

Agile & Scrum & Kanban Fundamentals





AGENDA

- Agile
- Scrum
- Kanban

AGILE

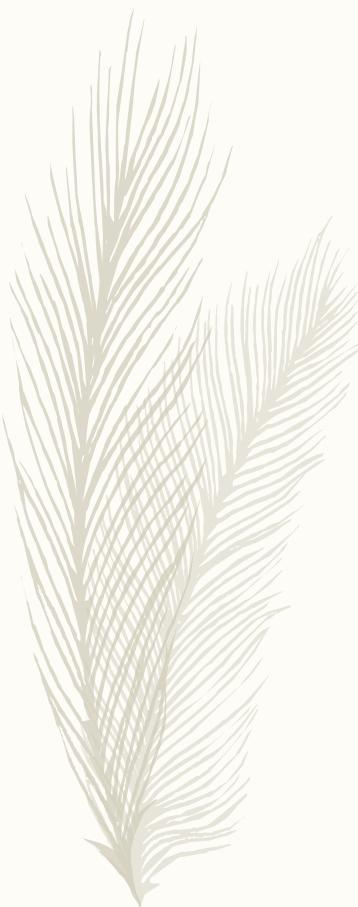
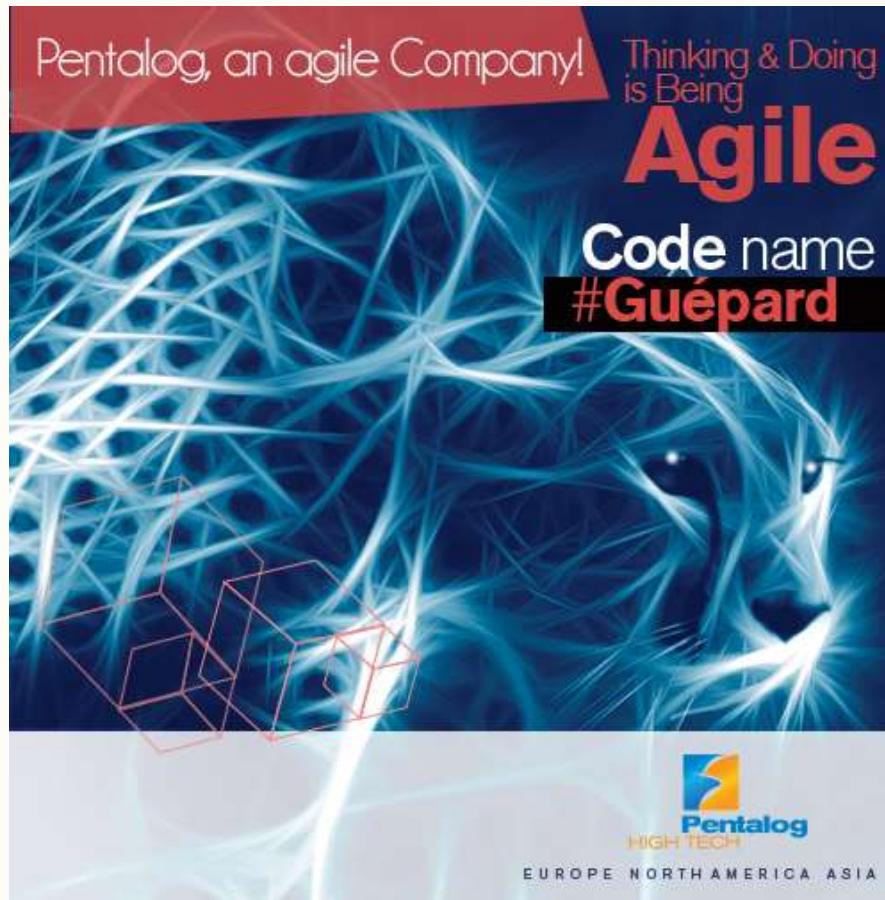




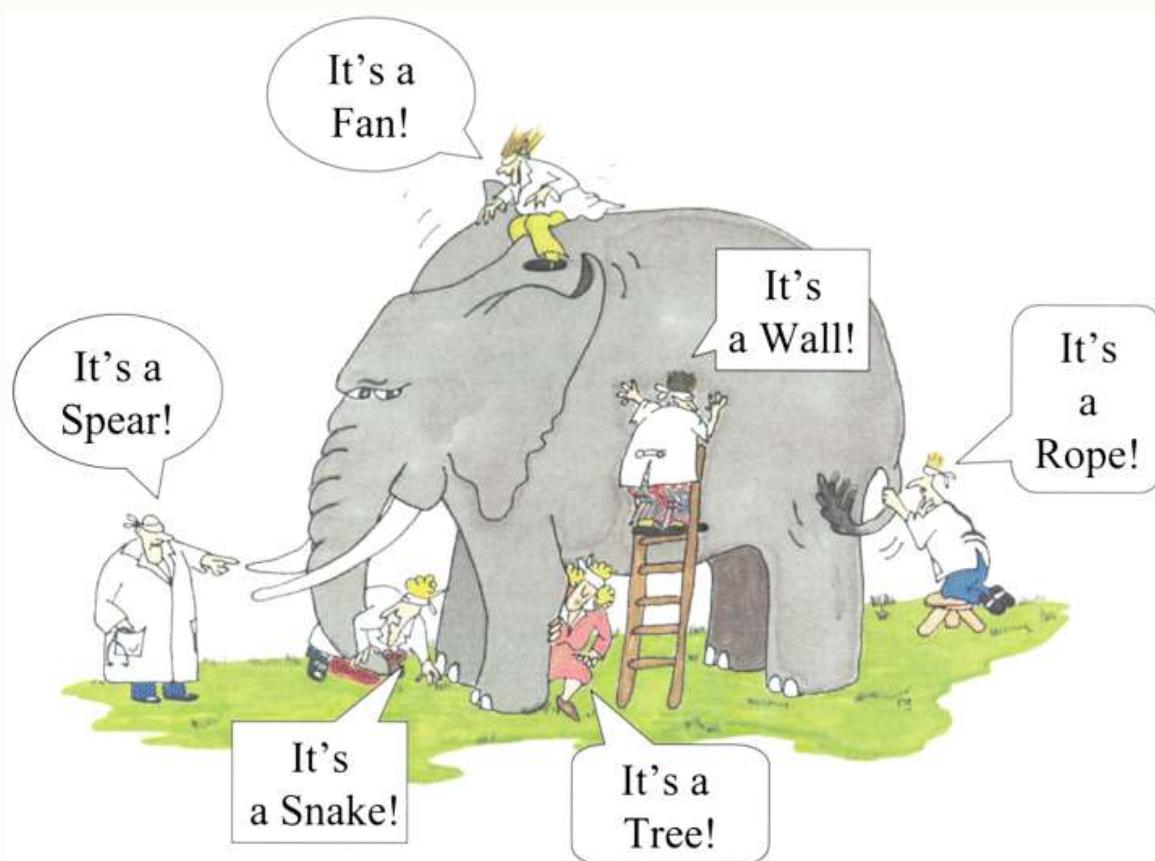
AGENDA – Agile

- Guepard
- Origins
- Agile Manifesto
- Principles
- Waterfall – Agile
- Pitfalls
- Statistics

How we aim to be agile?



The blind men and the elephant





Origins

1930	Plan-Do-Study-Act (PDSA) cycle to the improvement of products and processes - statistician Walter Shewhart of Bell Labs
------	---

1950	Incremental software development
------	----------------------------------

1960	Time boxed and iterative development - NASA
------	---

1970	Evolutionary development model - Tom Gilb
------	---

1990	Lightweight software development
------	----------------------------------

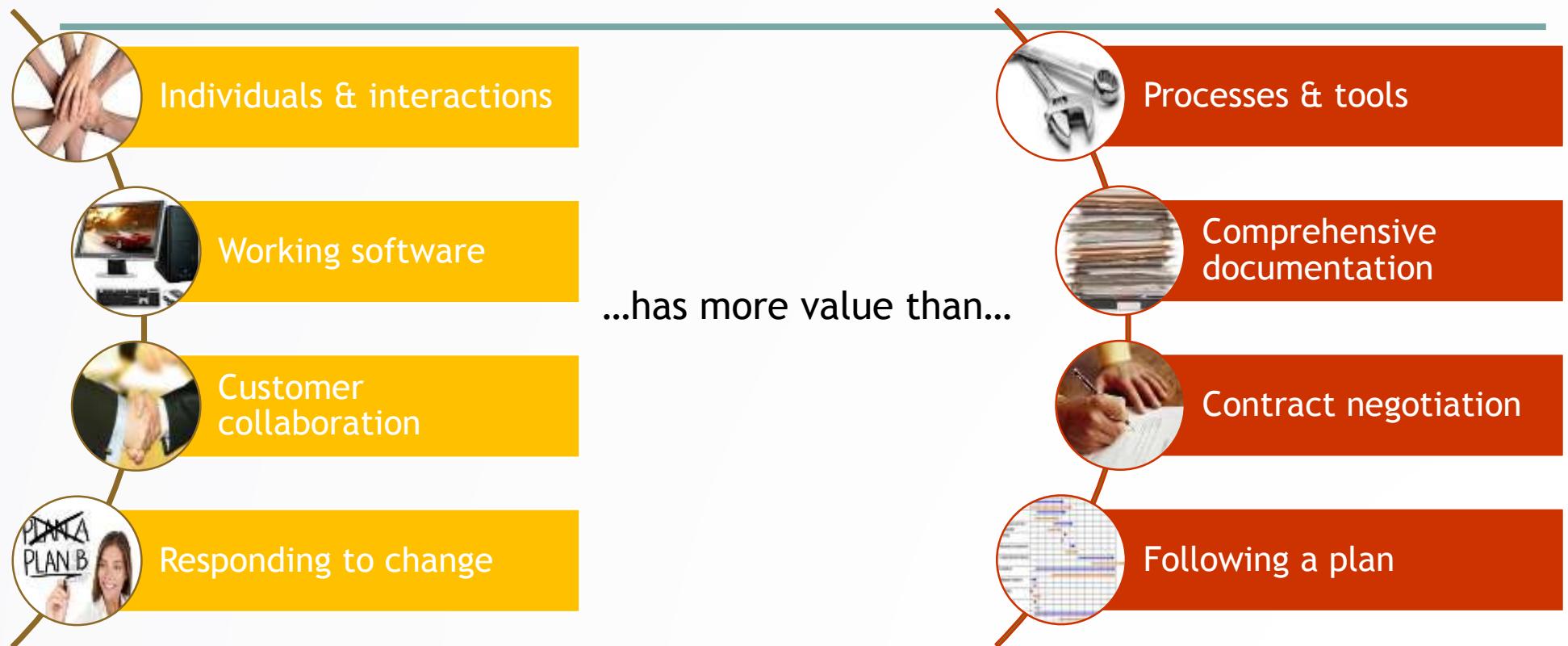
1995	SCRUM - Jeff Shuterland, Ken Schwaber
------	---------------------------------------

