



The Agile Service Management[®] Guide

By Jayne Groll

Sources and Acknowledgements

- ▶ *The Scrum Guide* by Ken Schwaber and Jeff Sutherland, July, 2013
- ▶ *The ITSM Process Design Guide* by Donna Knapp, ISBN: 978-1-60427-049-5 August, 2010
- ▶ *INVEST in Good Stories, and SMART Tasks* by Bill Wake, August, 2003
- ▶ Scrum.org
- ▶ ITIL® is a registered trade mark of AXELOS Limited. Figures with Based on AXELOS ITIL® material. Reproduced under license from AXELOS. are from ITIL® Service Lifecycle Publication Suite Books © Crown copyright 2011. Reproduced under license from AXELOS Limited.



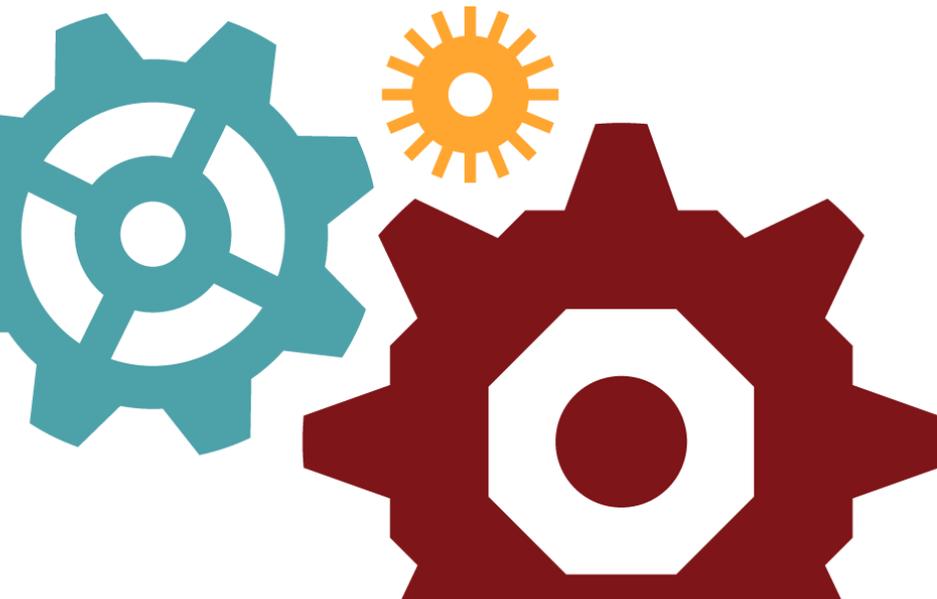
About the Author

Jayne Groll is an ITIL Expert, Certified ScrumMaster, Certified Agile Service Manager (CASM) and Certified Process Engineer (CPDE). She has over 25 years of IT management experience that spans multiple industries including legal, telecommunications, retail, non-profit, insurance and hospitality.

Jayne is CEO and co-founder of the DevOps Institute whose mission is to bring enterprise level DevOps training and certification to the IT market. Jayne is also President and co-founder of ITSM Academy, an ITIL and ITSM training organization. She is active in both the DevOps and ITSM communities and is a frequent webinar and conference speaker.

The inspiration for Agile Service Management® grew out of Jayne's recognition that end-to-end IT agility could only be achieved if Agile thinking and practices were exercised by both development and operational teams.

Agile Software Development + Agile Service Management® = DevOps



The Agile Service Management Guide®

Table of Contents

- Introduction 5
- Being Agile 6
- The Agile Manifesto 7
- What is Agile Service Management? 10
- Agile Frameworks and Methods 12
- Agile Process Design 17
- Scrum Basics 20
- Agile Service Management Roles 24
- Agile Service Management Artifacts 28
- Agile Service Management Events 32
- Agile Process Improvement 40
- Tools for Agile Service Management 43
- Getting Started 45
- Glossary of Terms 46



1

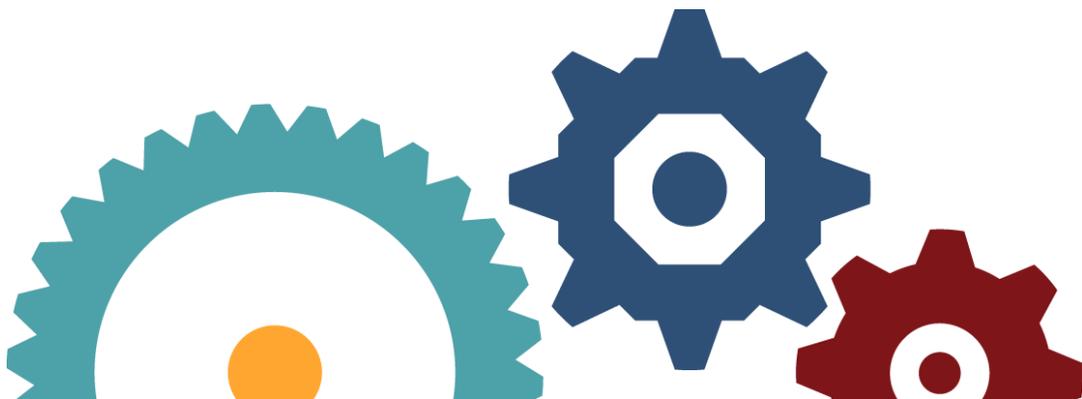
Introduction

Demands on IT for innovation and reliability have been steadily increasing since technology became a critical success factor for most businesses. IT has always been asked to do more with less, to improve its integration with business goals and to ensure the ongoing quality of IT services. With the rise of mobile technology, the cloud and an "app" mentality, IT is being asked to do all that and more at breakneck speed.

While devices and applications are being introduced faster than ever before, it is the service behind the technology that is still most important to the customer. As a result, IT will always need to manage its services and IT service management (ITSM) practices and processes will always be necessary. The challenge is adapting service management practices to changing times so they can enable IT to go faster and deliver more ongoing value to the customer.

Rapidly changing IT requirements require rapidly changing IT capabilities.

New capabilities require new ways of thinking and performing. IT must learn to be more agile.



2

Being Agile

The MacMillan Dictionary defines “agile” as

*Able to move quickly and easily;
able to think quickly, solve problems, and have new ideas.*

Too often in IT, the concept of “being agile” is equated to “doing Scrum.” While Scrum is an excellent framework for managing complex projects, the application of Scrum practices does not necessarily increase an organization’s agility. Software developers recognized this many years ago when they crafted the Agile Manifesto’s guiding values and principles. The tenets of agility must first be understood before embarking on agile practices such as Scrum and other frameworks.

Being agile is a state of mind. It is more perspective than prescription. In order for an organization to “be agile,” they must also be:

- ▶ Customer-centric
- ▶ Lean
- ▶ Collaborative
- ▶ Communicative
- ▶ Adaptive
- ▶ Measurable
- ▶ Consistent
- ▶ Results-oriented
- ▶ Reflective



3

The Agile Manifesto

In 2001, a group of seventeen developers met at a ski lodge in Utah to discuss the increasing complexities associated with modern day software development. The developers were frustrated by delays, rework and customer dissatisfaction that were resulting from constraints and were affecting their ability to get projects done on time and on budget. Their goal in crafting the Agile Manifesto was to refocus stakeholders and developers on the aspects of software development that matter most.

The Agile Manifesto

We Value

Individuals and Interactions
Working Software
Customer Collaboration
Responding to Change

Over

Processes and Tools
Comprehensive Documentation
Contract Negotiations
Following a Plan

*While we value the items on the right,
we value the items on the left more.*

The Agile Manifesto is underpinned by twelve principles of Agile software

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of minimizing the amount of work done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



These principles are not significantly different than the principles that underpin service management

- ▶ Be aligned with the business
- ▶ Focus on customer outcomes
- ▶ Ensure ongoing customer value
- ▶ Understand and enable business success
- ▶ Deliver quality IT services
- ▶ Restore service as quickly as possible
- ▶ Adapt to changing requirements
- ▶ Minimize risks
- ▶ Be effective and efficient
- ▶ Make processes sustainable and repeatable
- ▶ Fulfill IT governance requirements

There is clearly alignment between the objectives of the Agile Manifesto and the objectives of service management. Unfortunately, that alignment does not necessarily translate into end-to-end agility. IT must now learn to be agile throughout the entire service lifecycle - from concept to retirement.

