

A Brief Introduction to Scrum

An Agile Methodology

by

Craig D. Wilson
MS, PMP, CSM

MATINCOR, INC.

Information Technology Management Consulting

Presentation Outline

- Introduction to Scrum
- Origins of Scrum
- Definitions & Principles
- Benefits & Risks

Caveats & Disclaimers

- Focused on “core” Scrum as defined in “ScrumGuide” by Ken Schwaber and other introductory Scrum texts
- Presentation is a summary of a two-day class – all topics are touched upon but summarized and simplified
- Scrum and the Scrum community are evolving - many weaknesses have been or are being addressed

An “agile” methodology

Supports the Agile Manifesto:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

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

Agile Methodologies

Extreme Programming

MSF for Agile Software Development

Crystal

Scrum

• Feature Driven Development

Adaptive Software Development

Dynamic Systems Development Method

Why Talk About Scrum?

- Popular
- Powerful
- Easy to learn

But.....

misunderstandings abound

“All models are wrong, some are useful.....”

George Box, industrial statistician

Popularity of the Scrum Model

- Basic principles are easy to understand
- Technology and tool agnostic
- Built on several time-tested techniques
- Utilizes team-of-peers management approach

History of Scrum

- Inspired from approach defined in 1986 by H. Takeuchi and I. Nonaka
- Term “scrum” used in “Wicked Problems, Righteous Solutions” by DeGrace and Stahl in 1991
- Used as a methodology name in the book “Agile Software Development with SCRUM” by K. Schwaber and M. Beedle published in 2001.

A Rugby Scrummage



Definition of Scrum

“Scrum...is a framework within which you can employ various processes and techniques...within which complex products can be developed”

-Ken Schwaber,
ScrumGuide, May 2009

Scrum Principles

- Time-boxes
- Cross-functional teams
- Open communications
 - Within team
 - With stakeholders
- Priorities set by Product Owner
- Demonstrable results
- Responsive to change

