

Agile Software Process

Presented by Wiwat V.

The Linear Sequential Model

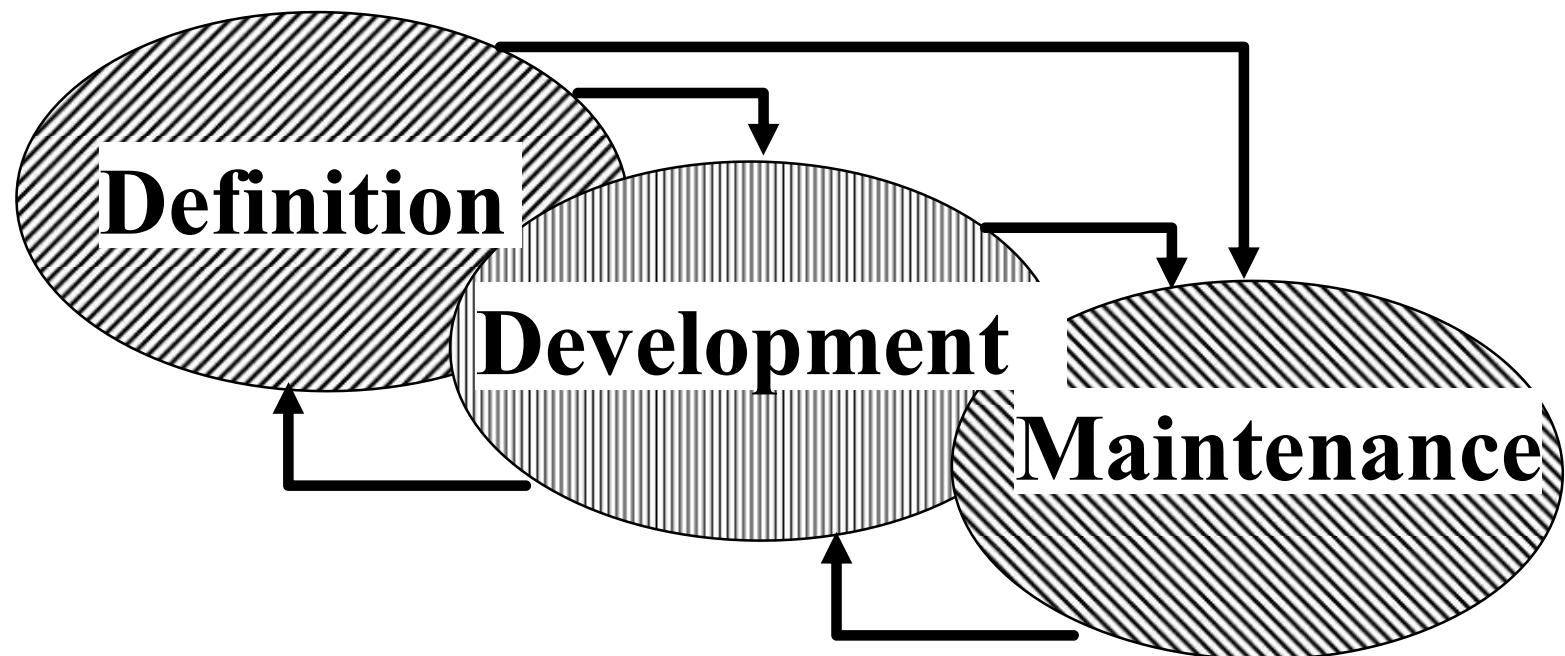
- Classical Life Cycle or Waterfall model consists of activities:
 - System/information and engineering modeling
 - Software Requirements Analysis
 - Design
 - Code Generation
 - Testing
 - Support

ใหม่โรงโดยการเล่าความเป็นมาก่อน

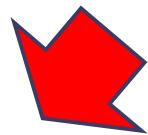
Q: Life cycle หนึ่ง ๆ ยawananแค่ไหน?



A Generic View of Software Process



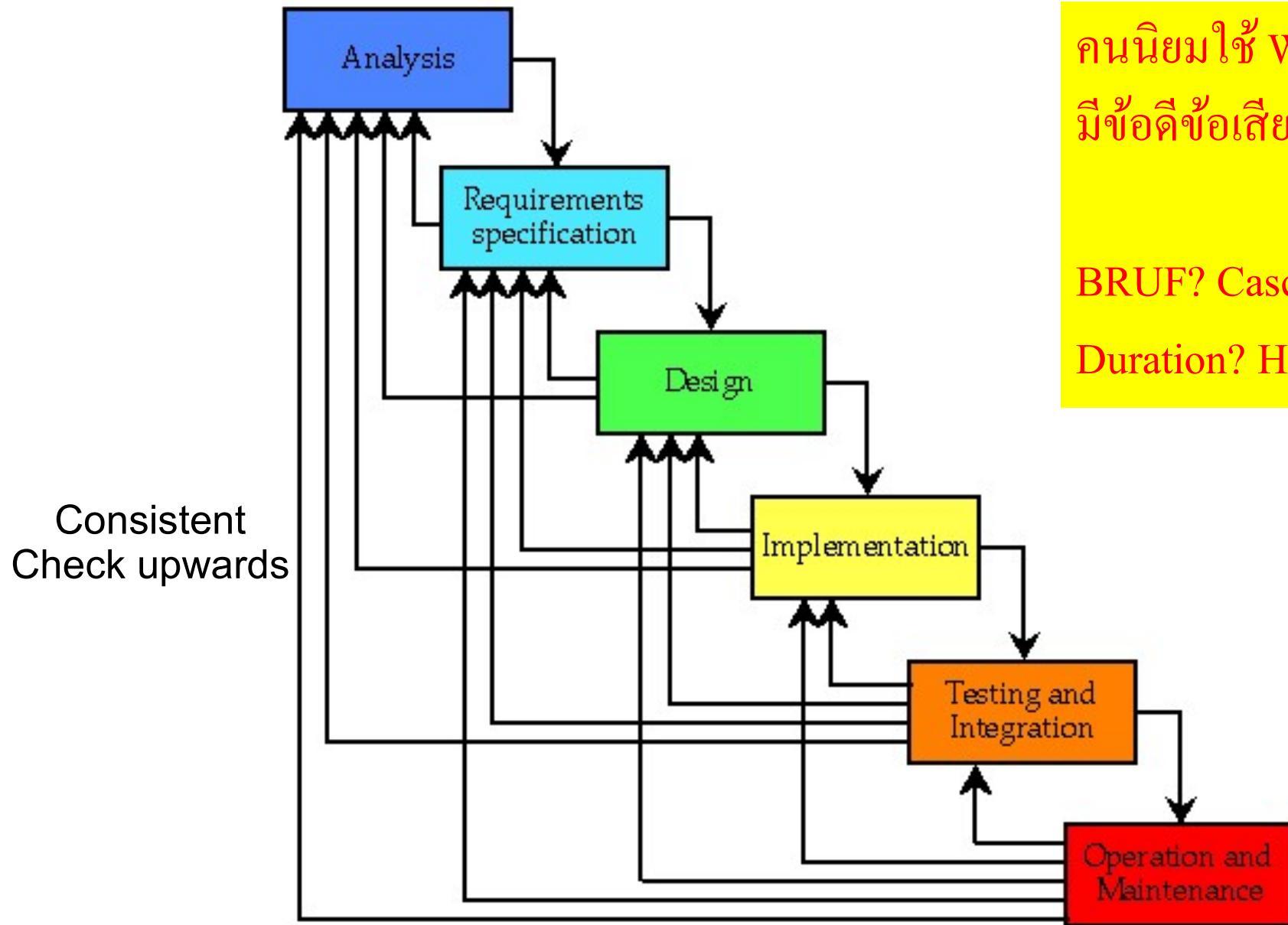
- | | | |
|---------------|----------|---------------|
| ■ Definition | focus on | WHAT |
| ■ Development | focus on | HOW |
| ■ Maintenance | focus on | Change |



Waterfall Model

- Discussed by Royce since 1970
- The stages are depicted as cascading from one to another
- One stage should be completed before the next begins
- Has been used in variety context, for example it was a basis for deliverables in U.S. Department of Defense Contracts and defined in DOD2167-A
- Deliverables are produced in each process
- The model will be very useful in helping developers lay out what they need to do

Waterfall Model



คนนิยมใช้ Waterfall?

มีข้อดีข้อเสียอย่างไร

BRUF? Cascading?

Duration? HR?

Drawbacks of the Waterfall Model

- It shows how each phase terminates in the production of some artifact, but there is no insight into how each activity transforms one artifact to another
- It provides no guidance on how to handle changes to products and activities
- It is too late to show the customers what the product looks like

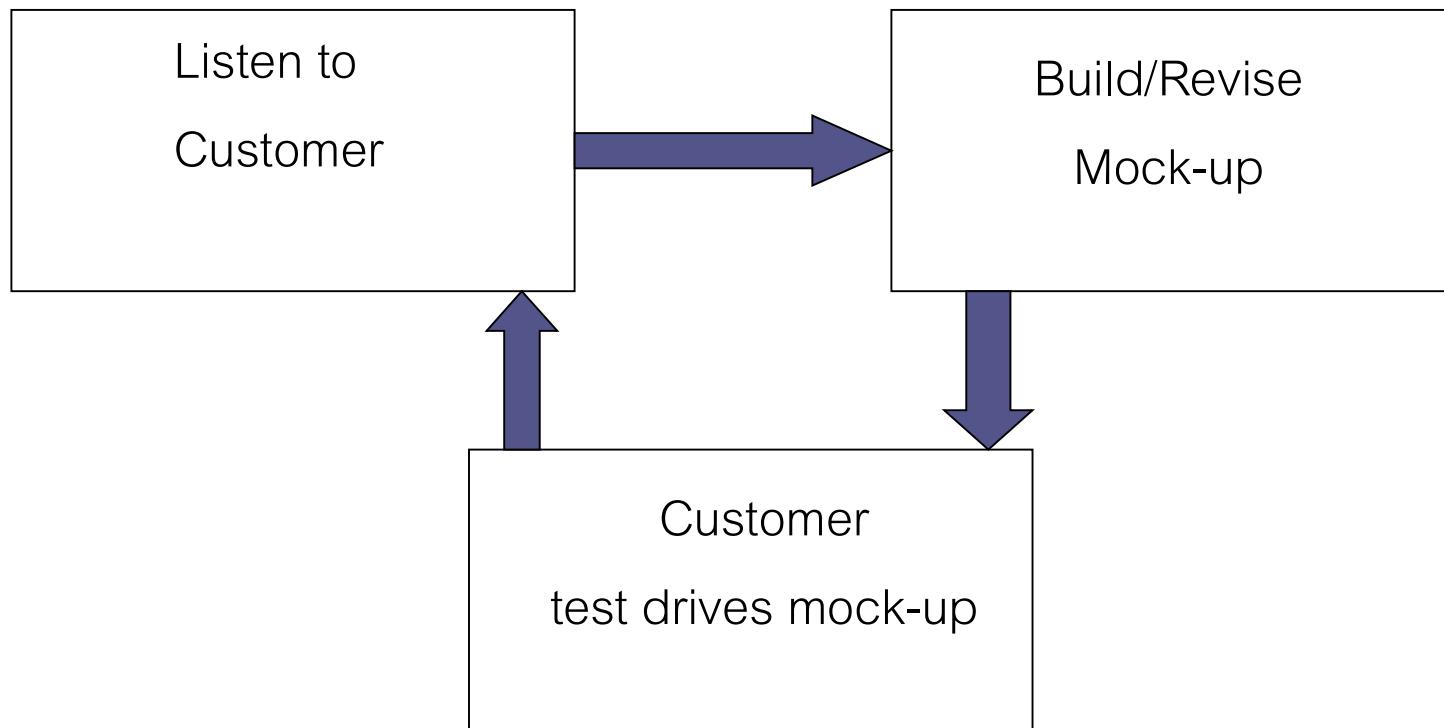
เพราฯไม่คาดว่าจะต้องมี change
เนื่องจากทุก phase ต้องทำดีที่สุด ไม่ย้อนกลับ[↑]
แต่ชีวิตจริงอาจจะไม่ใช่

Factors preventing reliable up-front specifications

- The clients or users are not sure what they want.
- They have difficulty stating all they want and know.
- Many details of what they want will only be revealed during development.
- The details are overwhelmingly complex for people.
- As they see the product develop, they change their minds.
- External forces (such as a competitor's product or service) lead to changes or enhancements in requests.

