
# Scrum SW Development Projects

## HowTo Guide: SWEN90016 Group Assignment

Eileen O'Callaghan

# Previous Lecture

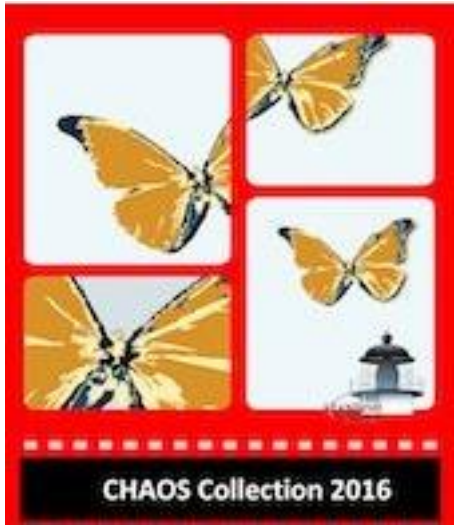
1. What is trending in IT, the knowledge industry?
   - Big Data
2. What is happening to products?
   - Quick Pivots
3. Why do products?
   - Strategic Initiatives, Team Building
4. When is an Agile project a good choice?
   - Adapt and respond to change
   - Requirements emerge

# Previous Lecture

- Fact Checker
  - Standish Group Authority
  - 2012 research outcomes
  - 2013 research outcomes
- Disadvantages of Agile
- Project Management used at NASA

# Fact Checker

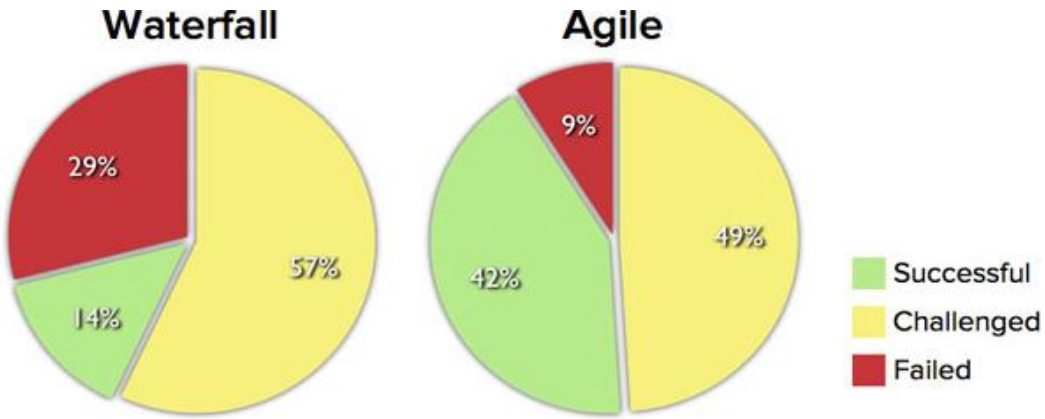
## Standish Group




- IT research advisors
- Focus on mission-critical software applications
- Focus on failures and possible improvement
- Reports must be purchased and are expensive
  - www.standishgroup.com/

# Projects Metrics 2012

## The CHAOS Manifesto, Standish Group




Source: The CHAOS Manifesto, The Standish Group, 2012.

- Refer to: www.slideshare.net/Placosta/chaos-report-2012
- Refer to: www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall

# Projects Metrics

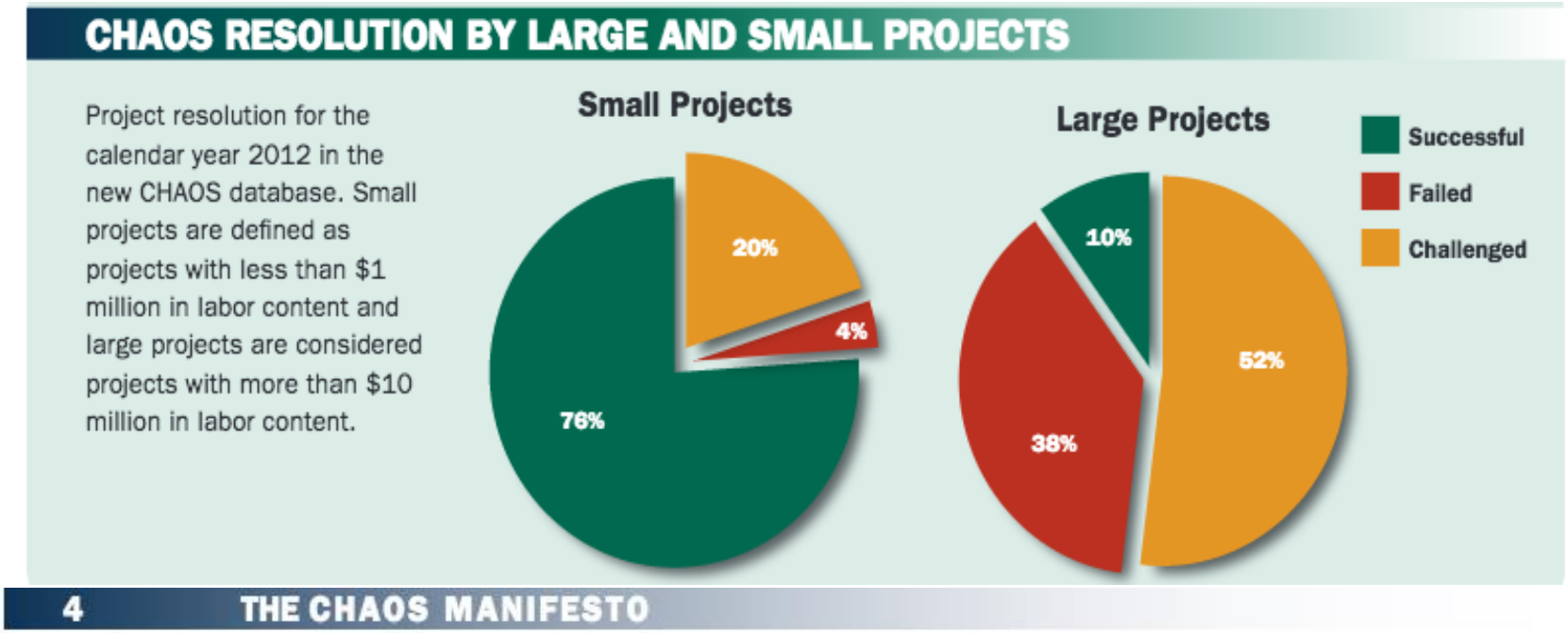
## The CHAOS Manifesto, Standish Group

Do their claims have authority?

