

point**b**.



Lean IT and Agile 101

October 2013

John Okoro – Point B Agile Service Offering
Lead – jokoro@pointb.com

Pat Edmonds – Agile Service Offering Team –
pedmonds@pointb.com

Objective

- Introduction to Agile (including definition, concepts, components, and benefits)
- Understand how Agile and Lean principles align, and how they can be complementary in the organization
- Understand how Agile fits into a Lean IT approach
- Learn from case studies of how other organizations have used Agile to create business value
- Gain a strong foundation for how your organization can leverage Agile

What is Agile?

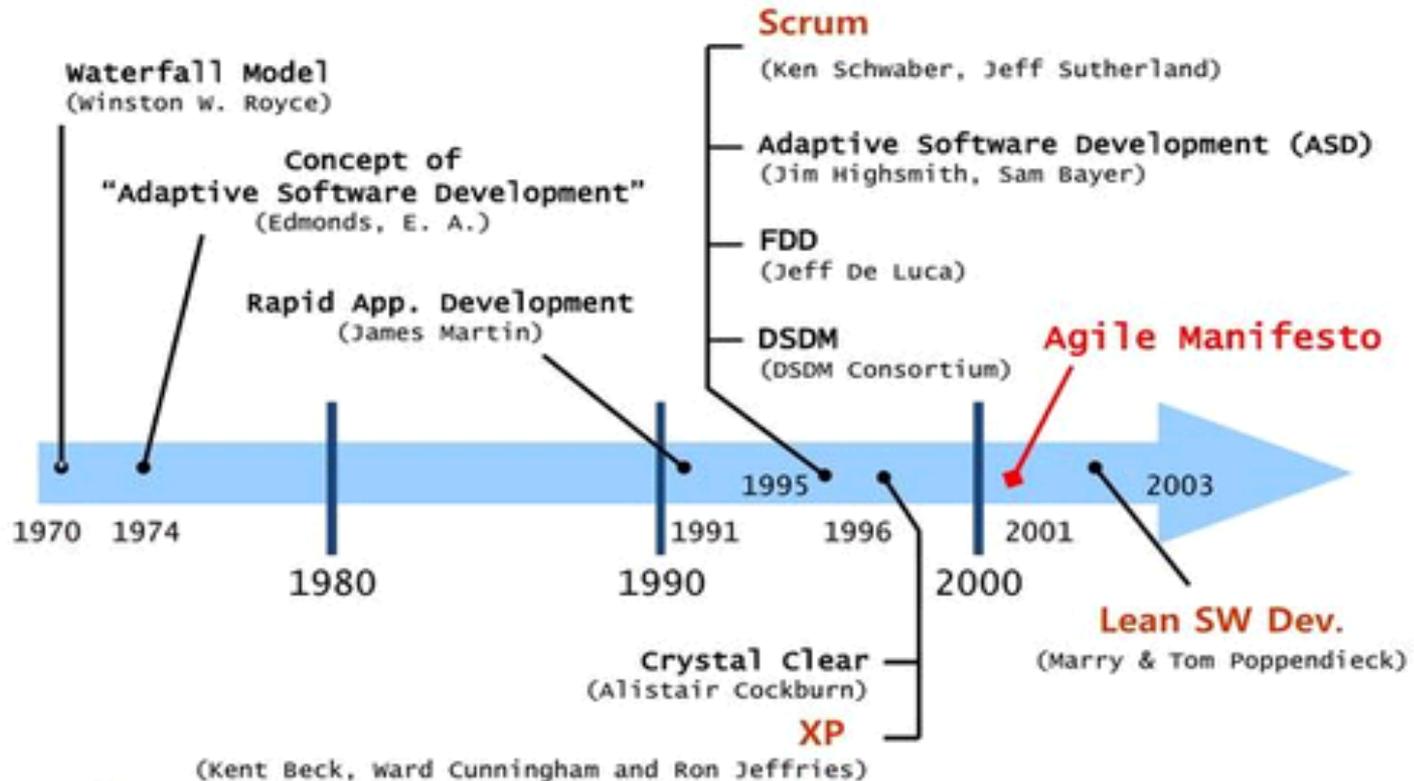
From Wikipedia:

- Agile software development is a group of **software development methods** based on **iterative and incremental development**, where requirements and solutions evolve through **collaboration** between **self-organizing, cross-functional teams**.
- It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle.

Examples of Agile methodologies: Scrum, XP, AUP, Kanban, etc.

A Brief History

History of Agile

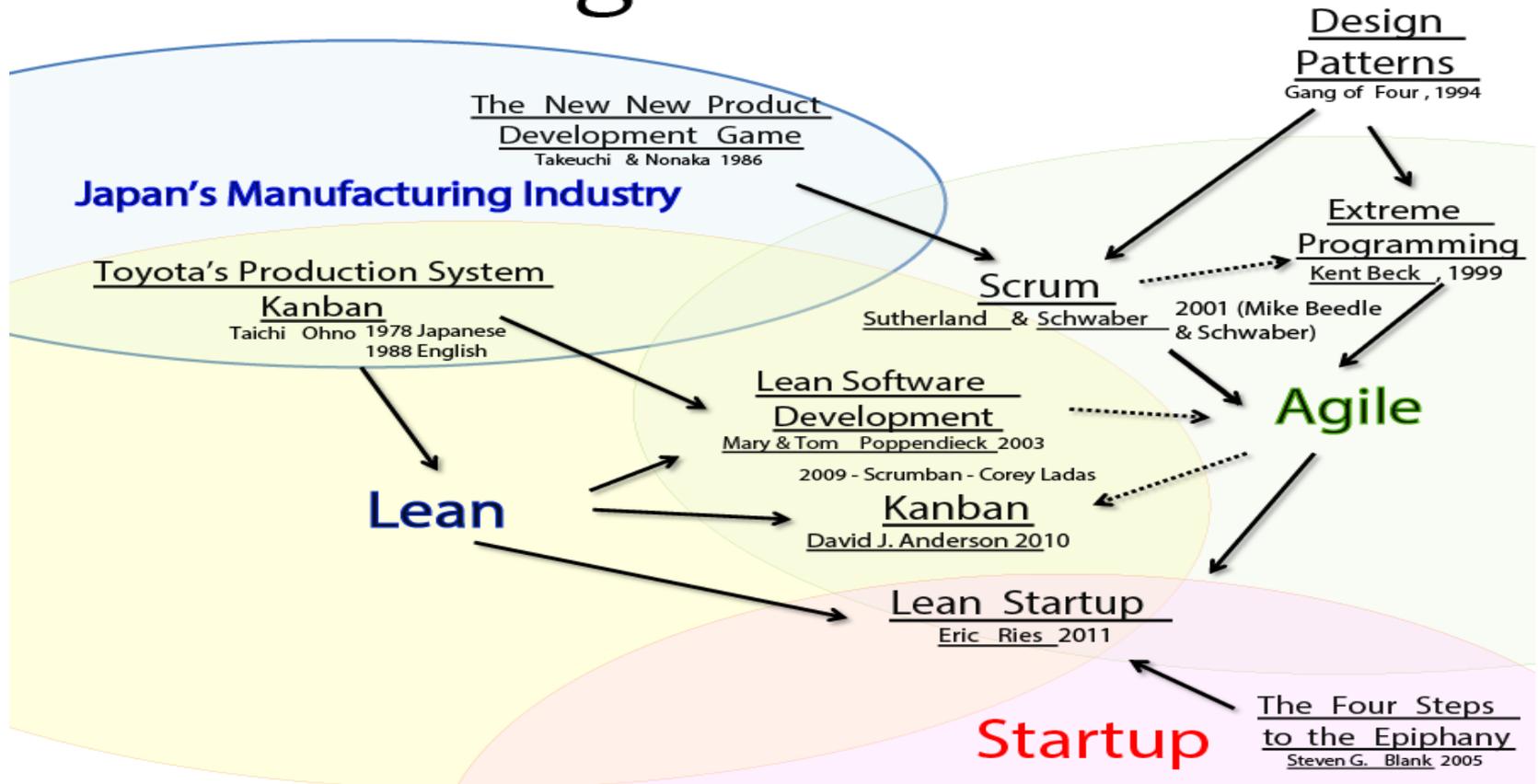


<http://bogus212.egloos.com/294217>

Agile & Lean

Agile & Lean

2013 Yasunobu Kawaguchi
Mashup @agustinvillena - @josephhurtado



<http://agilelion.com/agile-kanban-cafe/agile-and-lean-influences-where-did-kanban-scrum-scrumban-come-from>

