SPU

الجامعة السورية الخاصة

SYRIAN PRIVATE UNIVERSITY

كلـيـة الـهـنـدسـة

FACULTY OF ENGINEERING

Agile Software Development

Instructor: Dr. Mouhib Alnoukari

5 SCRUM

Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- IBM
- 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

Characteristics

- Self-organizing teams
- Product progresses in a series of 1-4 week “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 Scrum Framework

Business Ideas

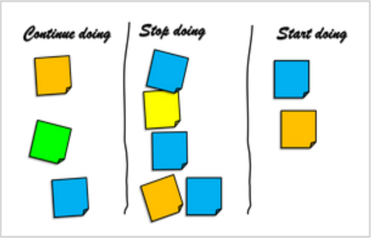
- Epics and Features for scrum team

Sprint Planning Meeting

- Review Product backlog
- Estimate Sprint backlog
- Commit to Sprint timebox
- Communicate Sprint Goals

Daily Standup

- What was done yesterday?
- What will be done today?
- Impediments - Blockers?

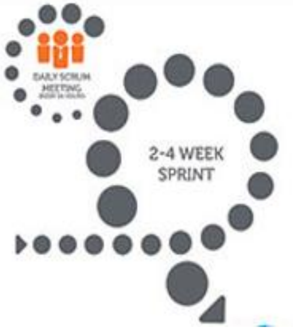


Retrospective

- Inspect and Adapt

Sprint Review Meeting

- Demo and Acceptance of Sprint stories



Product Backlog

- Groomed, prioritized Features in story form

Sprint Backlog

- Stories estimated by team
- Task breakdown by team

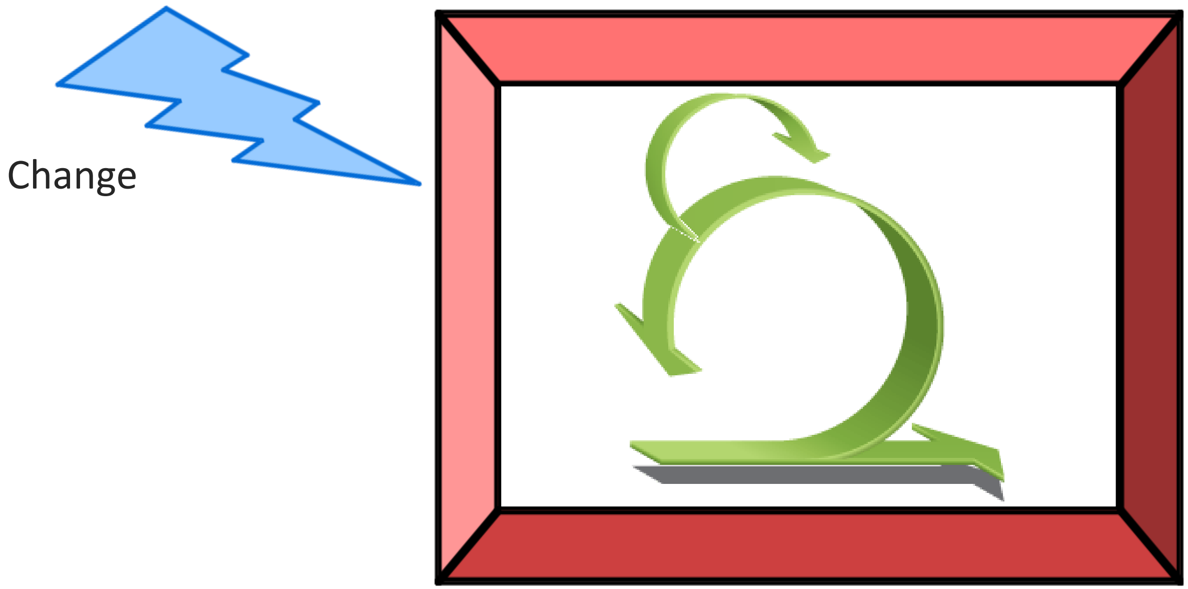
Iteration Timebox

- 2 to 4 weeks

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

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

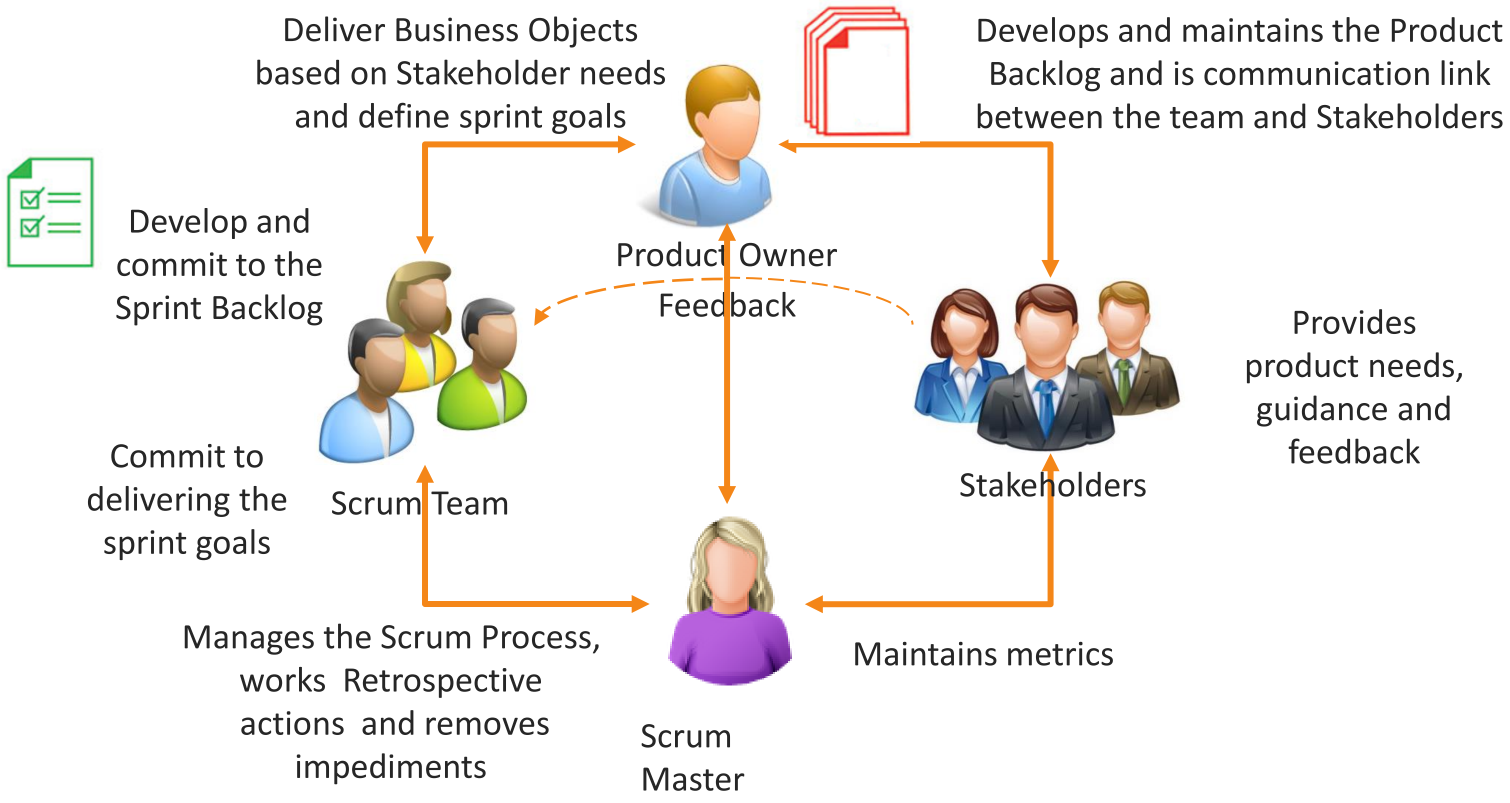
Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Roles and Responsibilities in Scrum



