Agile Dev
Better Software
DevOps WEST

A TECHWELL EVENT

AT5

Agile Practices
Thursday, June 7th, 2018, 11:30 AM

Let's (Re)Learn about Agile and Scrum in One Hour!

Presented by:

Steven Spearman

Swift Ascent, LLC

Brought to you by:

TECHWELL™

Steven Spearman
Swift Ascent, LLC

A founder and principal at Swift Ascent, LLC, Steve Spearman is a Certified Scrum Trainer and an agile coach working at the enterprise and team levels. Steve has more than thirty years' experience in corporate software development as a software developer, architect, project manager, ScrumMaster, and multiple management positions in a variety of enterprises. For the past four years, Steve has been providing agile-focused training and coaching to companies from late-stage startups to large corporations. Find out more about Steve on LinkedIn.

Let's (Re)Learn about Agile and Scrum in One Hour!
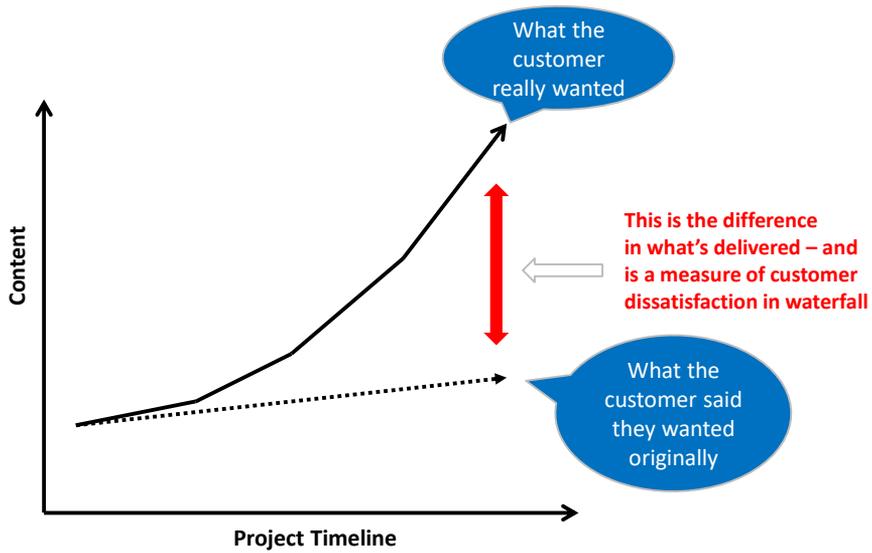
Welcome!

Steve Spearman

- 30+ years of software development experience
- CST, CSM, CSPO, PMP, PMI-ACP, (former) SAFe SPC, Certified LeSS Practitioner
- Certified Scrum Trainer and coach for Agile teams and organizations
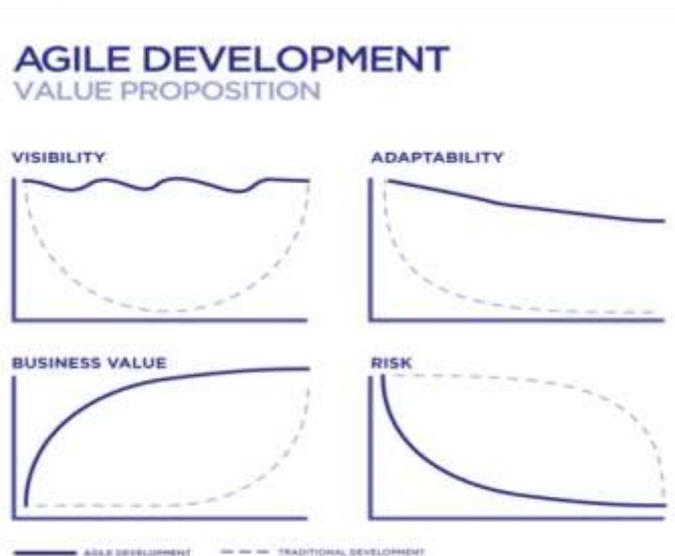- Based out of Denver (USA)

@sgspearman
stevespearman

Why the Move to Agile?