1996	Extreme Programming
------	---------------------

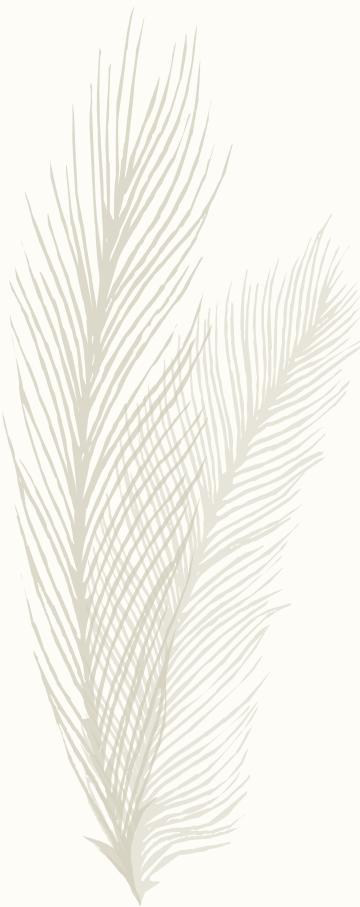
1999	Feature driven development
------	----------------------------

2001	Agile Manifesto
------	-----------------

Agile manifesto (2001)



Agile Principles





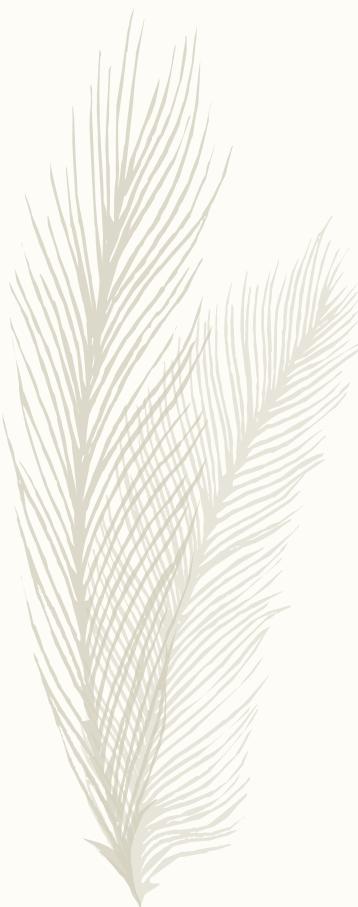
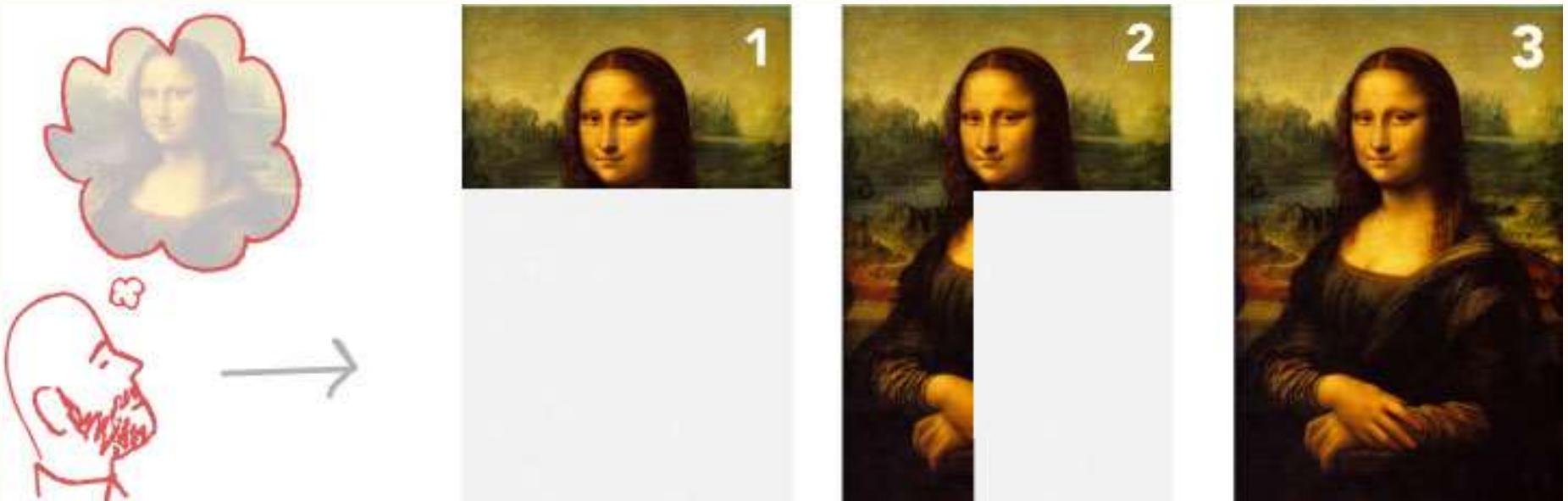
Agile Concepts: Main ideas

- Short feedback loops
- Release Ready deliverables
- Self organizing teams
- Deliver value for business
- Embracing change
- Inspect and adapt

Incrementing Mona Lisa

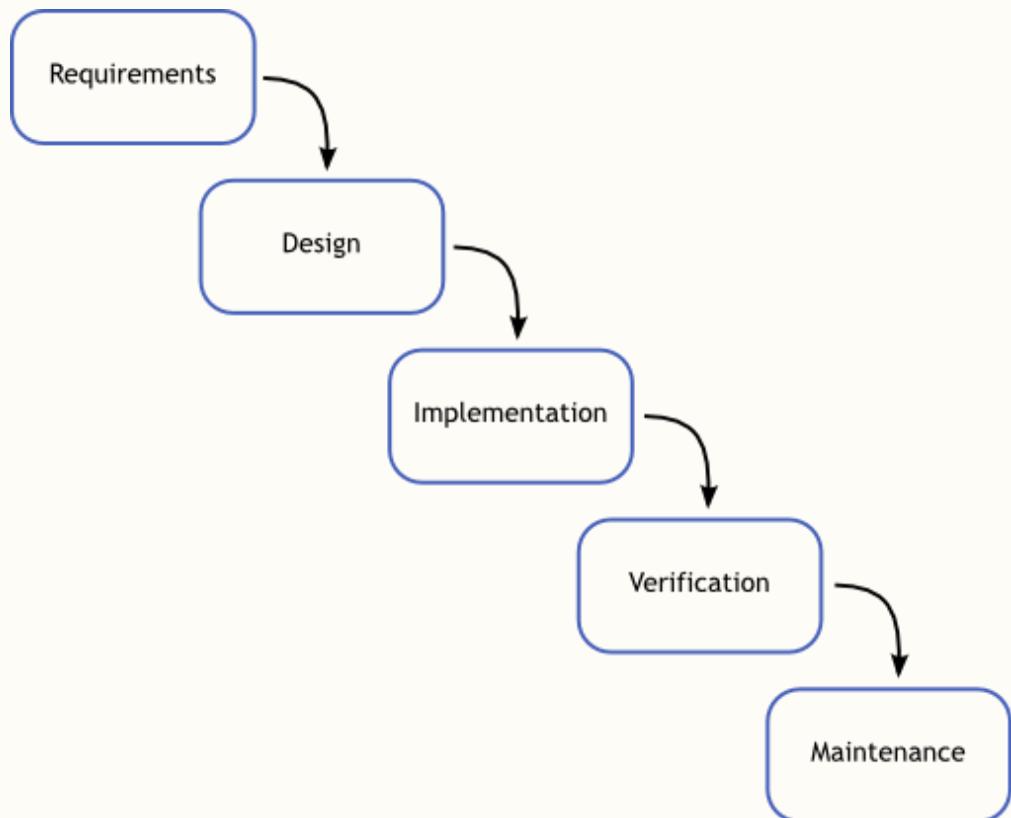


Iterating Mona Lisa

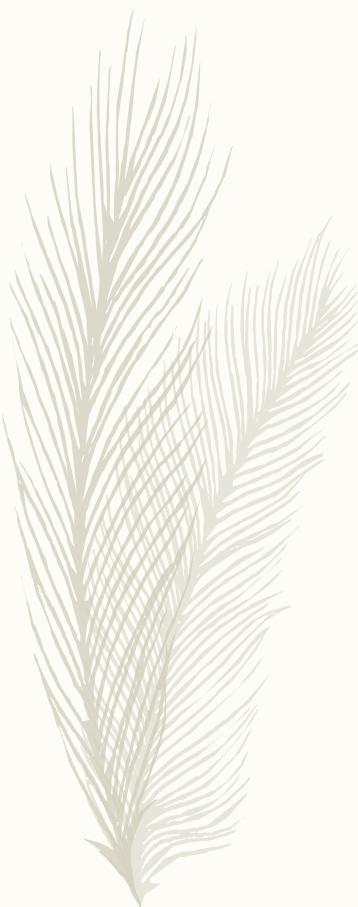


Waterfall

- The **waterfall** model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through different phases.
- 1970's



