The Agile Performance Holarchy

Book Sampling
Build Great Agile—
The Agile Performance Holarchy provides leaders with organizational and behavioral guidelines for scalable and sustainable agility.
Table of Contents

THE AGILE PERFORMANCE HOLARCHY

Introduction 2
Background 3
An Operating System for Agile Organizations 5
The Agile Performance Holarchy 5
The Performance Circles 6
Holons 8
Objectives 8
Actions 8
Ceremonies and Techniques 8
Outcomes 8

THE PERFORMANCE CIRCLES

Leading 10
Holon: Valuing 11
Holon: Engaging 17
Holon: Visioning 22
Holon: Enabling 25

Providing 28
Holon: Partnering 29
Holon: Contributing 33
Holon: Equipping 36

Envisioning 40
Holon: Defining 41
Holon: Roadmapping 45
Holon: Clarifying 50

Crafting 54
Holon: Planning 55
Holon: Solving 60
Holon: Delivering 65
Affirming

Holon: Understanding
Holon: Confirming

Teaming

Holon: Organizing
Holon: Growing
Holon: Governing

CEREMONIES AND TECHNIQUES

Acceptance (Usability) Testing
Agile Agreement
Agile Digs
Agile Partner Assessment
All Hands Raised
Arc of Conversation
Automated Build
Backlog Grooming
Best Practices Board
Brainstorming
Burn-Down Chart
Confirmation
Continuous Deployment
Continuous Integration
Class, Responsibilities, Collaborators (CRC) Cards
Daily Stand-Up
Definition of Done
Definition of Ready
Dot Voting
Epics
Evaluation
Frequent Releases
Gemba Walks
Gemba Kaizen
Goal, Question, Metric (GQM)
Incremental Development
Kamishibai (Board and Cards)
Kanban Board
Kano Model
Mob Programming
Obeya Room
Open Space Technology
Pair Programming
Peer Reviews
Planning Poker
Product Backlogs
Product Scenarios
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project (Team) Chartering/Team Agreement</td>
<td>134</td>
</tr>
<tr>
<td>Prototyping</td>
<td>135</td>
</tr>
<tr>
<td>Release Planning</td>
<td>137</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>138</td>
</tr>
<tr>
<td>Review</td>
<td>139</td>
</tr>
<tr>
<td>Roles and Accountabilities Game</td>
<td>140</td>
</tr>
<tr>
<td>Scrum of Scrums</td>
<td>141</td>
</tr>
<tr>
<td>Scrum Wall</td>
<td>142</td>
</tr>
<tr>
<td>Self-Selection</td>
<td>143</td>
</tr>
<tr>
<td>Sprint</td>
<td>145</td>
</tr>
<tr>
<td>Sprint Demo (Also called Show and Tell)</td>
<td>146</td>
</tr>
<tr>
<td>Sprint Planning</td>
<td>147</td>
</tr>
<tr>
<td>Stakeholder Identification and Management</td>
<td>148</td>
</tr>
<tr>
<td>State of the Team</td>
<td>149</td>
</tr>
<tr>
<td>SWOT Analysis (Strengths, Weaknesses, Opportunities, Threats)</td>
<td>150</td>
</tr>
<tr>
<td>Team Agreement</td>
<td>151</td>
</tr>
<tr>
<td>Team Estimation Game (Relative Estimation, Planning Poker)</td>
<td>152</td>
</tr>
<tr>
<td>Team Room Set-Up</td>
<td>153</td>
</tr>
<tr>
<td>Technical Debt</td>
<td>154</td>
</tr>
<tr>
<td>Test Driven Development</td>
<td>155</td>
</tr>
<tr>
<td>Training</td>
<td>156</td>
</tr>
<tr>
<td>Unit Testing</td>
<td>157</td>
</tr>
<tr>
<td>Velocity</td>
<td>158</td>
</tr>
<tr>
<td>Visual Information Management</td>
<td>159</td>
</tr>
<tr>
<td><strong>NEXT STEPS FOR LEADERS</strong></td>
<td>160</td>
</tr>
<tr>
<td>How to apply the APH Model to your Business</td>
<td>161</td>
</tr>
<tr>
<td>Assessing your organization with the APH</td>
<td>162</td>
</tr>
<tr>
<td>Certification for your Company, Organization, or Agency</td>
<td>163</td>
</tr>
<tr>
<td><strong>APPENDICES</strong></td>
<td>164</td>
</tr>
<tr>
<td>Authors, Illustrators, and Contributors</td>
<td>165</td>
</tr>
<tr>
<td>Sources</td>
<td>166</td>
</tr>
</tbody>
</table>
Companies adopt agile frameworks to reduce the cost of failure.