Agile Software Development + Agile Service Management = Agile IT (DevOps)



4

What is Agile Service Management®

Agile Service Management (Agile SM) ensures that ITSM processes reflect Agile values and are designed with “just enough” control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed.

The goals and objectives of Agile Service Management include

- ▶ Ensuring that agile values and principles are embedded into every service management process from design through implementation and continual improvement
- ▶ Improving IT’s entire ability to meet customer requirements faster
- ▶ Being effective and efficient (lean)
- ▶ Designing processes with “just enough” scalable control and structure
- ▶ Provide services that deliver ongoing customer value

Agile Service Management encourages a continuous learning environment and promotes better collaboration between development and operational teams by cross-pollinating vocabulary and methods.

There are two aspects of Agile Service Management: Agile Process Design and Agile Process Improvement.

Agile Process Design

Agile Process Design applies the same approach to process design that software developers apply to product development. Each process is built and potentially released in small, frequent increments. New procedures and behaviors are introduced gradually, providing greater opportunity for normalization as well as more frequent feedback and input to guide the future direction of the process.

An iterative and incremental approach to process design also allows ITSM processes to mature organically and holistically. Dependent increments can be built simultaneously or in succession. Most importantly, the organization can test the boundaries of “just enough” process throughout the service lifecycle.

Agile Process Design does not attempt to redefine the underlying principles of process design. There are solid, proven best practice approaches for process design, including those described in *The ITSM Process Design Guide* by Donna Knapp. Agile Service Management supplements those principles with agile thinking and practices.

Agile Process Improvement

Agile Process Improvement seeks to continually align service management with Agile values and principles as part of Continual Service Improvement (CSI). Processes are regularly audited and reviewed to ensure that they are at the right level of control and do not drift from “just enough” to “too much” or “not enough”. Most importantly, Agile Process Improvement identifies and eliminates bottlenecks or waste in order to keep ITSM relevant, efficient and effective in the face of changing customer requirements.

Agile Service Management is framework agnostic and does not attempt to redefine any of the ITSM processes. ITIL® and other service management frameworks have done an excellent job of describing best practices for managing IT services, including the processes that are necessary for a complete service lifecycle. Agile Service Management supplements those frameworks with agile thinking and practices.





5

Agile Frameworks and Methods

While Scrum is most commonly associated with Agile, there are several frameworks that are aligned with Agile values and Agile Service Management.

Scrum

Scrum.org defines Scrum as a

“simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create “just enough” structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge.”

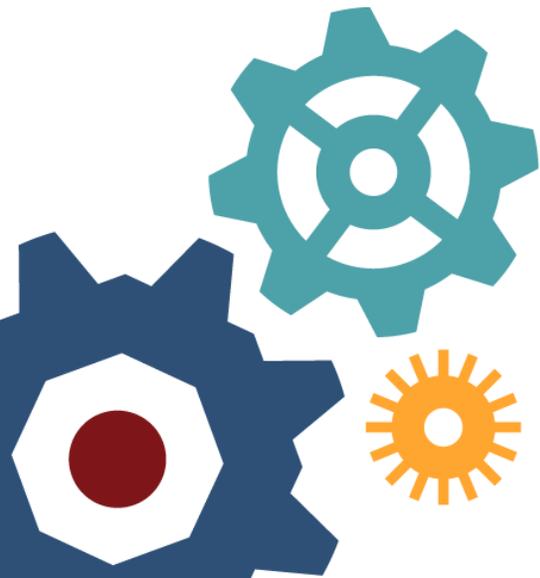
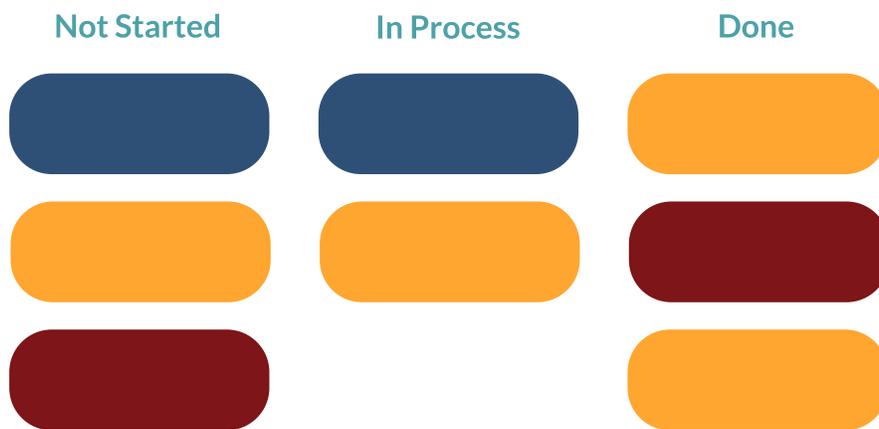
Scrum is not a technique or process for building products. It is a deceptively simple framework for managing projects. While originally created for software development, its iterative and incremental approach has been applied to many other types of projects, including Agile Service Management.

Agile Process Design adapts the Scrum roles, events and artifacts to the design and implementation of service management processes.

Kanban

Kanban is a deceptively simple but powerful method for visualizing and communicating workflow in order to reduce or eliminate work in progress. User stories expressed on sticky notes or index cards are moved through the Kanban columns until they are considered done. Any work that does not progress as expected is identified and addressed as excessive work in progress or an impediment. Kanban Boards are particularly useful tools for understanding impediments and team velocity.

Kanban Boards support Agile Service Management. A Kanban Board can be used to manage the flow of process design activities or to identify bottlenecks in processes such as Change Management, Release Management or Problem Management.



ITSM

While IT service management is often overlooked as an Agile practice, it is the integrated approach to managing IT services that actually enables IT to meet customer requirements in a timely manner. Whether formalized or not, ITSM processes transcend every aspect of the service lifecycle from design, development, deployment to operation and retirement.

By their nature, ITSM processes were not intended to be complex or bureaucratic. Agile Service Management strives to instill Agile values into scaled ITSM processes thereby increasing IT's end-to-end agility and ensuring consistency and speed. ITIL® is the most prominent ITSM framework.



Continual Service Improvement

Service Strategy
Service Transition
Service Operation
Service Design

Based on AXELOS® ITIL® material.
Reproduced under license.

DevOps

DevOps is a cultural and professional movement that stresses communication, collaboration, integration and automation between software developers and IT operations professionals.

The goal of DevOps is to cross traditional silos, instill shared accountabilities and improve the flow of work between development and operational teams. Improved workflow, shorter feedback loops, shared practices and automation help the entire IT supply chain increase its rate of production and time to value.

Continuous delivery

Continuous delivery is a software development practice where software is always in a releasable state. It allows organizations to rapidly deploy enhancements and fixes when needed. Continuous delivery relies on automated testing and deployment as well as good collaboration between development and operational teams (DevOps).

Continuous delivery is not the same as continuous deployment. Continuous deployment sends the release immediately into the production as soon as it is completed. Continuous delivery stages the release so it *could* be deployed quickly whenever it is needed. Processes such as Change Management and Release Management will need to be much more agile in environments where continuous delivery is practiced.

Continuous integration

Continuous integration is a software development practice where a team of developers create separate pieces of code that are regularly integrated onto a central server. Each integration goes through an automated build and test process to detect errors and defects.

Continuous integration leverages the capabilities and simultaneous work of multiple developers resulting in faster software builds. Early and regular integration testing identifies correctable defects at the source. Continuous integration aligns development standards within the organization and ensures that quality is built into the product throughout all phases of development.



Lean

Originally derived from the Toyota Production System, Lean is a production philosophy that seeks to create more value for customers with fewer resources and less waste. Lean considers any activity that does not contribute value as “waste.”