What the customer really wanted

This is the difference in what's delivered – and is a measure of customer dissatisfaction in waterfall

What the customer said they wanted originally

Content

Project Timeline

3

Copyright 2018, Agile for All

Why Agile - Executive Perspective

AGILE DEVELOPMENT VALUE PROPOSITION

VISIBILITY

ADAPTABILITY

BUSINESS VALUE

RISK

AGILE DEVELOPMENT   TRADITIONAL DEVELOPMENT

http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/

Copyright 2018, Agile for All

Another Reason for Agile - Complexity

Complex

the relationship between cause and effect can only be perceived in retrospect

probe – sense - respond

emergent practice

Complicated

the relationship between cause and effect requires analysis or some other form of investigation and/or the application of expert knowledge

sense – analyze - respond

good practice

novel practice

no relationship between cause and effect at systems level

act – sense -respond

Chaotic

best practice

the relationship between cause and effect is obvious to all

sense – categorize - respond

Simple

© Cynefin framework by Dan Snowden

Outcome vs Output

Output – a measure of what was built
Outcome – business value & impact

Table discussion: what's the relationship between these and which is more important?

Timebox:
2 mins

What is Agile?

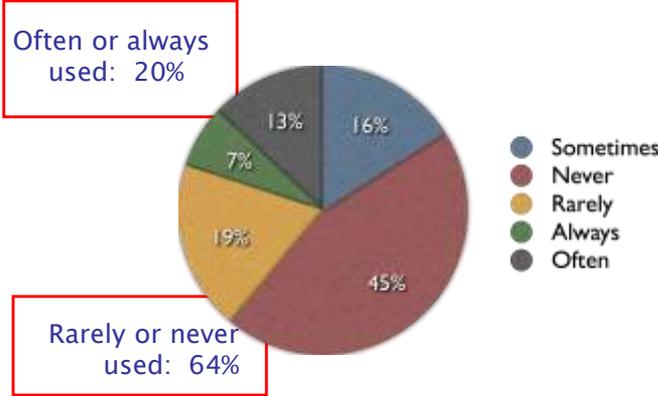It is often viewed as a process or something you adopt

But we would suggest it is more…..

A philosophy that great products come from:
collaborative, empowered teams
working closely with customers,
utilizing value-driven delivery,
adaptive planning &
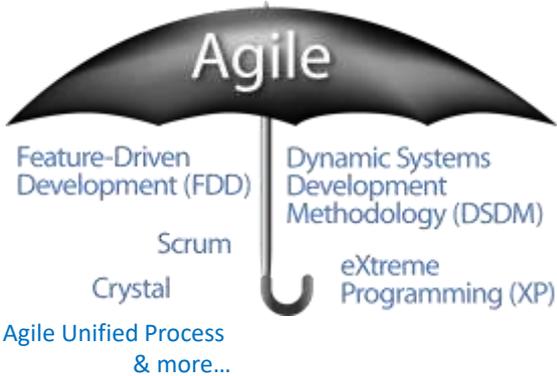continuous improvement of their methods

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck   James Grenning   Robert C. Martin
Mike Beedle   Jim Highsmith   Steve Mellor
Arie van Bennekum   Andrew Hunt   Ken Schwaber
Alistair Cockburn   Ron Jeffries   Jeff Sutherland
Ward Cunningham   Jon Kern   Dave Thomas
Martin Fowler   Brian Marick

2001

Snowbird, Utah

Simplicity – the Art of the Work Not Done

Often or always
used: 20%

13%	16%
7%	
19%	
	45%

- ● Sometimes
- ● Never
- ● Rarely
- ● Always
- ● Often

Rarely or never
used: 64%

Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

"Agile" includes a Family of Frameworks

Agile

Feature-Driven
Development (FDD)

Dynamic Systems
Development
Methodology (DSDM)

Scrum

Crystal

eXtreme
Programming (XP)

Agile Unified Process
& more…

And maybe scaling frameworks like SAFe, DAD, LeSS, Agile@Scale

Intro to Scrum

Scrum is an Empirical Process

Transparency
Inspection
Adaptation

Process evolves along with product

5 Scrum Values

- Focus
- Respect
- Openness
- Commitment
- Courage

Three Scrum Roles – an Introduction

Product Owner

"**What** and **why**"
Vision & goals
Backlog
Releasing
Business expert
Daily with the Dev Team
Most accountable

ScrumMaster

Scrum Expert
Coaches
Facilitates
Protects the Team
Removes impediments
Servant Leader
Supports
     PO, Dev Team
     Whole organization!

Development Team

"**How**" & "**how much**"
Cross-functional
Multi-discipline
Estimates
Delivers
Creates & solves
Self-organizing
Evolves processes

What about these Roles?

