AGILE – SCRUM
Introduction

Agile & Scrum is like your mother-in-law,
*it points out **all** your faults!*

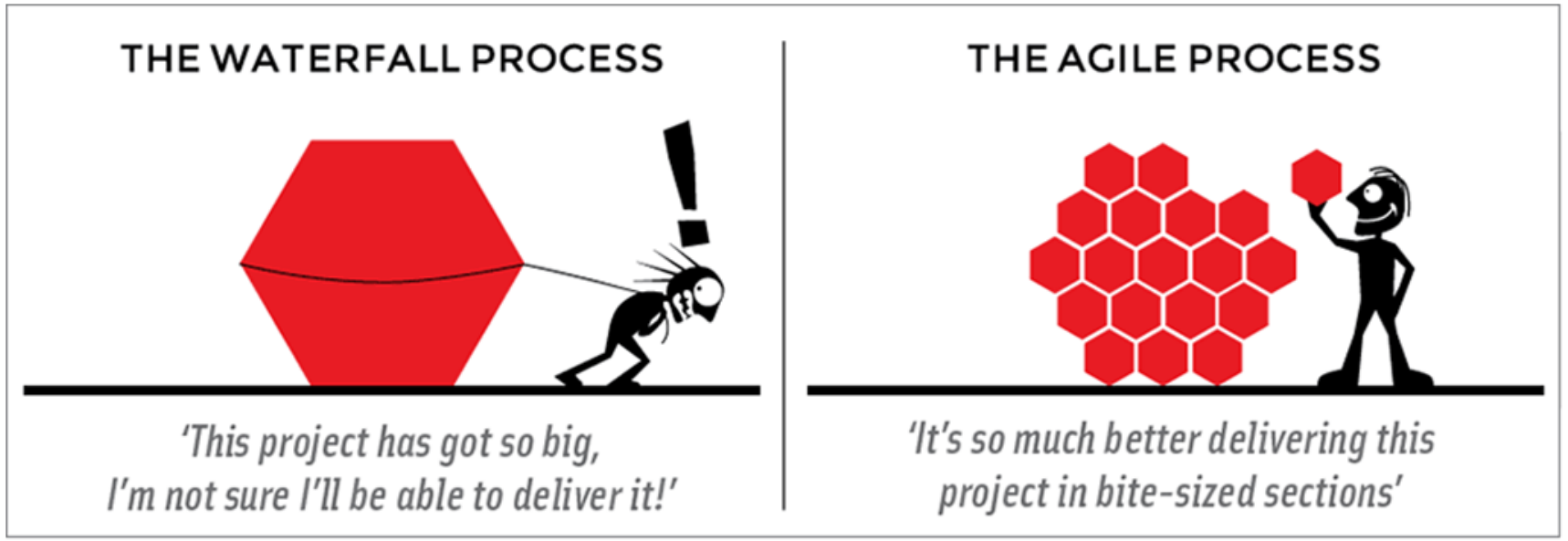
TOPICS

1. Introduction
2. Waterfall vs. Agile
3. Agile Methodologies
   - XP
   - TDD / FDD
   - Scrum
   - Disciplined Agile Delivery
4. Agile Values and Principles
5. What is Scrum?
6. History of Scrum
7. Functionality of Scrum
8. Minimum Viable Product
9. Kanban
10. Scrum Framework
    - Roles
    - Ceremonies
    - Artifacts
11. Sprint
12. Velocity
13. Burndown Chart
14. Estimation Techniques
    - Planning Poker
    - T-Shirt Sizes
15. Pros & Cons
16. Next Steps
    - SAFe
    - DevOps

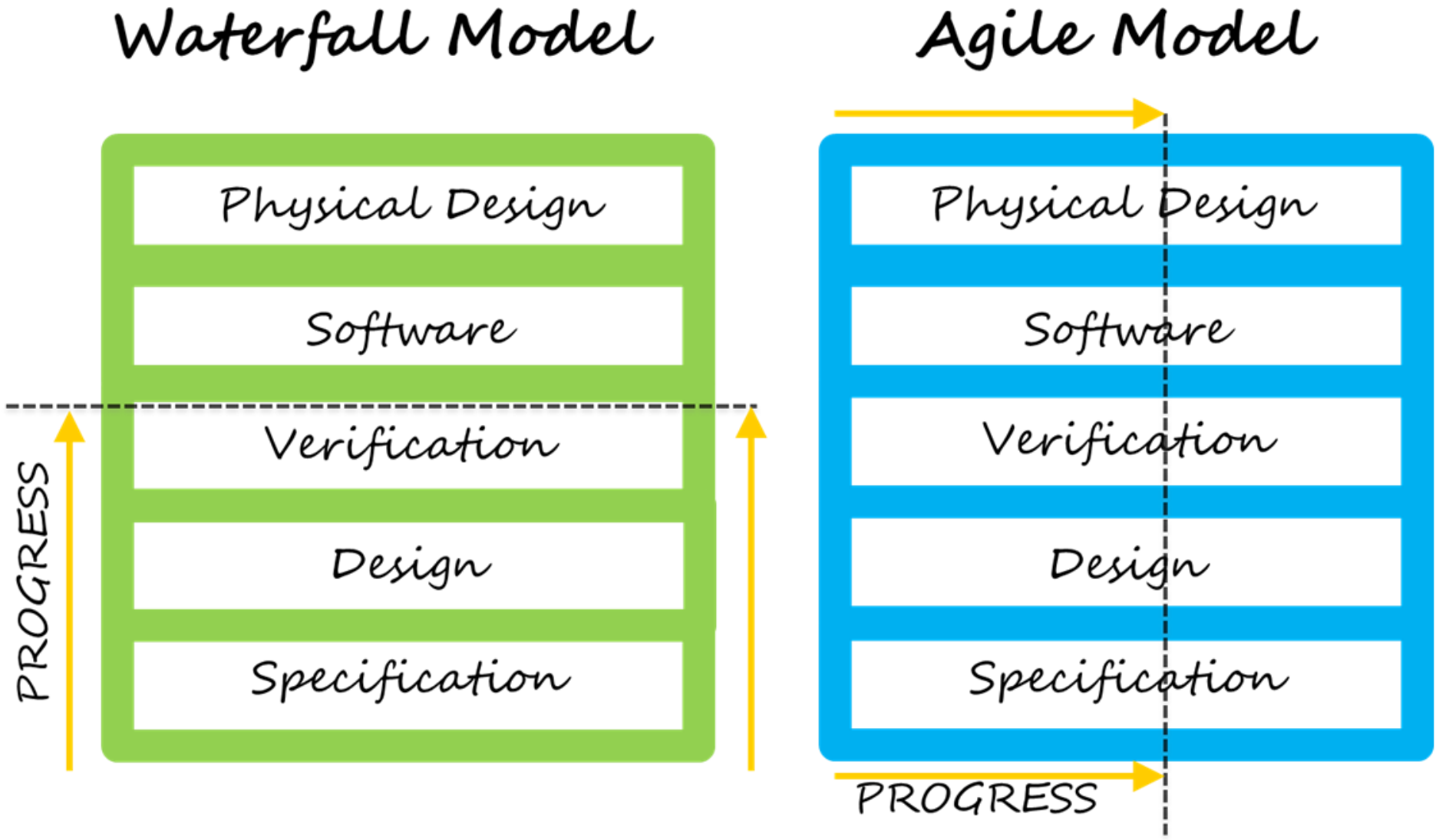
INTRODUCTION

Classical methods of software development have many disadvantages:

- huge effort during the planning phase
- poor requirements conversion in a rapid changing environment
- treatment of staff as a factor of production



WATERFALL VS. AGILE



AGILE

Introduction and Overview

THE AGILE MANIFESTO

In February 2001, 17 software developers met at the Snowbird, Utah resort, to discuss lightweight development methods. They published the *Manifesto for Agile Software Development* to define the approach now known as agile software development.

Some of the manifesto's authors formed the Agile Alliance, a non-profit organization that promotes software development according to the manifesto's values and principles.

AGILE VALUES

The Agile Manifesto reads:

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.