Principles of process optimization and Agile

Key Agile Concepts



- **Individuals and interactions** over processes and tools



- **Customer collaboration** over contract negotiation



- **Respond to change** over following a plan



- **Work [business outcomes]** over comprehensive documentation

Core Lean Principles



Respect for People



Value - Customer-centricity



Flow



Responsiveness & Flexibility



Pull



Pursuit of Perfection

Some Common Lean & Agile Practices

Agile Practices

- Daily Standups
- Backlog (WIP or Work in waiting)
- Retrospectives

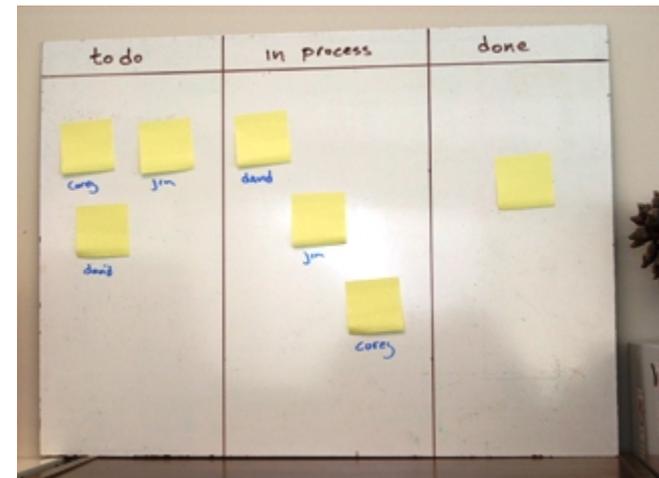


Some Lean Corollaries

- Daily Huddles
- Work Breakdown Structure
- Lessons Learned
- Continuous Improvement Plan

Case Study – Entertainment Client

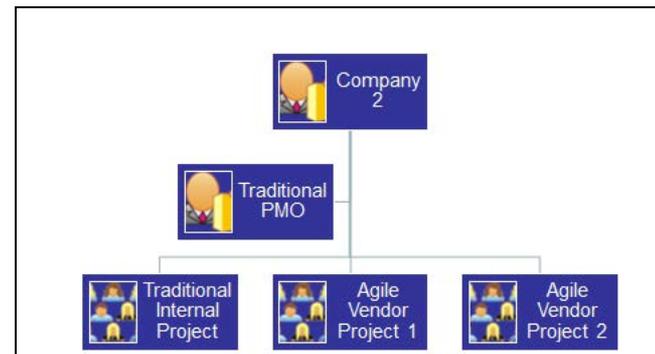
- Large Entertainment Client
- \$1 billion program, included physical properties and Web Site re-launch
- Outcome: Successful re-launch for this key program
- Scrumban was used within shared services groups that had frequent requests from VPs and executive team – “fly ins”
- Scrumban = Scrum + Kanban
- 500+ Person Agile Engagement



Source - <http://leansoftwareengineering.com>

Case Study – Financial Services Client

- Captive Financial Services Client (Automotive Manufacturer)
- Use Lean principles outside of the IT space
- Generally uses waterfall in the IT space
- Working to align internal projects with vendors that use Agile
- This is an Agile hybrid case – common in industry
 - In some cases by design, in some cases de-facto



Additional Case Studies

	Problem	Agile Practices Used	Outcome
Strategy Development	Looking for ways to deliver more value in a sustainable way and invert the pyramid to engage more senior leadership	<ul style="list-style-type: none">• Weekly Scrum Standups• Co-location• Timeboxes• Backlogs	Greater accountability and engagement achieved More solutions developed to deliver high quality and rich recommendations
Content Development	Looking for alternatives to delivering content at a more rapid and consistent pace in addition to managing production support	<ul style="list-style-type: none">• Scrum• Ideal Time• Kanban• Retrospectives	Ability to prioritize work in conjunction with production support issues. Ability to incorporate a pull system to bring additional work items into a sprint
Home Renovation	Looking for ways to renovate a home while not disrupting everyday living	<ul style="list-style-type: none">• Backlog• Epics• Timeboxes• Prioritization	Were able to breakdown the work into consumable chunks and lessen the inconvenience of rooms being out of commission

point**b**[®]



