UNIVERSITY OF RICHMOND

Software Development Life Cycle

CMSC 240 Software Systems Development

Today

- Software Development Life Cycle (SDLC)
- Waterfall Method
- Agile
  - Scrum
  - Kanban

Today

- Software Development Life Cycle (SDLC)

- Waterfall Method

- Agile
  - Scrum
  - Kanban

Software Development Life Cycle (SDLC)

- Purpose
  - Create good software
  - Reduce risk
  - Enable visibility and measurement
  - Enable teamwork

- Key Attributes
  - Outcomes/results of the life cycle are key deliverables or products
  - Clear roles
  - Pre and post conditions are understood and upheld

Key Elements to Any SDLC

1. Feasibility

2. Requirements Specification
3. Architecture and Design
4. Development
5. Validation

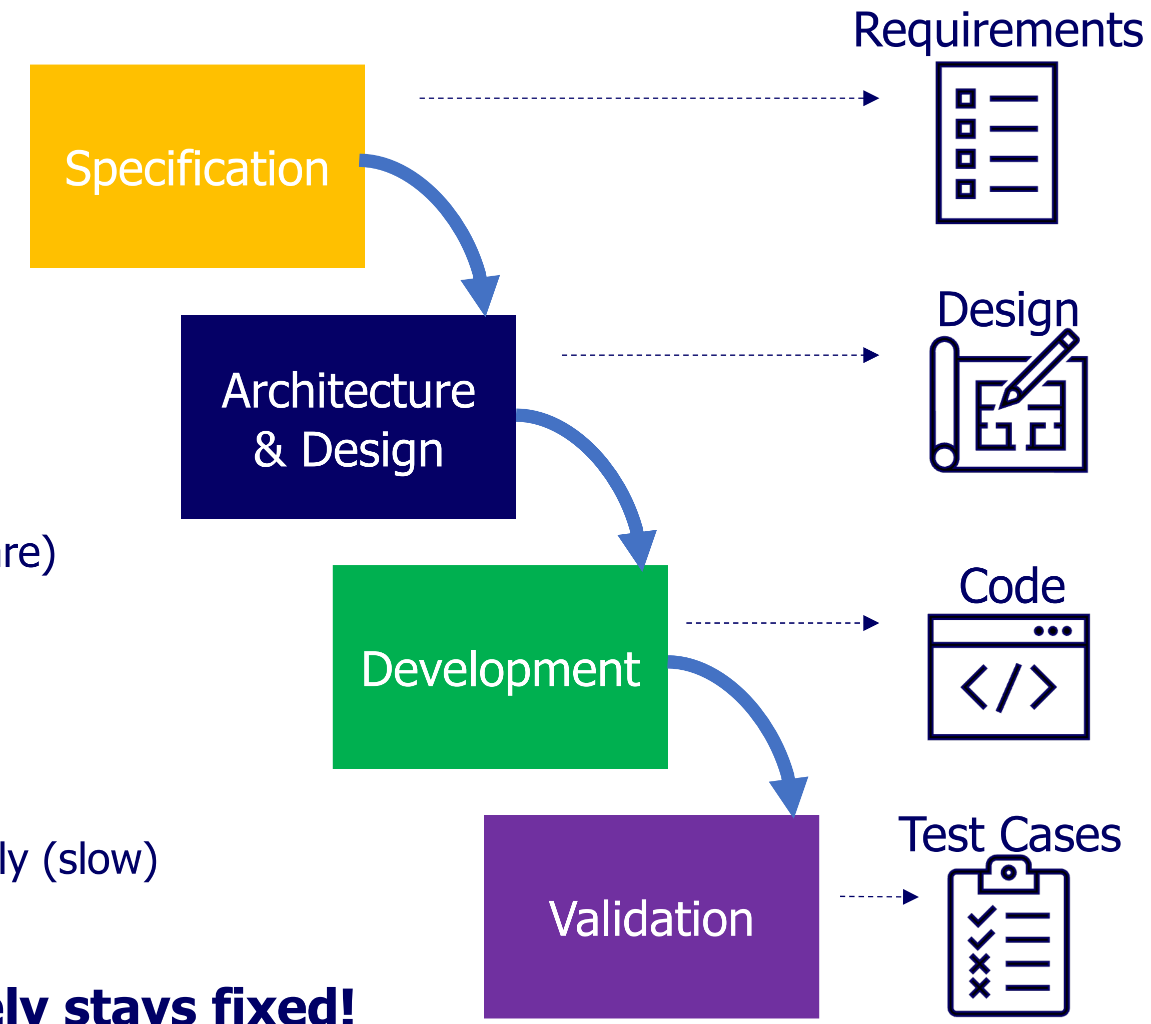
6. Evolution/Maintenance

Today

- ~~Software Development Life Cycle (SDLC)~~
- Waterfall Method
- Agile
  - Scrum
  - Kanban

Waterfall Model

- Sequential Process Phases
    - One step completes before next one starts
- Rational Process
    - Enables careful planning
        - This is exactly how construction is done
    - Good for
        - systems that cannot be easily changed (hardware)
        - when exhaustive testing is required
- Challenges
    - Heavyweight process
        - Process if followed systematically and completely (slow)
        - Specification is a negotiation process
        - Specifications precede the system
    - **World is rarely known upfront and rarely stays fixed!**
        - Hard to adapt to upstream changes once a step completes



