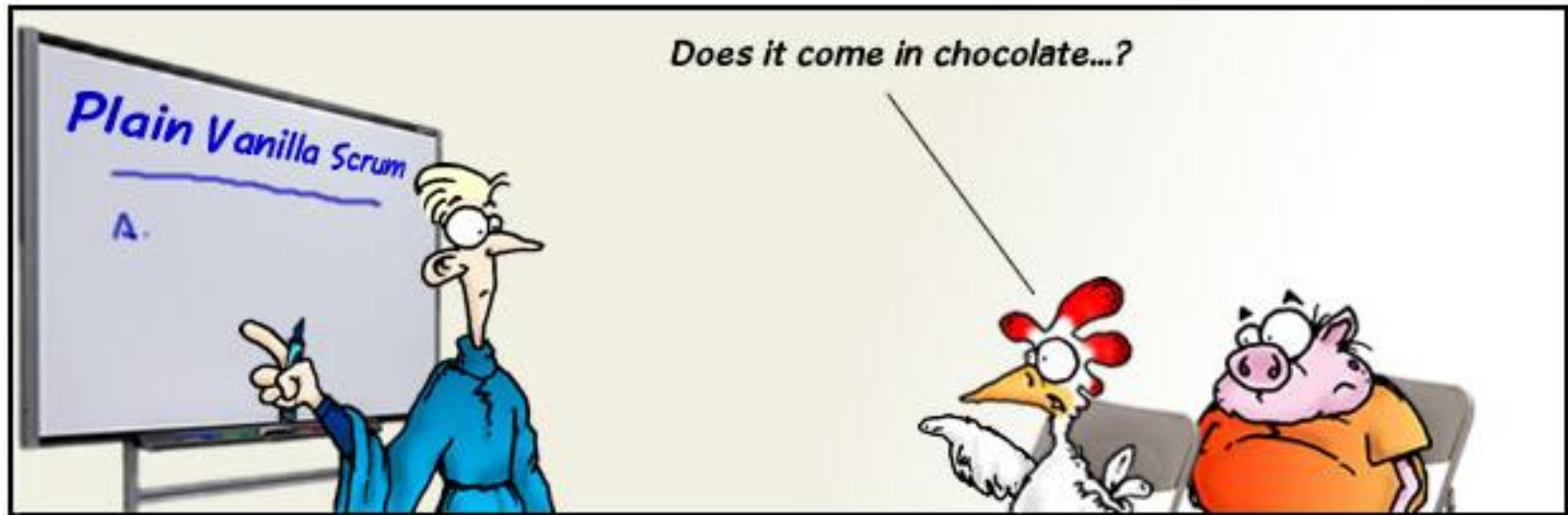


Student Scrums Workshop

Tom Reichlmayr
 Rochester Institute of Technology
 Department of Software Engineering
 tjr@se.rit.edu