AGILE VALUES

AGILE PRINCIPLES

The Agile Manifesto is based on 12 Principles:

1. Customer satisfaction with rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity – the art of maximizing the amount of work not done – is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

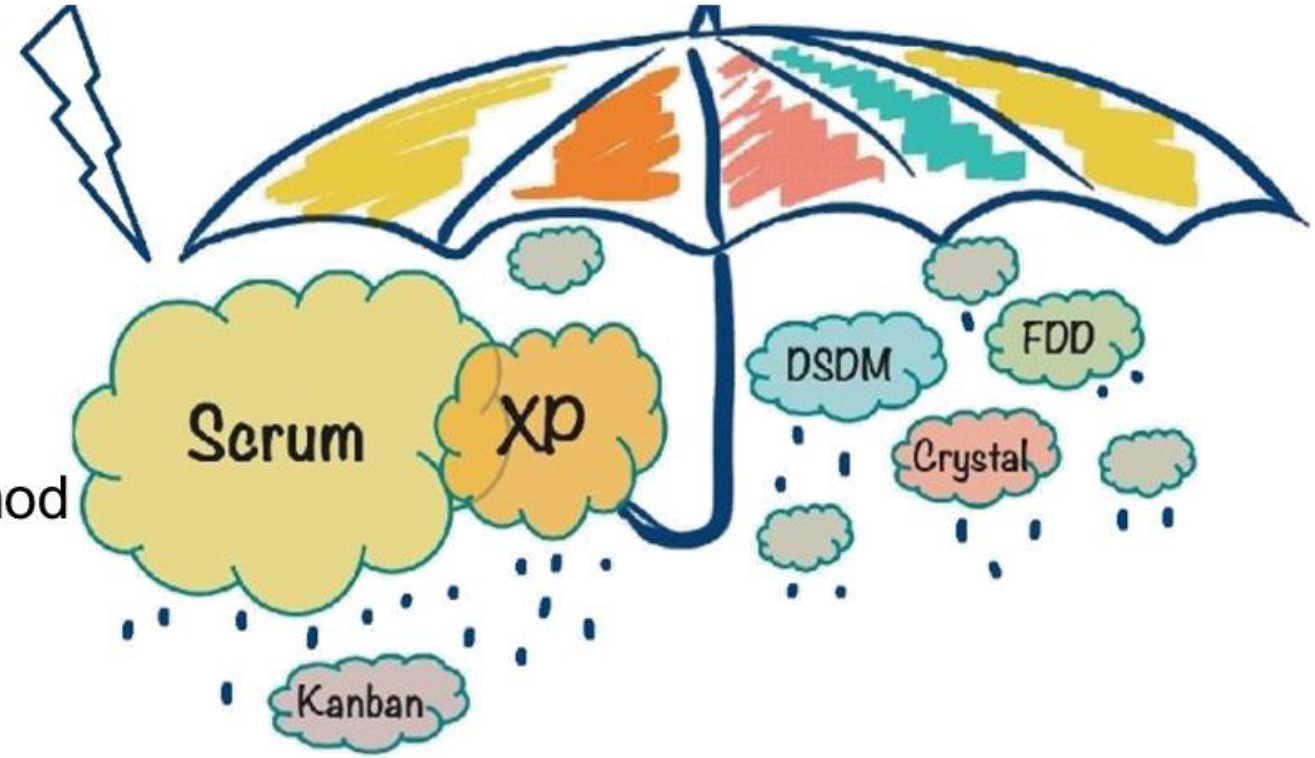
AGILE SOFTWARE DEVELOPMENT

Characteristics of Agile Software Development

- Light Weighted methodology
- Small to medium sized teams
- Vague and/or changing requirements
- Vague and/or changing techniques
- Simple design
- Minimal system into production

AGILE METHODOLOGIES

- Extreme Programing (XP)
- Test Driven Development (TDD)
- Feature Driven Development (FDD)
- Scrum
- Disciplined Agile Delivery (DAD)
- Dynamic Systems Development Method (DSDM)



SCRUM

Introduction and Overview

Scrum in 100 words

- Scrum is an agile methodology 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 the real working software and decide to release as it is or continue to enhance it for another sprint.

SCRUM?

Definition from rugby football:

a scrum is a way to restart the game after an interruption, where the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them

SCRUM - AN AGILE PROCESS

SCRUM is an agile, lightweight process for managing and controlling software and product development in rapidly changing environments.

- Iterative, incremental process
- Team-based approach
- Developing systems / products with rapidly changing requirements
- Controls the chaos of conflicting interest and needs
- Improve communication and maximize cooperation
- Protecting the team form disruptions and impediments
- A way to maximize productivity

HISTORY OF SCRUM

1995:

- Analysis of common software development processes → not suitable for empirical, unpredictable and non-repeatable processes
- Design of a new method: Scrum by Jeff Sutherland & Ken Schwaber
- Enhancement of Scrum by Mike Beedle & combination of Scrum with Extreme Programming

1996:

Introduction of Scrum at OOPSLA conference

2001:

Publication “Agile Software Development with Scrum” by Ken Schwaber & Mike Beedle

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
- 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
- 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

SCRUM FRAMEWORK

Roles

- Product Owner
- Scrum Master
- Team

Ceremonies

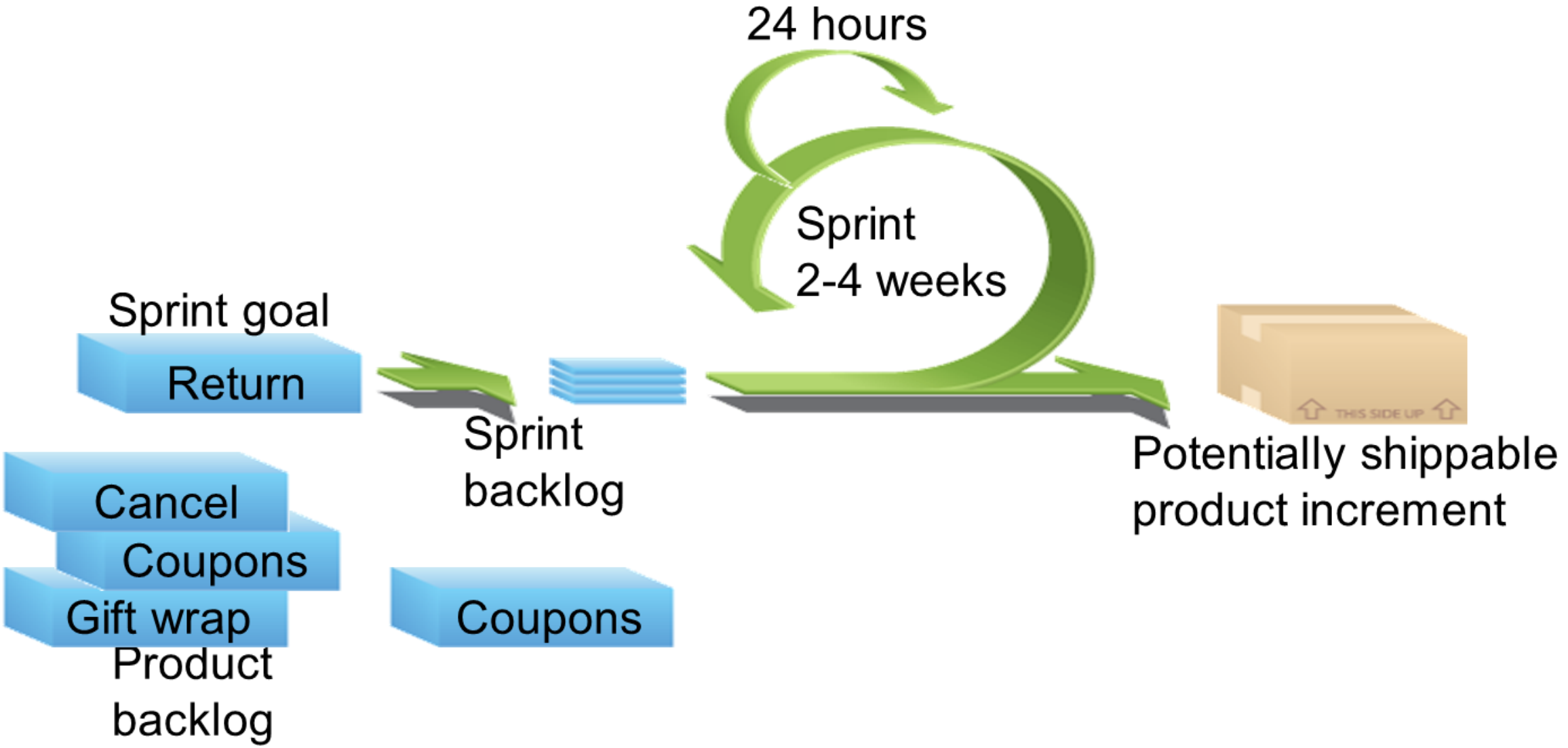
- Sprint Planning
- Sprint Review
- Sprint Retrospective
- Daily Scrum Meeting

