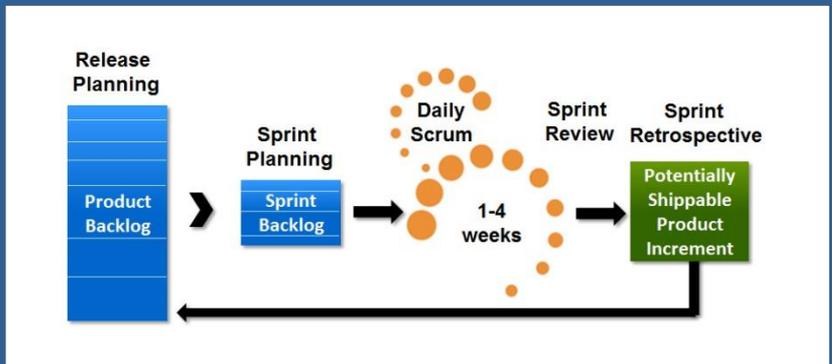


Scrum Companion

A Short Overview of the Scrum
Product Development Framework



by

Roger W. Brown

Scrum Alliance® Certified Scrum Trainer
and Certified Enterprise Coach

[www. AgileCrossing.com](http://www.AgileCrossing.com)

© 2016-2018 Roger W. Brown

Edition 2.0.1 Oct. 19, 2018



This work is licensed under CC BY SA 4.0. To view a copy of this
license, visit <http://creativecommons.org/licenses/by sa/4.0/>

Why This Book?

This book is a concise description of the Scrum Product Development Framework written by a Scrum Alliance® Certified Enterprise Coach and Certified Scrum Trainer. It is a great introduction to the subject as well as a reference for training and the ongoing application of Scrum. It was developed as a reference for participants of Roger Brown's and Jeff McKenna's Scrum Alliance® Certified ScrumMaster® and Certified Scrum Product Owner® Courses.

About the Author

Author: Roger Brown, Scrum Alliance® Certified Enterprise Coach®, Certified Scrum Trainer®

Contributions by Jeff McKenna, Scrum Alliance® Certified Enterprise Coach®, Certified Scrum Trainer® and Coach for the first Scrum Team

Roger Brown is an independent Agile Coach in San Diego who provides training, consulting and coaching services in Scrum and enterprise Agile adoption. He is a veteran software engineer with experience at the US Veterans' Administration, Dartmouth College, Vicinity Corporation, Microsoft and TeleAtlas. His Agile coaching and training clients include over 70 companies from small startups to multinational banks.

Roger is a Scrum Alliance® Certified Enterprise Coach since 2007 and Scrum Trainer since 2008. He is a reviewer and founding member of the Certified Scrum Coach, Certified Enterprise Coach and Certified Team Coach Programs at the Scrum Alliance. He has facilitated open coaching clinics at several Agile conferences since 2009. Roger is currently working in the areas of strategic planning and executive coaching. Roger has presented topics and led workshops at Scrum Alliance Gatherings, Agile Alliance Conferences, Agile Open California in 3 locations and San Diego Agile Meetups.

Terms and concepts in this work are consistent with Scrum Guide™ 2017
www.scrumguides.org

Contents

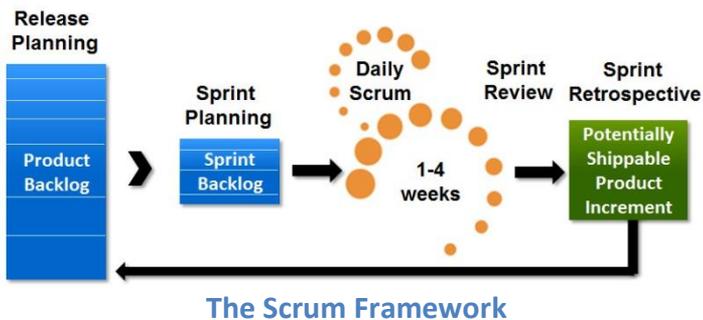
Why This Book.....	i
1. Introduction.....	1
2. Origins	3
3. Scrum Roles	4
<i>Product Owner</i>	5
<i>Development Team</i>	6
<i>Scrum Master</i>	7
4. Scrum Preparation	8
5. The Sprint	9
6. Scrum Meetings	10
<i>Sprint Planning Meeting</i>	10
<i>Daily Scrum Meeting</i>	11
<i>Sprint Review Meeting</i>	12
<i>Sprint Retrospective Meeting</i>	13
7. Scrum Artifacts	15
<i>Product Backlog</i>	15
<i>Sprint Backlog</i>	16
<i>Potentially Shippable Product Increment</i>	18
8. Product Backlog Refinement	19
9. Team Agreements.....	20
10. Scrum Planning	22
11. Agile Engineering Practices.....	24
12. Scrum Benefits	25
<i>Organizational Benefits</i>	25
<i>Customer Benefits</i>	26
<i>Employee Benefits</i>	26
13. Agile Values and Principles.....	27
14. Success Factors	29
15. References	29
Index	30

1. Introduction

Scrum is a framework for product development by self-organized teams. Scrum uses an iterative structure, defining short blocks of time called Sprints, each focused on the next most valuable features of the product. Each Sprint ends with a set of tested, shippable features. Scrum is typically associated with software development but has been applied to many other types of products. Scrum emphasizes adaptability to change, implementing principles defined for a class of frameworks called Agile Software Development.

Scrum is not a methodology as it does not define how work is to be done. Rather Scrum defines a container and mechanism for efficient, incremental product development. Using the Scrum Framework, each product team will implement and refine their own process.