It is conventional wisdom in the technology industry that failure is inevitable, with some research organizations estimating the rate of project failure at above seventy percent. There are many reasons for these results, not the least of which is an inability to accurately plan and envision solutions outside of a very limited time window, so it makes sense to seek solutions that assume failure, not success, and to simply seek to reduce its cost.

Failure is not just an option. It is normal and should be expected.

A foundational premise of agility is the acknowledgment that failure will almost always occur, and that it is advantageous to plan to fail fast and fail early, thereby reducing its cost while allowing teams to redirect toward a more successful approach through the use of experimentation, retrospectives, and short, time-boxed iterations. Quality professionals will recognize this as an application of Deming’s “Plan–Do–Check–Act” theory of continuous improvement applied in short iterations.

Agile methods deliver business value to end-users more quickly than traditional methods.

It has been said that agile frameworks result in faster delivery of solutions, although this point has been exaggerated and misunderstood by overly optimistic business managers. It is more accurate to say that “business value” is delivered more
quickly with an iterative and incremental delivery approach, where low value features are more quickly discarded thereby freeing up valuable resources to focus on the high priority needs of the customer. This results in both delighted customers, faster delivery of real solutions, and smaller, more focused product teams.

**Deployment of agile teams helps push decision-making down, freeing leaders to focus on strategy.**

For decades, the information technology industry has explored ways to push decision-making down in order to make business nimbler, and agile frameworks have finally provided a set of techniques that can make that a reality, if only leaders are willing to accept it. A healthy agile team requires minimal oversight, makes day-to-day operational decisions, collaborates with business customers, and delivers business value without the need for continuous management intervention, offering leaders an opportunity to focus on strategy, problem solving, and crafting a vision for both products and the business.

**Agile compliments IT industry models like CMMI, ISO 9001, and PMBOK.**

Many say that CMMI, ISO 9001, and PMBOK are models we use, but agile is something we are. There is truth to this. The former defines what needs to occur for a product or service to be successfully delivered, while the latter describes why we take those actions. In the case of agile, which is best defined as a set of values and guiding principles, the why modifies the what, resulting in behaviors and processes that are transparent, collaborative, and focused on delivering business value, as opposed to the documentation and process deliverables that are often perceived to be associated with the aforementioned models. The idea that why should always drive what, is essential to success with agile adoption, and has caused leaders to re-evaluate how process models like ISO 9001 are applied. In fact, according to the CMMI Institute, over seventy percent of organizations that have been “rated” at a CMMI Maturity Level report using one or more agile frameworks as part of their product development process.

**The world is changing, but we still have a long way to go.**

**Background**

In February 2001, a group of fourteen agile progenitors, representing a set of software development frameworks that included scrum, XP, Crystal, and more, gathered in Snowbird, Utah to collaborate, compare experiences, and explore new ways to advance the state of software development. Their motivation was driven by the over-processed, over-documented nature of the product development processes that was favored by corporate methodologies, and the industry standard process models that were preferred by the Department of Defense and Aerospace companies, including SW-CMM, PMBOK, AS-9100, and DO-178B.

The resulting “Agile Manifesto,” a collection of twelve guiding principles for improving the quality and experience of software product development, unleashed a wave that has washed over the software and information technology industry faster and more completely than any of its predecessors, leaving few traditional methods standing in its wake.

While the success of agile frameworks like scrum and XP cannot be understated, they have also been a victim of their own success.

**All is not well in the land of agility.**

Eager to replicate small company success, satisfy younger, more self-organizing employees, and to just simply “go agile,” large companies have jumped on the agile bandwagon, often ignoring the required changes to governance, infrastructure, measurement, and training required to succeed. The results have been chaotic, with large organizations adopting some elements of scrum (daily standups, sprints), and mixing them with more traditional roles and techniques that are in conflict with agile values. This conflict often negates the value of the agile ceremonies they are using, and leaves the organization without benefits they were hoping to achieve.

**Of two hundred large companies assessed by AgileCxO partners between 2010 and 2017:**

- Over 90% employed Project Managers for task management, oversight, and control of agile teams.
- Over half combined the Project Manager and Scrum Master roles.
Companies often fail to ensure that their leaders have the skills and capabilities to effectively lead and evaluate the transformation to high-trust, collaborative, and transparent behaviors that are essential to the success of agile adoption. In fact, the very behaviors that are required to “move up” in the large corporate environment today are antithetical to those needed to lead successful self-organizing agile teams.

This conflict results in an organizational type mismatch, whereby leaders desire agility, but continue to apply a low-trust Defined Process Control (DPC) model to run the business, when a high-trust, Empirical Process Control (EPC) model is required. This friction, often manifesting itself as “agile-like” or “scrum-but,” corrupts and degrades the very performance that agile leaders are seeking to achieve.