While conceived for manufacturing purposes, lean thinking has now been introduced across the business enterprise.

Lean IT

Applying the key ideas behind lean production to the development and management of IT products and services.

Lean Enterprise

Creating an organization that strategically applies the key ideas behind lean production across the entire business.

Agile Service Management strives to take a lean approach by eliminating waste, getting more done with fewer resources and creating customer value faster by making processes and services more agile.



6

Agile Process Design

What is an Agile Process?

An Agile process is one that delivers “just enough” structure and control to enable the organization to achieve its outcomes in the most expeditious, effective and efficient way possible. An Agile process is easy to understand, easy to follow and prizes its collaboration and outcomes more than its artifacts.

The characteristics of an Agile process include

- ▶ Having an accountable owner
- ▶ Clarifying everyone’s roles and responsibilities
- ▶ Benchmarking itself against Agile values and principles
- ▶ Being lean, efficient and expedient
- ▶ Being scalable
- ▶ Adapting to change

An Agile Approach to Process Design

The waterfall model is a sequential approach to software development where each phase of the process flows the project further downward until the product is built, tested, deployed and ready to maintain.

While the waterfall model is associated with software development, process designers often take a similar sequential approach in their projects.

There are several challenges when using the waterfall model to design processes including

- ▶ The rigidity of a sequential approach
- ▶ User feedback that comes late in the process
- ▶ The delays, rework and additional costs resulting from user feedback and testing errors
- ▶ The need for integration with processes not yet in design
- ▶ The extensive time required to build and deploy an entire end-to-end process
- ▶ The learning curve that users will experience when trying to normalize an entire process and its procedures

The bodies of knowledge behind ITIL® and other ITSM frameworks do not necessarily promote a waterfall or sequential approach to end-to-end process design. In fact, most frameworks recommend an integrated process approach as described ISO/IEC 20000, the international standard for service management.

While there is some benefit to methodically moving down a project waterfall, there is also a risk that climbing back up the waterfall may be more difficult and time consuming than expected.

Agile Process Design promotes a more adaptive approach by

- ▶ Implementing each process in smaller, more frequent increments
- ▶ Encouraging shorter feedback and feed-forward loops
- ▶ Shaping future increments based on current business conditions
- ▶ Taking a holistic approach to building, maturing and integrating process activities
- ▶ Giving process practitioners time to absorb and institutionalize new behaviors
- ▶ Getting more “done” and delivering value more quickly

The net result will be an agile process that delivers “just enough” structure and control while

- ▶ Tying success measures to business outcomes
- ▶ Engaging stakeholders and soliciting input and feedback
- ▶ Enabling effective communication
- ▶ Integrating other processes and frameworks
- ▶ Introducing timely improvements
- ▶ Having simple documentation



Minimum Viable Process (MVP)

Like a Minimum Viable Product in software development, a Minimum Viable Process has three characteristics

- ▶ It has enough value that people are willing to use it initially
- ▶ It demonstrates enough future benefit to retain early adopters
- ▶ It provides a feedback loop to guide future capabilities

It is much easier to add to a process incrementally than it is to scale a process back later. A MVP approach ensures that the core elements of a process are designed and introduced first. It strips away the “wants” from the “needs.” It provides a basis for dialogue and feedback so that future development will provide ongoing value to those who rely on the process.





7

Scrum Basics

In many organizations, Scrum has become the preferred method for managing software development projects. Scrum embodies the values and principles of the Agile Manifesto and focuses on getting more done. By its own admission, Scrum is lightweight, simple to understand yet difficult to master.

Agile Service Management captures the essence of Scrum within the context of process design and process improvement. While some of the roles, events and artifacts have been adapted, the core concepts, rules and processes are the same.

A comparison chart of Scrum and Agile Service Management counterparts precedes each section within Agile Service Management.

The Pillars of Scrum

Scrum is founded on empirical process control where knowledge comes from experience, decisions are based on what is known and three pillars underpin the entire framework.



The three pillars of Scrum are

Transparency

Workflows and progress towards the Sprint Goal are made visible through daily standups, Kanban Boards, planned events and other methods. Common standards, vocabulary and definitions are shared by all stakeholders.

Inspection

Scrum artifacts are regularly inspected to help assess progress towards or deviations from a Sprint Goal.

Adaptation

Workflows are adapted as soon as possible if a deviation, impediment or other need is detected during inspection.

Scrum Values

Scrum defines five values that Scrum teams should embrace and demonstrate at all times.



Important Scrum Terms

The following are key terms and concepts that will be used throughout this Guide. In Scrum, these are defined in the context of a “product”. In Agile Service Management, they may be adapted to the context of a “process”. A more complete glossary is appended to the end of this guide.

Scrum Guide

A document that describes Scrum concepts and practices, written by Ken Schwaber and Jeff Sutherland

Product/Process Backlog

A prioritized list of functional and non-functional requirements for a system or process; usually expressed as user stories

User Story

A statement written from the user’s perspective that describes what a user wants to do with a feature of the software or aspect of a process

Increment

Potentially shippable completed work that is the outcome of a Sprint

Sprint

A period of 2-4 weeks during which an increment of product work is completed

Sprint Goal

The purpose and objective of a Sprint, often expressed as a business problem that is going to be solved

Sprint Backlog

Defines the work that must be completed during the Sprint

Burndown Chart

Shows how much work is left over a period of time for a product or Sprint

Definition of Done

Shared understanding of what it means for work to be complete



Definition of Done

Shared understanding of what it means for work to be complete

Timebox

The maximum duration of an event

Daily Scrum

A fifteen-minute daily meeting that synchronizes work completed since the prior meeting and forecasts the work to be done before the next one

Impediment

Anything that prevents a Team member from performing work as efficiently as possible

Velocity

How much product or process backlog effort a Team can handle in a single Sprint

Additional definitions are contained in the glossary at the back of this guide.

Scrum Components

The Scrum framework is built around the interaction and rules that govern roles, artifacts and events.

- ▶ 3 Roles
- ▶ 4 Artifacts
- ▶ 5 Events

In the Scrum Guide, these are defined in the context of a “product.” In Agile Service Management these are adapted to the context of a “process.”

Product vs Process

Software products and service management processes are not fundamentally different. Both shape behaviors, enable people to “do something” and have defined inputs and outputs. Customer requirements drive design and development and are usually captured in some type of document or repository. Products and processes each benefit from having an accountable owner. Cross-functional expertise is essential in order to create and maintain the product or process. Products are often built to replicate processes.

Scrum roles, artifacts and events can be adapted to Agile Service Management, allowing ITSM processes to be designed iteratively and in complete, potentially releasable increments.

8

Agile Service Management Roles

Counterparts

Scrum Role

The Team
Product Owner
ScrumMaster

Agile Service Management Role

The Team
Process Owner
Agile Service Manager

The Agile Service Management Team

An Agile Service Management Team is

- ▶ Self-organizing
- ▶ Cross-functional
- ▶ Without titles
- ▶ Without sub-teams
- ▶ Accountable for the work produced as a whole regardless of individual skills or experience

A self-organizing team understands what it takes to get things done. For each increment of work, they are provided a goal, a backlog of tasks, a completion date and a clear and shared Definition of Done. The Team agrees on an approach for completing the work and meeting the goal. Essentially, the Team is given the “what” and they collectively determine the “how.”

Successful self-organizing teams are

- ▶ Stable
- ▶ Trusting
- ▶ Empowered
- ▶ Motivated
- ▶ Accountable
- ▶ Focused
- ▶ Business centric
- ▶ Communicative
- ▶ Quality driven

These characteristics are often matured over time and experience.

Different perspectives and cross-functional skills are essential to an Agile Service Management Team. Membership should include a

- ▶ Process Owner
- ▶ Agile Service Manager
- ▶ Customer and/or process practitioner
- ▶ Process architect
- ▶ Tool administrator
- ▶ Change Manager
- ▶ Documenter

The Agile Service Management Team must include a customer or practitioner representative.

Each member of the Team will work on some aspect of items from the Process Backlog. None are observers.