Artifacts

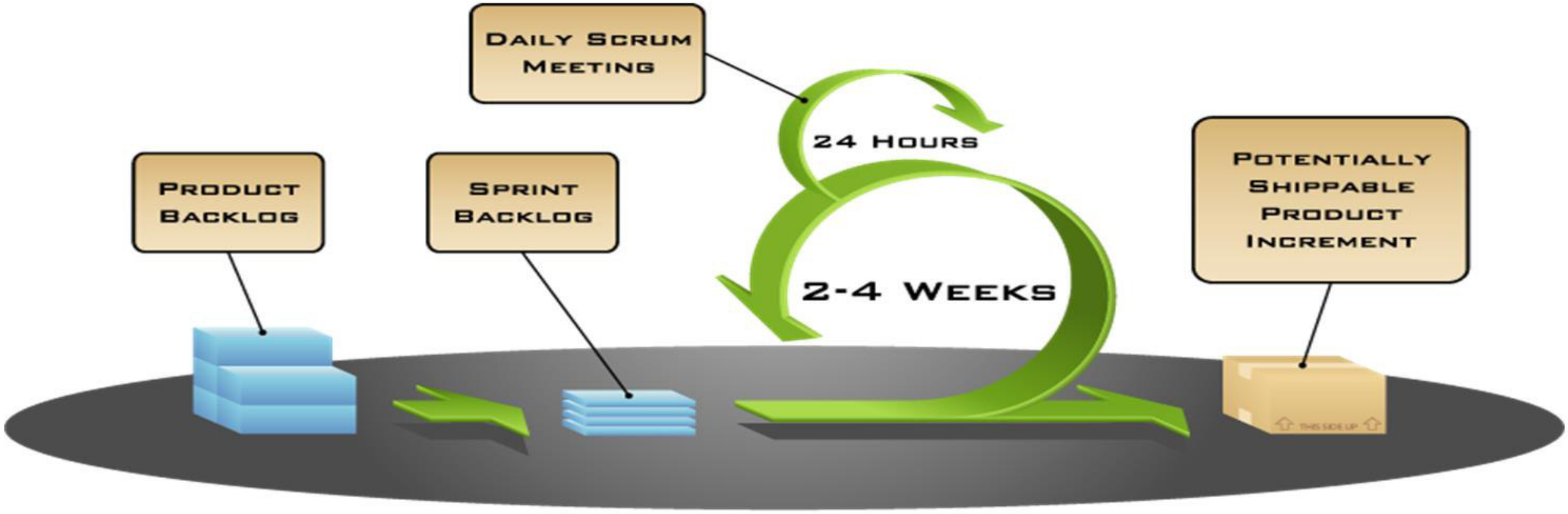
- Product Backlog
- Sprint Backlog
- Burndown Charts