Up-Front Spec. กลับกลายเป็นปัญหาใหญ่

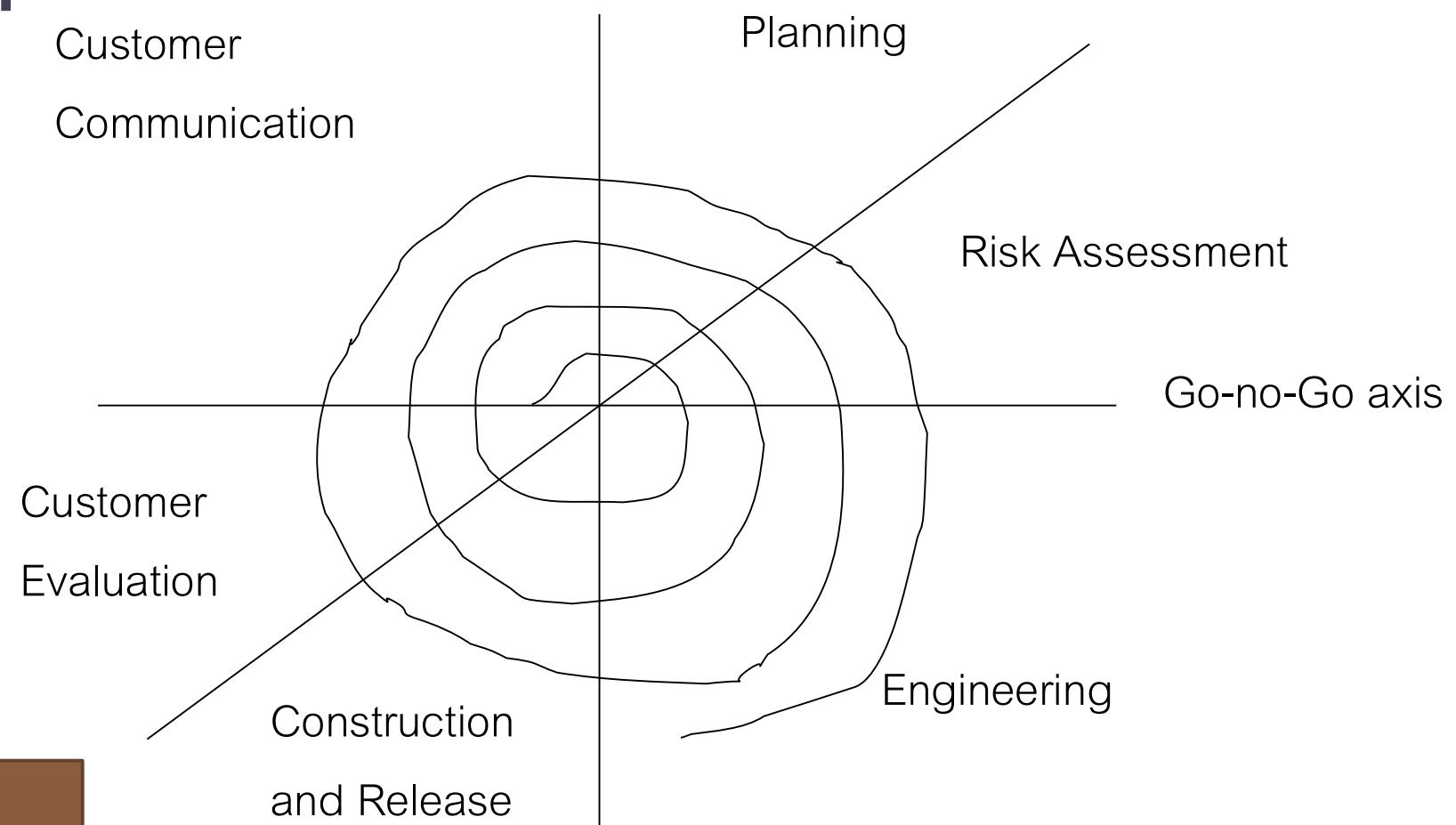
Prototyping Model



Introduce a Mock-Up
of each functionality
to be developed

สาเหตุที่ spec. กำหนดได้ยาก เพราะความไม่รู้ว่าจะได้อะไร...
ทางออกหนึ่งคือสร้าง Mock-up เพื่อนำทาง
และให้ User เริ่มรับรู้และ Feedback

The Spiral Model



**Introduce
Risk
Assessment**

Four Essential Best Practices!!



Some Basic Practices to Software Process



- Timeboxing
 - Fixing the development time intervals to 1-6 weeks for each timebox
- Adaptive
 - Adapt in response to feedback from prior work – feedback from users, tests, developers, etc.
- Iterative
 - describes and governs each move we make in almost every aspect of turning ideas into products
- Incremental
 - “That feature will be available of the next *increment* of the product.”

รวมเทคนิคเหล่านี้เข้ากับ Software Process เดิม

When it comes to AGILE!

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

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

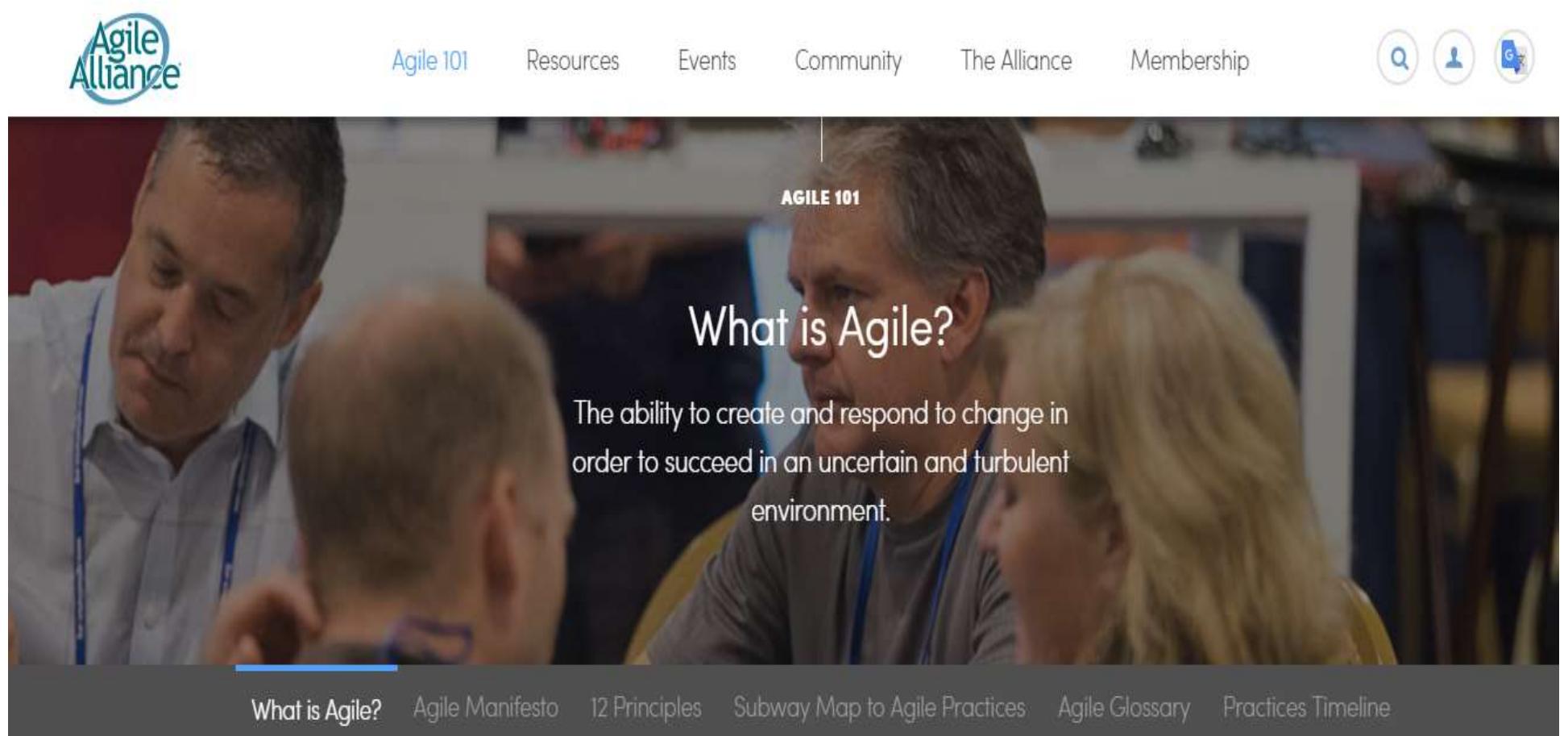
Jeff Sutherland

Dave Thomas

What is Agile

- Agile = “วงศิริ แข็งขัน กระฉับกระเฉง”
- Agile = “Rapid and flexible response to change”

Agile Alliance (<http://www.agilealliance.org>)



The screenshot shows the Agile Alliance website homepage. At the top, there is a navigation bar with links for "Agile 101", "Resources", "Events", "Community", "The Alliance", and "Membership". To the right of the navigation bar are three circular icons: a magnifying glass for search, a person for membership, and a speech bubble for community. The main content area features a large image of several people in a workshop or meeting setting. Overlaid on this image is the text "AGILE 101" and "What is Agile?". Below this, a definition is provided: "The ability to create and respond to change in order to succeed in an uncertain and turbulent environment." At the bottom of the page, there is a navigation bar with links for "What is Agile?", "Agile Manifesto", "12 Principles", "Subway Map to Agile Practices", "Agile Glossary", and "Practices Timeline".

AGILE 101

What is Agile?

The ability to create and respond to change in order to succeed in an uncertain and turbulent environment.

What is Agile? Agile Manifesto 12 Principles Subway Map to Agile Practices Agile Glossary Practices Timeline

What is Agile Software Development?

