

Scrum Fundamentals

Do It Right

Mike Vincent

MVA Software

mikev@mvasoftware.com



Mike Vincent

- Architect, ALM and Agile Coach
- Over 24 years as software developer and architect
- Marketing director, construction project manager and structural engineer previously
- Worked with Visual Studio Team System since initial beta releases
- TFS implementation and training experience in corporate and public organizations
- Microsoft MVP - Visual Studio ALM
- Passion for community
 - INETA
 - IASA
- Professional Scrum Developer Trainer
- Professional Scrum Product Owner



Agile / Scrum Sessions

- VAS01: Scrum Fundamentals
 - Do It Right
- VAS02: Top 10 Dysfunctions with Scrum
 - Don't Shoot the Messenger
- VAS03: The Scrum Product Owner
 - Big Responsibilities
- VAS04: Herding Pigs
 - Managing Self Organizing Teams

What you will learn

- What is Scrum
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why you should care
- Scrum for Visual Studio 2010 ALM

Assumptions

- You want to learn about Scrum
 - It is new to you, so start from the beginning
 - You are using it and ...
 - you want to know if you are doing it right
 - you want to learn more
- You want to improve ...
 - Get your organization on board with Scrum
 - Make your ALM process better
 - Manage project risk
- You are here to learn
 - This is not a debate

Scrum Fundamentals

- Scrum is incredibly simple on the surface
- But touches profoundly on people, process and technology
 - What should you expect if you do it right?
 - And what if you don't?
 - It all starts with understanding the Scrum Fundamentals

Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

Scrum Fundamentals

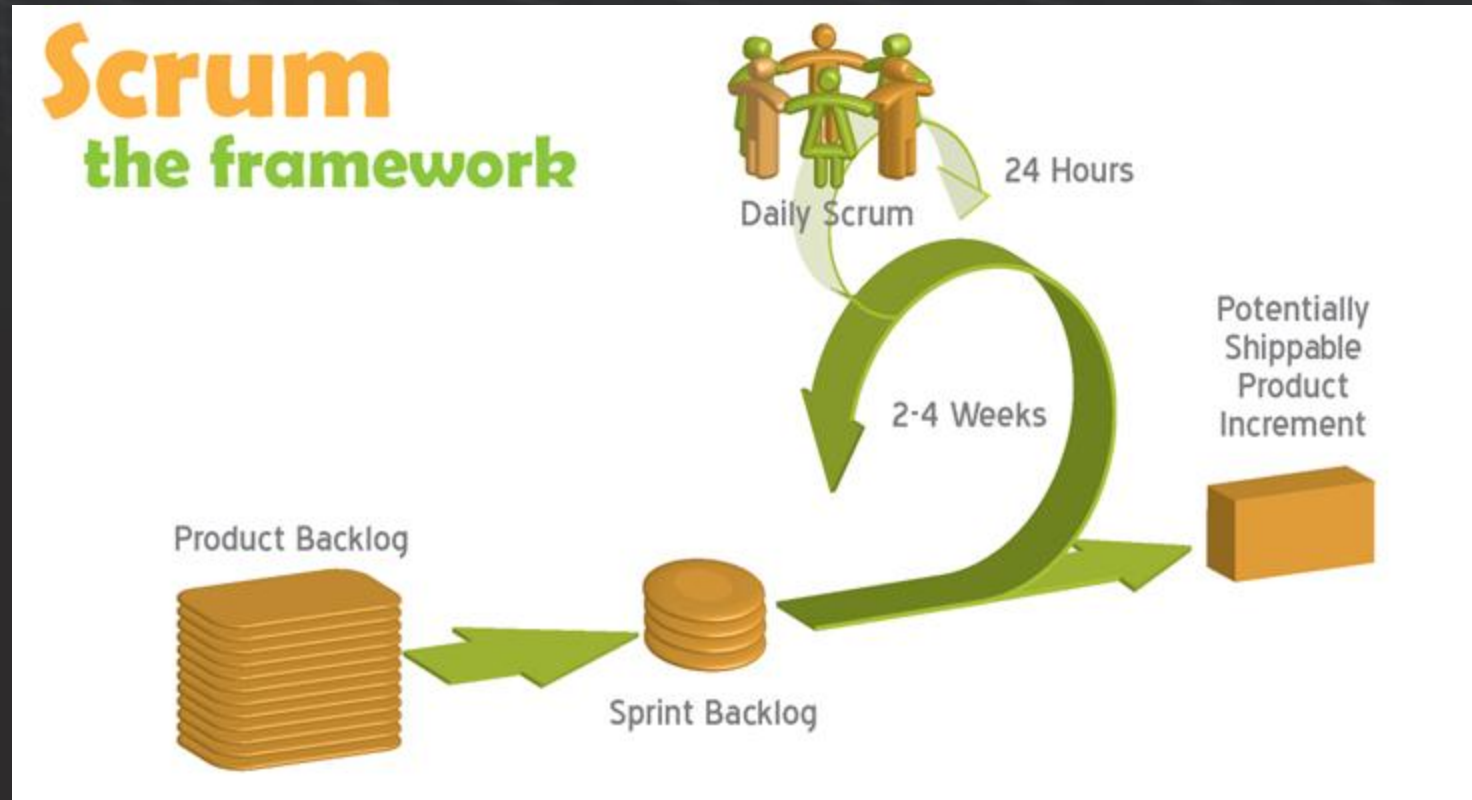
- **What is Scrum?**
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