Example - Functional Requirements

1. The system shall allow users to register and create personal accounts.

2. The system shall process credit card payments.

3. The software must integrate with the existing inventory management system.

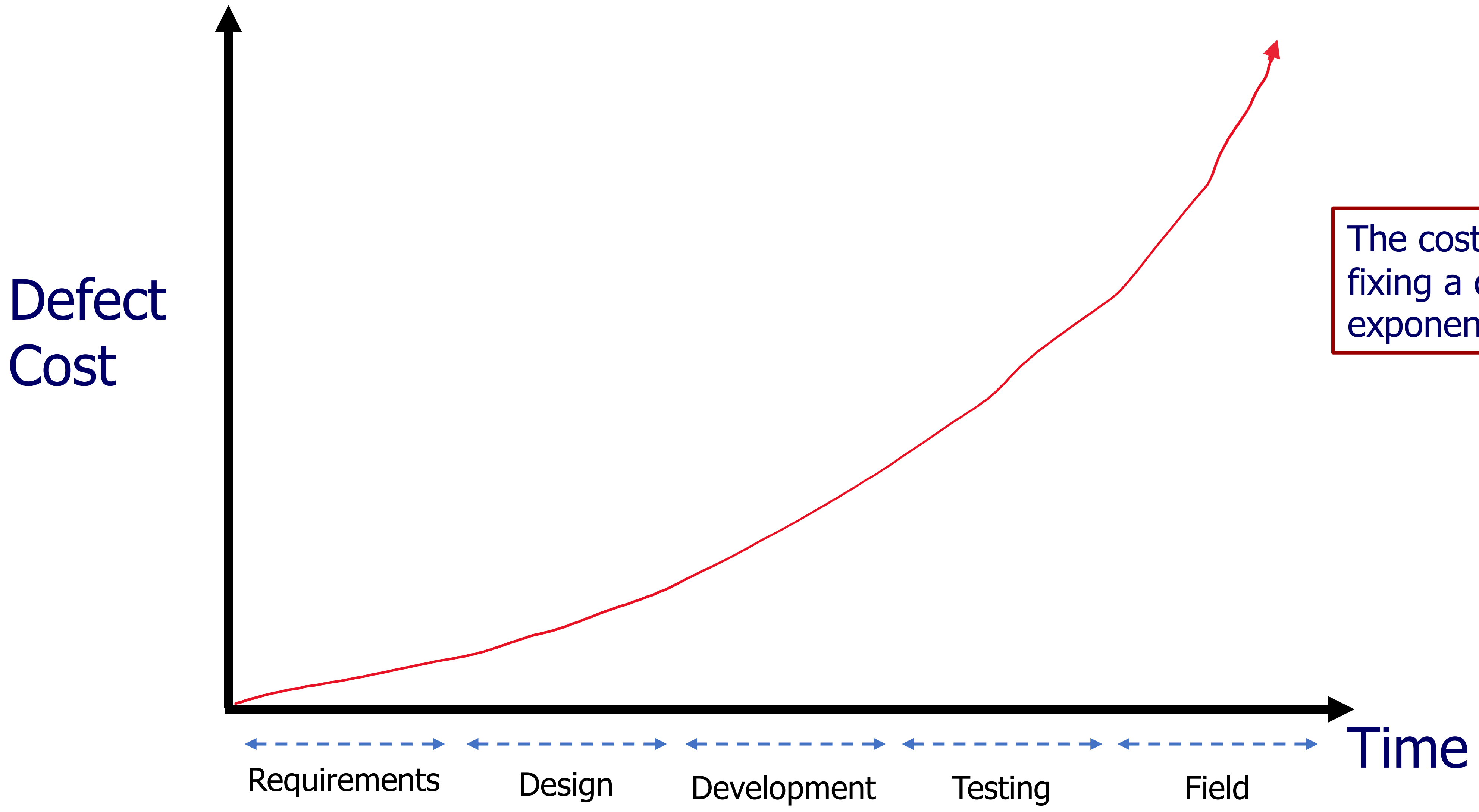
Waterfall Model

- Real projects rarely follow a sequential flow
- Hard to state all requirements explicitly
- No maintenance or evolution involved
- Customer must have patience
- Mistakes can be disastrous

Errors are most frequent during <u>requirements</u> and <u>design</u> activities

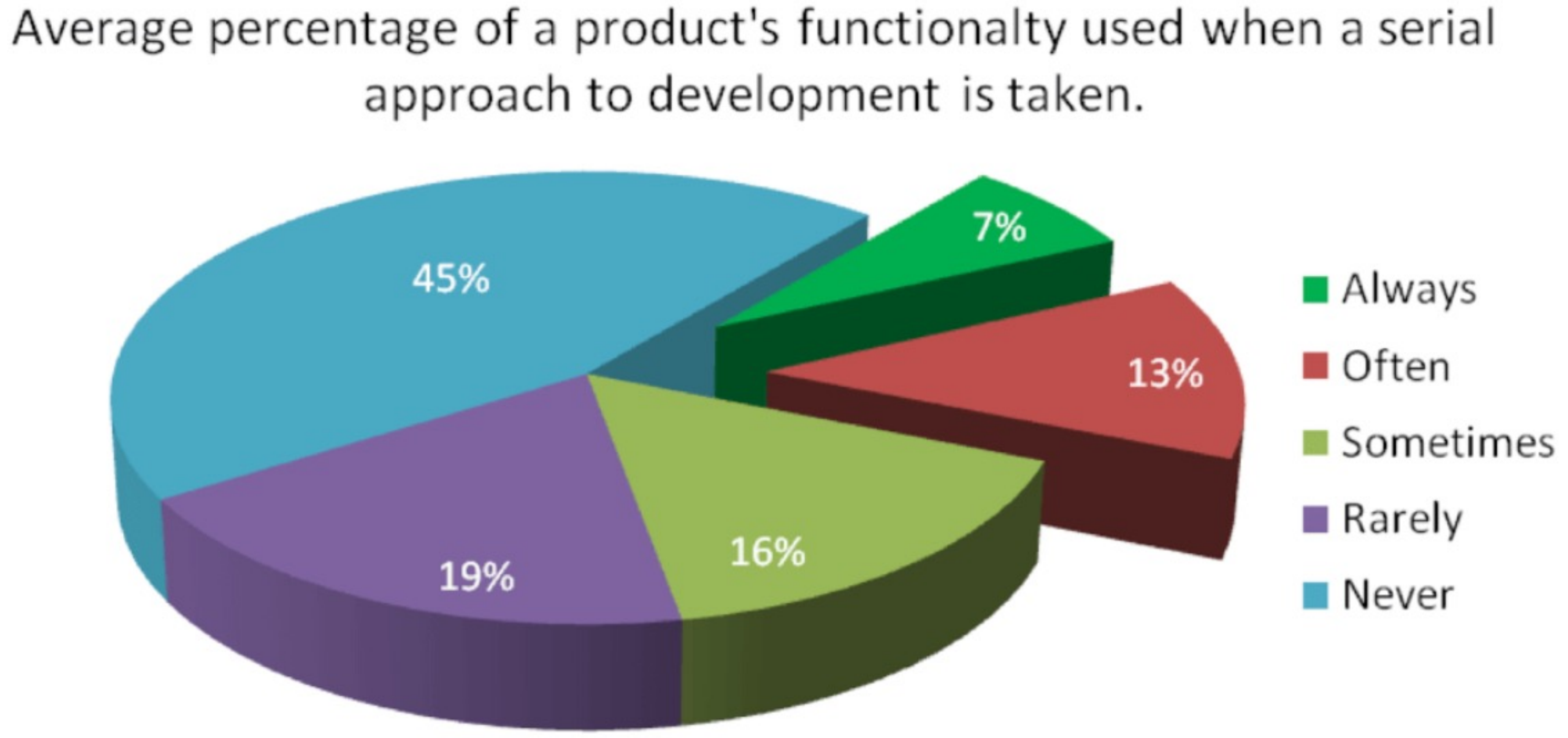
and are **more expensive** the later they are removed.