Manifesto for Agile Software Development

- **Individuals and interactions** over processes and tools
 - **Working software** over comprehensive documentation
 - **Customer collaboration** over contract negotiation
 - **Responding to change** over following a plan
-
- 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, John Kern, Dave Thomas, Martin Fowler and Brian Marick (in 2001)



Alistair Cockburn
– Crystal Clear



James Grenning –
planning poker



Kent Beck
- XP



Martin Fowler –
Architectural Patterns



Robert C. Martin
- Clean Architecture



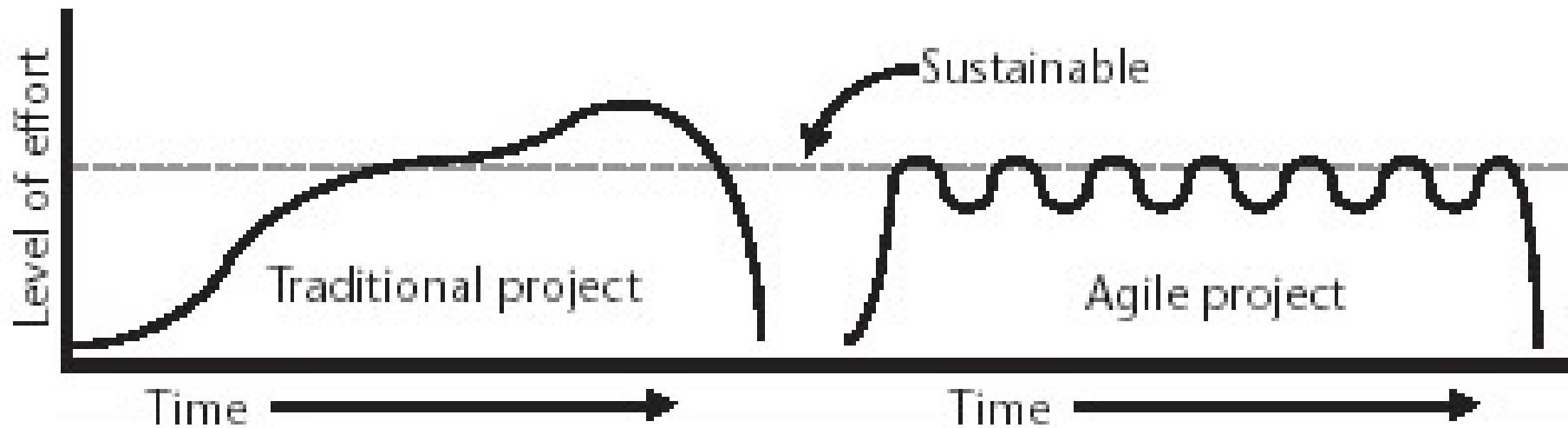
Ken Schwaber - Scrum

12 Principle of Agile Software

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

Sustainable Pace

ตัวอย่างการอธิบาย
Agile Principle

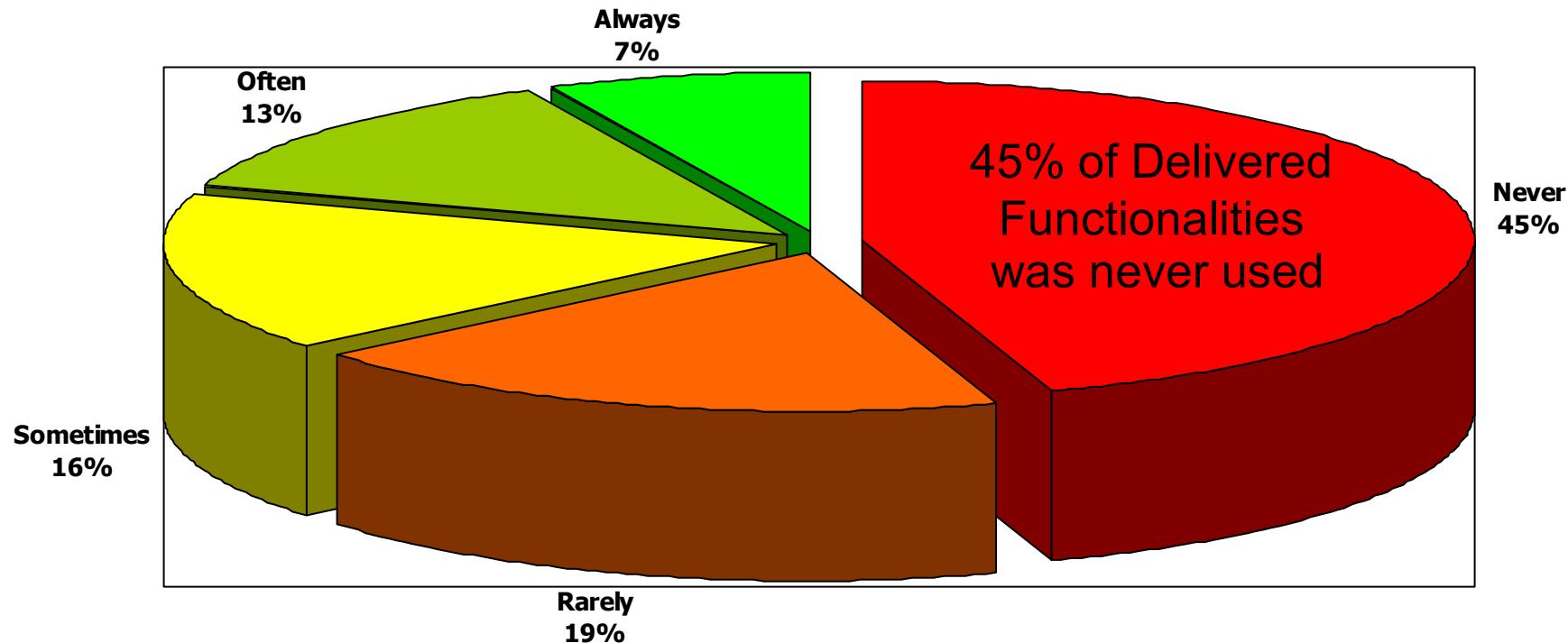


Agile Documentation

- Travel light – You need far less documentation than you think
- Agile documents:
 - Maximize stakeholder investment
 - Are concise
 - Fulfill a purpose
 - Describe information that is less likely to change
 - Describe “good things to know”
 - Have a specific customer and facilitate the work efforts of that customer
 - Are sufficiently accurate, consistent, and detailed
 - Are sufficiently indexed
- Valid reasons to document:
 - Your project stakeholders require it
 - To define a contract model
 - To support communication with an external group
 - To think something through

The Cost of Traditional BRUF (Big Requirements Up Front)

“Successful” Projects Still Have Significant Waste



BRUF

ນັກເປັນປໍລູຫາ

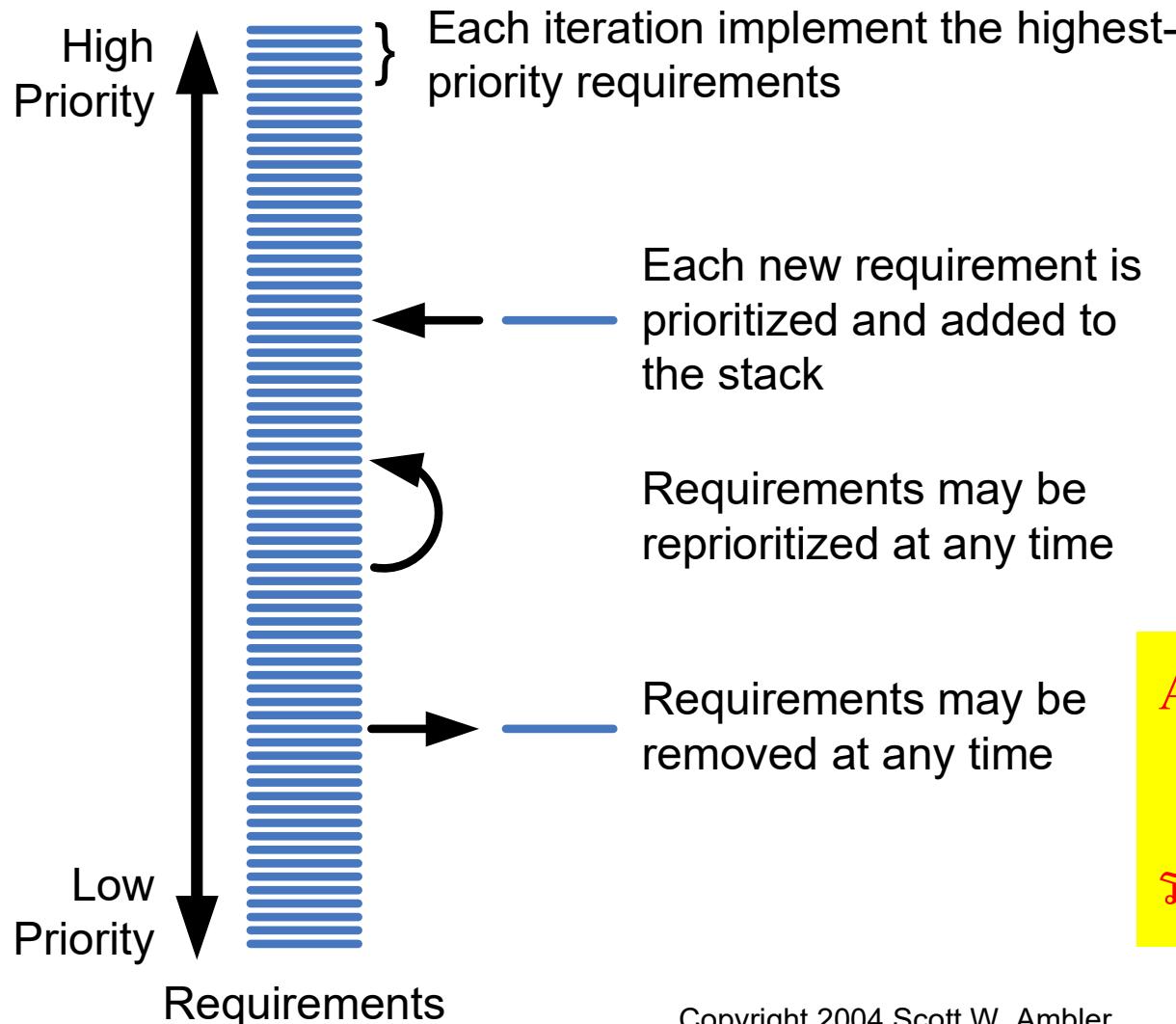
Source: Jim Johnson of the Standish Group, Keynote Speech XP 2002

Big Requirements Up Front (BRUF) Approach

- Typically, a software development lifecycle that is mostly serial in nature should yield:
 - A big requirements up front (BRUF) approach is the approach where you create a detailed requirements specification early in the lifecycle
 - A change management process is needed to avoid "scope creep" throughout the project
 - A big design up front (BDUF) approach is the approach where you attempt to specify your architecture in detail before coding begins

Agile Software Requirements Management

Changing Requirements Are a Competitive Advantage if You Can Act on Them



Agile Software Process
มักจัดการ Req. เพื่อ
รองรับการเปลี่ยนแปลง

Active Stakeholder Participation

- Project stakeholders should:
 - Provide information in a timely manner
 - Make decisions in a timely manner
 - Actively participate in business-oriented modeling

ดึง Active Stakeholder เข้ามา มีส่วนร่วม ด้วยกัน
และ มีส่วน ได้ส่วน เสีย⁺
นิทาน “Pig and Chicken”

Model With Others

- The modeling equivalent of pair programming
- You are fundamentally at risk whenever someone works on something by themselves
- Several heads are better than one