The Process

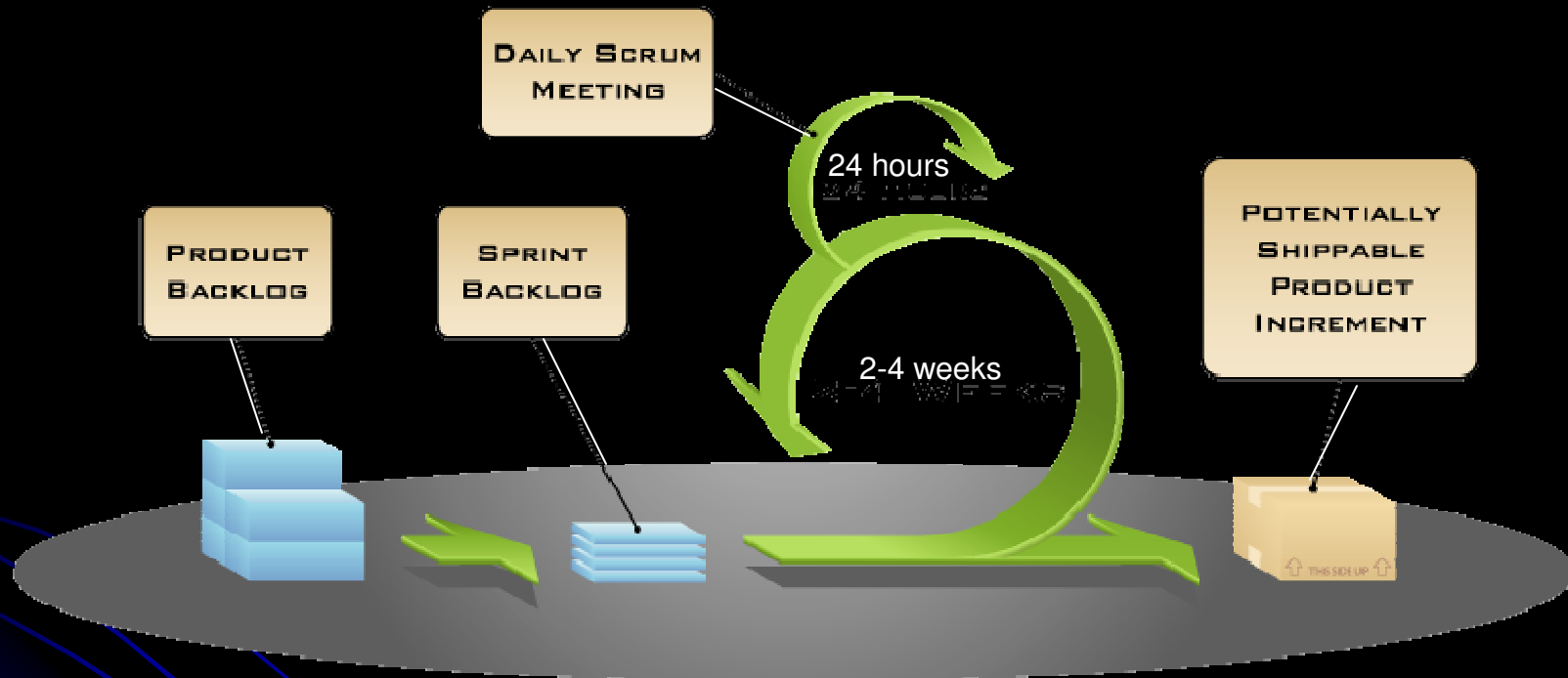


Image used by permission
Mountain Goat Software
Copyright 2005

Scrum Terms

- Team
 - ScrumMaster
 - Scrum Team
 - Product Owner
 - Users & Stakeholders
- Sprint
- Backlog
 - Product
 - Sprint
- Meetings
 - Daily scrum
 - Planning
 - Review
 - Retrospective
- Burndown
 - Chart
 - Velocity

The Team

- ScrumMaster
 - Not “command & control” project manager
 - Process coach and team facilitator
 - Remover of roadblocks
- Scrum Team
 - Individuals responsible for the Sprint results
 - Mix of skills representing multiple disciplines
 - Usually 6-8 individuals
- Product Owner
 - Individual responsible for product
 - Responsible for product profitability (ROI)
 - Adjusts feature list and priorities for each Sprint
 - Accepts or rejects work results
- Users & Stakeholders
 - Interested in results but not responsible for deliverables

The Sprint

- Time boxed effort
 - Usually 2 weeks to 1 month
 - Can be longer or shorter
- Defined workload
 - No changes once Sprint is begun
 - If workload changes, Sprint restarted
- Begins with Planning Meeting
- Ends with demonstrable Release

Product Backlog

- All features and functions for final product
- May be subdivided into releases
- Prioritized by Product Owner
- Initial backlog is established and releases are defined prior to start of Sprints
- Product backlog is reviewed and updated throughout the project
- Time must be allotted during Sprints to allow for this activity

Sprint Backlog

- Features and functions targeted for a single Sprint
 - Frequently broken down into User Stories; especially if function is complex
- May include technical requirements or objectives
 - e.g.; database design, UI standards, architecture documentation

Meetings

- Sprint Planning
 - Sprint goal and functionality objectives
 - Sprint tasks identified
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum meetings are time-boxed and occur on a regular schedule!

Sprint Planning

- Product Owner and Team
- Review of Product Backlog
- Product Owner provides definition and details of features and functions
- Negotiation of what will be in Sprint
- May identify new Product Backlog needs
- Results in Sprint Goals

Sprint Task Definitions

- Sprint Team meeting
- Immediately follows Sprint Planning
- Breaks work into tasks
 - 4-16 hours of effort each
 - Identifies interdependencies
- Results in Sprint Backlog
- Tasks are prioritized by team

Daily Scrum

- Standup 15 minute meeting
- Each team member answers 3 questions:
 - What have you done since the last meeting?
 - What will you do before the next meeting?
 - What is preventing you from accomplishing your tasks?
- For benefit of other team members
 - Not a “status report” to the ScrumMaster

Sprint Review

- Results of Sprint are demonstrated to Product Owner
- Owner accepts or rejects results
- Results are input to:
 - Next Sprint Planning meeting
 - Sprint Retrospective

Sprint Retrospective

- Meeting with Sprint Team
 - May include Product Owner
- Process review and modification
- Lessons learned applied in following Sprints

Scrum Estimation Technique

- Team effort
 - Only those doing the work
- Real time Delphi method
- Evaluate relative “effort size” of functions
 - Factors may include complexity, effort, uncertainty
- Assign relative “work effort value” to each function
- Track progress against effort estimate

