# Managing Projects of Chaotic and Unpredictable Behavior

by Richard “Dick” Carlson

ALEA

**Agile & Lean Education Associates**

*"Teaching Agile & Lean from A to Z"*

# Managing Projects of Chaotic and Unpredictable Behavior

Dick Carlson, Agile & Lean Education Associates (ALEA)

## Introduction

When the ***Manifesto for Agile Software Development*** was written in early 2001, it started a movement that has ignited the software development community. The movement also generated controversy and debate; connected related movements in manufacturing, construction, and aerospace, and extended into project management. The essence of this movement, whether in new product development, product service, software applications, or project management, rests on four foundational goals: delivering innovative products to customers (particularly in highly uncertain situations), reducing the cost of development, improving the quality and reliability, and creating working environments in which people look forward to coming to work each day.

## Controlling Chaos

Many of our processes, rather than being repeatable, defined, and predictable, are unrepeatable, undefined and unpredictable. Having said that, I shall attempt to explain the difference between predictable (*defined*) and unpredictable (*empirical*).

A defined process is one that can be defined fully with all things known about it so that it can be designed and run repeatedly with predictable results, and usually it can be subjected to automation. If all things about a process are not fully known, then only what generally happens when you mix these inputs and what to measure and control to get the desired output–these are called empirical processes. A defined process is predictable; it performs the same every time within acceptable boundaries. An empirical process requires close watching and control, with frequent intervention. It’s chaotic and unrepeatable, requiring constant measurement and control through intelligent monitoring.

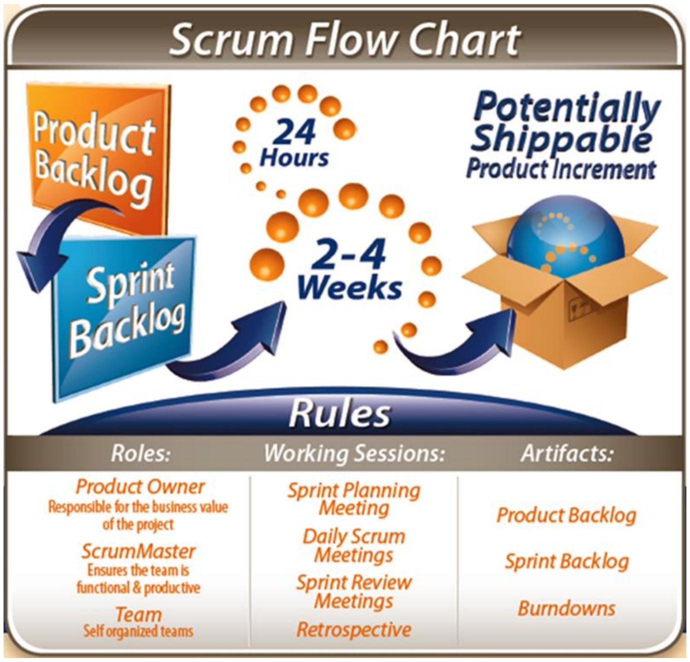
Models of empirical processes are derived by categorizing observed inputs and outputs and defining the controls that cause them to occur within prescribed bounds. Empirical process modeling involves constructing a process model strictly from experimentally obtained input-output data, with no recourse to any laws concerning the fundamental nature and properties of the system. No *a priori* knowledge about the process is necessary; a process is treated like a black box.

*Scrum* is a macro-process that defines and implements controls. Scrum, an agile approach to project management, consists of tasks that establish, monitor, and manage backlog, work, risk, issues, problems, and changes. Micro-processes, such as object-domain analysis, are used for actual product construction. A software project that implements Scrum is controlled by establishing, maintaining, and monitoring key control parameters. These controls are critical when a software development encompasses an unknown quantity of uncertainty, unpredictable behavior, and chaos. The use of these controls is the backbone of the Scrum development process. These controls are measured, correlated, and tracked. The main controls are executing the backlog and controlling risk by producing the highest value items first. The backlog should start relatively high, get higher during planning, and then be whittled away as the project proceeds – either by being solved or removed, until the software is completed. Risk will rise with the identification of backlog, issues, and problems, and fall to acceptable levels as the software is changed to meet the clarified goals.

[Note: Scrum as applied to product development was first referred to in "The New New Product Development Game" (Harvard Business Review 86116:137-146, 1986), and later elaborated in "The Knowledge Creating Company" both by Ikujiro Nonaka and Hirotaka Takeuchi (Oxford University Press, 1995).]