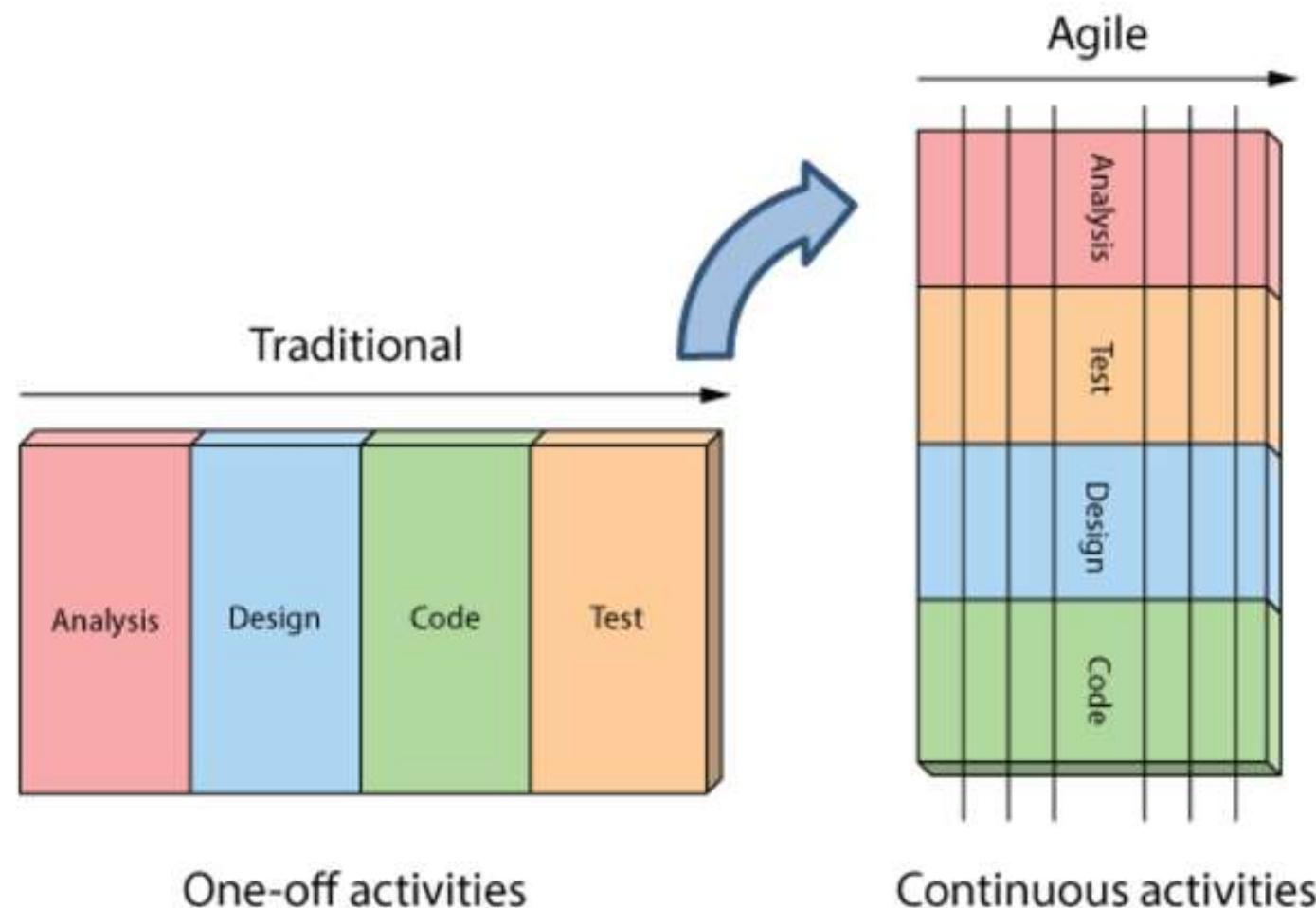
Non-Solo Development

Agile แนะนำให้ทำงานเป็นคู่
เป็นทีม อย่าทำคนเดียว

Effectiveness of Requirements Gathering Techniques



Agile Approach



Agile Modeling Methodology

- Scrum by Ken Schwaber and Jeff Sutherland, 1990
- eXtreme Programming (XP) by Kent Beck, 1996
- Feature Driven Development (FDD) by Jeff De Luca, 1999
- Crystal Clear: A Human-Powered Methodology for Small Teams by Alistair Cockburn, 2004

Timeline คร่าวๆ สำหรับเทคนิคที่ใช้หลักการ Agile Principle

An Introduction to Scrum

Slide from Mountain Goat Software

We're losing the relay race

“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.”

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

รักนี้ ดีกว่า วิ่งผลัด อย่างไร



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.

Scrum origins

- Jeff Sutherland
 - Initial scrums at Easel Corp in 1993
 - IDX and 500+ people doing Scrum
- Ken Schwaber
 - ADM
 - Scrum presented at OOPSLA 95 with Sutherland
 - Author of three books on Scrum
- Mike Beedle
 - Scrum patterns in PLOPD4
- Ken Schwaber and Mike Cohn
 - Co-founded Scrum Alliance in 2002, initially within the Agile Alliance



Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswich
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce



Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- the Joint Strike Fighter
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use



Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”