According to the Version One “State of Agile” survey, after fifteen years of agile expansion:

- 80% say they are “still maturing” with agile.
- 63% say corporate philosophy conflicts with core agile values.
- 60% have less than half of their teams actually practicing agile.
- 47% lack experience with agile methods and techniques.
- 51% do not have leadership skills needed for large-scale agility.
- 45% lack management support.

There is a missing layer in the architecture.

While the Agile Manifesto excels in describing why we do what we do, and industry frameworks and models describe what we need to accomplish, there is little guidance for leaders or teams on how we need to behave in order to experience consistent success with agile.

In order to succeed within large organizations, technology leaders and teams can categorize organizational agile capability into three inter-dependent levels: what, why and how.

“WHY-ABILITY” MODELS
The set of values and guiding principles that are traced directly to the goals and methods of the organization. These models tend to be philosophical in nature, and provide a set of guiding principles that we can use to anchor decisions about the methods and techniques we use to complete work. The Agile Manifesto, with its twelve guiding principles, is the most common example.

“WHAT-ABILITY” MODELS
The set of frameworks, methods, roles, and artifacts derived from industry-standard models or internal methodologies. These models define “what” needs to be done, and often provide examples that help us understand what we need to do while executing the software product development process. Examples include scrum, XP, and Kanban. Due to their empirical nature, these are purposefully more ambiguous than traditional methods, leaving room for teams to experiment, learn, and improve their own instantiation of the framework.

“How-ABILITY” MODELS
A set of behaviors, actions, and outcomes that help define and evaluate organizational success, and support the culture, goals, and objectives of the organization. How-ability models trace directly to established values, guiding principles, and frameworks to ensure that the behaviors exhibited by teams reflect the values of the organization. The Agile Performance Holarchy is a how-ability model that is an operating system for high performing agile organizations.
The Agile Performance Holarchy

The Agile Performance Holarchy (APH) is a how-ability model that provides agile leaders and teams with an operating system to build, evaluate, and sustain great agile habits and behaviors.

The APH is an organizational operating system based on a set of interdependent, self-organizing holons, where a holon is an entity that is simultaneously a whole and a part.

A holon is self-sufficient in its own right, but operates with limited control as part of a greater whole. Introduced in the 1967 book, *The Ghost in the Machine*, by Arthur Koestler, holons are described as self-reliant entities that “possess a degree of independence and can handle contingencies without asking higher authorities for instructions.”

Koestler also defines a holarchy as a “hierarchy of self-regulating holons that function first as autonomous wholes in supra-ordination to their parts, secondly as dependent part in sub-organization to controls on higher levels, and thirdly in coordination with their local environment.”
The use of a holarchy works well for describing and evaluating agile performance, where behaviors are self-organizing and empirical, rather than defined (EPC vs. DPC), and the sequence of actions and outcomes is unpredictable, iterative and recursive, rather than procedural.

The Agile Performance Holarchy is not a process, but an operating system comprised of interdependent actions and outcomes that manifest themselves as behaviors, ceremonies, and techniques performed by team members and organizational leaders. Sequence, rigor, and intensity of the actions and outcomes are determined by functional and project teams, not by the process.

The components of the APH include:

• Performance Circles
• Holons
• Objectives
• Actions
• Ceremonies/techniques

The Performance Circles
Each performance circle in the APH encapsulates a discrete set of behaviors, with a unique set of interrelated actions and outcomes that are essential to successfully adopting, transforming, and mastering large-scale agile performance. Each performance circle can be built, evaluated, and sustained independently, but all circles should be completed to demonstrate mastery.

Organizations wishing to benchmark performance against the APH may evaluate one or more performance circles to become certified as either Adopting, Transforming, or Mastering the behaviors of that circle.

THERE ARE SIX PERFORMANCE CIRCLES IN THE APH OPERATING SYSTEM:

As an agile leader, I will project agile values, provide the environment, and establish a vision, so that my teams can be agile and successful in everything they do.

As an agile leader, I want agile team members engaged in the planning and building of high quality products, so that we deliver the solution as expected.
As an agile leader, I want to foster a continuous improvement environment, and engage with agile partners, so that agile teams can grow their capabilities.

As an agile leader, I want teams and functional areas to learn and master self-organization and agile ceremonies and techniques, so that the entire organization can benefit fully from agile adoption.

As a product owner, I will establish a roadmap, release plan, and backlog, so the overall vision of the product/service can be realized.

As an agile leader, I want to confirm that teams are demonstrating agile values, methods and techniques as expected, so that I can understand what is working well and what needs improvement.
Holons

Holons are encapsulated within each performance circle, and represent a set of actions and outcomes that can effectively stand alone, but are also an integral part of a greater whole. All of the actions and outcomes should be implemented in order to realize the value of each holon. See Section Two for implementation details for each holon.

