Foreword

This book documents my knowledge of working as a business analyst on *agile* projects. It covers 6 years of work experience with desktop, agile and cloud based software systems.

- ♦ **I emphasized the word agile, to indicate that the 1 thing that all of these projects had in common are short delivery cycles, or sprints.**

What I have learned is that the Scrum process framework does not describe the complete story.

Scrum identifies 3 roles, the development team, product owner and Scrum master. Scrum does not include roles for the business analyst, system architect, quality assurance team, tester, UI designer, deployment manager or writer. Using Scrum, the work that is normally assigned to these roles, is performed by the development team. Scrum assumes that the development team includes people with all of the skills normally associated with these roles.

The problem with this situation is that all of the work performed by the development team happens within a sprint cycle.

- ♦ **The only common Scrum activity performed outside of a sprint cycle is product backlog maintenance (and even this activity is not documented in the Scrum framework).**

In this book, I describe activities performed outside of the sprint cycle and identify the benefits that they bring to implementation of a deliverable product.

In my experience, some projects included a Scrum master and others have no official Scrum master role. The roles that all of these projects had in common are, a product owner (sometimes known as the product manager), a development team, and of course, a business analyst.

- ♦ **With no identified Scrum master role, as the business analyst, I would often be assigned many of the Scrum master responsibilities**

Most development teams were supported by a separate user interface design team and an independent quality assurance (QA) or testing team. Other roles that are outside of development include deployment and operations, system architects and document writers. These roles provide specific responsibilities in support of releasing software, even when an agile process is employed.

This book documents responsibilities, deliverables and best practices for all these roles as they relate to agile systems development. These activities are captured from a business analyst’s point of view. This is my experience of working on Scrum-like agile projects.

Definitions

While writing this book, I discovered that I was using some terms interchangeably and inconsistently. For this reason, I have defined the essential terms used in this book up front. In this manner, the reader can easily clarify the intended meaning of words when their use might be ambiguous.

The following terms take this meaning when used in this book:

- Acceptance Criteria – This is a set of instructions detailing how the product is used.
- Activity – Something that is done as part of the process that delivers 1 or more artifacts and is the responsibility of a single role.
- Agile – The ability to develop software while responding to change, as defined on the Agile Alliance website at https://www.agilealliance.org/.
- Backlog – A repository of work to be done.
- Build – Working software that is the result of a development cycle (1 or more builds may be combined to make a release).
- Business Analyst – A role that is responsible for ensuring the quality of the product that is delivered to the customer.
- Development (Scrum) Team – A role (played by a group people) defined on the Scrum website as 'consisting of professionals who do the work of delivering a potentially releasable Increment of "done" product at the end of each Sprint'.
- Done – A state that is assigned to a user story when all development work is complete. It is left up to the Scrum team to identify when development is complete, but for the purpose of this book, all completed user stories in a sprint are considered done at the end of a sprint.
- Epic – A user story that captures the needs of the business; an epic is independent of development life cycle, spawns 1 or more user stories which may span several sprints.
- Framework - An outline for a process; the framework is not intended to be adopted as specified, but adapted to suit your organization. This book describes a framework in which its components can be adopted and modified to make a software development life cycle process that is appropriate for your situation.
- Model – An abstract representation of a system. A model captures information that is useful to its audience while hiding unnecessary details about the system.
- Process – A predefined set of activities, roles, and artifacts which when put together produces a product that can be delivered to a customer.
- Product – The software that is delivered by the development team.

- Product Owner – A role in a Scrum process framework and defined on the Scrum website as, ‘responsible for maximizing the value of the product resulting from the work of the Development Team’.
- Project Team – All of the people in the process that do work towards delivering a system release.
- Quality – A measure of the satisfaction that the product provides to the customer of that product. Quality can be measured by the cost to ‘fix’ the product when it does not perform as the customer expects and the loss of revenue as a result.
- Quality Assurance – The act of ensuring that every component in the product lifecycle is adding quality to the product that is released to the customer.
- Ready-for-dev - Ready for development; is a state that is assigned to a user story when it has been prioritized for consideration as a candidate for a sprint backlog.
- Release – A product build that is made available to the customer.
- Release Cycle – The period of time between software builds and producing a release. Several builds may be produced during 1 release cycle.
- Requirement – A specification of behavior that the system will exhibit. This may take the form of a use case, a user story, text or a diagram.
- SAFe – When references are made to the Scaled Agile Framework they are taken from the SAFe website at https://www.scaledagileframework.com/.
- Scrum – This book uses the resources at https://www.scrum.org/ when referencing Scrum
- Software Development Life Cycle – A series of processes that encompass delivery of a software product from inception through to retirement of the software. Whereas a process delivers a complete product in a single release iteration, a software development life cycle may involve cycling through many iterations of more than 1 process.
- Sprint – A fixed period of time where Scrum ceremonies are performed and an incremental build is produced.
- System – The combination of hardware and software that is used to satisfy stakeholder needs.
- User Story – A description of a piece of work that may be completed in a single sprint.
- View – A diagram or related set of diagrams, that provide information from the model for a specific audience or purpose.

♦ **Note that historically all items in the backlog were called user stories. Epics are often referred to as user stories that are too large for a single sprint. I distinguish between user stories and epics by listing both epic and user story as separate items. For the purpose of this book the term user story does not refer to an epic.**