Tip! https://www.youtube.com/watch?v=9TycLR0TqFA

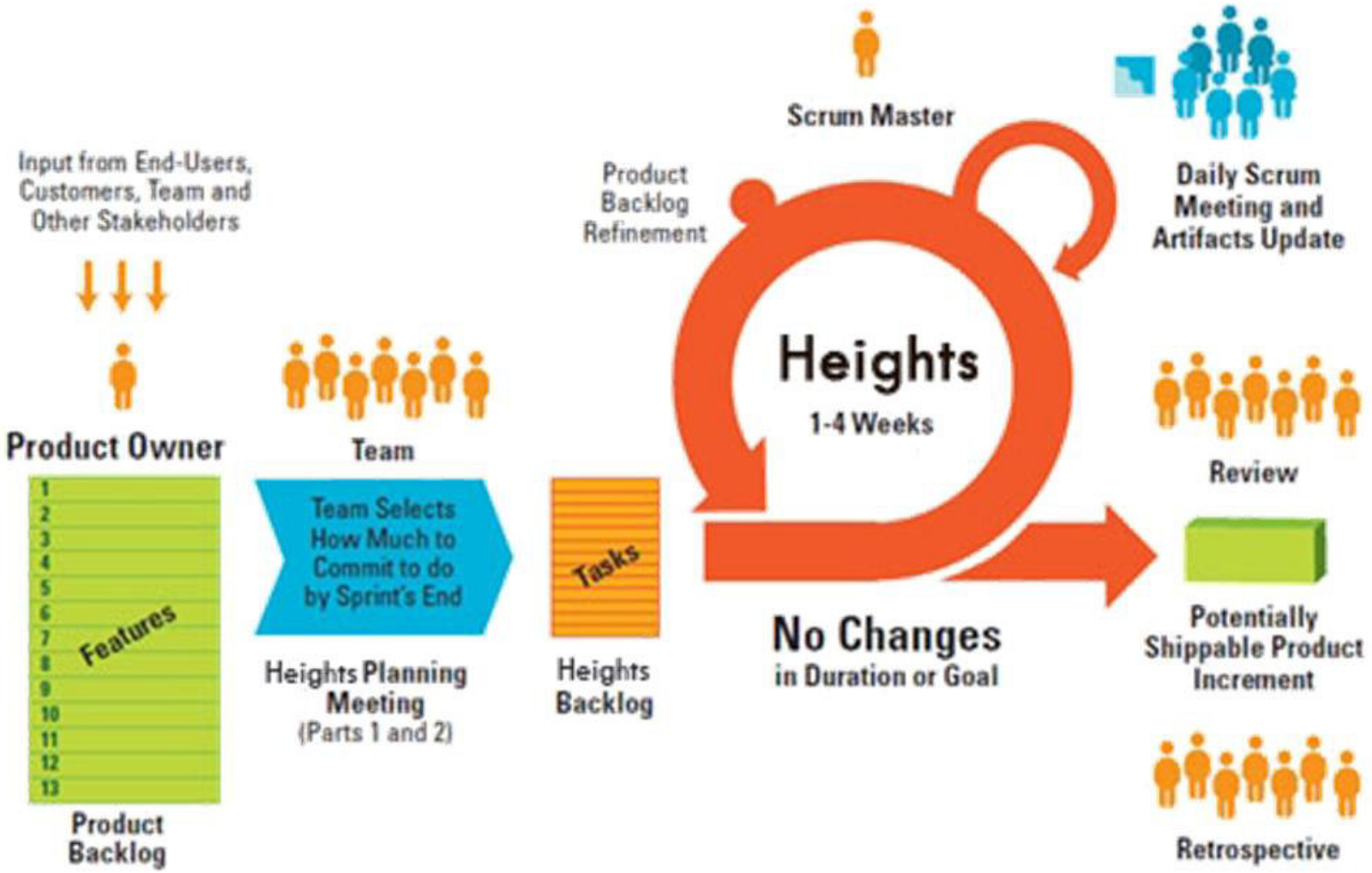
FUNCTIONALITY OF SCRUM



PUTTING IT ALL TOGETHER



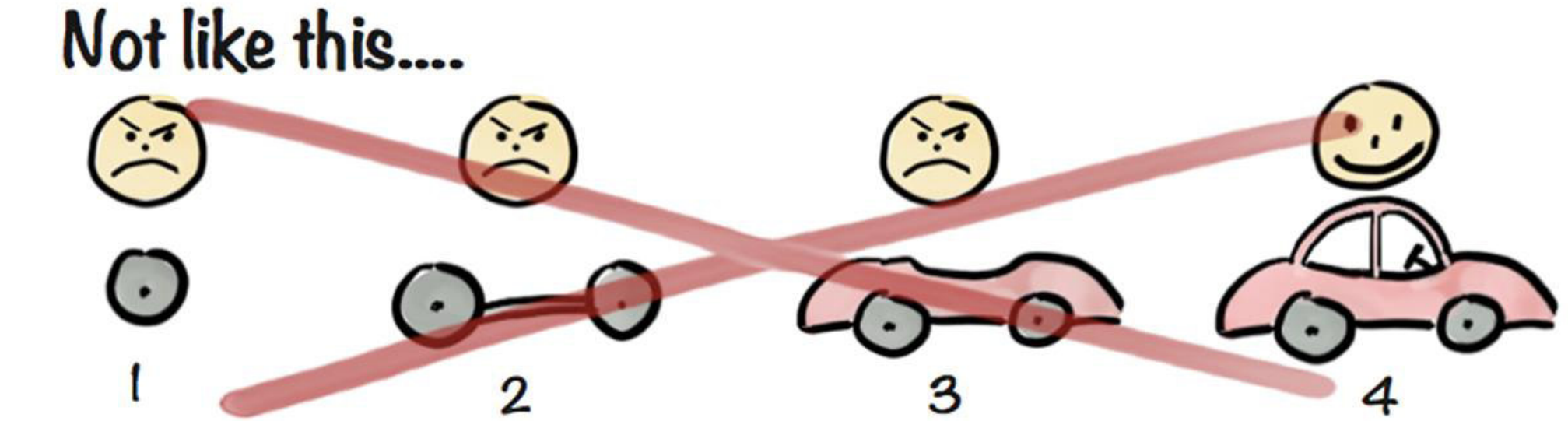
PUTTING IT ALL TOGETHER

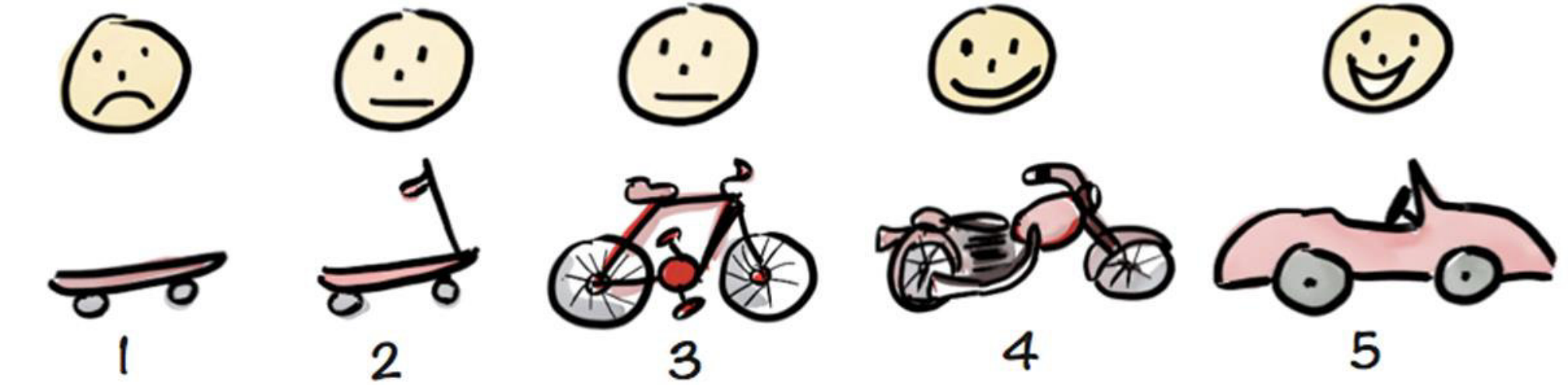


SCRUM TERMINOLOGY

- MVP
- Scrum Team
- Sprint
- Product Owner
- Scrum Master
- Product Backlog
- Sprint Backlog