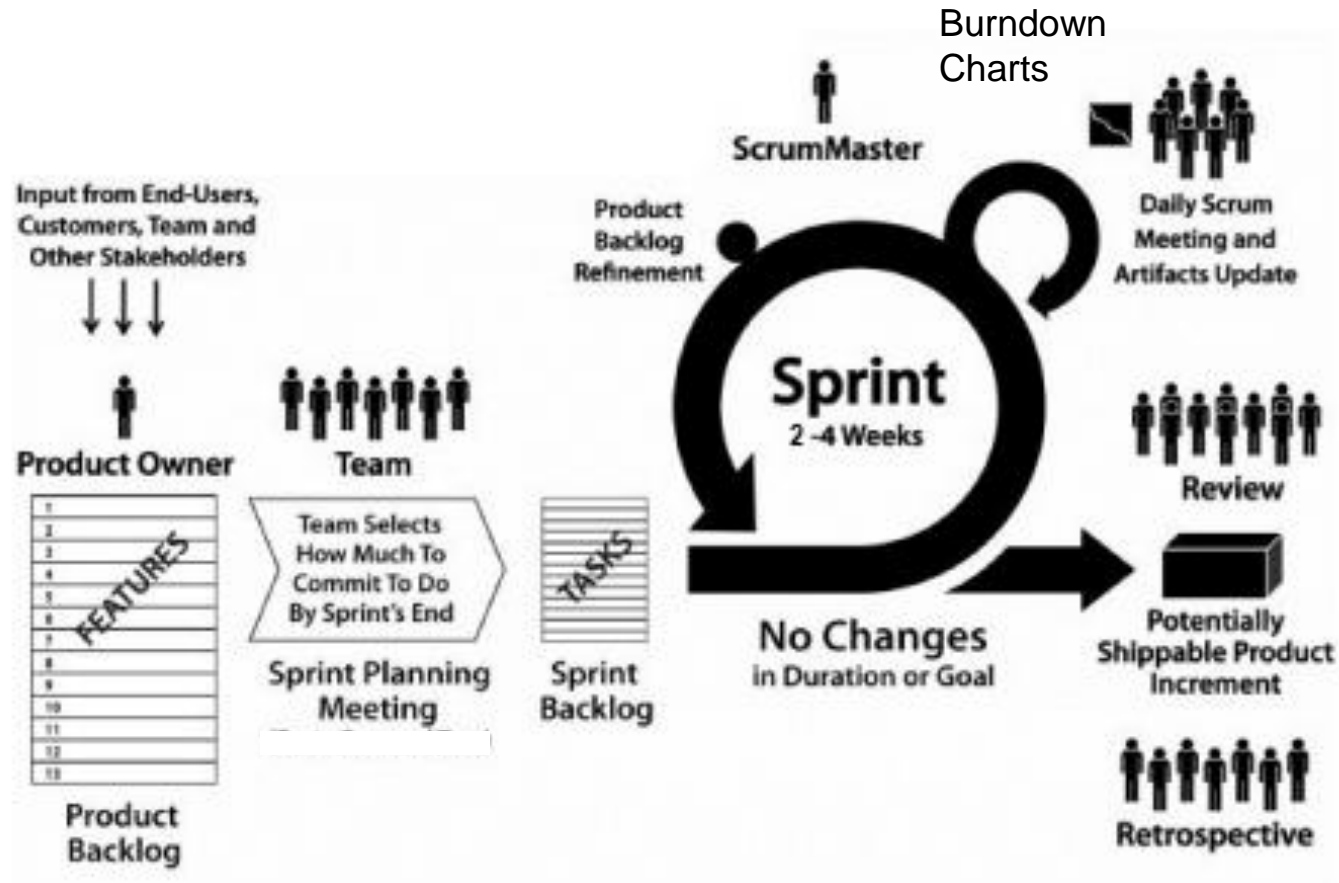
The Scrum Framework



By Clark & Vizdos

© 2007 implementingscrum.com

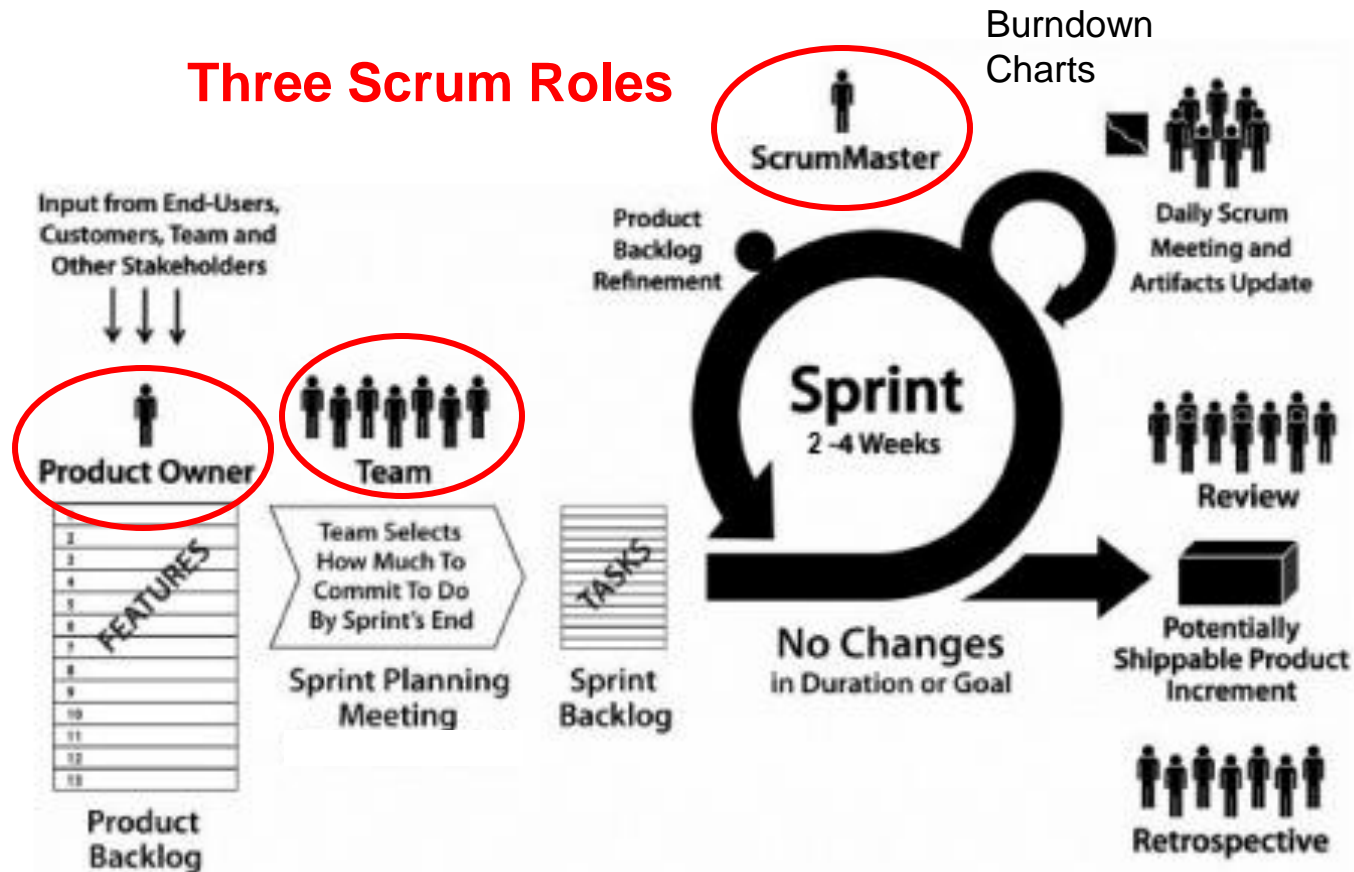
The Scrum Framework



Chris Noffke: <http://www.noffke.com/>

The Scrum Framework

Three Scrum Roles



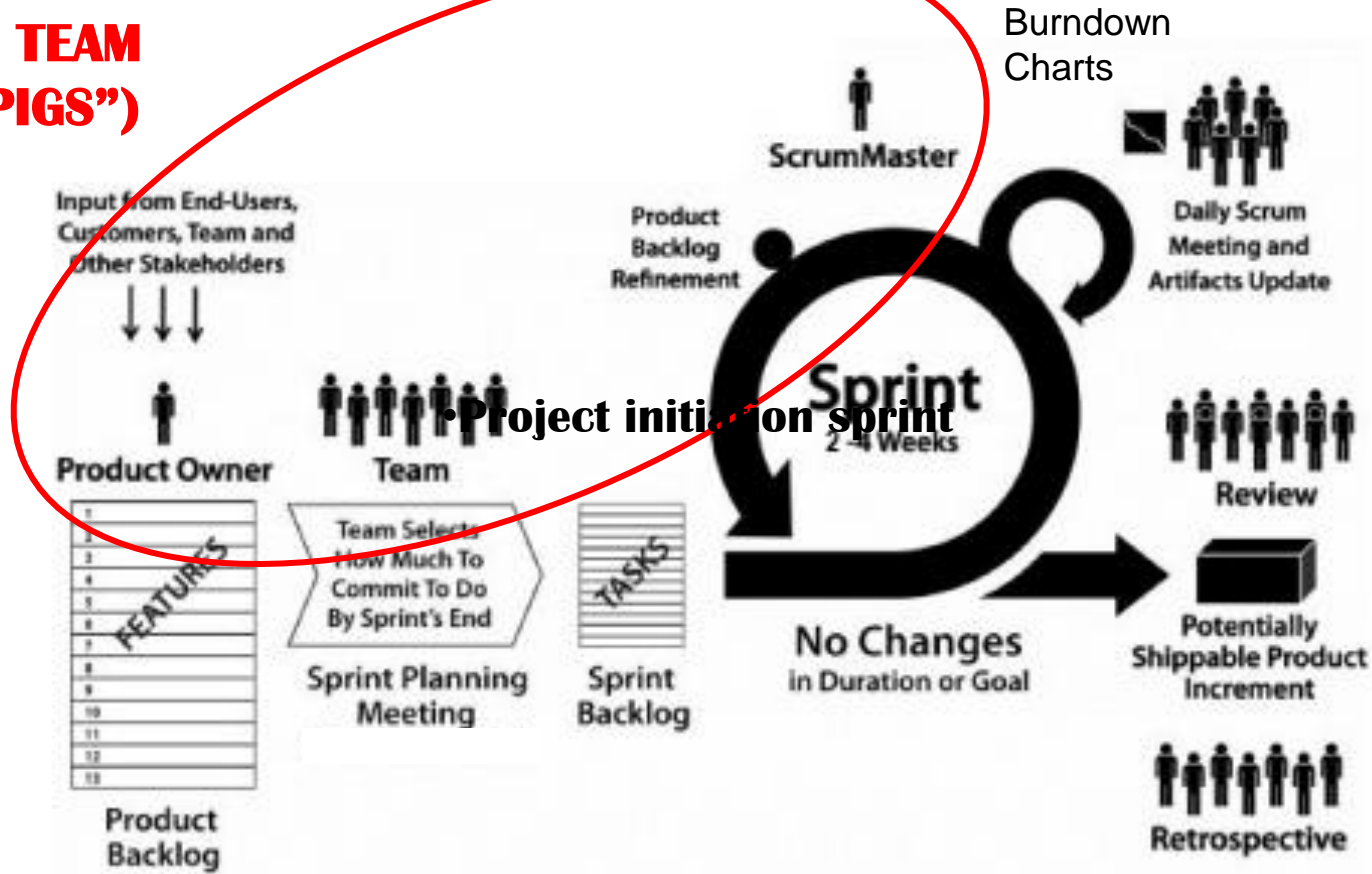
Chris Noffke: <http://www.noffke.com/>

Ham-n-Eggs



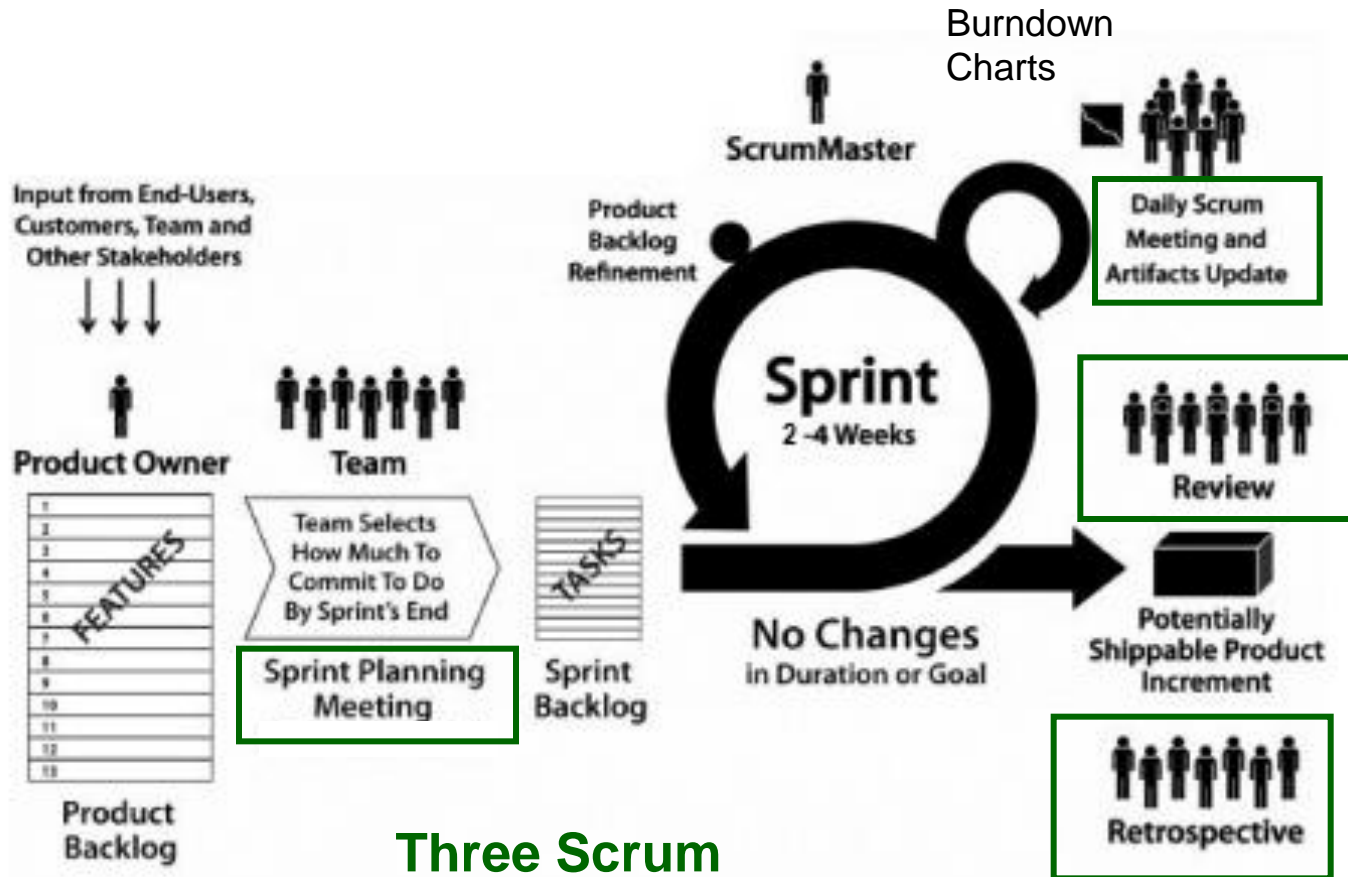
The Scrum Framework

**THE
SCRUM TEAM
(aka "PIGS")**



Chris Noffke: <http://www.noffke.com/>

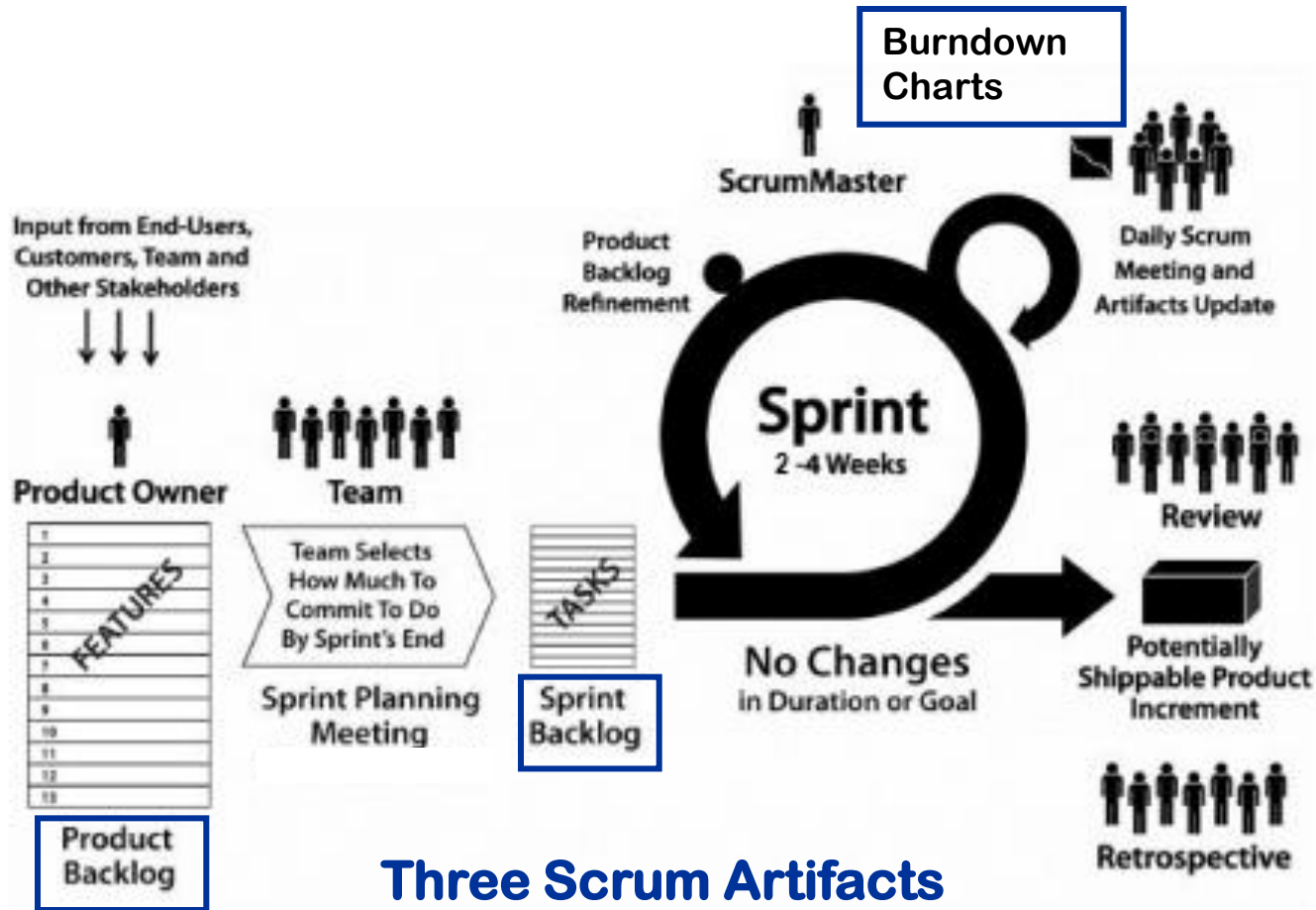
The Scrum Framework



Three Scrum Ceremonies

Chris Noffke: <http://www.noffke.com/>