There are eighteen independent holons within the Agile Performance Holarchy:

<table>
<thead>
<tr>
<th>Leading</th>
<th>Crafting</th>
<th>Envisioning</th>
<th>Teaming</th>
<th>Affirming</th>
<th>Providing</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALUING</td>
<td>PLANNING</td>
<td>ROADMAPping</td>
<td>ORGANIZING</td>
<td>CONFIRMING</td>
<td>CONTRIBUTING</td>
</tr>
<tr>
<td>ENABLING</td>
<td>SOLVING</td>
<td>DEFINING</td>
<td>GROWING</td>
<td>UNDERSTANDING</td>
<td>PARTNERING</td>
</tr>
<tr>
<td>VISIONING</td>
<td>DELIVERING</td>
<td>CLARIFYING</td>
<td>GOVERNING</td>
<td></td>
<td>EQUIPPING</td>
</tr>
<tr>
<td>ENCAGING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Objectives

Each holon describes one or more objectives that must be met in order to instantiate the value of the holon. An objective can be met by taking one or more of the defined actions in a manner that is consistent with agile values.

Actions

An action is the specific behavior that is applied to meet an objective. All behaviors must be demonstrated to meet the intent of the objective, and must be aligned with agile values. The APH recommends one or more agile ceremonies and techniques that, when executed successfully, will meet the intent of the actions.

Ceremonies and Techniques

Each action provides a recommended set of ceremonies and techniques that can be adopted to demonstrate the desired behavior and meet the intent of the action. Section Three provides a detailed description of agile ceremonies and techniques that can be used as a guideline.

Outcomes

Each holon contains a set of outcomes that can be used to evaluate, improve, and sustain high performance. The outcomes are categorized into three levels: Adopting, Transforming, and Mastering, and are used during the APH certification process to evaluate and describe an organization’s state of agile performance.
The Performance Circles

THE FOLLOWING IS A SAMPLE FOR THE PERFORMANCE CIRCLE SECTION
"As an agile leader, I want agile team members engaged in the planning and building of high quality products, so that we deliver the solution as expected."

The Crafting performance circle describes objectives, actions and ceremonies that address the capability lift and craftsmanship required to consistently deliver high quality products and services.

Agile leaders need to instill a culture of craftsmanship in their own organization as well as other organizations within the product or service value stream. Customers, procurement, and sales all have a role to play, and for technologists to be successful, leaders from those organizations need to demonstrate craftsmanship as well.

This includes:

- Integrated coding-design-testing techniques such as Test-Driven Development (TDD) and Business-Driven Development (BDD).
- Effective strategies for managing technical debt and refactoring.
- Eliciting and recognizing high quality requirements, epics, and stories.
- Engagement of the business and end-users for product ownership and relevant ceremonies.
- Implementation of an effective tool chain with sufficient automation across the entire product or service development lifecycle.
Holon: Planning

Planning is a holon within the Crafting performance circle. The Planning holon contains the actions and ceremonies required to estimate and plan for the upcoming sprint or iteration, grooming the backlog by the team, demonstrating successes, and inspecting and adapting team performance as part of continuous learning and improvement.

OBJECTIVE:
As an agile leader, I want agile team members to estimate and plan for the upcoming sprint and groom the backlog mid-sprint, so that we meet the sprint forecast as planned.

PERFORMANCE LEVEL OUTCOMES:
The Planning holon has the following outcomes defined at each performance level. An organization can achieve performance outcomes by performing the actions and behaviors associated with the specific performance levels.

<table>
<thead>
<tr>
<th>Adopting Level Outcomes</th>
<th>Transforming Level Outcomes</th>
<th>Mastering Level Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Projects demonstrate the use of established agile planning ceremonies and techniques.</td>
<td>✓ Essential planning stakeholders engage with projects while demonstrating agile values.</td>
<td>✓ Projects select planning ceremonies and techniques based on project needs and objectives.</td>
</tr>
<tr>
<td>✓ Agile values are demonstrated during planning.</td>
<td>✓ Agile leaders are trained in agile planning ceremonies and techniques, and use them for their own work.</td>
<td>✓ Agile leaders engage with projects using agile values.</td>
</tr>
<tr>
<td>✓ Project team members are trained in the agile planning ceremonies and techniques.</td>
<td>✓ All projects and functional groups use agile planning ceremonies and techniques.</td>
<td>✓ Planning ceremonies and techniques are improved and expanded over time.</td>
</tr>
</tbody>
</table>
**ACTION 1.0**

**Enable Team Commitment**

A culture of agile leadership demonstrates core agile values top-down and bottom-up at all times to enable self-organization of all project teams. Project chartering and Team Agreements define the agile brand of each team.