Scrum is simple to understand but difficult to implement in organizations used to a more rigorous and rigidly defined development method. Partial implementation of Scrum is unlikely to succeed because its design is based on a synergistic set of values and principles that complement each other in a way that provides adaptability to change, quick feedback and an enjoyable, creative working environment.



Scrum has three roles: the Product Owner who is responsible for building the most valuable product; the Scrum Master who guides the team in Scrum and facilitates the team process; and the self-organized Development Team that builds the product.

Scrum has four standard meetings: Sprint Planning, Daily Scrum, Sprint Retrospective and Sprint Review.

Scrum has three artifacts: the Product Backlog, the Sprint Backlog, and the Potentially Shippable Product Increment.

2. Origins

Jeff Sutherland and his team created the Scrum framework in 1993. The term “scrum” comes from a Rugby analogy used in a 1986 paper by Takeuchi and Nonaka, published in the *Harvard Business Review*. In that study, the authors described product development methods in use at five hardware companies that differed from industry standard practice. Major differences included the formation of stable, cross-functional, self-managed teams and overlapping development phases to enable quick application of new information. Scrum was first publicized by team member Ken Schwaber in a paper at OOPSLA 1995.

Scrum implements Empirical Process Control, a model for constantly improving process based on observations of what is working well. This approach is analogous to the Scientific Method in which experiments are conducted to learn new information that can then be used to improve future outcomes.

Empirical process is contrasted to the “defined process” model traditionally applied to software development. The traditional methodology, popularly known as “Waterfall” uses the same process for every project regardless of outcome. It follows a construction metaphor, keeping to a defined, phased plan that is difficult to change. Scrum provides agility to achieve more successful business outcomes through feedback and adaptation of process for each circumstance.

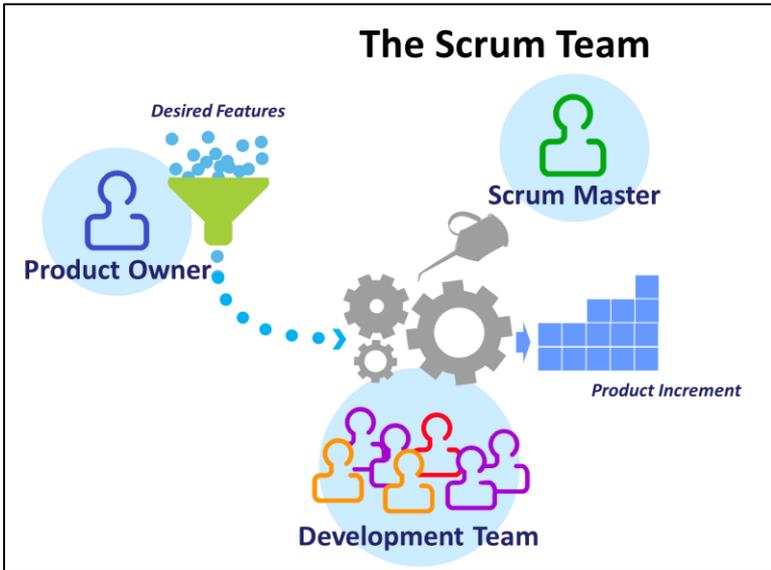
Scrum leverages three “pillars” of empiricism:

- Transparency: having all relevant data visible for review
- Inspection: analyzing the available data to identify opportunities for improvement
- Adaptation: applying new knowledge to improve current process

In Scrum, empirical process control is applied to the process in use as well as to the product created.

3. Scrum Roles

Scrum has three roles: Product Owner, Scrum Master, and Development Team. Together, these three roles form the **Scrum Team**.



All Scrum Team members are dedicated to a single team, typically for a long time (1 year or more) to achieve a state of high performance. Membership is on a full-time basis for the duration. Any variance from this design will result in inefficiencies, bottlenecks and time lost due to context switching and knowledge dilution. Roles are not to be shared by a single team member because each role requires full-time focus for the Scrum Team to achieve optimal results.

Product Owner

The **Product Owner** role has the following characteristics:

Basic Requirements for Success

- Knowledge of the Product business domain
- Ability to make business decisions with confidence
- Fulfilled by a single person to avoid the delay and conflicts of “decision by committee”

Responsibilities

- Consults with business stakeholders to determine goals and priorities
- Strives to produce the most valuable product possible within the given timeframe, measured by Return on Investment (ROI)
- Determines which features will be released and when, given available budget and team

Activities

- Prioritizes features by overall business value
- Manages the Product Backlog by continually re-prioritizing as needed
- Elaborates the most important items on the Product Backlog for consideration at the Sprint Planning meeting
- Reviews and then accepts or rejects each item as the Team completes it

Authority

- Independent authority to make necessary priority decisions
- Has final say what is included in the released product, when to release and when to stop developing

Development Team

The **Development Team**, also known as “The Team” is responsible for building the product with high quality.

The Team has the following characteristics:

Basic Makeup

- Includes all of the skills required to build the product
- Has three to nine members, full time dedicated to this one team
- Self-organizes for optimal efficiency

Responsibilities

- Is responsible for producing the Potential Shippable Product Increment that is complete to the Team’s Definition of Done
- Strives to continually improve its development practices and work efficiency through empirical process control
- Defines and adheres to a Definition of Done for the product increments to achieve smooth flow and the high product quality
- Shares knowledge of domain, technology and skills

Activities

- Decides the Sprint Goal in collaboration with the Product Owner
- Provides estimates of work required per Product Backlog Item if estimation is a Team practice
- Collaborates to achieve a smooth flow of quality production during each Sprint
- Demonstrates the product increment to the Product Owner and Stakeholders
- Reflects on process for continuous improvement

Authority

- Authorized to defer external work requests that are not in the current Sprint Plan in order to maintain momentum and flow
- Is empowered to make decisions on how work is done, who does it and how the process will evolve
- Achieves stability by staying together for a year or more to achieve high productivity

Scrum Master

The **Scrum Master** or ScrumMaster is a guide, coach and facilitator for the Product Owner and Development Team.