MINIMUM VIABLE PRODUCT

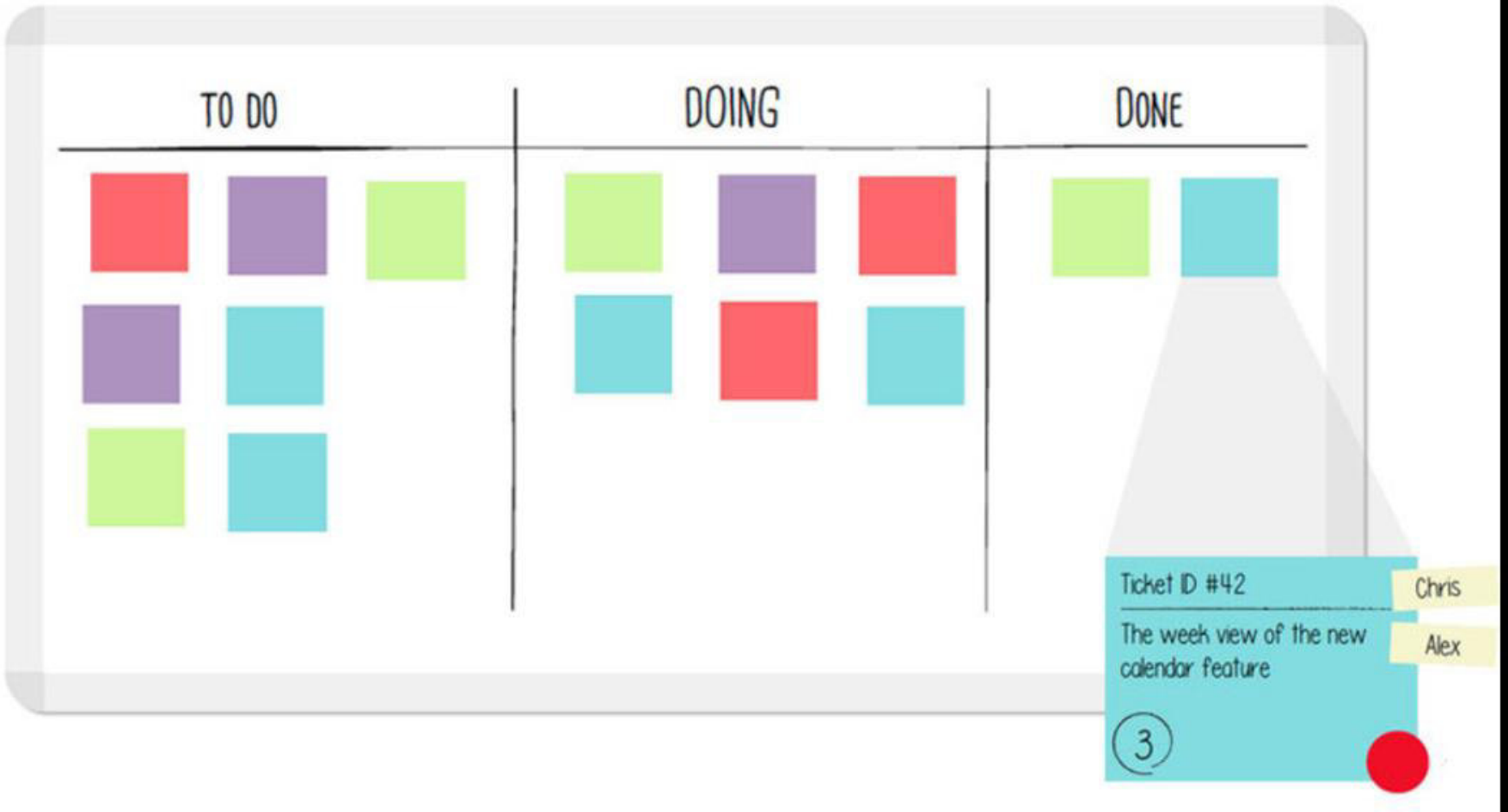




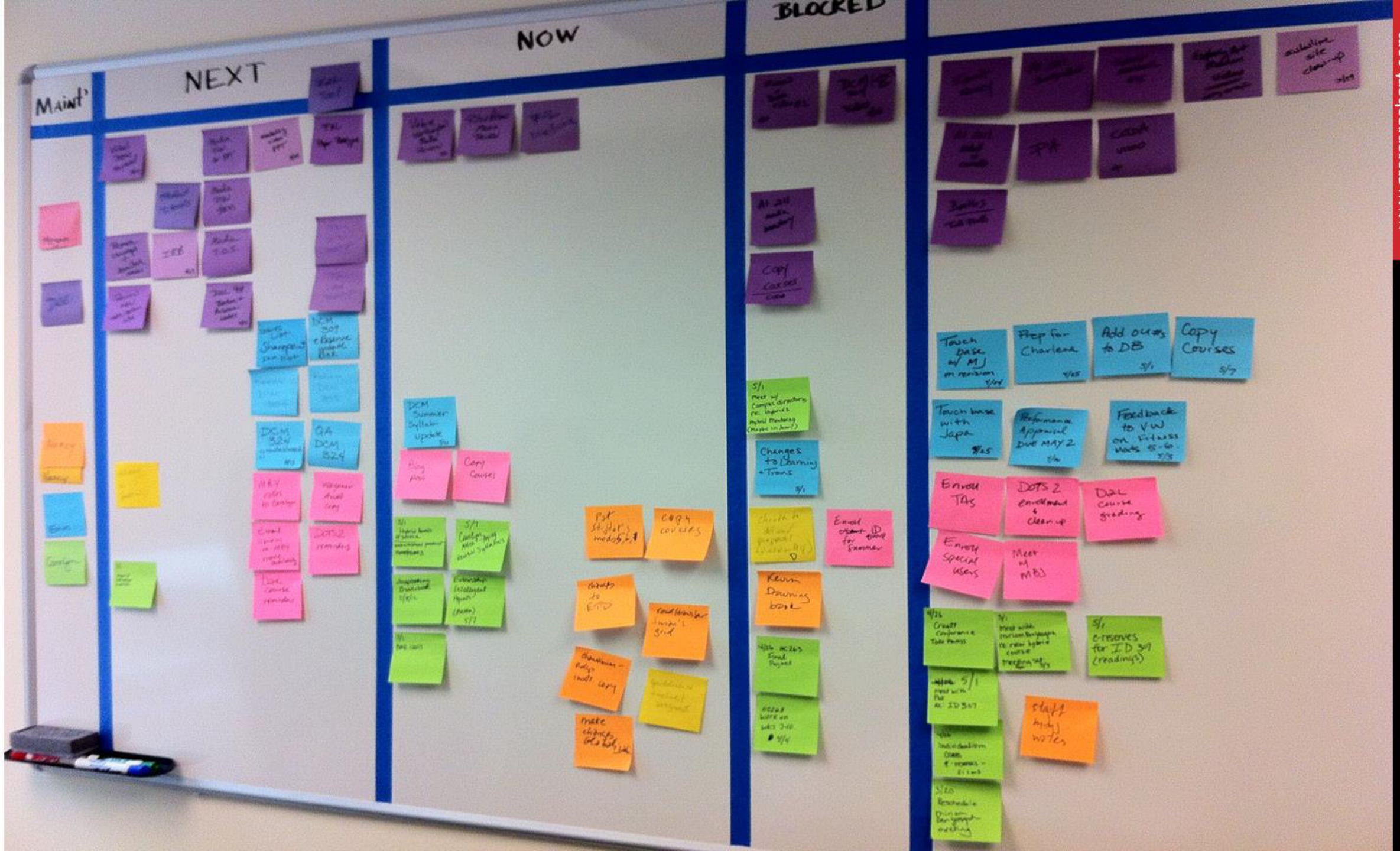
by Henrik Kniberg

KANBAN / VISUAL SIGNS

- Visualize Work
- Limit Work in Process
- Definition of Done
- Visualize Impediments
- Continuous Improvement



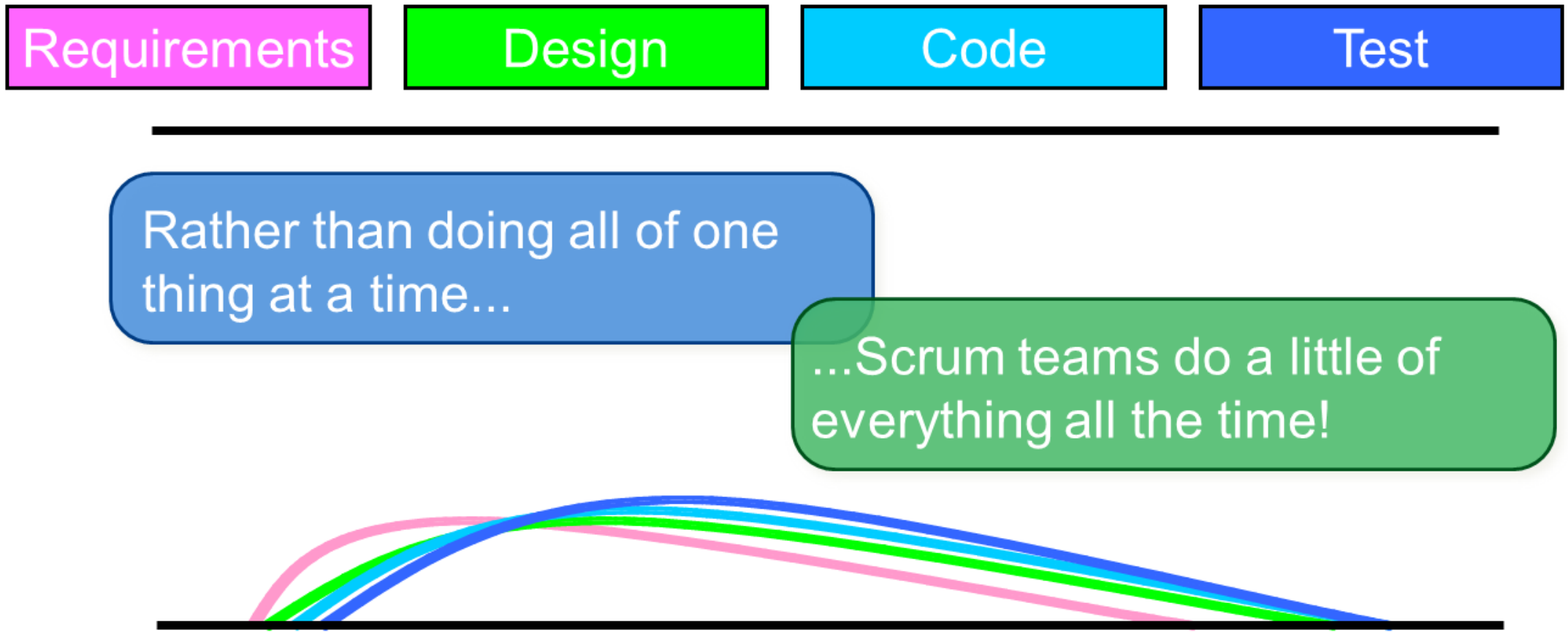
KANBAN - EXAMPLE



SPRINTS

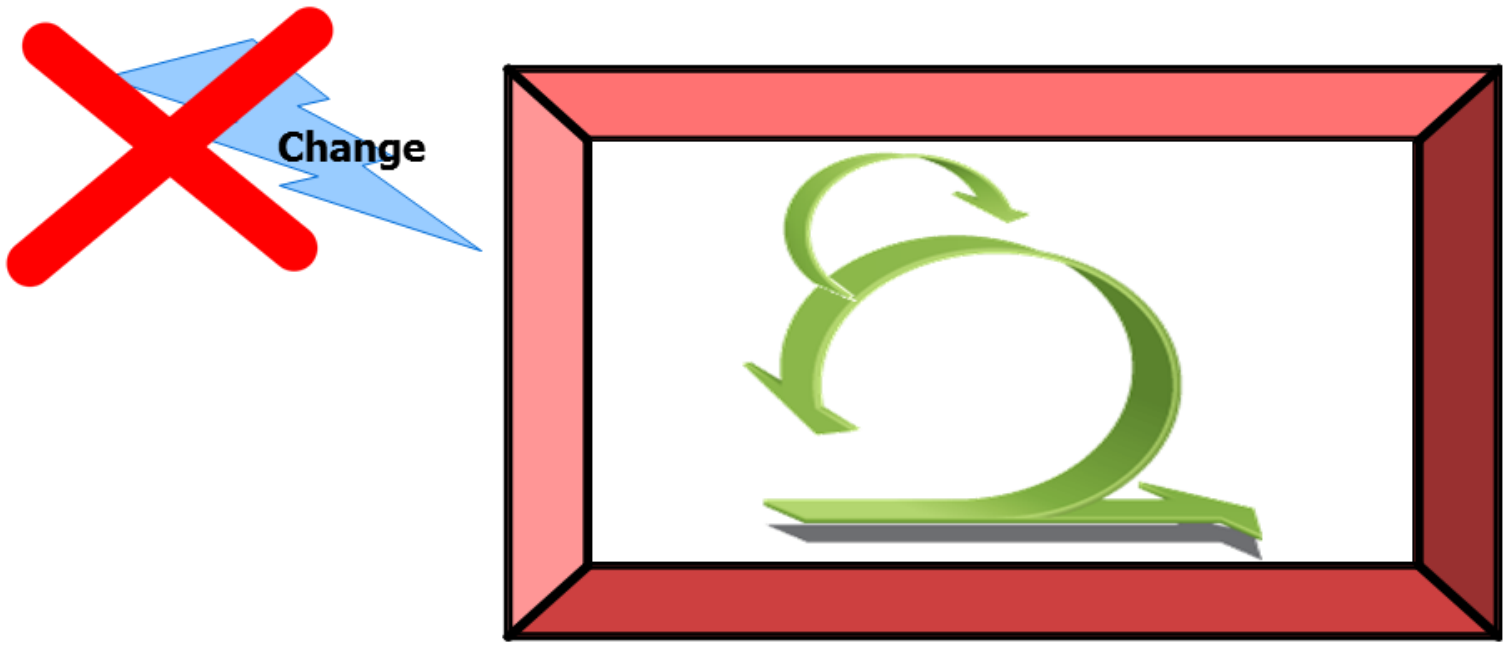
- Scrum projects make progress in a series of “sprints”
  - Analogous to Extreme Programming iterations
- Typical duration is 2 – 4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

