UNSW Business School

Information Systems and Technology Management

INFS2603 Lecture Series

Agile SCRUM I

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.

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

Agile Approaches

- **Scrum**
- Extreme Programming (XP)
- Kanban
- Crystal
- Dynamic Systems Development Method (DSDM)
- Agile Unified Process (AUP)
- Feature Driven Development
- Adaptive Software Development

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 has been used by:

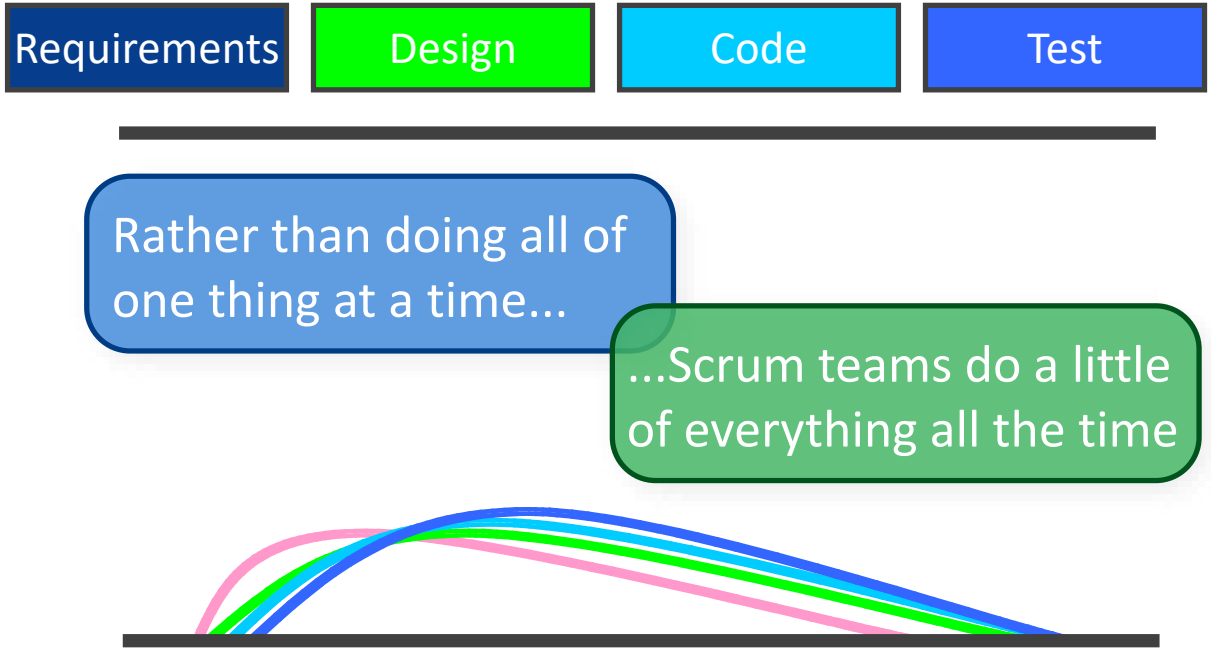
- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- **Lockheed Martin**
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting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

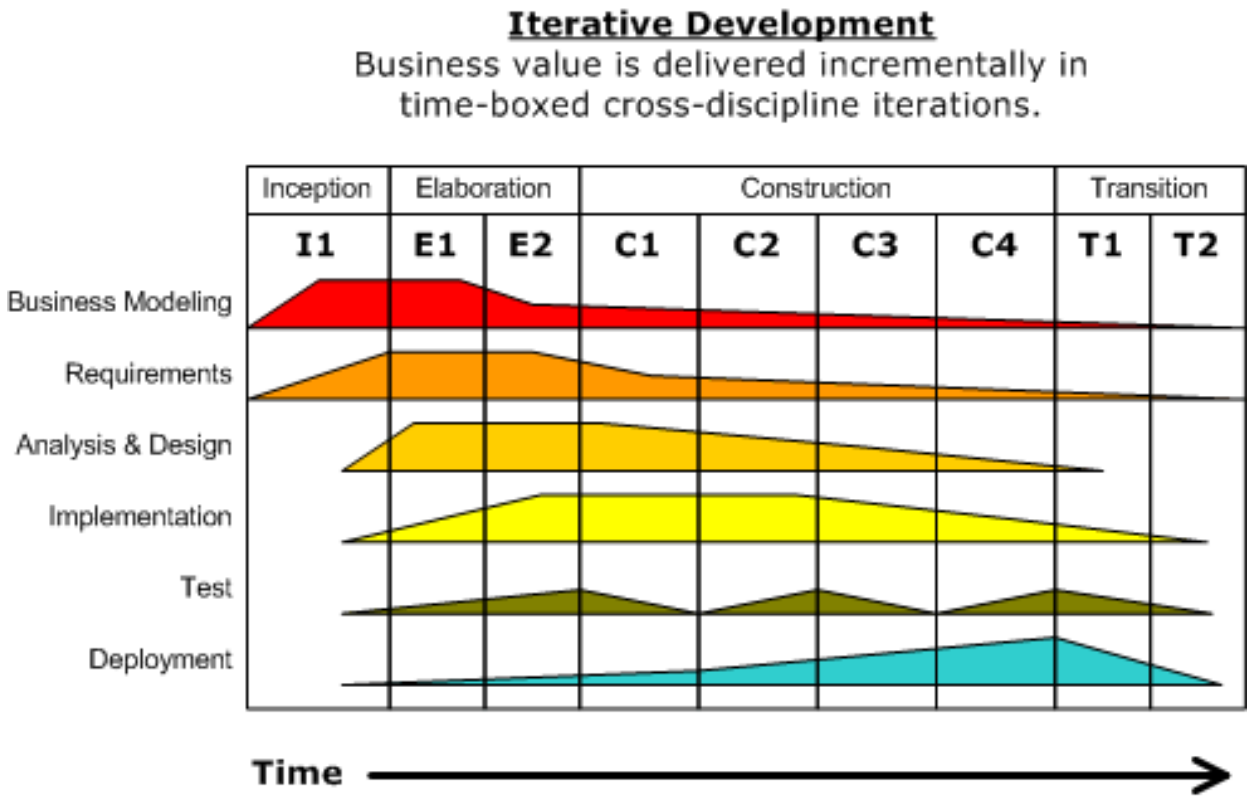
- Video game development
- FDA-approved, life-critical systems
- **Satellite-control software**
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Sequential (Waterfall) vs. Overlapping development