Agile Software Development Using Scrum

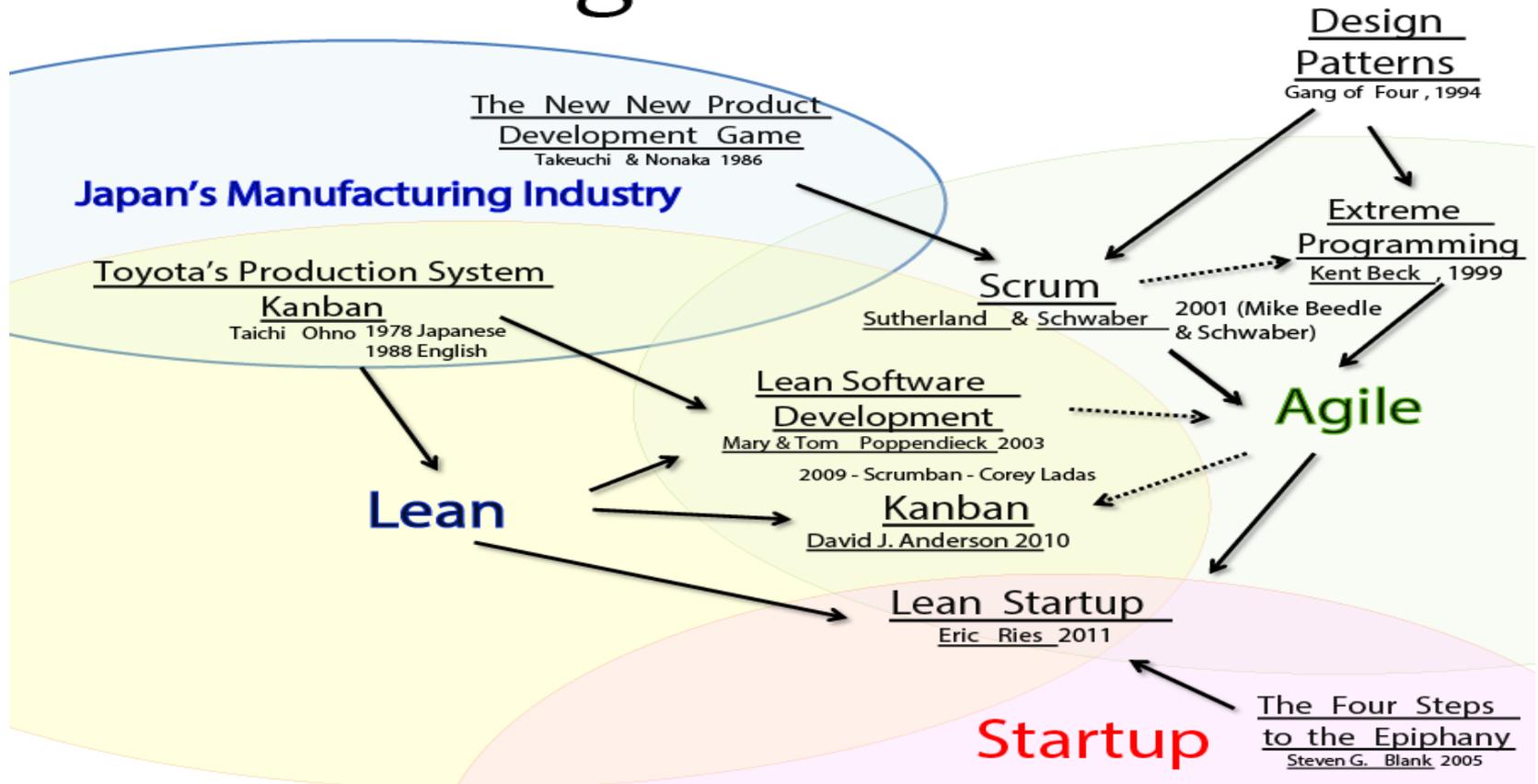
October, 2013

John Okoro – Point B Agile Service
Offering Lead – jokoro@pointb.com

Agile & Lean

Agile & Lean

2013 Yasunobu Kawaguchi
Mashup @agustinvillena - @josephhurtado

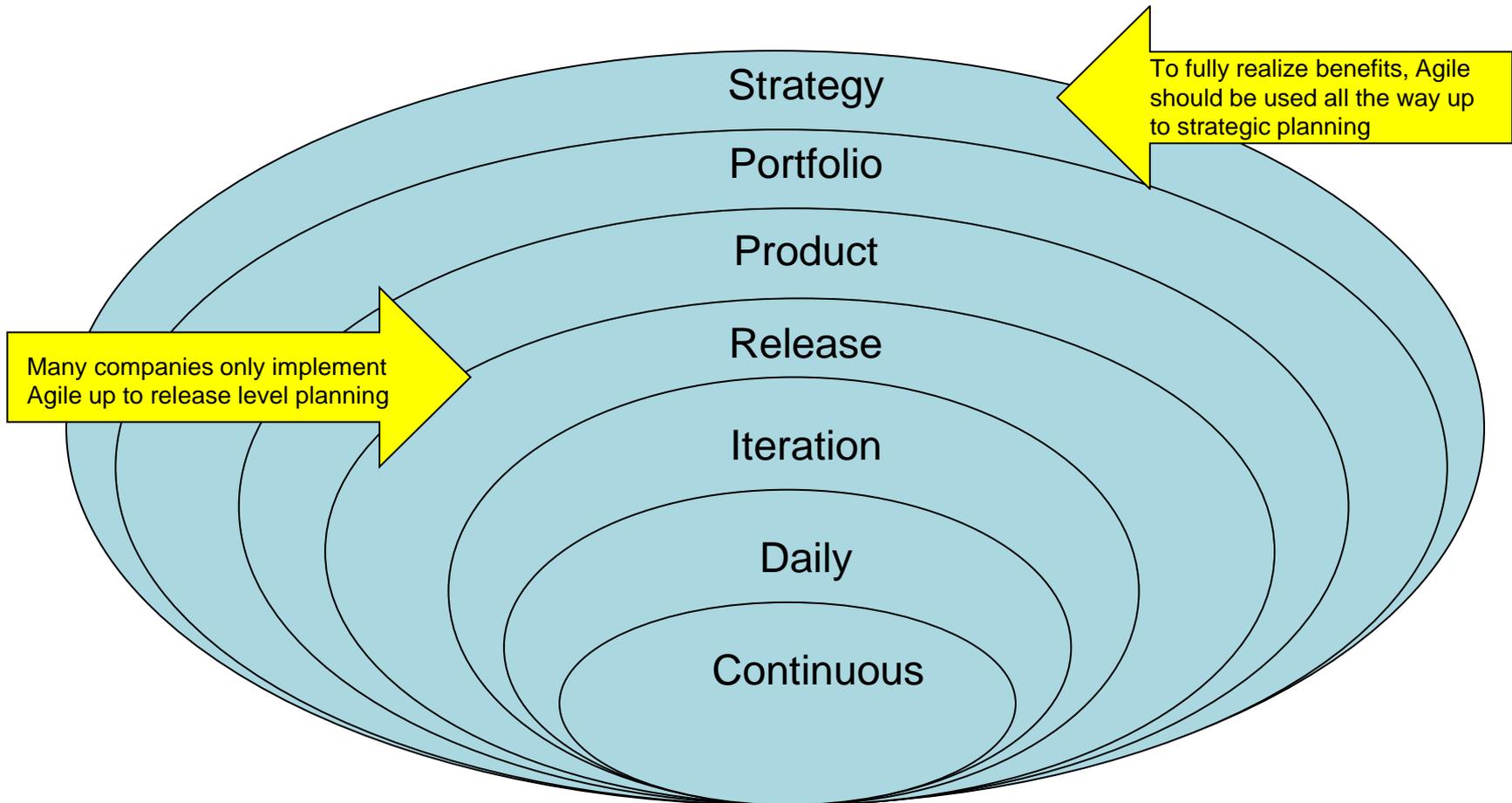


<http://agilelion.com/agile-kanban-cafe/agile-and-lean-influences-where-did-kanban-scrum-scrumban-come-from>

Contents

- Agile Planning
- Why Scrum?
- Scrum – in 30 Slides or Less :)
- Wrap Up

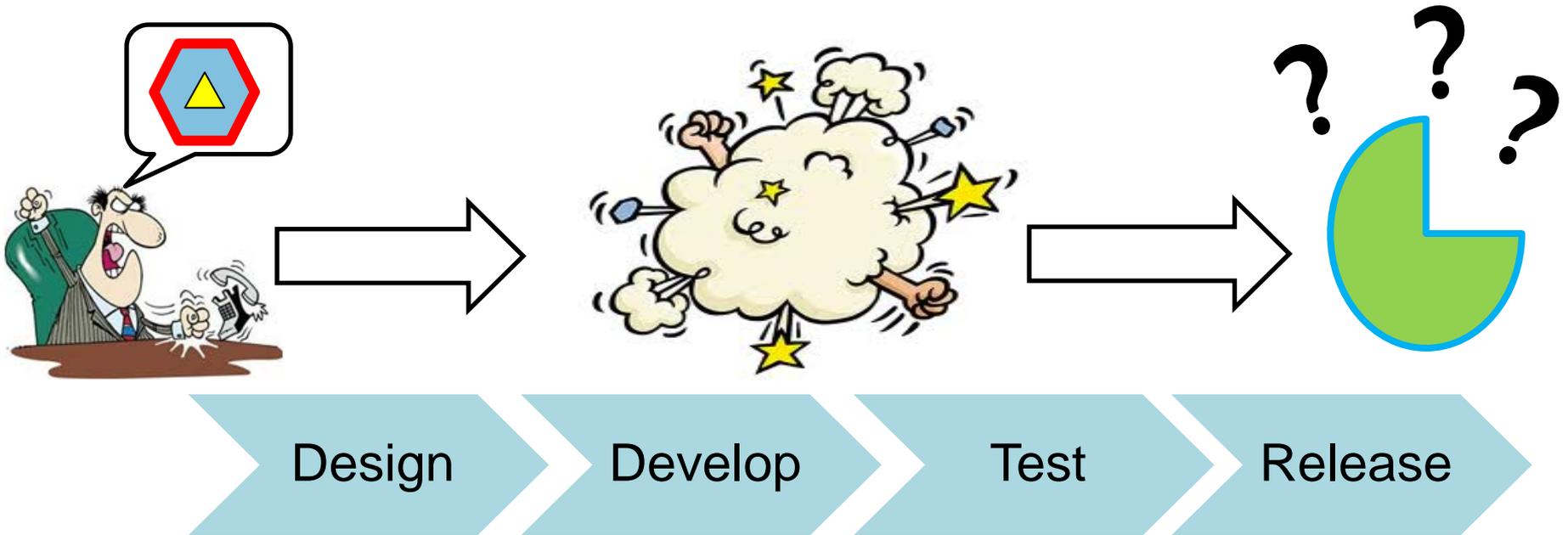
Agile Planning



Why Scrum?