What this means in practice



The cost of finding and fixing a defect grows exponentially with time

What this means in practice



Chaos Report v3, Standish Group

Today

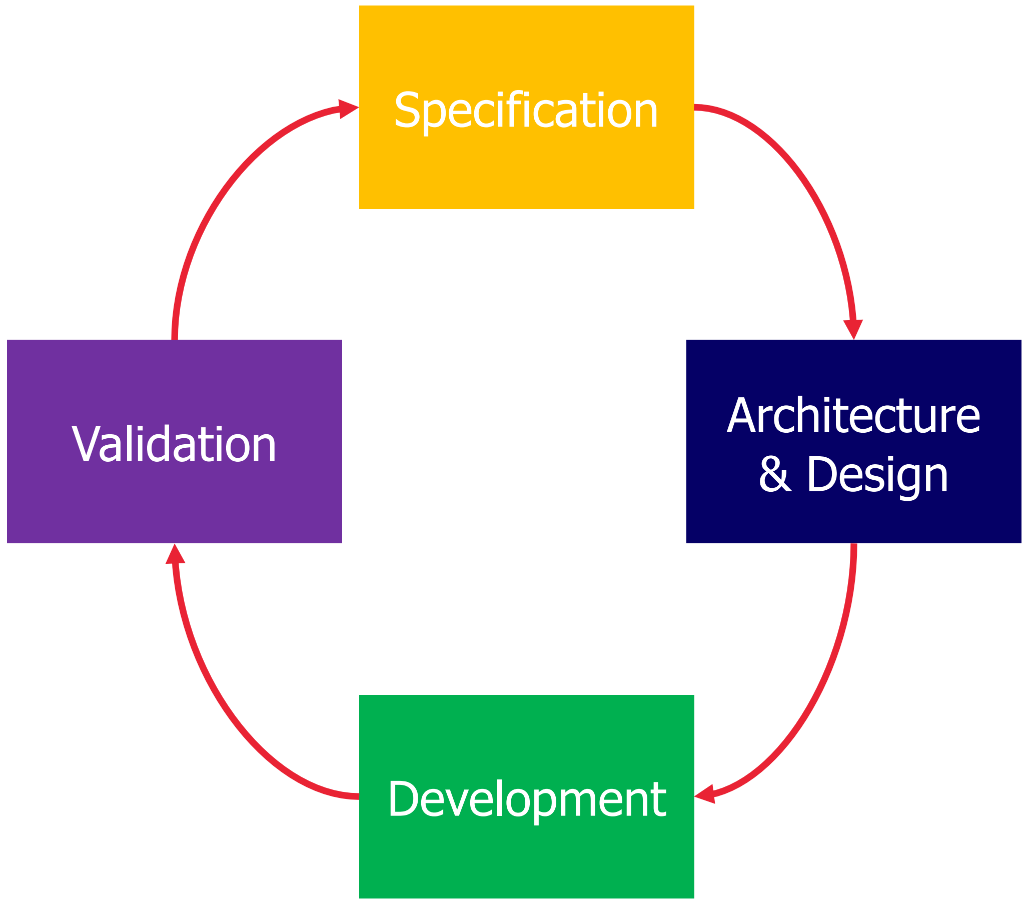
- ~~Software Development Life Cycle (SDLC)~~

- ~~Waterfall Method~~

- Agile
  - Scrum
  - Kanban

Iterative Models

- System is created by successive versions
  - Go through each process step, then iterate
  - Includes feedback between steps
- Lowers the cost of requirement changes
- Allows some client/user feedback
- Smaller step sizes
  - means delivery of something comes sooner
  - and value is created earlier
- Challenges
  - Changes can lead to messy designs
  - and implementations



Agile Manifesto

We 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.

https://agilemanifesto.org

Agile is a Set of SDLC Approaches

Agile Umbrella

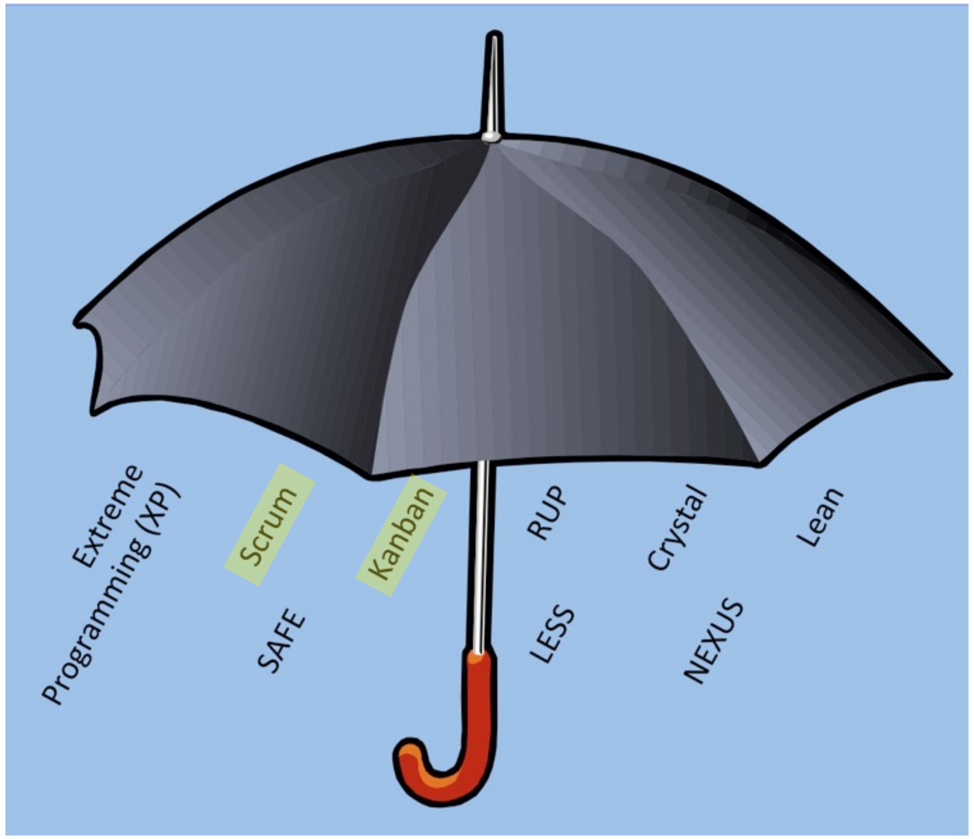
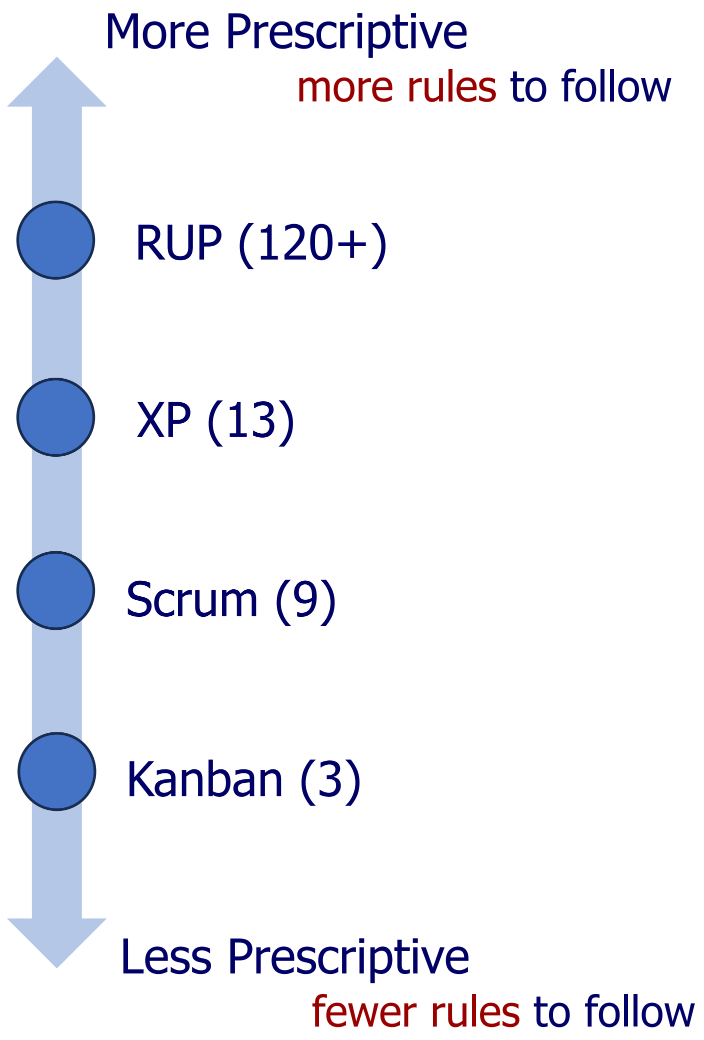