The Team should have at least three members but no more than nine to ensure sufficient cross functional skills and the ability to self organize. Members may be on multiple teams, although it is recommended that an individual not work on more than two process Teams at any given time.

Velocity

Velocity is a metric that estimates how much of the Process Backlog a Team can handle in a single Sprint. The more mature and stable the Team, the higher the Team's velocity (or ability to absorb and complete work).

Velocity is often measured by work accomplished during past Sprints and serves as a predictor of future Team performance.

The Agile Process Owner

Most service management frameworks advocate for a Process Owner role that is accountable for the end-to-end results of the process.

Frameworks such as ITIL® do a good job of describing the responsibilities of a Process Owner for a specific process. The Agile Process Owner role supplements the Process Owner role description by adding responsibilities for integrating Scrum practices and instilling agile thinking into the process.

The key responsibility of the Process Owner is to create, manage, prioritize and own the Process Backlog. The Process Backlog is the single source of current or future requirements for a single ITSM process including activities, tools, plans, interfaces, documentation, training and improvements.

The Process Owner has ultimate authority over the items in the Process Backlog and ensures that the items are clear and visible. This role understands how to prioritize items in the Process Backlog and helps the Team understand the next process increment. The Process Owner is the only individual who can change the Team's direction and/or add, remove or cancel items in a Sprint.

Other responsibilities of a Process Owner include

- ▶ Communicating the process' vision and goals
- ▶ Ensuring that Agile values are embedded into the process so that outcomes and collaboration are prized over tools and artifacts
- ▶ Clarifying a Definition of Done for each process increment
- ▶ Inspecting the progress and status of the process after each Sprint
- ▶ Auditing and reviewing the process on a regular basis
- ▶ Prioritizing improvements in the Process Backlog
- ▶ Being accountable for overall process quality and deliverables

The Process Owner is not necessarily responsible for performing any or all of the tasks associated with managing a process. Depending on the size and complexity of the organization, the Process Owner may assign one or more roles to oversee day-to-day process execution.



The Agile Service Manager

The Agile Service Manager is the operational process counterpart to development's product ScrumMaster. While the context is different, the role and its responsibilities are very similar.

Responsibilities of the Agile Service Manager include

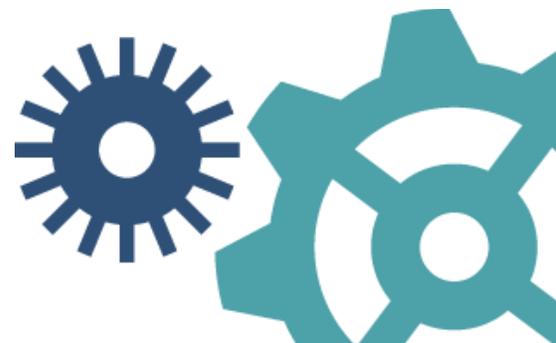
- ▶ Instilling agile thinking into the management of IT services.
- ▶ Ensuring that Agile values and principles are understood and applied
- ▶ Helping the Team adhere to Scrum practices and rules
- ▶ Refocuses IT Teams to the items on the left of the Agile Manifesto instead of prizing the items on the right
- ▶ Removing impediments whenever possible
- ▶ Facilitating Scrum meetings
- ▶ Serving as a facilitator, educator, protector and coach

The Agile Service Manager works closely with the Process Owner and the Team to get the work done.

The Agile Service Manager does not manage the Team. The Team is self-organizing. The Agile Service Manager is a servant-leader that helps the Process Owner integrate the guidance between ITSM and Scrum in order to build and maintain an accurate and relevant Process Backlog. The Agile Service Manager coaches the Team and helps the members write effective process-related user stories.

Most importantly, the Agile Service Manager protects the Team and does everything possible to ensure its success. This includes helping those outside the Team understand how to (and how not to) interact with the Team. The Agile Service Manager educates the organization on Agile values and Scrum practices so that everyone knows what to expect.

The Agile Service Manager bridges a relationship with the organization's software development ScrumMasters. Cross-populating Agile practices, vocabulary and automation across all sides of IT will serve to increase speed and consistency. Collaboration between Agile Service Managers and ScrumMasters helps to create and maintain a DevOps culture.



9

Agile Service Management Artifacts

Counterparts

Scrum Artifact

Product Backlog
Increment
Sprint Backlog
Burndown Chart

Agile Service Management Artifact

Process Backlog
Process Increment
Sprint Backlog
Burndown Chart

The Process Backlog

The Process Backlog is the single source of current or future requirements including process activities, tool updates, plans, interfaces, documentation, training and improvements for a single ITSM process. The Process Backlog continually evolves, is regularly re-prioritized and is never complete. It exists as long as the process exists. It is solely owned and managed by the Process Owner.

The form and format of the Process Backlog is not prescribed – items can be captured in anything from a Kanban Board to a spreadsheet to a database. It should be visible to all process stakeholders and readily available for inspection.

Each item in the Process Backlog should be expressed as a user story.

The Process Backlog and User Stories

A user story is a simple statement that describes what a user or process practitioner wants from an aspect of the process. It is always written from the user's perspective and in their words. It is not meant to include all of the details about the process aspect but is intended to encourage further dialogue and collaboration. User stories are generally captured on index cards or sticky notes. That fact alone should demonstrate how succinct the user story should be.

*User stories generally follow the formula
"As a (role), I want to (do something) so I can (achieve something)"*

In 2003, Bill Wake recommended the INVEST model to describe the elements of a good user story

- ▶ Independent
- ▶ Negotiable
- ▶ Valuable
- ▶ Estimable
- ▶ Small
- ▶ Testable

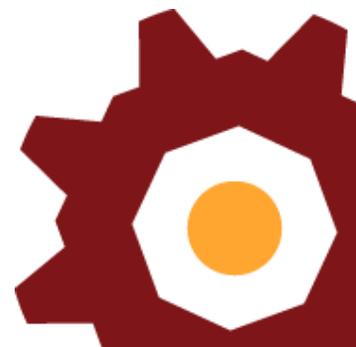
A process user story can be written for any aspect of the process including an activity, a procedure or process artifact.

Process Backlog Refinement

The Process Backlog should be refined regularly to add detail, estimates and prioritization to Process Backlog items. The Process Owner and the Team will determine when and how the Process items should be reviewed and refined. As items become higher priorities, the amount of detail needed will become greater and therefore refinement more necessary. Details can come from a variety of sources, but the Team is responsible for updating the work estimates as important inputs into Sprint Planning.

Each user story in the Process Backlog should be refined with at least the following details

- ▶ A unique reference number for querying
- ▶ The stakeholders or customers
- ▶ An assigned priority
- ▶ The estimated number of hours to complete
- ▶ Who the story has been assigned to
- ▶ The anticipated Sprint that will include this story
- ▶ An approximate date of completion



Process Increments

A Process Increment is a potentially releasable and completed aspect of the process that was the pre-defined outcome of a Sprint. A Process Activity Increment could be an activity, procedure or work instruction.

The Process Increment is defined during the Sprint Planning Meeting. It is built during the Sprint from items in the Sprint Backlog.

A Process Increment is considered finished when it meets the agreed Definition of Done. It is demonstrated and discussed during the Sprint Review meeting. The Process Owner then decides whether and when the Process Increment should be released.

The “Definition of Done”

The Team and process stakeholders must share an understanding of the “definition of done” for each Process Backlog item or Process Increment.

The Definition of Done is critical to Sprint Planning. It guides how many items can be added to the Sprint Backlog and reasonably accomplished during the Sprint. As the Team’s velocity increases, their ability to get more “done” in each Sprint will also increase.

When is a Process Increment Done?

The Definition of Done may vary from Process Increment to Process Increment depending on scope of work in the Sprint Backlog. Process Activity Increments should be considered “done” when the following questions have been answered

- ▶ Have the inputs, outputs, triggers and outcomes been defined?
- ▶ Have procedures been defined and documented?
- ▶ Have roles and responsibilities been mapped?
- ▶ Have tools and automation been updated?
- ▶ Have policies been reviewed and updated if necessary?
- ▶ Has training been developed and scheduled?
- ▶ Has communication been drafted?
- ▶ Have suppliers been engaged?
- ▶ Have all of these been reviewed and tested by stakeholders and process practitioners?