Waterfall model



Waterfall

Advantages	Disadvantages
Simple to explain and recall: “Do the requirements, then design, and then implement.”	Long cycles
It gives the illusion of an orderly, accountable and measurable process, with simple document-driven milestones (such as “requirements complete”)	Fatal loss of knowledge in phase shifts
	Progress measured based on documentation
	Lack of effective communication
	No feedback
	Does not allow change during project

Agile

- Agile is a collection of principles, values and frameworks that can be applied on an software development projects. This model is more flexible than traditional methods, making it a better fit in a fast changing environment.
- 2001

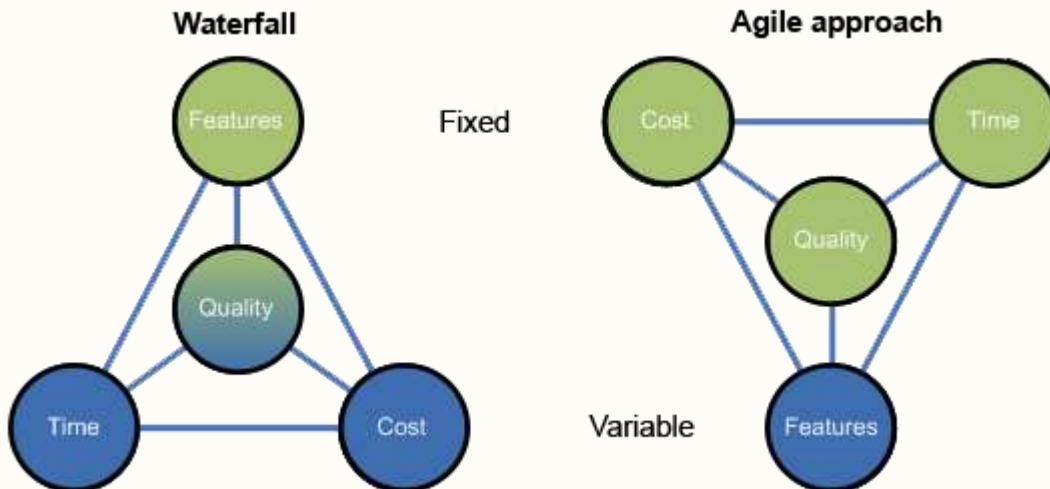


Agile

Advantages	Disadvantages
Focus on the features that business needs most	Agile is more complex to understand and describe
Deliver working and tested software frequently (max every 4 weeks)	
Business and IT work together to complete all the work (collaboration and communication)	
Improve continuously software, processes, practices and team (feedback / inspect and adapt)	



Waterfall vs Agile



What is Agile?

<https://www.youtube.com/watch?v=Z9QbYZh1YXY&list=PL8laJpDvOqJ9tJGnu9u10eZZC-b5QqbNM&index=5>



Flavors of Agile

	SCRUM	KANBAN	XP
Cadence	Iterations up to 4 weeks.	Continuous flow	Iterations up to 2 weeks.
Release method	At the end of each sprint if approved by the product owner.	Continuous delivery or at the team's discretion.	At the end of each iteration.
Roles	Product owner, Scrum Master, Development team	No existing roles. Some teams enlist the help of an agile coach.	Customer, Developer, Tracker, Coach
Key metrics	Velocity	Cycle time	Velocity (the ratio of ideal time estimated versus time spent)
Change philosophy	Changes should wait for next sprint.	Change can happen at any time.	Embrace change. Release early and release often, that is the philosophy.
Task Order	The items are dragged in the sprint based on PO order.	Task are taken in a strict priority order.	Task are taken in a strict priority order.
Engineering practices	No	No	Yes



Agile pitfalls

- Lacking sustained leadership commitment
- Settling for unempowered product owners
- Falling back into old habits
- Modifying stock agile frameworks before the organization has thoroughly absorbed an agile philosophy
- Scaling up before there is success
- Reliance on cliché agile practices for success
- Failure to employ empirical forecasting
- Neglecting distributed teams

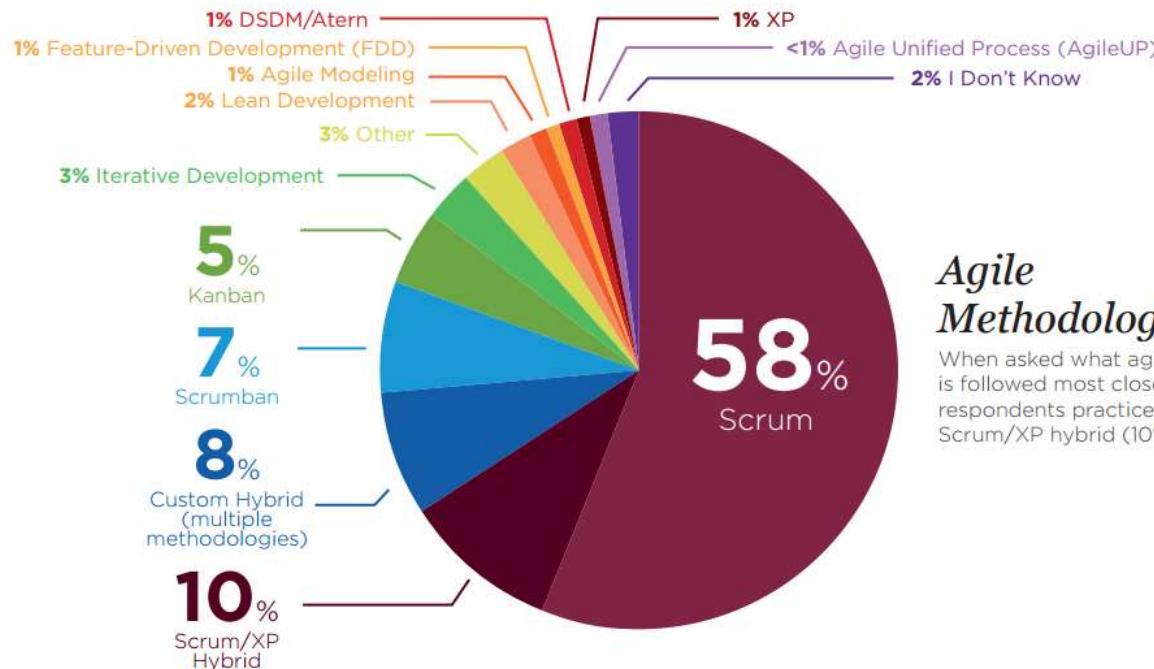
What Agile is not?

- Agile is NOT an excuse to stop producing documentation.
- Agile is NOT an opportunity to eliminate planning.
- Agile is NOT open season on scope creep.
- Agile is NOT about blindly following a set of “best” practices, whether or not they’re best for your project.



Agile Statistics

AGILE METHODS AND PRACTICES



Agile Methodologies Used

When asked what agile methodology is followed most closely, nearly 70% of respondents practice Scrum (58%) or Scrum/XP hybrid (10%).