Scrum

- **Scrum is a *framework***
 - For managing the development and deployment of complex products
- **It is not a process or a technique**
 - But can employ processes and techniques
- **Scrum exposes what isn't working well**
 - So you can fix and improve



The Scrum Framework



- The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules.
- Each component within the framework serves a specific purpose and is essential to Scrum's success and usage

Scrum

- Created in the early 1990's by Ken Schwaber and Jeff Sutherland
- Co-presented Scrum as a formal practice at OOPSLA in 1995
- In 2011, 88% of teams identifying themselves as “agile” use Scrum

- Forrester Research

Scrum

- Is grounded in empirical process control theory
- Employs an iterative, incremental approach to optimize predictability and control risk
- It is dependent on frequent inspection and adaptation to reach goals
- Inspection is dependent on transparency
- Uses self-managing teams that are cross-functional
- Scrum generates increments that are done and functional

What is Agile?

- Agile Software Development is an *umbrella term* for approaches to software development that follow the principle of “*Inspect and Adapt*” and advocate *team empowerment*
- “Agile” approaches emerged concurrently from a number of leading thinkers who were successfully delivering software with “lighter” methods in the 1990s
- Out of a meeting at Snowbird in 2001 to find commonality between their approaches came the *Agile Manifesto*
- Agile methods include DSDM, Extreme Programming (XP), Scrum, FDD, ...

The Agile Manifesto

While there is value in the items on the right, we value the items on the left more

- ✓ *Individuals & Interactions* over *Processes & Tools*
- ✓ *Customer Collaboration* over *Contract Negotiation*
- ✓ *Responding to Change* over *Following a Plan*
- ✓ *Working Software* over *Comprehensive Documentation*

Agile Manifesto Principles 1 of 2

1. Satisfy the customer
2. Welcome changing requirements
3. Deliver working software frequently
4. Work together
5. Motivated individuals
6. Face to- face conversation

Agile Manifesto Principles 2 of 2

7. Working software

8. Sustainable development

9. Technical excellence

10. Simplicity

11. Self-organizing teams

12. Reflect

Pigs and Chickens



By Clark & Vizdos

© 2006 implementingscrum.com

Scrum Fundamentals

- What is Scrum?
- **The roles**
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