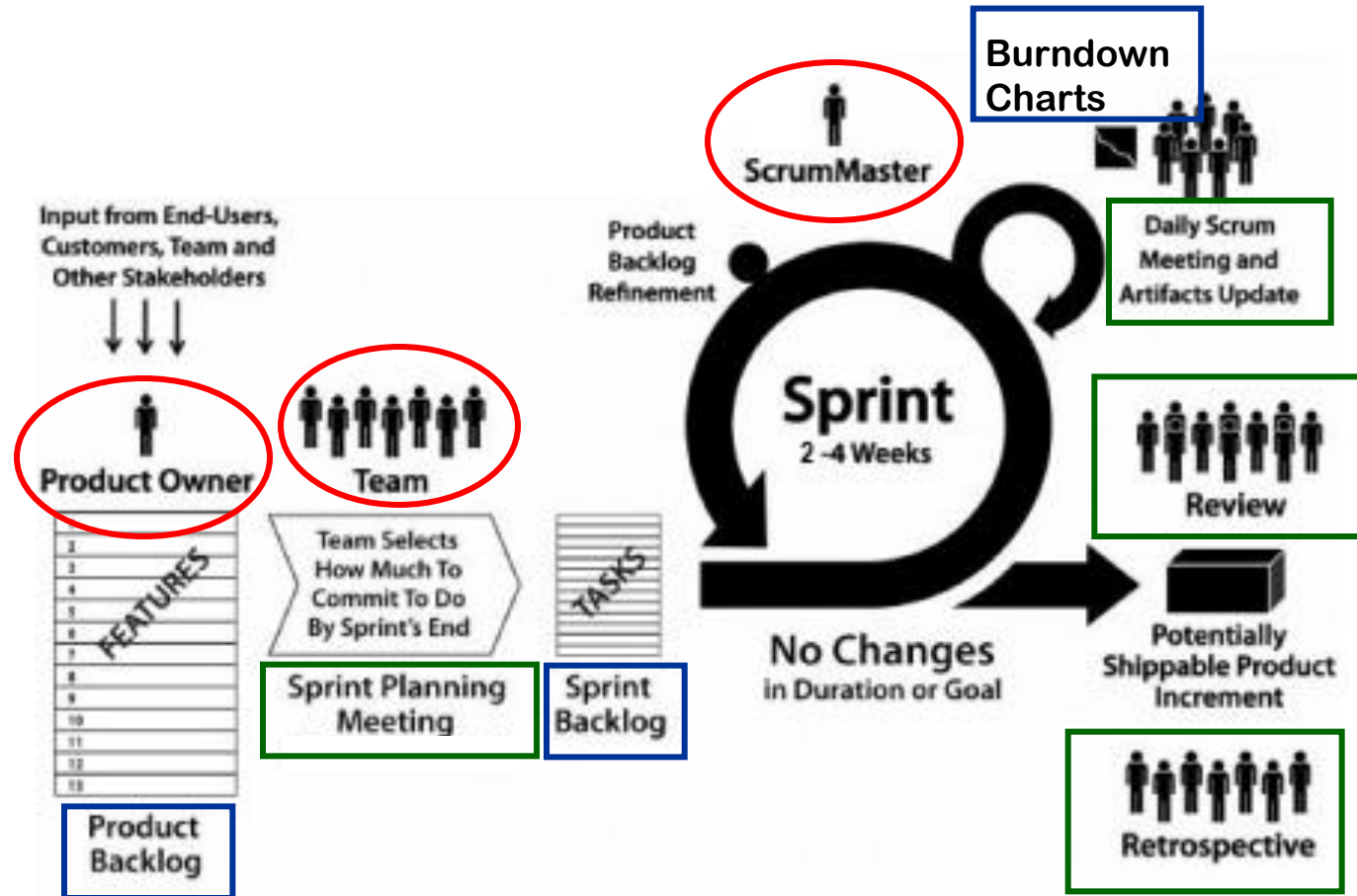
The Scrum Framework



Three Scrum Artifacts

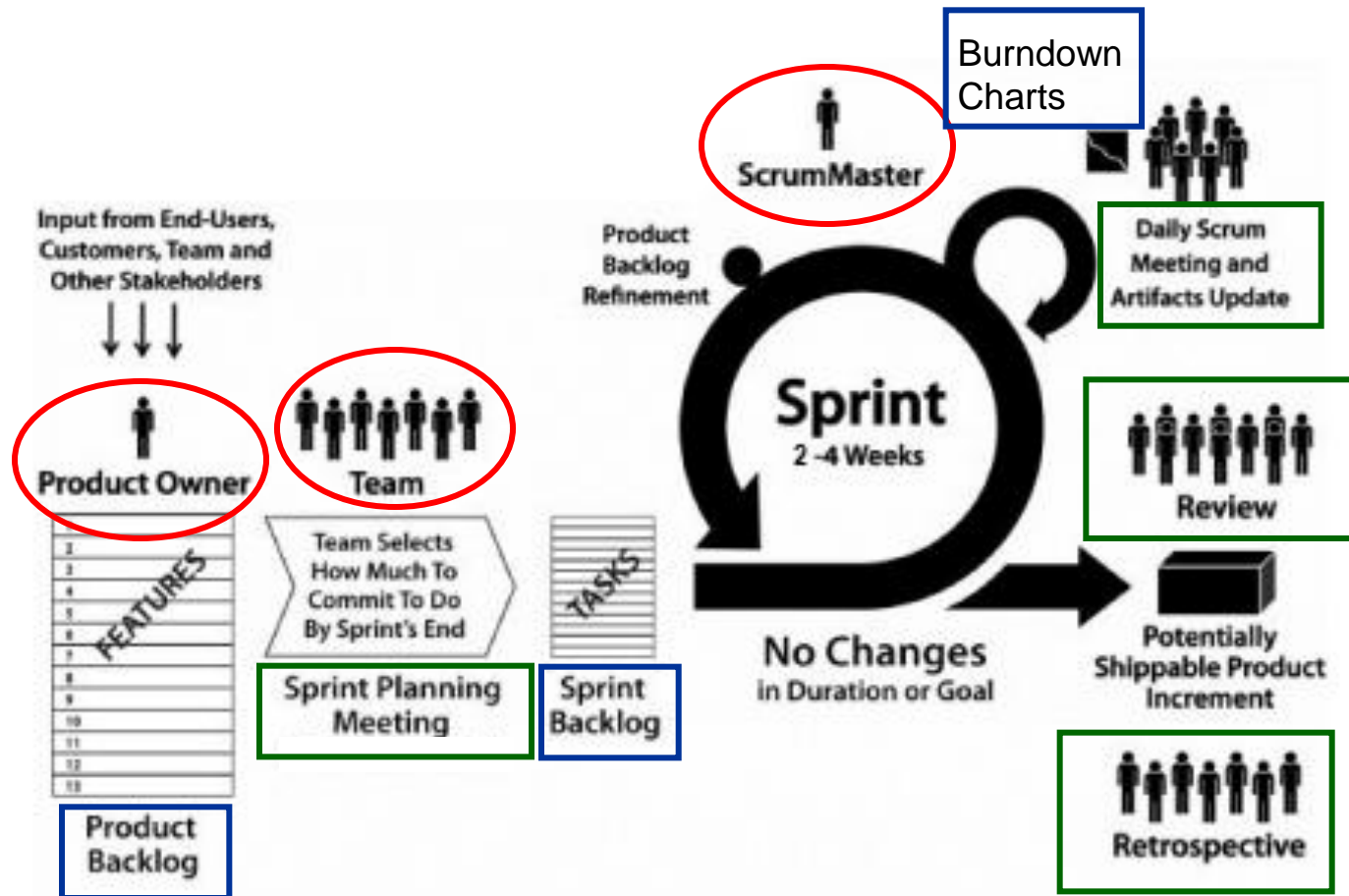
Chris Noffke: <http://www.noffke.com/>

The Scrum Framework



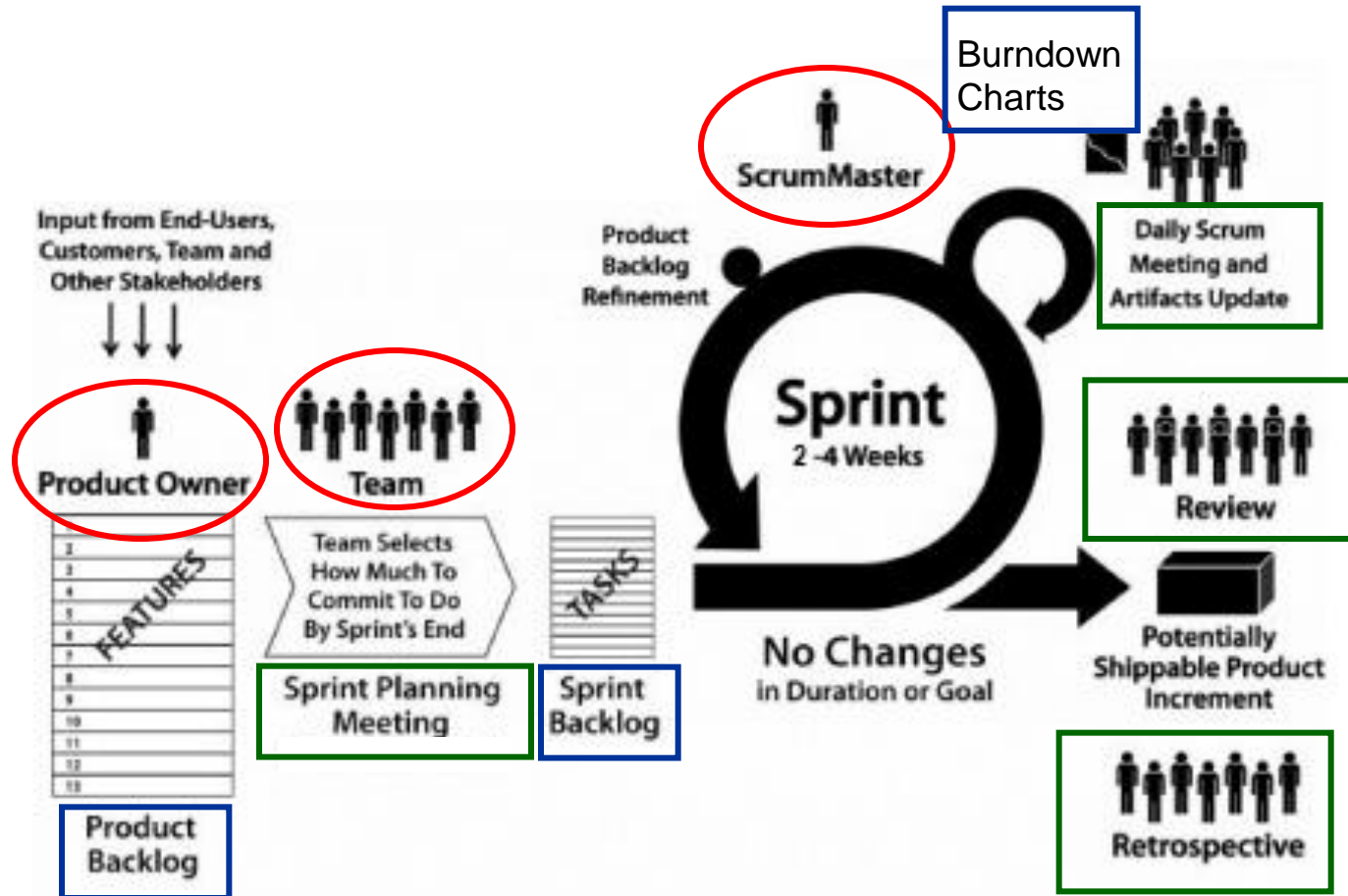
Chris Noffke: <http://www.noffke.com/>

Scrum is not about software...



Chris Noffke: <http://www.noffke.com/>

Scrum is a Project Management FRAMEWORK



Chris Noffke: <http://www.noffke.com/>

Scrum Origins

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

Hiroataka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”, *Harvard Business Review*, January 1986.