SCRUM



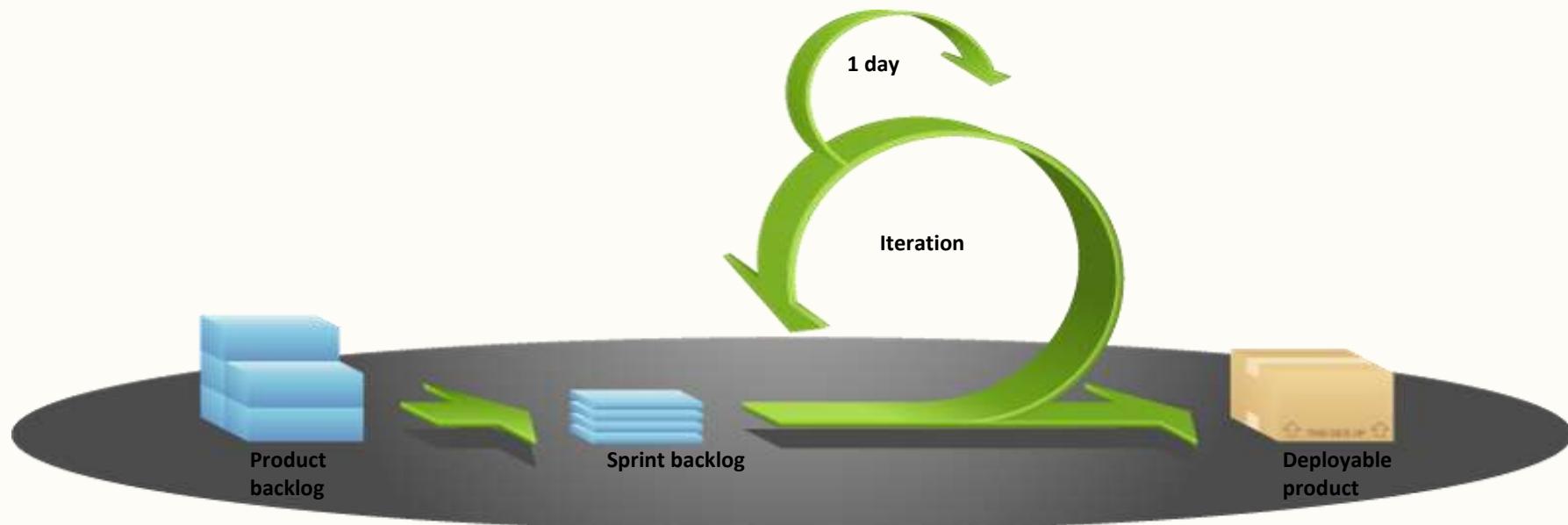


AGENDA – Scrum

- Values
- Roles
- Events
- Artifacts
- DoR / DoD
- KPIs
- Mistakes
- Statistics

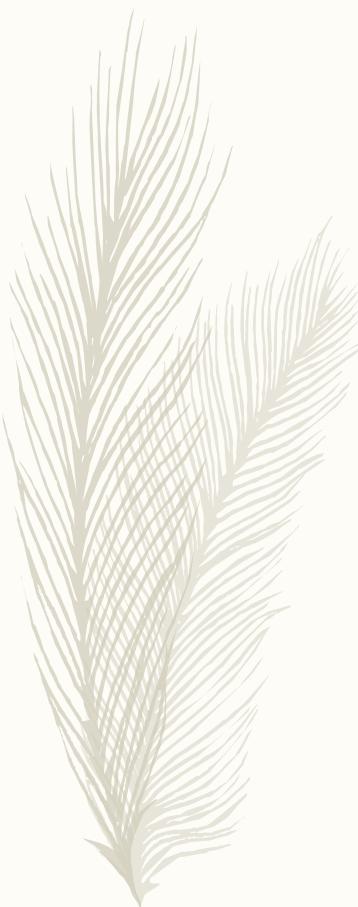
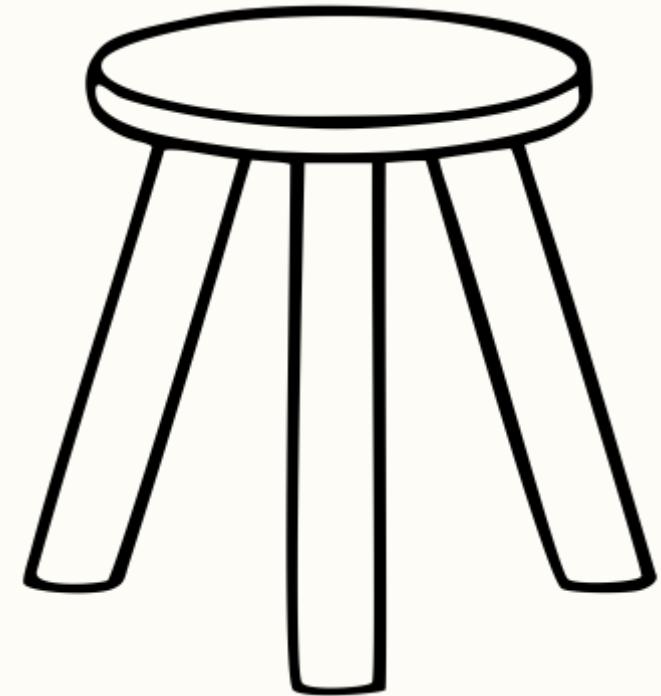
Scrum

- 1993 Jeff Sutherland created the Scrum process
- Consistent fixed length iterations → “sprint” (max 1 month)
- Deliver working software frequently
- References: Scrum Guide



Scrum Values

https://www.youtube.com/watch?v=o_grcE4tkzY&list=PL8laJpDvOqJ9tJGnu9u10eZZC-b5QqbNM&index=8



Scrum - Roles

- Self-organizing, cross-functional
- Composition: Development team (3-9 persons), Scrum Master, Product Owner
- Motto: Agile is about the success of the team, not individual success or heroic behavior.

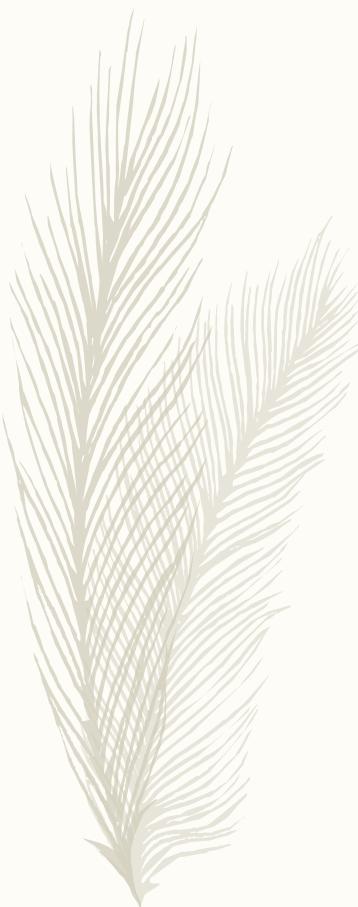




Product Owner

Product Owner is the sole person responsible for managing the Product Backlog:

- Clearly expressing Product Backlog items;
- Ordering the items in the Product Backlog to best achieve goals and missions;
- Optimizing the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next;
- Ensuring the Development Team understands items in the Product Backlog to the level needed.



Product Owner's Deliverables

- Vision
- User Personas
- User Scenarios / Flows
- Product Roadmap / Releases
- Product Backlog
- Product Launch

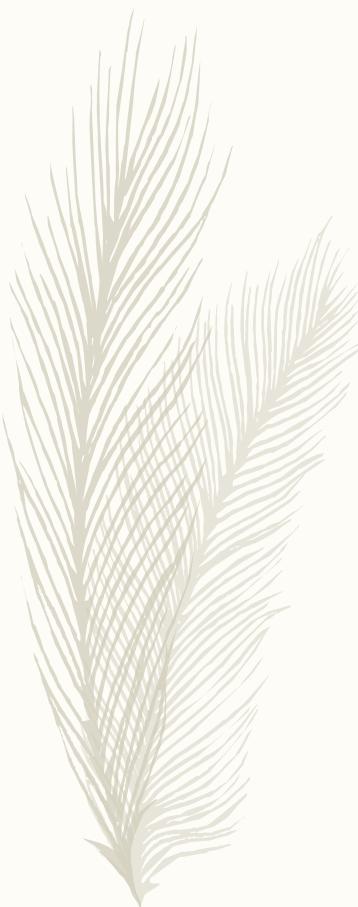
Product Owner

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=502ILHJX9EE](https://www.youtube.com/watch?v=502ILHJX9EE)

Agile Product Ownership

in a nutshell





Scrum Master

Scrum Master Service to the Product Owner

- Finding techniques for effective Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Understanding product planning in an empirical environment;
- Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
- Understanding and practicing agility;
- Facilitating Scrum events as requested or needed.

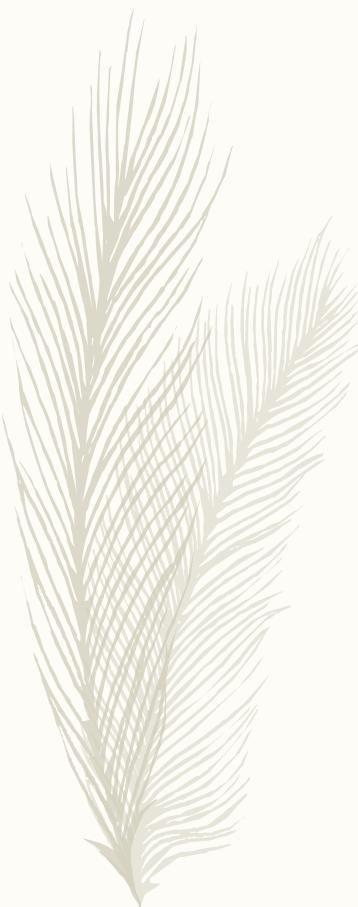
Scrum Master Service to the Development Team

- Coaching the Development Team in self-organization and cross-functionality;
- Helping the Development Team to create high-value products;
- Removing impediments to the Development Team's progress;
- Facilitating Scrum events as requested or needed;
- Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

Scrum Master Service to the Organization

- Leading and coaching the organization in its Scrum adoption;
- Planning Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact Scrum and empirical product development;
- Causing change that increases the productivity of the Scrum Team;
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