Deliver Business Objects based on Stakeholder needs and define sprint goals

Develops and maintains the Product Backlog and is communication link between the team and Stakeholders

Product Owner

Develop and commit to the Sprint Backlog

Feedback

Scrum Team

Commit to delivering the sprint goals

Stakeholders

Provides product needs, guidance and feedback

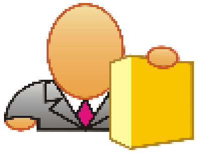
Manages the Scrum Process, works Retrospective actions and removes impediments

Scrum Master

Maintains metrics

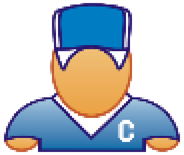
Roles and Responsibilities in Scrum

– Product Owner



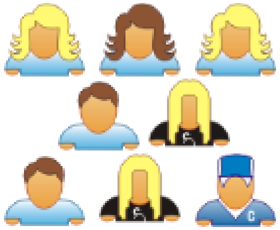
- Possibly a Product Manager or Project Sponsor
- Decides features, release date, prioritization, $$$

– Scrum Master



- Typically a Project Manager or Team Leader
- Responsible for enacting Scrum values and practices
- Remove impediments / politics, keeps everyone productive

– Project Team

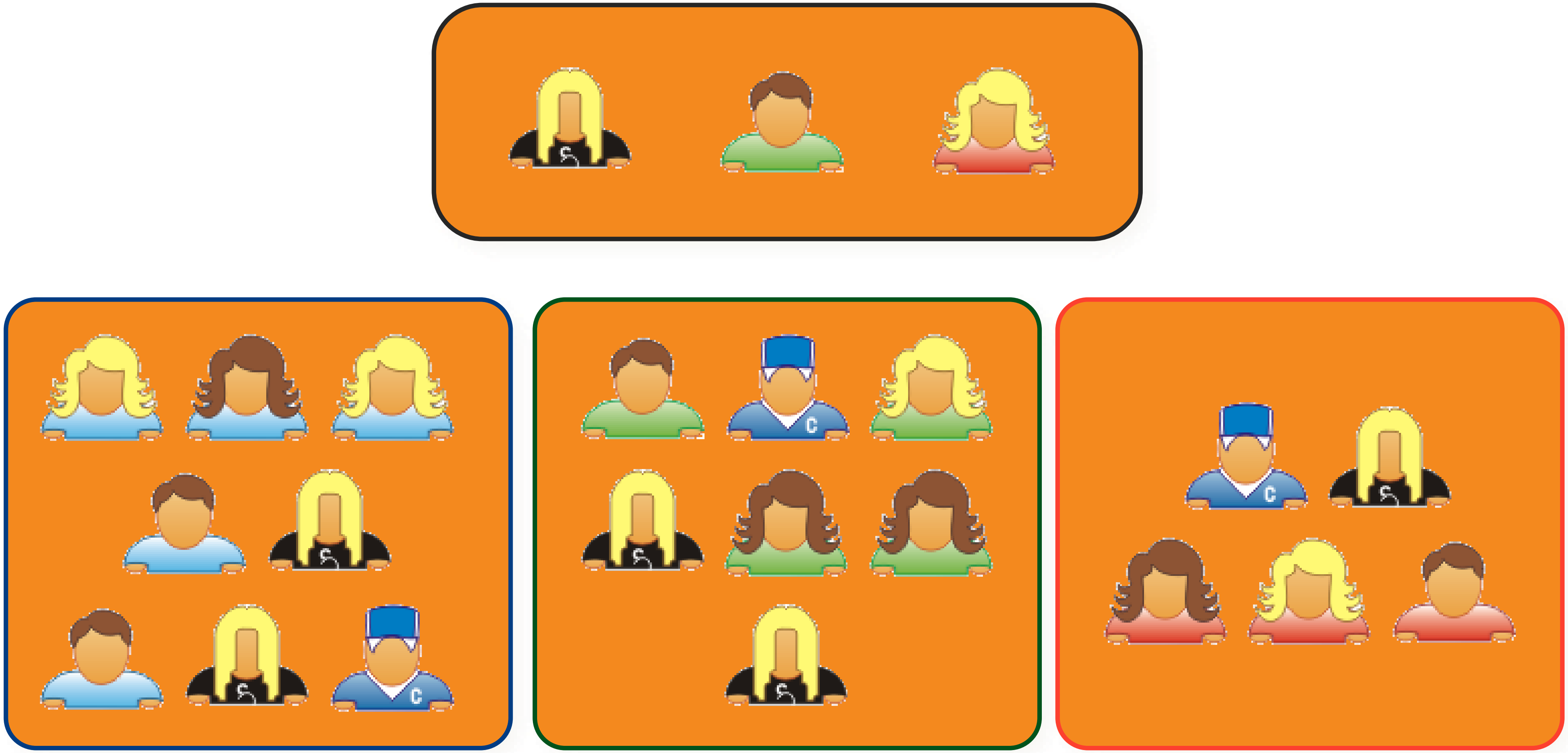


- 5-10 members; Teams are self-organizing
- Cross-functional: QA, Programmers, UI Designers, etc.
- Membership should change only between sprints

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: Scrum of Scrums



Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

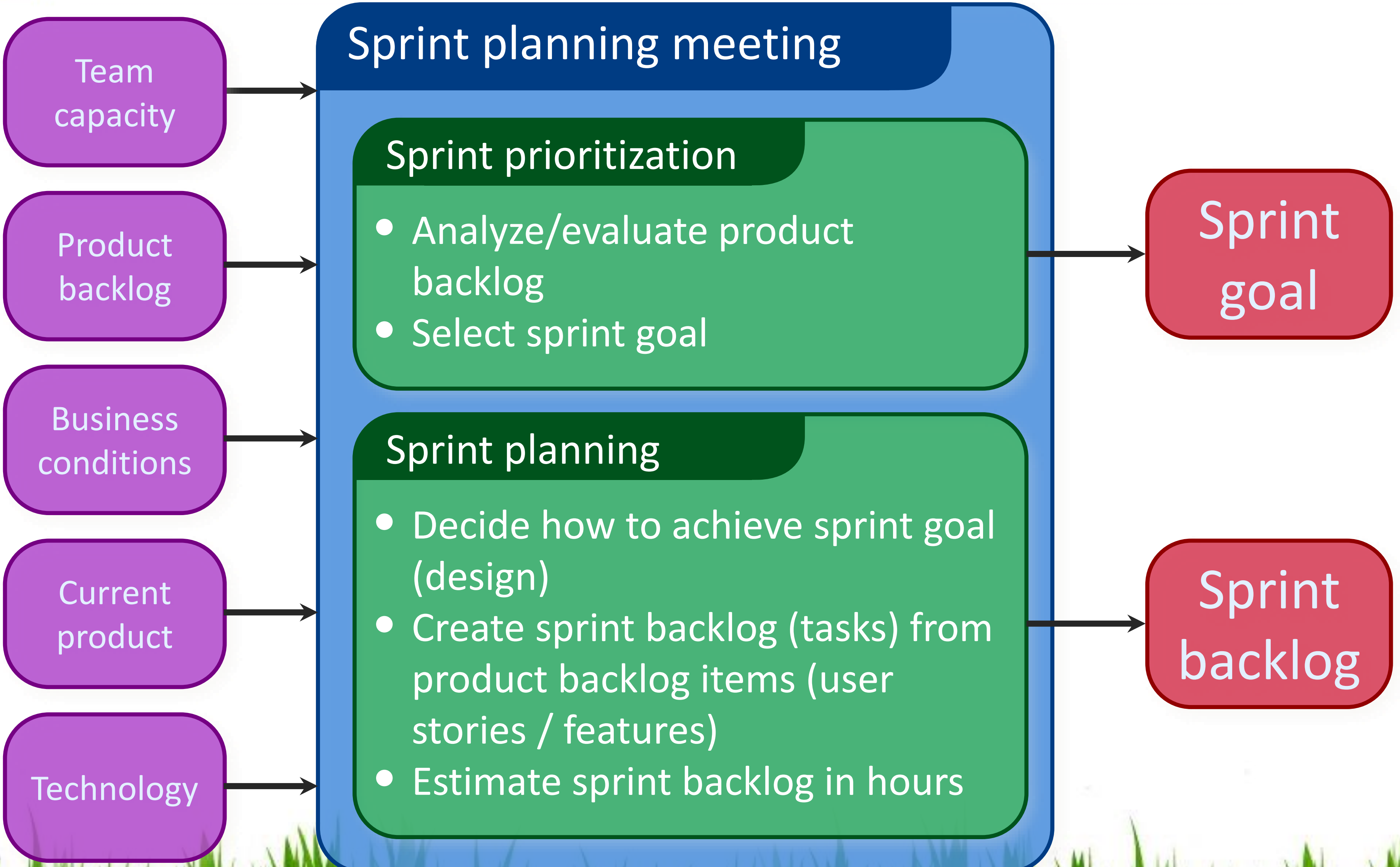
Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Sprint Planning Mtg.