Mike Cohn, Mountain Goat Software



Scrum is not (always) about software...

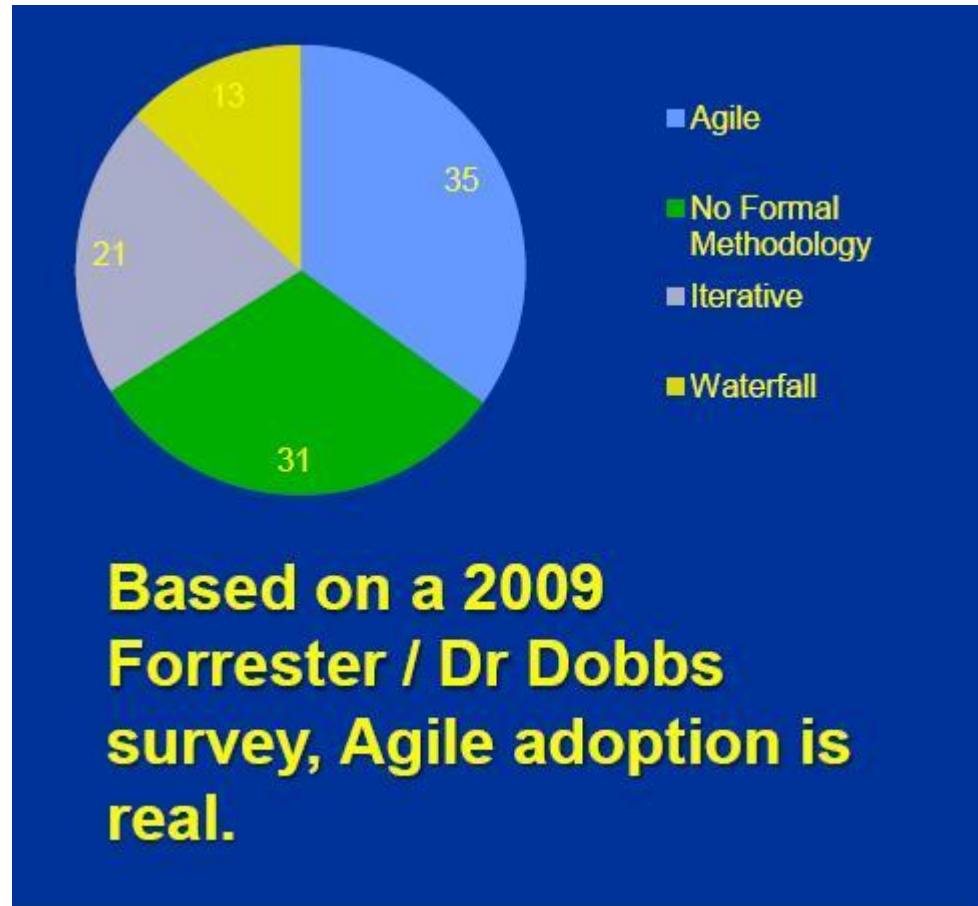


“Scrum in church? Of course! How else did God create the world in seven days?”

Rev. Arline Conan **Sutherland**,
First Parish Lexington - Lexington, MA

- 1986 Harvard Business Review Paper*
- 1993 Ken Schwaber & Jeff **Sutherland**
- 1995 OOPSLA Presentation
- 1996 Extreme Programming (XP)
- 2001 Agile Manifesto
- **“Agile Values & Practices”**

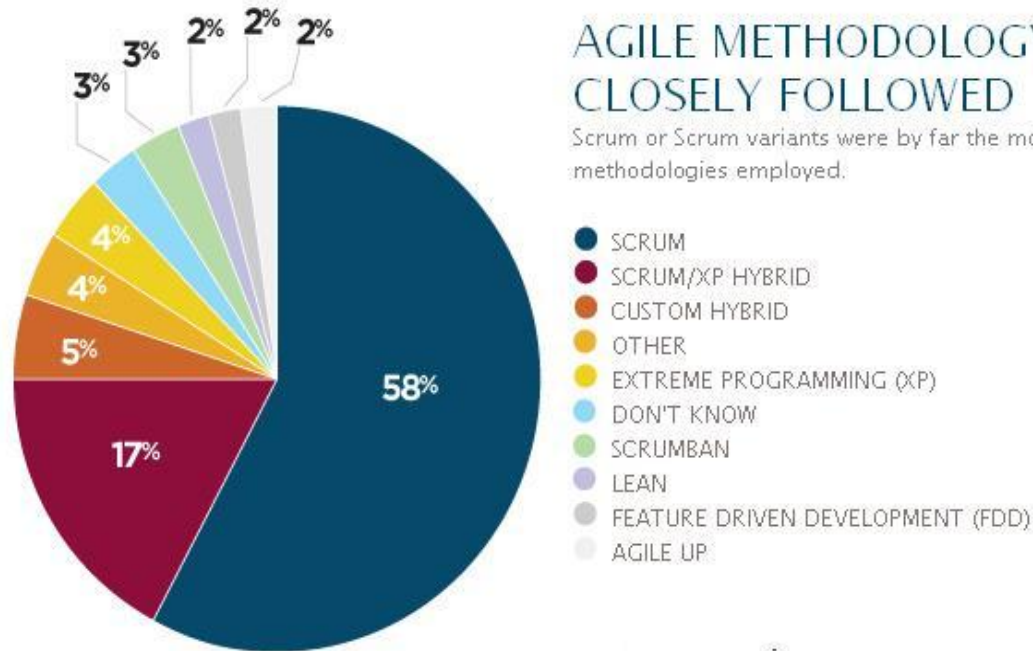
Is Scrum Mainstream?



“Perils of Agile, But”,
Dottie Acton – Lockheed Martin Senior Fellow, 2010



AGILE METHODS & PRACTICES



2010 State of Agile Development Survey Results
VersionOne



Scrum is used at...

Microsoft

Yahoo

Google

Electronic Arts

IBM

Philips

Siemens

Nokia

Lockheed Martin

Capital One

BBC

Intuit

Apple

Nielsen Media

Qualcomm

Texas Instruments

Salesforce.com

John Deere

Lexis Nexis

Sabre

Oracle

Time Warner

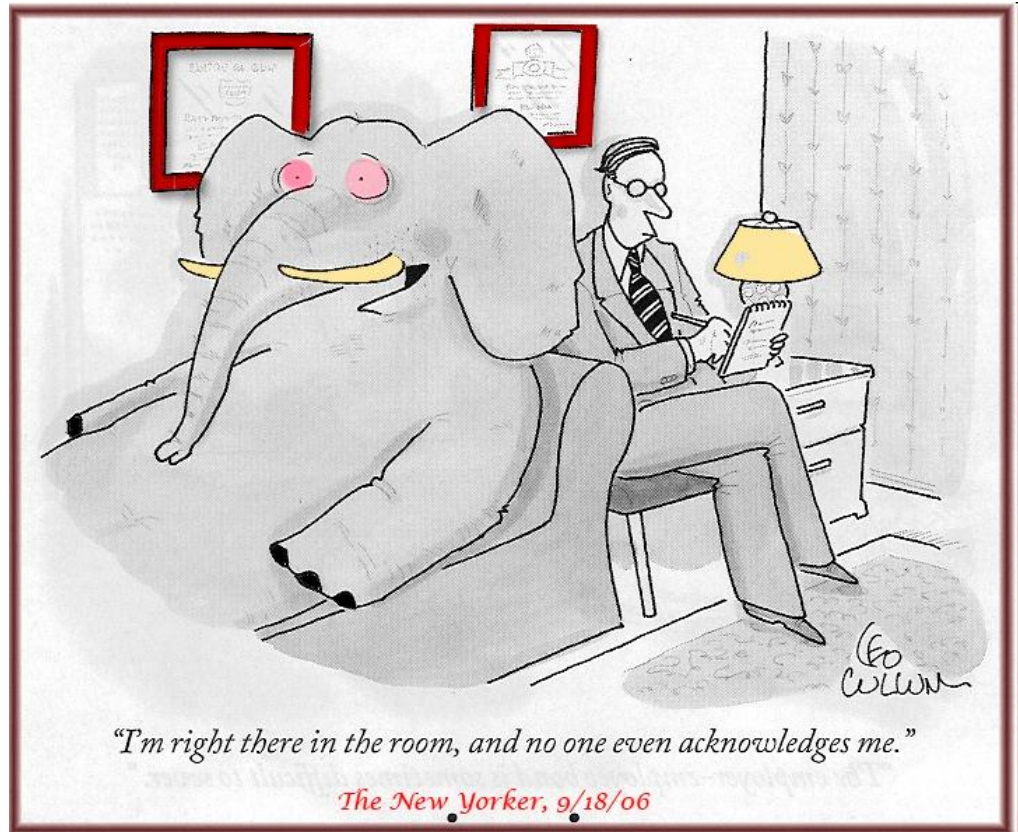
Turner Broadcasting



Scrum Illuminates Opportunities

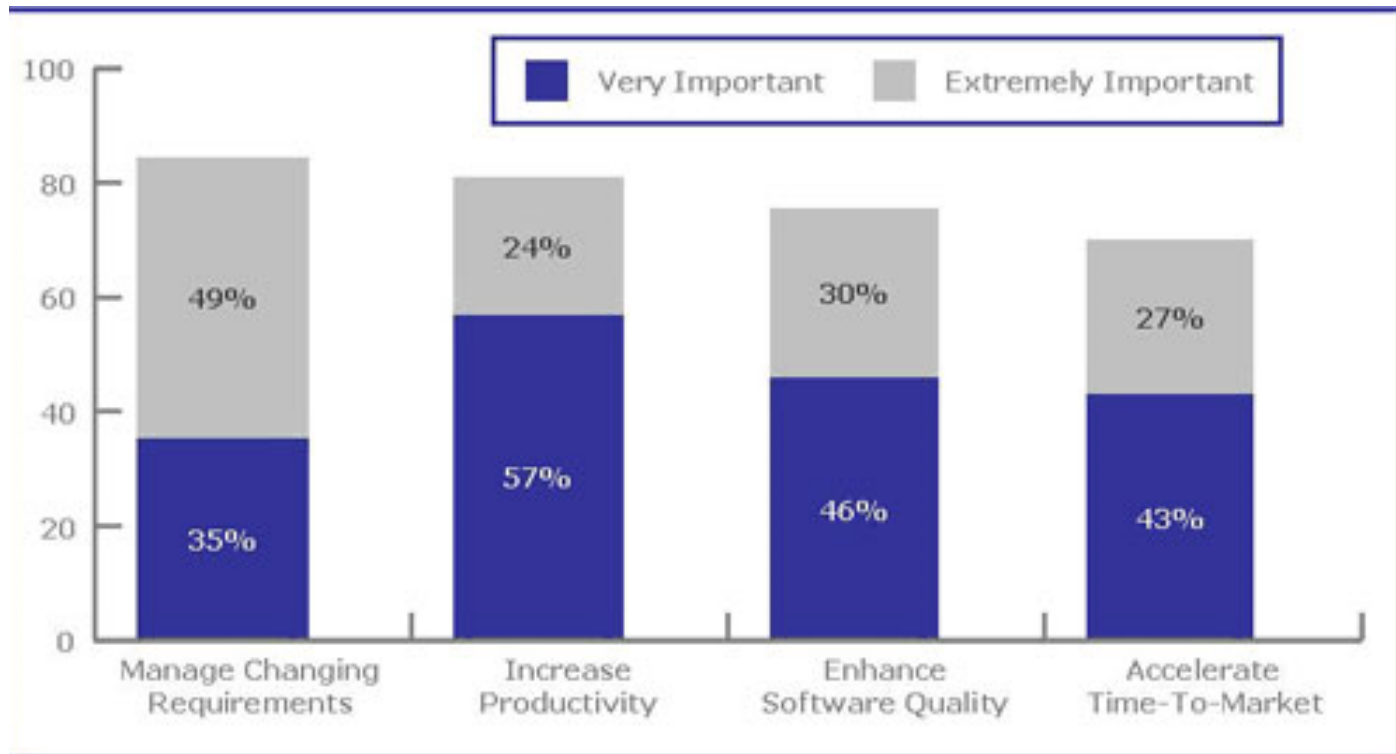
Scrum works by exposing the impediments in our processes, by shoving them in our face so we have no choice but to do something about them.