Delphi Process

- Developed by Rand Corporation in late 1940's
- Documented in software development in 1970's by Barry Boehm and John Farquhar
 - Defined as "Wideband Delphi" due to more interactive discussions
- In Scrum
 - Team members evaluate and compare a list of functions
 - Next, compare opinions of relative effort required
 - Work to an agreement on estimates
 - Occurs during backlog review and updates

Relative Estimation

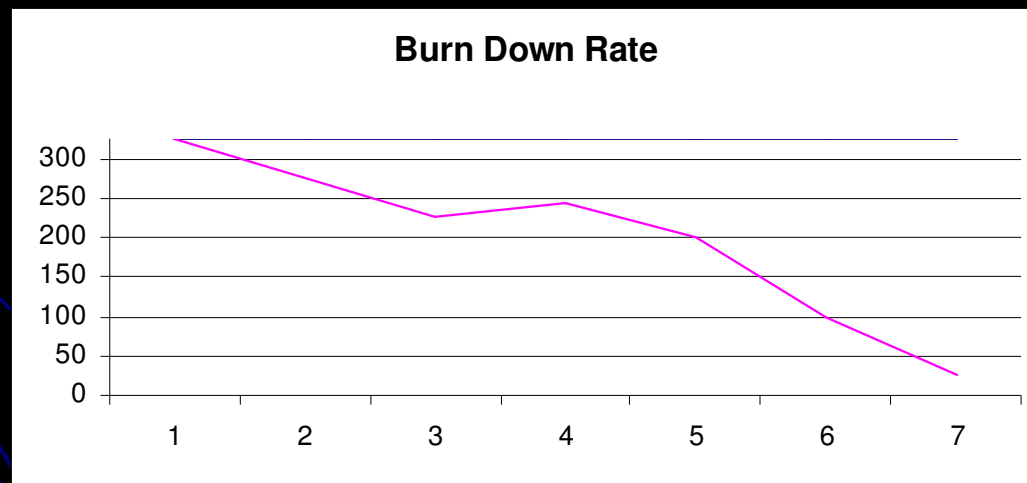
- Define relative complexity & effort
 - Much larger, larger, equal, smaller, much smaller
 - Several “fun” techniques (Number of pizzas, T-shirt sizes, buckets, Planning Poker®)
- Assign numeric value to each category
 - Numbers have no intrinsic value; only relative value
- Estimators discuss results and continue re-estimating until everyone in agreement
- Results in work points

Using Work Points

- At end of Sprint total work points achieved by adding estimates for all accepted functions
- Track number of work points earned in each iteration to determine
 - Product Backlog Burndown
 - Velocity
- Difficult to carry work point estimations over to other projects
 - Different teams, tools, technologies, etc.

Burndown

- Measurement of accomplishments
 - Product Backlog Burndown
 - Sprint Task Burndown
- Burndown Chart



Velocity

- Compare Velocity to total Estimated Work Points for Product Backlog to estimate project duration
- Need several Sprints to determine team's velocity
- Same technique is used to estimate Sprint Burndown
 - Track tasks instead of functions

Benefits of Scrum

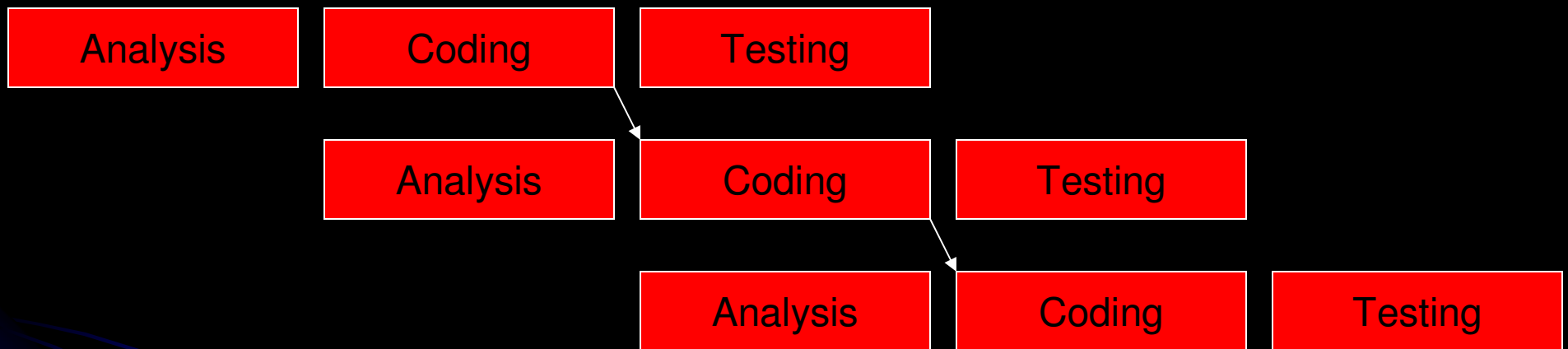
- Targets Product Owner's functions-of-value
- Focus on team communications
 - Frequent and ready access to knowledge
 - Co-location improves communications
- Frequent demonstrations for early feedback from stakeholders
- Team spirit and camaraderie
- Sense of accomplishment
- Quality of product