SEQUENTIAL VS. OVERLAPPING



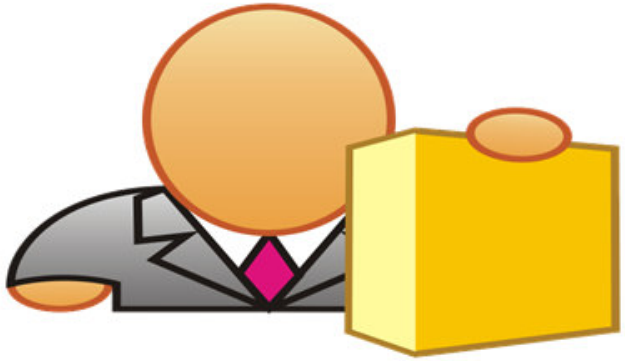
Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

NO CHANGES DURING A SPRINT



Tip! Plan sprint durations around how long you can commit to keeping change out of the sprint.

PRODUCT OWNER



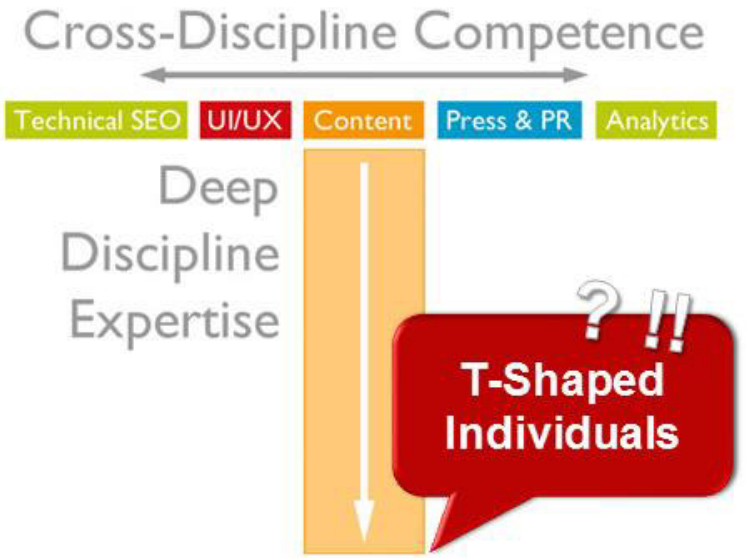
- Defines the features of the product
- Decides on release date and content
- Be responsible for the success & profitability of the product (ROI)
- Prioritizes features according to the market value
- Adjusts features and priority every iteration, as needed
- Accepts or rejects work results

THE SCRUM MASTER

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

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



SPRINT PLANNING MEETING

- A collaborative meeting in the beginning of each Sprint between the Product Owner, the Scrum Master and the Team
- Takes 8 hours and consists of 2 parts (“before lunch and after lunch”)
- 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 Scrum Master
- High-level design is considered

PARTS OF SPRINT PLANNING MEETING

1st Part:

- Reviewing Product Backlog
- Determining the Sprint Goal
- Participants: Product Owner, Scrum Master, Scrum Team

2nd Part:

- Creating Sprint Backlog
- Participants: Scrum Master, Scrum Team

DAILY SCRUM

- Is a short (15 minutes long) meeting, which is held every day before the Team starts working
- Participants: Scrum Master (which is the chair), Scrum Team
- “Chickens” and “Pigs”
- Every Team member should answer on 3 questions

EVERYONE ANSWERS 3 QUESTIONS

1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

Tip! These are *not* status for the Scrum Master, these are commitments in front of peers.

DAILY SCRUM

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team

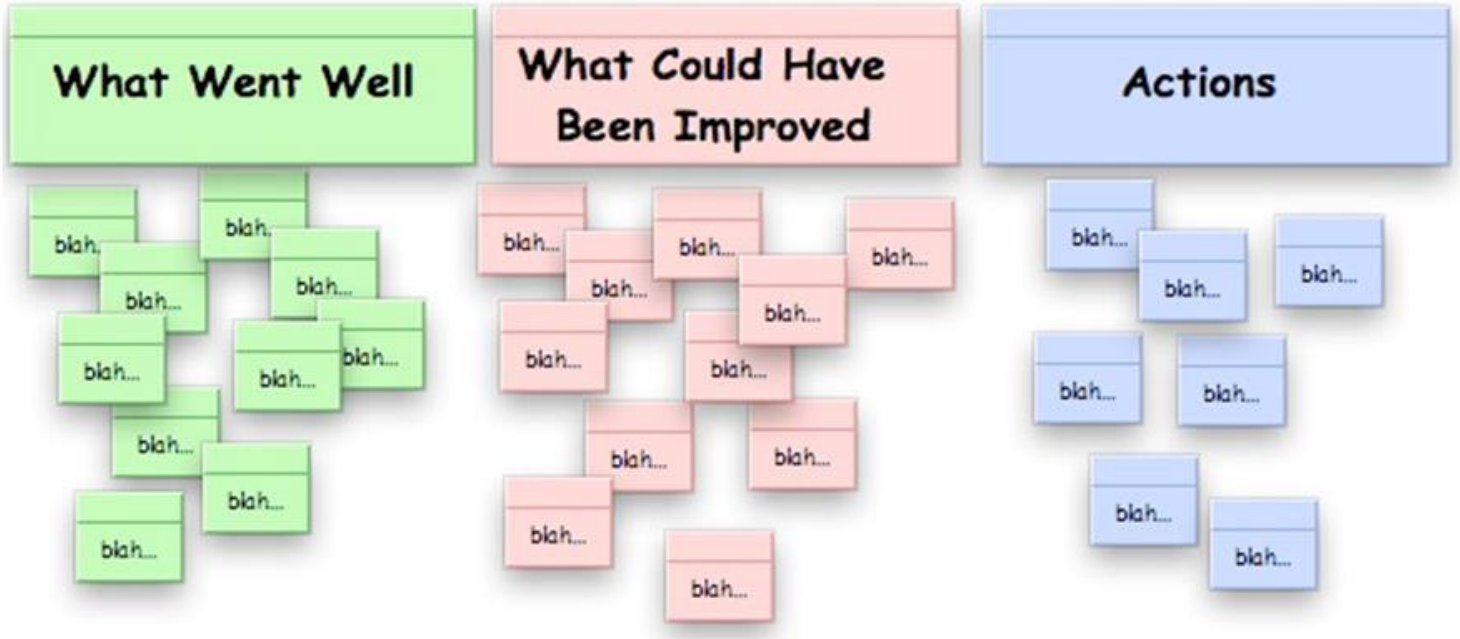
DAILY SCRUM

THE SPRINT REVIEW

- Is held at the end of each Sprint
- Business functionality which was created during the Sprint is demonstrated to the Product Owner
- Informal, should not distract Team members of doing their work
- The team presents what it accomplished during the sprint
  - Typically takes the form of a demo of new features or underlying architecture
  - Informal
  - 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



Tip! Retrospectives can be conducted as “Mute” Meetings.