Let's have a quick discussion at your table.   Scrum does not say anything about these roles. What do you think about them?
➢ Project Managers
➢ People Managers (manager, director, VP)
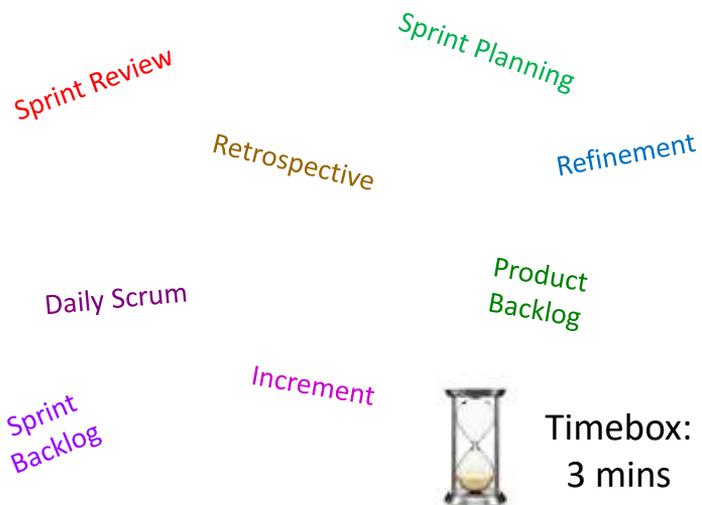➢ BAs / Architects

Timebox:
3 mins
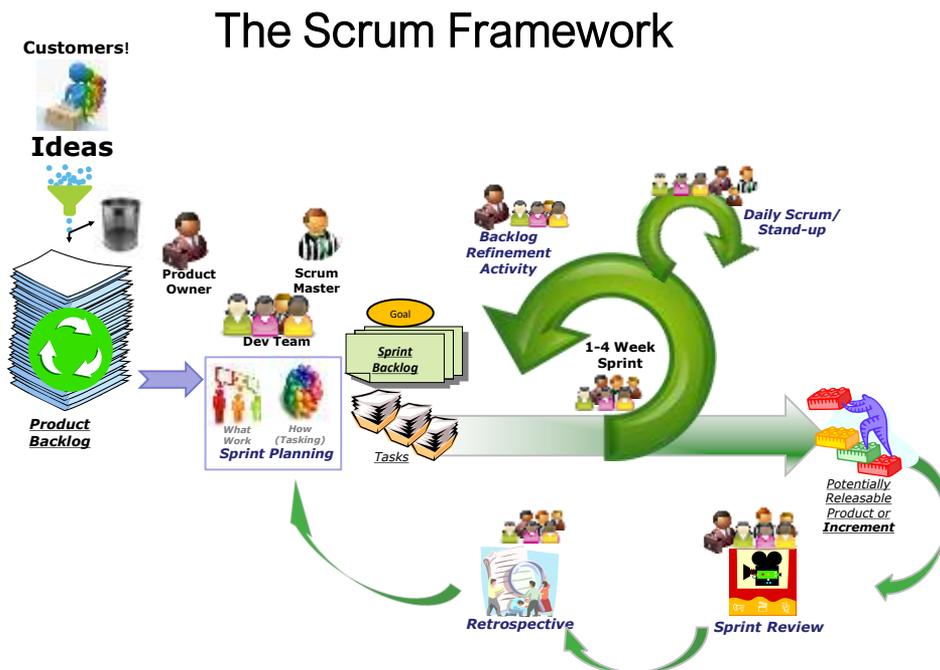
The Scrum Framework

Key Scrum Artifacts

The 3 primary artifacts for Scrum are:

➢ Product backlog
➢ Sprint backlog
➢ Product increment

Group Exercise: Draw the Scrum Framework

Sprint Planning

Sprint Review

Retrospective

Refinement

Daily Scrum

Product Backlog

Increment

Sprint Backlog

Timebox:
3 mins

Scrum Timeboxes

Sprint:	30 days
Sprint Planning:	8 hours
Daily Scrum:	15 minutes
Sprint Review:	4 hours
Retrospective:	3 hours

This are maximums for a 30 day sprint

Scrum is not a ...

... but it will reveal everything that the ScrumMaster or team needs to fix

Listen to Customers – but Also Disrupt

Do customers always know what they want?