Team capacity

Product backlog

Business conditions

Current product

Technology

Sprint planning meeting

Sprint prioritization

- Analyze/evaluate product backlog
- Select sprint goal

Sprint planning

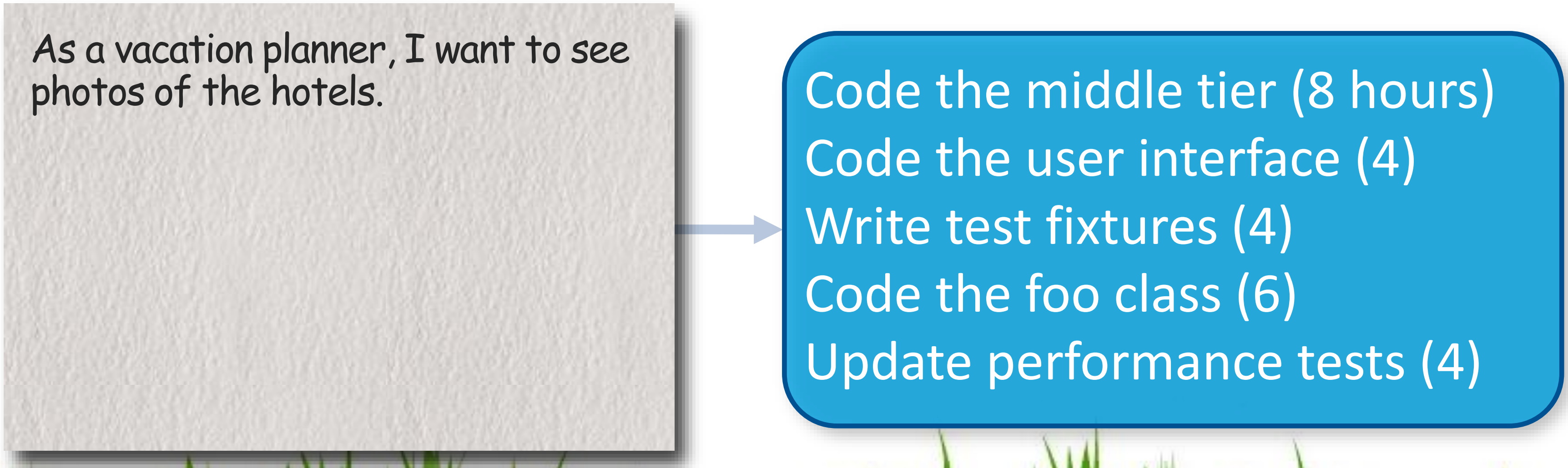
- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

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 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

Continue doing

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

Daily SCRUM Meeting

- Parameters
  - Daily, ~15 minutes, Stand-up
  - Anyone late pays a $1 fee
- Not for problem solving
  - Whole world is invited
  - Only team members, Scrum Master, product owner, can talk
  - Helps avoid other unnecessary meetings
- Three questions answered by each team member:
  1. What did you do yesterday?
  2. What will you do today?
  3. What obstacles are in your way?

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

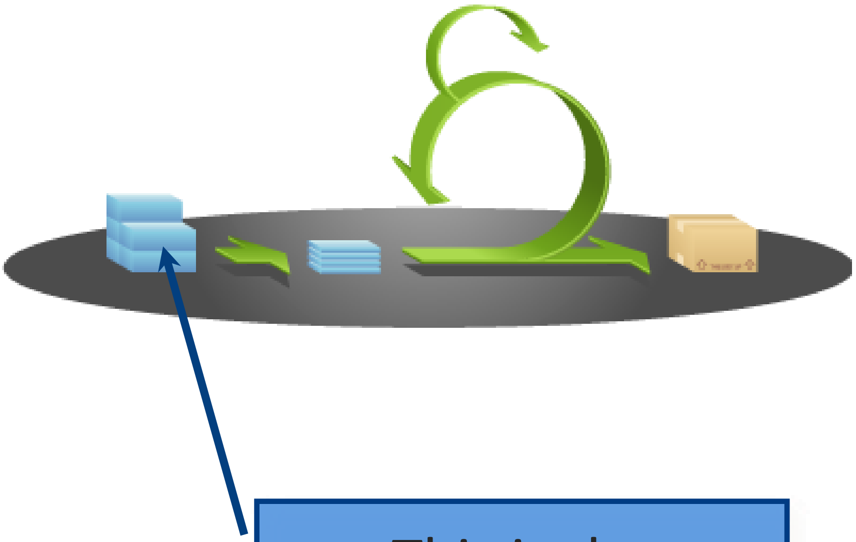
- Product backlog
- Sprint backlog
- Burndown charts

Scrum's Artifacts

- Scrum has remarkably few artifacts
  - Product Backlog
  - Sprint Backlog
  - Burndown Charts

- Can be managed using just an Excel spreadsheet
  - More advanced / complicated tools exist:
    - Expensive
    - Web-based – no good for Scrum Master/project manager who travels
    - Still under development

Product Backlog

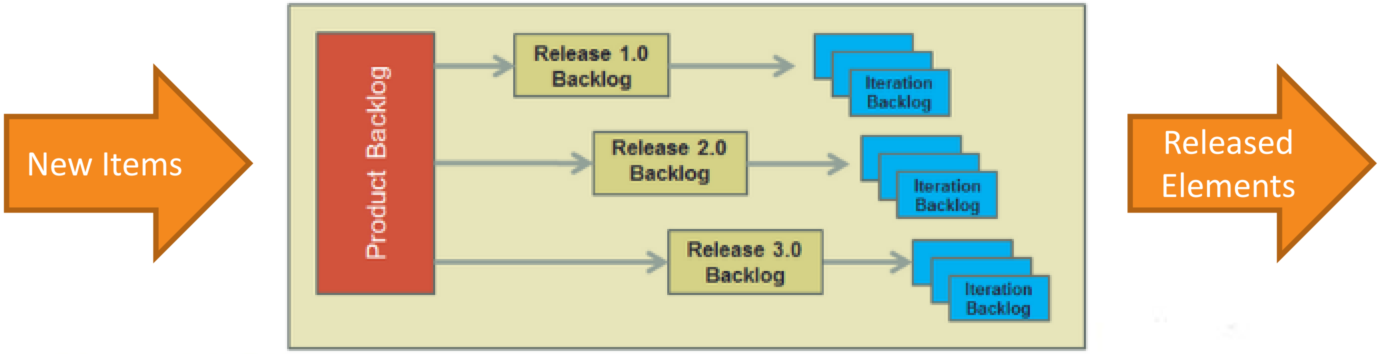


This is the product backlog

- The requirements
- A list of all desired work on project
- 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
- Prioritized by the product owner
- Reprioritized at start of each sprint