Image: Agile Fundamentals Workshop



The Basis for Agile

1. Agile is built around the notion that change is normal
    - Requirements are not and cannot be fully known up front
    - That existing requirements will change
    - That the team will learn as they go through development

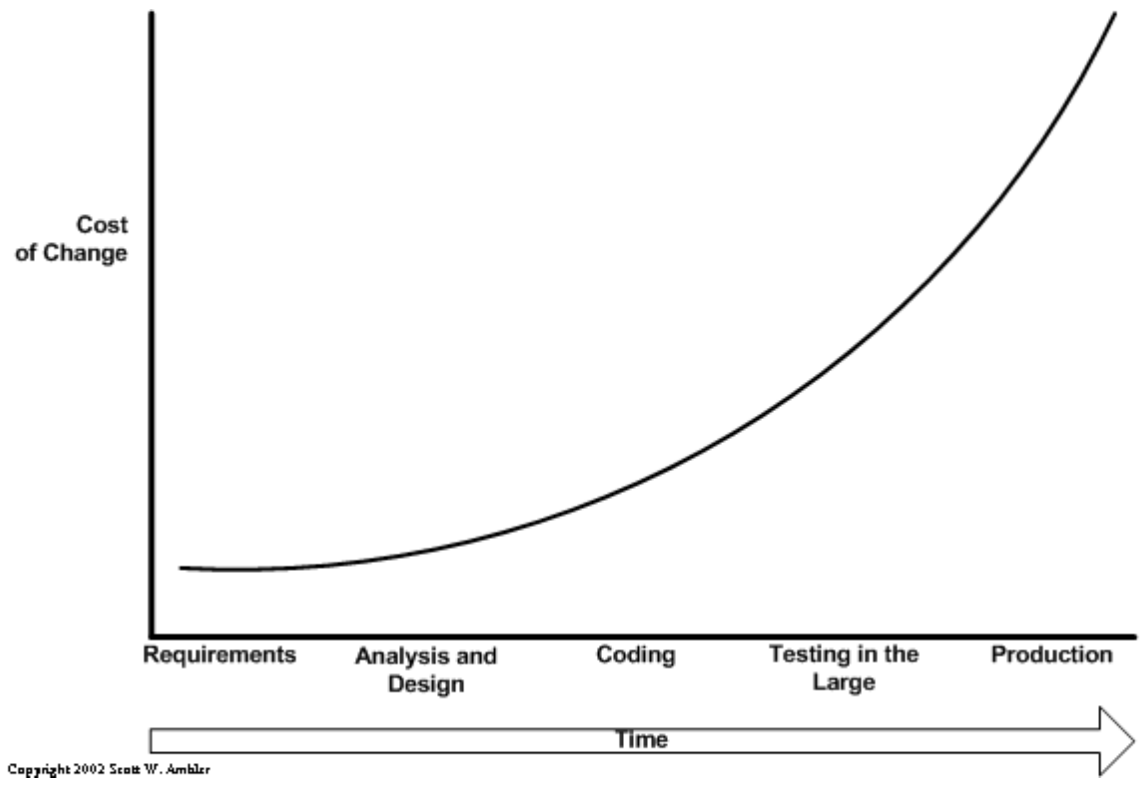
2. Up-front work is a liability
    - Invest in too many details that are wrong, forgotten, …
    - Complexity not discoverable up front

Agile

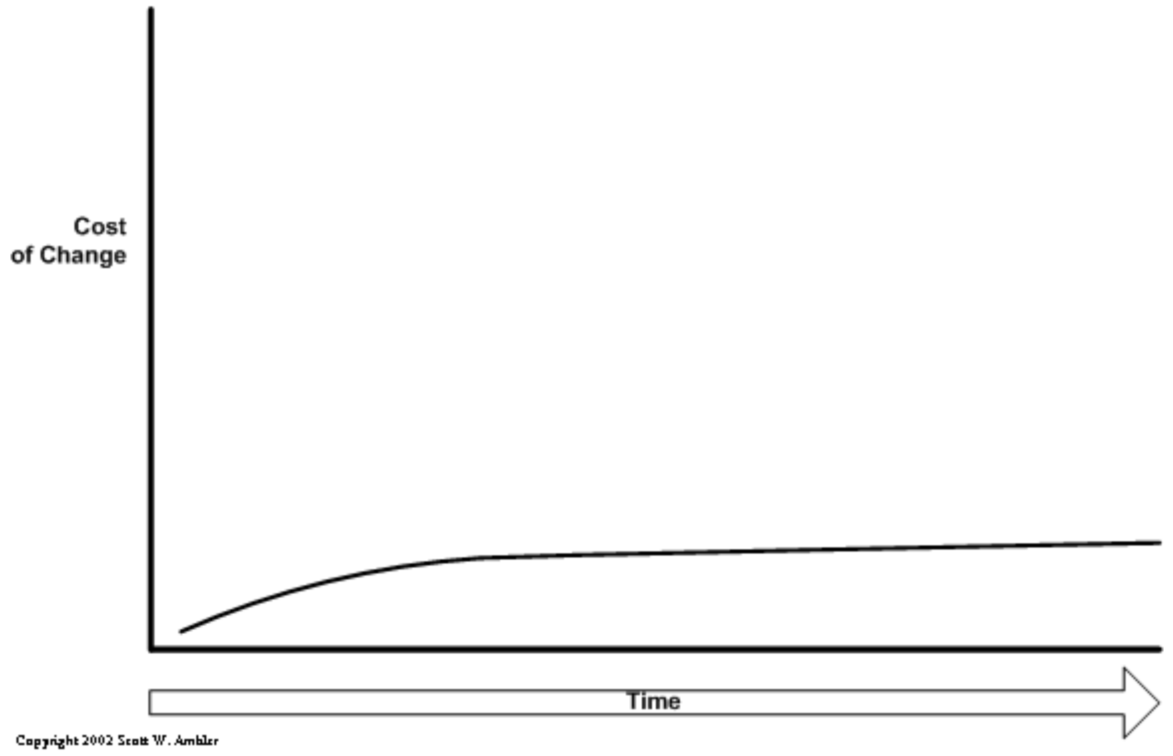
- Focus on
  - Producing small increments of software in a reasonably short time frame
  - Entire process is run during a sprint
  - Sprint results are deployed to production
- Opposite of Waterfall
  - Plans develop incrementally and evolve
- Client collaboration versus client negotiation
- Specification follows from working system, not the reverse
  - Immediate feedback from stakeholders
- Responding to change rather than following a plan
  - Enhancements, new features, and bug fixes are all prioritized as candidates for completion during the next sprint
  - Emphasis on keeping scope small
- Impact of changes will grow over time