## Scrum Nuts & Bolts

With Scrum, projects progress through a series of iterations, usually 30-days in length, called sprints. Scrum is ideally suited for projects with rapidly changing or highly emergent requirements. The work to be done on a Scrum project is listed in the ***Product Backlog***, which is a list of all desired behavior and changes to the product. At the start of each Sprint a ***Sprint Planning Meeting*** is held during which the ***Product Owner*** prioritizes the Product Backlog and the ***Scrum Team*** selects the tasks it can complete during the coming Sprint. These tasks then are moved from the Product Backlog to the ***Sprint Backlog***. Each day during the sprint a brief daily meeting called the ***Daily Scrum*** is held, which helps the team stay on track. At the end of each Sprint the team demonstrates the completed functionality at a ***Sprint Review Meeting***. Daily scrums raise the visibility of each person's work to facilitate knowledge sharing, reduce overlapping tasks, and ensure that their work is integrated. The Scrum process is depicted in the following diagram.



**Stories are prepared, prioritized, and maintained by Product Owner**

**Stories are decomposed into quantifiable tasks to ensure all aspects of work are completed**

**The main criteria for a successful sprint is an increment of working software**

**A simple set of rules agreed to by the Team maximizes productivity**

The Scrum Framework is an "inspect and adapt" framework consisting of three roles, three ceremonies, and three artifacts designed to deliver working software during Sprints.

# Scrum Roles

Scrum roles include the Product Owner, Scrum Master, and Team.

The **Product Owner** is responsible for the following:

- Defining the features of the product;
- Deciding on release date and content;
- Being responsible for the profitability of the product (ROI);
- Prioritizing features according to market value;
- Adjusting features and priority every 30 days, as needed; and
- Accepting or rejecting work results.

The **Scrum Master** is a facilitative team leader working closing with the Product Owner. The Scrum Master is responsible for:

- Ensuring that the team is fully functional and productive;
- Enabling close cooperation across all roles and functions;
- Removing barriers;
- Shielding the team from external interferences; and
- Ensuring that the process is followed, including issuing invitations to Daily Scrum, Sprint Review, and Sprint Planning meetings.

The **Team**:

- Is a group of 5 to 9 members who are motivated and cross-functional;
- Works with the Product Owner to define the goals of each Sprint;
- Has the right to do everything within the boundaries of the project guidelines to reach Sprint goals;
- Selects work of the highest priority from the product backlog;
- Organizes and manages itself and its work; and
- Demonstrates work results to the Product Owner and key stakeholders.

# Scrum Ceremonies

There are four Scrum ceremonies (work sessions): Sprint Planning, Daily Scrum, Sprint Review, and the Sprint Retrospective meetings.

### Sprint Planning Meeting

The Sprint Planning Meeting is time-boxed to a maximum of eight hours, depending on the length of the sprint. Preparation for a Scrum sprint begins when the Product Owner develops a plan for a product or a project. A Product Owner needs a vision for the product that frames its ultimate purpose and a roadmap that plans out several releases, with features ordered by contribution to a return on investment (ROI). The Product Owner prepares a list of customer requirements prioritized by business value. This list is the Product Backlog – a single list of features prioritized by value delivered to the customer. [Note: The Product Owner role can be filled by a domain / technical expert within the company, a customer representative, or a customer proxy.]

The Scrum begins when enough of the Product Backlog is defined and prioritized to launch the first sprint. A Sprint Planning Meeting is used to develop a detailed approach for the iteration. It begins with the Product Owner reviewing the vision, the roadmap, the release plan, and the Product Backlog with the Scrum team. The team reviews the estimates for features on the Product Backlog and confirms that they are as accurate as possible. The team decides how much work it can take successfully into the sprint based on the team's size, availability of the members, and the level of the team's productivity. It is important that each member of the team "pulls" an item or items from the top of the Product Backlog that each member can commit to deliver in a sprint. Pull systems have been shown to deliver significant productivity gains in ***Lean product development***.

[Note: Sprint lengths can vary from one week to four weeks, and the length is decided by the team at the start of the project. While the length of a sprint remains the same duration throughout a project's lifecycle, there are instances when a team may change the sprint length after becoming familiar with its ability to perform or its productivity performance.]

When the Scrum team has selected and committed to deliver a set of top priority features from the Product Backlog, the Scrum Master facilitates the team in a session by itself to decompose the Product-Backlog items into quantifiable, sprint tasks. These tasks are the specific, development activities required to implement a feature or functionality and help form the Sprint's Backlog.

### Daily Scrum Meeting

Once sprint planning is complete, the sprint begins its iteration cycle. Each day the Scrum Master facilitates the team in the Daily Scrum Meeting. This is a time-boxed meeting that is limited to 15-minutes that promotes communication and transparency amongst team members. Each team member responds to three questions:

1. "What have you done since the last Scrum meeting?"
2. "What are you planning to do next?"
3. "What obstacles are preventing me from performing my work?"