- Ron Jeffries



LOCKHEED MARTIN

We never forget who we're working for™



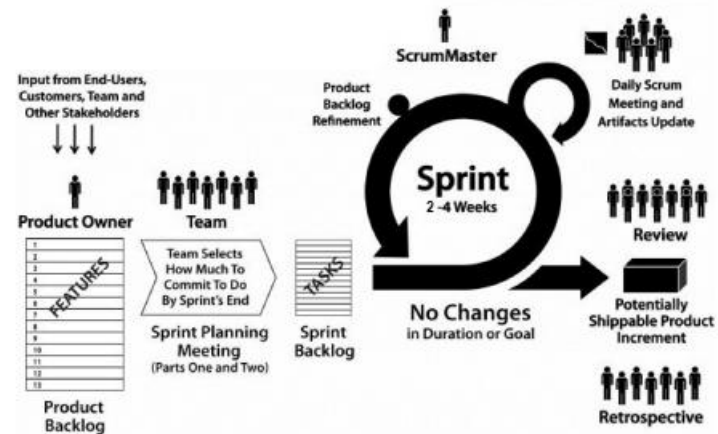
“War Stories – Fighter Jets and Agile Development at Lockheed Martin”, Agile Journal, 2007

Scrum & Agile Practices

Agile Practices

Time Boxed Iterations
Self Organizing Teams
Continuous Integration
User Stories
Test First/Driven Development (TFD/TDD)
Pair Programming
Refactoring
Retrospectives
... others

Scrum is a ...FRAMEWORK



Practices used in a Scrum Project

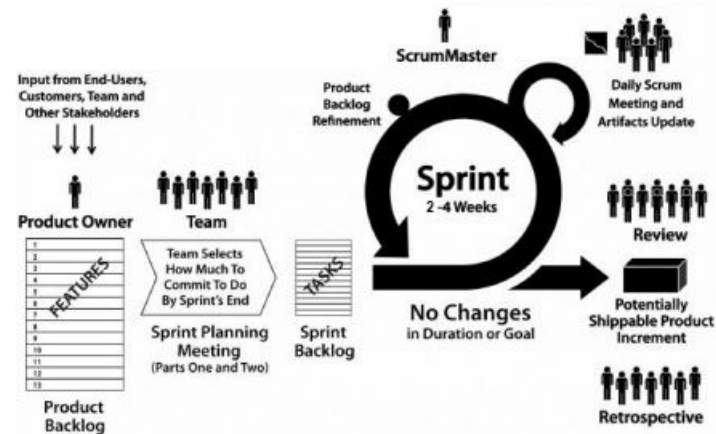
Scrum Roles
Scrum Artifacts
Scrum Ceremonies

Scrum & Software Engineering Practices

SE Practices

Time Boxed Iterations
Self Organizing Teams
Continuous Integration
User Stories
Test First/Driven Development (TFD/TDD)
Pair Programming
Refactoring
Retrospectives
... others

Scrum is a ...FRAMEWORK

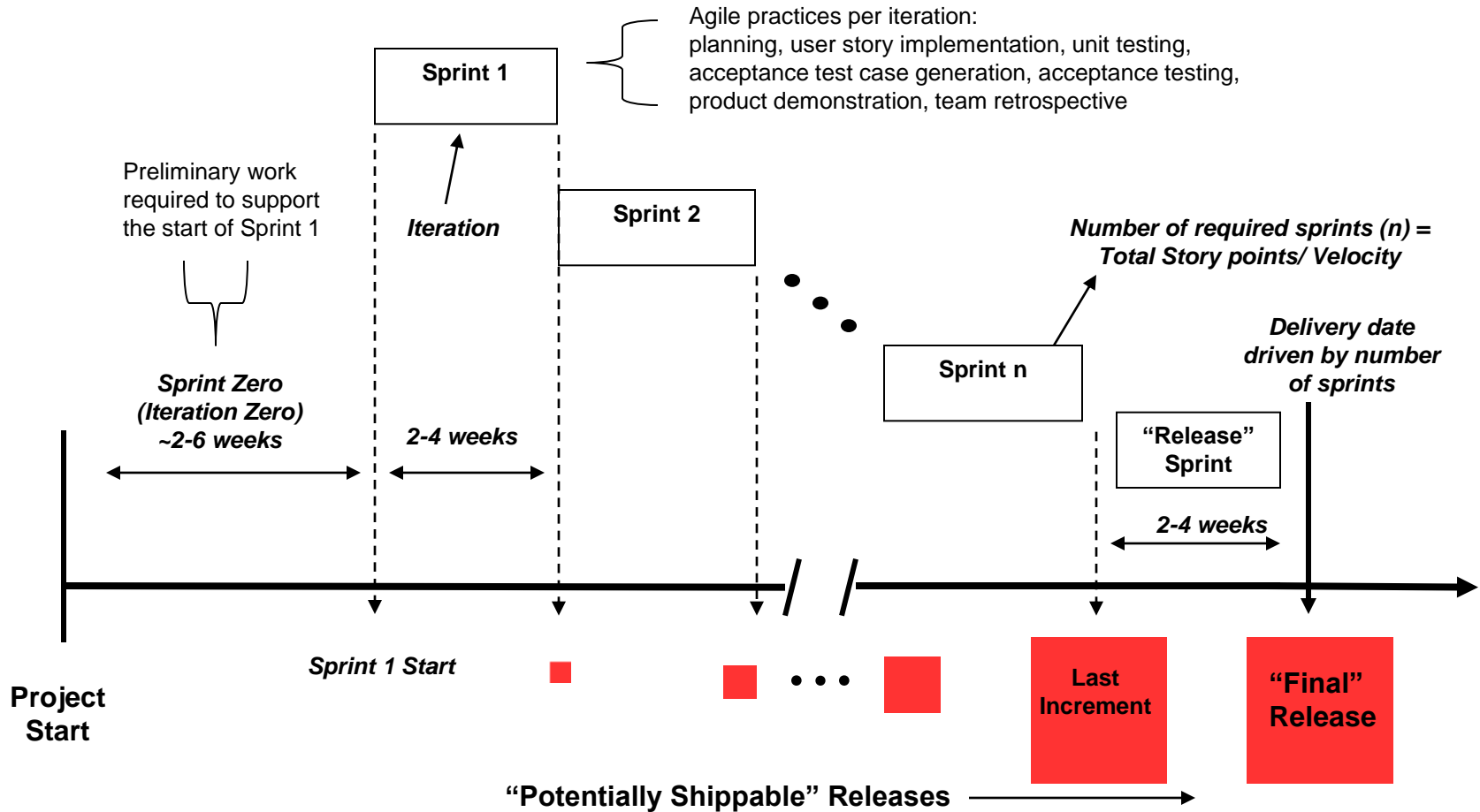


Practices used in a Scrum Project

Scrum Roles
Scrum Artifacts
Scrum Ceremonies

Ideal Scrum

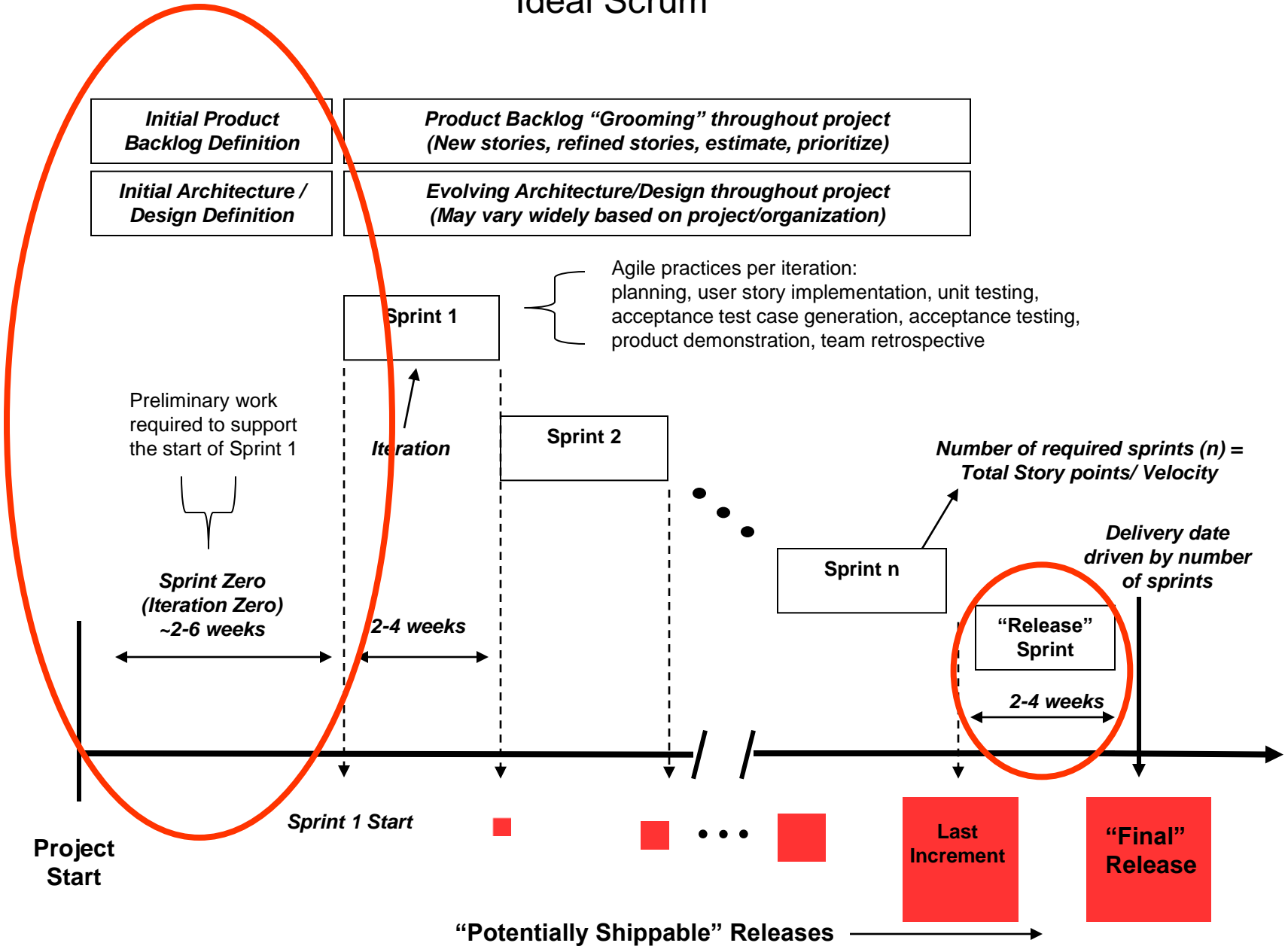
Initial Product Backlog Definition	Product Backlog "Grooming" throughout project (New stories, refined stories, estimate, prioritize)
Initial Architecture / Design Definition	Evolving Architecture/Design throughout project (May vary widely based on project/organization)