- First, some truths about software development:
 - Building software rarely means building repeatable widgets
 - It's difficult to plan a project that's more than a few weeks, since you don't know when you'll encounter something unexpected, or when business priorities will change
- For a **product manager**, Scrum permits:
 - Frequent changes in scope
 - Flexible changes to a project's end-date
- For a **development team**, Scrum permits:
 - A guarantee that scope won't change for a predictable period of time
 - A team to self-organize as opposed to being told what to do
- For an **organization**, Scrum permits:
 - Quality software that more reliably meets the needs of the organization
 - Software projects that can be delivered on time

A Typical Project



Giving Credit Where It's Due...

- The bulk of the following slides have been borrowed in full or in part from Mike Cohn (with Mike's permission, of course...)
- Mike Cohn:
<http://www.mountangoatsoftware.com/>

What is Scrum

- Scrum is one of the many flavors of Agile
 - Scrum is the flavor of Agile used by many Point B clients
- Scrum was started by Jeff Sutherland in 1993
 - He borrowed the term Scrum from a 1986 study by Takeuchi & Nokata Published in the HBR
 - The HBR study compares high performing cross functional teams to the Scrum formation used in Rugby
- Scrum is the leading Agile development framework used by Agile development teams around the world
- There are several other flavors of Agile including XP, AUP, *Kanban*



The Agile Manifesto – a Statement of Values

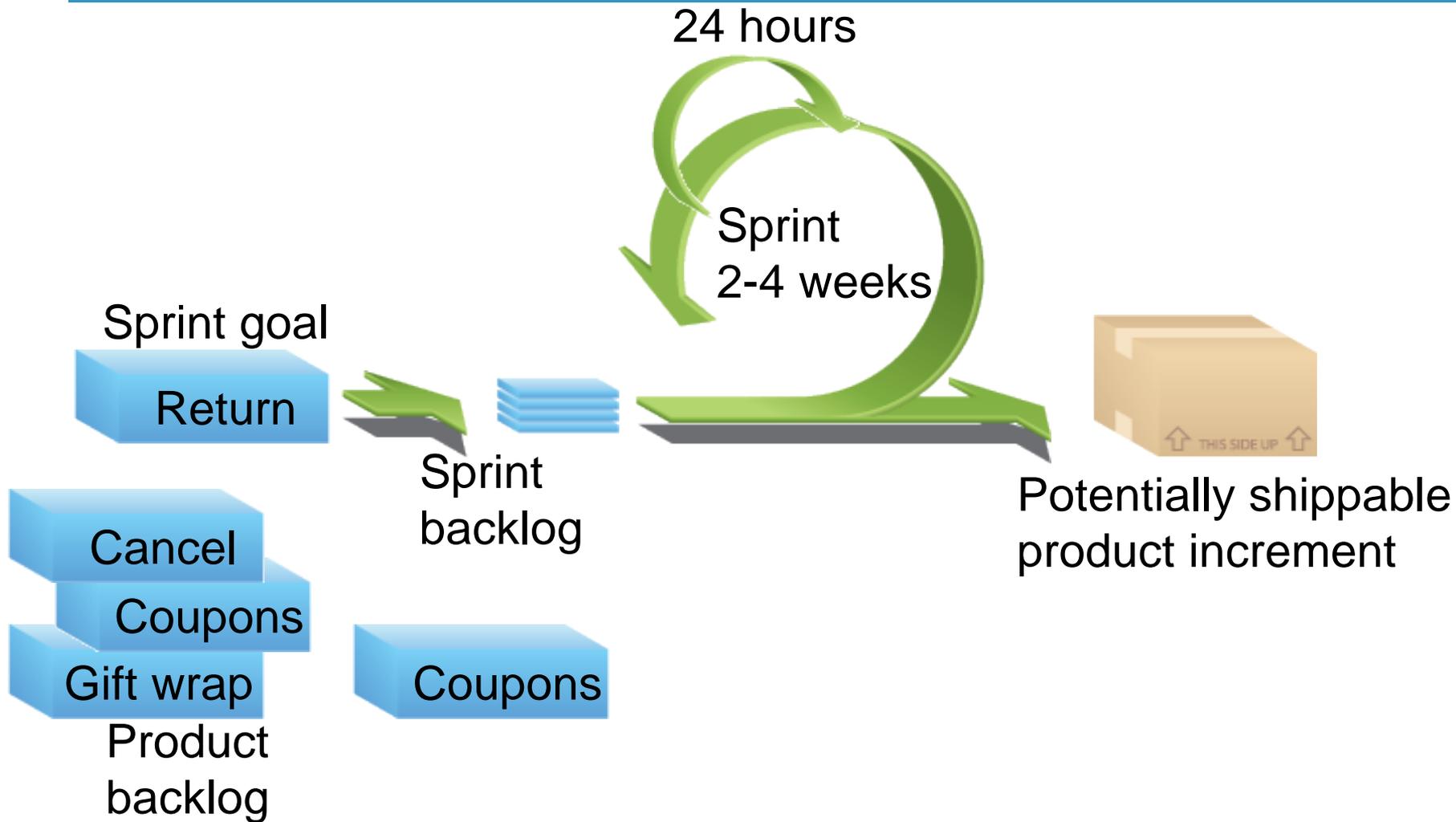
Individuals and Interactions	over	Process and Tools
Working Software	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to Change	over	Following a Plan