Responsibilities

- Fosters Scrum values and Agile principles
- Observes the team and its process to identify opportunities for improvement
- Mediate conflicts within the Team
- Helps the Team define and excel at their chosen process for getting work done
- Enables and promotes innovative ideas
- Protects the Team from interruptions

Activities

- Builds relationships between the Team and other Teams and operational collaborators
- Identifies and promotes opportunities for improved efficiency, waste reduction and team harmony
- Facilitates collaboration between Team Members
- Manages impediments or blockers to Team work flow
- Helps the Team stay focused by protecting them from external interference
- Facilitates meetings and tracks Sprint progress

Authority

- Has no intrinsic authority but earns trust through ongoing Servant Leadership
- Serves as a Change Agent to help shift the organization to a more agile business culture

Requisite Skills

- Facilitation
- Communication
- Mediation
- Negotiation
- Coaching

4. Scrum Preparation

A new product or release of an existing product begins with the “Product Vision”. The Product Owner is responsible for creating the Vision based on prior research into market or customer base needs or desires.



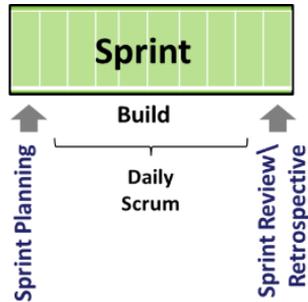
The Product Owner represents the desires of the end-customer and/or business stakeholders for the product to be created or enhanced. The Product Vision describes in simple terms its market justification, its ultimate purpose, who it is for, how it will create value and what benefits it will provide. The vision may encompass several Releases or a single release to the market in a Product Roadmap.

The members of the Development Team are identified and, together with the Product Owner and Scrum Master, define the Scrum Team. Often, a set of activities called “chartering” are then undertaken to identify shared team goals and agreements. A collaborative refinement of the Product Vision may be included.

With help from the Development Team, Scrum Master and other Stakeholders, the Product Vision is expressed in more detail in a Product Backlog, a list of features prioritized primarily by value delivered to the customer. Creation of the initial Product Backlog may happen during an event called Release Planning. The Product Backlog is then revised continuously as priorities change during development. Items on the Product Backlog (Product Backlog Items or PBIs) are typically estimated by size and complexity during Release Planning to aid both long term and Sprint planning.

5. The Sprint

The Sprint is the time-box that contains the Scrum meetings and development work. A time-box is a fixed length of time allocated to accomplishing a specific goal. The Sprint is over when the time is up regardless of how much product is developed. Meetings are time-boxed as an upper-limit but may conclude sooner.



The bulk of the Team's time in a Sprint is spent building and validating the product. For example, in a 2 week Sprint there are about 8 hours of planning and review concentrated in the first and last days with nine days of product development in between. There are no breaks between Sprints except, perhaps, after a formal release of the product.

Sprint length is determined prior to the first Sprint and kept constant for the duration of the product development. Sprints are typically either 1, 2, 3 or 4 weeks long in calendar days, all starting on the same day of the week. A Sprint containing holidays would have fewer days available for work.

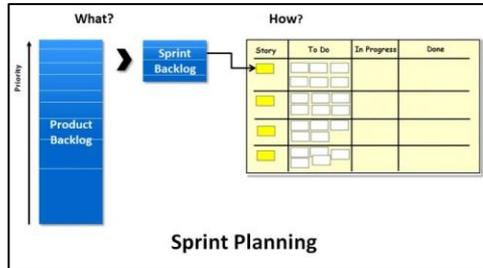
Note that the Sprint length does not vary. It is fixed by the Team and changed only in unusual circumstances. A fixed length allows the Team to develop a steady rhythm, find a sustainable pace for completing a reliable amount of product and to measure how much work can be accomplished in a normal Sprint.

6. Scrum Meetings

Scrum has four standard meetings, also called ceremonies or events: Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective.

Sprint Planning Meeting

The first Sprint begins when enough of the Product Backlog is defined and prioritized to give the Team sufficient work to do in the chosen time period. The first and all subsequent Sprints start with the Sprint



Planning Meeting in which the Team develops a plan of what they will work on and how they will accomplish it. The Scrum Team may define a Sprint Goal, a thematic description of the work to be done.

The entire Scrum Team is present for Sprint Planning. Other stakeholders and experts may be present to help answer questions about the context of certain PBIs and to share useful information for the detailed Sprint plan.

The first topic of this meeting is the Sprint forecast, how much work the Sprint Team can successfully complete in the Sprint. The forecast is based on the Team members' availability or "team capacity" and its "velocity", a record of the amount of product developed in previous Sprints. Sprint Planning is a "pull" process in which the Team plans for as much work as it can accomplish with high quality and at a sustainable pace. This forecast is then used to choose the best high-priority items on the Product Backlog to work on during the Sprint.