  Copyright 2018, Agile for All

Scrum is a meta-process

Work Processes

Such as:
Coding Approach
Testing tools
Definition of Done
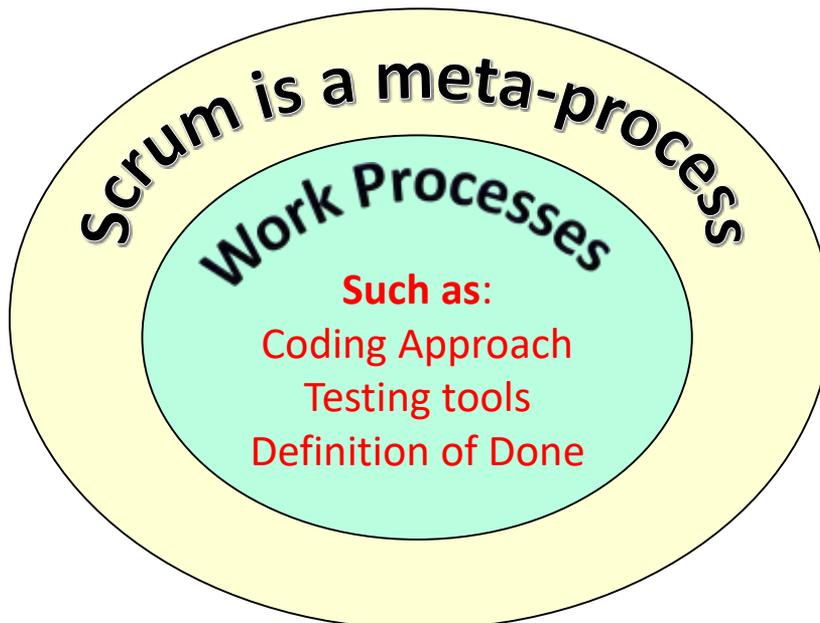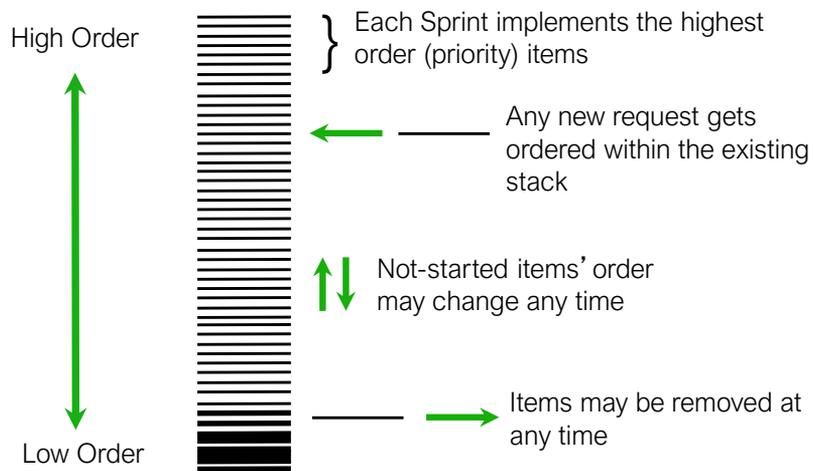
Emergent Knowledge

Because most things are emergent, Scrum teams:

➢ Talk more, write less

➢ Regularly show working product to users
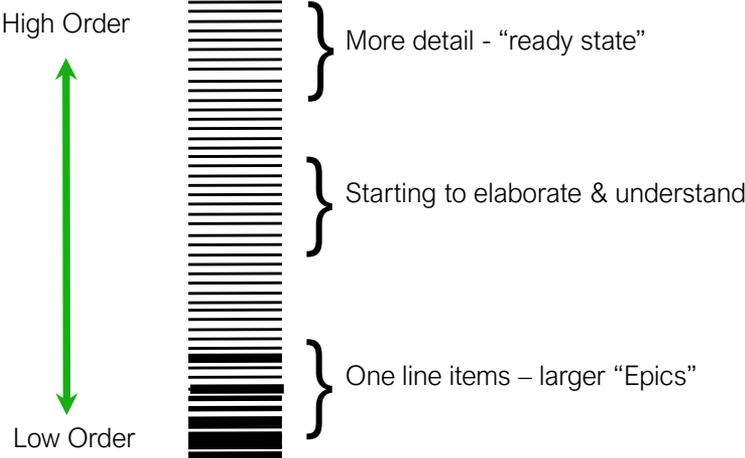
➢ Embrace change