The Product Backlog Lifecycle

- The Product Backlog has no end – it is never finished unless the activity is not longer needed
- The Product Backlog should be groomed on a regular cadence
- The Product Backlog evolves to meet the Feedback received from customers and the marketplace



New Items

Product Backlog

Release 1.0 Backlog

Iteration Backlog

Release 2.0 Backlog

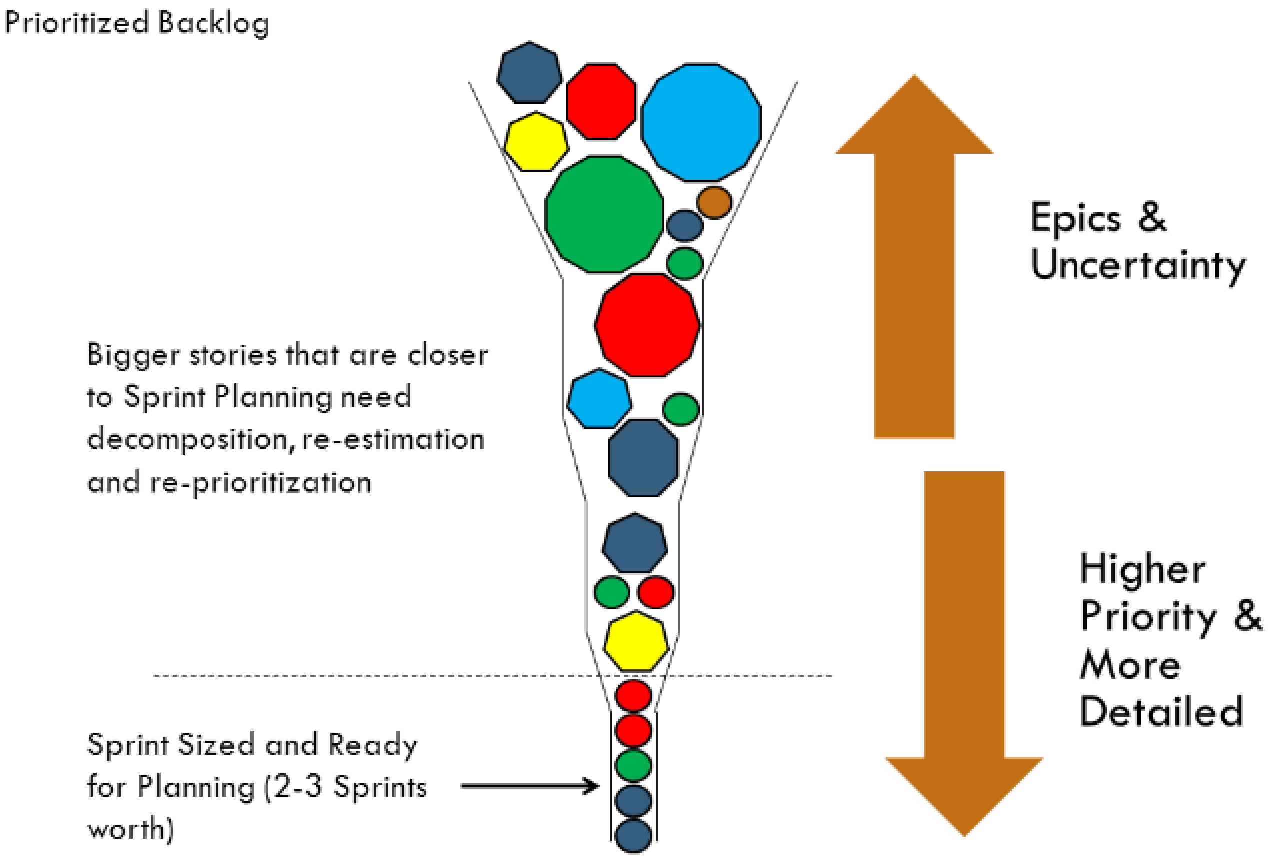
Iteration Backlog

Release 3.0 Backlog

Iteration Backlog

Released Elements

Building a backlog



Prioritized Backlog

Bigger stories that are closer to Sprint Planning need decomposition, re-estimation and re-prioritization

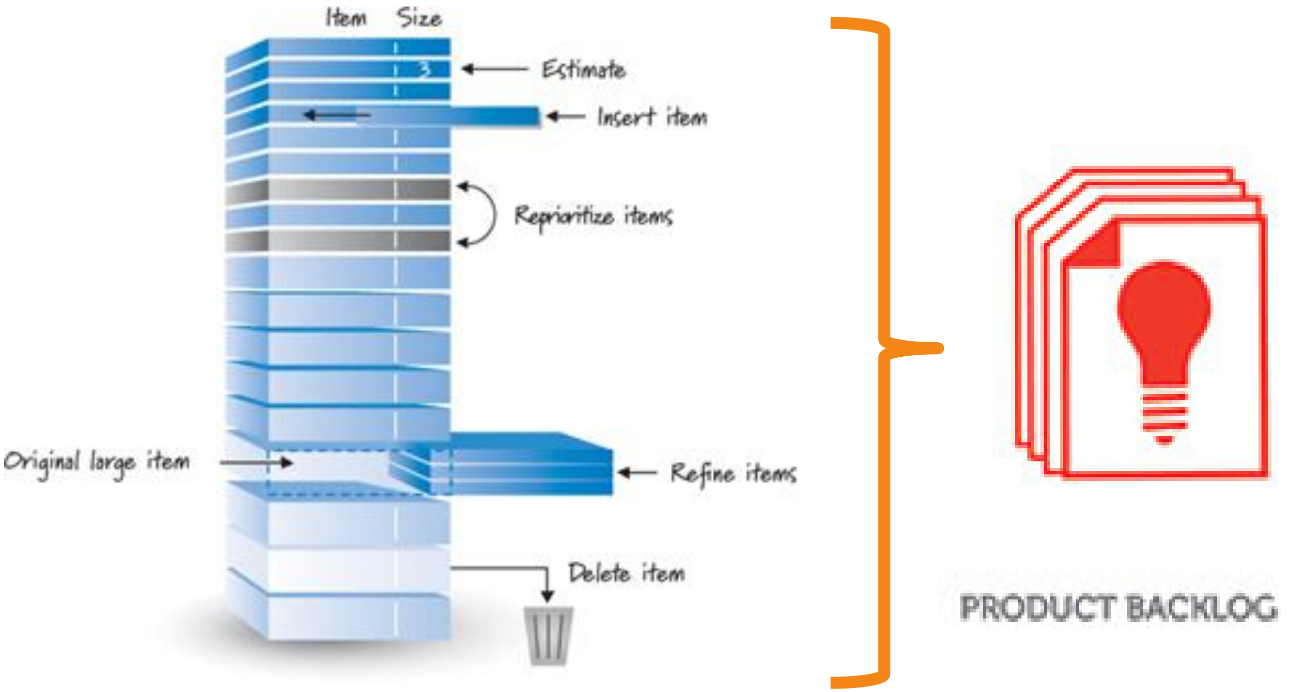
Sprint Sized and Ready for Planning (2-3 Sprints worth)