- Original report is updated regularly
  - updates make information unstable
  - archive is not readily accessible
- Do not publish their research method
  - sampling of projects
  - how measures are defined
  - how they asked
  - who they asked
- Do not release their raw data
- The full report is expensive, not readily accessible

# Projects Metrics 2013

## The CHAOS Manifesto, Standish Group



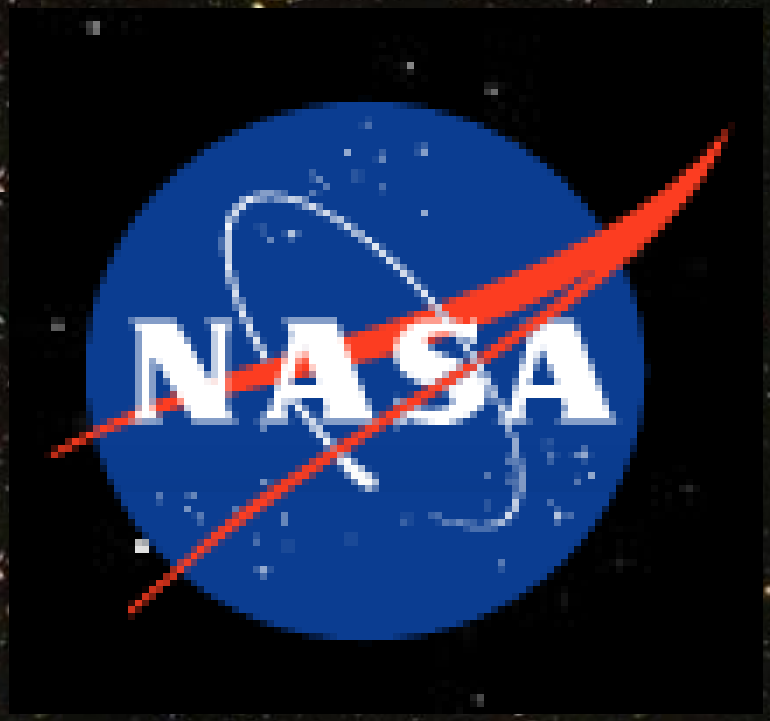

– Refer to: larlet.fr/static/david/stream/ChaosManifesto2013.pdf

# Disadvantages of Agile

- Scrum – has Sprints, self-organizing team
  - + Deliver chunks of stable and packaged code
  - + Time boxed Sprint fosters design opportunity
  - ? Meetings surround each Sprint can become ceremonial
  - **- Maintenance of code less documented and easy**
  - **- Quality less trusted**
  - **- Contracting and subcontracting less supported**

- Kanban – intuitive, visual, smooth work flow
  - + Widely adopted
  - + Deliver code with “Just-In-Time” efficiently
  - - Production line of features, without design frame

# What would NASA do?



# An Agile project for Rocket Flight Software: 2017

Development of mission-critical human-rated embedded flight software

Marshall brings a responsive, agile approach to program and project development efforts. The Marshall flight software team performs the complete range of flight software activities, including requirements development and analysis, software processes and planning, design and development, systems integration, and development testing. Marshall also provides the facilities for flight software development and testing and software formal verification through the development and management of test activities.

Marshall was NASA's first field center to achieve CMM Level 3 required for human missions, is an early adopter of UML, uses agile software development techniques, and continually embraces new approaches and tools to be more efficient in software development.

Marshall's expertise in real-time hardware-in-the-loop (HWIL) capabilities complements its software development efforts by enabling the integrated software and avionics hardware systems of launch vehicles to be modeled, simulated, and tested early, before finalizing designs. The modular HWIL approach is easily extensible to multiple types of spacecraft,

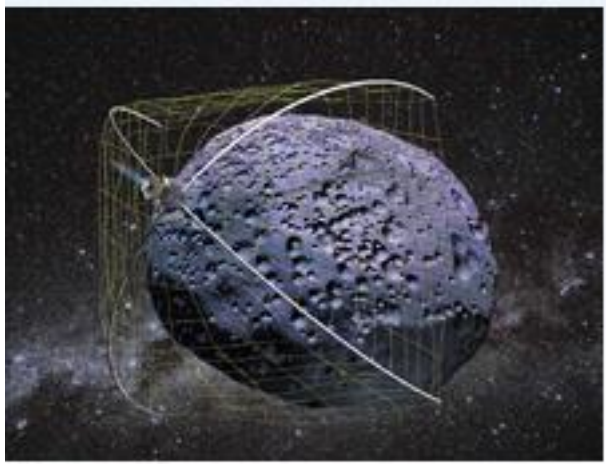
**At-A-Glance**

Flight software is critical to mission success, both in development and execution. For Class A and human missions, it also must meet exceedingly stringent requirements. Marshall's flight software team was the Agency's first to be certified as Capability Maturity Model (CMM) Level 3 (required for Class A and human missions) and an early adopter of agile modular development and industry standards such as Unified Modeling Language (UML). Combining robustness with agility, Marshall has the capability to design flight software for NASA's future flagship science and human exploration missions.