Once the Scrum Team has selected and committed to give their best effort to deliver the set of chosen PBIs, the Team proceeds to the second topic of the meeting - a plan for implementing those items. There is no set format for such a plan, but a common form is to break the features down into Sprint tasks. Tasks are the specific development activities

required to implement a Product Backlog Item. The resulting plan, called the Sprint Backlog, contains the PBIs to be implemented and the tasks they will require. The Sprint Forecast may be revised upwards or downwards following this more detailed planning activity. Sprint progress is then tracked in comparison to the plan as tasks and features are completed.

A typical meeting time-box for new teams is 2 hours per week of Sprint (eg. 4 hours for a 2-week Sprint). Experienced teams will complete the meeting in half of this time.

Daily Scrum Meeting

Work on the next product increment begins as soon as planning is done. Each day the Team holds the Daily Scrum, a fifteen-minute meeting to synch the team up on the current state of work.



A common format for this meeting is to have each team member answer these three questions:

- What did I do yesterday to achieve our Sprint Goal?
- What do I plan to do today to achieve our Sprint Goal?
- What impediments are blocking us from implementing our Sprint Goal?

An alternate format to the three questions focuses more on the PBIs in progress rather than on individual Team Members. The Scrum Master asks simply “Where are we on this item?” And team members who have worked on it or would like to work on it can speak up.

The Daily Scrum is open to anyone for observation but only Scrum Team members can speak. The goals are to quickly see how things are going, adjust for any new findings and make a plan for the day. This meeting is typically held in the morning but can happen whenever the team decides.

Impediments that are not easily managed by the Development Team Members are passed to the Scrum Master for resolution. Impediments may be noted on the Sprint Task Board or in some other form for visibility.

It is important to keep this meeting to 15 minutes so that the Team can get to work and maintain momentum. Some guidelines that help achieve this include:

- Hold the meeting at the same time and place each day
- Meet in front of a physical Sprint Task board
- Stand up to help keep discussion shorter (This meeting is commonly referred to as The Standup Meeting.)
- Always start on time, with consequences for anyone being late
- Defer any detailed discussion or “solutioning” of particular challenges until after the Daily Scrum and include only interested Team members

The Daily Scrum may include participation by the Scrum Master and Product Owner as either may have relevant news or observations to share with the Development Team members.

Failure to have this meeting each Sprint day will lead to inefficiencies including duplication of effort, individuals stuck on an issue that someone else can resolve, reinforcement of “silo” behaviors and hidden impediments. It is important to remember that collaboration is continuous in Scrum and not limited to the Daily Scrum meeting.

Sprint Review Meeting

A public Sprint Review Meeting is held at the end of the Sprint. The Development Team demonstrates the completed product increment to the Stakeholders to get their feedback. The Product Owner has already reviewed and accepted the work being demonstrated.



The Product Increment is demonstrated in as realistic an environment as possible. For software this means real code running in a real or simulated production environment. For non-software products it can take many other forms. The goal is to instill confidence in the stakeholders that the Product Increment is complete and robust.

Many Teams refer to this meeting as the “Demo”, a historical term that misrepresents its intended purpose.

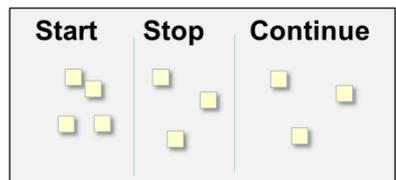
The Product Owner updates Stakeholders to the current progress of the Release with regard to the calendar, budget and remaining scope of features. This meeting may also be used to assess possible shifts in emphasis or priority for the Release Plan. The Sprint Review typically takes 1 to 2 hours depending on the length of the Sprint.

Note: Some early Scrum literature defines the Sprint Review as a single meeting that includes the Product demonstration for feedback and the Retrospective. Since the two meetings have different purposes and attendees, we now consider them to be separate.

During this meeting, the Product Owner decides to continue development, stop development, shift product direction in response to new information and/or ship unreleased features to customers

Sprint Retrospective Meeting

The final meeting is the Sprint Retrospective, facilitated by the Scrum Master. In this private meeting the entire Scrum Team reviews their process and interactions during the Sprint to identify potential process improvements.



Gathering Retrospective Data

“The Retrospective is the most important meeting in Scrum.”

- Jeff McKenna, Coach for the first Scrum Team

New ideas, interpersonal conflicts and impediments impacting the whole Team may be discussed. The goal is to identify ways to improve the work

of the Team. A list of potential process changes is created and a plan of action is drawn up for the top one or two opportunities discovered. A typical meeting length is 1.5 hours.

Continuous process improvement is a primary goal of Scrum. If the Scrum Team skips the Retrospective Meeting, it is a sign that there is an opportunity for improvement. Typical causes are complacency, insertion of unplanned work or lack of adjustment of the Sprint forecast to unforeseen events. Failure to inspect and adapt the process will result in a compounding of issues.

The Retrospective Meeting benefits from variation in format to keep it creative and productive. Some variations are offsite meetings, field trips to another team's space or sharing sessions with other teams.

Feedback Loops

Each Scrum Meeting is designed to close a feedback loop that creates an opportunity for inspection and adaptation to achieve increasingly better results. Specifically:

- The Sprint Planning meeting includes an assessment of what has been accomplished so far, how much the team is able to finish in an average Sprint and what factors may impact delivery in the upcoming Sprint.
- The Daily Scrum is a chance to balance work between team members, compensate for absences, adjust the Sprint Backlog to meet current conditions and identify impediments to completing particular PBIs planned for the Sprint.
- The Sprint Review is designed specifically to obtain feedback from Stakeholders on the Product Increment – fitness for purpose (does it solve the stated problems), fitness for use (can users leverage it efficiently) and new ideas triggered by the review process.
- The Sprint Retrospective is an official time for the Scrum Team to reflect on their process to seek and implement improvements in quality, flow and sustainability.