Epics & Uncertainty

Higher Priority & More Detailed

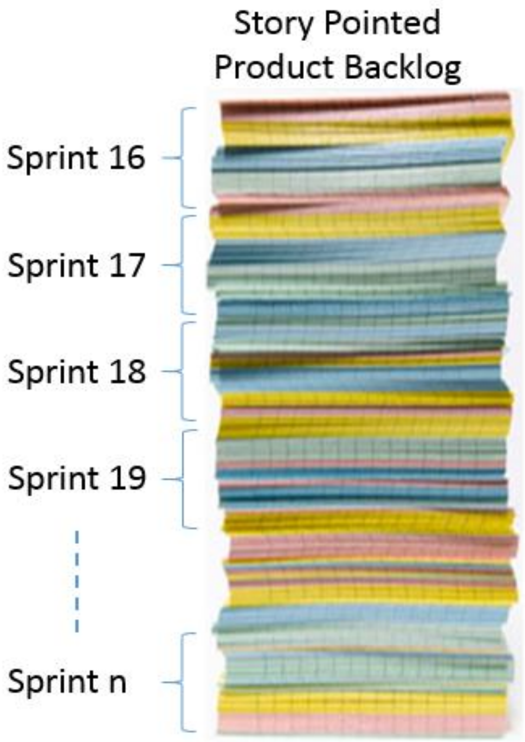
- The Backlog is made up of a hierarchy of Epics → Features → Stories
- Decompose an Epic into Features
- Decompose Features into Stories in preparation for Sprint Planning

A deeper view of the Product Backlog

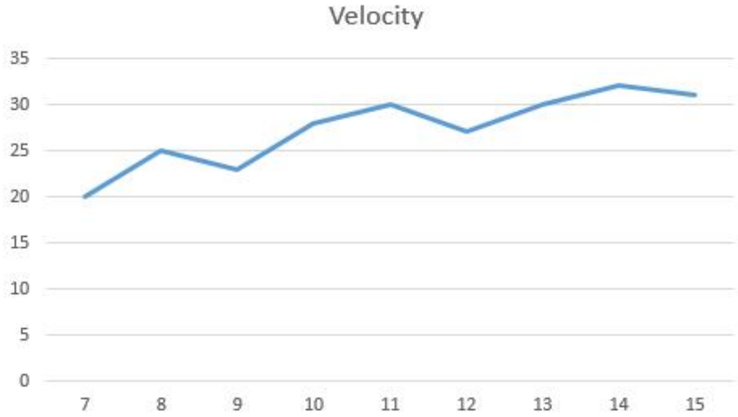


- Owned and prioritized by the Product Owner
- Anyone in the Org can add to the backlog
- Discipline: “If it is not on the backlog it doesn’t exist

Backlog grooming based on Team velocity



Average Capacity 27 Story Points



Sprint	Velocity
7	20
8	25
9	23
10	28
11	30
12	27
13	30
14	32
15	31

In Story Points

This effort generates a Sprint Roadmap that has been estimated by the team

- Once there is some experience with this model it will provide our monthly planning data

User Stories

- Instead of Use Cases, Agile project owners do "user stories"
  - **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?

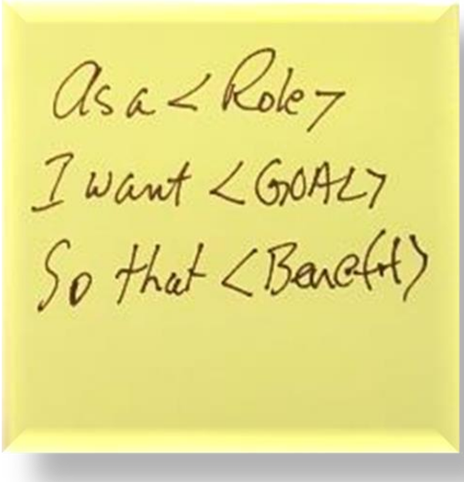
As a [user role], I want to [goal], so I can [reason].

- 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.

User Stories and the Backlog

- **ROLE** – is the user of the element
- **GOAL** – is what they want to accomplish
- **BENEFIT** – is the business value it provides



Describes the value from a users perspective of the value of the element to be developed

The backlog is a collection of Stories that are arranged in priority order by value

Product Backlog

Rqmts

- Epic
  - Feature
    - Story
      - Task

Description – details of the story, technical, logistic, timing, etc.

Acceptance Criteria”

- Give – initial condition
- When – stimulus
- Then – Final condition

User Story Examples

- As a consumer, I want shopping cart functionality to easily purchase items online.
- As an executive, I want to generate a report to understand which departments need to improve their productivity.
- As a developer, I want to publish the current state of my application to an update set, so that I can deploy it to a production system.
- As a customer, I want to receive notifications when an incident is commented, so that I am updated on the status.
- As a change manager, I want to enable the assessment of risk for any given change by establishing a list of questions with multiple choice answers.

Given that we have Emerging Requirements what do we do?

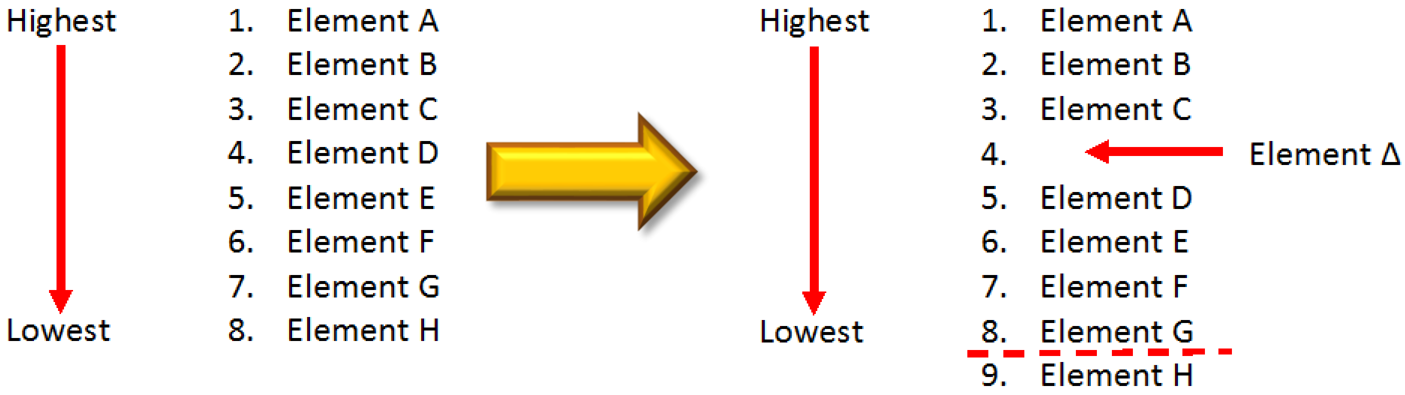
- First embrace that new requirements will be coming in as we progress through the project/program
- Review the business value of each new Feature or Story to ensure the highest business value
- Ensure that we are not over-specifying or adding extra features that add to wasted time and effort for which the customer or marketplace in not willing to pay
- Groom the backlog on a cadence so that new requirements can be added and the backlog order reprioritized

Method for determining an effective Story

Independent	• Able to be build in any order
Negotiable	• Make tradeoffs with Stakeholders
Valuable	• If there is no value, then it is a waste
Estimable	• There is enough information to estimate the effort
Small	• Small enough to be estimable
Testable	• Without testing there is no knowledge of achievement: Given, When, Then

Relative Priority to achieve a singularity

- Develop a single priority list which has the order of importance visible in the ranking