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

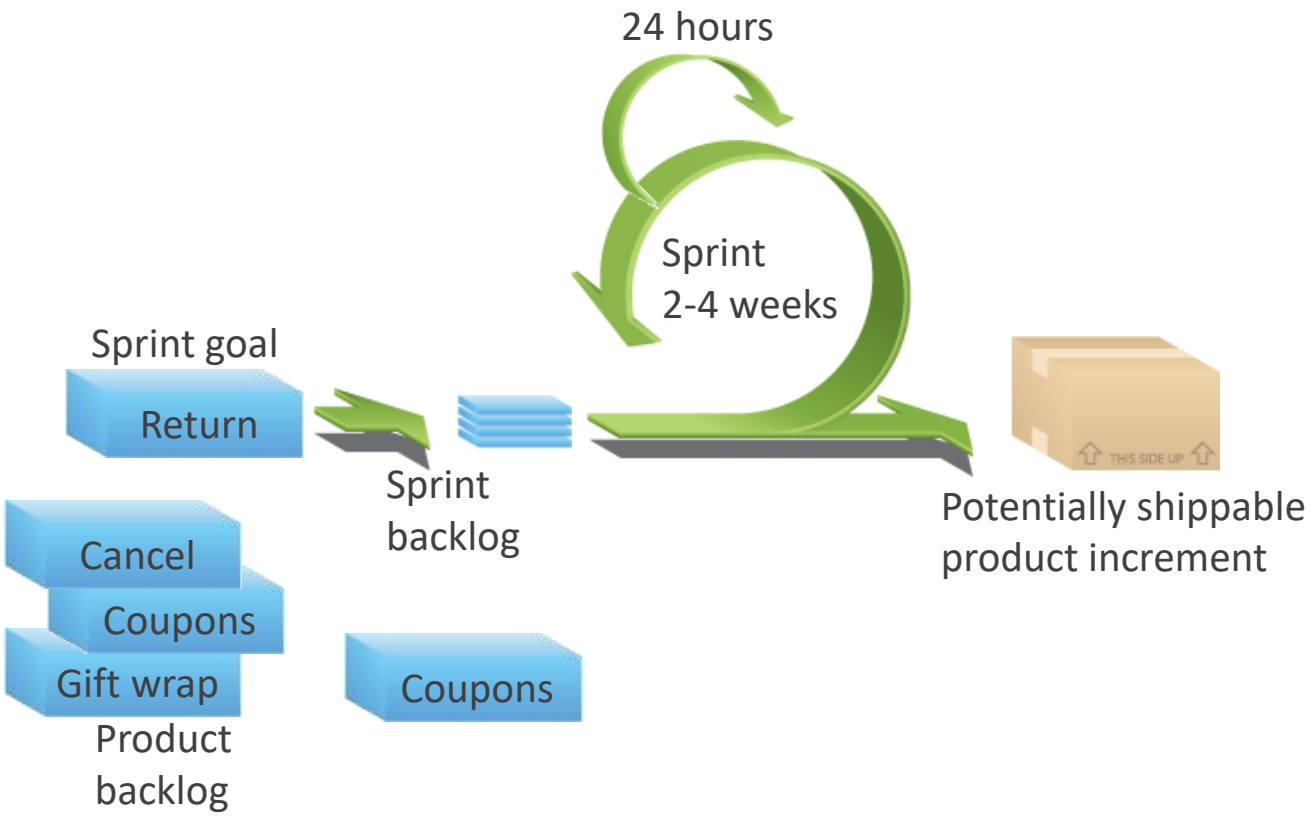
But wait…I thought we had this covered in Unified Process?



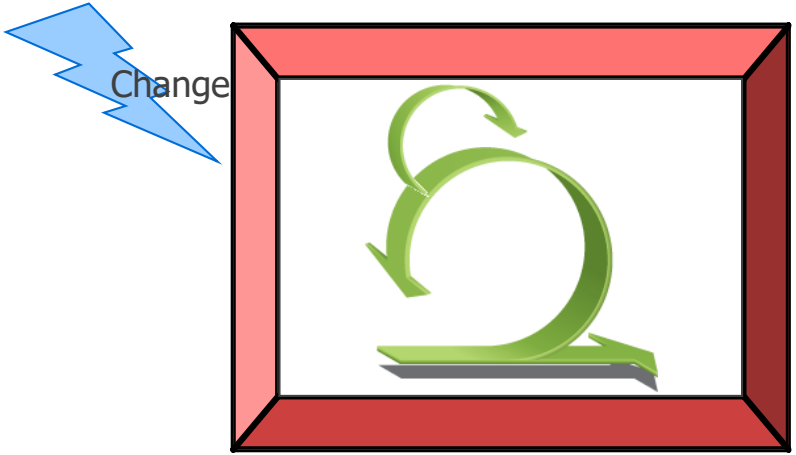
Sprints

- 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


24 hours
Sprint
2-4 weeks
Sprint goal
Return
Sprint
backlog
Potentially shippable
product increment
Cancel
Coupons
Gift wrap
Product
backlog
Coupons