Source: www.agilemanifesto.org

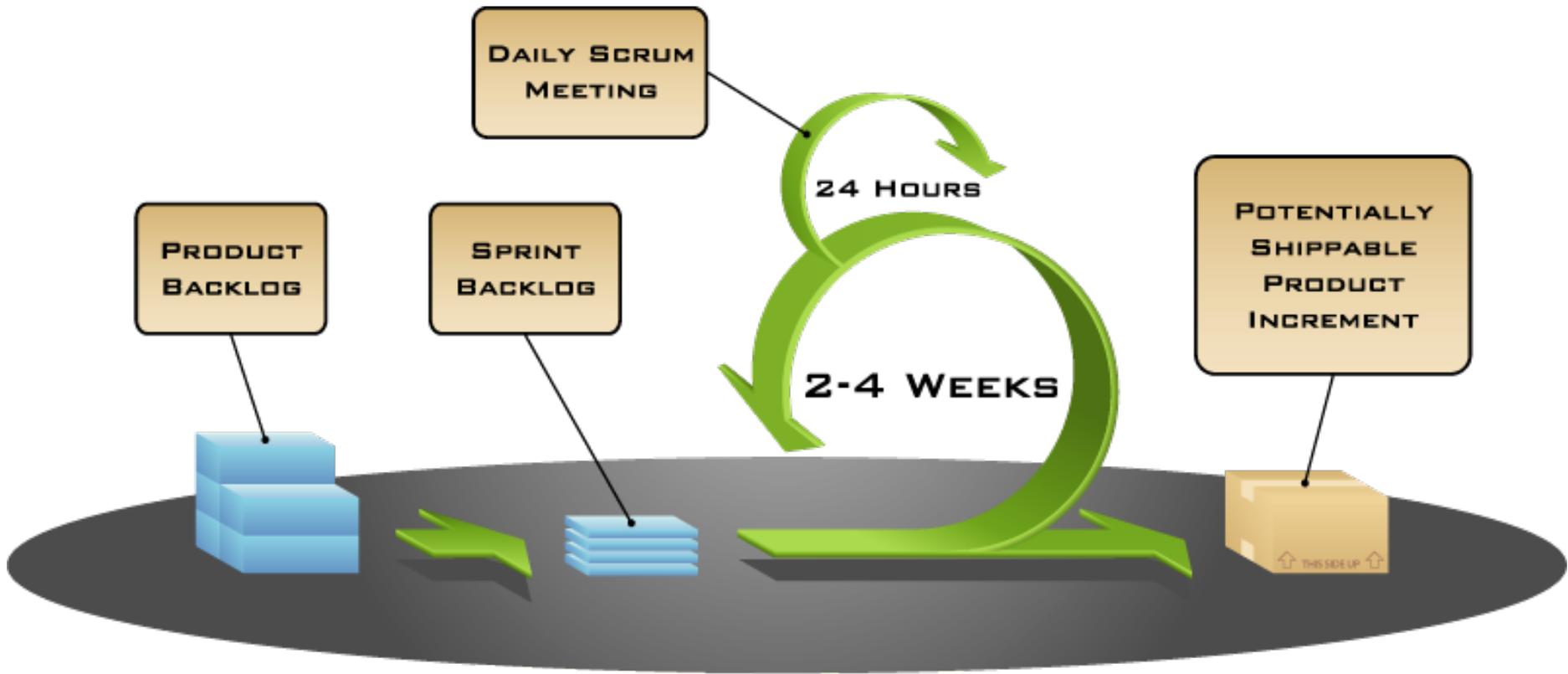
Sprints

- Scrum projects make progress in a series of “sprints”
 - Analogous to Extreme Programming iterations
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

Scrum Lifecycle



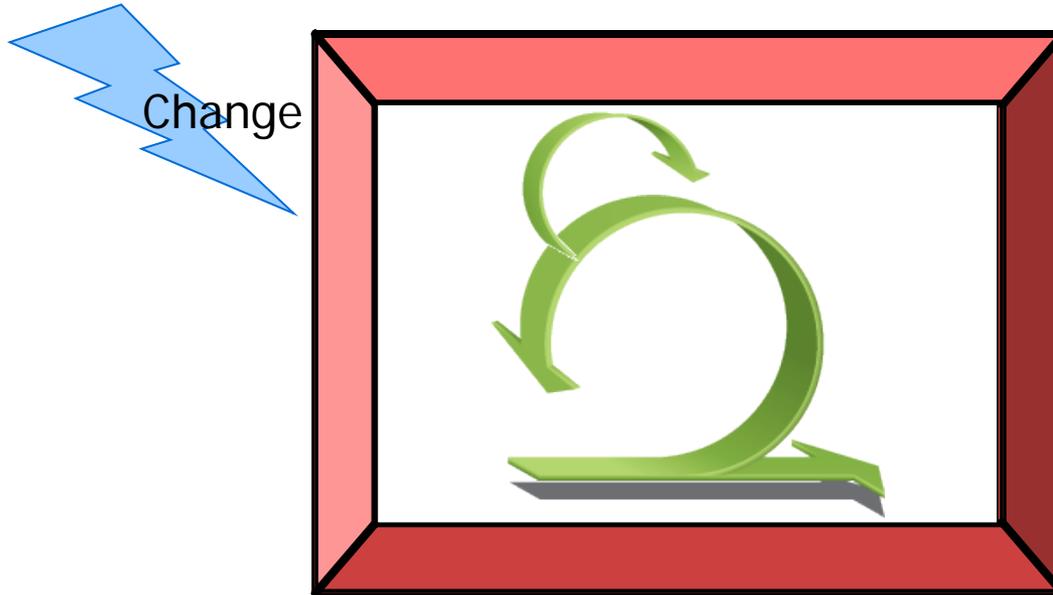
Putting it All Together



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Image available at www.mountaingoatsoftware.com/scrum

No Changes During a Sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

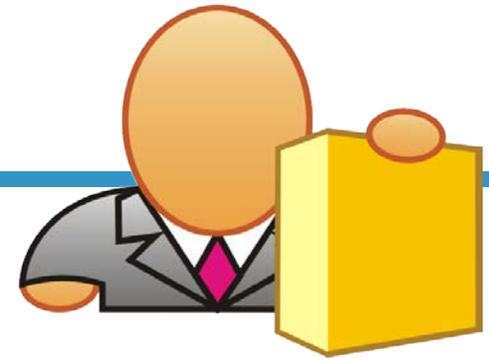
Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

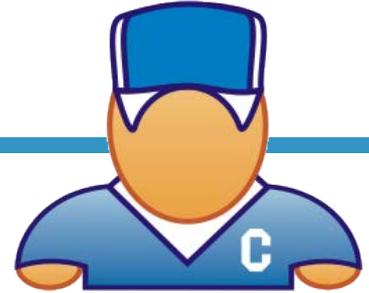
- Product backlog
- Sprint backlog
- Burndown charts

Product Owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

The ScrumMaster



- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

The Team



- Typically 5-9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
- Members should be full-time on the project
 - May be exceptions (e.g., database administrator)
- Teams are self-organizing
 - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Everyone Answers 3 Questions

1

What did you do yesterday?

2

What will you do today?

3

Is anything in your way?

- These are *not* status for the ScrumMaster
 - They are commitments in front of peers