The Agile Manifesto—a statement of values

Individuals and interactions

Working software

Customer collaboration

Responding to change

over

over

over

over

Process and tools

Comprehensive documentation

Contract negotiation

Following a plan

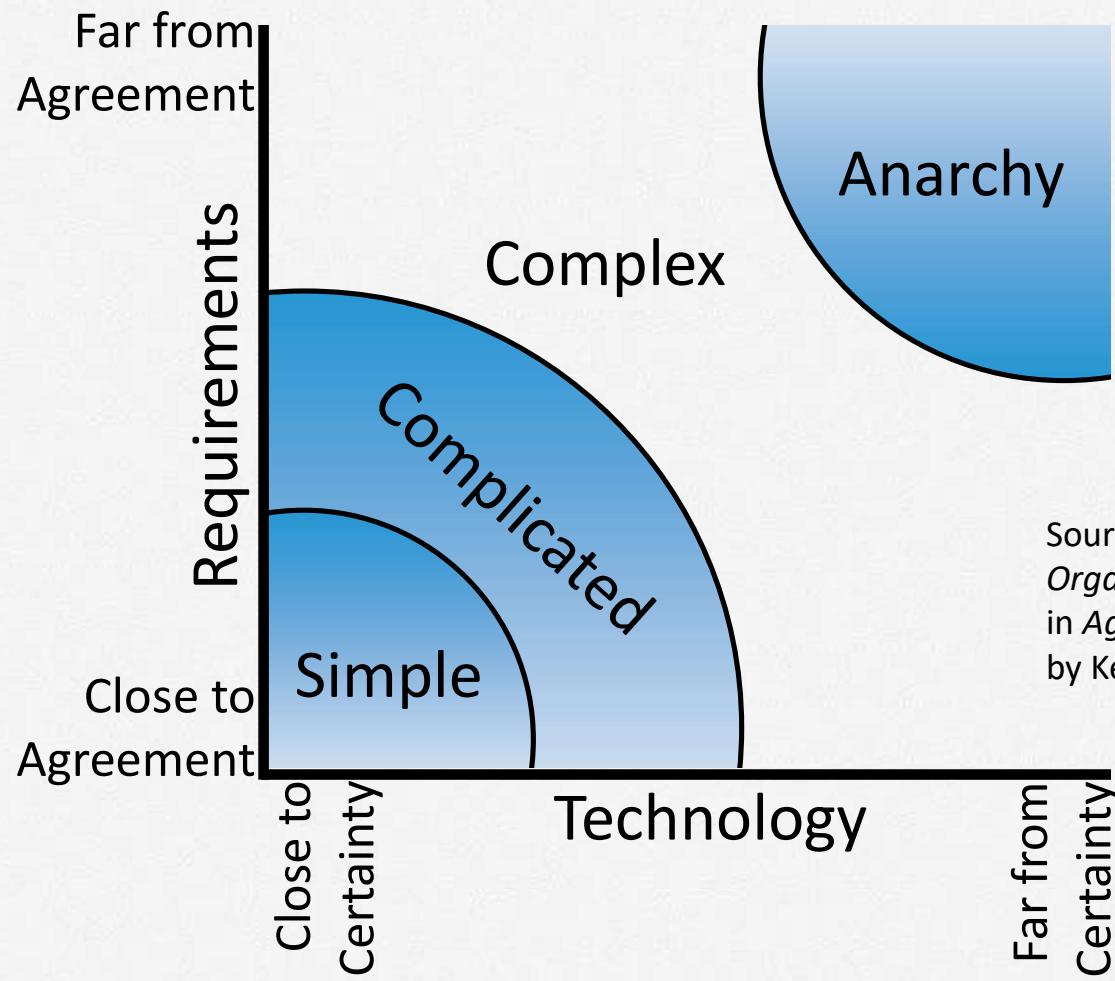
Source: www.agilemanifesto.org



Mountain Goat Software, LLC

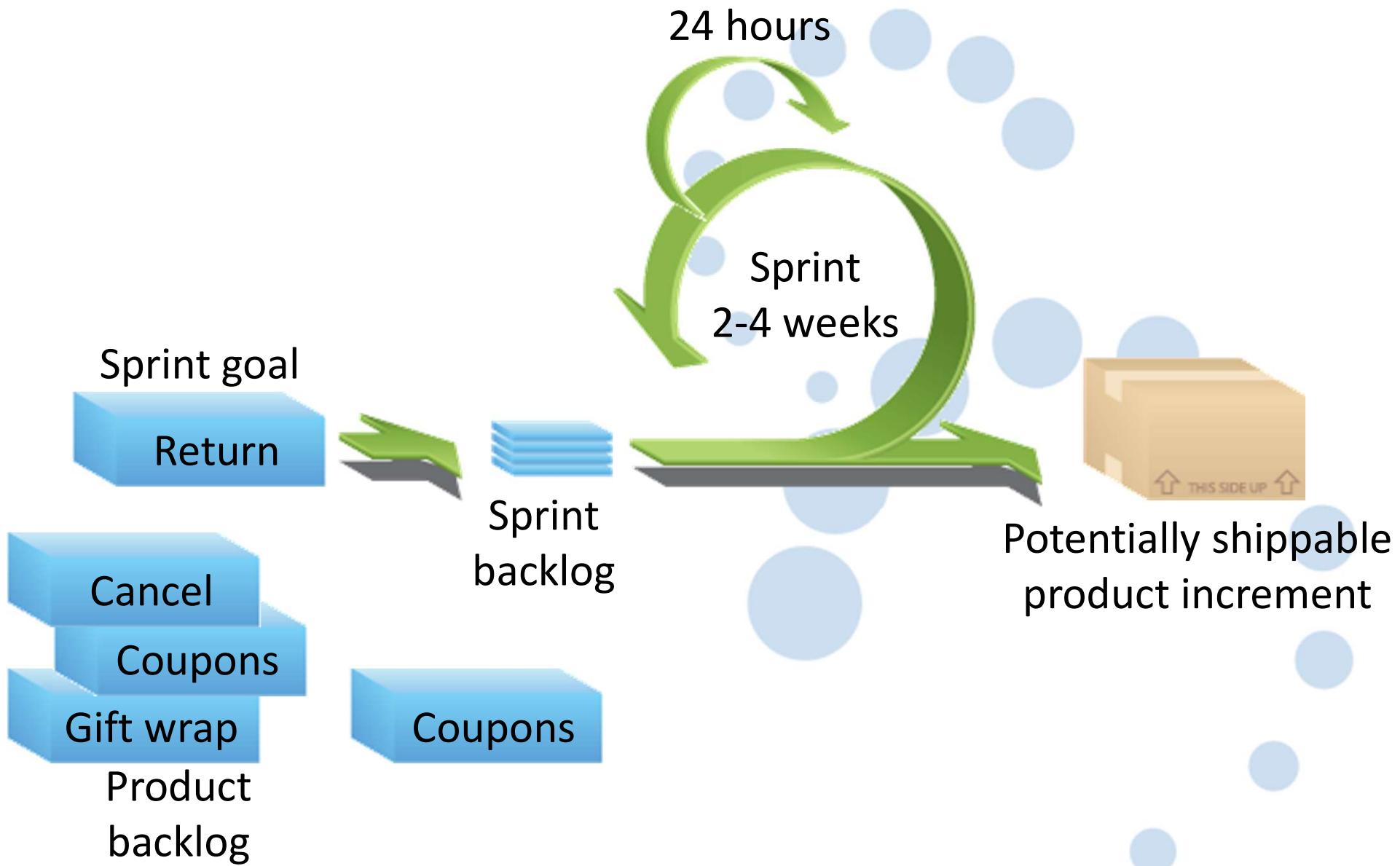


Project noise level

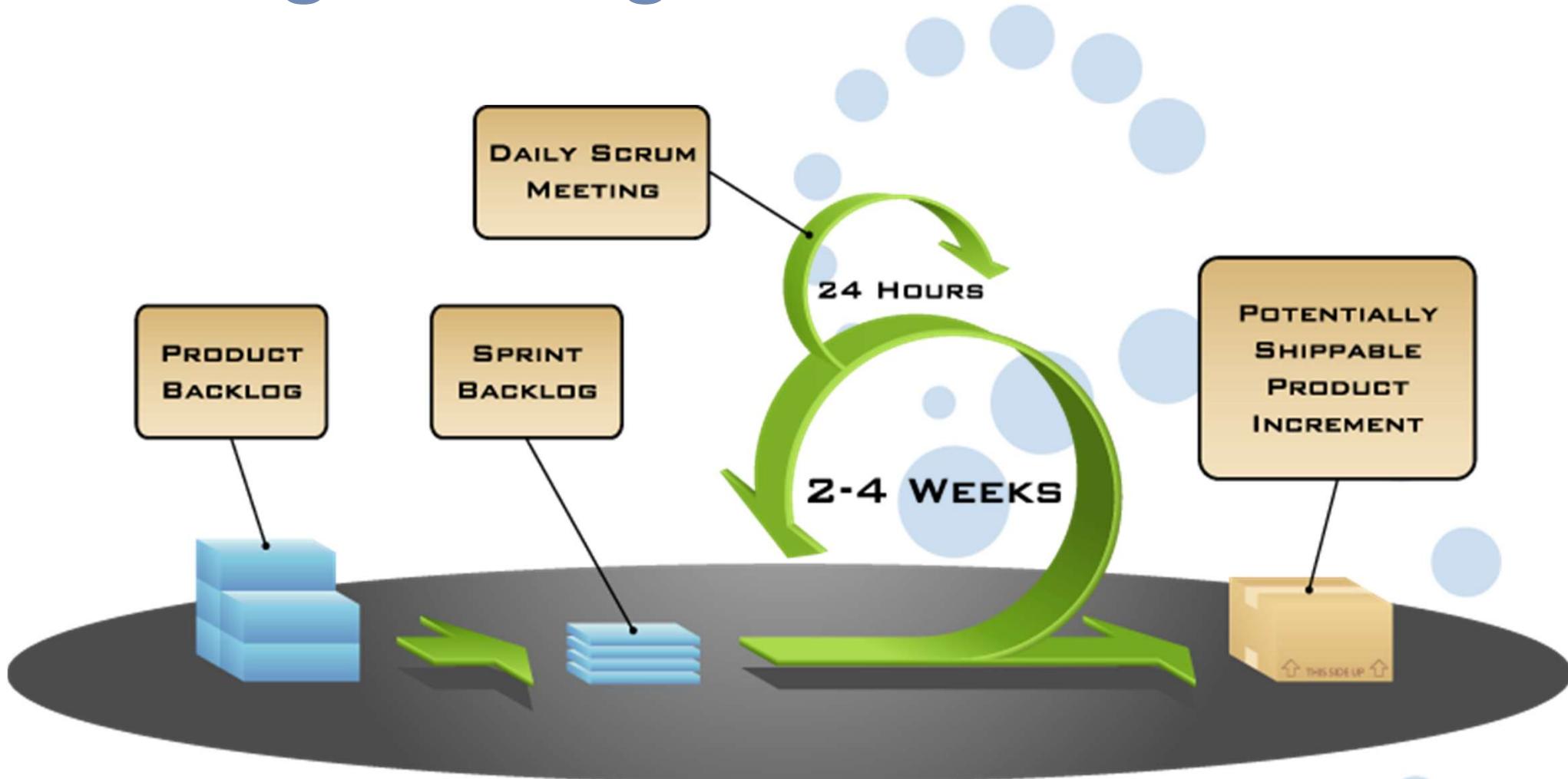


Source: *Strategic Management and Organizational Dynamics* by Ralph Stacey
in *Agile Software Development with Scrum*
by Ken Schwaber and Mike Beedle.

Scrum



Putting it all together



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Image available at
www.mountaingoatsoftware.com/scrum



Mountain Goat Software, LLC

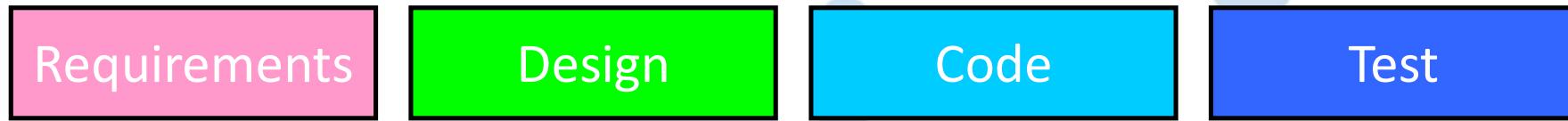


Sprints

- 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

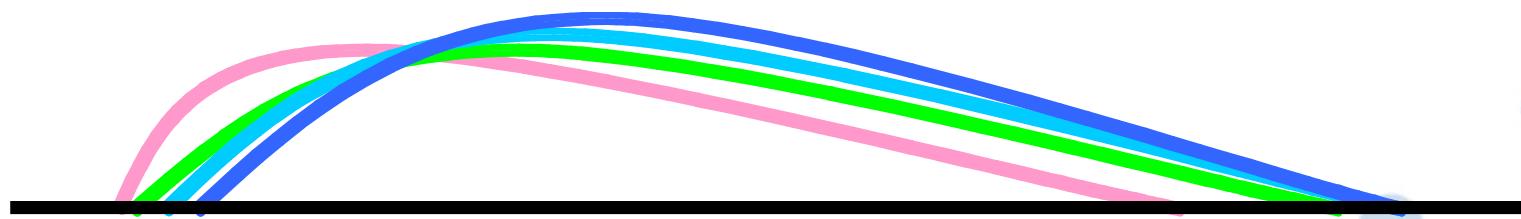


Sequential vs. overlapping development



Rather than doing all of
one thing at a time...

...Scrum teams do a little
of everything all the time



Source: "The New New Product Development Game" by Takeuchi and Nonaka.
Harvard Business Review, January 1986.



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

ท่องจำเลย

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

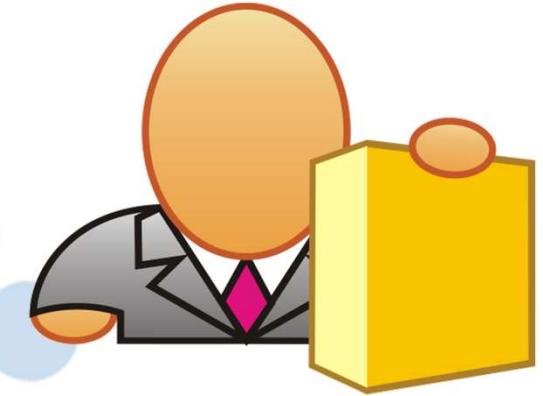
Artifacts

- 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

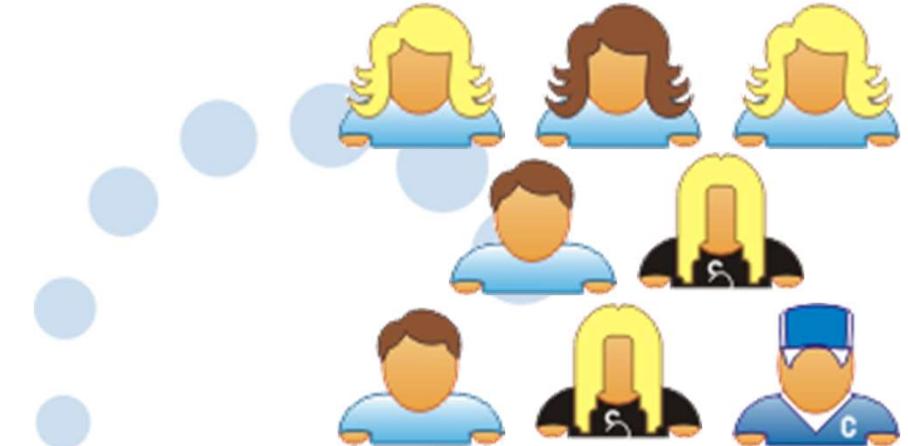


- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences



The team

- Typically 5-9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
 - Members should be full-time
 - May be exceptions (e.g., database administrator)

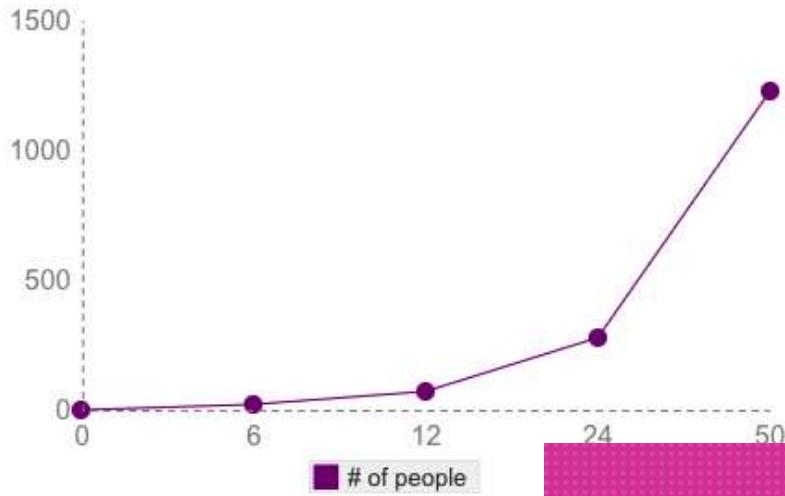


Cross-functional หมายถึงอะไร?