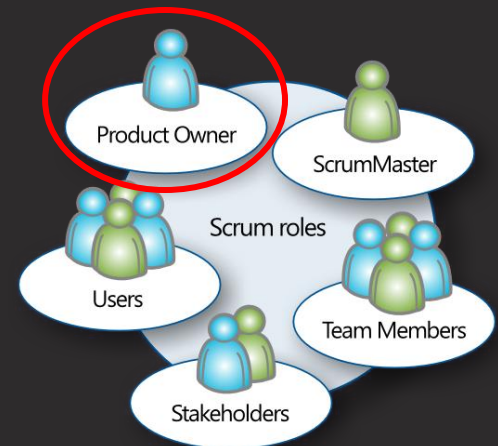
The Scrum Team Roles

- Product Owner
- Development Team
- Scrum Master

Roles and Responsibilities

- **Product Owner**

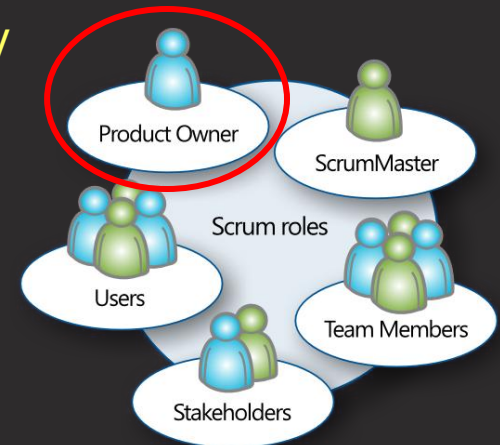
- Maximizes the value of the Product
 - Ensures the worth of the work the Team performs
- Creates and maintains the Product Backlog
 - Makes sure it is clear, understandable and has sufficient detail
 - Ensures that it is visible, transparent to everyone
- Chooses what and when to release
 - Optimizes delivery of completed functionality
- Represents stakeholders and customers to the team
 - Defines the vision and features of product
 - Maintains frequent communication and availability with both stakeholders and team



Roles and Responsibilities

- **Product Owner**

- The Product Owner is the one and only person responsible for managing the Product Backlog
 - Ensures the value of the work the Team performs
- Everyone knows what items have the highest priority
 - So everyone knows what will be worked on
- The Product Owner is one person, not a committee
 - Committees may exist that advise or influence this person, but people who want to change an item's priority have to convince the Product Owner

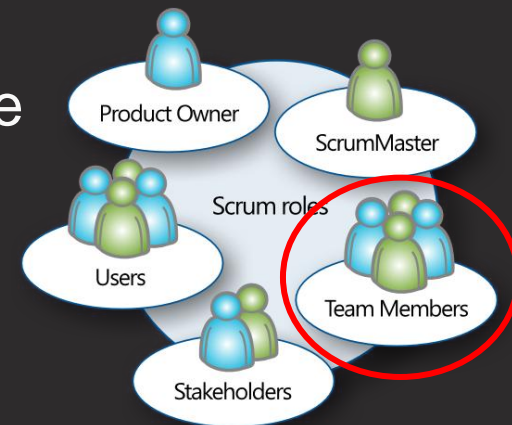


Companies that adopt Scrum may find it influences their methods for setting priorities and requirements over time

Roles and Responsibilities

- **Development Team**

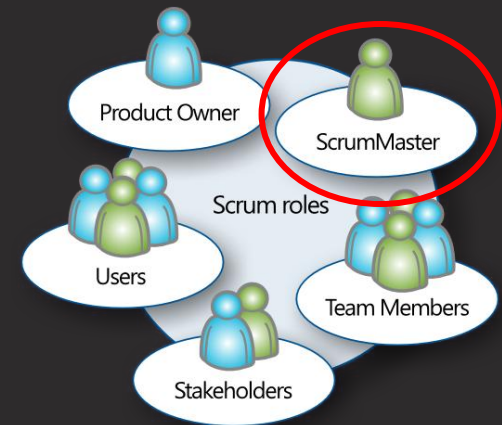
- Turns Product Backlog into increments of potentially shippable functionality every Sprint
- Selects the iteration goal and specifies work results
- Commits to what it feels it can accomplish
- Has authority to do everything within existing standards and guidelines to reach the iteration goal
- Manages itself and its work
- Collaborates with Product Owner to optimize value
- Demos work results to the Product Owner



Roles and Responsibilities

- **Scrum Master**

- Ensures that the team is fully functional, productive and improves quality
- Enables close cooperation across all roles and functions and removes barriers
- Shields the team from external interferences
- Ensures that the process is followed
- Teaches Product Owner and Team how to fulfill their roles



Roles and Responsibilities

- **Scrum Master**

- Is responsible for ensuring that the Organization and Team adheres to Scrum values, practices, and rules
- Helps the Scrum Team and the organization adopt Scrum
- Teaches the Team by coaching and by leading it to be more productive and produce higher quality products
- Ensures the Team is functional and productive
- Helps the Team understand and use self-management and cross-functionality
 - However, the Scrum Master does not manage the Development Team; the Team is self-managing

Scrum Fundamentals

- What is Scrum?
- The roles
- **The players**
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

The Scrum Team Players

- Product Owner
- Development Team
- Scrum Master

Product Owner

- May be a product manager
- For in-house development efforts, could be the manager of the business function that is being automated
- May be an executive
- May be a customer
 - Must understand Scrum
- The Product Owner can be a Team member, also doing development work
 - This additional responsibility may cut into the Product Owner's ability to work with stakeholders
- The Product Owner can never be the Scrum Master