**CEREMONIES // TECHNIQUES**

**PROJECT CHARTERING/TEAM AGREEMENT**

Use project chartering/team agreements to enable agile teams to self-organize around the key characteristics of their team’s values, goals, and objectives. Each team creates their personal “agile brand,” which can include behaviors, operations, and a unique team name.

**SELF-SUBSCRIPTION**

Foster a culture of team commitment and self-organization by ensuring all agile team members are empowered to self-subscribe (pull) work from a backlog based on their availability, expertise, and workload. Self-subscription is a core behavior of lean thinking, and is a powerful tool for promoting agility across teams.
ACTION 2.0

Create and Agree on a Definition of Done

Backlog grooming mid-sprint can be used to ensure that a surplus of user stories are always available for the next iteration or sprint. A Definition of Done is clearly defined for each story, with common criteria are captured in Team Agreements as part of the teams’ agile brand.

CEREMONIES // TECHNIQUES

PROJECT CHARTERING/TEAM AGREEMENTS

Use project chartering/team agreement to self-organize around the key characteristics of their team, and the Definition of Done for Epics, stories, code, tests, and more. Let them create their “agile brand.”

TEAM AGREEMENTS

Use Team Agreements to capture common Definitions of Done.

DEFINITION OF DONE

Use Definition of Done to capture, in unambiguous language, how the team knows a user story is complete and ready to present to the Product Owner or other stakeholders in a sprint demo.

BACKLOG GROOMING

Use Backlog Grooming to generate and capture the forecast for subsequent sprints, and define the Definitions of Done for individual user stories that are being considered.
SELF-SUBSCRIPTION

Foster a culture of ownership by empowering agile teams to employ Self-Subscription while pulling their own work from the backlog. Self-subscription supports ownership and commitment with self-organizing teams. Employ Self-Subscription in all Sprint planning and grooming meetings.

DEFINITION OF READY

Use a common definition of ready to increase the likelihood that the team is working with high quality stories, and that the implementation of a user story in each increment will be successful. Applying screening criteria such as INVEST to user stories is an example of Definition of Ready.

INCREMENTAL DEVELOPMENT

Use an incremental approach to break development into small, consistent, manageable, and predictable pieces.

CEREMONIES // TECHNIQUES

ACTION 3.0

Select Work to Complete for Each Sprint

Self-subscription enables the agile team members who are doing the work to pull from a backlog(s) of prioritized user stories. A Definition of Ready is clearly defined and verified for each story on the backlog to ensure they are ready to be selected.
ACTION 4.0

*Estimate the Work to Complete in the Sprint*

Sprint Planning meetings are used to size user stories relative to one-another using techniques such as Planning Poker, T-Shirt Sizing, or the Team Estimating Game. As part of planning the upcoming sprint or iteration, team members break the selected user stories into tasks, and then estimates each task in hours to complete. Tasks include activities beyond coding proven to support craftsmanship such as design, unit testing, prototyping, and validation/verification.

**CEREMONIES // TECHNIQUES**

**RELATIVE ESTIMATION**

Use relative estimation to size epics and user stories as larger or smaller than a known, agreed-up-on reference. Limit choices and ranges in the Team Agreement to increase estimation consistency, and reduce the time needed to create estimates.

**TEAM ESTIMATION GAME (PLANNING POKER)**

Use the Team Estimation Game or Planning Poker, to size user stories as part of Sprint Planning. Encourage “relatively right” versus “accurately wrong” estimates using Planning Poker based on a limited-range Fibonacci sequence.

**BACKLOG GROOMING**

Use Backlog Grooming to prioritize stories for future sprints.

**SPRINT PLANNING**

Include the Sprint planning meetings to engage agile team members and the product owner in the estimation and planning process. Team member should break down user stories into the tasks (in hours) needed to complete the story and meet the predefined definition of done.
Holon: Solving

Solving is a holon within the Crafting performance circle. The Solving holon contains the actions and ceremonies needed to create and sustain high-quality products and services from the viewpoint of the customer. Solving is a structured and disciplined approach to the design and development of products and services.

OBJECTIVE:
*As an agile leader, I want to help agile team members meet their sprint forecast, so that we develop a high-quality solution using an iterative and incremental approach.*

PERFORMANCE LEVEL OUTCOMES:
The Solving holon has the following outcomes defined at each performance level. An organization can achieve performance outcomes by performing the actions and behaviors associated with the specific performance levels.