Scrum Framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

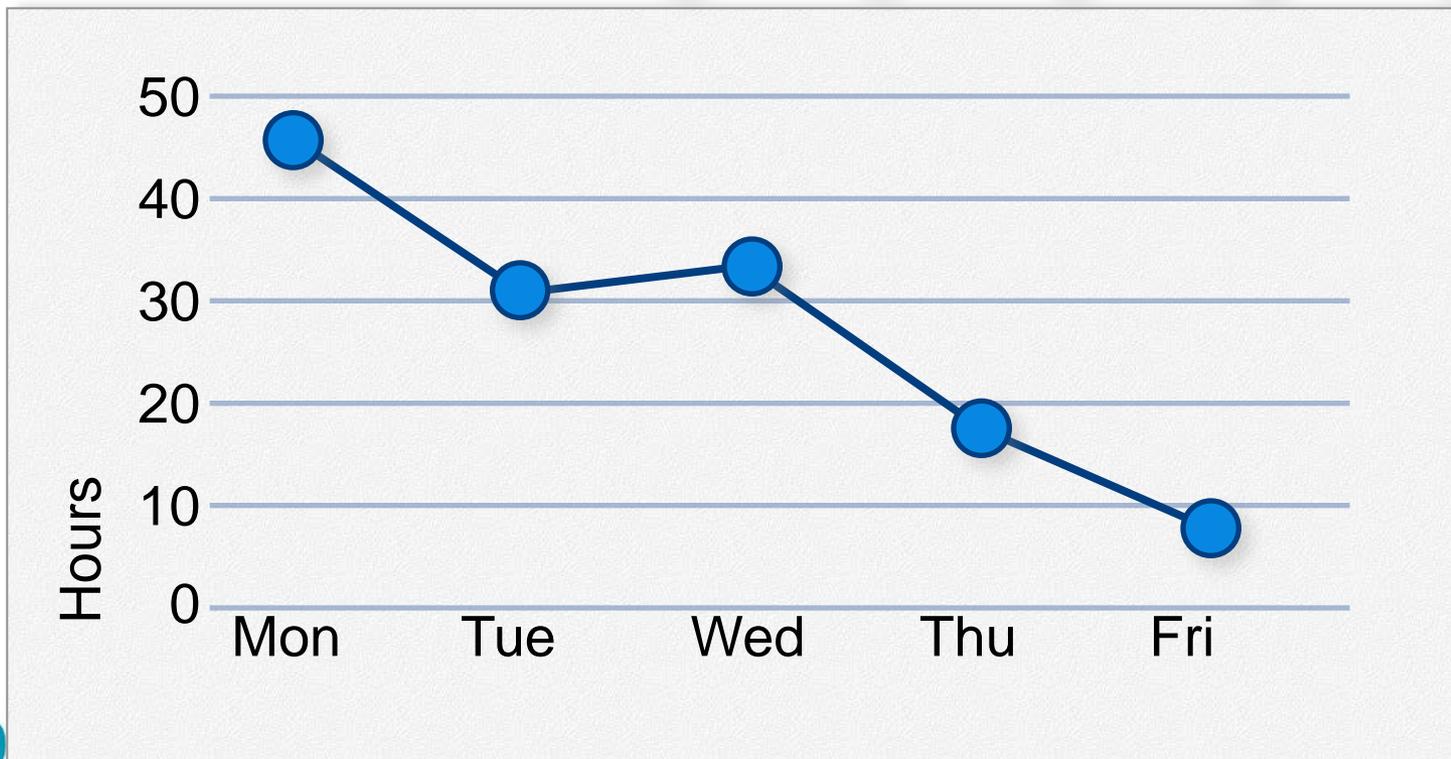


This is the
product backlog

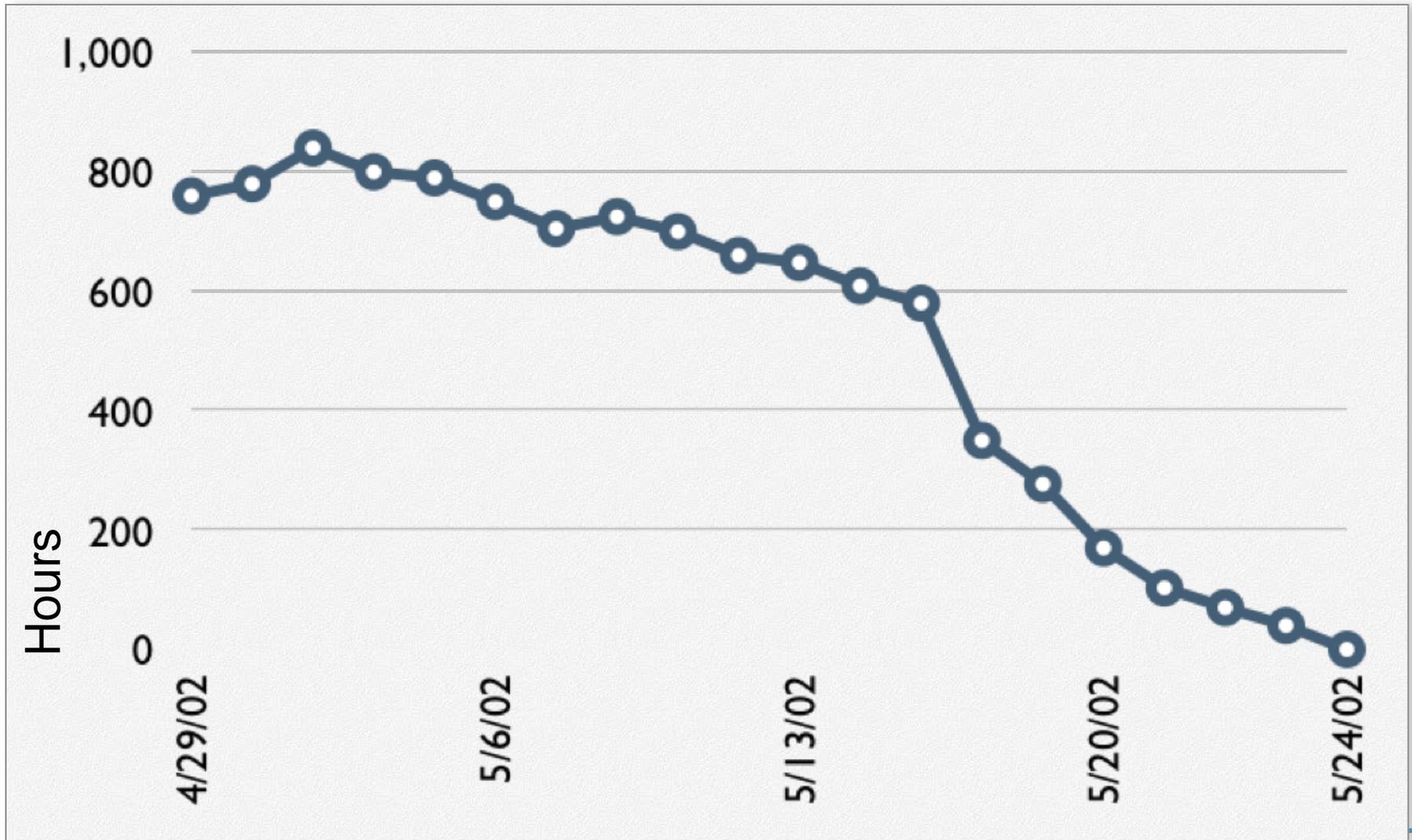
A Sprint Backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

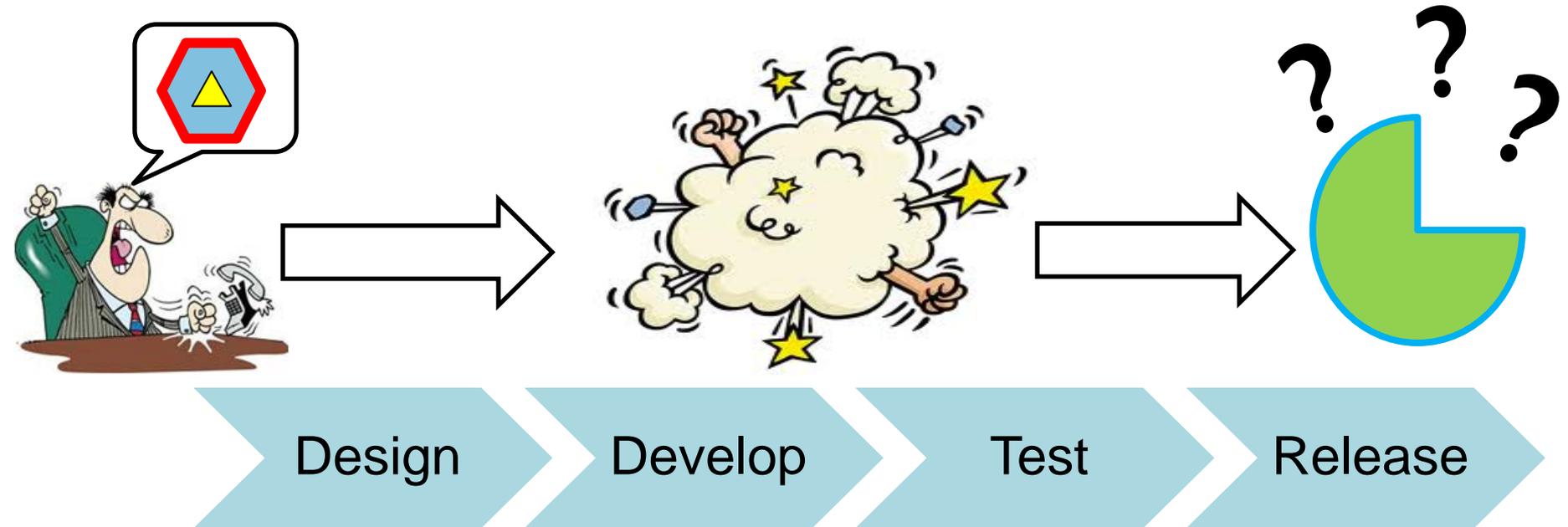
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