- https://www.nasa.gov/sites/default/files/files/FlightSoftware.pdf

# Wrangler Project: June 4 2014

- an incremental development program
- capture and de-spin asteroids and space debris while minimizing risk to the primary spacecraft
- validates the technology with test flights



www.nasa.gov/content/wrangler-capture-and-de-spin-of-asteroids-and-space-debris/

# Industry Best Practices in Project Management and Safety Assurance

- Each project has a Project Management Plan that includes at minimum:
  - Gantt schedule
  - Objectives
  - Requirements
  - Reference to Safety Plan
  - Reference to Data Mgmt. Plan
- Components of plans shared and confirmed by each customer
- plan is updated throughout the duration and customer apprised of updates

https://www.nasa.gov/offices/ocio/itsummit/index.html

# Purpose of lecture

- Show the Group Scrum Assignment
  - Ask questions please!
- How to participate in an Agile process
- How to do Reporting and Monitoring

# Group Assignment Artifacts

- Team Administration
  - Meeting Agenda
  - Meeting Minutes
  - Team Timesheet
  - Individual Reflection
- Assignment Specification
- myFarmXchange Case Study

# The Assignment

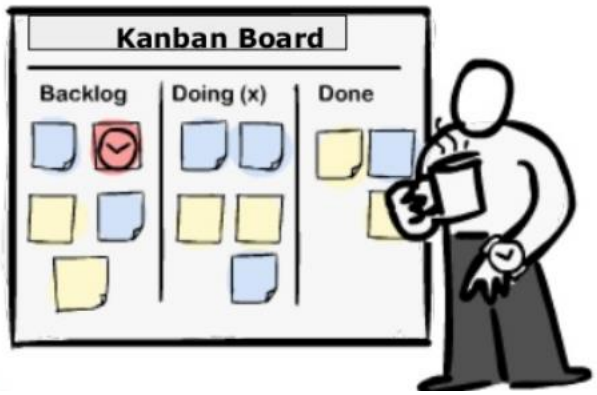
- Part a: the teamwork component 5%
  - » 19th May 2017 5pm Friday
- Part b: the draft Narrative Overview 0%
  - » 24th April 2017 5pm Monday
- Part c_i: an initial draft PMP and SDLC 10%
  - » 5th May 2017 5pm Friday
- Part c_ii: an updated and final PMP and SDLC 15%
  - » 19th May 2017 5pm Friday

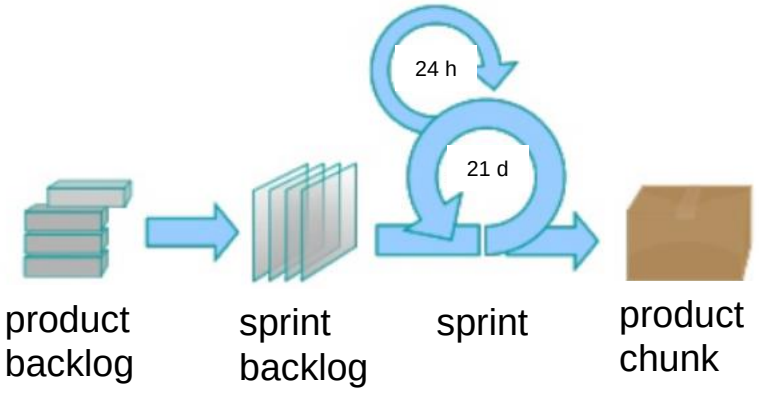
# PMP Document Overview

## Assignment Specification

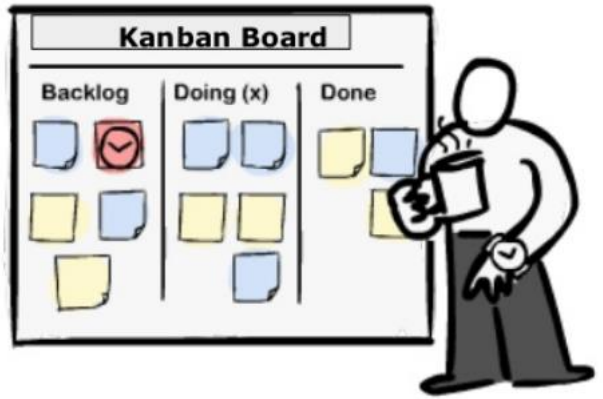
| | |
|---|---|
| – Introduction: | section 1 … 4 |
| – Scrum PMP: | section 5, a … h |
| – Narrative Overview: | section 6 |
| – Solution Overview | section 7 |
| – Scrum SDLC: *hypothetical* | section 8, a … f |
| – Summary | section 9 |
| – References | section 10 |

# Agile Features in SWEN90016



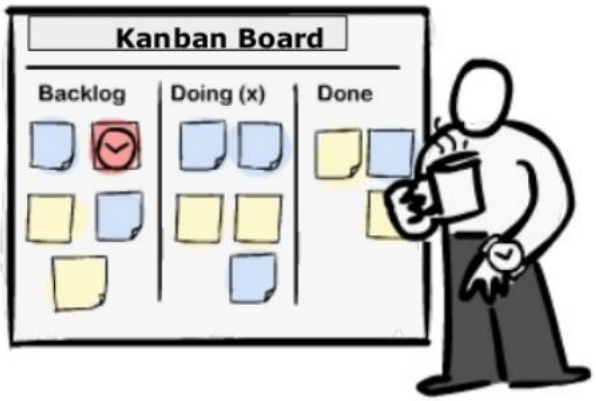
# Kanban Philosophy




1. Start with what you do now
   - No specific roles or processes
2. Pursue incremental, evolutionary change
   - Continuous small changes that will not generate hostility
3. Respect the current standard way
   - Gain support and foster good team karma

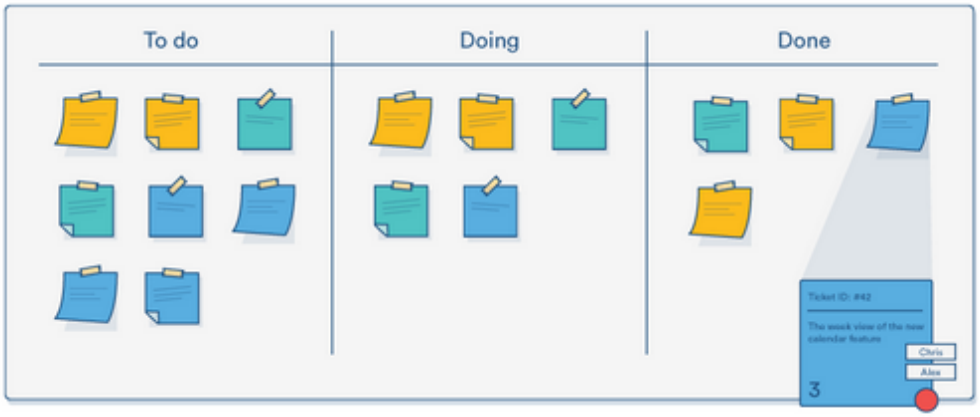
# Kanban Methods

1. Visualize the work flow
   - Divide board into swimlanes
2. Limit Work_In_Progress
   - Pull tasks from ”todo” swimlane
3. Manage Flow
   - Visual board make monitoring obvious
4. Make any policy change explicit
   - Accurate description of any process
5. Collaborate
   - Use scientific measures before adopting improvements