(the often overlooked) Sprint Zero

- Minimal Backlog to start Sprint 1
- Project initiation sprint
- Environment setup / CI Support
- Architecture / High Level Design
- Team setup / Team norms
- Training / Instruction
- **Define Deliverable Artifacts (DONE!)**

Ideal Scrum



Technical Debt

- Incurs when project team chooses an approach that's expedient in the short term but that increases complexity and is more costly in the long term.
- Debt can be incurred both intentionally and unintentionally
- As in our financial lives, there is “good” and “bad” debt

Scrum & Technical Debt

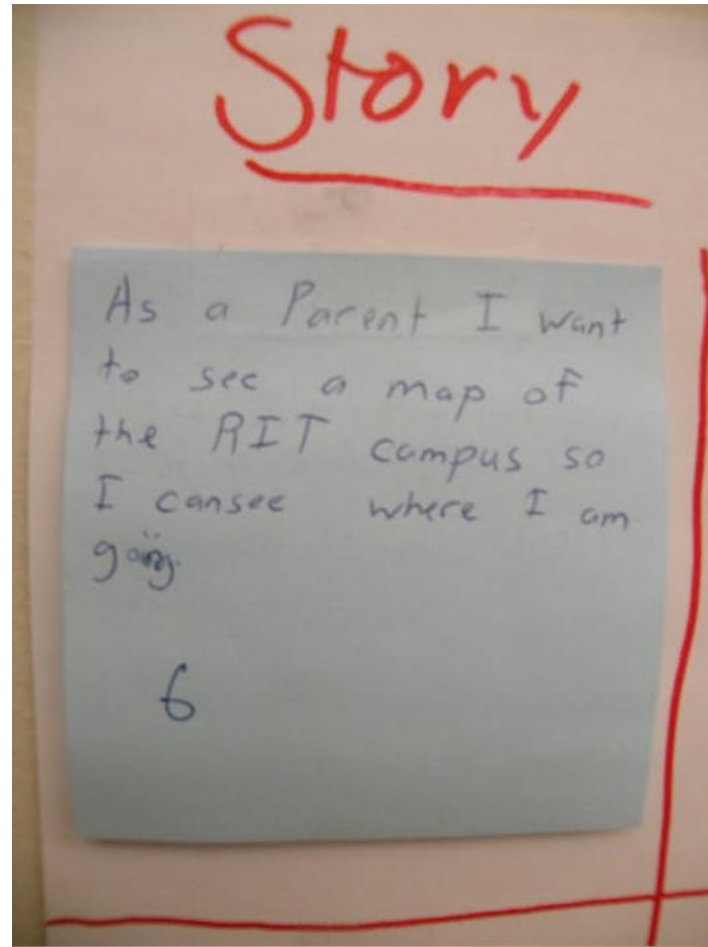
- **A complete description of what it means to be DONE-DONE with a user story**
 - Addresses intentional debt
 - Conscious decision to complete stuff “later”
 - Affects credibility of velocity metric
- **Small work increments (minimal WIP)**
 - Addresses unintentional debt
 - Defect propagation
 - Avoid long term hand-offs, work queues

Done-Done? Did you do...

- Performance testing
- Stability testing
- Refactoring
- Integration with the work of the other six teams
- Integration testing with the work of the other six teams so the increment is the totality of all seven teams
- Release notes
- Internationalization to the six cultures where the product will be sold
- User acceptance testing
- Regression testing



Creating the Product Backlog



Product Backlog

- Prioritized list of product features as determined by the **Product Owner**
- Reprioritized are the start of each new sprint.
- *Scrum does not specify the format of items on the Product Backlog, but typically they expressed as **User Stories**.*
- Each User Story also defines the **acceptance criteria** the Product Owner will use to accept the release at the end of the Sprint - **DONE!**
- Acceptance criteria is expressed as Acceptance (Functional) Test Cases



Product Backlog

Current Product Backlog

ID	User Story	Acceptance Criteria	Story Points
1	As an RIT student, I want the application to know where I am on campus via GPS.	Verify GPS is on. Verify student is on campus. Verify a location are returned within 10 minutes.	3
2	As an RIT student, I want the application to not drain battery when I am not using it.	Verify application is not in the foreground. Verify the GPS is not being accessed.	3
3	As a developer, I want the application to run on android 2.2+	Verify the manifest uses SDK 8 or higher. Verify the application is tested on android 2.2+ .	1/2
4	As a visitor on campus, I want to find build 70 for a meeting	Verify user location. Verify building location.	5
5	As a parent, I want to know the magnetic orientation of the phone.	Verify phone displays a direction on top of the map. Verify the direction is correct.	5
6	As a parent, I want to see a map of the RIT campus so I can see where I am going.	Verify the phone can access Google Maps. Verify location.	1
7	As a new student, I want to be given walking directions from my current location to a destination building	Verify phone shows directions. Verify arrival at destination building.	13



User Stories



www.dilbert.com
scottadams@aol.com



1/14/03 © 2002 United Feature Syndicate, Inc.



What is a “User Story”

- **Card**
 - Stories are traditionally written on index cards.
 - Cards may be annotated with notes, estimates, etc.
- **Conversation**
 - Details behind the story come out during conversations with customer, product owner
- **Confirmation**
 - Acceptance tests validate the story was correctly implemented in the application

The Card

As a user, I want to purchase a book.

As a user, I want to cancel an order.

Communciating the Details

“As a user, I want to cancel an order”

- **Does the user get a full or partial refund?**
 - Credit card? Site credit? Other?
- **Is a confirmation provided to the user?**
 - How?
- **Can you specify a subset of items from an order?**

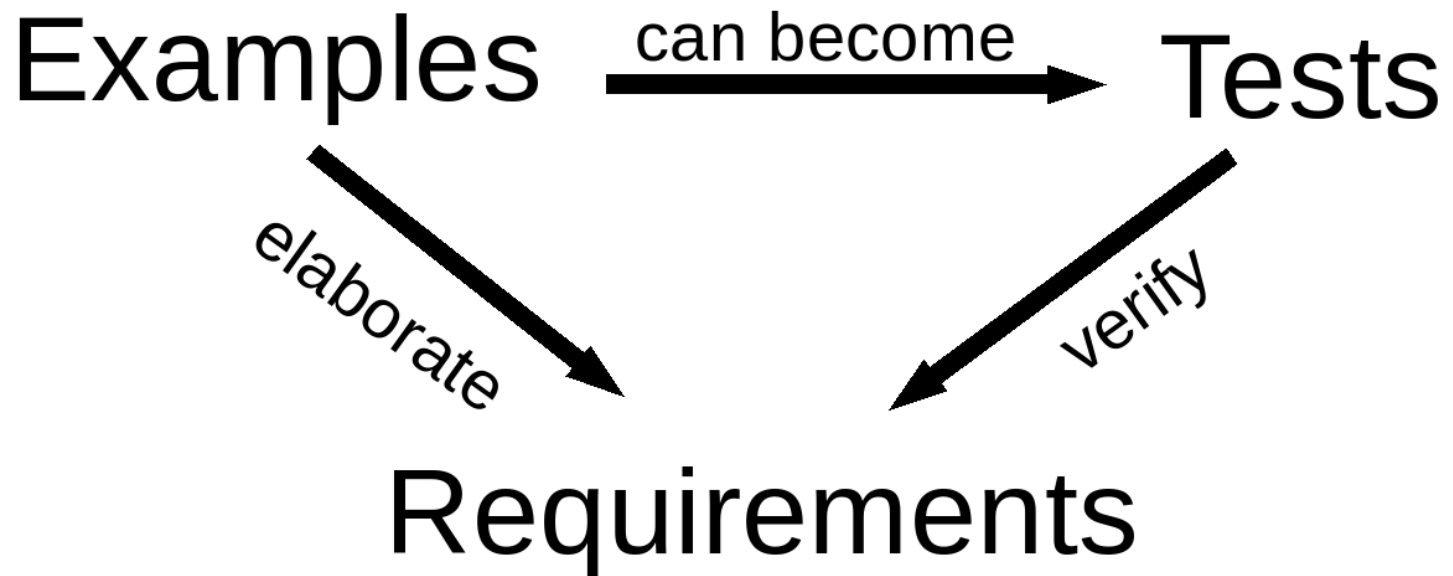
Details are in the Confirmation

“As a user, I can cancel an order”

Acceptance Criteria:

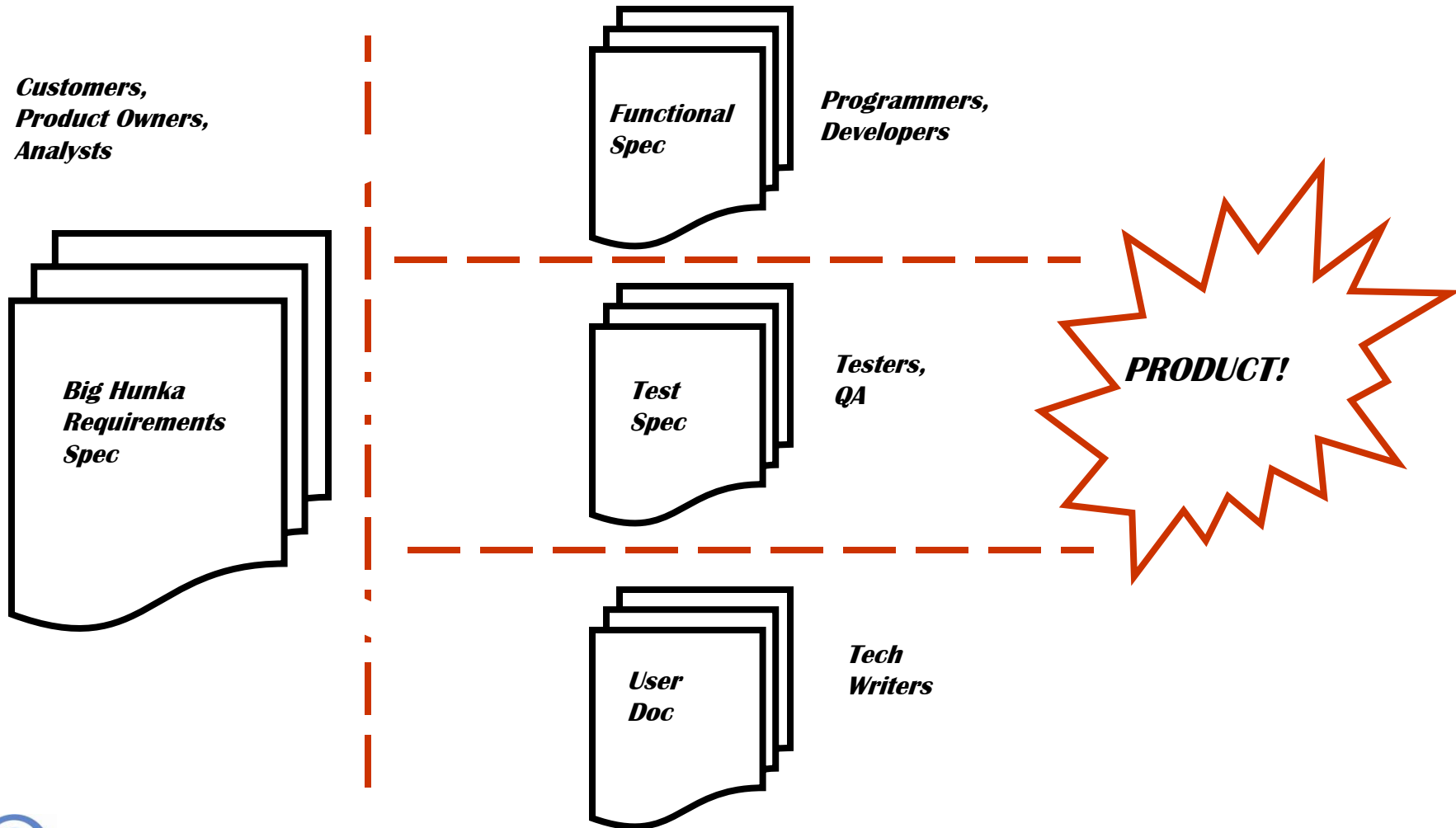
- **Verify that the user canceling a credit card order is credited on their account.**
- **Verify the user receives an email confirmation.**
- **On an order of multiple items, cancel a subset of those items and verify the remaining items are still processed.**