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

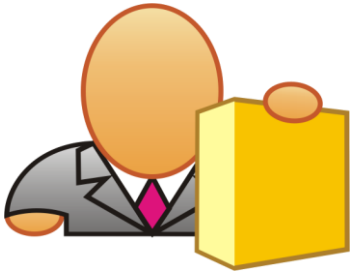
Ceremonies

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

Artifacts

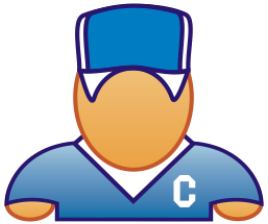
- Product backlog
- Sprint backlog
- Burndown charts

The Product owner



- The Visionary
- 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 Scrum Master



- The Servant Leader
- 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 (?) teams**
- Members should be **full-time**
  - 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

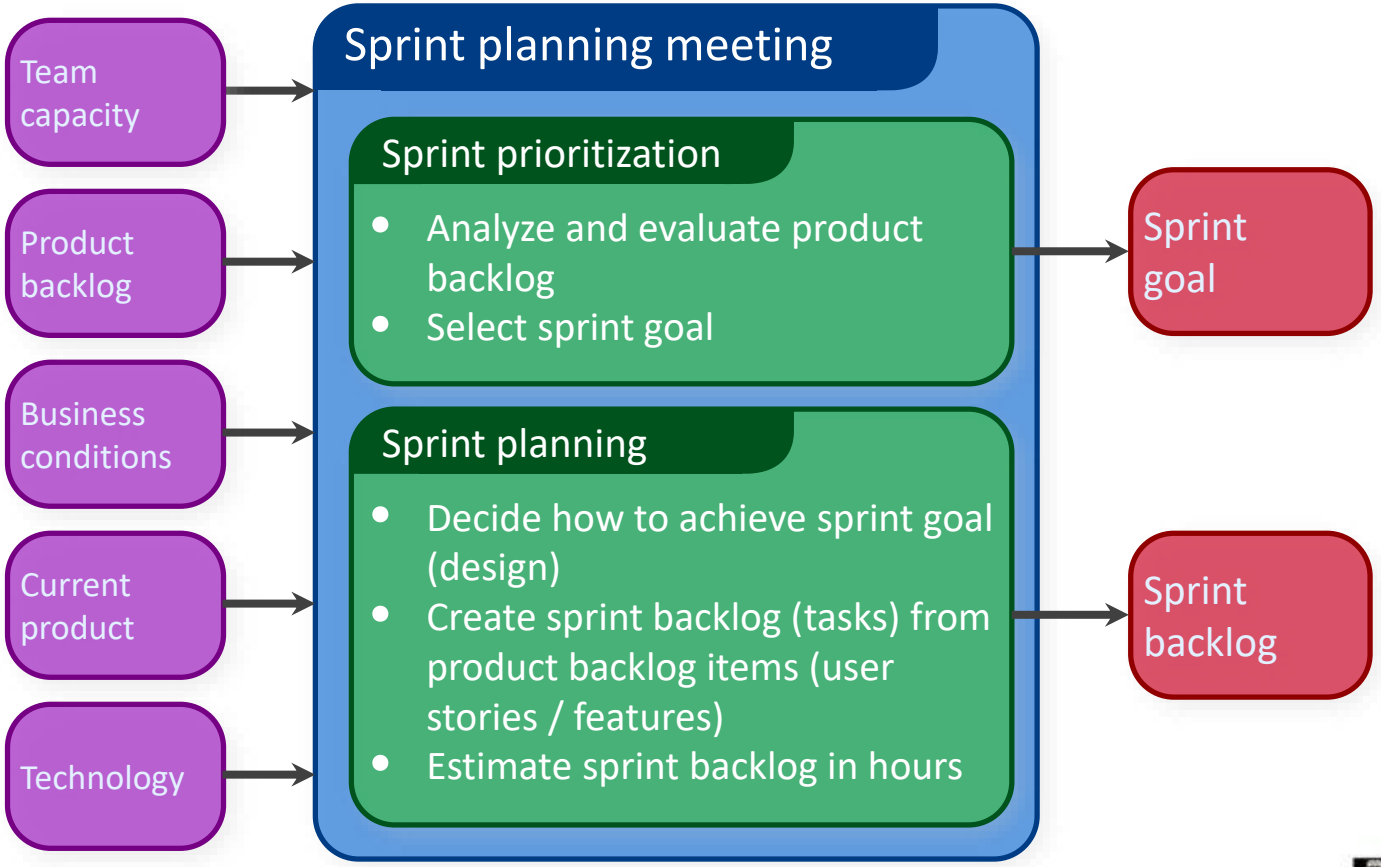
- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts


Team capacity
Product backlog
Business conditions
Current product
Technology
Sprint planning meeting
Sprint prioritization
Analyze and evaluate product backlog
Select sprint goal
Sprint planning
Decide how to achieve sprint goal (design)
Create sprint backlog (tasks) from product backlog items (user stories / features)
Estimate sprint backlog in hours
Sprint goal
Sprint backlog

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.