# Kanban Visualization

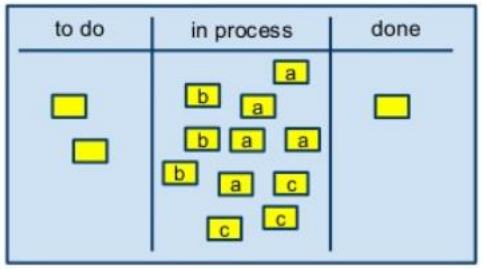
- Start with 3 swimlanes in a task board
- Name swimlanes: “todo”, “doing” and “done”
- Add User Story card to represent a work task
- Associate a team member’s name to card in “doing” swimlane




- Create an area for team social interaction

# Kanban – naïve

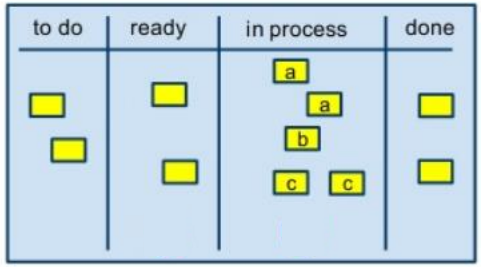
- What problem does this board demonstrate?
- Team members are attempting to multi-tasking
- There is too much work in “doing” swimlane




- Allow one “doing” item each member, for good focus

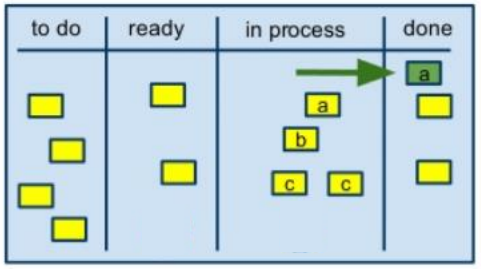
# Kanban – more specific

- Regulate the work in “doing” swimlane
- Assign member to task Just_In_Time

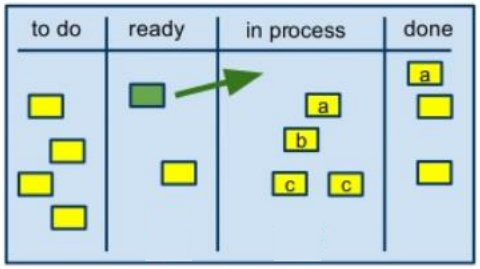



- Define work status more precisely
- Create “ready” swimlane

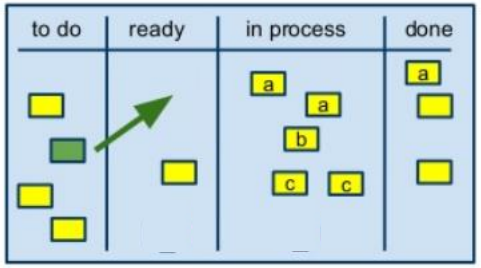
# Kanban Flow




Team member A completes a card and moves it to "done"




Team member A "pulls" a new card from "ready" and moves it to "doing"

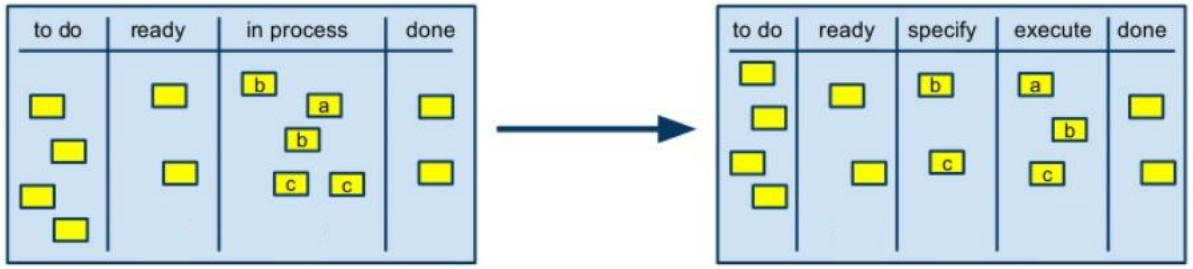



The whole team selects the next priority card and moves it to "ready"

The *"pull"*

# Velocity

- Create optimum number of swimlanes

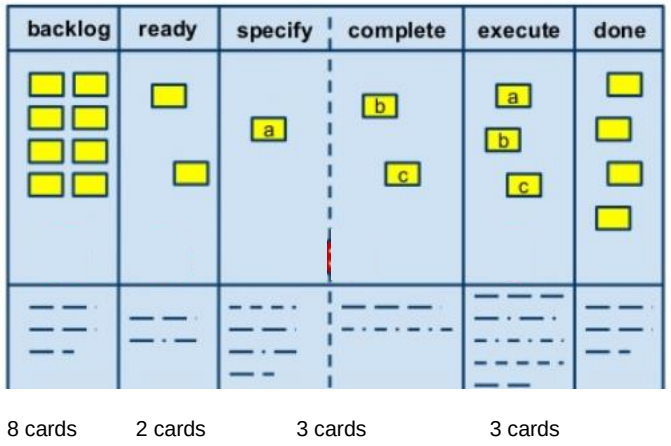



- How many cards where done over the time-boxed Sprint?
- Establish team velocity over time

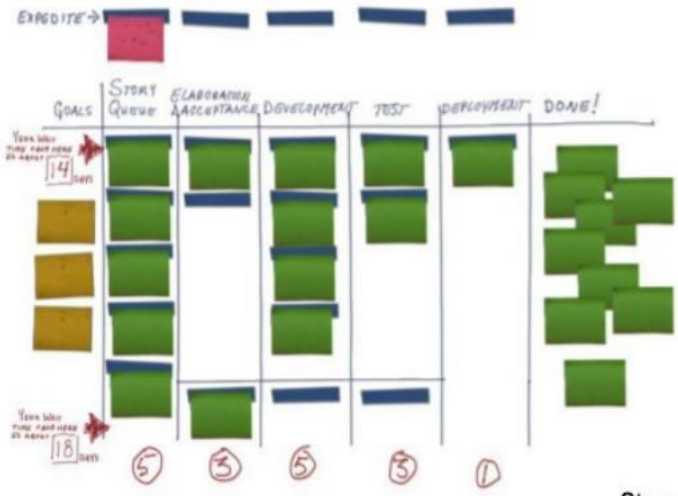
- Visual metrics are intuitive, transparent progress