- 1) ใน Team มีสมาชิกที่ทำงานหลายๆ อย่าง ได้
 - 2) สมาชิกคนใดๆ สามารถทำงาน ได้หลายๆ อย่าง ได้
- ในระหว่าง Sprint อาจจะต้องมีการช่วยเหลือกันในแต่ละหน้าที่ เช่น Programmer มาช่วย Test

Amazon Jeff Bezos' Two Pizza Team

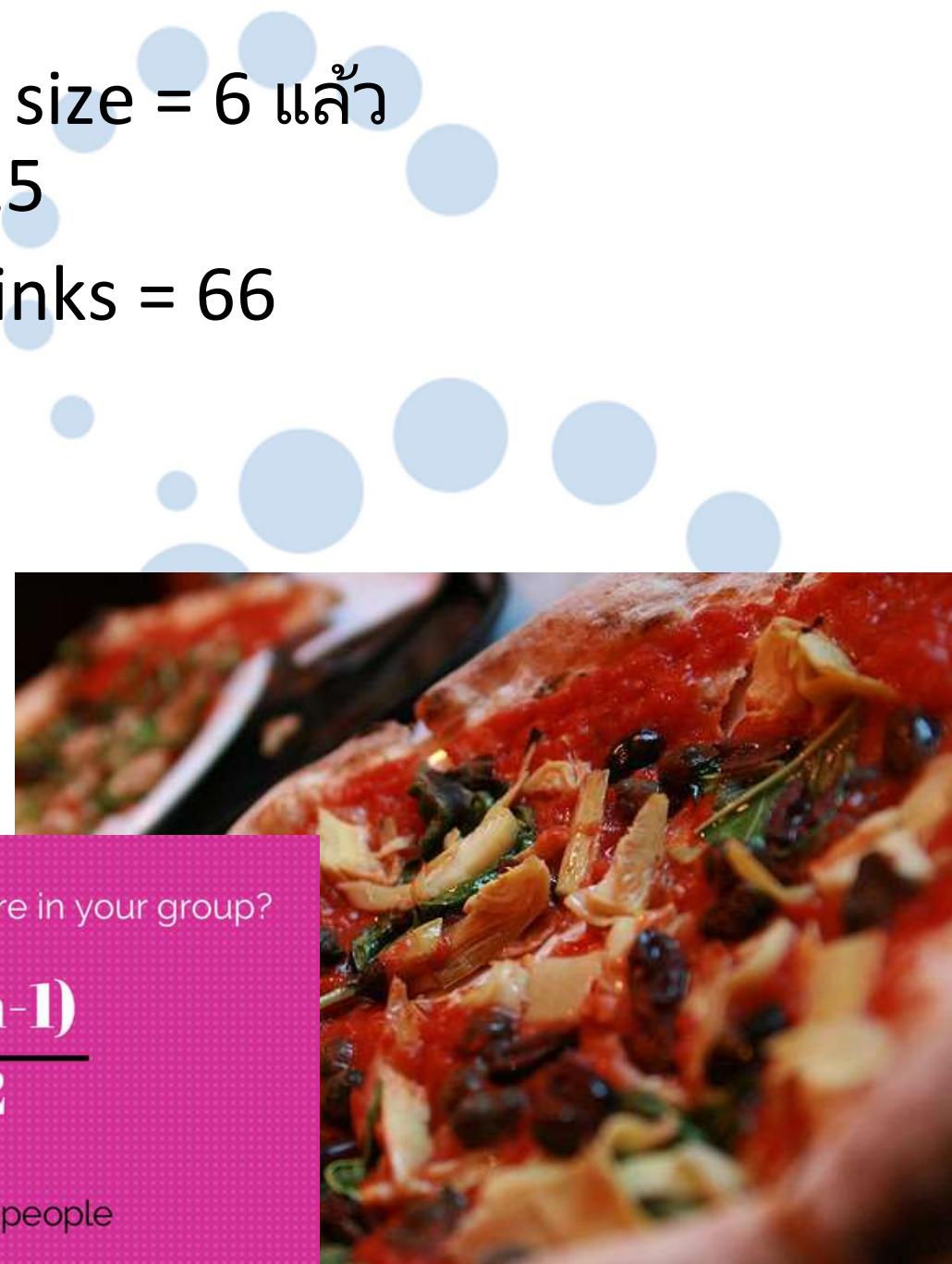
- ถ้า Basic two pizza team size = 6 แล้ว communication links = 15
- ถ้า team size = 12 then links = 66



How many links are in your group?

$$\frac{n(n-1)}{2}$$

$n = \# \text{ of people}$



The team

- Teams are self-organizing
 - Ideally, no titles but rarely a possibility
- Membership should change only between sprints



Self-organizing หมายถึงอะไร?



Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

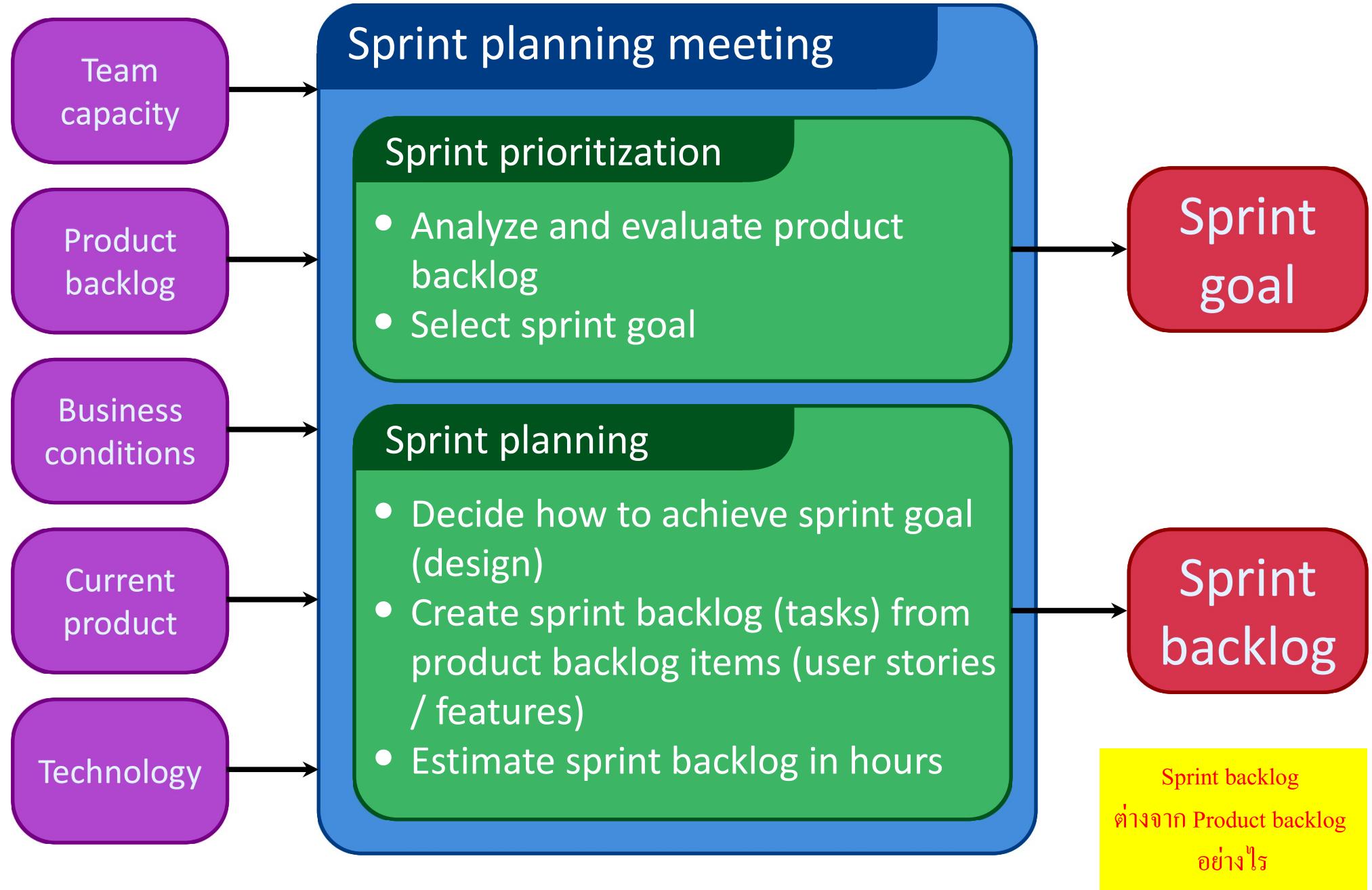
Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts





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)
 - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

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

รายการใน Product backlog

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

รายการใน Sprint backlog



The daily scrum

- Parameters
 - Daily
 - 15-minutes
 - Stand-up
 - Same place (recommended)
- 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

ประชุมเพื่อส่งงาน

- 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



Concern “Pig and Chicken” roles



Mountain Goat Software, LLC

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

ประชุมเพื่อ
เรียนรู้และปรับวิธีการทำงาน
ใน Sprint รอบต่อไป



Start / Stop / Continue

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

Start doing

ตัวอย่างในการทำ Sprint Retro.
ให้ตาม 3 คำตามนี้

Stop doing

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

Continue doing



Mountain Goat Software, LLC



Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

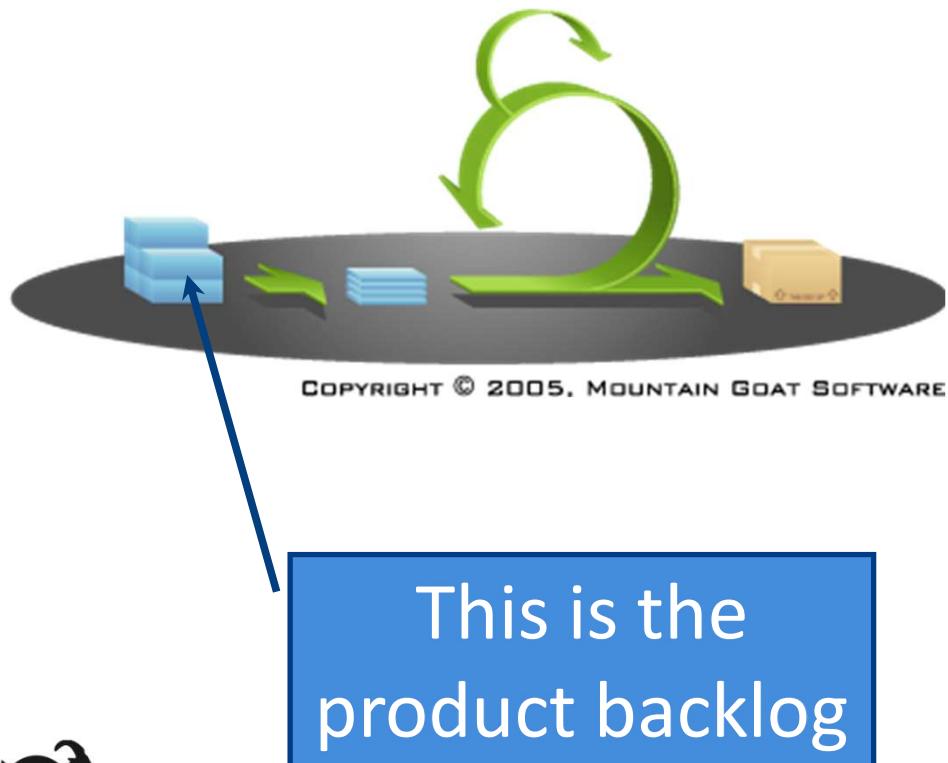
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

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



A sample product backlog

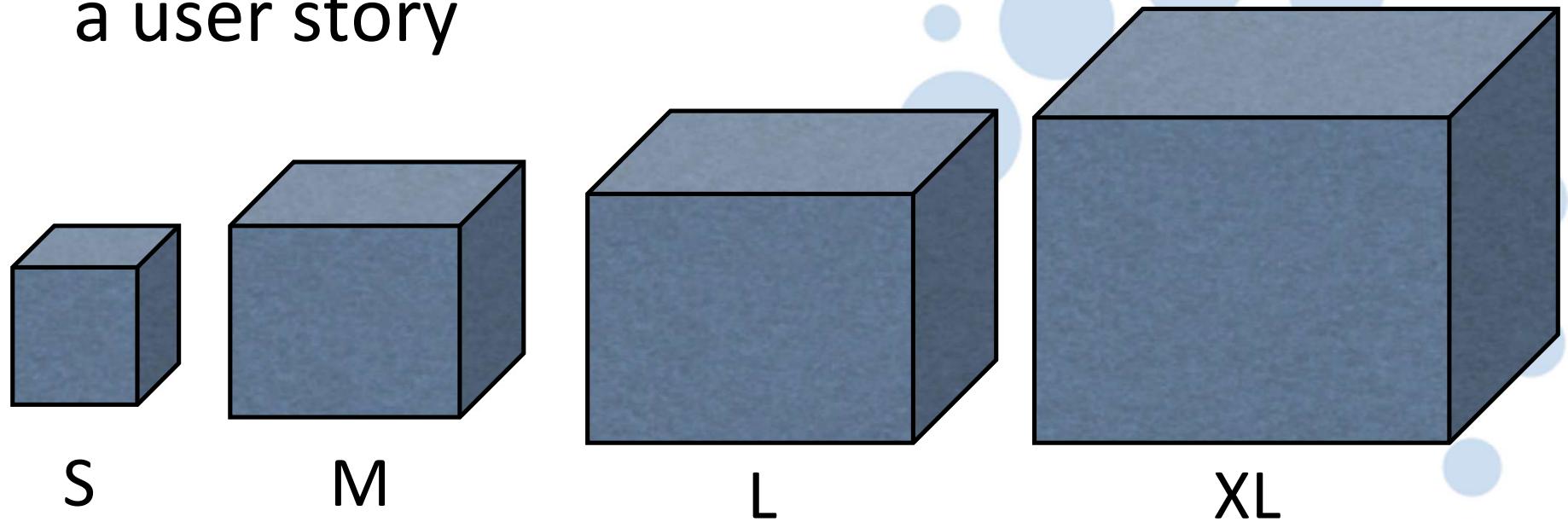
អនុវយ?

