	The most common form of Agile delivery approach. Makes use of time-boxed iterations and regular product and process reviews.
Scrumban	A hybrid of Scrum and Kanban. Uses fixed iterations, roles and events from Scrum in combination with the focus on lowering work in progress and improving flow from Kanban.
Specification by example	A form of requirements elaboration that uses examples and creates automatable tests as part of the requirements specification process (‘requirements as test cases’).
Spike	A time-bound attempt to resolve a technical issue or gain more information about the size/difficulty of solving it. Ideally through doing rather than white-paper research.
Spotify Model	Agile in the style of the Swedish company Spotify using structural forms such as Guilds, Chapters, and Tribes.
Story	The standard name for an Agile work item. A story must be able to be completed within a sprint or iteration.
Story Points	An abstract, relative measure of size used by Agile teams to estimate work quickly to 'good enough' accuracy.
TDD or Test Driven Development	A development approach with three basic steps: write tests to prove the desired functionality works, then write simple code to create the desired functionality, then tidy up (‘refactor’) the code to make it pretty. This practice improves design and creates the first layer of test automation in a system (unit tests).
T-shaped Person	A person with depth in one skill (the upright of the T) and breadth across a range of skills (the crossbar of the T). For example, an analyst who can write testable requirements as code or perform as a tester following test cases.
XP	Extreme Programming - an early form of Agile delivery approach. Many of the technical aspects of Extreme Programming are used by other Agile frameworks (e.g. pair programming, test-driven development, refactoring, etc.).