# Board Summary

- Create optimum number of swimlanes
- Define work status precisely, Just_In_Time for max productivity
- The ideal work flow is a smooth & sustainable pace
- Pull card across board

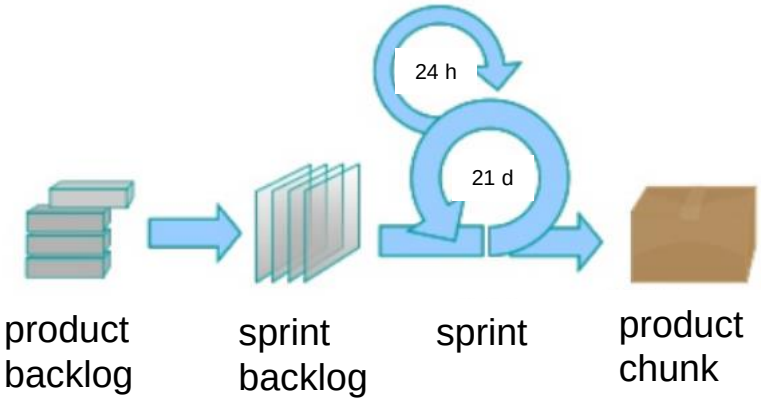


# Different Visual Boards



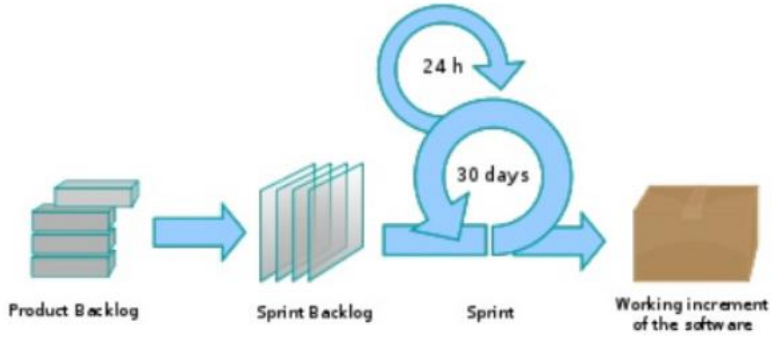

| Story | To Do | | In Process | To Verify | Done |
|---|---|---|---|---|---|
| As a user, I.. 8 points | Code the.. 9 | Test the.. 8 | Code the.. DC 4 | Test the.. SC 6 | Code the.. |
| | Code the.. 2 | Code the.. 8 | Test the.. SC 8 | | Test the.. |
| | Test the.. 8 | Test the.. 4 | | | Test the.. SC |
| | | | | | Test the.. SC |
| | | | | | Test the.. SC 6 |
| As a user, I.. 5 points | Code the.. 8 | Test the.. 8 | Code the.. DC 8 | | Test the.. SC |
| | Code the.. 4 | Code the.. 6 | | | Test the.. SC |
| | | | | | Test the.. SC 6 |

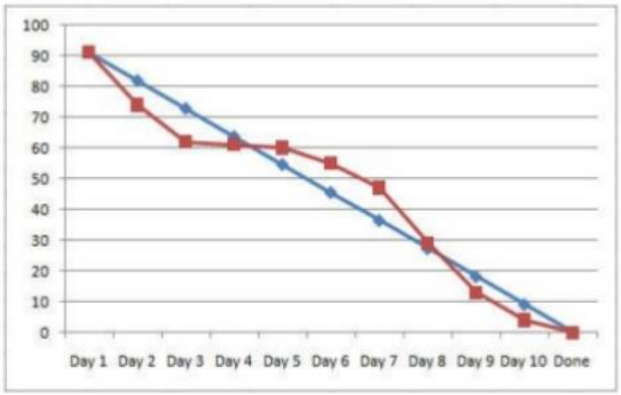
# Scrum Philosophy




- Focus on delivering high value is shortest time
- The business sets the priorities, the team self-organize how to deliver
- Strive for maximum stability on fixed User Stories during each Sprint
- Measure progress in actual working software
- Every Sprint show-cases working software to all interested stakeholders

# Scrum Methods




## Roles

Product Owner
Scrum Master
Development Team

Pigs and Chickens?