In simple terms, the Definition of Done is when you do not need to think about it anymore.



The Sprint Backlog

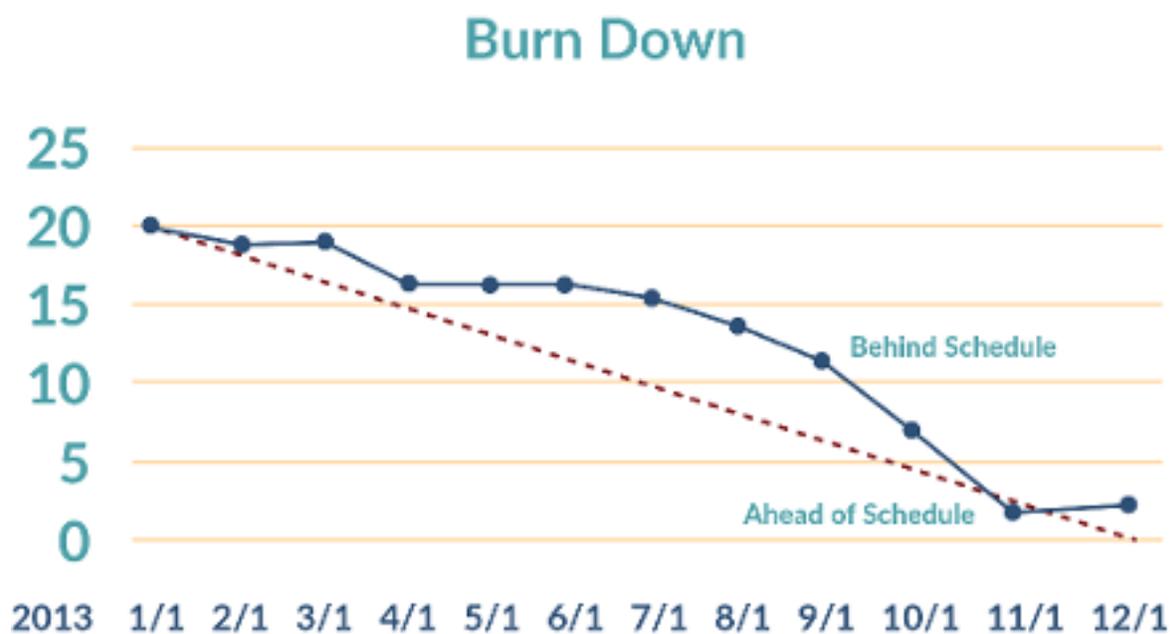
The Sprint Backlog is a subset of the Process Backlog and forecasts what increment of the process will be designed during the next Sprint. It is created during the Sprint Planning Meeting and documents all of the items that will be necessary in order to meet the Sprint Goal. It should be highly visible and available for inspection.

The Sprint Backlog provides a central artifact around which the Team can self-organize in order to meet the Sprint Goal. It should have enough detail so that the Team understands the Definition of Done and can inspect progress during the Daily Scrum.

The Sprint Backlog expires at the end of the Sprint – hopefully with all items completed. Outstanding items do not automatically carry over to the next Sprint. They are reprioritized with other Process Backlog items and considered during the next Sprint Planning Meeting.

Burndown Charts

A Burndown Chart is a graph that shows the trend of completed and remaining work over a specified time period such as the timebox of the Sprint or the planned rollout of the new or improved process. The most common types of Burndown Charts are the Process Burndown and the Sprint Burndown.



The Sprint Burndown is particularly important since it visually demonstrates whether the Team is on course to complete the Sprint on time. It also shows where they may be ahead or behind schedule, whether they are under or over-allocated. The Burndown Chart is a useful tool for conducting a post-sprint analysis of the Team's velocity.

10

Agile Service Management Events

Counterparts

Scrum Event

- Release Planning Meeting (Optional)
- Sprint Planning Meeting
- Sprint
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Agile Service Management Event

- Process Planning Meeting (Optional)
- Sprint Planning Meeting
- Sprint
- Daily Scrum
- Sprint Review
- Sprint Retrospective



Timeboxes

Scrum prescribes a maximum duration or “timebox” for each event. The timebox range depends on the length of the Sprint (from two weeks to one month).

Timeboxes	
Event	Timebox
Process Planning Meeting (Optional)	Not Timeboxed
Sprint Planning Meeting	4 to 8 Hours
Sprint	2 to 4 Weeks
Daily Scrum	15 Minutes
Sprint Review	2 to 4 Hours
Sprint Retrospective	1.5 to 3 Hours

The Process Planning Meeting

The Process Planning Meeting is a high level planning event that establishes the Process Definition Document for a single process. Outcomes of the meeting include a definition of the

- ▶ Goals, objectives, inputs and outputs of the process
- ▶ Features/activities of the process
- ▶ Expected integration with other processes
- ▶ Stakeholders
- ▶ Necessary tools
- ▶ Regulatory, governance or policy requirements
- ▶ Major risks
- ▶ Delivery date and cost

This event is not timeboxed, mainly because it may take multiple meetings to establish the high level Process Definition Document. While Scrum considers a Release Planning Meeting to be optional, Agile Service Management strongly recommends that this event occur in order to plan the end-to-end process before it is broken down into Process Increments.

The ITSM Process Design Guide by Donna Knapp provides detailed guidance on creating a Process Definition Document.

The Sprint Planning Meeting

The Sprint Planning Meeting is timeboxed for 4 to 8 hours which demonstrates the importance of proper Sprint Planning. The Agile Service Manager facilitates the meeting and the Process Owner describes the next Process Increment. The entire Team collaborates on planning the details of the next Sprint.

The primary purpose of the Sprint Planning Meeting is to

- ▶ Establish the Sprint Goal
- ▶ Define what increment of the Process Backlog will be completed during the Sprint
- ▶ Determine how the increment will be done
- ▶ Ensure that the Team has all of the necessary skills and resources to complete the increment
- ▶ Define any dependencies or integrations with other processes that need to be considered
- ▶ Create a Sprint Backlog

Inputs to the Sprint Planning Meeting include the Process Backlog, the past velocity of the Team, the availability of Team members and the dependencies on other processes and tools. Only the Team can determine how much it can accomplish during the next Sprint.

The Sprint Planning Meeting is also where the Team begins to self-organize by determining how they will accomplish the Sprint Goal. They plan their approach and prioritize the items going into the Sprint Backlog. By the end of the Sprint Planning meeting, the Team should be able to articulate what they are going to accomplish and how they are going to do it.

The Sprint

A Sprint is a period of 2 to 4 weeks during which the work needed to meet the Sprint Goal is performed. The Process Increment is built from items in the Sprint Backlog based on the approach agreed to during Sprint Planning. Progress is inspected during the Daily Scrum and visualized on the Sprint Burndown Chart. The Sprint is guided by the Definition of Done.

During the Sprint, the Agile Service Manager keeps the Team focused, coaches the members and stakeholders on Scrum practices and protects the Team from outside distractions. The Agile Service Manager also removes impediments whenever possible. The Process Owner ensures that no one else attempts to change the Team's priorities or tasks during the Sprint.

Agile Service Management embraces the Scrum principle of being iterative and incremental. Every Sprint is considered an iteration that progresses the service management process forward by building Process Increments. When one iteration is completed, another is planned and repeated until all increments of the process are done.



Sprint Types

Agile Service Management defines three basic types of Sprints

Strategic Sprint

A Strategic Sprint is committed to working on the underpinning items from the Process Backlog that are essential to the process but do not usually appear on the process flowchart. They include

- ▶ Establishing a high level process definition document
- ▶ Allocating resources
- ▶ Inventorying and assessing existing tools
- ▶ Creating new or updating existing policies
- ▶ Mapping stakeholders to high-level activities
- ▶ Drafting training and communication plans

Strategic Sprints follow the rules of any other type of Sprint. They are guided by a Sprint Goal, agreed Definition(s) of Done and produce a Process Increment that is demonstrated during a Sprint Review.