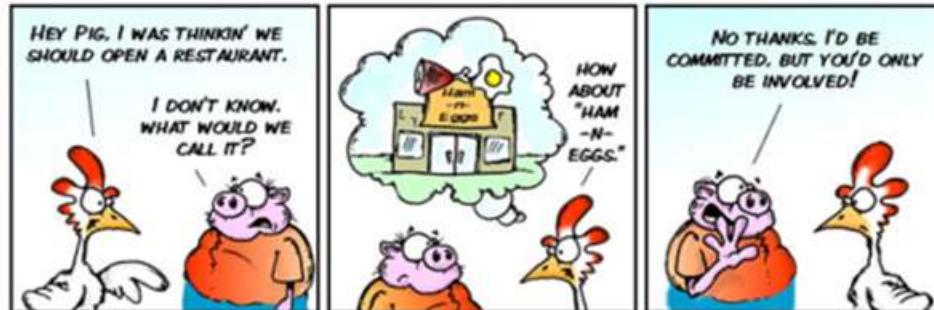
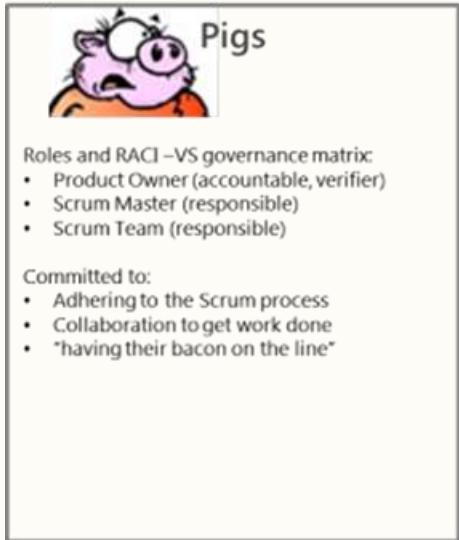
Development Team



Development Teams have the following characteristics:

- The team has between 3 and 9 members;
- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;
- Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule;
- Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule;
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

What about stakeholders?



Is someone who's impacted by the outcome of the solution and is clearly more than an end-user.

A stakeholder may be one of the following:

- The “gold owner” who funds the project
- A direct or indirect user
- A senior manager
- An operations or IT staff member
- A developer(s) working on other systems



Scrum - Events

Sprint Planning

(maximum 8 hours for 1 month Sprint)

Sprint Planning Structure

- Sprint Goal
- What will contain the next increment?
- How will achieved it?

Sprint Planning Actions

- Clarify Requirements
- Identify and estimate tasks
- Commit to the Sprint Goal
- Forecast the Sprint Backlog

Scrum - Events

Daily Scrum

(maximum 15 minutes)

- What did I do yesterday that helped the Development Team meet the Sprint Goal?
- What will I do today to help the Development Team meet the Sprint Goal?
- Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?



Scrum - Events

Sprint Review – Demo

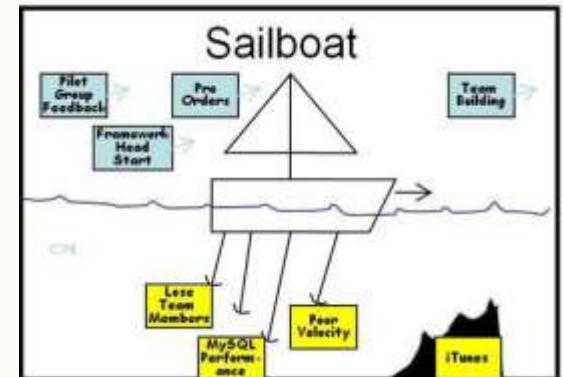
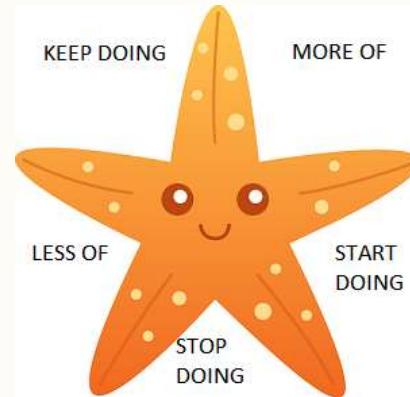
(maximum 4 hours for 1 month Sprint)

- Attendees include the Scrum Team and key stakeholders invited by the Product Owner;
- The Product Owner explains what Product Backlog items have been “Done” and what has not been “Done”;
- The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;
- The Product Owner discusses the Product Backlog as it stands. He or she projects likely completion dates based on progress to date (if needed);
- The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;
- Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,
- Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product.

Scrum - Events

Sprint Retrospective (maximum 3 hours for 1 month Sprint)

- Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- Identify and order the major items that went well and potential improvements; and,
- Create a plan for implementing improvements to the way the Scrum Team does its work.





Scrum - Artifacts

Product Backlog

The Product Backlog is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.

PROGRESS - Burndown chart

A burn down chart is a graphical representation of work left to do versus time.

FORECAST – Velocity

Velocity is a metric that predicts how much work an Agile software development team can successfully complete within a sprint.

Product Backlog

- Single list of requirements based on high level analysis
- Contains the scope of the project, listed in epics, features and user stories
- Contains:
 - Functional requirements
 - Non functional requirements, e.g. performance
 - Housekeeping, e.g. ongoing improvements
 - Infrastructure, e.g. technical setup
 - Deployment (product packaging), e.g. documentation

Backlog 10 of 38 issues visible Clear all filters				Create Sprint
		TIS-48 Large Group Participant should be able to cancel reservation	3.0	3
		TIS-44 Reward Customers an extra 5-10% when they book a large trip	2.1	3
		TIS-46 Update LocalTransportController to handle multiple travel providers in one reser	3.0	5
		TIS-42 Extend booking experience in UI to include multiple hotels on one reser	2.1	5
		TIS-43 Extend booking experience in UI to include multiple flights on one reser	2.1	13
		TIS-41 Update LodgingController to handle multiple travel providers in one reser	3.0	1
		TIS-40 Update FlightController to handle multiple travel providers in one reser	2.1	0
		TIS-39 Update UI controls on travel booking page to handle groups up to 30	2.2	0
		TIS-47 UI should allow users to book on large or personal space craft	3.0	2
		TIS-45 Email non registered users to sign up with Teams In Space	2.1	2

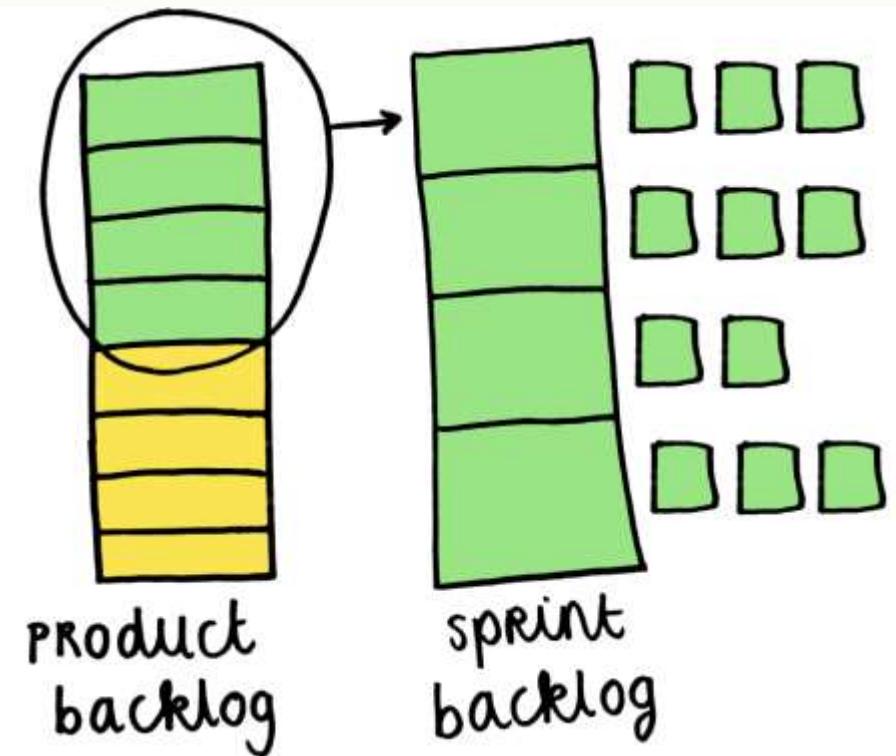
Product Backlog Validators

- Detailed: order items more detailed in top
- Estimated: expressed in Story points
- Emerged: keep it up to date, new items added, existing items modified;
- Prioritized: all items are ordered
- Owned: by the Product Owner



Sprint Backlog

- Contains the scope of the iteration
- Focus on the customer's highest priority stories





User Story

- One action for one user satisfying one business goal
 - As <user> I <action> so <business goal>
- Little piece of functionality (end to end)
- Discussed between business and team
- Understandable for business and team
- Is a work document, not documentation
- Elaborated just-in-time
- Small enough to fit in a single iteration
- Tested to the satisfaction of the customer



User story – Best Practices

INVEST

- Independent
- Negotiable
- Valuable
- Estimate-able
- Sized
- Testable

Minimum viable product (MVP) = a new product developed with sufficient features to satisfy early adopters

Minimum marketable features (MMF) = Smallest piece of functionality that can be sold / released to the end user

User Story with Acceptance Criteria

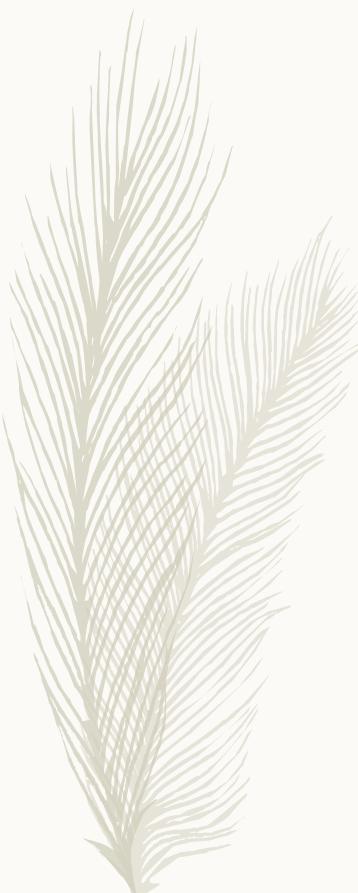
User Story example:

As an Administrator, I want to be able to create User Accounts so that I can grant users access to the system.