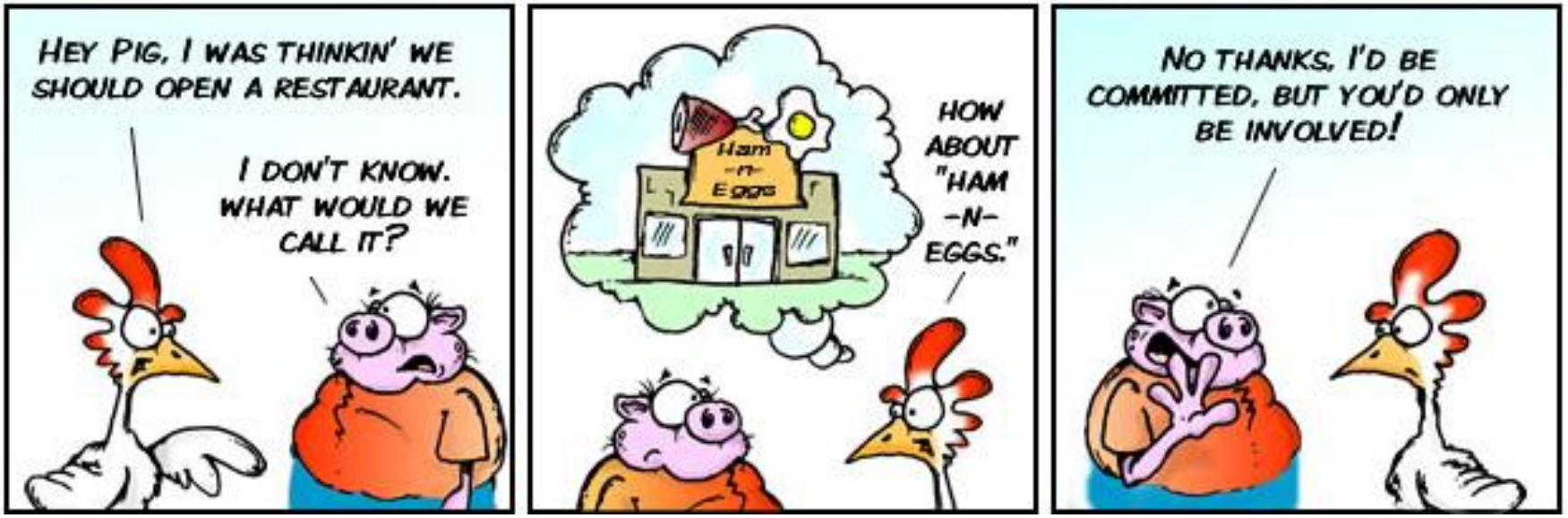
## Ceremonies

Daily Stand Up
Sprint Planning
Sprint Review
Sprint Retrospective

## Artifacts

Product Backlog
Sprint Backlog
User Stories
Burndown Chart
Burnup Chart

# Scrum Metaphor




## Roles (retired)

Pigs are committed to the team success. **Think of developers**

Chickens are observers, someone who has something to gain by the Pigs performing, but do not contribute day to day to "getting things done." **Think of Scrum Masters, users, …..**

# Scrum Artifacts

- User Stories
  - *As a <user>, I want < goal> so that < reason>.*
- Product Backlog
  - Tasks listed in client priority order
- Sprint Backlog
  - Tasks selected for this project release
- Burn Down Chart
  - represents the amount of work done
- Burn Up Chart
  - represents the amount of work remaining

# Why User Stories?

## Easier to communicate with users

- Users do not need to be trained to understand User Stories
- Real time feedback
- Encourages collaboration

## Simplified Plan – if it's too big & you can't estimate, make it smaller

- Avoids locking in design detail too early
- Leave technical functions to the architect, developers, testers
- Never out of date … Just In Time

## The Product Backlog is made up of Epic User Stories

- List the stories in order of importance => "Groom the Backlog"

# User Story Size

## User Story

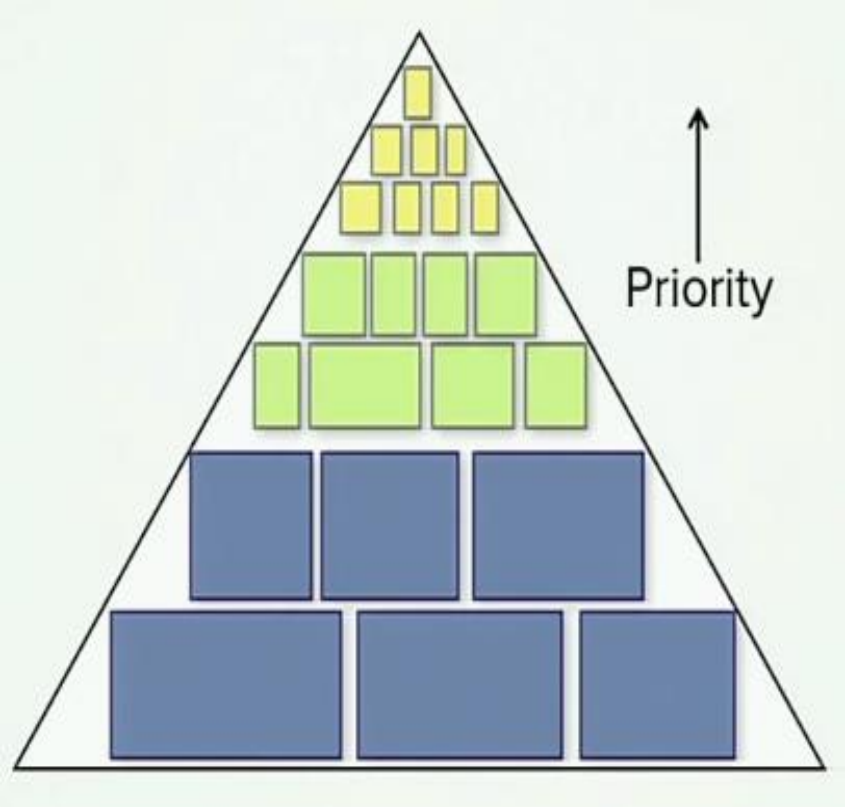
- A planning item
- A conversation placeholder

## Feature

- Capabilities the Product Owner values
- Value realized by multiple User Stories