The Development Team

- **Teams are self-organizing**
 - No one tells the Team how to turn Product Backlog into increments of shippable functionality
- **Teams are cross-functional**
 - There are no sub-teams dedicated to a particular domain like testing or business analysis
- **No titles on Scrum Teams**
- **Everyone chips in**
 - Team accountability

The Development Team

- Optimal size for a team is seven people, plus or minus two
 - Doesn't include Product Owner or Scrum Master unless they are also developers
- Team composition can change at the end of a Sprint
 - But, be careful about diminishing productivity gains

Scrum Master

- **May be a manager or executive**
 - Must have appropriate servant-leader skills
- **May be a member of the Development Team**
 - For example, a developer performing Sprint tasks
 - However, this often leads to conflicts when the Scrum Master has to choose between removing impediments and performing tasks
- **The Scrum Master can never be the Product Owner**
- **May be fired by the Team**

Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- **The rules**
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

Sustainable Pace

- Teams work at a sustainable pace
 - Typically 40 hour weeks
 - Productivity higher
 - Quality higher
 - Lot of creative thinking in “off hours”
- No “death marches”

Documentation Plan

- Documentation

- Just enough to communicate with the developer – it's what you have to say and what you don't have to say
 - Overall style guide
 - Light wireframes
 - High level use cases
 - Real detailed test plans (definition of done)
 - Pass
 - states Fail
- The only deliverable that should count is the final product

Definition of Done

- Team together with product owner defines what “done” means
- Done defines the current technical capability of the team
 - Over time Done should include everything needed before deployment
 - Not done backlog items should not be demonstrated at Sprint Review