→

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, Scrum Master, 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*

UNSW | AGSM Business School

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
  - Scrum Master
  - Product owner
  - Team
  - Possibly customers and others

Start / Stop / Continue

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

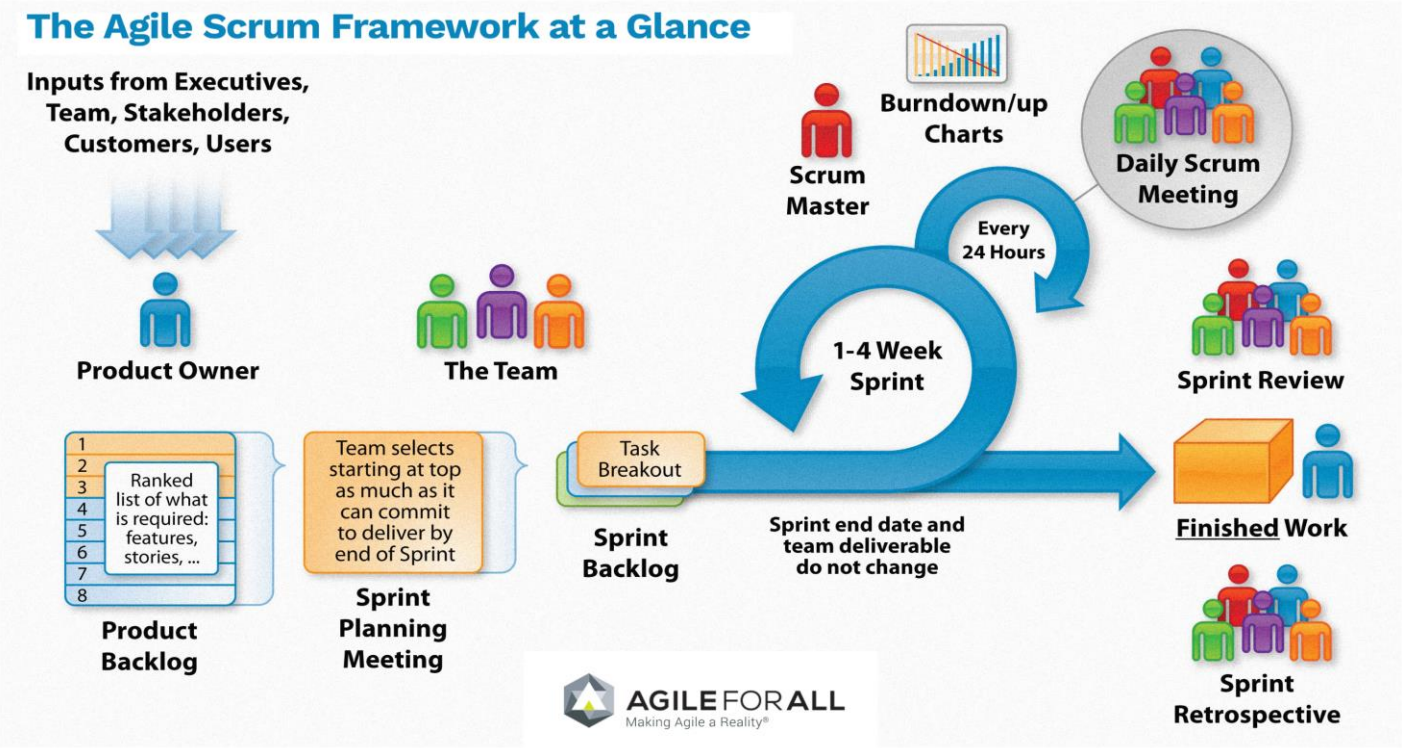
Start doing

Stop doing

Continue doing

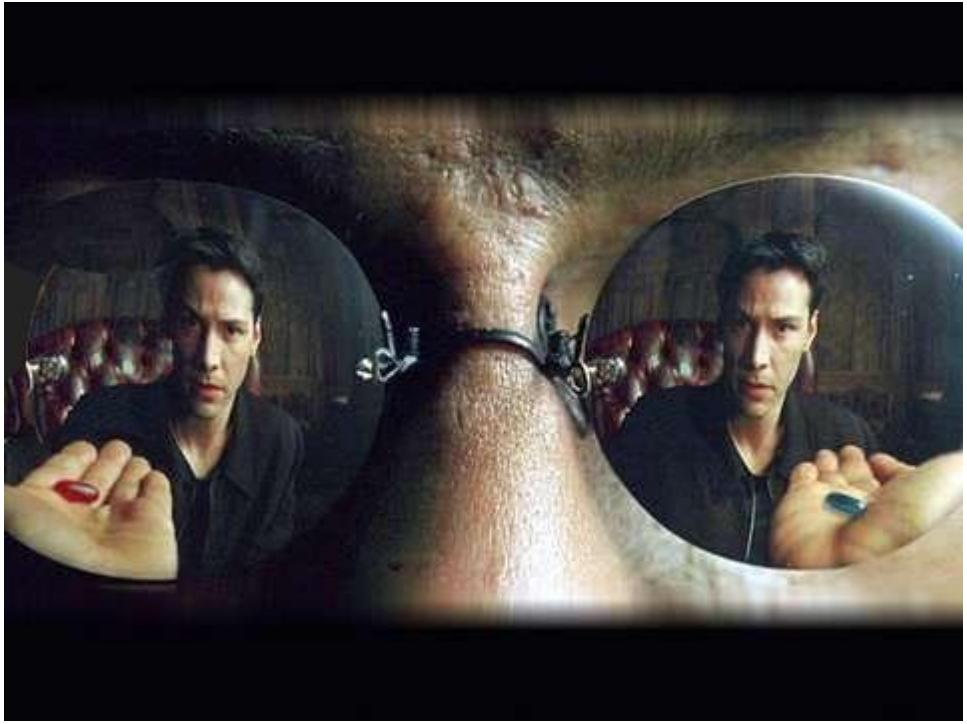
This is just one of many ways to do a sprint retrospective.