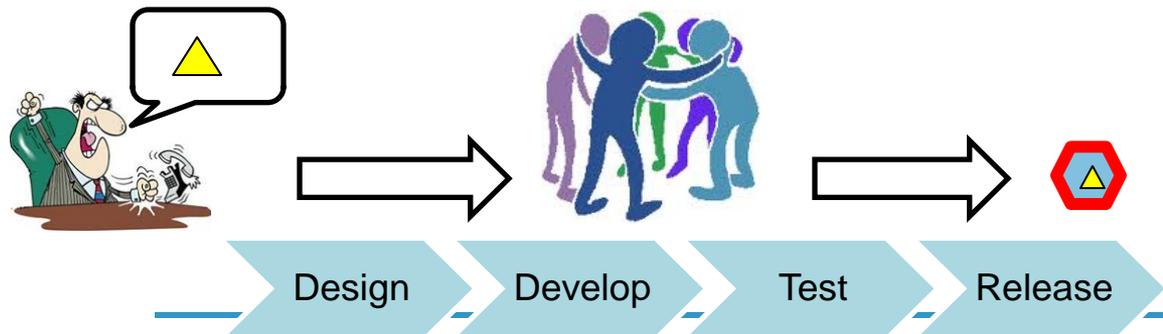
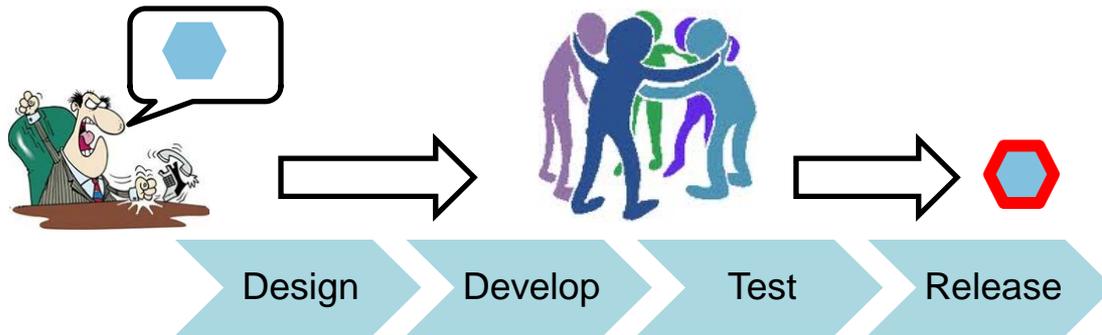
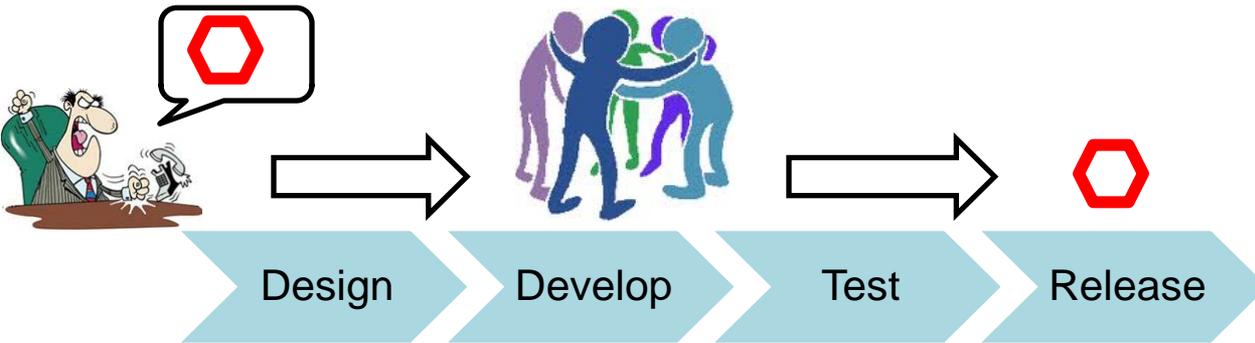
A Sprint Burndown Chart



Remember this?



How about this instead?



Wrap Up

- Why Scrum?
 - **Focus** – Sprints allow teams to focus on an achievable goal
 - **Ownership** – Self-organization foments a feeling of ownership of the product
 - **Transparency** – everyone can see the requirements and the progress, which allows for course corrections to happen before it's too late
 - **Continuous Improvement** – It's ok to re-do what wasn't done right the first time. Retrospectives cause teams to constantly look for areas of improvement.
 - **It's All About the Team** – over time, team members develop respect for one another and are more willing to pitch-in to reach a common goal

Resources

- Where to learn more:
 - Scrum Alliance: <http://www.scrumalliance.org/>
 - Mike Cohn's Site: <http://www.mountangoatsoftware.com/topics/scrum>
 - Point B Agile Thought Leadership & Case Studies : <http://www.pointb.com>
 - Wikipedia ([Agile](#), [Scrum](#))
 - YouTube (lot's of great short videos)
 - Books:
 - Agile Software Development with Scrum (Schwaber/Beedle)
 - Agile Estimating and Planning (Cohn)
 - Agile Project Management: Creating Innovative Products (2nd Edition - Highsmith)
 - Lean IT: Enabling and Sustaining Your Lean Transformation (Bell)

point **b**.

Appendix

Resources

- PointB.com Web Site – <https://pointb.com>
- Agile Service Offering Team: John Okoro, Alex Stockdale, Pat Edmonds (Lean), Scott Ikeda
- Agile Case Study - <http://www.pointb.com/consulting-services/project-leadership/case-studies/applying-agile-methods-in-support-of-creating-a-world-class-guest-experiencc>
- Agile Thought Leadership - <http://www.pointb.com/latest-insights/agile-beyond-the-it-department>

The Business Environment

- Many organizations have embraced Agile for their software development practices
- A growing percentage of these organizations are starting to use it outside of the IT department
- Application of Agile practices in non-IT environments include:
 - Daily stand-ups
 - Backlog
 - Retrospectives
- With a slight tweak of the Agile Manifesto from software to **business outcomes**, the intent behind the manifesto becomes relevant to most implementation-type projects

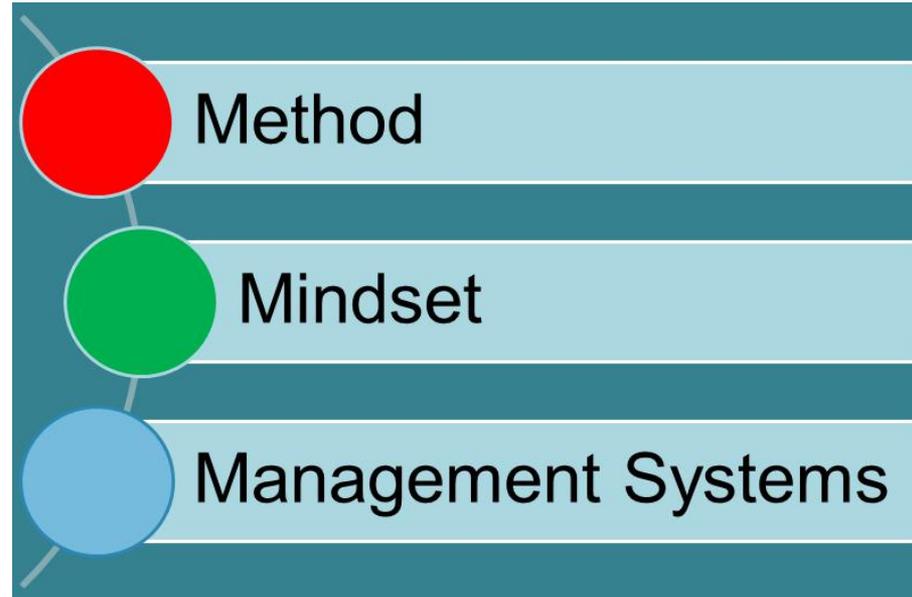
Agile Principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