Acceptance Criteria example:

- If I am an Administrator, I can create User Accounts.
- I can create a User Account by entering the following information about the User: a. Name, b. Email address, c. Phone Number d. License Number (Power/Basic/None), e. Account Status (Active/Inactive), f. Reports to (from a list of "Active" Users)
- The system notifies me that it sent an email to the new User's email address, containing a system-generated initial password and instructions for the person to log in and change their password.

User Story with Acceptance Criteria



User Story: As an Account holder I want to withdraw cash from an ATM so that I can get money when the bank is closed.

Scenario 1: Account has sufficient funds

Given: The account balance is 100\$ and the card is valid and the machine contains enough money

When: The account holder requests 20\$

Then: The ATM should dispense 20\$ and the account balance should be 80\$ and the card should be returned

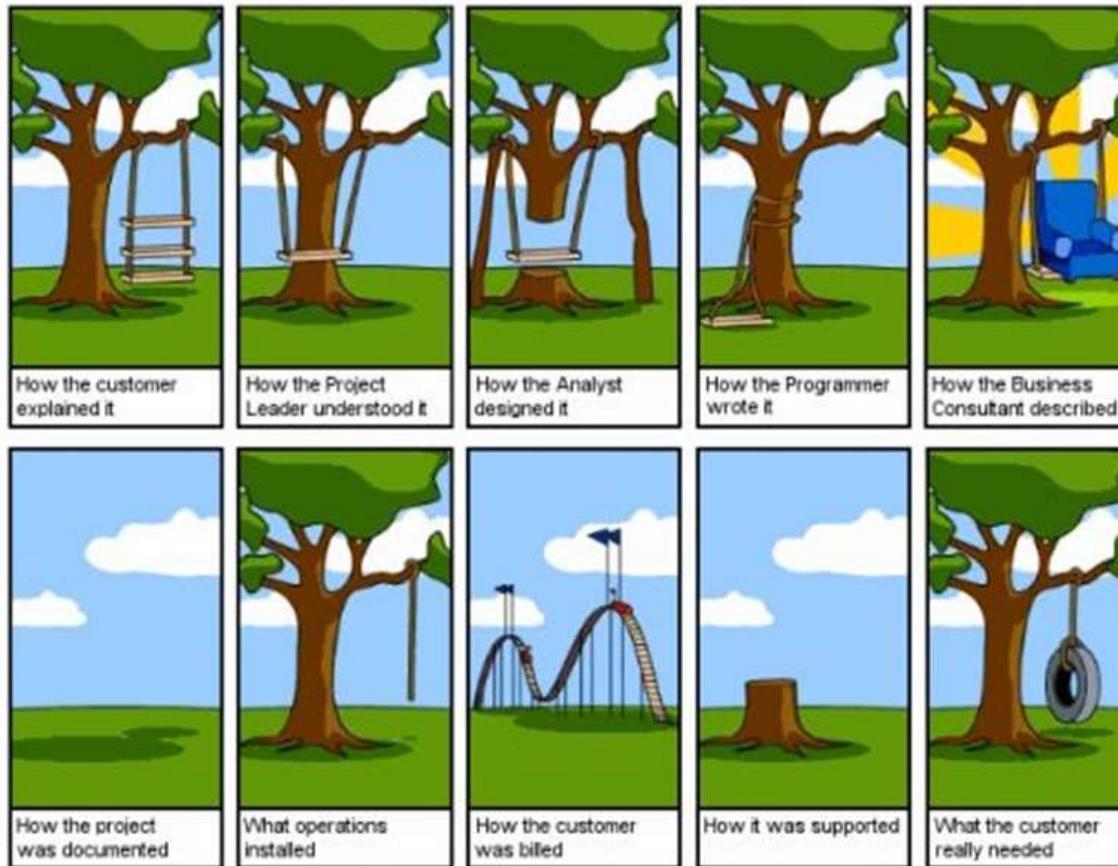
Scenario 2: Account has insufficient funds

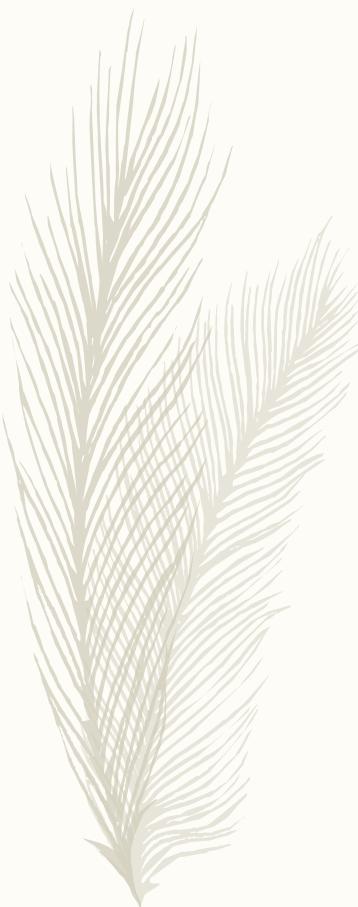
Given: The account balance is 10\$ and the card is valid and the machine contains enough money

When: The account holder requests 20\$

Then: The ATM should not dispense any money and the ATM should say there are insufficient funds and the account balance should be 10\$ and the card should be returned

Be careful





User Story - Estimation

- Relative sizing in story points
- Estimate functional complexity
- Fast, easy and simple
- Points do not decay over time
- Accurate up-front estimates are not possible
- Planning Poker

User story - Prioritization

- Focus on features that business really needs
- Return on investment and cost of delay
- Cost of development
- Risks

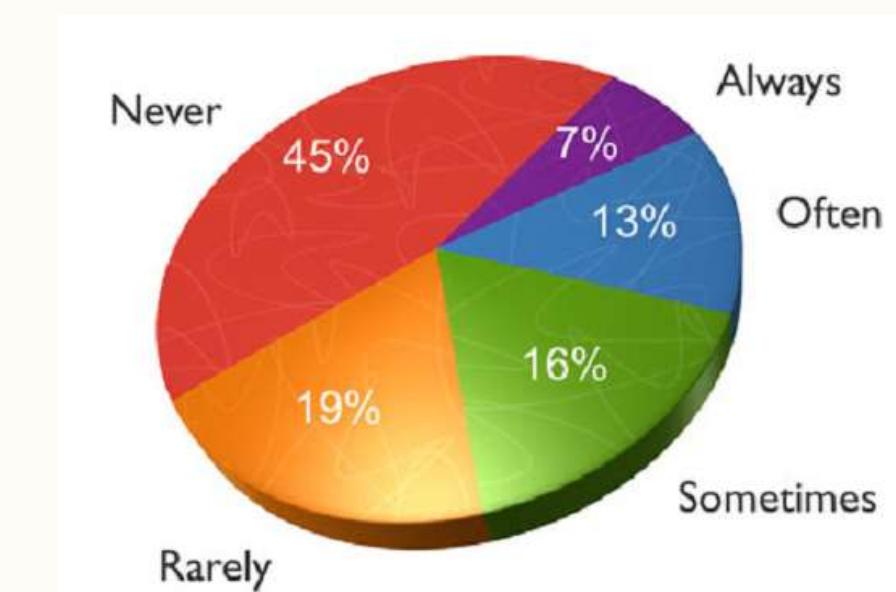
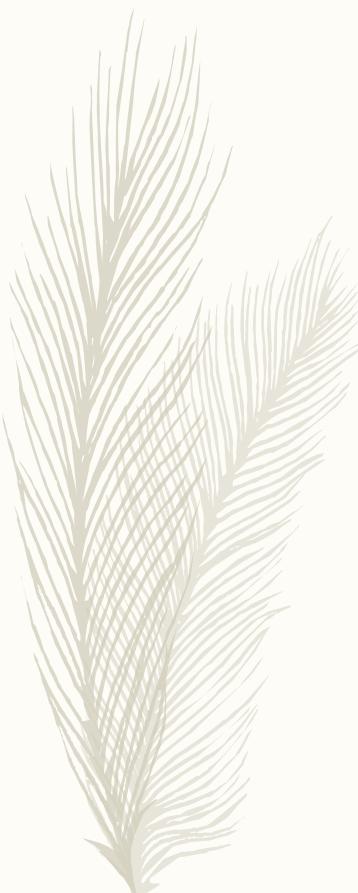


Figure 1.3 A study by the Standish Group indicates how often features are used in a typical application.