While others are encouraged to attend this meeting, only team members who have committed to deliver work to the Scrum are allowed to speak. The goal is to get a snapshot of the project at its current state, discover any new dependencies, address any personal team needs, identify impediments, and adjust the work plan in real time to the needs of the day.

[Note: As a general rule, Product Owners and management are not invited to attend the Daily Scrum as the intimidating nature of some of these individuals destroys the intent of the meeting. However, in many personal cases, I have seen a significant number of Product Owners and project managers who were strong Agile advocates, thus were allowed and even encouraged to attend these meetings.]

### Sprint Review Meeting

At the end of a sprint, a Sprint Review Meeting is held. This meeting is time-boxed from 4 to 6 hours depending on the length of the sprint and the complexity of functionality. The first half of the meeting is set aside to demonstrate to the Product Owner the potentially shippable features that have been developed during the sprint. The Product Owner leads this part of the meeting and invites all interested stakeholders to attend. The state of the business, the market, and the technology also are reviewed. The Product Owner determines which items on the Product Backlog have been completed in the Sprint, and discusses with the team and stakeholders how best to reprioritize the Product Backlog for the next sprint. The goals for the next sprint are defined.

### Sprint Retrospective Meeting

The Sprint Retrospective Meeting is a reflection of what the team thought of the sprint. The Sprint Retrospective Meeting is facilitated by the Scrum Master. The team assesses the way they worked together during the sprint and identifies positive ways of working together that can be encouraged as future practice. The team also identifies the things that could work better and develops strategies for improvement. After the Sprint Retrospective Meeting, the process begins again. Sprints continue to proceed until enough features have been implemented to complete or release a product, or until the customer cessation.

## Scrum Artifacts

There are three Scrum Artifacts: Product Backlog, Sprint Backlog, and Burndown Chart.

### Product Backlog

At the beginning of the project, the Product Owner prepares a list of customer requirements prioritized by business value. This list is the Product Backlog – a single list of features prioritized by value to deliver to the customer. The Scrum Team contributes to the Product Backlog by estimating the size and effort of developing features. The Product Backlog should include all features visible to the customer, as well as the technical requirements needed to build the product. The highest priority items in the Product Backlog must be decomposed into small enough chunks (in detail) to be estimable for the next sprint. Features that will be implemented further out in time can be less detailed.

### Sprint Backlog

The Sprint Backlog is an artifact of the Sprint Planning Meeting. When the Scrum Team has selected and committed to deliver a set of top priority features from the Product Backlog, the features are further decomposed into tasks that are added to a Sprint Backlog, which is a list of the specific development tasks required to implement a feature. Task decomposition should reflect a minimum of 4 hours and a maximum of 16 hours of work. (Decomposition times of 4 to 8 hours should occur for sprints of 2 weeks; longer sprints can take more time to decompose.) When the Sprint Backlog is complete, the total work estimated is compared with the original estimates from the Product Backlog. If there are significant differences, the team will negotiate with the Product Owner to get the right amount of work to take into the Sprint with a high probability of success.

[Note: While there is value to using a Sprint Backlog, I have found that most teams prefer a more visible and eye-catching tool for tracking tasks such as a taskboard, where individual tasks are written on PostIt Notes$^©$ and placed on the taskboard for all to see.]

### Burndown Charts

Burndown Charts show the cumulative work remaining in a Sprint on a day-by-day basis. At the Sprint Planning Meeting, the Scrum Team identifies and estimates specific tasks that must be completed for the Sprint to be successful. The total of all Sprint Backlog estimates of work remaining to be completed is the cumulative backlog. When tasks are completed as the Sprint proceeds, the Scrum Master recalculates the remaining work to be done and the Sprint Backlog decreases, or *burns down* over time. If the cumulative Sprint Backlog is zero at the end of the Sprint, the Sprint is successful. Product Backlog items brought into the Sprint are fixed for the duration of the Sprint. However, the Sprint Backlog may change for several reasons:

- The development team gains a better understanding of work to be completed as time progresses and may find that they need to add new tasks (additional work) to the Sprint Backlog to complete the Product Backlog items selected.
- Defects may be identified and logged as additional tasks. While these are viewed primarily as unfinished work on committed tasks, it may be necessary to keep track of them separately.
- The Product Owner may work with the Team during the sprint to help refine the team's understanding of sprint goals. The Scrum Master and Team may decide that minor adjustments that do not lengthen the sprint are appropriate to optimize customer value.
- Burndown Charts are used as a tool to guide the development team to successful completion of a Sprint on time with working code that is potentially deliverable as a product.