The first Strategic Sprint will establish a high level Process Definition Document. Subsequent Strategic Sprint iterations can be planned when they make sense to do so. Planning simultaneous Strategic Sprints from multiple processes may help to ensure alignment and integration.

Process Activity Sprint

Process Activity Sprints are planned in order to complete a Process Increment for a single activity, procedure or work instruction including

- ▶ Roles and responsibilities
- ▶ Timelines and escalations
- ▶ Documentation
- ▶ Metrics
- ▶ Updated tools and automation
- ▶ Interfaces or dependencies on other processes
- ▶ Training or communication

Some activities have too many user stories or are too large to complete in a single Process Activity Sprint. In this case, the Process Owner should logically group related user stories into smaller Process Increments that can be planned over multiple Process Activity Sprints. The collective Process Increments could be released either separately or together.

Continual Service Improvement (CSI) Sprint

A CSI Sprint commits a cycle of work to implementing prioritized improvements from the Process Backlog. CSI Sprints are based on Deming's Plan-Do-Check-Act (PDCA) improvement cycle.

A CSI Sprint is usually undertaken as part of Agile Process Improvement. It is an opportunity to adapt the process based on input and feedback from prior Process Increment releases.

CSI Sprints should be regularly planned throughout the lifecycle of the process to maintain or increase the process' agility.

Typecasting Sprints is done solely for the purpose of ensuring that all aspects of the process are addressed. There is no limit to the number or frequency of each type of Sprint. There may also be other Sprint cycles that do not fall into a particular type and are just iterations to progress the process forward.

The Daily Scrum

The Daily Scrum (sometimes called a Daily Standup) is timeboxed for 15 minutes. It is not a status meeting but a daily opportunity to inspect progress towards the Sprint Goal and identify impediments as quickly as possible.

During the Daily Scrum, each Team member in turn shares

- ▶ What he/she has accomplished since the last meeting
- ▶ What he/she is going to do before the next meeting
- ▶ What obstacles are in his/her way

While observers and stakeholders may attend, Team members are the only ones allowed to speak. Questions are not allowed during the timebox. The Agile Service Manager facilitates the meeting.

There may be a temptation to hold the Daily Scrum less frequently. The importance of this daily inspection should not be undervalued – the faster deviations and impediments are identified, the greater the opportunity to meet the Sprint Goal and get more done. Fifteen minutes a day during an active Sprint is usually time well spent.



The Sprint Review

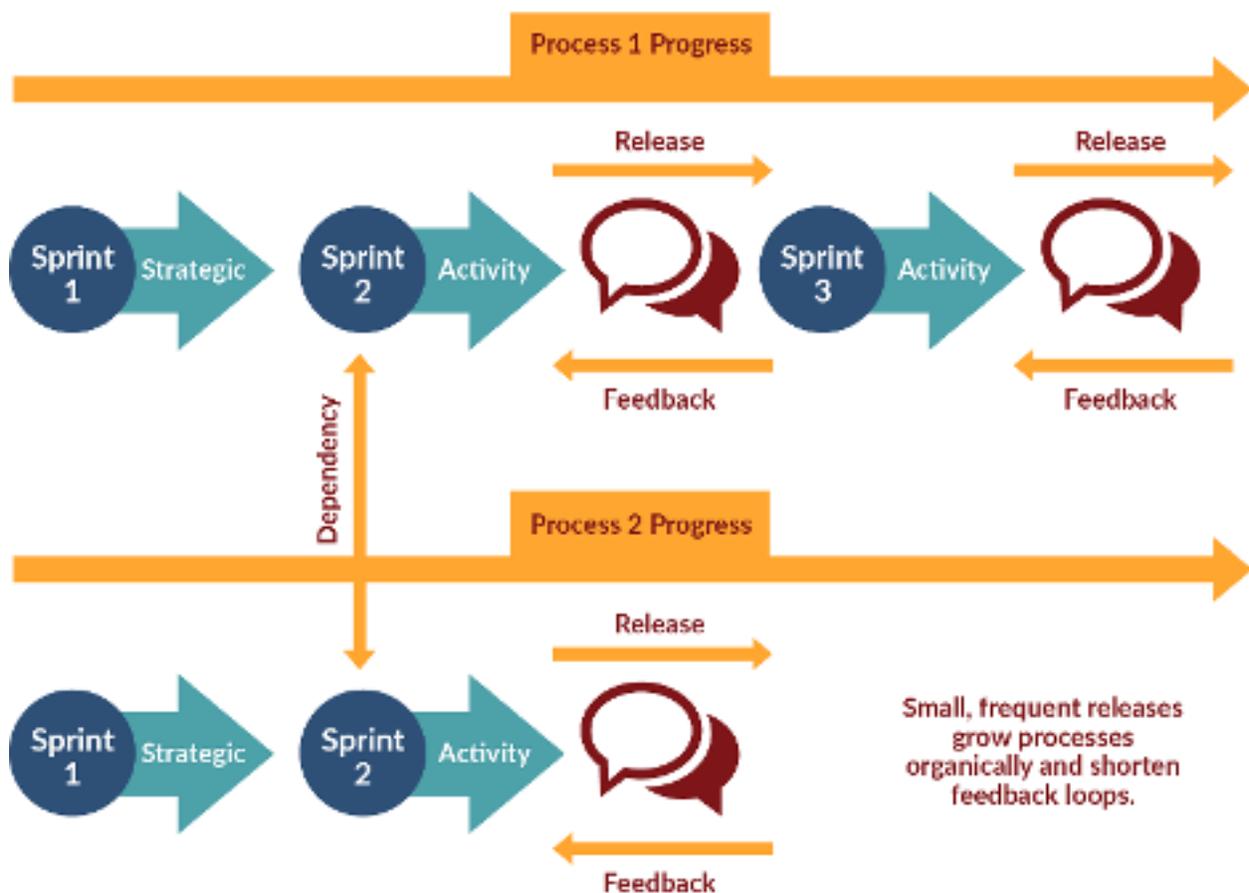
The Sprint Review is timeboxed for 2-4 hours and is attended by the Team and stakeholders. It is an important opportunity for transparency, inspection and adaptation (the pillars of Scrum).

During the Sprint Review, the Team demonstrates the aspects of the process that were designed during the last Sprint. The Team shares the challenges they faced, successful resolutions and outstanding issues. The Process Owner explains the current state of the process and the Process Backlog. The Process Owner also describes any feedback received from process practitioners about any previously released Process Increments. A decision on whether the current Process Increment will be released is made.

The Sprint Review allows the Team and stakeholders to discuss the next steps for the process as input to the next Sprint Planning Meeting.

Should the Increment be Released?

Process Implementation One Increment at a Time



One of the key decisions made during the Sprint Review is whether or not to release the Process Increment. While releasing aspects of service management processes incrementally gives the organization time to adopt and adapt to new behaviors, there are several considerations that should be discussed including whether

The Process Owner will decide if the Process Activity Increment should be released. Consideration include whether

- ▶ The organization is ready and receptive
- ▶ It won't confuse practitioners
- ▶ It delivers business value
- ▶ The increment can stand alone
- ▶ The status of any dependencies
- ▶ The process increment will not affect the accuracy or validity of data or reporting
- ▶ It does not contribute to "change fatigue"

Change fatigue can occur when too many changes are made to a process in rapid succession.

Some of the benefits of releasing a Process Increment include

- ▶ Changing organizational behaviors one increment at a time
- ▶ Capturing more data or information
- ▶ Shortening feedback loops and using feedback to influence future Process Increments
- ▶ Helping the process adapt to changing requirements as it is slowly being matured
- ▶ Identifying and aligning dependencies on other processes
- ▶ Encouraging an integrated approach to service management
- ▶ Keeping tools relevant and updated



Sprint Retrospective

The Sprint Retrospective is an internal opportunity for the Team to reflect on and inspect the progress and organization of the last Sprint. In some ways, it resembles the form and format of a post-implementation review in that it addresses

- ▶ What did we do right?
- ▶ What could we have done better?
- ▶ What have we learned?
- ▶ What will do differently next time?