➢ Experiment more than follow a plan

Product Backlog 101

High Order

} Each Sprint implements the highest order (priority) items

Any new request gets ordered within the existing stack

Not-started items' order may change any time

Items may be removed at any time

Low Order

Product Backlog 102 – Just in Time

High Order

} More detail - "ready state"

} Starting to elaborate & understand

} One line items – larger "Epics"

Low Order

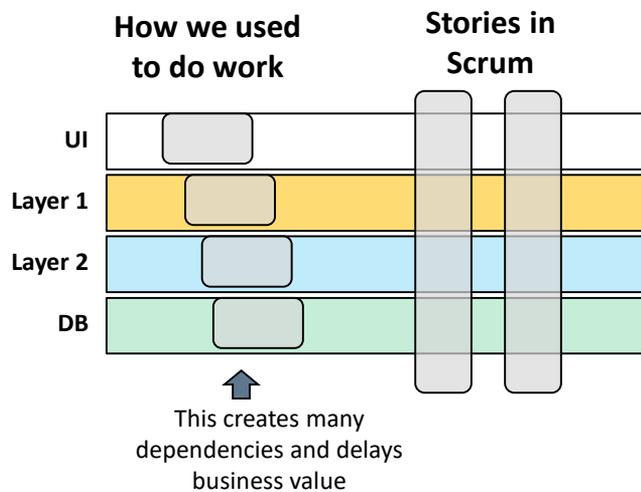# User Stories

Beyond Core Scrum
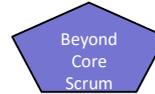
Some User Stories

Is this a complete user story?

As a Facebook user, I want to update my profile so that potential friends may find me.

Video Series:
"Chet Rong"
By Atlassian

27

Stories are Slices

How we used to do work　　**Stories in Scrum**

UI	
Layer 1	
Layer 2	
DB	

This creates many dependencies and delays business value

28

Scrum Roles Exercise

I have a little "Roles Game" for you at your table!

Timebox: 5 mins

ScrumMaster as Servant: Ask, Don't Tell

- ✓ I noticed <situation>; what shall we do?
- ✓ I observed <this>; is that important?
- ✓ I feel <feeling>; do you share that?
- ✓ Shall we try to find out why <situation>?
- ✓ What do you think we should do?

   …….

Does this sounds like they don't do much?

Is It an SM Function (True or False)

True!   Helping the Dev Team to create high-value products

No   Owning the processes for getting work done

True!   Leading and coaching the <u>organization</u> in its Scrum adoption

True!   Planning Scrum implementations within the organization

No   Serving as an interface between the PO and the Dev Team to convey information and clarify expectations

True!   Causing change that increases the productivity of the Scrum Team

No   Coordinating between multiple Scrum Teams so they can remain focused on their work

True!   Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization

Is It an SM Function #2 (True or False)

True!   Ensuring that goals, scope, and product domain are understood by everyone on the Scrum Team

True!   Finding techniques for effective Product Backlog management

True!   Helping the Scrum Team understand the need for clear and concise Product Backlog items

No   Running the Daily Scrum meeting to optimize team effectiveness

True!   Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value

No   Scheduling meetings and producing charts for the team

No   Presenting in the Sprint Review to ensure complete understanding by all those involved.

Scrum is Not Easy

1. Learning Scrum is a long journey
2. Feel free to reach out for more information

THANK YOU!

Steve Spearman
steve@agileforall.com

Suggested Reading

- Autry, James A, **The Servant Leader: How to Build a Creative Team, Develop Great Morale, and Improve Bottom-Line Performance,** Crown Business, 2004
- Beck, Kent, with Andres, Cynthia, **Extreme Programming Explained**. Addison-Wesley.
- Broderick, Stacia and Michele Sliger. **A Software Project Manager's Bridge to Agility**. Addison-Wesley
- Cohn, Mike. **User Stories Applied**. Addison-Wesley, 2004.
- Cohn, Mike. **Agile Estimating and Planning**. Addison-Wesley, 2006.
- Crispin & Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley Professional, 2009
- Derby, Esther and Larsen, Diana. **Agile Retrospectives - Making Good Teams Great.** Pragmatic Bookshelf, 2006.
- Poppendieck, Mary and Tom. **Lean Software Development - An Agile Toolkit**. Addison-Wesley, 2003.
- Rubin, Kenneth, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison-Wesley Signature Series, 2012
- Schwaber, Ken. **Agile Project Management with Scrum**. Microsoft Press, 2004.
- Tabaka, Jean. **Collaboration Explained: Facilitation Skills for Software Project Leaders**. Addison-Wesley, 2006.