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
Writing in User Story Format	
...	50



Agile Estimation of User Story

- Prefer measure “size” to “time”
- Prefer Relative score to Absolute score
- Using relative “Story Point” (indicating size) of a user story



or 1, 2, 3, 5, 8, 13, 21, 34, ... Fibonacci-like sequence

The sprint goal

เขียนประโยชน์เพิ่มเติมสั้นๆ ของ Sprint รอบนี้

- 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



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



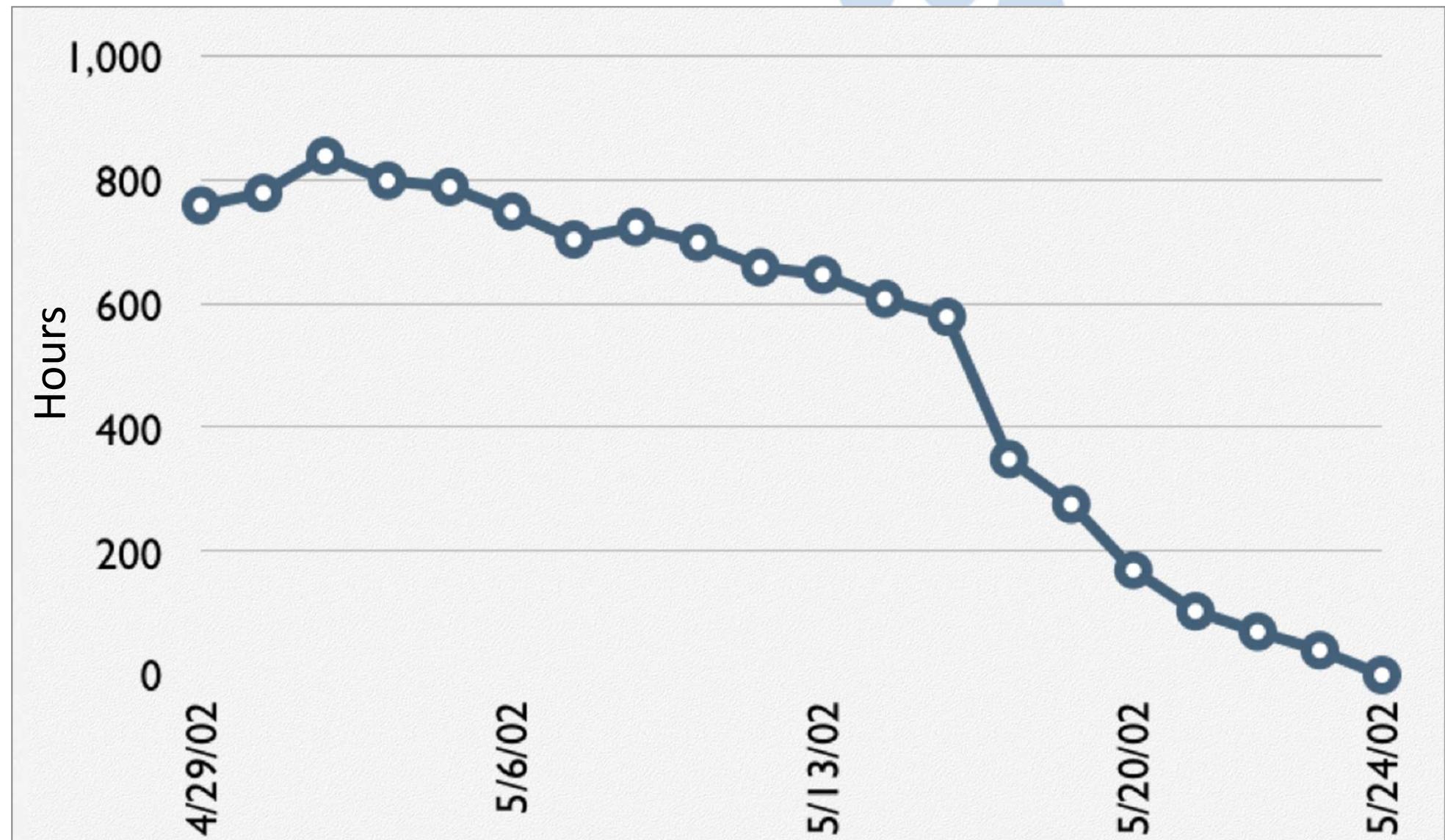
A sprint backlog

Estimated work remaining ของแต่ละ Task แต่ละวัน

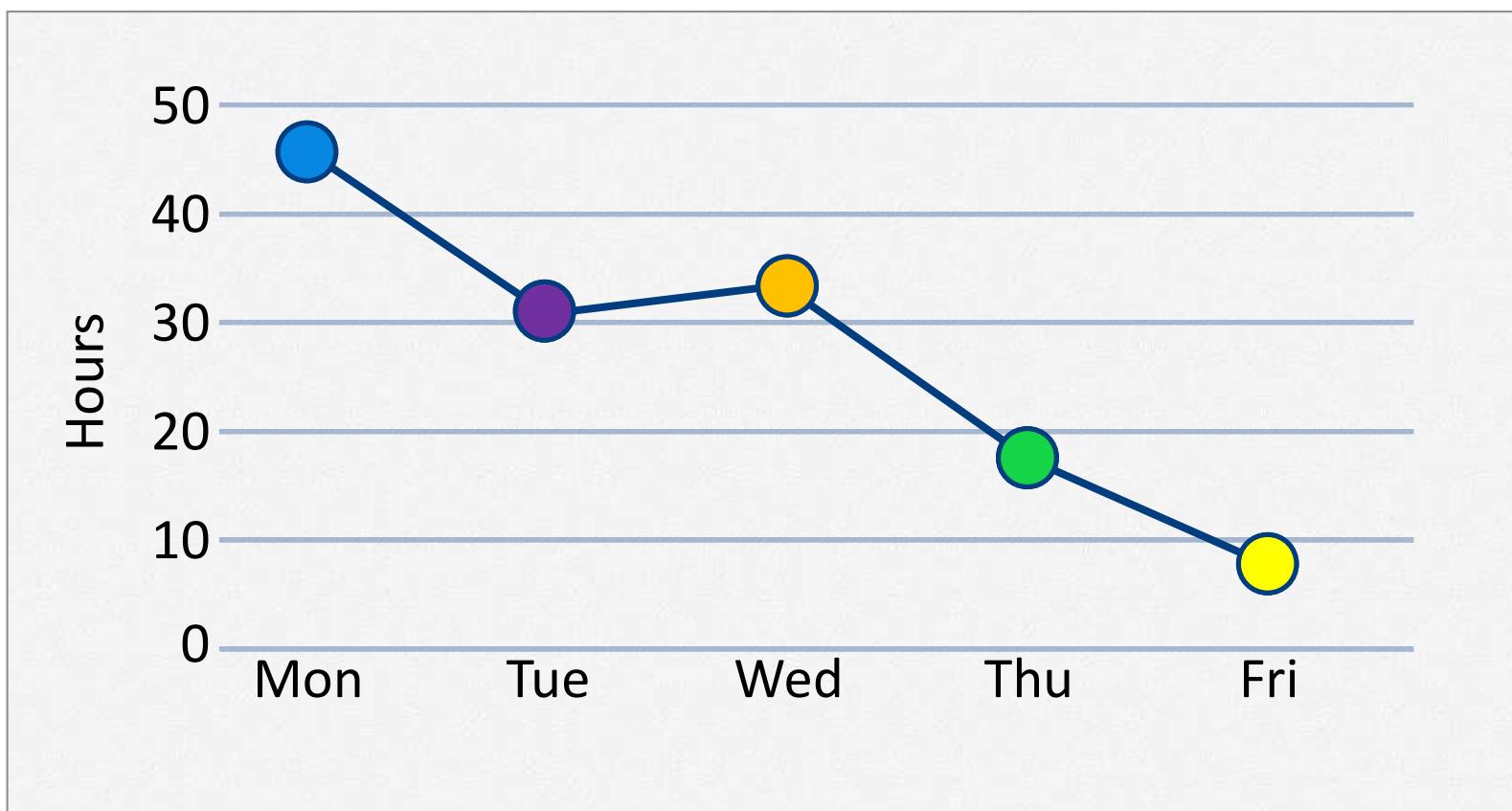
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	