Agile Cost of Change Curve

Waterfall Method



Agile



https://agilemodeling.com/essays/costOfChange.htm

Today

- ~~Software Development Life Cycle (SDLC)~~

- ~~Waterfall Method~~

- ~~Agile~~
  - Scrum
  - Kanban

Scrum

Emphasis on small teams delivering discrete pieces of a system

Team has total responsibility for the components it produces

- Team
  - Small, cross-functional, self-organizing units
- Scope
  - Small deliverable scope
  - Delivered in consensus priority order
  - Priorities can be adjusted
- Timeline
  - Small iterations (2 to 3 weeks is typical) emphasizing delivery at the end

Scrum Terminology

- Sprint
  - One iteration through the process
- Backlog
  - Contains all the work that needs to be done
- User stories
  - Describe the function from the user's perspective
  - The user may be another software component or system
- Daily Stand-Up Meeting
  - Discuss progress and plans for what is next in the sprint
  - Keeps the team focused: what is done, what needs to be done, impediments

Scrum

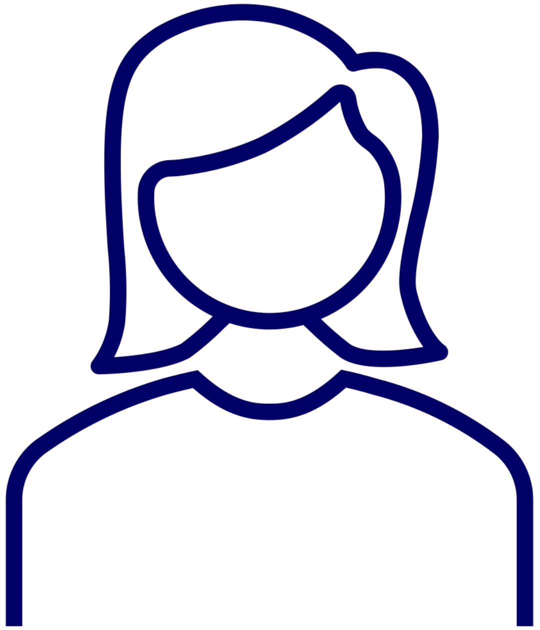
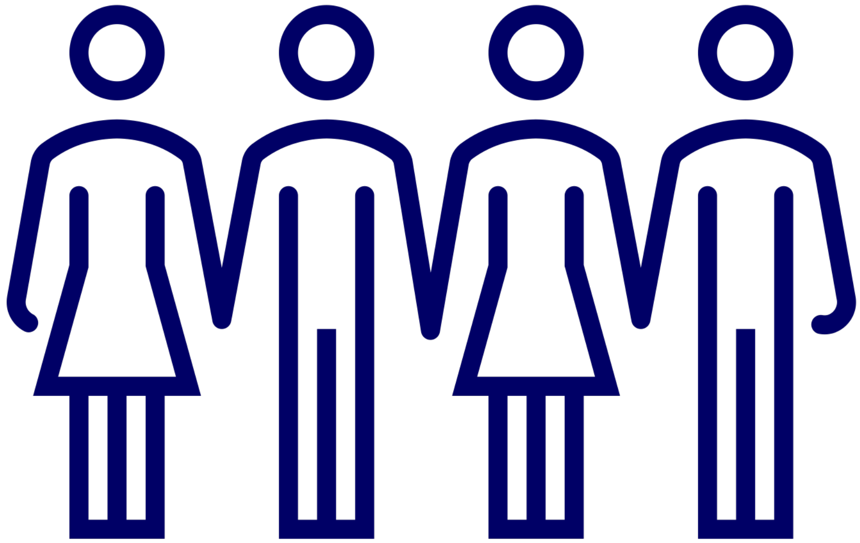
3 Roles