START / STOP / CONTINUE

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

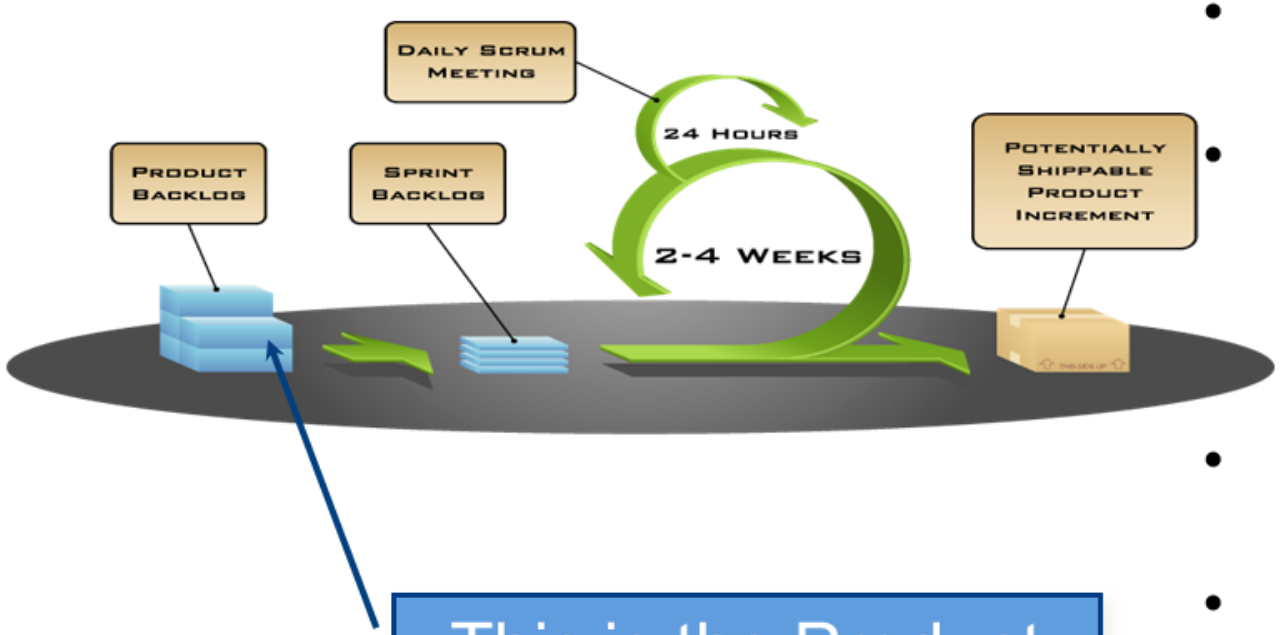
Start doing

Stop doing

Continue doing

Point out!
Vote out!

ARTIFACT - 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

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

THE SPRINT GOAL

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

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.