A sprint burndown chart

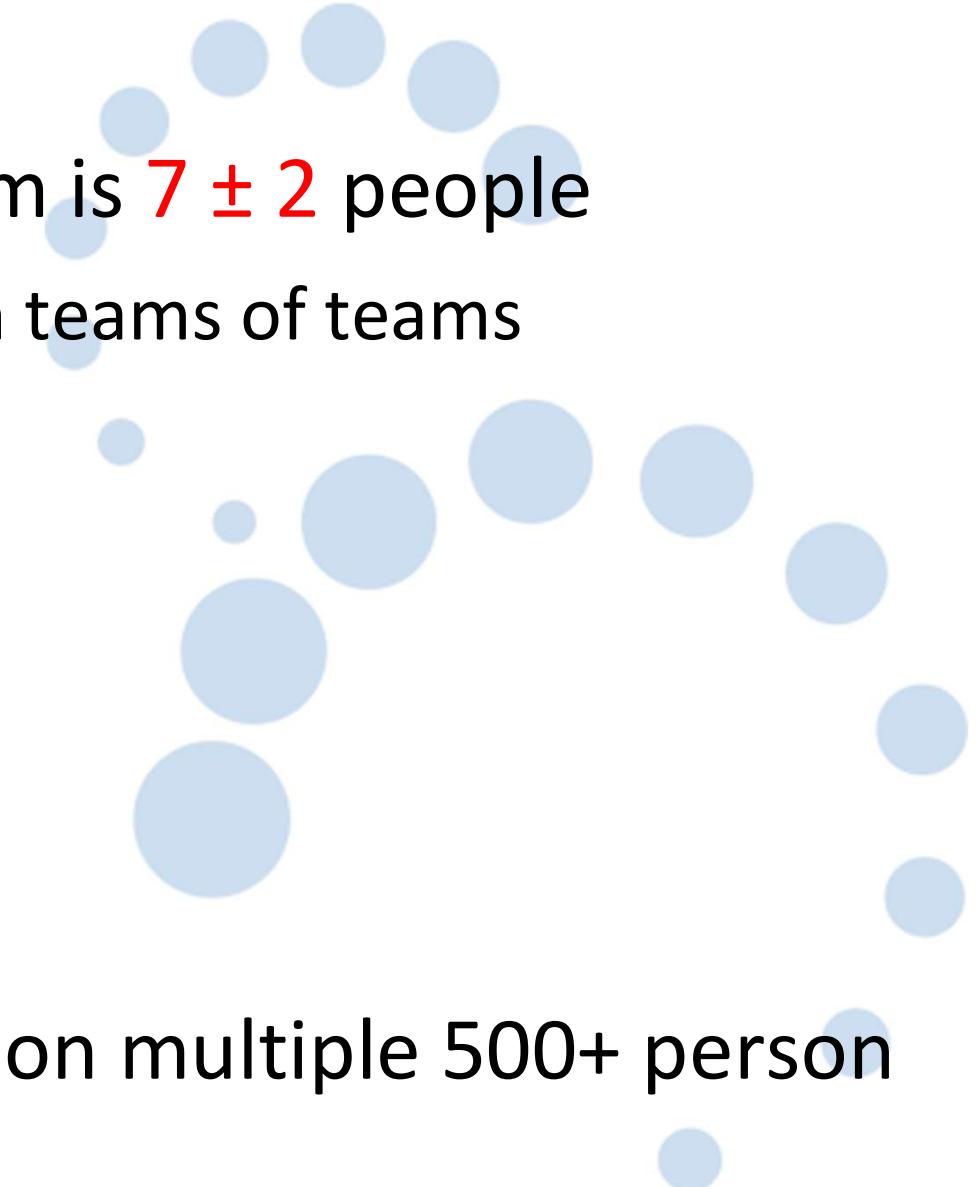


Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				
Total					

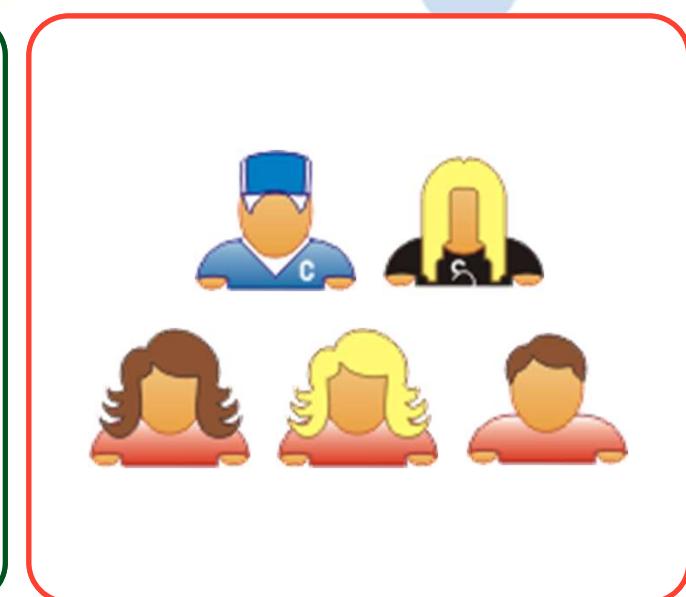
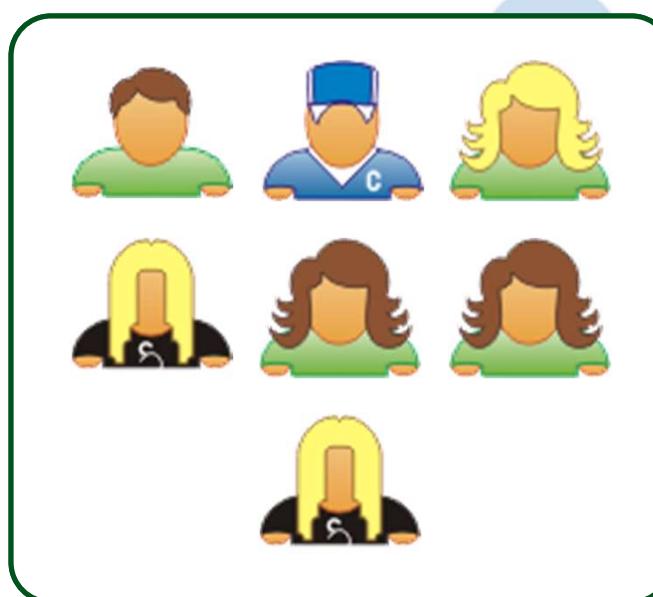
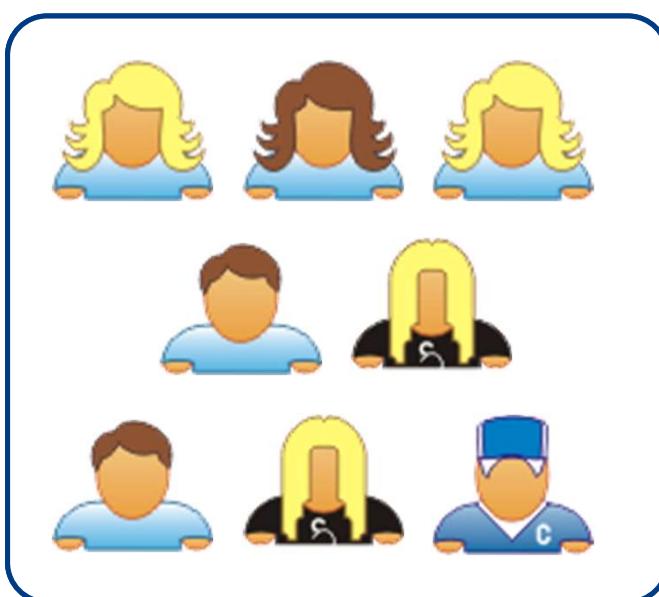
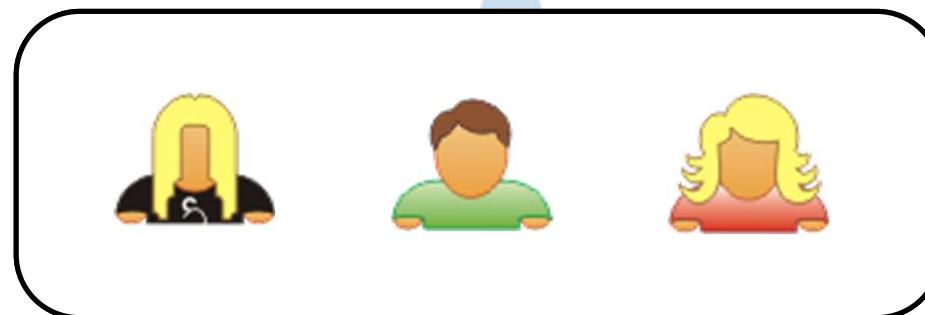


Scalability

- Typical individual team is 7 ± 2 people
 - Scalability comes from teams of teams
- Factors in scaling
 - Type of application
 - Team size
 - Team dispersion
 - Project duration
- Scrum has been used on multiple 500+ person projects



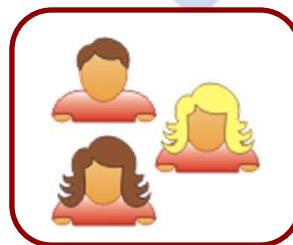
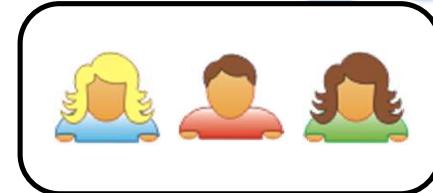
Scaling through the Scrum of scrums



Mountain Goat Software, LLC



Scrum of scrums of scrums



Where to go next

- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahooroups.com



Mountain Goat Software, LLC



A Scrum reading list

- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen



A Scrum reading list

- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- *Scrum and The Enterprise* by Ken Schwaber
- *Succeeding with Agile* by Mike Cohn
- *User Stories Applied for Agile Software Development* by Mike Cohn



Copyright notice



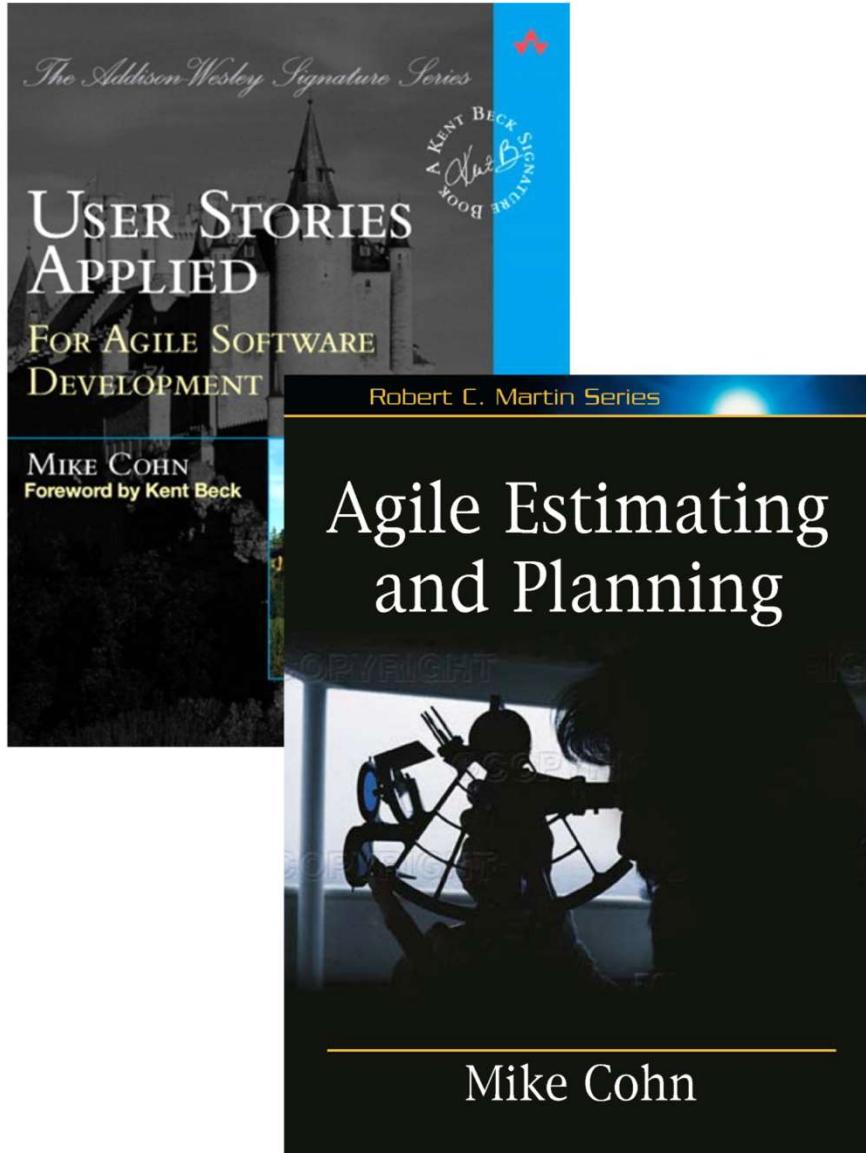
- You are free:
 - to Share—to copy, distribute and transmit the work
 - to Remix—to adapt the work
- Under the following conditions
 - Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- Nothing in this license impairs or restricts the author's moral rights.
- For more information see
<http://creativecommons.org/licenses/by/3.0/>



Mountain Goat Software, LLC



Contact information



Presentation by: Mike Cohn
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
(720) 890-6110 (office)

You can remove this (or any slide) but you must credit the source somewhere in your presentation. Use the logo and company name (as at bottom left, for example) or include a slide somewhere saying that portions (or all) of your presentation are from this source.
Thanks.



Mountain Goat Software, LLC