<table>
<thead>
<tr>
<th>Adopting Level Outcomes</th>
<th>Transforming Level Outcomes</th>
<th>Mastering Level Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ Teams use established product development ceremonies and techniques.</td>
<td>✗ Essential product development stakeholders demonstrate agile values.</td>
<td>✗ Teams proactively select product development ceremonies and techniques based on project needs, objectives, and constraints.</td>
</tr>
<tr>
<td>✗ Agile values are demonstrated during product development.</td>
<td>✗ Agile leaders are trained in how to recognize established product development ceremonies and techniques.</td>
<td>✗ Agile leaders engage with teams using agile values.</td>
</tr>
<tr>
<td>✗ Team members are trained in the established product development ceremonies and techniques.</td>
<td>✗ All teams in the organization select product development ceremonies and techniques from a common set.</td>
<td>✗ Product development ceremonies and techniques are improved and expanded over time.</td>
</tr>
</tbody>
</table>
ACTION 1.0

Burn Down User Stories

Project teams demonstrate self-organization and commitment by completing the implementation of user story tasks for the current sprint or iteration without oversight from a project manager. Disciplined methods are used to complete work, with a focus on quality, and Visual Information Management Systems create transparency for all team members and extended stakeholders.

CEREMONIES // TECHNIQUES

**BURNDOWN CHART**
Display daily progress and work that has been completed during each sprint using a Burndown Chart. Any stakeholder can easily determine if the sprint forecast will be met. Action can be taken early to reprioritize stories and tasks.

**SCRUM WALL**
Use a Scrum Wall to show the progression of user stories through the workflow states defined by the team in the team agreement. The Scrum Wall and Burndown Charts, used together, create transparency for all stakeholders.

**KANBAN BOARD**
For teams that use Kanban to manage work, use a Kanban Board for visual information management.

**VELOCITY**
Velocity, in story points, represents the capacity of a team to deliver value during a sprint. Once established, it creates predictability for each sprint and the overall release plan, given a relatively consistent team.

**PAIR PROGRAMMING**
Use Pair Programming for low-defect product development. Although counter-intuitive, it is consistently more efficient than working individually.

**MOB PROGRAMMING**
Mob Programming adds the whole team, not just the developers, to the development effort. Adding different perspectives early reduces all sources of defects.

**TEST DRIVEN DEVELOPMENT**
Use Test Driven Development to create tests first, and then write code that fails or passes the tests. Working this way increases efficiency by getting both the tests and the code right early in the development cycle.

**UNIT TESTING**
Unit Testing finds and eliminates defects early in the development cycle. Automated unit testing is common with agile teams, and is encouraged for one hundred percent of all code.
**ACTION 2.0**

**Review Activities and Impediments**

The team discusses progress and transparently identifies impediments as a group. The daily standups have a short, focused script, defined in the team agreement, are time boxed, and are facilitated by a Scrum Master or equivalent. Impediments are recorded in an impediment backlog so they can be addressed outside of the daily standup.

**CEREMONIES // TECHNIQUES**

**DAILY STAND-UP/DAILY SCRUM**

The Daily Stand-Up, also known as the Daily Standup, Huddle, or Daily Meeting, significantly increases transparency and collaboration for the team. The better and more open the daily stand-up is, the higher the likelihood of rapidly solving problems to ensure meeting sprint commitments.

**IMPEDIMENT BACKLOG**

Use an Impediment Backlog to quickly record impediments that cannot be addressed during the daily stand-up. Use Visual Information Management for the Impediment backlog to increase visibility and transparency.
**ACTION 3.0**

*Remove Impediments*

Impediments in the impediment backlog are eliminated, if possible, by the Scrum Master or agile leader. Impediments that cannot be eliminated are used to identify future improvements, or are escalated to the agile leadership so they can be resolved for the benefit of entire organization.

**CEREMONIES // TECHNIQUES**

**IMPEDEMENT BACKLOG**

Continuously improve quality and performance by eliminating impediments recorded in the Impediment Backlog. Instill the value of “Inspect and Adapt.”

---

**ACTION 4.0**

*Review Completed Work*

The customer or end-user of the product or service reviews and accepts or rejects what was completed by the team during the current sprint or iteration. The customer or end-user accepts the work, identifies changes to be made, or identifies new needs or stories for a future sprint or iteration. Prioritization of changes or new stories is defined and agreed upon.

**CEREMONIES // TECHNIQUES**

**SPRINT DEMO**

Use the Sprint Demo, also known as a Sprint Review, or Show and Tell, to capture immediate feedback from the Customer or Business Owner of the product being demonstrated. Capture both defects and change requests at this time to efficiently manage changes and improvements.
ACTION 5.0

Identify Improvements

The project team conducts a Retrospective after each sprint to identify improvements and learning opportunities. The Retrospective is facilitated by a Scrum Master, or equivalent, and appropriate learning information is shared beyond the boundaries of the project team so the entire organization benefits.