7. Scrum Artifacts

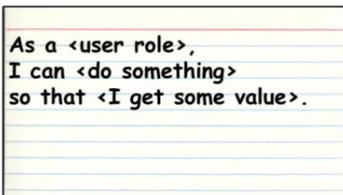
Scrum has three primary artifacts: the Product Backlog, the Sprint Backlog, and the Potentially Shippable Product Increment.

Product Backlog

Early in product development, the Product Owner with the help of stakeholders and the Team will create a prioritized list of features that will achieve the Product Vision. This list is called the Product Backlog. The Development Team contributes to the Product Backlog by breaking features into smaller packets of value, estimating the size of each and offering feedback on development order from a technical viewpoint.

The Product Backlog is a container for all features requested by the customers and stakeholders. Additional items may be included for development work that is not visible to the customer such as foundational work, refactoring (improved re-implementation of existing functionality) and short research efforts called “spikes”. Items on the Backlog that have highest value will be broken down into smaller, more detailed items. Less valuable features may remain large and less detailed.

Scrum Teams often define the product features using User Stories, a device borrowed from Extreme Programming (XP), a complementary Agile development framework. User stories describe functionality in a simple, high-level format: Who, What, Why.



User Story

User stories remind us of two important concepts that help to maximize the ROI: each product increment must deliver value to the user and smaller chunks of work gives us a smoother flow from concept to customer feedback.

Items on the Product Backlog are typically estimated in units of relative size as an input to Sprint Planning and longer term Release Planning. A common unit is Story Points, a simple number that compares each PBI

with all others in terms of size and complexity. This estimation is typically done at Release Planning time and as PBIs are refined over time.

The number of Story Points a Team actually completes in each Sprint is called “velocity”, a measure that can be averaged to forecast the amount of Product Backlog that is likely to be implemented in some future time period. Velocity can also be measured simply by the number of PBI’s completed per Sprint when items are all defined to be approximately the same size.

Acceptance Criteria

A helpful Scrum practice is the progressive elaboration of PBIs with **acceptance criteria**, details that are analogous to specifications used in traditional planning. In Scrum, these details only apply to the small slice of work identified by the PBI. They are created just-in-time as a PBI becomes more likely to be worked on in an upcoming Sprint.

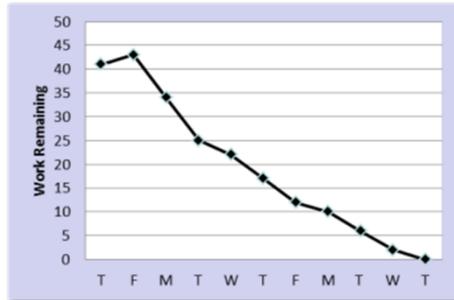
Acceptance criteria can include screen layouts, parameter ranges, test cases, sample data and activity flows. These details make it easier for the Development Team to understand the solution identified in the PBI and to jump start testing.

Sprint Backlog

The Sprint Backlog is a subset of the Product Backlog, elaborated to more detail. In the Sprint Planning Meeting, the Sprint Backlog is created to contain the set of chosen Product Backlog Items and a plan for implementing them. A common Sprint Planning technique is to break each Item into implementation tasks. The tasks are defined in chunks that will likely take 1 to 8 hours of work for a Development Team member.

The work remaining on the Sprint Backlog after each day of the Sprint can be tracked on a Burndown Chart. As tasks are completed, the number of estimated work hours or total tasks remaining is reduced or

“burned down” accordingly. If all work is completed in the Sprint, the work remaining is zero at the end. The Sprint Burndown chart is a simple but powerful device for seeing progress and trends as well as showing the Team how they are tracking toward the planned Sprint goal.



Sprint Burndown Chart

Once a Sprint is planned, the Sprint Backlog is fixed so that the Team can focus on the work. While the Product Backlog is open to new feature requests, the Sprint Backlog is not. Adding an additional request to a Sprint will, on average, replace two planned items due to the context switching overhead of re-planning.

It is important to keep in mind that a Sprint Goal and Sprint Backlog are targets rather than commitments. Teams are expected to work at a “sustainable pace”, a reasonable level of effort that they can sustain indefinitely. There is no requirement in Scrum that everything forecast in Sprint Planning be accomplished at any cost. Such an expectation would create a “push system” dynamic that will quickly break Scrum. Rather, Scrum Teams implement a “Pull System” in which the system pulls in work at a rate that matches it’s capacity while continuously watching for ways to be more effective.

The Development Team may adjust the Sprint Backlog in three ways:

- PBIs can be removed from the Sprint Backlog and returned to the Product Backlog if it becomes clear that they will not be accomplished in the Sprint. This can happen for many reasons including new understanding of a feature request, loss of one or

more team members for part of the Sprint and unanticipated external events.

- The Team is allowed to pull additional PBIs into the Sprint if their goal is met early. The Sprint continues until the end of its regular time-box, so the Scrum Team can choose to build more features or otherwise use the time productively.
- If the Sprint Goal becomes obsolete due to unforeseen circumstances, for example by a competitive surprise or team sickness, the Sprint can be cancelled by the Product Owner. A legacy term for this safety valve is “Abnormal Termination”, abnormal because it should be a rare occurrence.

Potentially Shippable Product Increment

The new product functionality created during a Sprint is called the **Potentially Shippable Product Increment**. Each Sprint outcome must meet the high quality bar desired for Scrum. Each PBI must be completed to the satisfaction of the Team’s Definition of Done, a process and quality checklist, and achieve a part of the Product Owner’s vision. When these criteria are true, the product is potentially shippable, i.e. it is fully tested and it works.