Highest

1. Element A
2. Element B
3. Element C
4. Element D
5. Element E
6. Element F
7. Element G
8. Element H

Lowest

Highest

1. Element A
2. Element B
3. Element C
4.
5. Element D
6. Element E
7. Element F
8. Element G
9. Element H

Element Δ

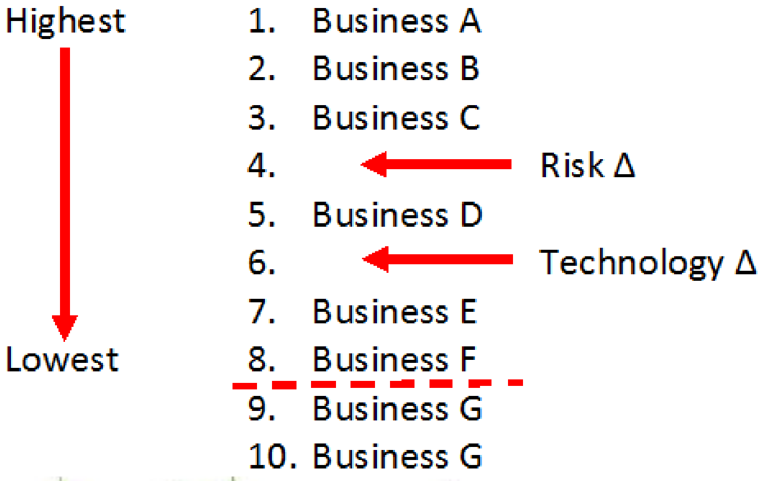
Lowest

- Using this method provided a means for adding, subtracting or changing order as new information is provided
- It clearly shows the relative priority of one element to another

Risk/Technology adjusting a Backlog

- Perform a risk analysis on the backlog to determine items that need to be learned early in the development cycle
  - Move the most risky element up in priority
- Review the backlog to see what technology items must be completed before business items
- Backlogs can be adjusted by

  ROI or Risk calculations



Highest

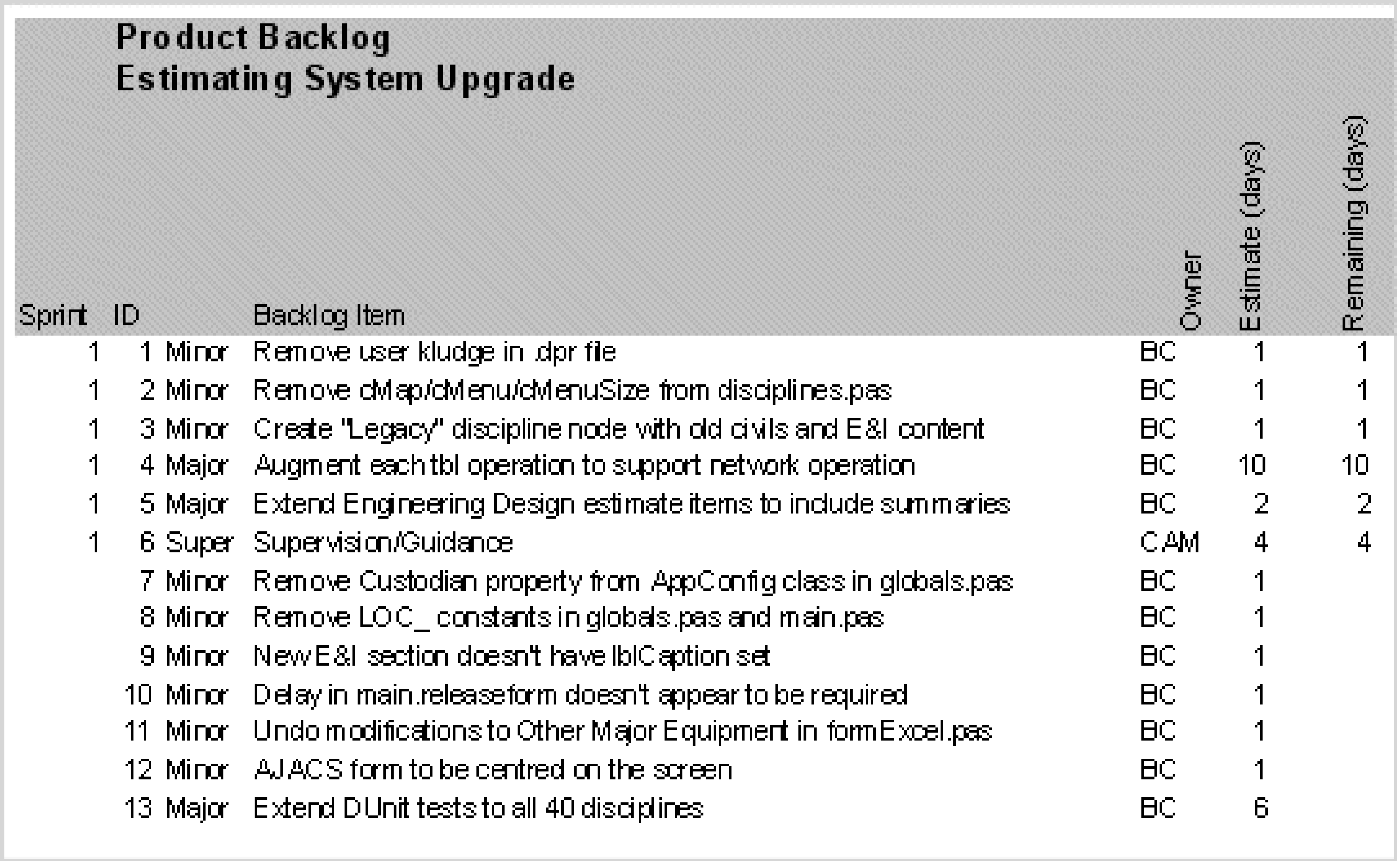
Lowest

1. Business A
2. Business B
3. Business C
4. ← Risk Δ
5. Business D
6. ← Technology Δ
7. Business E
8. Business F
9. Business G
10. Business G

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

Sample Product Backlog



Product Backlog
Estimating System Upgrade

Sprint	ID		Backlog Item	Owner	Estimate (days)	Remaining (days)
1	1	Minor	Remove user kludge in .dpr file	BC	1	1
1	2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	1	1
1	3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	1	1
1	4	Major	Augment each tbl operation to support network operation	BC	10	10
1	5	Major	Extend Engineering Design estimate items to include summaries	BC	2	2
1	6	Super	Supervision/Guidance	CAM	4	4
	7	Minor	Remove Custodian property from AppConfig class in globals.pas	BC	1	
	8	Minor	Remove LOC_ constants in globals.pas and main.pas	BC	1	
	9	Minor	New E&I section doesn't have lblCaption set	BC	1	
	10	Minor	Delay in main.releaseform doesn't appear to be required	BC	1	
	11	Minor	Undo modifications to Other Major Equipment in formExcel.pas	BC	1	
	12	Minor	AJACS form to be centred on the screen	BC	1	
	13	Major	Extend DUnit tests to all 40 disciplines	BC	6	

The Sprint Goal

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

Life Sciences

Support features necessary for population genetics studies.