<http://agilemanifesto.org/principles.html>

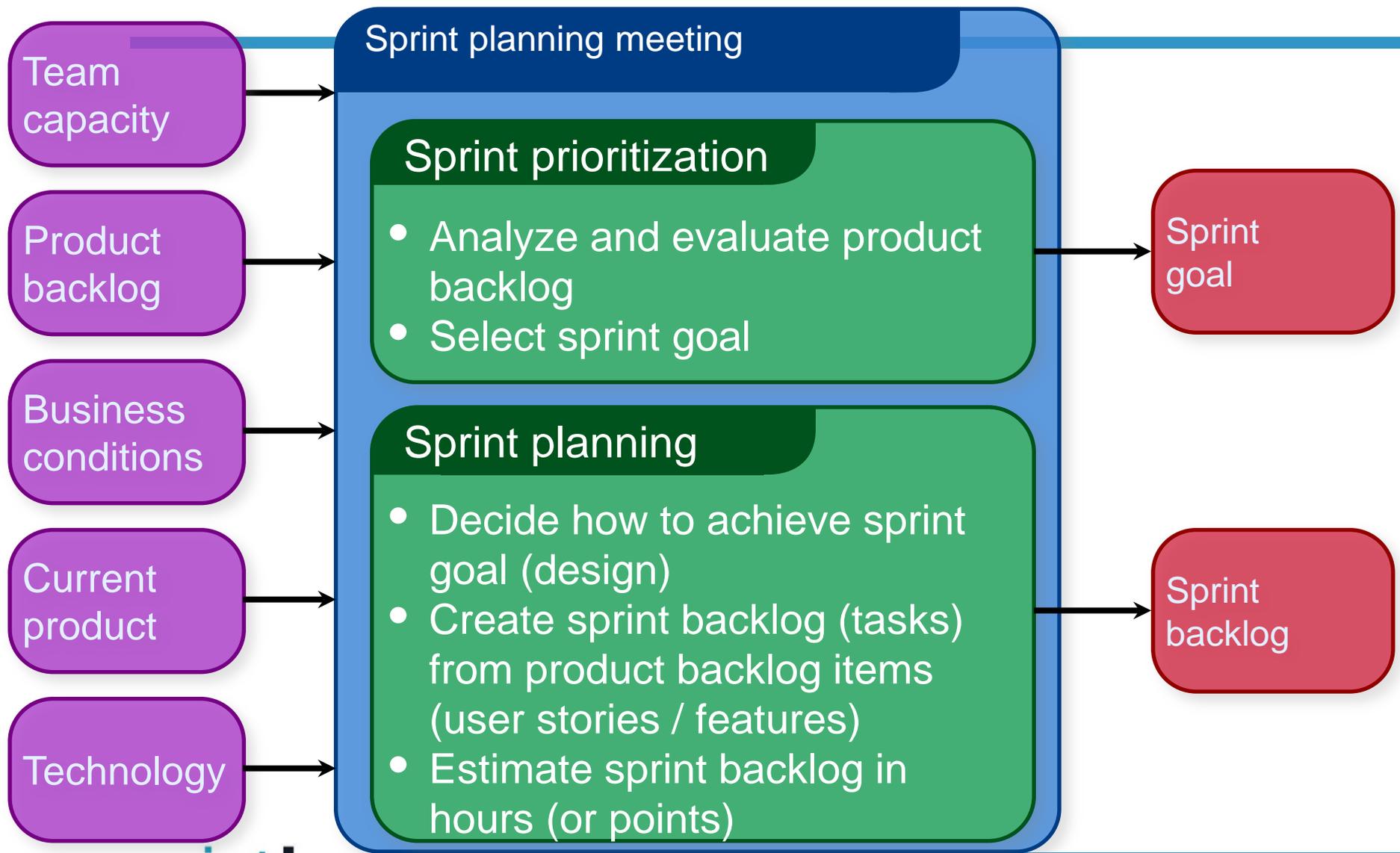
What is Lean?

A **philosophy, mindset** and **set of tools** focused on **delivering value** to **customers** through the **elimination of waste** in the business process



Scrum in 100 Words

- Scrum is an agile process that allows us to focus on delivering the **highest business value** in the **shortest time**.
- It allows us to rapidly and repeatedly inspect **actual working software** (every two weeks to one month).
- The **business sets the priorities**. Teams **self-organize** to determine the best way to **deliver the highest priority features**.
- Every **two weeks to a month** anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.
 - For creative teams this may mean delivering demonstrable experiences (working deliverables)



Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
 - Tasks are identified and each is estimated (1-16 hours or points)
 - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

The Daily Scrum

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
- Not for problem solving
 - Whole world is invited
 - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings



Everyone Answers 3 Questions

1

What did you do yesterday?

2

What will you do today?

3

Is anything in your way?

- These are *not* status for the ScrumMaster
 - They are commitments in front of peers

The Sprint Review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
 - 2-hour prep time rule
 - No slides
- Whole team participates
- Invite the world



Sprint Retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
 - ScrumMaster
 - Product owner
 - Team
 - Possibly customers and others

Start / Stop / Continue

- Whole team gathers and discusses what they'd like to:

Start doing

Stop doing

This is just one of many ways to do a sprint retrospective.

Continue doing

User Stories

- **What is it?** A short, simple description of a feature told from the perspective of the actor who desires the new capability.
- **Template:** As an <actor> I want to <goal> so that <reason or intent>.
- **Example:** As a Customer, I want to have my product gift-wrapped so that when I buy a gift for someone, it appears personalized and special
- Notes:
 - Focus is always on providing business value
 - Usually considered the start of a conversation and are not intended to contain all requirements for a feature
 - Assume that IAs/designers/developers/testers are experienced enough to make good design decisions on the details
 - Epics are used for large features and should contain many corresponding user stories

A Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

The Sprint Goal

- A short statement of what the work will be focused on during the sprint

Database Application

Make the application run on SQL Server in addition to Oracle.

Life Sciences

Support features necessary for population genetics studies.

Financial services

Support more technical indicators than company ABC with real-time, streaming data.

Managing the Sprint Backlog

- Individuals sign up for work of their own choosing
 - Work is never assigned
- Estimated work remaining is updated daily
- 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

Estimating Work

- Estimating the time it takes to build software or design a UI is really difficult to *nail*
- ... But it's not so tough to get it *pretty close* to right
- Estimating to achieve perfection can waste a ton of time
- So instead of spending all your team's time estimating, let them take their best guesses and see how things work out
- Patterns will emerge...
- Treat estimates as **best guesses** rather than **contracts**

Challenges Adopting Scrum

- Org structure, support and sponsorship
 - Scrum adoption requires executive sponsorship as well as complete buy-in from Creative/dev/test/product mgmt teams
- Physical environment
 - Scrum values communication, putting a team in the same room is best, but often can't be achieved
- Geographical location
 - In today's world of distributed teams, time zones and distance can make it difficult to communicate
- External customers
 - Implementing Scrum when the Product Owner is outside of your company can be difficult – distance, contracts and trust can get in the way.
- Scope
 - Letting go of the notion of fixed scope throughout the course of the project