CEREMONIES // TECHNIQUES

**SPRINT RETROSPECTIVE**

Ensure the team conducts Sprint Retrospectives to identify what went well (continue doing), what did not go well (stop doing), and what should be improved (learning) for the next sprint. Share appropriate learning with all teams doing similar work.

**MILESTONE RETROSPECTIVE**

Do Milestone Retrospectives to collaborate with additional stakeholders outside the localized agile team.
Holon: Delivering

Delivering is a holon within the Crafting performance circle. The Delivering holon contains the actions and ceremonies needed to plan product or service releases at a higher level than individual sprints or iterations. Delivering contains planning how product assembly and testing are done at the sprint or iteration level, and at the release level, as well as how and when the assembled and tested product is made available to the customer or end-user.

OBJECTIVE:
As an agile leader, I want team members to integrate and test the solution, so that it meets the needs of the customer and end-user.

PERFORMANCE LEVEL OUTCOMES:
The Delivering holon has the following performance outcomes defined at each performance level. An organization can achieve performance outcomes by performing the actions and behaviors associated with the specific performance levels.

<table>
<thead>
<tr>
<th>Adopting Level Outcomes</th>
<th>Transforming Level Outcomes</th>
<th>Mastering Level Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>✴ Projects use defined delivery ceremonies and techniques.</td>
<td>✴ Essential delivery stakeholders engage with projects using agile values.</td>
<td>✴ Projects proactively select organizational delivery ceremonies and techniques based on project needs, objectives, and constraints.</td>
</tr>
<tr>
<td>✴ Agile values are demonstrated during delivery.</td>
<td>✴ Agile leaders are trained in the delivery ceremonies and techniques.</td>
<td>✴ Agile leaders engage with projects using Agile values.</td>
</tr>
<tr>
<td>✴ Project team members are trained in the defined delivery ceremonies and techniques.</td>
<td>✴ All projects use defined delivery ceremonies and techniques.</td>
<td>✴ Delivery ceremonies and techniques are improved and expanded over time.</td>
</tr>
</tbody>
</table>
**ACTION 1.0**  
*Plan each Delivery*

The Product Owner, or their equivalent, creates a release plan that connects the completion of the stories in each sprint or iteration to individual releases. The release plan is at a level higher than the individual sprints or iterations, and is continuously updated and communicated to the team as part of Backlog Grooming.

**CEREMONIES // TECHNIQUES**

**RELEASE PLANNING**  
Identify the high-level release schedule and the goal for each release by doing Release Planning. Work to deliver the highest value (from the perspective of the end user) user stories early in the development.

**BACKLOG GROOMING**  
Use Backlog Grooming to target user stories for planned releases.

**ACTION 2.0**  
*Assemble Product or Service Component for the Sprint*

The assembly of the product or service components are accomplished using the methods and tools defined as part of release planning. Automated Builds and Continuous Integration are used to improve efficiency, quality, and the craftsmanship of the end product or service.

**CEREMONIES // TECHNIQUES**

**AUTOMATED BUILD**  
Use Automated Builds to build code modules, and run automated unit tests each time code is checked-in. Bias the process towards rapidly finding and fixing defects early (fail fast).

**CONTINUOUS INTEGRATION**  
Use Continuous Integration to build the product at the component level, and run automated integration tests each time the product is built. Bias the process towards rapidly finding and fixing defects at the interface level.
**ACTION 3.0**

**Test Assembled Product or Service**

The assembled product or service is tested to make sure performance matches the expectations and commitments made during the sprint or iteration. Detailed information is captured for defects, enabling root cause analysis, and resolution occurs within the sprint or iteration, or de-prioritized and added to the backlog. The team analyzes defects or bugs, and uses the information gathered for future Retrospectives to drive continuous learning and improvement.

**CEREMONIES // TECHNIQUES**

**ACCEPTANCE TESTING**

Conduct Acceptance Testing to ensure that product works correctly for the end user. If possible, the customer or end-user should perform testing in the presence of the team via the Sprint Demo.

**USABILITY TESTING**

Leverage Usability Testing to ensure that the end-user will have a positive impression of the user experience.

**ACTION 4.0**

**Deliver the Assembled Product or Service**

The product or service is delivered to the customer or end-user as defined in the release plan. Changes or defects identified after delivery are placed on the backlog, reviewed with the customer or end-user, prioritized, and sequenced into future sprints or iterations.

**CEREMONIES // TECHNIQUES**

**CONTINUOUS DEPLOYMENT**

Using automation, promote the assembled and tested product to the production environment at the end of each sprint to receive rapid feedback from the user community.