Database Application

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

Financial services

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

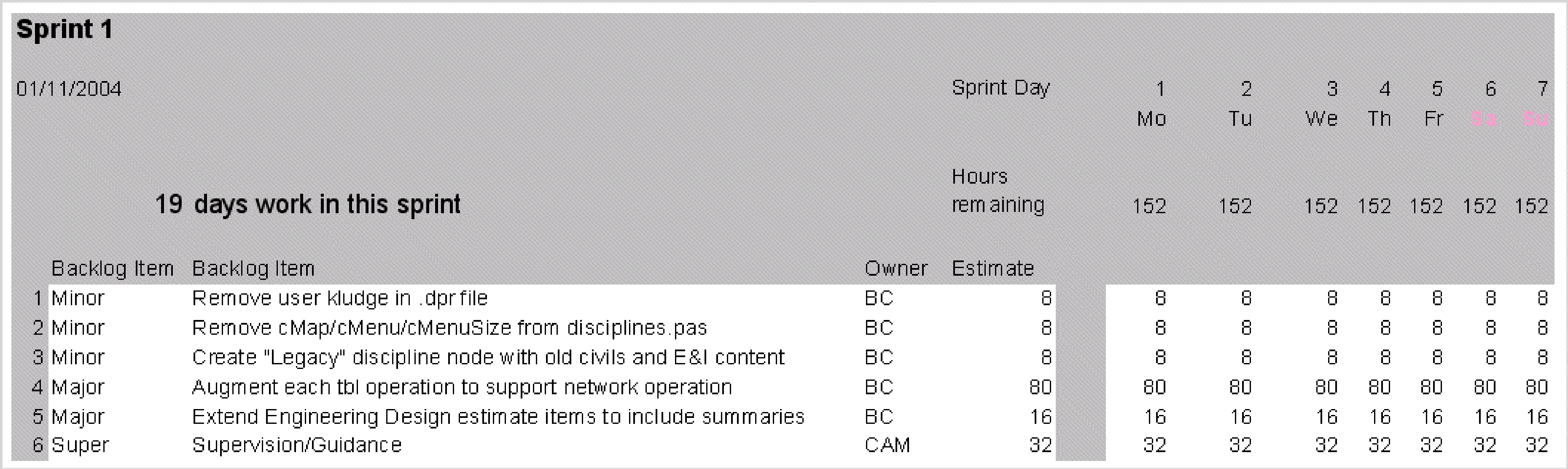
Managing the sprint backlog

- 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

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	

Sample Sprint Backlog



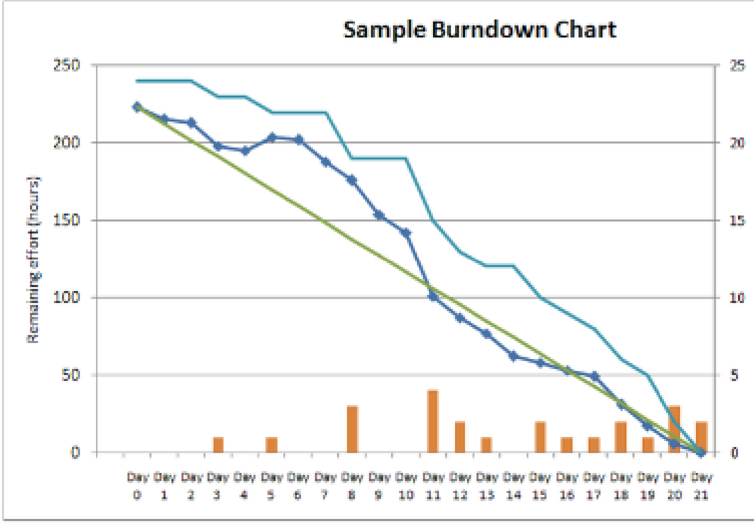
Sprint 1

01/11/2004				Sprint Day	1	2	3	4	5	6	7
					Mo	Tu	We	Th	Fr	Sa	Su
	19	**days work in this sprint**		Hours remaining	152	152	152	152	152	152	152
	Backlog Item	Backlog Item	Owner	Estimate							
1	Minor	Remove user kludge in .dpr file	BC	8	8	8	8	8	8	8	8
2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	8	8	8	8	8	8	8	8
3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	8	8	8	8	8	8	8	8
4	Major	Augment each tbl operation to support network operation	BC	80	80	80	80	80	80	80	80
5	Major	Extend Engineering Design estimate items to include summaries	BC	16	16	16	16	16	16	16	16
6	Super	Supervision/Guidance	CAM	32	32	32	32	32	32	32	32

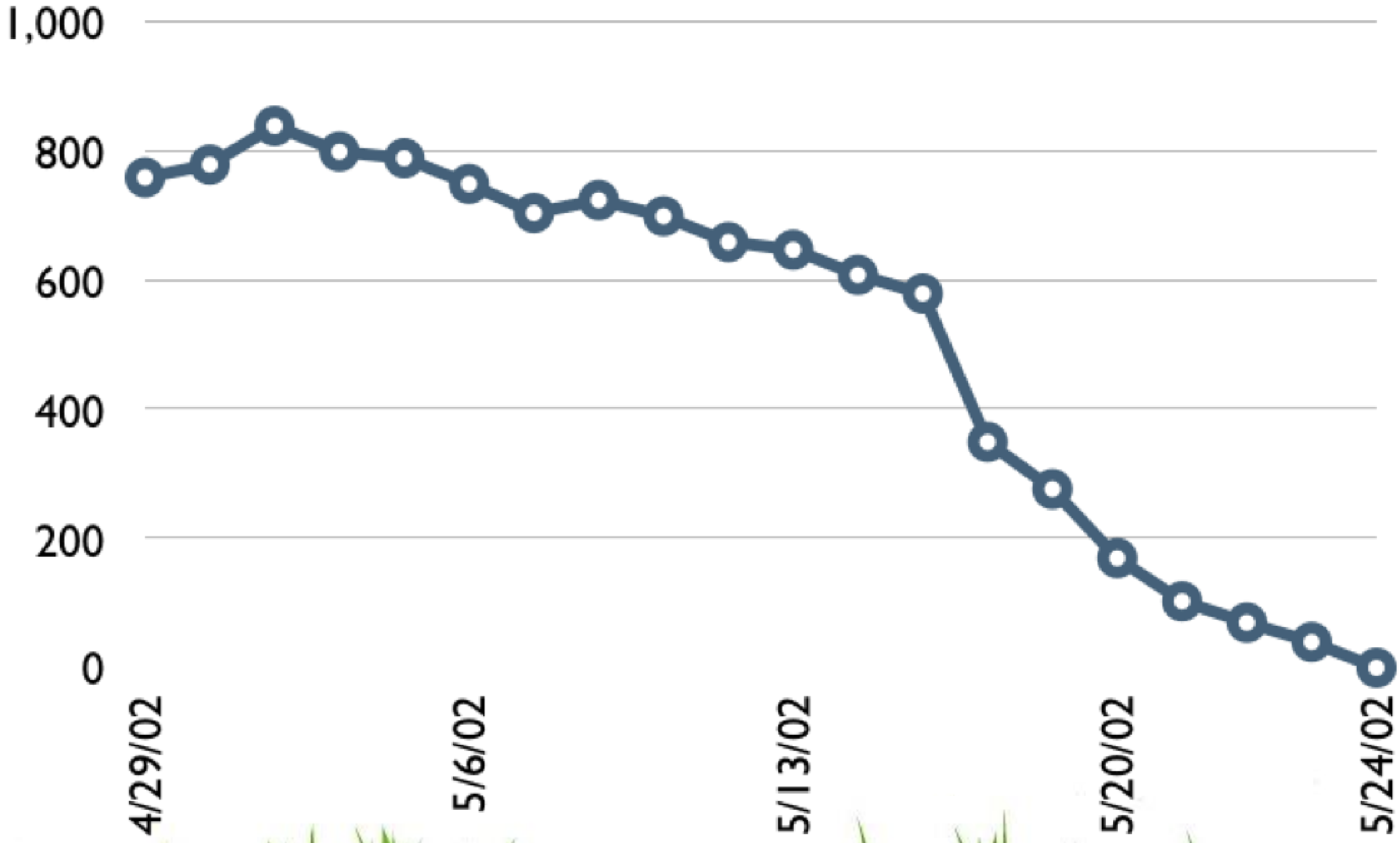
Sprint Burndown Chart

- 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)

- *variation:* Release burndown chart
  - shows overall progress
  - updated at end of each sprint