But keep the following in mind.....

Sprint Process

- Sprint is not a “mini-waterfall”
- Must result in quality, demonstrable function(s) of value to Product Owner
 - Beware of defect build-up (aka technical debt)
- Sprints will include requirements clarification, development, and testing
 - Sprints may include architectural design
 - Full regression testing may parallel next Sprint

A Phase Is Not A Sprint



Time-boxed coding phases following on the footsteps of each other is not Scrum and violates several principles of the methodology.

Team Roles – ScrumMaster

- Lacks many Project Manager responsibilities as defined by Project Management Body of Knowledge
 - Someone needs to perform these responsibilities
 - May be PM overseeing several related Scrum teams
- Lacks authority given to some Project Managers – which may be needed on large scale or difficult projects

ScrumMaster Certification

- Currently achieved by attending 2-day lecture provided by Certified Scrum Trainer
- Effective October 1, 2009 must also pass a Certified ScrumMaster online certification exam

Team Roles – Product Owner

- Role filled by prime user or sponsor
 - Responsible for resulting product and its ROI
- Role may be supported by Business Analyst
 - Representing the interests of users and stakeholders
 - Must be careful not to become a wall between users and Sprint Team
 - Should be a “communication enabler”
 - Facilitate communications between users and team

Team Roles – Team Members

- Choose their own tasks
 - Not assigned by Scrum Master
 - Works better with a “process mature” team
- Beware “task hogs” who
 - Take on more than they can handle
 - Grab the best tasks for themselves
- Some may perceive themselves as filling a traditional role rather than co-owner of Sprint Release

“I’m a QA guy. Call me when you’re ready to test.....”

Intense Iterations

- Full team is “always on” during a Sprint
- Must be cautious of team burnout
- Limit overtime
- Set a sustainable pace

Project Duration Estimates

- In order to estimate project duration you need to know
 - Full inventory of Product Backlog
 - Know enough about each function to perform estimation technique
- Address this by using Product Releases
 - Break large projects in to several production releases
- Do not rely solely on the Delphi estimation technique for project duration estimates
 - Experience and common sense should not be ignored

Product Engineering

- Scrum does not address product engineering
 - For software development you will need a software engineering process such as Object Oriented Analysis & Design
- May benefit by adding methods that are more focused on software creation
 - Extreme Programming (XP)

Target Fixation

- Beware team target fixation
 - If goal is velocity and burn-down, quality could suffer
- Focus may be on developing to requirements and a miss on getting the requirements right
 - Product owners may not always know “right” requirements. Still need effective research, analysis, process re-engineering, etc.

Product Backlog

- In “core” Scrum, this is very poorly defined
 - Assumes Product Owner has, or can, fully define the requirements
 - Does not address requirements discovery, non-functional requirements, or requirements analysis
- Product Backlog requires on-going “grooming” and this activity must be part of Sprints
 - Breakdown functions into User Stories
 - Provide greater detail for upcoming Sprints

Suggestions For Further Study

- Scrum Alliance www.scrumalliance.org
- Mountain Goat Software *Mike Cohn is the founder and great presenter on the topic!* www.mountaingoatsoftware.com
- 30-Day Blitz *Another time-boxed methodology:* [www.michaelhugos.com/30-day Blitz.html](http://www.michaelhugos.com/30-day_Blitz.html)
- Case study: “Issues and Challenges of Agile Software Development with Scrum” by Juyun Cho, Colorado State University

More presentations by Craig D. Wilson
www.matincor.com