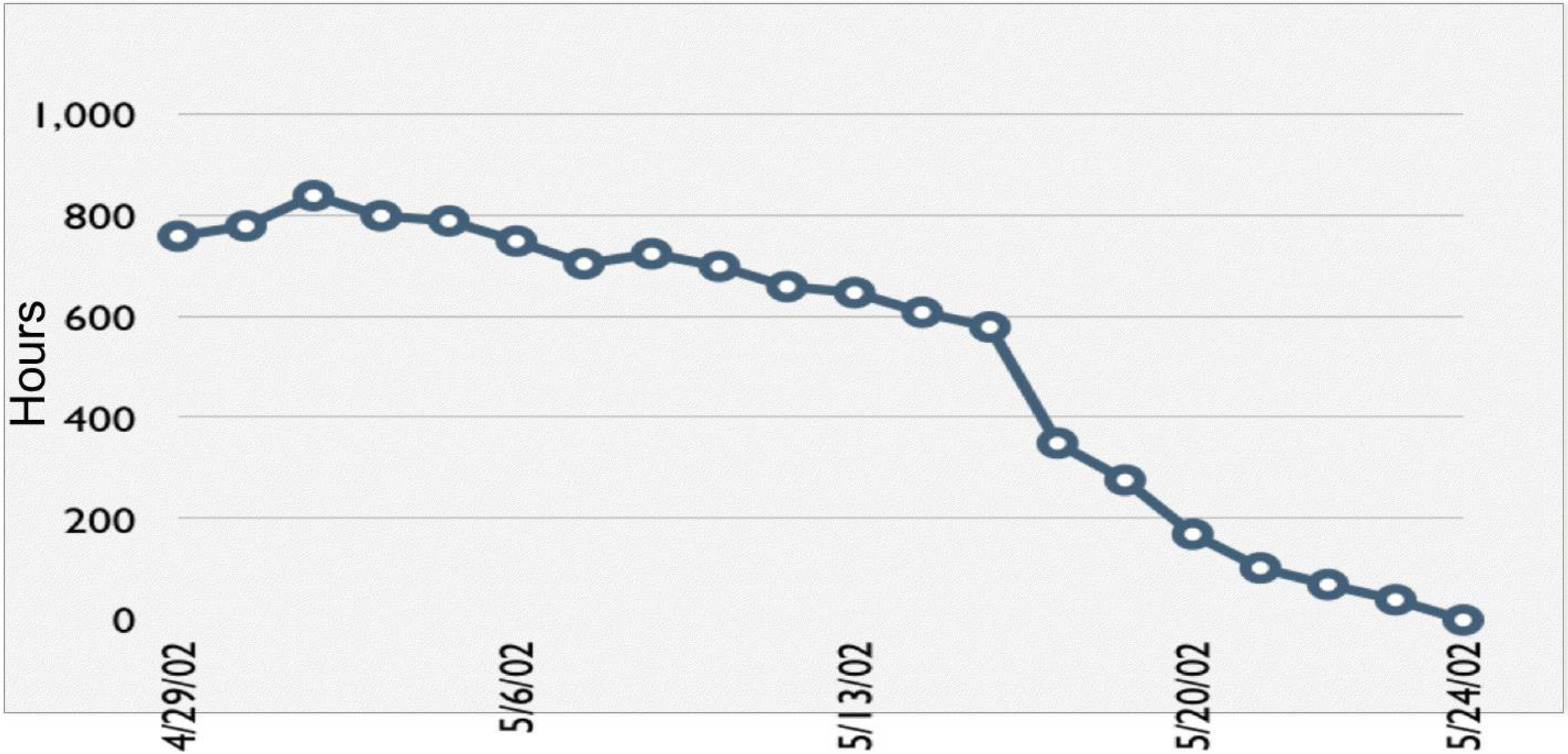
ARTIFACT - A SPRINT BACKLOG

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

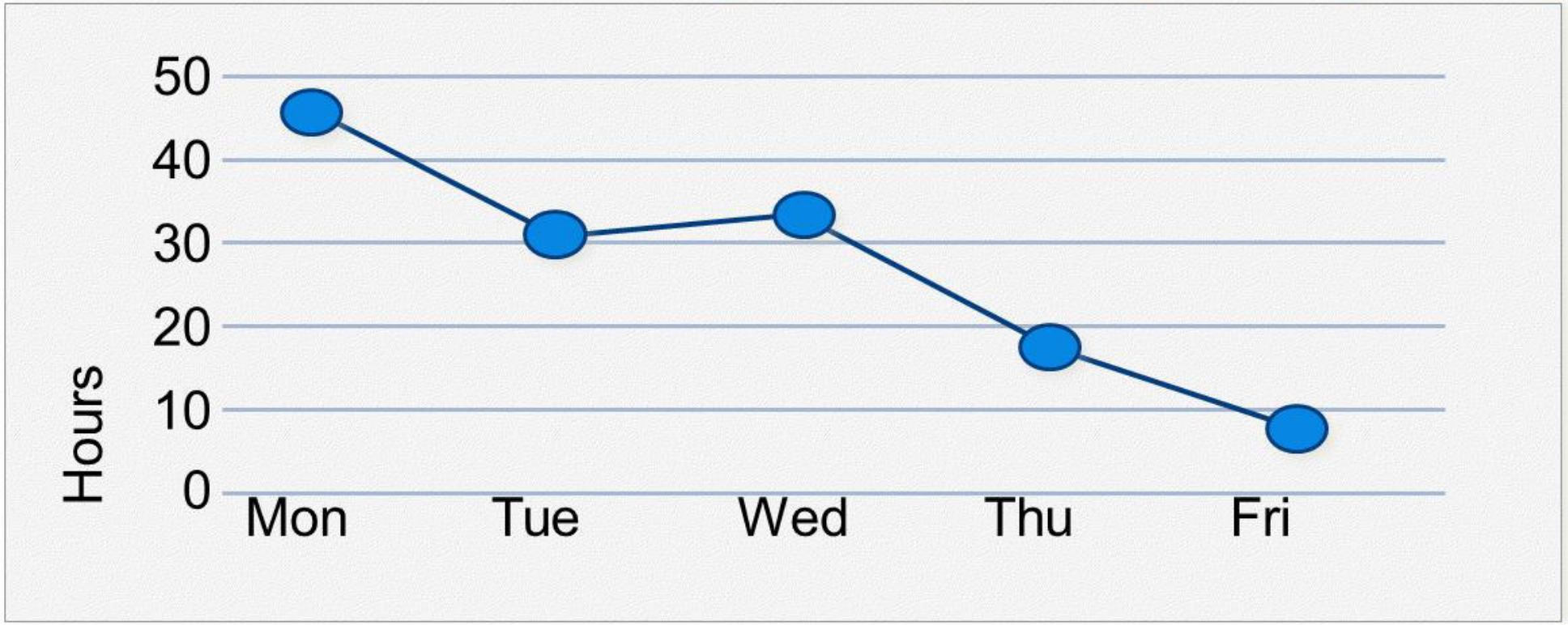
MANAGING THE SPRINT BACKLOG

- Any team member can work on the sprint backlog
- 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
- Individuals sign up for work of their own choosing
  - Work is never assigned!
- Estimated work remaining is updated daily

ARTIFACT - A SPRINT BURN DOWN CHART



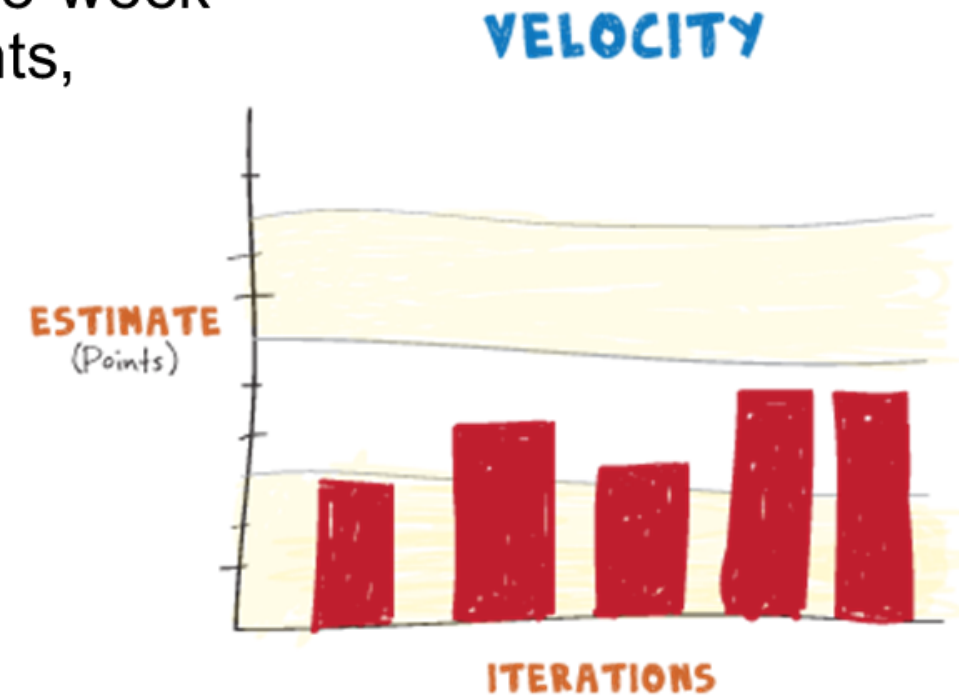
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



VELOCITY

- Velocity is a metric that predicts how much work team can successfully complete within a sprint

- If the team completed 10 stories during a two-week sprint and each story was worth 3 story points, then the team's velocity is 30 story points per sprint.



PLANNING POKER

- In the Sprint Zero as a part of release planning, the Scrum team comes up with effort estimation for all the stories in the release
- Planning Poker is a popular method of effort estimation which ensures that the entire team to be involved in the estimation exercise

PROS & CONS

PROS

- Completely developed and tested features in short iterations
- Simplicity of the process
- Clearly defined rules
- Increasing productivity
- Self-organizing
- Each team member carries a lot of responsibility
- Improved communication

CONS

- “Undisciplined hacking” (no written documentation)
- Violation of responsibility
- Current mainly carried by the inventors

NEXT STEPS?

- Read Agile Scrum Guide – Know it better!
- Use Agile methods in your day to day work or life!
- Seek industry known certifications like CSM / ACP!
- Understand SAFe & DevOps!

THANK YOU!

Prashant Arora

www.aroraprashant.com | ✆ +91 855-0990-933| ✉ pprashan@gmail.com

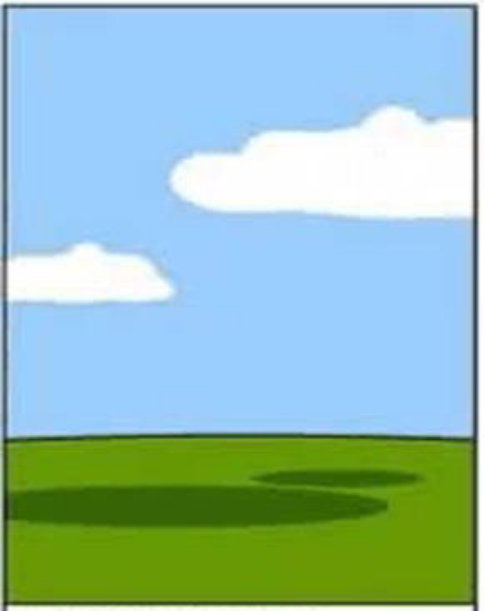
How Client Explained

How Business Analyst Understood it

How Architectes Designed it

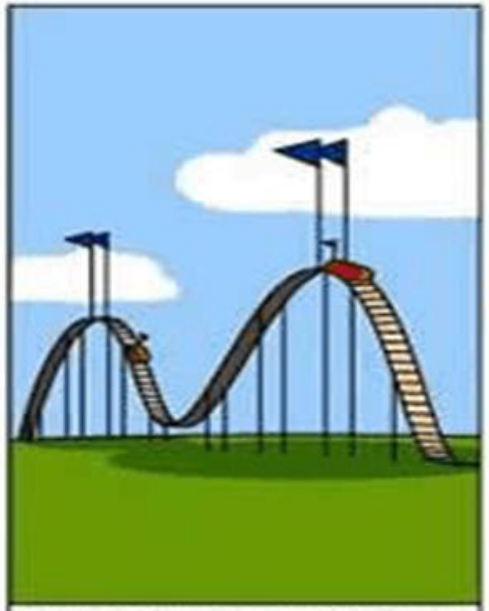
How Programmer Wrote It

How Business Consultant described it

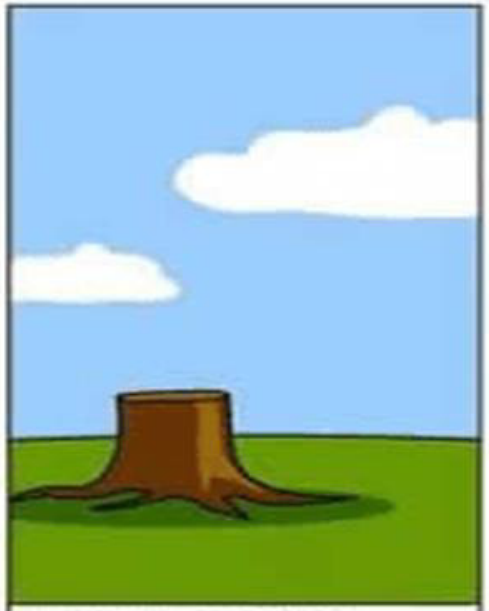


How the Project was Documented

What Operations installed



How the Client was billed



How it was supported

What the Client really needed!