Putting it all together



Implications for organisations

- If you want a radical/breakthrough innovation – go Agile
  - Need to create self-contained, autonomous units
- But there is a “cost” – are organisations prepared?
  - Changes to existing **mindsets**
  - Changes to existing **roles**
  - Changes to existing **routines**



In Summary

- From Sequential to “iterative and incremental”
- 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

Note: Presentation adapted from: Mike Cohn | www.mountaingoatsoftware.com

UNSW | AGSM Business School

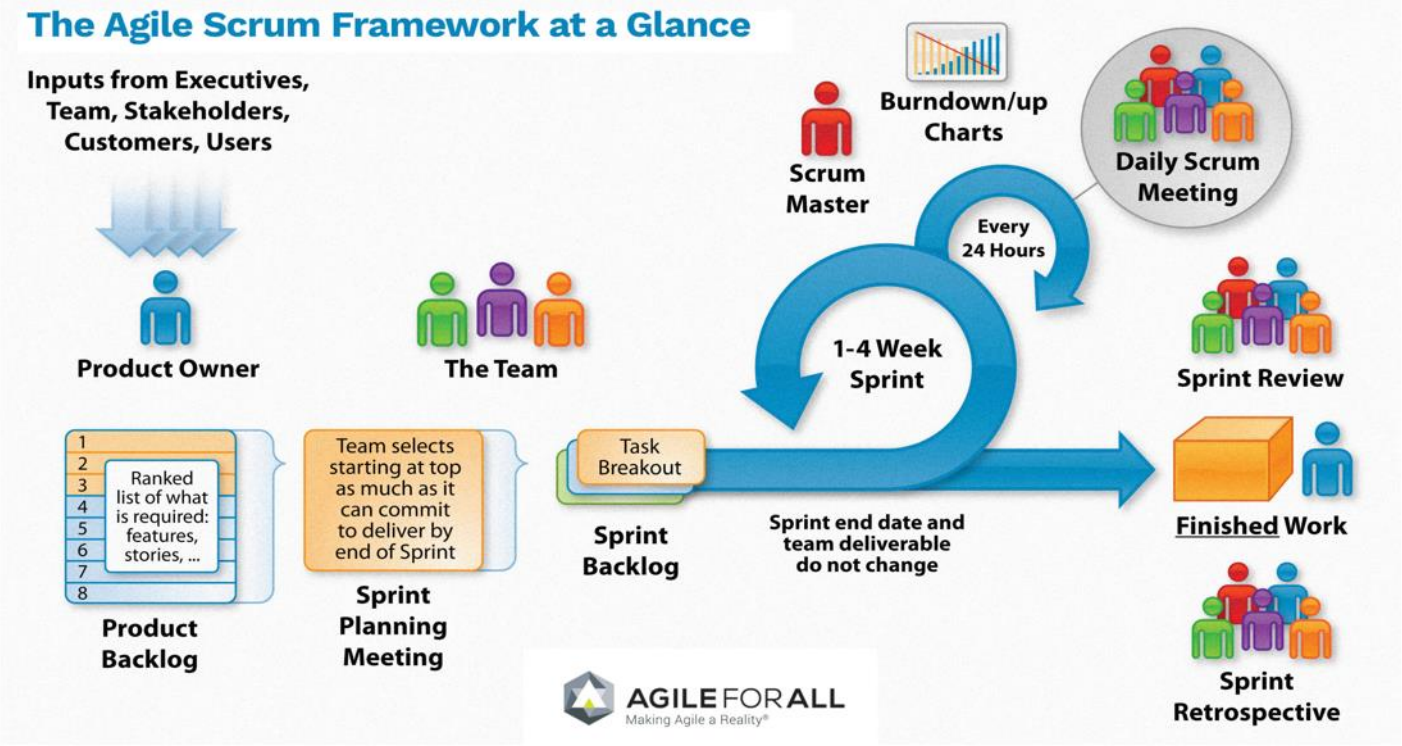
UNSW Business School

Information Systems and Technology Management

INFS2603 Lecture Series

Agile Scrum II

Week 07 Recap



UNSW | AGSM Business School

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Kicking off an Agile Scrum project

- Business vision → Product vision
- "The minimum plan necessary to start a Scrum project consists of a (**Product) Vision** and a **Product Backlog**. The vision describes why the project is being undertaken and what the desired end state is." (Ken Schwaber, co-creator of Agile Scrum)
- Creating a **Product Vision** (Pichler, 2009)
- **Format:**

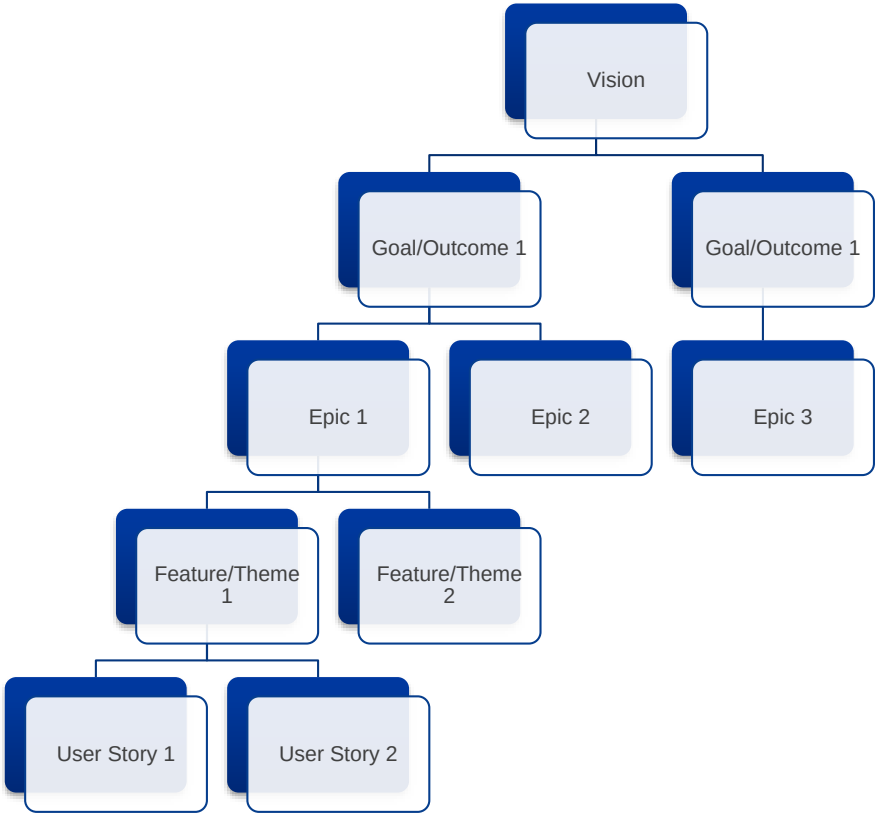
Vision Statement	Crisp summary of the vision / idea.		
Target group	**Needs**	**Product**	**Value**
Which market segment does the product address? Who are the target users and customers?	Which needs does the product fulfil? How does it create value for its users? Which emotions will it evoke?	What are the three to five top features? What are its unique selling points?	How is the product going to benefit the company? Will it, for instance, increase revenue, enter a new market, develop the brand, reduce cost, create valuable knowledge?