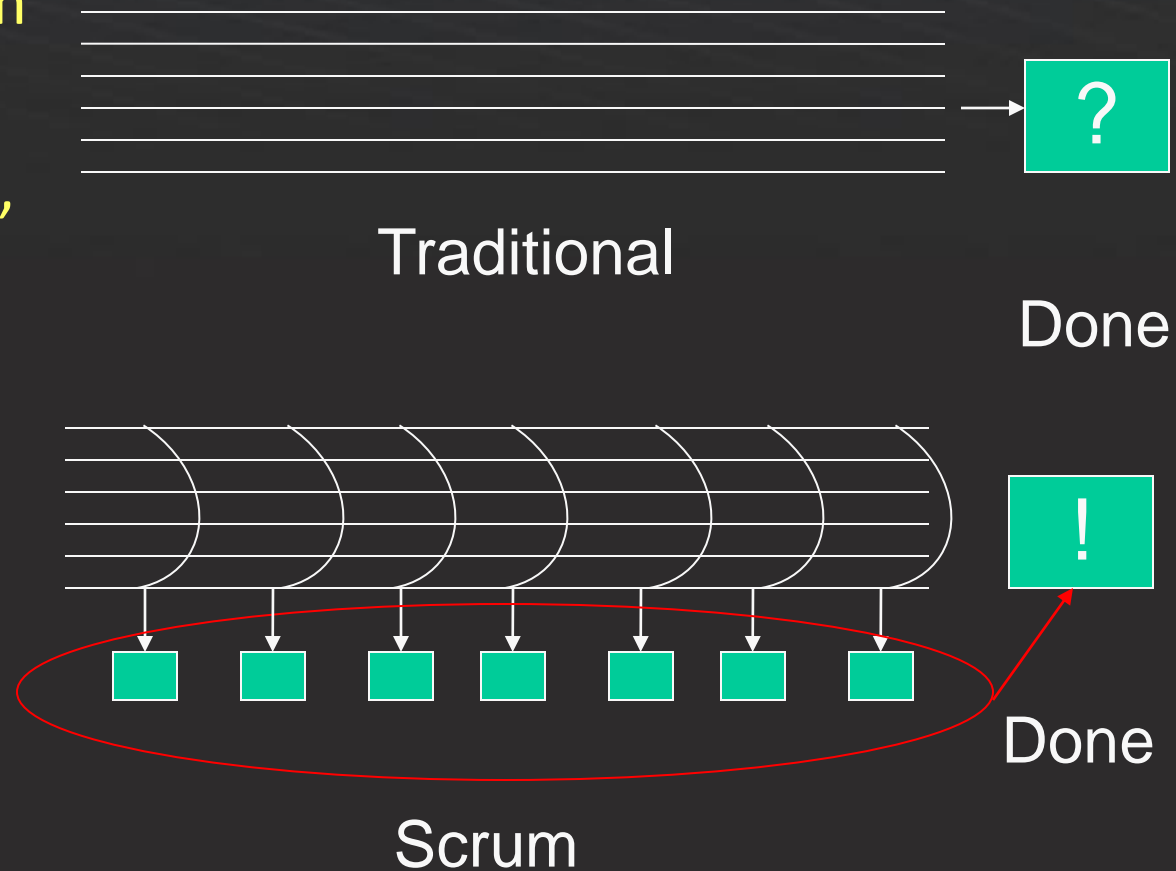


Done

- ✓ Designed
- ✓ Refactored
- ✓ Coded
- ✓ No clever techniques
- ✓ Code review
- ✓ Design review
- ✓ Unit tested
- ✓ Functional tested
- ✓ Unit test harness
- ✓ Integration tested
- ✓ Regression tested
- ✓ Performance tested
- ✓ Security tested
- ✓ User Acceptance tested

Done

- Development team must be able to produce a completely “done” piece every Sprint
- Product Owner must inspect and adapt to optimize ROI every Sprint
- “Undone” work must be identified



Time-Boxes

- The Time-Boxes in Scrum are the:
 - Release Planning Meeting
 - Sprint
 - 2 – 4 weeks
 - Sprint Planning Meeting
 - What and how, each 4 hours for one month sprints
 - Daily Scrum
 - 15 minutes Stand up
 - Sprint Review
 - 4 hours for one month sprints
 - Sprint Retrospective
 - 3 hours

Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- **The process and ceremonies**
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

Scrum Process

Release
Planning

Sprint

Retro-
spective

Sprint
Planning

Daily
Scrums

Sprint
Review

Release Planning Meeting

- Establish plan and goals on how to turn a vision into a product
- Don't plan the entire product
 - Additional just-in-time planning will be performed
- Outcomes
 - Overall product features and functionality
 - The goal of the release
 - Major risks identified
 - Highest priority product backlog

The Sprint

- Sprint = iteration
- It is the time when the team achieves their Sprint goal
 - Turning requirements into an increment of potentially shippable functionality
- Sprints are time-boxed
 - Typically 2, 3 or 4 weeks
 - Beyond 4 weeks is a smell



The Sprint

- Consists of the
 - Sprint planning meeting
 - Development work
 - Sprint Review
 - Sprint retrospective
- Occurs one after another, no time in between



The Sprint

- If the Team senses that it has overcommitted
 - Meets with the Product Owner to reduce scope of the Product Backlog selected for the Sprint
- If the Team senses that it has extra time
 - Meet with the Product Owner to select additional Product Backlog
- Sprints can be cancelled before the Sprint time is over
 - Only the Product Owner has the authority to do this

Sprint Planning Meeting

- Time boxed to 8 hours for a one month Sprint
- The whole Scrum Team attends
 - 1st 4 Hour time box – What will be done
 - Product Owner presents top priority Product Backlog to Team
 - Team assesses what it can accomplish
 - Sprint goal is crafted

Sprint Planning Meeting

- Time boxed to 8 hours for a one month Sprint
 - 2nd 4 hour time box – How will we do it
 - Team figures out how it will turn the Product Backlog selected into a done increment
 - Tasks are decomposed so they can be done in less than one day – list is the Sprint Backlog
 - Team self organizes to assign and undertake the work
 - Usually only 60-70% of the Sprint Backlog will be devised in the Sprint Planning meeting, the rest is stubbed out for later detailing during the Sprint
 - The Product Owner is present for questions and clarification

Daily Scrum

- Daily 15 minute status meeting
- Team stands in a circle facing each other
- Each team member answers 3 questions:
 1. What have you done since the last Scrum?
 2. What will you do between now and the next Scrum?
 3. What is in your way?
- For synchronization
 - Not problem solving!
 - Pigs, not chickens
- Not a Scrum Master status meeting
 - Commitments between peers



Sprint Review

- Review sprint goal and forecasted/committed PBIs
 - Demo completed features and collect feedback
 - Describe and discuss uncompleted work
- The Team presents, not the Scrum Master
- Informal
 - Minimal preparation
 - e.g. Room logistics
 - Avoid slides
- Whole team participates
- Invite anyone and everyone
- Reviews are not PR exercises