3 Artifacts

3 Ceremonies

3 Roles

Product Owner Scrum Leader Team

3 Artifacts

Product Backlog

Sprint Backlog

Burndown Chart

3 Ceremonies

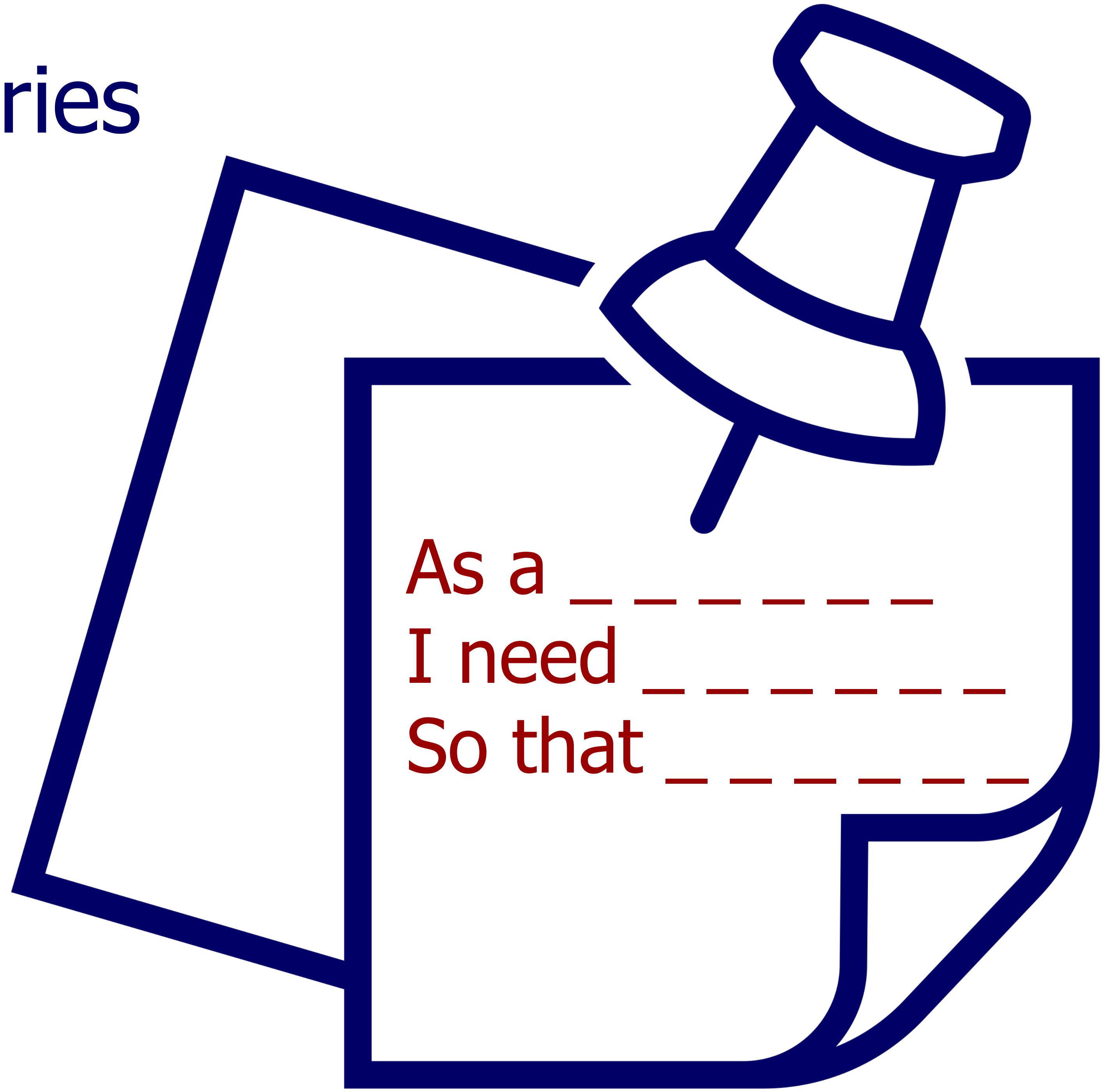
Sprint Planning

Daily Scrum

aka. stand-up
time-boxed

Sprint Review

User Stories



User Stories

- Each of these user stories includes three key components:
  - Who the story is about (the role)
  - What they want to achieve (the feature)
  - Why they want to achieve it (the benefit)
- This format helps ensure that the development team understands the value behind the work they are doing, keeping the user's needs at the forefront of product development

User Stories for E-Commerce Website

As an online shopper,

I want to filter product search results by price range, brand, and rating,

So that I can quickly find the best products within my budget.

User Stories for Mobile Banking App

As a bank customer,

I want to receive real-time notifications for any transactions over $100,

So that I can monitor my account activity and detect unauthorized transactions promptly.

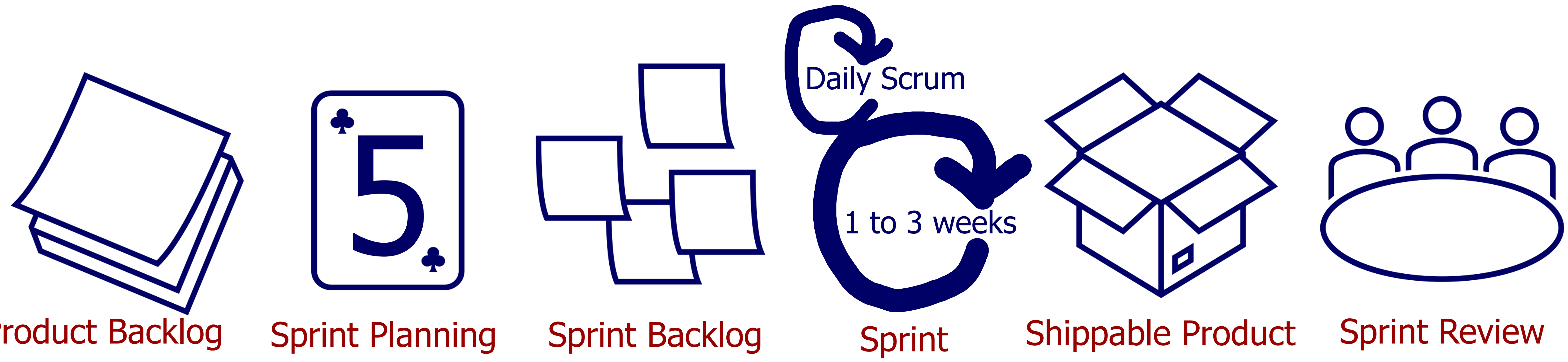
Understanding Story Points in Scrum

- Story Points are a unit of measure for expressing the overall effort required to fully implement a story
  - Not directly related to time or hours, but rather difficulty and effort
  - Story points abstract away from *time* to focus on *size* and *complexity*

How Scrum Works

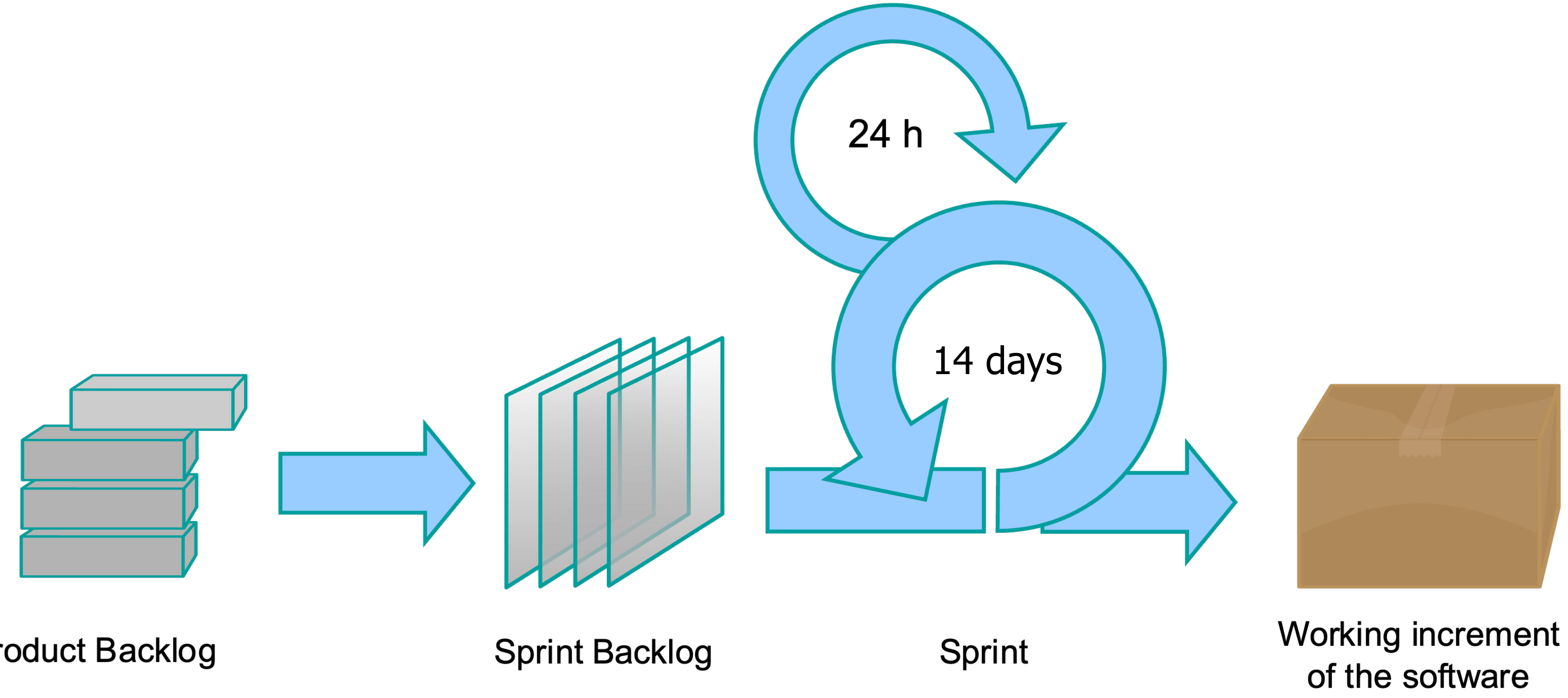
1. Team estimates how much effort each user story will take
2. Product owner provides input on the priorities in the backlog
3. Team updates priorities to allow for dependencies or difficulties
4. The backlog is now a roadmap for the sprint
5. Daily stand-up meetings are held each day during the sprint
6. At the end of the sprint the software is deployed to production