## Epic

- Large initiatives delivering new services
- A collection of features



# User Story Template

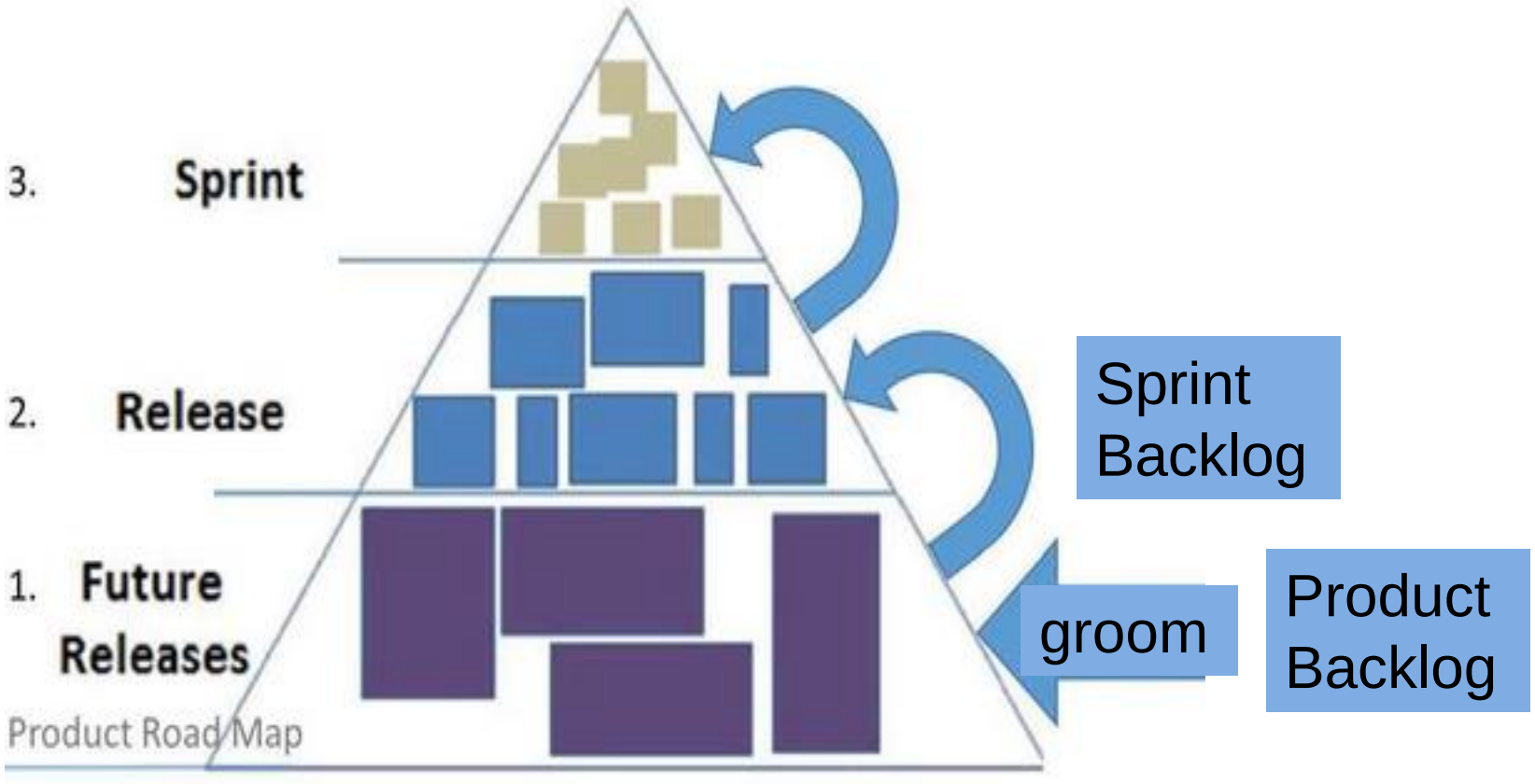
- PMP Speculate Stage:
  - create Epics from the high-level features
- Template structure:

  ***"As a <type of user>,***
  ***I want <some goal> ,***
  ***so that <some reason>"***
- Examples:
  - Epic: As a Marketing Executive, I want to review previous campaigns, so that I can repeat profitable decisions
  - Feature: As a registered user, I want to log in, so I can access the system
  - User Story: As a forgetful user, I can request a password reminder email, so that I can still log in

# User Story Common Mistakes

- Too formal / too much detail
  - Could result in skipping the conversation
  - Risk moving in the wrong direction
  - Overlooking specific customer needs
- Technical tasks impersonating user stories
  - Does this represent what the user wants?

# Scrum Backlogs



# Narrative Overview

Provide a **non-technical** reader an understanding of the project and the solution.

- Build trust with the client, and establish you have understood their Case Study.
- List all the features you've gathered from the Case Study.

## Product Backlog

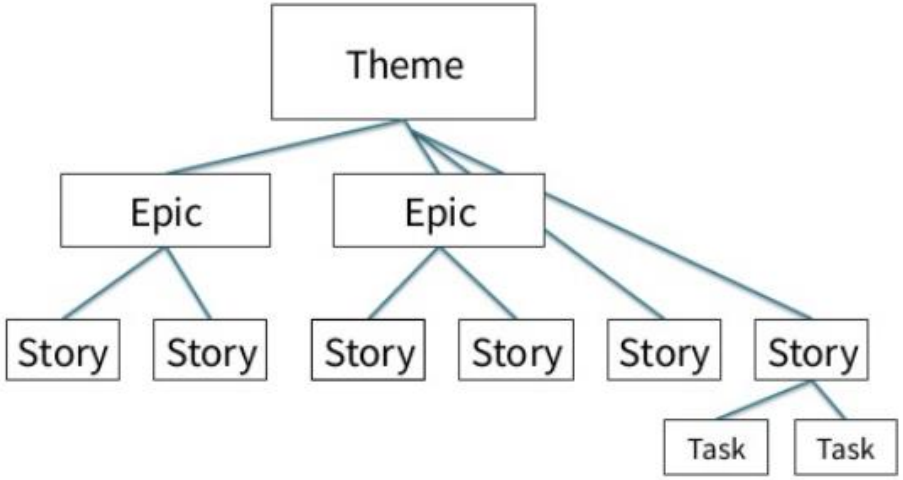
- Assume you have the Scrum role of the Product Owner
- Groom the Product Backlog and list in order, from highest value to lowest value
- This ordered list will determine the scope of future projects
- Low priority User Stories never get done

# Solution Overview

## Sprint Backlog

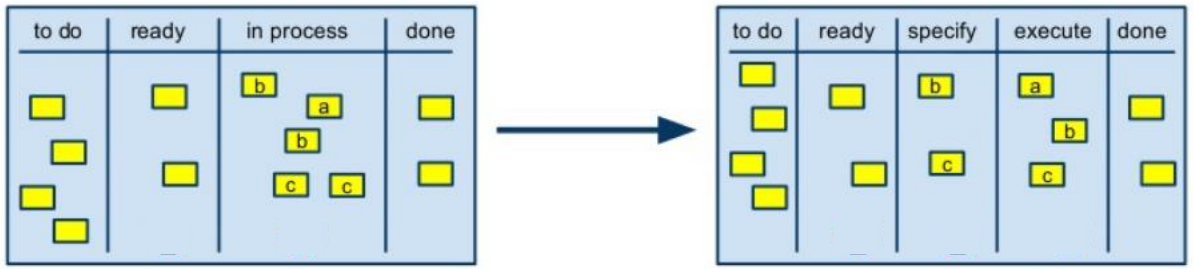
.