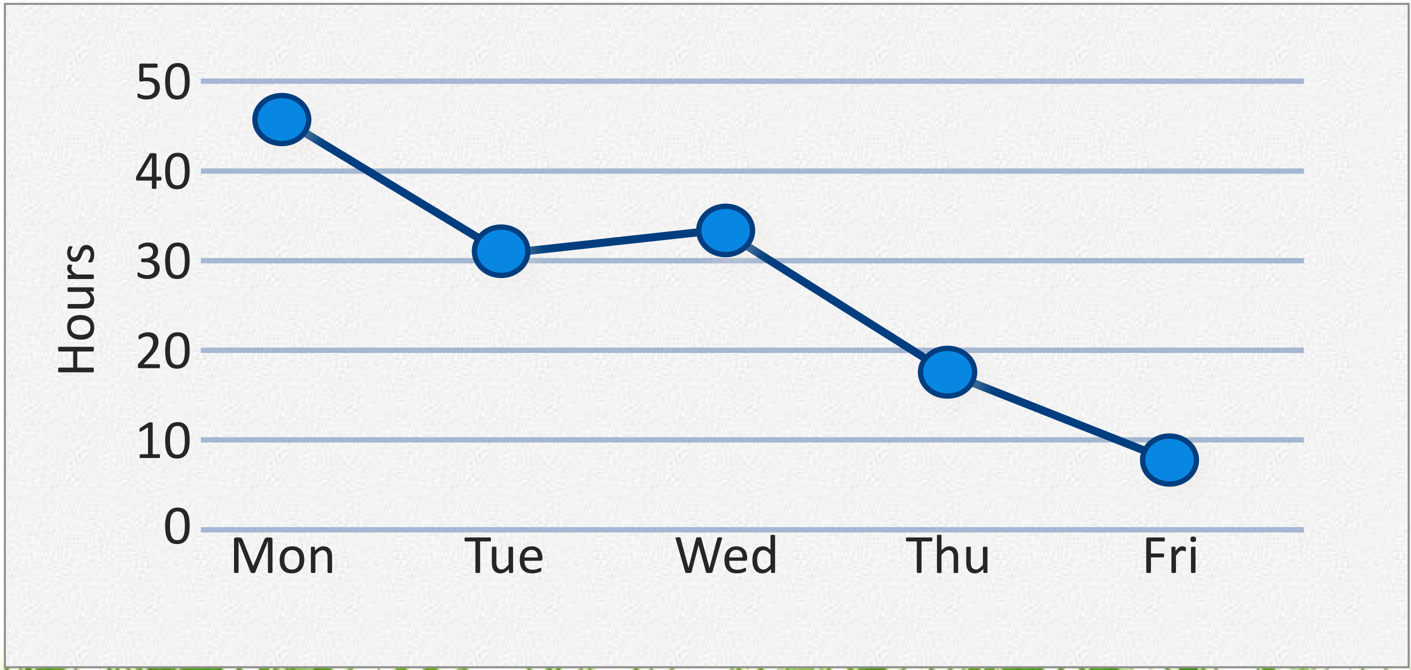


Sample Sprint Burndown Chart



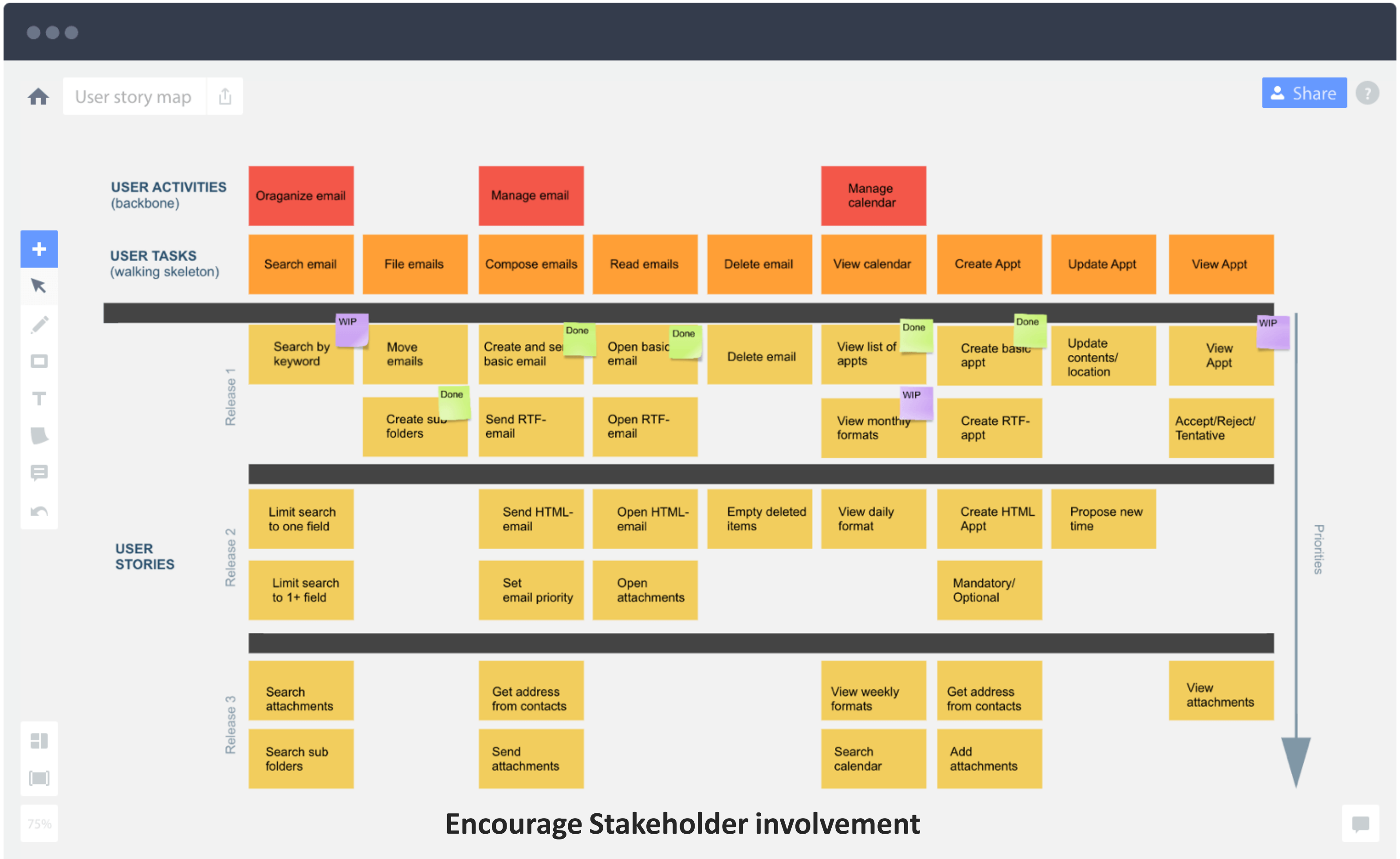
Sample Sprint Burndown Chart

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				



Product Roadmaps

Story map



USER ACTIVITIES (backbone)	Oraganize email		Manage email			Manage calendar			
USER TASKS (walking skeleton)	Search email	File emails	Compose emails	Read emails	Delete email	View calendar	Create Appt	Update Appt	View Appt
USER STORIES Release 1	Search by keyword WIP	Move emails	Create and se basic email Done	Open basic email Done	Delete email	View list of appts Done	Create basic appt Done	Update contents/ location	View Appt WIP
Release 1		Create sub folders Done	Send RTF-email	Open RTF-email		View monthly formats WIP	Create RTF-appt		Accept/Reject/ Tentative
Release 2	Limit search to one field		Send HTML-email	Open HTML-email	Empty deleted items	View daily format	Create HTML Appt	Propose new time	
Release 2	Limit search to 1+ field		Set email priority	Open attachments			Mandatory/ Optional		
Release 3	Search attachments		Get address from contacts			View weekly formats	Get address from contacts		View attachments
Release 3	Search sub folders		Send attachments			Search calendar	Add attachments		

Encourage Stakeholder involvement

Release map

Next Lecture

- XP