Sprint Retrospective

- **Look back**
 - What did we do well?
 - What didn't we do well?
- **Look forward**
 - Generate actions for the next sprint
 - Tasks for next backlog
 - Add to a wall chart in team area
- **Is our Product Owner Happy?**
- **Metrics**
 - Look at key reports
- **Whole team participates**
- **Scrum Master facilitates**



Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- The process and ceremonies
- **The artifacts**
- Getting started
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Product Backlog

- **Prioritized list of product or project requirements**
 - Expressed in business language
 - Prioritized at beginning of each Sprint
- **Ideally expressed such that each item has value to the user or customers of the product**
- **One person (Product Owner) is responsible for prioritization and making sure requirements are well formed**
 - Anyone can contribute items for the backlog
 - Public and available
- **High level estimates**
 - Size only - can use Story Points
 - Planning Poker

Product Backlog Item

- The increment of deliverable work
- Contains clear acceptance criteria
 - Criteria for test and successful completion
 - From Team or Product Owner
- May reference other artifacts like:
 - Specifications, Mockups, Architecture Models
- Sized appropriately
 - May be completed within a single Sprint
 - Typically with a few other PBIs
- Managed by the Product Owner
 - Appropriate detail just in time

Sprint Backlog

- The Sprint Backlog is a list of tasks that defines a Team's work for a Sprint
- Tasks emerge during Sprint planning
 - And during the Sprint as more information emerges
- The tasks are what the Team has defined as being required to turn committed Product Backlog items into system functionality
 - Task size: ideally no more than a day
 - Estimated as a team

Sprint Backlog

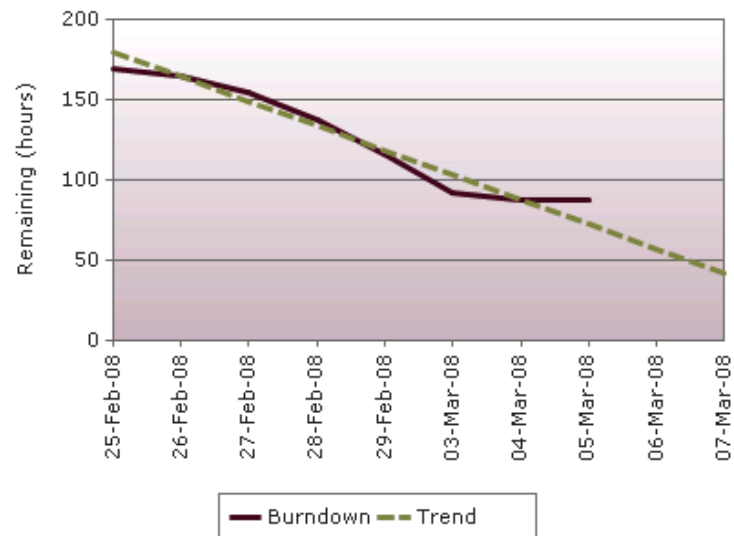
- The Sprint Backlog should be updated daily
 - After the Scrum Meeting works well
 - Or when a task is completed
 - By the team
- Anything that gets done should be visible on the backlog
 - Emergent tasks are added by the team to the backlog throughout the Sprint

Monitoring Progress

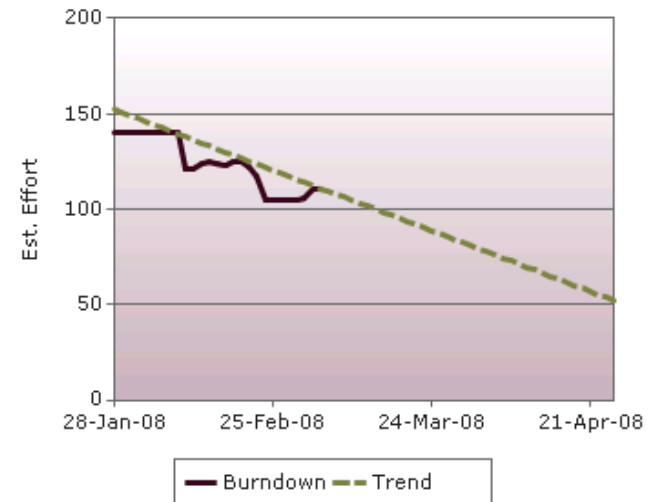
- **Burndown Charts**
 - A burndown is a measure of remaining backlog over time
 - A Sprint Burndown measures remaining Sprint Backlog items across the time of a Sprint
 - A Release Burndown measures remaining Product Backlog across the time of a release plan

Burndown Charts

Sprint Burndown Chart



Product Burndown Chart



Burndown charts are no longer a formal Scrum artifact

Increment

- The increment is the sum of all PBI's created in the Sprint
 - And all previous Sprints
- Is *usable* and it works
 - And is integrated with previous work
- Is *potentially* shippable
- Must be **DONE**
 - As per team standards
 - With no work remaining

Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- **Getting started**
- Organizational commitments
- Why should I care?
- Scrum for Visual Studio 2010 ALM

How do I get Started?