http://www.romanpichler.com/

Example:

Vision Statement	Develop a digital product canvas to help teams create great products		
Target group	**Needs**	**Product**	**Value**
Users: Product managers and product owners Customers: Mid-size to large enterprises	Have an effective tool for creating UX-rich products while taking advantage of GreenHopper Leverage the existing investment; minimise the cost of acquiring a new tool	Tablet app; data is held in GreenHopper Looks like a physical canvas; intuitive to use Provides guidance and templates	Open up a new revenue stream Develop our brands and reputation

Creating the Product Backlog: The Big Picture



UNSW | AGSM Business School

Understanding Users and Customers

- **Users interact directly with the system**
- They are important to understand because:
  - Knowledge of current usage patterns helps to design better, more usable systems
  - Unsatisfied users will work around the system, nullifying its advantages and eventually eliminating it
- **Customers (or project sponsors) make buying/adoption decisions**
- They are also important, because:
  - They have their own wish lists that may have little to do with their users' needs
  - They make the purchasing decisions, so if they're not happy, you won't get in the door

User modelling: Creating Personas

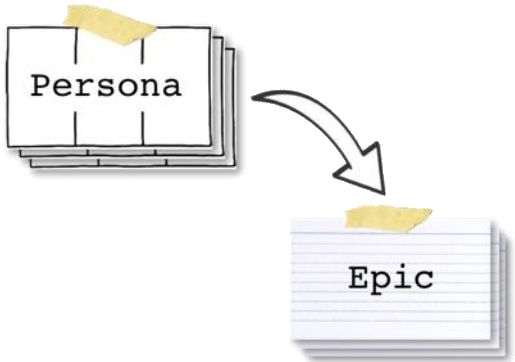
- **Personas represent a type of user across usage contexts**
  - One member of the *current or desired audience* in a tangible, less ambiguous way
  - A user, a user segment or a user profile
  - Provide a name, a face, and a description, giving a mental model of users allowing us to empathize with them and **predict how they will use the system**
  - Helps the team understand **motivations**, **level of expertise**, **context of use**, **workflow** and **goals and needs**
- **Level of detail**
  - Add just enough detail to aid empathy, more details can be distracting
  - Lightweight personas will suffice for many

User modelling: Creating Personas

Name & sketch	Description
Behaviors	Needs & goals

Derive Epics from Persona Goals

- Epics are **high-level features or activities**
  - E.g., any elementary business process
- Epics, as coarse-grained, high-level stories, typically spanning multiple sprints
- Write all the epics necessary to meet the persona goals but keep them rough and sketchy at this stage.



Derive Features from Epics

- Features are tangible expressions of functionality, but still too large to build (within a single sprint)
  - E.g., As a shopper, set up a mobile wallet so I can pay for purchases via NFC
- Created by the Product Owner with input from the team
- Often defined prior to release planning
- Decomposed over time to smaller “User Stories”
- Typically represent weeks of effort of implementation

Derive User Stories from Features

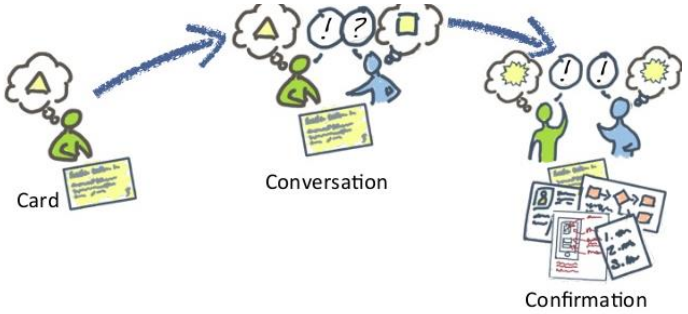
- Instead of Use Cases, Agile project owners do "user stories”
  - From **writing** requirements to **talking** about requirements
- A short story from the user’s perspective about what they find valuable in the product
  - **Who** (user role) – Is this a customer, employee, admin, etc.?
  - **What** (goal) – What functionality must be achieved/developed?
  - **Why** (reason) – Why does user want to accomplish this goal?

As a [user role], I want to [goal], so I can [reason].

- **Example:**
  - "As a user, I want to log in, so I can access subscriber content."
- **Story points**: Rating of effort needed to implement this story
  - common scales: 1-10, shirt sizes (XS, S, M, L, XL), etc.