Requirements Communication



"Bridging the Communication Gap" - Gojko Adzic

Traditional Flow



Agile Flow

- User stories are the initial entry into the elicitation process.
- Further conversations and questions identify requirement details.
- Details are captured as acceptance criteria.
- **Acceptance criteria becomes executable test cases.**
- Developers implement to the acceptance test cases.
- Clarifications, changes generated new test cases.
- New requirements generate new user stories



Why this Works

1. Words are imprecise – stories shift the focus from writing to talking.
2. Stories are equally understood by customers and developers.
3. Stories support iterative development.
4. Stories are the right side for planning.
5. Stories support participatory design.
6. Stories emphasize the user’s goals.

“The words we write on the story card are less important than the conversations we have”

“User Stories Applied” – Mike Cohn



User Story Template

As a <user role>,
I want <goal>,
so that <reason>

As a premium site member, I
can cancel my reservation up
to 24 hours in advance if my
travel plans change.

INVEST

- **Independent**
 - As much as possible, stories should not be dependent on each other.
- **Negotiable**
 - Details identified in the conversation.
- **Valuable**
 - The story has value to the customer/user.
- **Estimable**
 - Story allows prioritization and planning
- **Short**
 - Story can be implemented in one sprint
- **Testable**
 - We do not develop what we can't test. Defines **DONE!**



Story Writing Activity



Volunteer Support Site

- In groups hold a short user story workshop to write story cards for a generic web application that will support the coordination of volunteers for an event (festival, sports tournament, fundraiser, etc.).
- Identify the primary user roles that would be using this application.
- Start by brain-storming potential stories, then collect similar stories and begin splitting as necessary.
- Collect acceptance criteria as it is identified.



Agile Estimation & Planning

- **Planning Levels:**
 - Strategy
 - Portfolio
 - Product
 - Release
 - Iteration (Sprint)
 - Daily

User Stories & the Planning Process

- Stories are estimated in unit-less “story points”
 - Estimates are based on size – not duration
 - Size estimate are relative to other stories
- As stories are selected from the product backlog for a sprint:
 - Teams identify tasks, estimated in duration
- The number of story points completed during a sprint is the team’s **velocity**
- Velocity is used to predict what features can be completed for a release (collection of sprints).

Planning Poker

- **Variation on the Wideband Delphi technique used in the Rand Corp (~1946).**
- **Those who do the work, estimate the work.**
- **Requires justification of estimate.**
- **Involves ALL team members.**

Homeowner chores

1. Replace sash cord on two bedroom windows
2. Power wash deck (12x12)
3. Stain deck (12x12)
4. Install ceiling fan in living room
5. Strip wallpaper in bedroom (10x11)
6. Hang mirror in dining room
7. Replace electrical outlet (1) in kitchen
8. Seal driveway (20x100)
9. Apply fertilizer to front lawn (2500 sq ft)
10. Assemble new gas grill

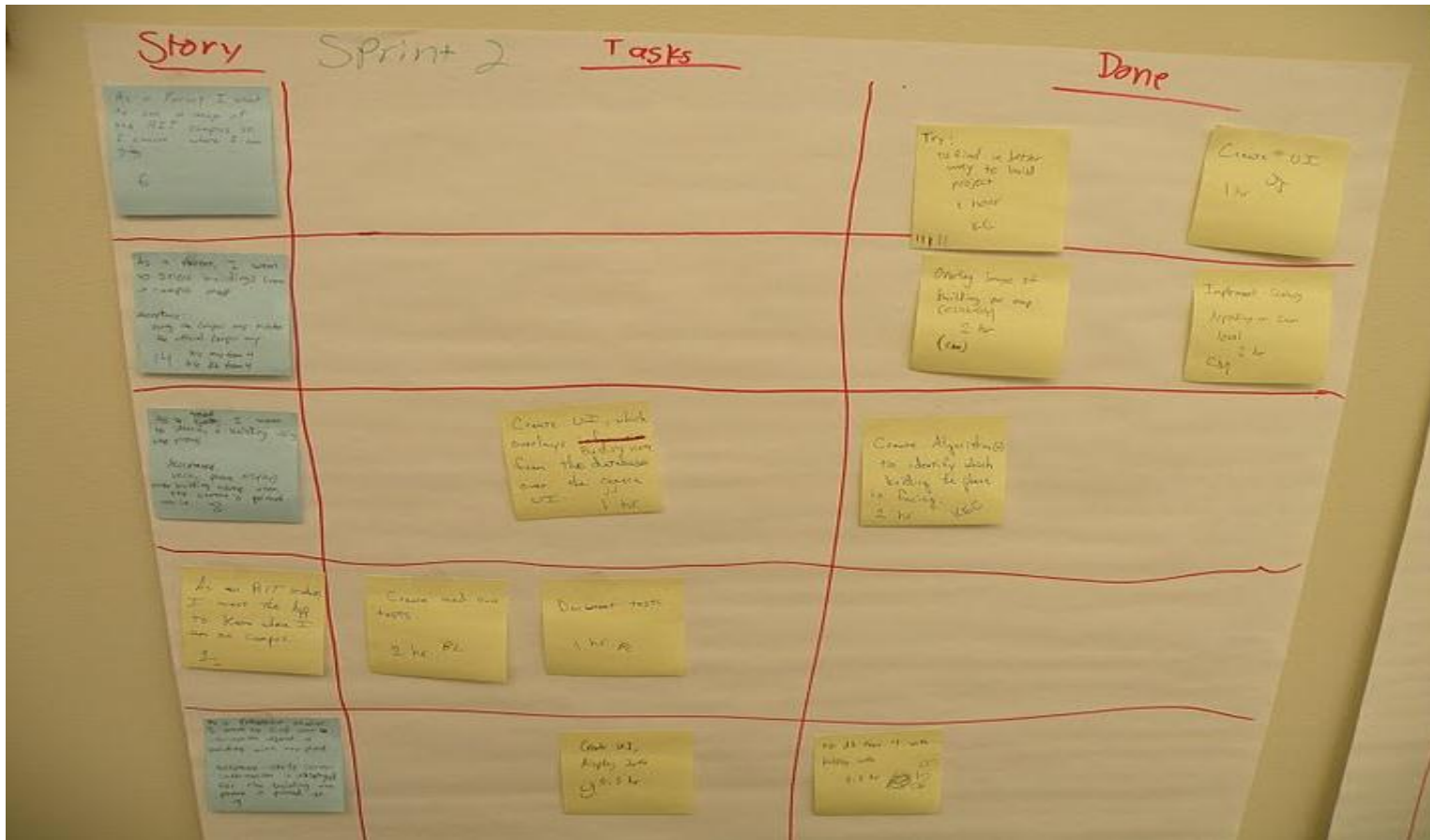


Scrum Sprints

- Starts with a **Sprint Planning Session** where a **Sprint Goal** is identified.
- 2-4 Week Iterations of all required life-cycle activities (Requirements-Test-Design-Construct-Test-Package-Document)
- Team creates a plan to FULLY complete features from the **Sprint Backlog**.
- Sprint ends with a **Sprint Review** – demo for Product Owner and acceptance or rejection
- **Sprint Retrospective** is held at the end to identify process improvements (keep, toss, try)



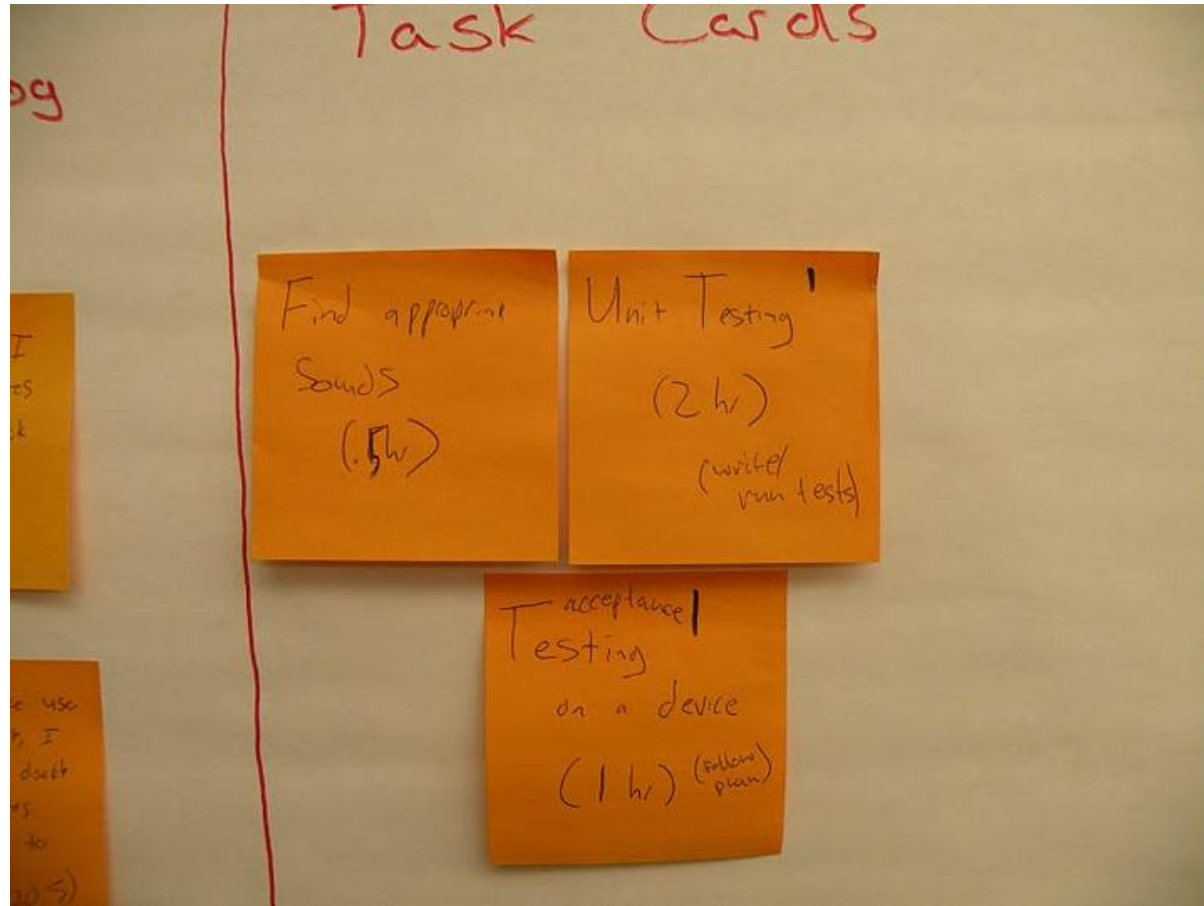
Sprint Planning



Sprint Backlog

- Teams select items from the (prioritized) Product Backlog that they can commit to completing.
- Product Backlog items are decomposed into **development tasks** and estimated by the team.
- Team members commit (sign up) to individual tasks. Tasks are not “assigned”.
- Any team member can add, delete or update tasks on the Sprint Backlog.
- Estimates of remaining work are updated daily (or each time team meets).
- Additional tasks added and estimated as the Sprint proceeds.