## Scrum Attributes and Benefits

Scrum is an iterative and incremental approach for developing any product or managing any kind of work. Scrum produces an increment of potentially shippable set of functionality at the end of every sprint. Scrum attributes are summarized as follows:

- Scrum is an Agile approach to managing and controlling development work.
- Scrum is a wrapper for existing engineering practices.
- Scrum is a team-based approach to iteratively and incrementally developing a system and other products when requirements are rapidly changing.
- Scrum controls the chaos of conflicting interests and needs.
- Scrum is a way to improve communications and maximize co-operation.
- Scrum is a way to detect and cause the removal of anything that gets in the way of developing and delivering products.
- Scrum practices enable transparency.
- Scrum maximizes productivity.
- Scrum is scalable from single projects to entire organizations.
- Scrum has the ability to control and organize development and implementation for multiple and interrelated products and projects with over a thousand developers and implementers.
- Scrum is a way for everyone to feel good about their job, their contributions, and that they have done the very best they possibly could.

Scrum focuses an entire organization on building successful products. Without major changes, often within 2 to 4 weeks, teams are building useful, demonstrable product functionality. Scrum can be implemented at the beginning of a project or in the middle of a project or for a product development effort that is in trouble.

Scrum is a set of interrelated practices and rules that optimize the development environment, reduce organizational overhead, and closely synchronize market requirements with iterative prototypes. Based in modern process control theory, Scrum causes the best possible software to be constructed given the available resources, acceptable quality, and required release dates.

Useful product functionality is delivered every 2 to 4 weeks (as determined by the project team) as requirements, architecture, and design emerge, even when using unstable technologies.

Scrum wraps an organization's existing engineering practices, which are tailored as necessary to support the Scrum framework while product increments are delivered. Scrum is spelled out as values, practices, and rules in a development framework that can be quickly implemented and repeated.

## Some Conclusions

After reading this article, you should have a better understanding of the process. So, why is it so effective?

1. By producing only the functions with the highest value, the likelihood of producing a function of little or no value is nearly eliminated. This reduces or eliminates waste.
2. Planning sessions for every sprint ensures that only the most important and highest value features will be built or implemented, only the team will estimate the items they will build during the sprint, and all sprint activities will be transparent for all to see.
3. The Daily Scrum provides a daily insight into progress, and this provides into better control, thus project management has a better insight into what is occurring. This should please the Project Manager and executive management.
4. The Daily Scrum identifies problems sooner rather than later. The longer a problem exists, the greater the likelihood of creating more problems or of allowing existing problems to grow in magnitude.
5. Better control reduces the magnitude of the chaos, so this makes control easier.
6. Smaller variations in chaos reduce the variations in unpredictability. The likelihood of success is improved by reducing unpredictability.
7. Sprint reviews are held for the benefit of all stakeholders so they can observe progress, determine what needs to be built next, offer constructive criticism when appropriate, provide valuable feedback to the team on their performance and development methods, and allow customers and users opportunities to recommend new technologies and introduce changes that adds value to the product being built.
8. Unresolved impediments form a drag on the development of the product, so the sooner that the impediment is mitigated or removed, the sooner the drag is reduced or eliminated.

## References

1. *Manifesto for Agile Software Development* (http://www.agilealliance.org)
2. *Scrum Alliance*, (http://www.scrumalliance.org)
3. Jim Highsmith, *"Agile Project Management,"* Addison Wesley, 2004
4. Ken Schwaber and Mike Beedle, "*Agile Software Development with Scrum,*" Prentice Hall, 2001

# About the Author

Dick Carlson has an extensive engineering background that includes many years of practical knowledge and hands-on experience in the implementation and deployment of Agile, Lean, and Scrum values and principles in communications-electronics engineering, software engineering, and systems engineering within the aerospace, DoD, IT, and industry domains. He has developed and actively conducted comprehensive training courses for Scrum Teams, Scrum Masters, Product Owners, project/program managers, customers, executives, organizational leaders, and others interested in learning how to implement and deploy Agile, Lean, and Scrum.

Recently retired from Boeing, Dick has been an active transformational leader for many small and large Agile projects, and frequently shares his experiences of successful Agile implementation at conferences, workshops, and symposia. He regularly counsels executives and leaders on the benefits of using Agile, actively coach's teams on Scrum and Lean-Agile Project Management fundamentals, and then follows up with mentoring activities to ensure successful project implementation.

Mr. Carlson has a Bachelor of Science degree from the University of Maryland, and is a Certified Scrum Professional, Certified Scrum Master, and Certified Scrum Product Owner, and holds a Lean-Agile Project Management certification. Dick has presented Agile topics at the IEEE Software Technology Conferences every year since 2010.