The 3 C’s of User Stories (developed by Ron Jeffries in Extreme Programming)

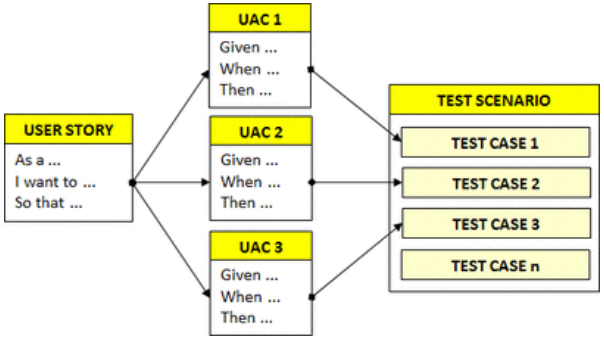
- **Card**
  - Cards are easy representations
  - Excel files overwhelm team members
  - Typically 3”x5” index cards
- **Conversation**
  - User stories are a means to an end, not an end in themselves
  - User stories are for having a group conversation
- **Confirmation**
  - Confirmation that everyone knows how to deliver the story
  - Is the “Acceptance Criteria”; written up on the back of the card
  - Should closely match the user story on the front of the card



User Stories and Acceptance Criteria

What are acceptance criteria?

- Are the conditions that a user story must satisfy to be accepted by a user, customer or the consuming system
- Format: **Given [preconditions] When [Event] Then [Post conditions]**
  - **E.g.,** Given the login page is working, When I input username and click on login button, Then I am on the Home page
- Are a set of statements, each with a clear pass/fail result
- Can be measured and specify both functional and non-functional requirement



Why do we write acceptance criteria?

- Define boundaries, gain shared understanding before development has started
- Help developers and testers derive tests and help developers know when to stop adding more functionality to a story

Example

As a user
I can cancel a registration
So that I don't have
to pay

- Premium member can cancel the same day without a fee
- Non-premium member is charged 50% of first day for a same-day cancellation
- Email confirmation is sent to members primary and secondary email addresses
- Hotel is notified of any cancellation

Add Prospect

As a property manager I want to add a new prospect to the lead management system so I can track my interactions with the prospect.

Conditions of Satisfaction

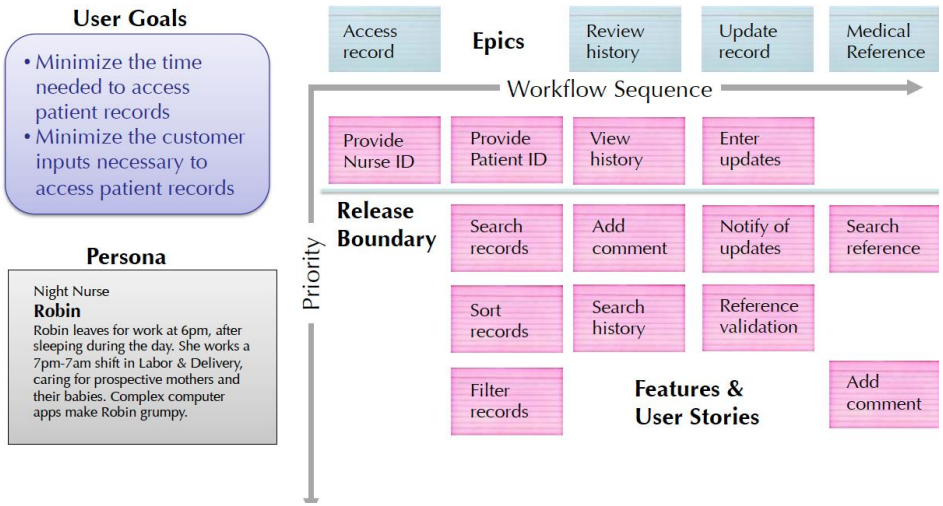
Capture name, email, phone #, contact date, contact format, lease type, and move-in date

Verify prospect is associated with an existing campaign

Writing User Stories: What makes a good user story?

Letter	Meaning	Description
I	Independent	The User Story should be self-contained, in a way that there is no inherent dependency on another User Story.
N	Negotiable	User Stories, up until they are part of an iteration, can always be changed and rewritten.
V	Valuable	A User Story must deliver value to the end user.
E	Estimable	You must always be able to estimate the size of a User Story.
S	Scalable (small sized)	User Stories should not be so big as to become impossible to plan or Task or prioritize with some level of certainty.
T	Testable	The User Story or its related description must provide the necessary information to make test development possible.

Putting it all together: user story mapping



- Helps build empathy with your user, visually depicting the journey undertaken
- Helps managing the project roadmap, especially when you're dealing with big projects
- "Slice" the user story map to help with release planning
- A release typically spans multiple sprints
- Many variations of story mapping

A user story map arranges user stories into a useful model to help understand the functionality of the system, identify holes and omissions in your backlog, and effectively plan holistic releases that deliver value to users and business with each release (from Jeff Patton's [The New User Story Backlog Is a Map])

Product Backlog

This is the product backlog

- The requirements
  - Ideally expressed as **a list of user stories** along with **story points,** such that each item has value to users or customers of the product
- Backlog also contains technical stories, spikes, research stories (but they are beyond the scope of discussion here)
- Prioritized by the product owner
- Reprioritized at start of each sprint

Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3 (story points)
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

"Backlog grooming"

- **removing** user stories that no longer appear relevant
- **creating new** user stories in response to newly discovered needs
- re-assessing the **relative priority** of stories
- **assigning estimates** to stories which have yet to receive one
- **correcting estimates** in light of newly discovered information
- **splitting user stories** which are high priority but too coarse grained to fit in an upcoming iteration

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 change 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