**FREQUENT RELEASES**

When automated build is not possible, promote the assembled and tested product in the production environment at a rapid cadence to receive frequent feedback from the user community.
Ceremonies & Techniques

THE FOLLOWING IS A SAMPLE FOR THE CEREMONIES & TECHNIQUES SECTION
Backlog grooming (sometimes called “story-time”) is a common agile technique used by teams to prioritize the backlog of epics and/or user stories before and during a sprint. Backlog grooming typically includes a negotiation between the product owner and the agile team on which will be added, removed, or revised for the upcoming sprint or iteration. The product owner and team members have input into this collaborative decision, which is a critical activity related to the planning and execution of a sprint. The product owner has accountability for the product backlog, while the team provides an important support role based on their knowledge of the product or service being developed. New epics and user stories may emerge as a result of backlog grooming, and it is the responsibility of the Product Owner to capture these within the backlog along with associated Definition of Ready (acceptance criteria).

Additional Backlogs, other than the Product Backlog, may be groomed to prioritize other work related to continuous improvement, or non-product related activities. In these cases, the stakeholders and frequency of grooming may vary since these may not be tied to a specific project. Such as:

- Impediment Backlog – Prioritized list of impediments across the enterprise.
- Product Training Backlog – Prioritized list of training required for the product or organization.
- Improvement Backlog – Prioritized list of improvements for the organization.

**DESIRED BEHAVIORS**

The steps to perform backlog grooming include:

1. Creating the initial product backlog.
2. Sequencing the user stories based on business value and priority with the highest priority user story at the top and the lowest priority story at the bottom.
3. Reviewing the stories (product owner and agile team) using a defined set of criteria.
4. Identifying user stories that do not meet the criteria and update them accordingly.
5. Establishing or updating traceability between epics, user stories, and child user stories.
6. If Backlog Grooming is occurring mid-sprint, identify the stories that are most likely to be included in the upcoming sprint/iteration.
Daily Stand-Up

The Daily Standup Meeting (aka “Daily Scrum” or “Daily Meeting”) is an agile technique that is often identified with Scrum, but is popular with all types of agile approaches. It is used as a way to identify issues and risks earlier than a traditional project (“fail fast”), and to increase collaboration between agile team members.

There are two methods of running a daily standup. The default method is to conduct a standup where information is shared based on user stories. The other method is called “round-robin” and is used for discussion related to tasks that are not user story bound like defect management, continuous improvements, and so on. The benefit of a user story focused discussion is a shift in focus from the individual to the story, as opposed to turning the daily standup into a status report without benefits.

Scrum identifies attendees as the “Scrum team” or core team members, and does not recommend that customer or management representatives be involved. In practice, extended team members are often in attendance. It is typical that each person who contributes toward completion of the current sprint participates in the meeting. Agile leaders should use caution when opening daily standups to external stakeholders, as conflicts in culture and desired outcomes could introduce chaos into the system.

**DESIRED BEHAVIORS**

Daily stand-up practices include:

1. Hold the daily standup at the same time each day.

2. Team meets face-to-face, or remotely using appropriate technology to promote information sharing, transparency, and peer accountability within the team.

3. The meeting is facilitated by a Scrum Master or other team member filling the facilitator role.
4. Typically, 15 minutes in length, the Daily Standup features input from each team member that focuses on three key areas:
   a. What has been completed since the last Daily Standup.
   b. What the team member plans on completing today.
   c. Any impediments, risks, or issues being experienced by the team member.

5. Teams may use some kind of token to indicate the current person is allowed to speak. This ensures that only one person speaks at a time and is fully heard by the team without interruption.

6. The Scrum Master or facilitator should help manage the conversation flow to avoid interruptions and conclude each stand-up by asking for announcements, requests or comments.

7. Scrum Master records impediments, risks, or issues to be resolved and the team collaborates on how best to resolve the issues identified by the team members.
Appendices
Authors, Illustrators, and Contributors

AgileCxO would like to thank all of the companies, agencies, and Partners that provided research and data to make this publication possible.

PRINCIPLE AUTHOR
Jeffrey R. Dalton

CONTRIBUTING AUTHORS
Michelle Rauch
Ross Timmerman
Darian Poinsetta
Timothy Zeller
Julie Calfin
Laura Adkins
Cathy Henderson

ILLUSTRATOR
Sam Melroy

MANAGING EDITOR
Michelle Rauch

COPY EDITOR
Patricia S. Dalton

CONTRIBUTING PARTNER
Broadsword Solutions Corporation
Sources

“A Guide to Scrum and CMMI: Improving Agile Performance with CMMI.”
CMMI Institute and Broadsword Solutions Corporation © 2016

www.agilealliance.org
“Agile Estimation Techniques,” January 18, 2018

www.mountaingoatsoftware.com

www.thebalance.com

www.projectmanagementdocs.com

www.openspaceworld.org