Not all Scrum Teams actually ship the product after every Sprint. In many environments it is more efficient for the Team to accumulate several Sprints worth of new functionality into a release that has sufficient value to justify the cost of packaging and deploying to the end-users.

8. Product Backlog Refinement

The Scrum Team will spend 5% to 10% of the Sprint time-box preparing for the future. The time is devoted to Product Backlog Refinement which consists of two activities.

The first activity is ongoing refinement of the Product Backlog, done by the Product Owner outside of the standard Sprint meetings. It includes:

- Addition of new PBIs that have emerged from new knowledge
- Specification of acceptance criteria for upcoming PBIs
- Preparing PBIs for the next Sprint by elaborating them to a level defined in the Team's agreed-upon Definition of Ready
- Removal of obsolete PBIs
- Reprioritization as needed
- Estimation of new and newly split PBIs

The second activity is an unofficial but common meeting mid-Sprint in which the Product Owner and Development Team discuss new PBIs and PBIs that are likely to be in the next Sprint Backlog. This meeting is commonly called the Backlog Refinement Meeting, Backlog Grooming Meeting, or Story Time.

These Product Backlog Refinement activities enable the Scrum Team to maintain a smooth flow from Sprint to Sprint and achieve a higher throughput at a sustainable level of effort. Flow is improved by having a shorter Sprint Planning Meeting, fewer questions from Development Team to Product Owner during the Sprint and quicker completion of PBIs.

9. Team Agreements

The Scrum Team will mature more quickly and achieve significantly better results if they create a number of working agreements. These agreements are initially made when the team is formed and then refined through retrospection. A common way to get these started is through a Team Chartering exercise with the help of an Agile Coach or an experienced Scrum Master. Agreements include operational elements such as meeting days and times, process elements such as the choice of tools for managing the Scrum artifacts and collaboration agreements such as methods for achieving consensus and collecting retrospective data.

There are two especially valuable agreements that can improve team throughput significantly, described below.

Definition of Done

Each Sprint should end with working software. The Definition of Done defines what “working” means. The DOD may contain process items, quality definitions and non-functional system requirements.

Sample DOD

- ✓ Unit tested to 90% coverage
- ✓ Code reviewed
- ✓ Acceptance tests pass
- ✓ UI Tested
- ✓ User Help updated
- ✓ Scales to 1 Million Users
- ✓ Meets response time targets

The Definition of Done will evolve as the Team matures. Evolution is driven by observed quality issues, skills development, new technical tools and practices and changes in operational requirements.

The Team may also have a DoD for the Sprint as a whole and another for a Release.

Multiple Teams working on the same Product will share most elements of their DoDs.

Note that the Definition of Done applies to all or most PBIs whereas acceptance criteria are further details that apply to individual PBIs.

Definition of Ready

The Product Owner and Development Team will benefit from a set of readiness criteria for PBIs to be pulled into a Sprint Backlog. These criteria are collectively called the Definition of Ready. Any PBI that does not meet the agreed criteria can be rejected by the Development Team.

Sample DOR

- ✓ PBI right size
- ✓ Screen sketches
- ✓ Acceptance criteria
- ✓ Dependent PBIs done
- ✓ Sample data

Identification of wait states in the flow of work can reveal additional items for the Definition of Ready. Anything that delays development activity or impedes work flow can indicate the need for more preparation of some aspect of the work.

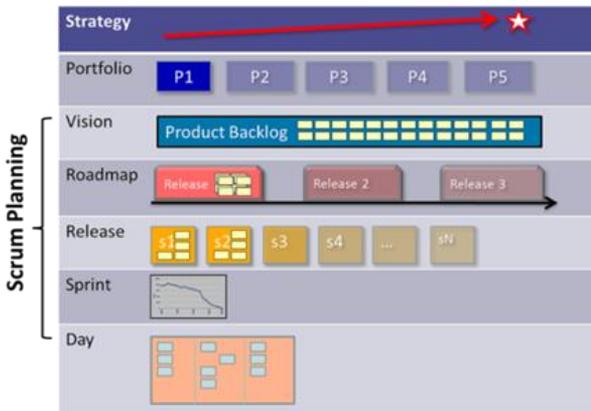
These two agreements greatly improve the flow of each Sprint by simplifying Sprint Planning, reducing uncertainty about the Sprint work, eliminating Sprint work delays and helping to avoid the spill-over of unfinished work into the next Sprint.

10. Scrum Planning

Planning is continuous in Scrum and occurs at five levels of detail related to five timeframes. The levels are defined for a single product in an overall portfolio that is designed to achieve the organization’s overall strategic goals.

Planning Levels

- **Product Vision:** The reason we are delivering one product
- **Roadmap:** The long-term (9-18 months) release goals for one product
- **Release:** The set of features we plan to release by the end of a sequence of Sprints
- **Sprint:** The goal and Product Backlog Items we forecast to deliver in the current Sprint
- **Day:** What the Team plans to accomplish today



Visioning can be for the entire product timeline and for the next defined release. Road-mapping is applicable to long-term development for a single product and may be unnecessary for shorter efforts. Sprint Planning is handled in the meeting of the same name. Daily planning is handled in the Daily Scrum. The Release Planning level is described below.

Release Planning

Release Planning is a process of building a Product Backlog with sufficient information to allow forecasting the answer to one of these questions:

- When can we ship this backlog?
- How much of this backlog can we ship by a specified date?