Scrum Glossary

- **Agile** the name coined for the wider set of ideas that Scrum falls within; the Agile values and principles are captured in the Agile Manifesto
- **Chicken** (obsolete) term for anyone not on the team
- **Daily Scrum** a fifteen-minute daily team event to share progress, report impediments and inspect & adapt
- **Done** also referred to as "Done" or "Done Done", this term is used to describe a product increment that is considered releasable; it means that all design, coding, testing and documentation have been completed and the increment is fully integrated into the system
- **Emergence** the principle that the best designs, and the best ways of working come about over time through doing the work, rather than being defined in advance, cf. Empiricism, Self Organization
- **Empiricism** the principle of "inspect and adapt" which allows teams or individuals to try something out and learn from the experience by conscious reflection and change, cf. Emergence, Self Organization
- **Epic** a very large user story that is eventually broken down into smaller stories; Epics are often used as placeholders for new ideas that have not been thought out fully. There's nothing wrong with having an Epic, as long as it is not high order
- **Estimation** the process of agreeing on a size measurement for the stories in a Product Backlog. Done by the team, perhaps using Planning Poker
- **Impediment** anything that prevents the team from meeting their potential (e.g. build servers are down). If organizational, it is the Scrum Master's responsibility to eliminate it. If it is internal to the team, then they themselves should do away with it
- **Impediment Backlog** a visible list of impediments in a priority order according to how seriously they are blocking the team from productivity. Not a core Scrum concept.

Scrum Glossary

- **Pig** (arch.) term for a team member, the term offended some people so is now rarely used
- **Planning Poker** a game used to apply estimates to stories; it uses the Delphi method of arriving at consensus
- **Product Backlog** a prioritized list of features or stories that are waiting to be worked on
- **Product Backlog Item** any feature that is on the backlog list, which will include user stories, Epics and possibly technical stories to deal with technical debt, etc.
- **Product Owner** person whom holds the vision for the product and is responsible for maintaining, ordering and updating the Product Backlog
- **Release Burndown Chart** a visible chart to show progress towards a release
- **Retrospective** a session where the Team and Scrum Master reflect on the process and make commitments to improve
- **ScrumMaster** (or Scrum Master) a servant leader to the team, responsible for removing impediments and making sure the process runs smoothly so the team can be as productive as possible
- **Scrum events** Planning, Review, Retrospective, Daily Scrum
- **Scrum Roles** there are only three: Product Owner, Scrum Master, team
- **Spike** a short, time-boxed piece of research, usually technical, on a single story that is intended to provide just enough information that the team can estimate the size of the story
- **Sprint** a time boxed iteration
- **Sprint Burndown** a visible chart that indicates on a daily basis the amount of work remaining in the Sprint
- **Sprint Goal** aka Sprint Theme, the key focus of the work for a single Sprint
- **Sprint Planning** an event primarily involving the Team and the Product Owner to plan the Sprint and arrive at an agreement on the forecast / commitment
- **Sprint Task** a single small item of work that helps one particular story reach completion

Scrum Glossary

- **Stakeholder** anyone external to the team with an interest in the product being developed
- **Story** a backlog item usually using the template form: as a [user] I want [function] so that [business value]
- **Story Point** a unit of measurement applied to the size of a story, cf. Fibonacci Sequence
- **Story Time** the regular work session where items on the backlog are discussed, refined and estimated and the backlog is trimmed and prioritized
- **Task** see Sprint Task
- **Task List** the tasks needed to complete the set of stories committed to a Sprint
- **Taskboard** a wall chart with cards and sticky notes that represent all the work of a team in a given Sprint; the task notes are moved across the board to show progress
- **Team** the Development Team, responsible committing to work, delivering and driving the product forward from a tactical perspective
- **Team Member** any member of the team, including developers, testers, designers, writers, graphic artists, database admins...
- **Timeboxing** setting a duration for every activity and having it last no more than that (i.e. neither events nor Sprint are ever lengthened)
- **Velocity** the rate at which a team completes work, usually measured in story points.
- **Vision Statement** a high-level description of a product which includes who it is for, why it is necessary and what differentiates it from similar products
- **XP Practices** the set of development practices, including pair-programming, test-first, or test-driven development (TDD) and continuous refactoring, which are drawn from the XP methodology; many Scrum teams find these practices greatly improve productivity and team morale