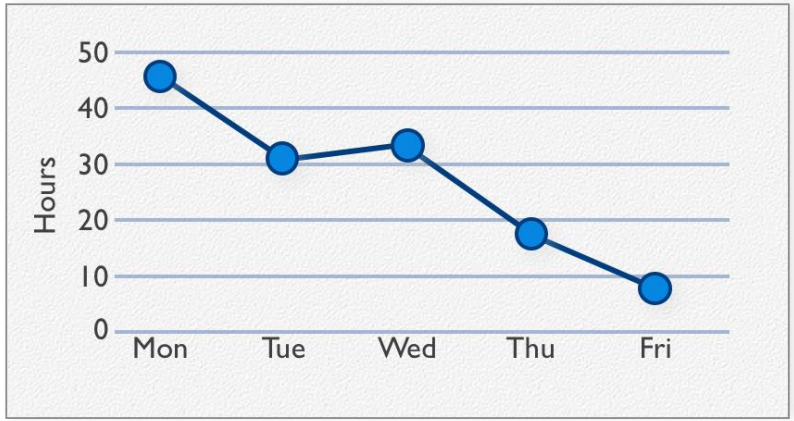
Sample Sprint backlog

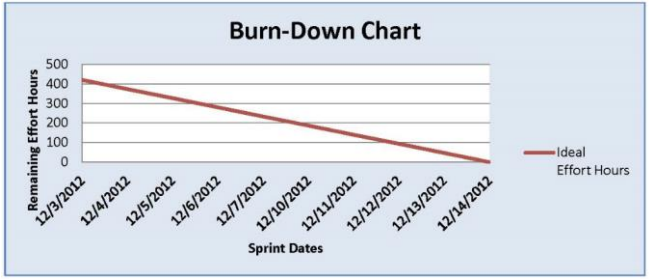
Tasks	Mon	Tue	Wed	Thu	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the Foo class	8	8	8	8	8
Add error logging			8	4	

Sprint Burndown Chart

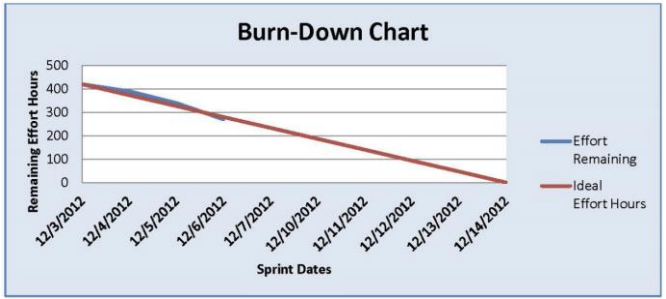
- Sprint tracking mechanism
- A display of ***what work has been completed and what is left to complete***
  - one for each developer or work item
  - updated **every day**
  - (make best guess about hours/points completed each day)
- Step 1: Prepare task breakdown and assign effort
- Step 2: Plot the “ideal” progress
- Step 3: Update chart through the sprint as and when tasks are completed

Tasks	Mon	Tue	Wed	Thu	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				

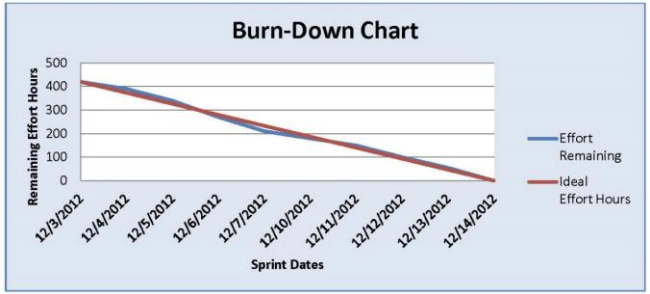




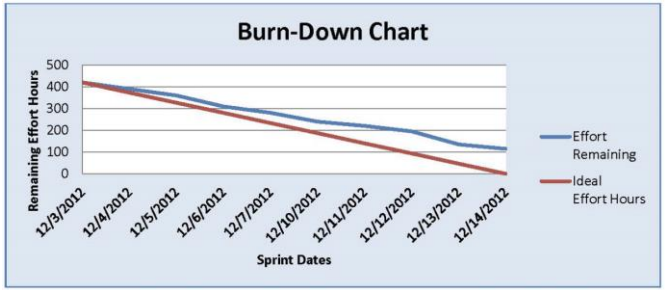
Ideal Burndown



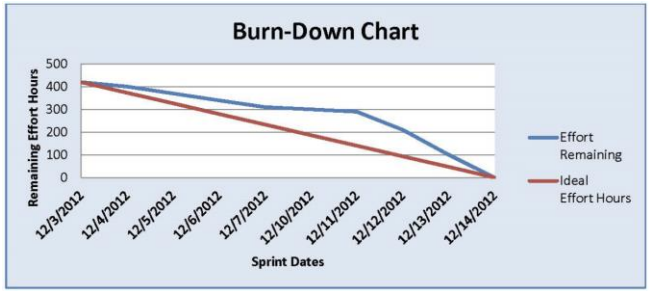
Update with real efforts as you go



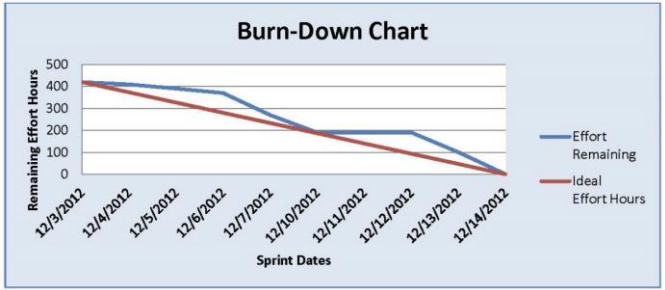
Sprint Commitment Met



Sprint Commitment Not Met



Team stretched toward the end to meet commitment



Team is not consistent

UNSW | AGSM Business School

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
- Scale with Scrum of Scrums
  - Representatives from each scrum attend a “daily standup” type meeting
  - Time-boxed to 15 mins