In the spirit of continual improvement, the Team also discusses

- ▶ Team composition and skill sets
- ▶ Tools
- ▶ Meeting logistics
- ▶ The Definition of Done
- ▶ Internal and external communications
- ▶ Input and feedback from stakeholders
- ▶ Velocity
- ▶ Process performance thus far

The Sprint Retrospective is timeboxed for 1.5 to 3 hours. While the temptation may be to go from the Sprint Review directly into the next Sprint Planning Meeting, encouraging the Team to take the time to review and improve their past performance will absolutely increase their maturity and velocity.



11

Agile Process Improvement

Designing agility into a process is a great first step towards Agile Service Management. However, the ability to remain agile once a process is in use is the bigger challenge and requires a long term commitment. If left unchecked, processes can become complex and bureaucratic over time. The leap from “just enough” to “too much” process can happen almost overnight.

There is also a risk that people will revert to old ways and potentially “not enough” process control. At that point, the temptation might be to put in place a more bureaucratic approach.

Agile Process Improvement’s goal is to ensure that service management processes are not

- ▶ Bureaucratic
- ▶ Unclear
- ▶ Constrained
- ▶ Time consuming
- ▶ Irrelevant
- ▶ Circumvented
- ▶ Nice on paper, but...

Agile Process Improvement is a key aspect of Continual Service Improvement. Whether a new or existing process, improvement opportunities should be assessed against Agile values and principles from concept to retirement.

Agile Process Improvement Audits and Reviews

Agile Process Improvement requires Process Owners to conduct regular audits and reviews of their processes. While process audits are often undertaken to determine compliance, Agile Process Improvement audits and reviews also help to

- ▶ Identify and eliminate waste and bottlenecks
 - ▶ Detect drifting trends
 - ▶ Benchmark against Agile values and principles
 - ▶ Assess ongoing relevancy
 - ▶ Maintain or move closer to “just enough” structure and control
 - ▶ Improve effectiveness, efficiency and agility
 - ▶ The process audit will include a review of process artifacts including
 - ▶ The Process Definition Document
 - ▶ Documented procedures
 - ▶ Plans
 - ▶ Documentation
 - ▶ Tools and databases
 - ▶ SLAs, OLAs, contracts
- The agility of the process artifacts will help determine whether the process is providing “just enough” structure and control.

For each artifact, process stakeholders, Team members and others may be asked

- ▶ Is it simple to use, read and/or understand?
- ▶ It is timely?
- ▶ Does it enable the process' effectiveness and efficiency?
- ▶ Can it easily measure compliance, usage or achievements?
- ▶ Is it readily available?
- ▶ Is it relevant to the current business environment?
- ▶ Do you use it and is it helpful? When was the last time it was updated?

Different perspectives will help the Process Owner understand how to keep or improve the value of the process in the management of IT services.

The Agile Service Manager should help facilitate the audits and assist the Process Owner in collecting and evaluating the output in line with Agile values and principles.

Agile Process Improvement as an essential element of Continual Service Improvement ensures that agility is as important to the ongoing relevancy of a process as effectiveness and efficiency. Agile Process Improvement should therefore overarch the entire service lifecycle.

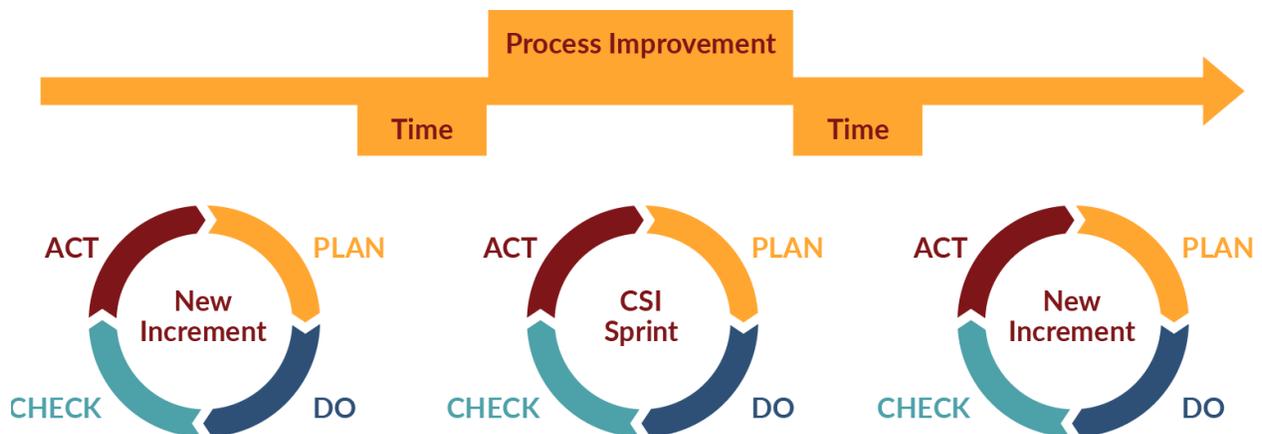
The Process Backlog as the CSI Register

There will be many opportunities and recommendations for improvements during Continual Service Improvement and Agile Process Improvement. ITIL® and other ITSM frameworks encourage the creation and maintenance of a Continual Service Improvement (CSI) Register – a repository for capturing and prioritizing recommended improvements for a process or service.

In Agile Service Management, the Process Backlog serves as the CSI Register in that it

- ▶ Documents improvement opportunities and recommendations
- ▶ Maps improvements to user stories
- ▶ Prioritizes user story improvements according to business requirements
- ▶ Can be used as the basis for planning CSI Sprints

As described above, a CSI Sprint is an opportunity to commit a cycle of work to implementing a Process Increment of prioritized improvements. It is based on Deming's Plan-Do-Check-Act approach and is critical to successful Agile Service Management. A CSI Sprint can occur whenever it makes sense for new or existing processes.



12

Tools for Agile Service Management

Agility is often supported by automation. If done well, automated processes or procedures can be more consistent, effective, efficient, expeditious and provide long term data repositories.

All tools currently used as part of an ITSM program are still (if not more) relevant to Agile Service Management including

- ▶ ITSM suites (some of which have Agile modules)
- ▶ Automated testing, quality assurance and deployment
- ▶ Monitoring and event management
- ▶ Dashboards
- ▶ Metrics and analytics
- ▶ Flowcharts and drawing
- ▶ Project management



There are also many Agile tools that can

- ▶ Capture and maintain the Product Backlog
- ▶ Track user stories
- ▶ Plan and manage a Sprint
- ▶ Burndown the process and Sprint
- ▶ Visualize workflows (Kanban)
- ▶ Analyze results and report on velocity
- ▶ Track testing and pilots
- ▶ Capture feedback
- ▶ Automate aspects and activities of each process

Some of the ITSM tools may already be in use by the organization's operational teams and software development team. The ability to leverage and share tools may help to cut costs while potentially increasing collaboration.

It is important to note that technology alone will not make an organization agile.