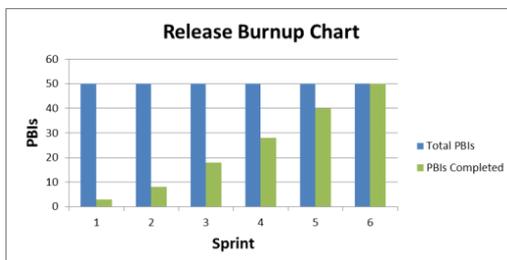
Information needed for the forecast is developed during a Release Planning meeting that typically lasts from half a day to two days depending on how far into the future we are forecasting and how many teams are going to be working on the backlog. A typical agenda for this meeting is:

- Define and study the Product Vision
- Identify relevant User Roles or personas for the product
- Brainstorm User Stories for those roles
- Prioritize the User Stories
- Estimate the size of the User Stories
- Forecast Team velocity

The Release forecast is then made by dividing the size of the backlog by the forecast velocity. For a new team there will be a fairly wide margin of error in the forecast velocity. It takes at least four Sprints for a reliable team velocity to be observed. In later releases the velocity of an established, stable team will be more predictable.

For budgeting purposes, a total labor cost for the Release is simple to calculate for a stable team since Sprints have a constant cost.

The Release Plan is revised each Sprint based on actual Team velocity and changes to the Product Backlog. Release progress is tracked on a Release Burnup or Burndown chart.



11. Agile Engineering Practices

The Scrum framework was originally designed within the context of software development. It is now used for other types of product development. When creating software, Scrum is more successful when complemented by a number of development practices defined in another framework known as Extreme Programming or XP. Since the 1990's when both frameworks were first applied, other technical practices have evolved to increase the success of Scrum-based work. Teams that use Scrum for software product development and delivery will be much more effective if they leverage these technical practices:

- **Co-location:** all Team members sitting together in one room
- **Pair Programming:** two or more people writing code together at the same computer
- **Test-First Development,** also known as Test-Driven Development: incrementally writing automated unit tests before writing the production code
- **Automated Acceptance Testing** or Story Testing: automated functional testing of each Product Backlog Item
- **Refactoring:** Improving the design and implementation of existing functionality for easier extension, higher operational efficiency and improved quality
- **Continuous Integration:** continuously running the automated tests to ensure all are passing before deployment
- **Continuous Delivery/DevOps:** automated deployments with monitoring and rollback
- **Managing Technical Debt:** continually improving poor design, steadily removing known defects and avoiding quick-fix solutions, all of which reduce quality and add to future development time
- **Evolutionary Design:** coding for change and allowing the system design to evolve as needed
- **Agile Architecture:** high-level design that anticipates change with minimal disruption

12. Scrum Benefits

There are many benefits to implementing Scrum that apply to organizations, employees and customers. Scrum has proven to be a big change for existing organizations, however, so the benefits come with the costs of change – new practices, mindset, tools, organizational structures and eventually corporate culture. Parallel to the adoption of Scrum in industry, the career of “Agile Coach” has evolved to help guide organizations in making the change.



Organizational Benefits

The adoption of Scrum and related Agile product development frameworks allows an organization to be more adaptable to changing customer desires, market opportunities and competitive pressures. Scrum enables companies to quickly change direction based on new information, leading to a faster time to market for new products and product enhancement.

The short iterative cycles and dedicated, stable teams used in Scrum provide resilience to surprises. Constant feedback reduces internal surprises. Short development increments provide quicker response to external surprises. Cross-trained teams reduce the risk of missing expertise during any one Sprint.

Agile companies can be more profitable. Scrum Teams are more efficient than traditional “resource planning” models due to close collaboration and extended time to “mature” into a high performing team. Scrum Teams are expected to build high quality product, reducing the frequency and cost of repairs. Scrum Teams focus on the highest value features and build the minimally necessary solution, reducing the time to market and later time required for maintenance.

In most companies, development is slowed down by issues identified as impediments during the daily meetings or planning and review meetings. With Scrum, these impediments are prioritized and systematically

removed, further increasing productivity and quality. Mature Scrum teams can achieve 4-8 times the productivity of the same number of people in a traditional process model and with significantly higher quality.

Companies that implement Scrum as designed typically have higher retention rates, reducing employee recruiting and onboarding costs.

Customer Benefits

Customers receive new functionality sooner. Through short feedback cycles, customers are more likely to receive a targeted, tested solution to the problems they are paying to solve. Scrum can provide solutions at lower cost to consumers. Customers have more frequent opportunities to give feedback to the providing company.

Employee Benefits

Scrum Team members are “self-organizing”, empowered to make many of their own decisions individually and through consensus. This autonomy results in more rapid problem solving, more frequent innovation and higher personal satisfaction.

Teams working in a “pull system” work at their own healthy pace, resulting in smoother flow, quicker response and a higher quality of life.

Team members are encouraged to learn from each other, creating an environment and culture of continuous learning and skill development. People are expected to not only master their career skills but also expand their knowledge and skill set.

13. Agile Values and Principles

Scrum works by implementing basic values and principles described in the Agile Manifesto at www.agilemanifesto.org.

Agile Manifesto Values

Individuals and interactions *over* processes and tools
Working software *over* comprehensive documentation
Customer collaboration *over* contract negotiation
Responding to change *over* following a plan

The Agile principles leverage the Toyota design and manufacturing tradition, now commonly called “Lean Product Development”. The following key principles at work in Scrum come from the Lean experience, Systems Thinking, robotics and Queuing Theory.