- Understand the Fundamentals of Scrum
- Begin with a target project and team
- Start
- Don't get hung up on big planning up front
- Learn from initial projects
- Adapt to mainstream projects

Don't Change Scrum

- Don't be tempted to change Scrum
- Scrum exposes inefficiency
 - Fix the problem
 - Don't shoot the messenger!

Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- **Organizational commitments**
- Why should I care?
- Scrum for Visual Studio 2010 ALM

Scrum Exposes Need for Change

- When adopting Scrum, organizational changes will be needed
- Remember 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
- Measurement will change
 - Burndown rather than “productivity indices”
- New people management
 - Facilitating rather than checking

Scrum's Impact on Culture

- **Empirical management replaces predictive management.**
 - The art of the possible replaces the mandate of the desired.
 - The desire to be certain is replaced by controlled risk.
- **Transparency is value neutral.**
 - Waste, impediments, and dysfunctions are highlighted along with progress.
 - Transparency disables politics.
- **Authority moves down the organization.**
 - Scrum Teams are self-organizing and self-managing.
 - Accountability is specific
- **More attention and hard choices are required.**
 - What if the project isn't delivering what is needed for an acceptable cost?

Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- **Why should I care?**
- Scrum for Visual Studio 2010 ALM

Why you might care about Scrum

Current

- Releases take longer and longer
- Release schedules slip
- Stabilization at end of release takes longer and longer
- Planning seems to take too long
- Changes are hard to introduce mid-release
- Quality is deteriorating
- Dissatisfied, alienated customers
- Death marches are hurting morale
- Products becoming prohibitively expensive to support and enhance

Scrum

- Agility – flexibility to turn on a dime
- Improved productivity and quality
- Early elimination of risk
- Early realization of value
- Always knowing exactly where you are in a development/deployment cycle
- Easier to make changes
- Reduced waste
- Lean products that reach market faster and are more targeted
- Increased Return on Investment
- Improved relationship with customers
- Engaged, empowered workers
- Reduced Total Cost of Ownership

Framework for Success

Delivering a steady flow of successful and valuable software, by

1. Empowering collaborative, trusting and motivated teams
2. Responding to feedback and adapting to change
3. Building quality in from the start

Produce increments of potentially shippable functionality every sprint...



Why Should I Care

- Improve your processes
- Be more efficient
- Respond to your customers
- Stay ahead of the competition
- Enjoy your work



Scrum Fundamentals

- What is Scrum?
- The roles
- The players
- The rules
- The process and ceremonies
- The artifacts
- Getting started
- Organizational commitments
- Why should I care?
- **Scrum for Visual Studio 2010 ALM**

Scrum for Visual Studio ALM

- **Process templates in VS 2010**
 - Visual Studio Scrum, v 1.0
 - MSF Agile 5.0 (~80% Scrum)

And...

Don't forget the fundamentals



Resources for More Information

The Scrum Guide

The Definitive Guide to Scrum:
The Rules of the Game



Jeff Sutherland



Ken Schwaber

July 2011

Developed and sustained by Ken Schwaber and Jeff Sutherland

Scrum.org

- Read the Scrum Guide
- The Scrum Guide represents the official Scrum Body of Knowledge
- The Scrum Guide is hosted by [Scrum.org](https://www.scrum.org), which provides Scrum resources, training, assessments, and certification as part of the [Professional Scrum Master](https://www.scrum.org/certification/professional-scrum-master), [Professional Scrum Developer](https://www.scrum.org/certification/professional-scrum-developer) and [Professional Scrum Product Owner](https://www.scrum.org/certification/professional-scrum-product-owner) programs

Questions



Mike Vincent

MVA Software

mikev@mvasoftware.com

www.mvasoftware.net



Your Feedback is Important

Please fill out a session evaluation form
drop it off at the conference registration
desk.

Thank you!