12

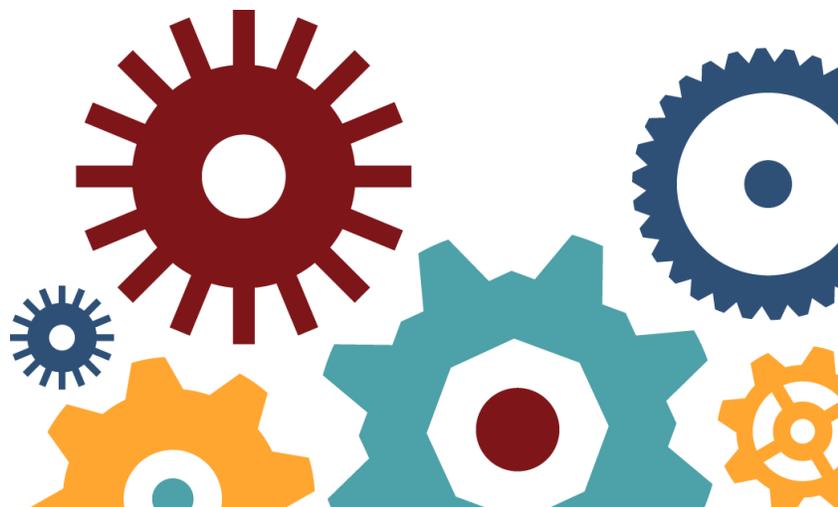
Getting Started

Agility does not happen overnight. Moving an organization to an Agile mindset and an Agile Service Management approach takes practice and perseverance. Identifying an organization's "just enough" level takes time and experience. Changing the thinking and behavior of individuals takes repetition, openness and patience. Embracing the Scrum values of Commitment, Focus, Respect, Openness and Courage is essential.

Wherever you are in your Agile Service Management journey, remember that it is important to understand what it means to "be agile" before you attempt to "do agile (or Scrum)."

Start simple and stay simple. Pick one process to pilot as a learning experience. Identify a Process Owner, Agile Service Manager and stakeholders. Build a small self-organizing team with cross-functional skills and appropriate levels of ITSM and Agile Service Management training. Engage stakeholders and encourage feedback.

Don't rush. Start with a Minimum Viable Process and move forward from there. Introduce the new or improved process in small, frequent increments. Give the organization time to absorb, adopt and adapt to new behaviors. Mature the processes holistically and organically. Small, short term wins will deliver greater wins in the long term.



Agile Service Management

Glossary of Terms

Term	Definition
Agile	A project management method for complex projects that divides tasks into small “sprints” of work with frequent reassessment and adaptation of plans.
Agile Manifesto	A formal proclamation of four key values and 12 principles to guide an iterative and people-centric approach to software development.
Agile Principles	The twelve principles that underpin the Agile Manifesto.
Agile Process Design	The aspect of Agile Service Management (Agile SM) that applies the same Agile approach to process design as developers do to software development.
Agile Process Improvement	The aspect of Agile SM that aligns Agile values with ITSM processes through continuous improvement.
Agile Service Management (Agile SM)	A framework that ensures that ITSM processes reflect Agile values and are designed with “just enough” control and structure in order to effectively and efficiently deliver services that facilitate customer outcomes when and how they are needed.
Burndown Chart	A chart showing the evolution of remaining effort against time.

Term	Definition
Continuous Delivery	A software development practice where software is always in a releasable state.
Continuous Integration	A software development practice where members of a team code separately but integrate their work at least daily. Each integration goes through an automated build and test to detect errors and defects so as to allow faster deployments.
Critical Success Factor	Something that must happen for a process, plan, project or other activity to succeed.
CSI Register	A vehicle for recording and managing improvement opportunities throughout their lifecycle.
Daily Scrum	A daily timeboxed event of 15 minutes or less for the Team to re-plan the next day of work during a Sprint.
Definition of Done	A shared understanding of what it means for work to be complete.
DevOps	A cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.
Impediment	Anything that prevents a Team member from performing work as efficiently as possible.
Increment	Potentially shippable completed work that is the outcome of a Sprint.
ITIL®	Set of best practice publications for IT service management. Published in five core books representing the five stages of the IT service lifecycle: Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement.

Term	Definition
INVEST	A mnemonic was created by Bill Wake as a reminder of the characteristics of a quality user story.
Kanban	A method for visualizing and communicating workflow in order to reduce or eliminate work in progress.
Key Performance Indicator	Key metric used to measure the achievement of critical success factors. KPIs underpin critical success factors and are measured as a percentage.
Lean Thinking	The goal of lean thinking is to create more value for customers with fewer resources and less waste. Waste is considered any activity that does not add value to the process.
Minimum Viable Product	The most minimal version of a product that can be released and still provide enough value that people are willing to use it.
Plan-Do-Check-Act	A four-stage cycle for process management and improvement attributed to W. Edwards Deming. Sometimes called the Deming Cycle or PDCA.
Post Implementation Review	A review that takes place after a change or a project has been implemented that assesses whether the change was successful and opportunities for improvement.
Potentially Shippable Product	An increment of work that is “done” and capable of being released if it makes sense to do so.
Procedure	Step-by-step instructions that describe how to perform the activities in a process.
Process	Interrelated work activities that take specific inputs and produce specific outputs that are of value to a customer.
Process Backlog	A prioritized list of everything that needs to be designed or improved for a process including current and future requirements.

Term	Definition
Process Customer	A recipient of a process' output.
Process Owner	Role accountable for the overall quality of a process and owner of the Process Backlog.
Process Planning Meeting	A high level event to define the goals, objectives, inputs, outcomes, activities, stakeholders, tools and other aspects of a process. This meeting is not timeboxed.
Process Supplier	A creator of process input.
Product Backlog	A prioritized list of functional and non-functional requirements for a system usually expressed as "User Stories."
Product Backlog Refinement	An ongoing process of adding detail, estimates and order to backlog items. Sometimes referred to as Product Backlog grooming.
Product Owner	An individual who manages the Product Backlog and ensures the value of the work that the Team performs.
Release Planning Meeting	A non-timeboxed event that establishes the goals, risks, features, functionality, delivery date and cost of a release. It also includes prioritizing the Product Backlog.
Scrum	A simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create "just enough" structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge.
Scrum Components	Scrum's roles, events, artifacts and the rules that bind them together.

Term	Definition
Scrum Guide	The definition of Scrum concepts and practices, written by Ken Schwaber and Jeff Sutherland.
ScrumMaster	An individual who ensures that the Team adheres to Scrum practices, values and rules.
Scrum Team	A self-organizing team consisting of a Product Owner, Development Team and ScrumMaster.
Scrum Values	A set of fundamental values and qualities underpinning the Scrum framework: commitment, focus, openness, respect and courage.
Self-Organizing	The management principle that teams autonomously organize their work. Self-organization happens within boundaries and against given goals. Teams choose how best to accomplish their work, rather than being directed by others outside the team.
Sprint	A period of 2-4 weeks during which an increment of product work is completed.
Sprint Backlog	Defines the work that must be completed during the Sprint.
Sprint Goal	The purpose and objective of a Sprint, often expressed as a business problem that is going to be solved.
Sprint Planning Meeting	A 4-8 hour timeboxed event that defines the Sprint Goal, the increment of the Product Backlog that will be done during the Sprint and how it will be done.
Sprint Retrospective	A 1.5-3 hour timeboxed event during which the Team reviews the last Sprint and identifies and prioritizes improvements for the next Sprint.

Term	Definition
Sprint Review	A timeboxed event of 4 hours or less where the Team and stakeholders inspect the work resulting from the Sprint and update the Product Backlog.
Strategic Sprint	A 2-4 week timeboxed Sprint during which strategic elements that were defined during the Process Planning Meeting are completed so that the Team can move on to designing the activities of the process.
Timebox	The maximum duration of an event.
User Story	A statement written from the user's business perspective that describes how the user will achieve a goal from a feature of the product. User stories are captured in the Product Backlog.
Velocity	How much Product Backlog effort a team can handle in a single Sprint.
Waste	Any activity which does not add value to a process.
Waterfall	A linear and sequential approach to software development.



Certified Agile Service Manager (CASM)[®]

Integrate Agile Thinking into Service Management



COURSE OBJECTIVES

The learning objectives for Certified Agile Service Manager (CASM) include an understanding of:

- What does it mean to “be agile?”
- The Agile Manifesto, its core values, and principles
- Agile concepts and practices including ITSM, Kanban, Lean and DevOps
- Scrum roles, artifacts, and events as it applies to both products and processes
- The two aspects of Agile Service Management:
 - 1 - Agile Process Improvement—ensuring processes are lean and deliver “just enough” control
 - 2- Agile Process Design—applying Agile practices to process design projects



Find an Education Partner Today
[DevOpsInstitute.com](https://www.DevOpsInstitute.com)