Definition of Ready (DoR)



A “ready” item should be clear, feasible and testable:

- **clear** if all Scrum team members have a shared understanding of what it means.
- **testable** if there is an effective way to determine if the functionality works as expected, acceptance criteria ensure that each story can be tested
- **feasible** if it can be completed in one sprint, according to the definition of done, this implies two things: the item must be small enough, and it must not be too complex



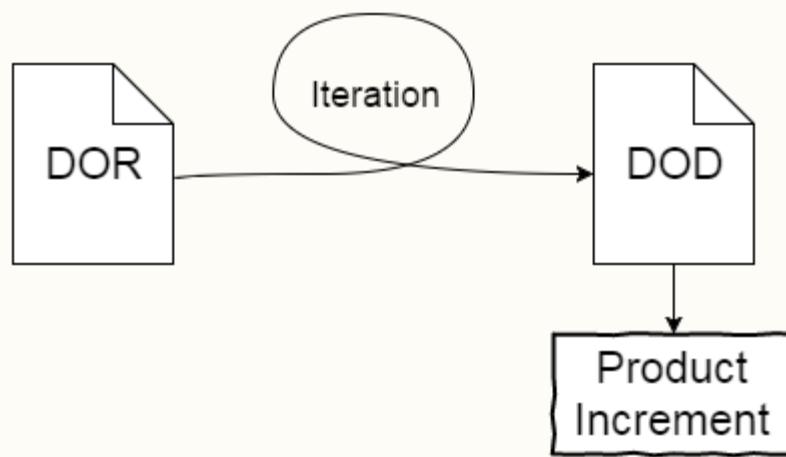
Definition of Done (DoD)

- Clarify when a user story is “done”
- Prevents half work
- Respected by entire team(s)

Example

- All tasks completed
- Code refactored
- Green build
- End-to-end test / acceptance test
- Proxy check completed
- No more bugs (known)
- Deployed to test environment

Acceptance of the stories?

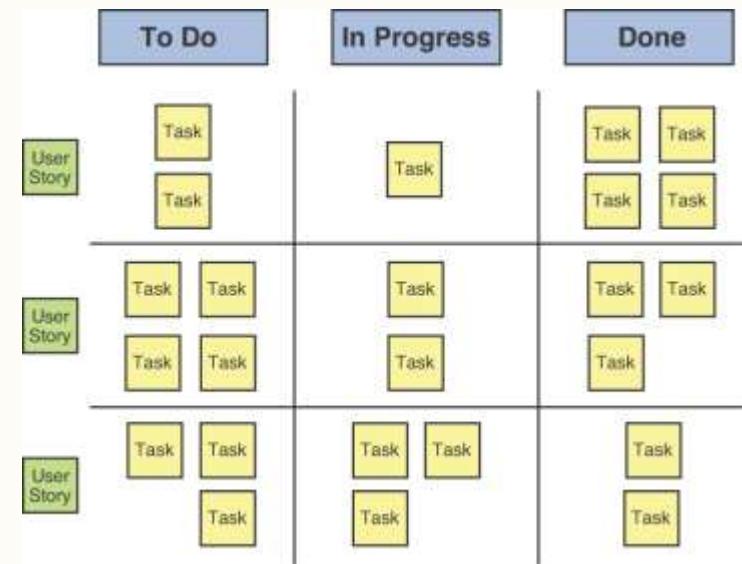


Scrum Board

- Visualise work
- Discover impediments

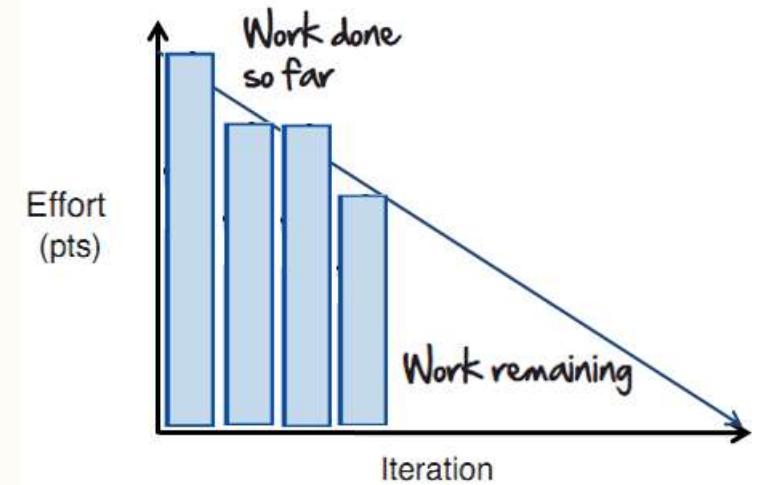
Tools:

- JIRA
- Trello

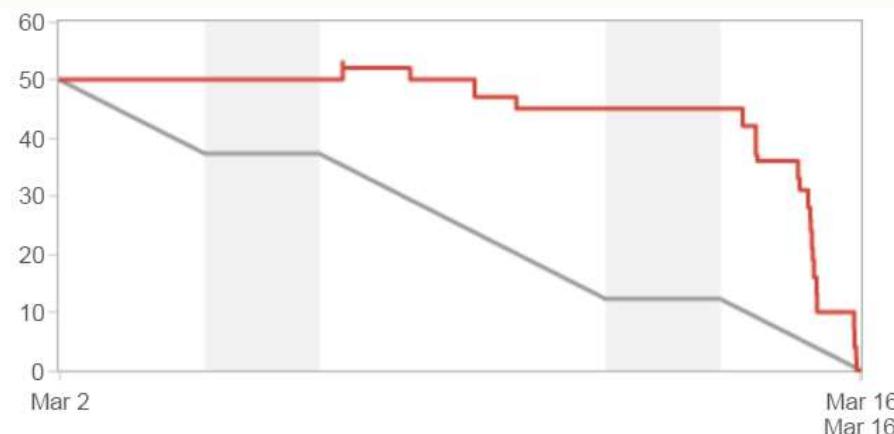
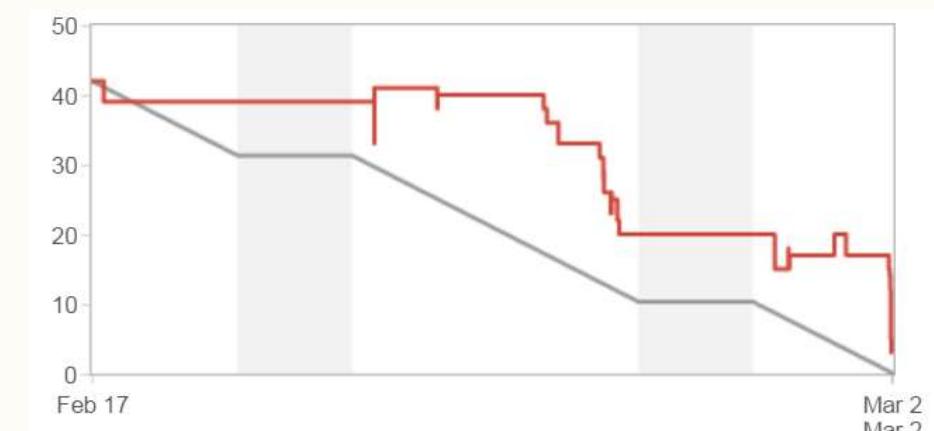


Sprint Burndown chart

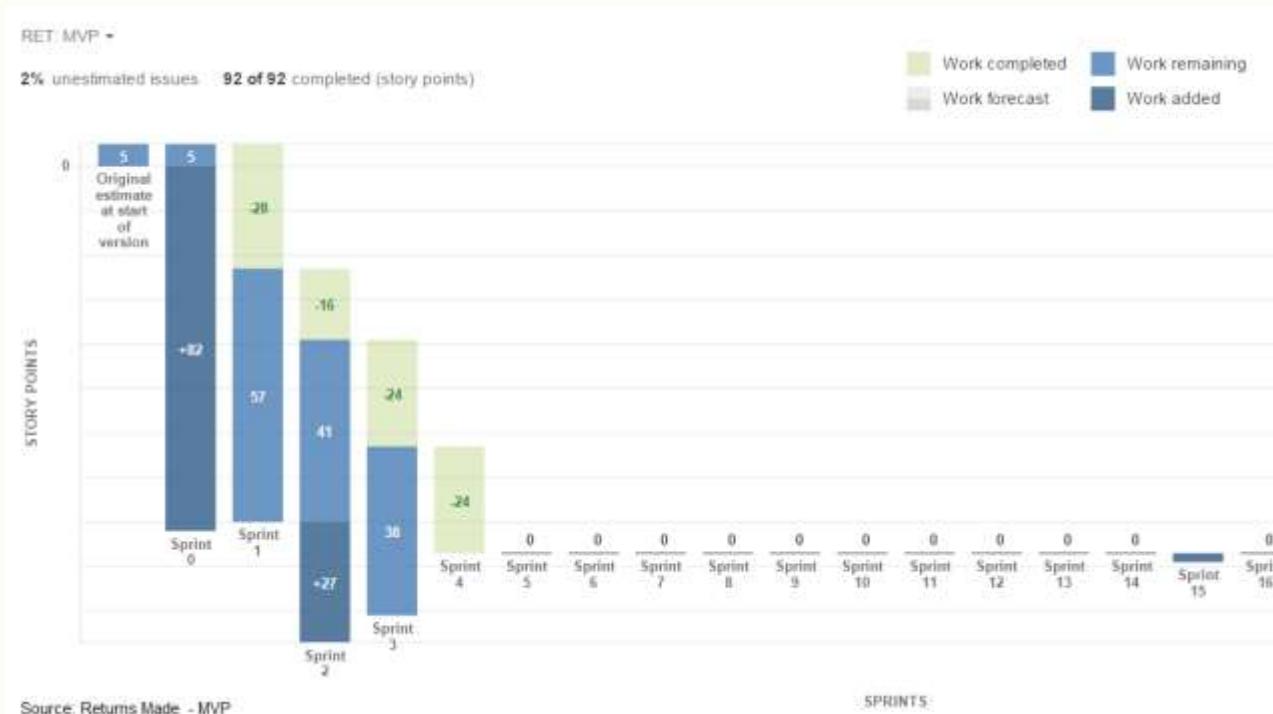
- Visualizes how much work has been done
- Visualizes how much work remains