Scrum Workflow

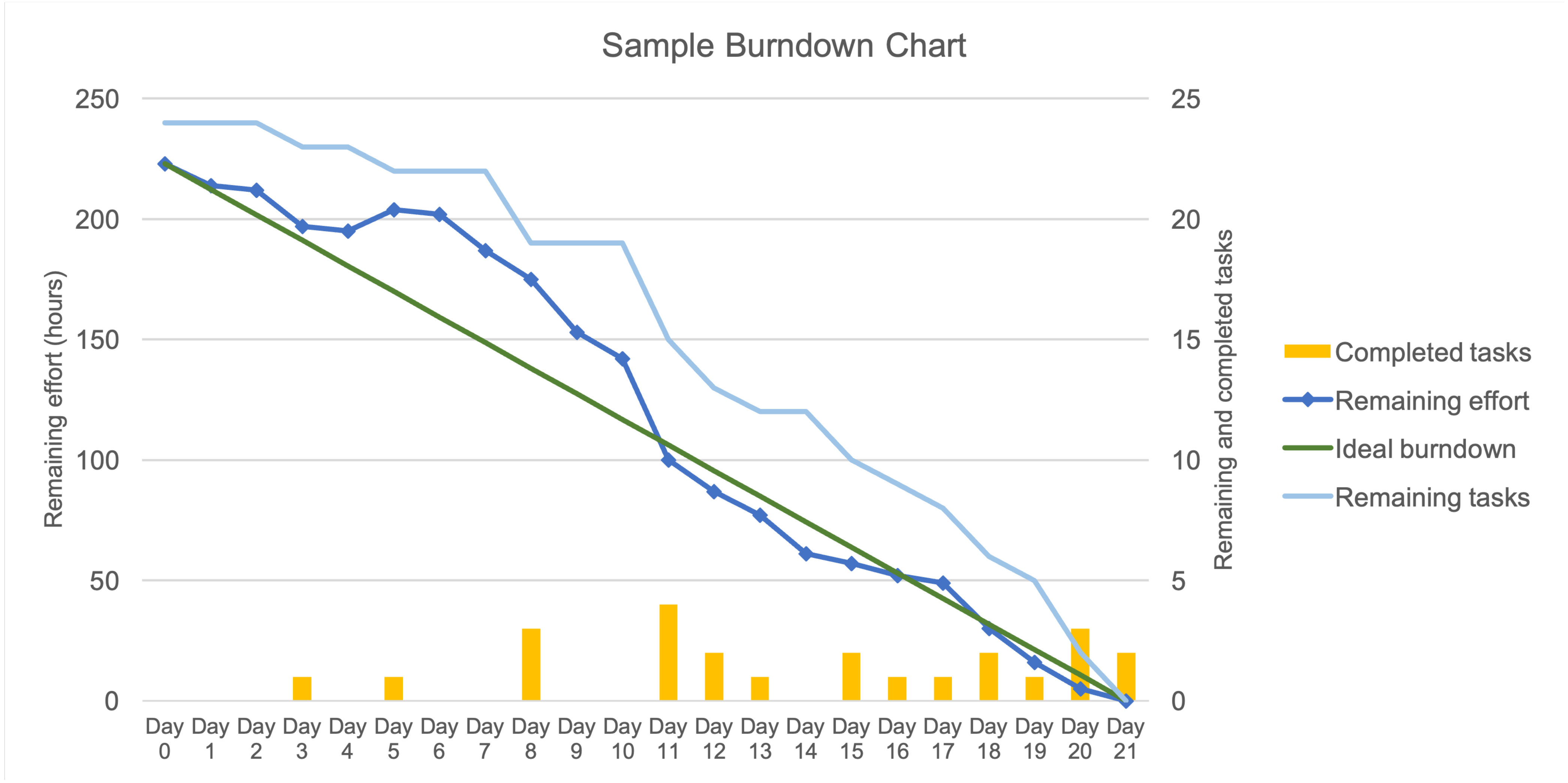


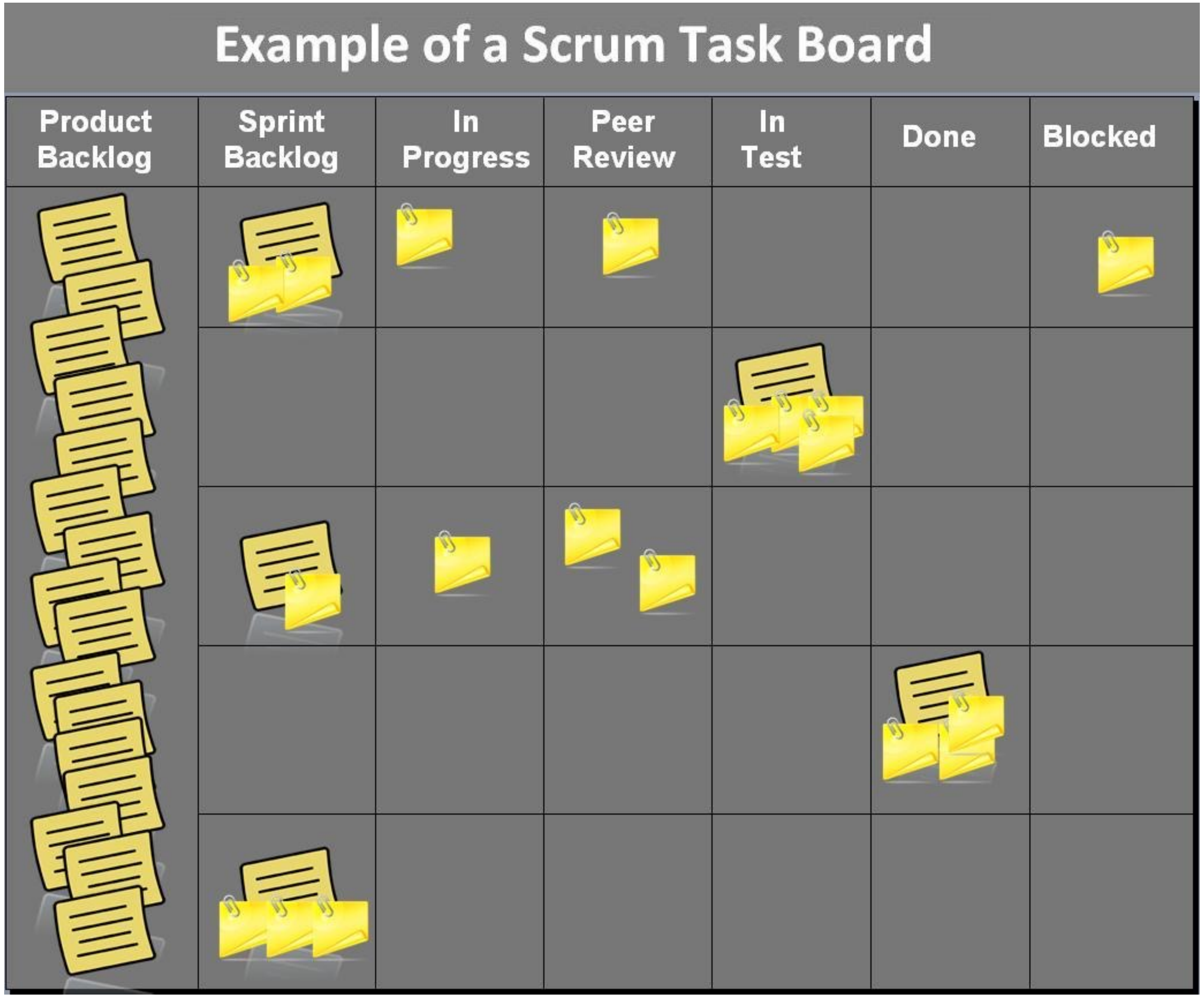
Repeat this workflow for each sprint

Scrum Process



Burndown Chart Example




Example of a Scrum Task Board
Product Backlog
Sprint Backlog
In Progress
Peer Review
In Test
Done
Blocked

Today

- ~~Software Development Life Cycle (SDLC)~~

- ~~Waterfall Method~~

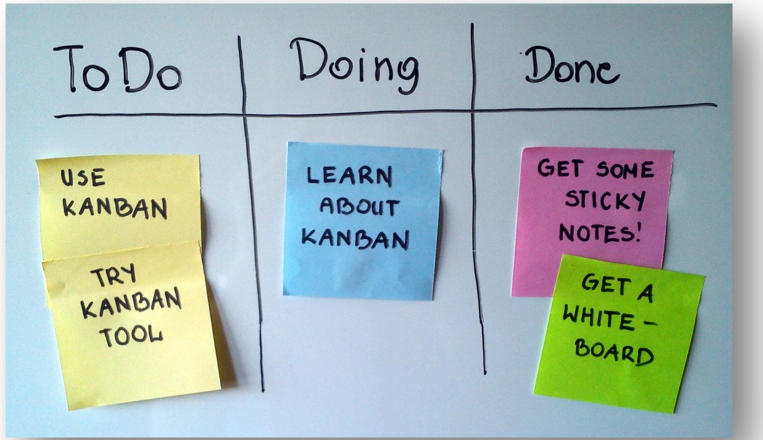