- Pull systems self-regulate by limiting work demand to available capacity with sufficient slack to allow for smooth flow and innovation. Pull systems have been shown to deliver significant productivity gains over “push” systems in lean product development. Pushing a system to full capacity or beyond will cause turbulence, delay, breakage and unpredictability.
- Process improvement is a continuous goal. There is always a way to do better.
- Small batches move through a system more quickly than large batches. An item of work is completed in the shortest time (cycle time) when taken from start to finish with no wait states (single piece flow).
- The more work in progress, the longer each work item will take to accomplish. Multi-tasking is inefficient because of the overhead spent switching between tasks.
- Low quality products that require maintenance will have a higher total lifecycle cost and therefore a lower ROI. It is actually quicker and more profitable to build quality in from the start.

- Teams can do things that no one individual can do. Synergy enhances productivity, knowledge growth and innovation.
- The people doing the work are the best experts on how to improve the work. Management sets the strategy, Team Members determine the tactics. This is called “de-centralized control”.
- Complex systems are self-organizing. To respond quickly, systems need feedback loops with minimal delay times. Traditional product development has long feedback cycles that are ineffective for learning what works and what needs improvement.
- People-based processes are dependent on trust and respect, both of which take time to achieve. Teams need time to mature.

Scrum Values

Most people like to work in teams. Teamwork fosters these **Scrum Values** which in turn lead to greater Team success:

- Focus to create product increments quickly
- Respect for teammates to promote collaboration
- Openness to new ideas and opinions
- Courage to experiment
- Commitment to doing high quality work

The Scrum Values are described in more detail in the Scrum Guide at www.scrumguides.org.

14. Success Factors

The most successful Scrum Teams have these characteristics:

- Full-time dedicated Development Team Members
- Full-time dedicated Scrum Master and Product Owner
- Longevity
- Decentralized decision-making
- Clear Definition of Ready in use
- Clear Definition of Done in use
- Operate as a Pull System

15. References

- Scrum Alliance www.scrumalliance.com
- Jeff Sutherland [Scrum: The Art of Doing Twice the Work in Half the Time](#)
- Ken Schwaber [Agile Project Management with Scrum](#)
- Mike Cohn [Succeeding with Agile: Software Development Using Scrum](#)
- Ken Rubin [Essential Scrum](#)
- Nonaka and Takeuchi [The New, New Product Development Game](#)
- Scrum Guide www.scrumguides.org
- Brown www.AgileCoachJournal.com

Index

- Abnormal Termination (see: Cancellation)
- Acceptance Criteria 16, 19, 20
- Adaptation 3, 14
- Agile Coach 20, 25
- Agile Manifesto 27
- Agile Product Development (see: Agile Software Development)
- Agile Software Development 1, 15, 24, 25
- Backlog Grooming (see: Product Backlog Refinement)
- Blockers (see: Impediments)
- Budgeting 5, 13, 23
- Business Value 5
- Cancellation 18
- Continuous Improvement 6, 14, 27
- Daily Scrum 2, 10-12, 14, 22
- Definition of Done 6, 18, 20, 29
- Development Team 2, 4, 6-8, 12, 15-17, 19, 21, 29
- Elaboration 16
- Empirical Process Control 3, 6
- Engineering Practices 24
- Estimation 6, 19
- Extreme Programming 15, 24
- Feedback 1, 3, 12-15, 25, 26, 28
- Forecast 10, 11, 14, 16, 17, 22, 23
- Impediments 7, 11-14, 25
- Inspection 3, 14
- McKenna, Jeff i, 13
- Multi-tasking 27
- Non-software Scrum 13
- PBI (see: Product Backlog Item)
- PBI Sizing 16
- Potentially Shippable Product Increment 2, 15, 18
- Prioritization 15, 23
- Process Improvement (see: Continuous Improvement)
- Product Backlog 2, 5, 6, 8, 10, 15-17, 19, 22-24
- Product Backlog Item 6, 15, 16, 18, 21, 24
- Product Backlog Refinement 19
- Product Owner 4-8, 12, 13, 15, 18, 19, 21, 29
- Product Roadmap 8, 22
- Product Vision 8, 15, 22, 23
- Pull System 10, 17, 26, 29
- Quality 6, 10, 14, 18, 20, 24-28
- Release Planning 8, 15, 16, 22, 23
- Return on Investment 5, 15, 27
- ROI (see: Return on Investment)
- Schwaber, Ken 3, 29
- Scrum 1-4, 7-20, 22, 24-29
- Scrum Alliance i, 29
- Scrum Master 2, 4, 7, 8, 11-13, 20, 29
- Scrum Team 4, 8, 10, 11, 13, 14, 18-20, 26
- Scrum Values 1, 7, 27, 28

Self-Organization 1-3, 6, 26, 28
Skills 6, 7, 20, 26
Spikes 15
Sprint 1, 2, 5-23, 25
Sprint Backlog 2, 11, 14-17, 19, 21
Sprint Goal 6, 10, 11, 17, 18
Sprint Length 9
Sprint Planning Meeting 2, 5, 8, 10, 14-17, 19, 21, 22
Sprint Review Meeting 2, 10, 12-14
Stakeholders 5, 6, 8, 10, 12-15
Story Points 15, 16
Sustainable Pace 9, 10, 17, 19
Sutherland, Jeff 3, 29
Tasks 10, 11, 16, 27
Team Capacity 10, 17, 27
Team Chartering 8, 20
Time-Box 9, 11, 18, 19
Transparency 3
User Stories 15, 23
Velocity 10, 16, 23
Waterfall 3
Working Agreements 20
XP (see: Extreme Programming)

Notes