- Describe your proposed Release
  - select from the Product Backlog into the Sprint Backlog
- Identify the boundary between what features are included and excluded.
- Use appropriate diagrams.



# Velocity

- Use visual Kanban process
- How many User Stories are done over the time-boxed Sprint?
- Only count complete stories

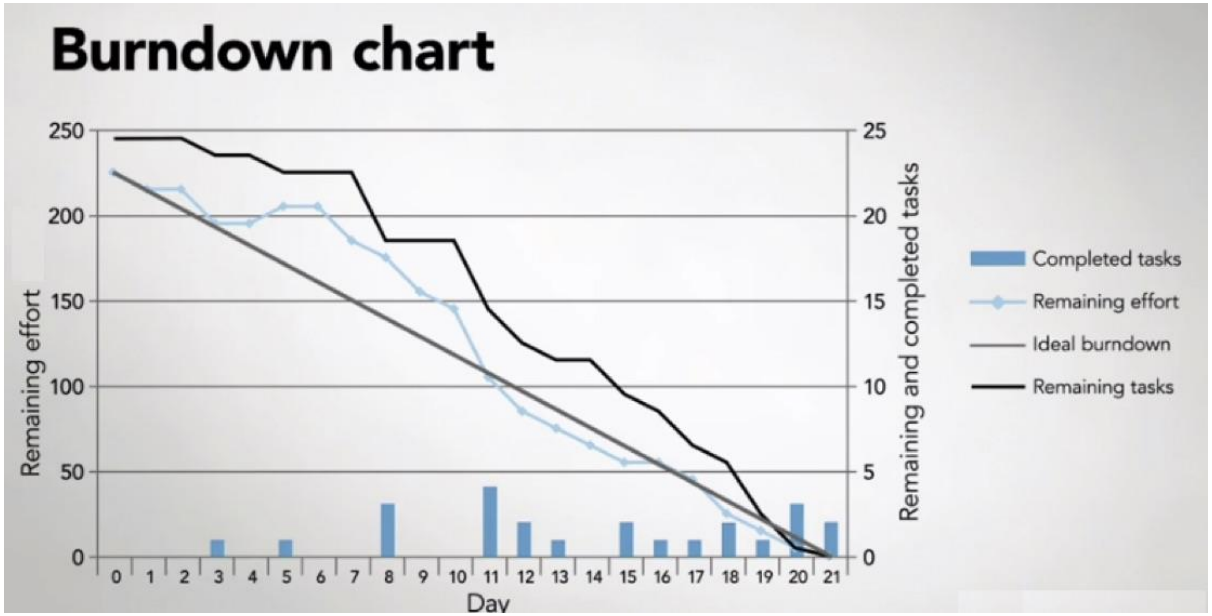



- Establish reliable team velocity over a number of Sprints
- When you'll be able to deliver all User Stories?

# Scrum Burn Down

Track remaining effort

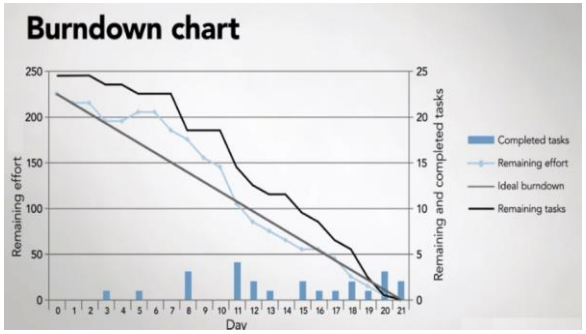
- Y-axis is the remaining Story Points
- X-axis is the elapsed time



# Burn down Example

21 day Sprint

250 Story Points of work




- plan: do 12 Story Points per day
- day 0 – 5: plan to completed 80, leaving 170
- day 5: estimated work remaining actually burned up, to 205
- day 10 estimated work remaining 140
- days 5 – 10: velocity was (205– 140) / 5 = 13 per day
- days10 – 15: velocity was (140– 55) / 5 = 17 per day
  - » velocity starts slow and tends to increase
- days 15 – 20: velocity was (55– 5) / 5 = 10 per day

# Scrum Burn Up

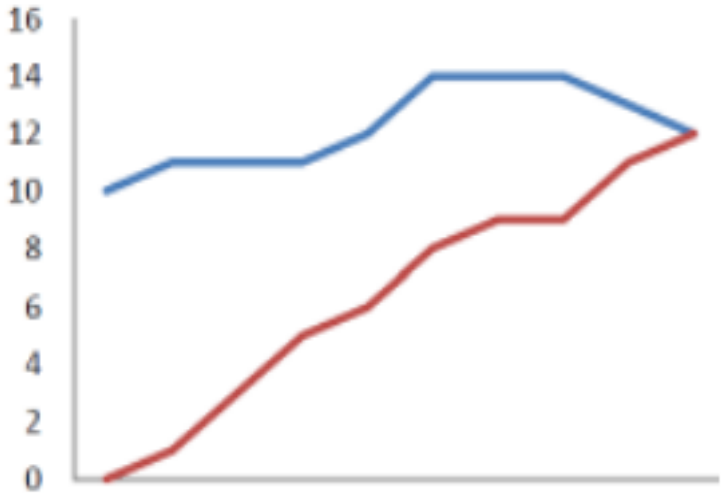
- important to predict when the release will be done
  - is team slower, or was more work added to the release?
- burndown chart:
  - the height of the line is the amount of work remaining
  - work added to the project is sometimes not shown
- burnup chart:
  - scope changes made explicitly visible
  - more work to do shown

# Scrum Burn Up

- Two charts for the same project
- Show scoop creep using Burn Up chart

Burn Down: show amount of work not done


Burn Up: show amount of work done



# Resources

- Agile Software Development with Scrum. Ken Schwaber and Mike Beedle
- Chaos Manifesto https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf
- Kanban: Successful Evolutionary Change for Your Tech Business. David J. Anderson
- Kanban https://www.atlassian.com/agile/kanban
- User Stories https://help.rallydev.com/writing-great-user-story
- User Stories Mike Cohen, https://vimeo.com/43601248

# Purpose of lecture

- Show the Group Scrum Assignment
  - More questions
- How to participate in an Agile process
  - Know the culture
- How to do Burn Down Charts
  - For monitoring and reporting