- ~~Agile~~
  - ~~Scrum~~
  - Kanban

Kanban

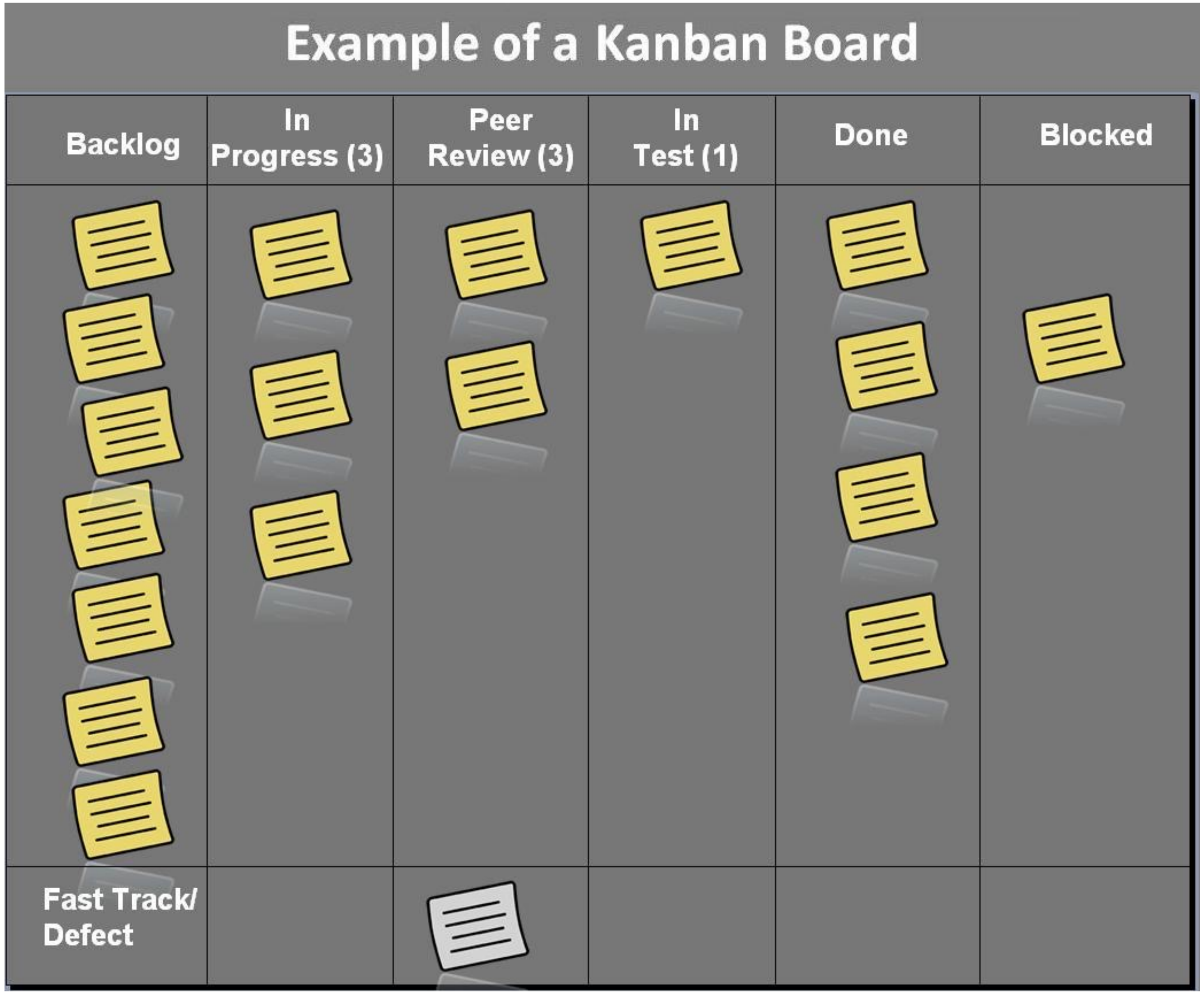
3 rules

1. Visualize Workflow
2. Limit Work In Progress (WIP)
3. Measure Flow

1 tool



Kanban Board



Kanban is a **pull system**

When you have bandwidth you look to the left and pull cards from the left to right

Each stage in the workflow has its own column

Kanban cards are work items, one card per work item