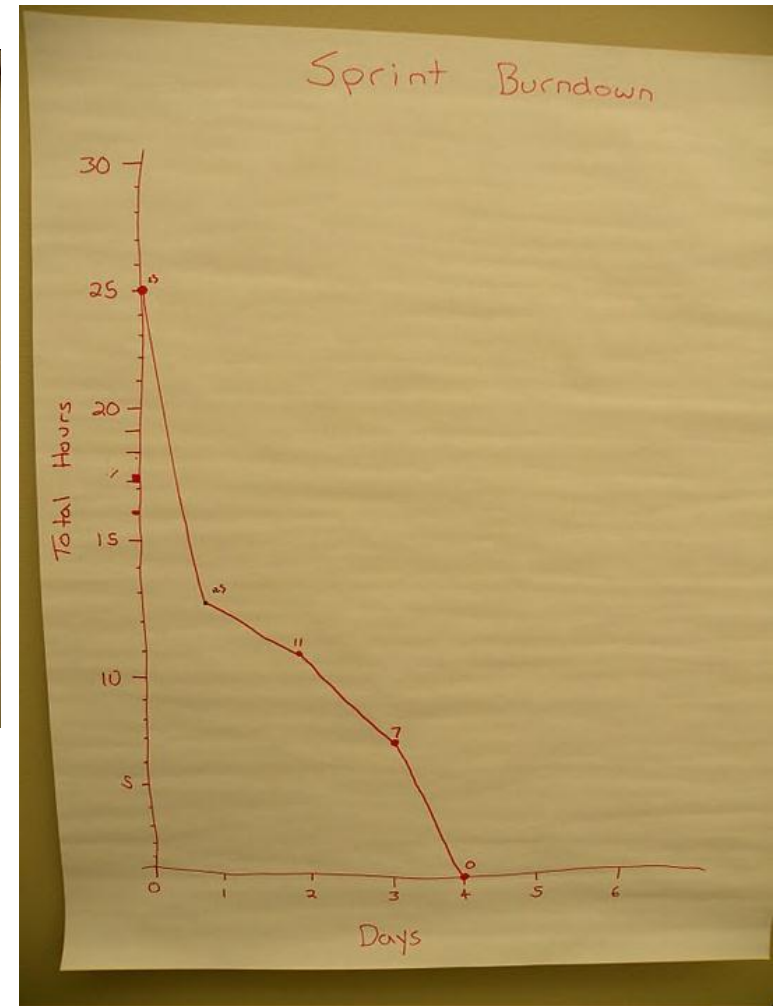
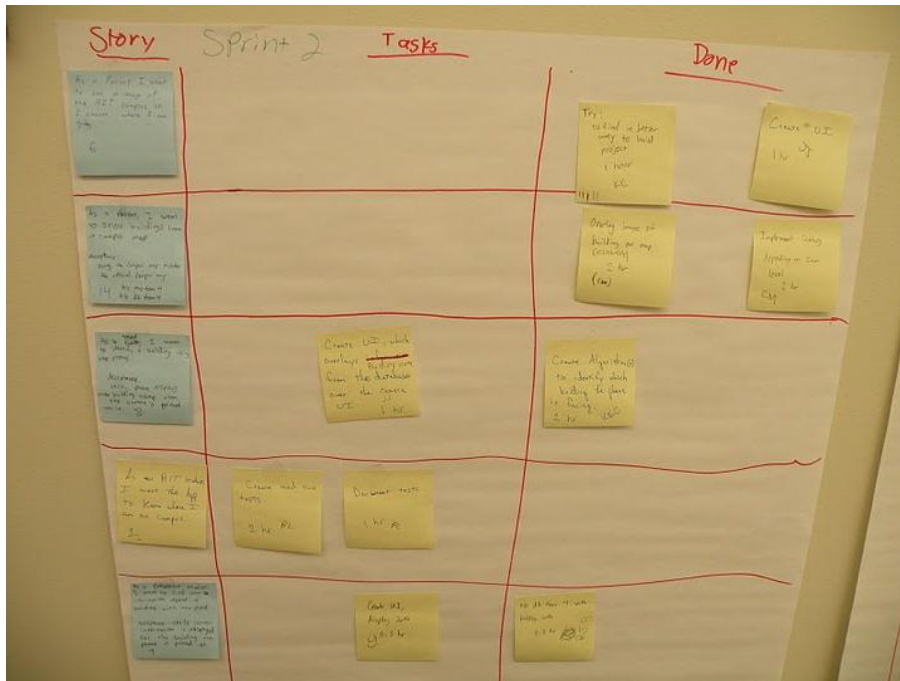
Task Estimation



Sprint Planning Activity



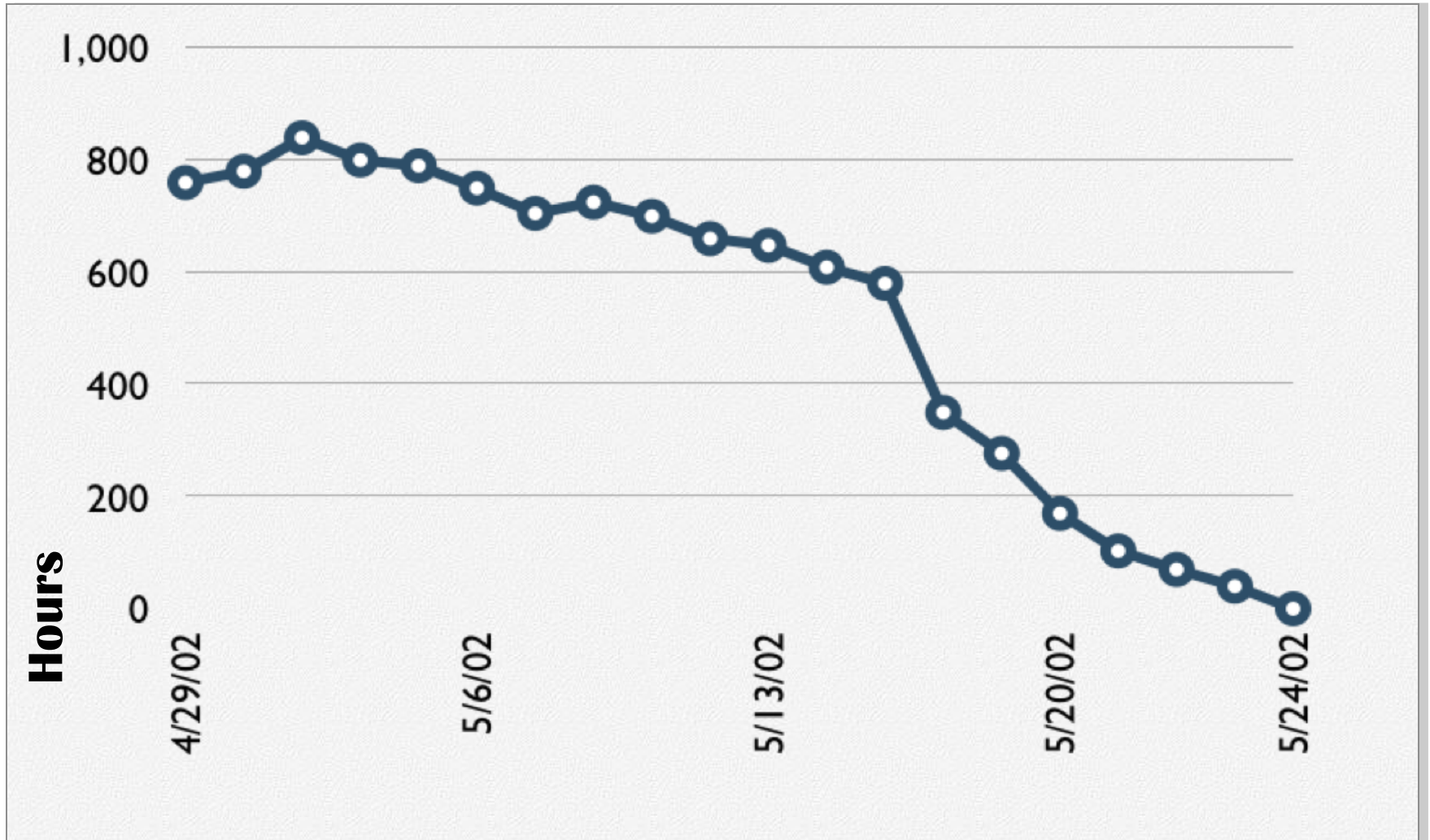
Sprint Execution



Managing the sprint backlog

- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known
- Estimated work **remaining** is updated daily

A sprint burndown chart



“Daily” Stand-Ups

At the start of each classroom session:

1. What did I accomplish since the last stand-up? (last session)
2. What am I planning on accomplishing before the next stand-up? (today)
3. What roadblocks are in my way?

**Making a verbal commitment to your team*



“Daily” Student Scrums

- Start by doing mock stand-ups
- Individually critique teams, the have class observe provide critique
- Make sure students are addressing their team, not you.
- “What I did on my summer vacation” team members become obvious.
- (Cheech & Chong: “Sister Mary Elephant”)



Technical Practices

- Collaborative Software Development
- **Continuous Integration**
- Test-Driven Development
- Acceptance Testing
- Unit Testing
- **Automated Testing**
- Simple Design
- Refactoring
- Coding Standards



Sprint Retrospective

Examples of Keep, Toss, Try...

- **Allowing more group members to participate in programming tasks**
- **Better estimation of how long tasks would take**
- **More frequently working as pairs or in small groups on tasks**
- **Including more group members in the project overall**
- **More testing at the unit and acceptance levels**
- **More fine grained break down of user stories into tasks**
- **Making sure to include all types of tasks needed to be completed for a story on the task breakdown chart**



Student Scrum Challenges

- **Course context – relate to learning outcomes**
- **Time (lack of) – Scrum teams get better after a series of Sprints**
- **Scrum roles – Product Owner, Scrum Master**
- **Initial state of requirements**



Student Scrum Opportunities

- **Course context – relate to learning outcomes**
- **It's a framework!**
- **PM practices – task breakdown, estimation, planning, process improvement ...**
- **Discipline and rhythm**
- **Communication and collaboration**
- **TESTING – TESTING - TESTING**



“Are we doing Scrum right?”

- **Better to ask:**
 - “Is Scrum doing right by us?”
 - “How useful has Scrum/Agile been to this project?”
- **Famous quotes:**
 - “A design is not good or bad, it is just more or less useful”
 - “There are no best practices, just darn good ones within a certain context”

Any Questions?



By Clark & Vizdos

© 2006 implementingscrum.com

References

- **Mike Cohn – Mountain Goat Software**
- **Scrum.org**
- **Alistair Cockburn**