Sprint Burndown chart

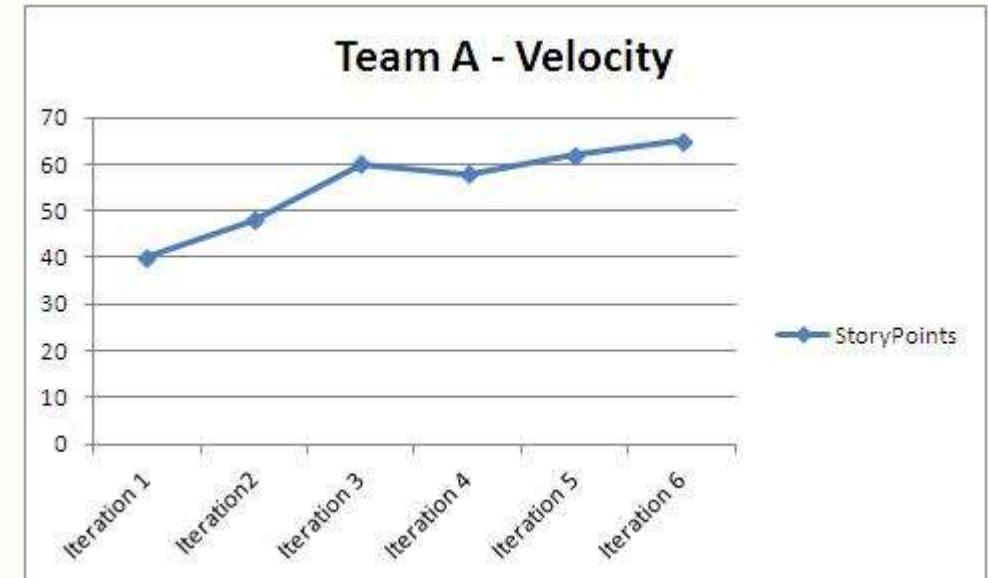


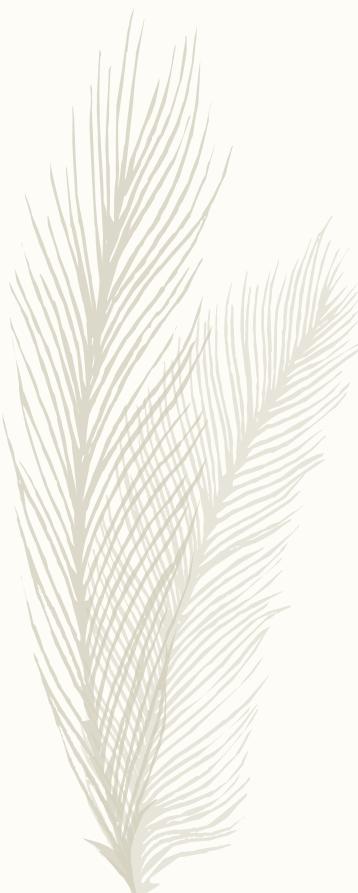
Release Burn down chart



Team Velocity

- Story points completed every iteration
- First iteration is a guess
- Presents a forecast for future Sprint





Scrum Common Mistakes

- Excessive Preparation/Planning
- Focus on tools
- Problem-Solving in the Daily Scrum
- Assigning Tasks
- Scrum Master As Contributor
- Product Owner Doesn't Show
- Team Organizes Product Backlog
- Product Owner Specifies Solutions
- Urgent Interruptions
- Making Assumptions
- Non-Scrum Roles on the Scrum Team (specialized roles)
- Teams are often not able to complete the stories in the time box and adjust the Sprint each time
- Frequently Stories are kicked off the sprint due to fire drills as team members get pulled into other activities
- Retrospectives are not actionable and there is a lack of continuous improvement culture

Scrum Statistics

Actual Improvements from Implementing Agile

The top three benefits of adopting agile have remained steady for the past five years: manage changing priorities (87%), team productivity (85%), and project visibility (84%).



*Respondents were able to make multiple selections.

KANBAN





AGENDA – Kanban

- Lean
- Adoption
- Principles
- Practices
- Continuous Integration / Continuous Delivery
- Tools
- Statistics

Lean Introduction



Lean manufacturing - History

- Pioneered at Toyota as from late 1940's
- Toyota's market was small, they had to make a variety of vehicles on the same assembly line to satisfy its customers.
- Short lead times & flexible production lines =>higher quality, better customer responsiveness, better productivity, and better utilization of equipment and space.

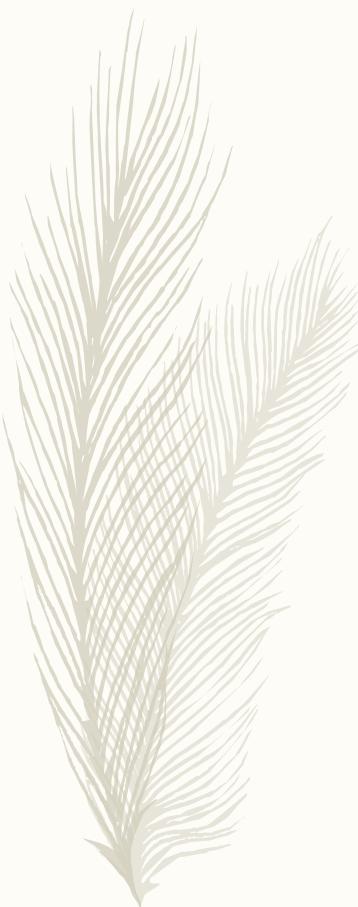
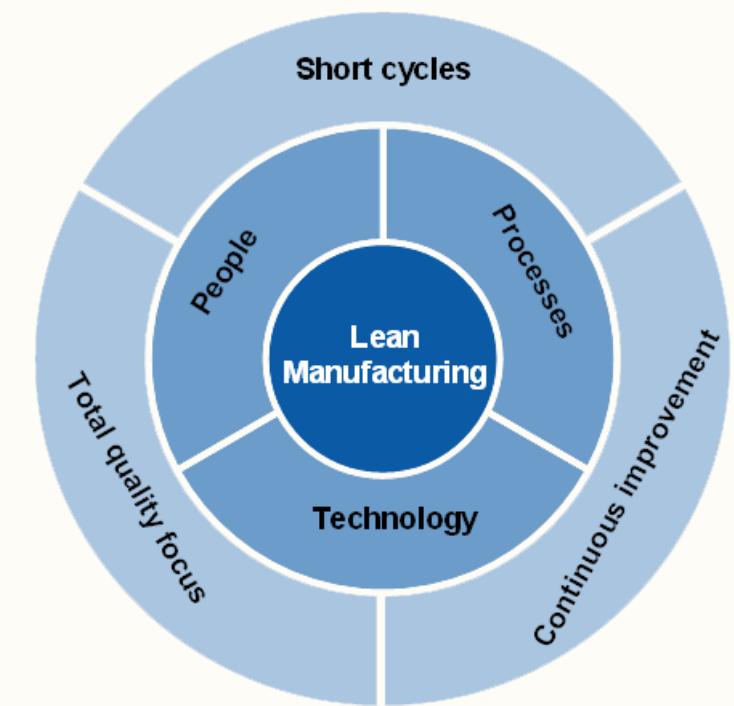
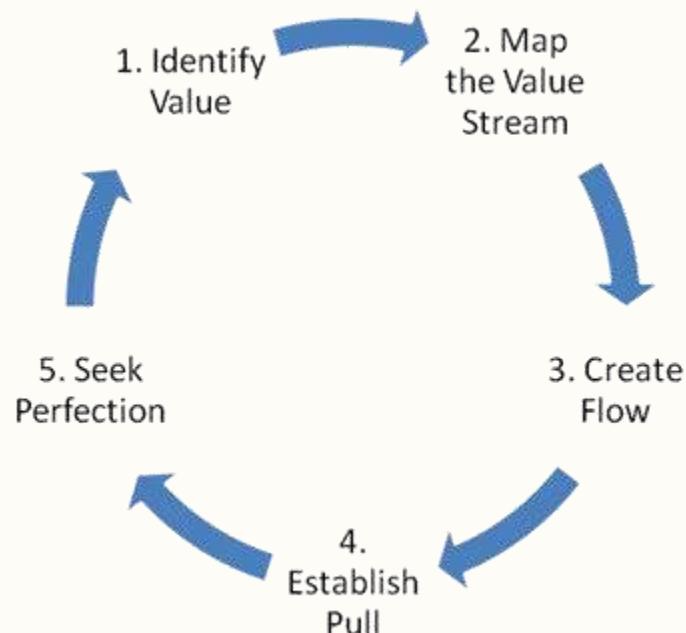


Lean manufacturing - Concept

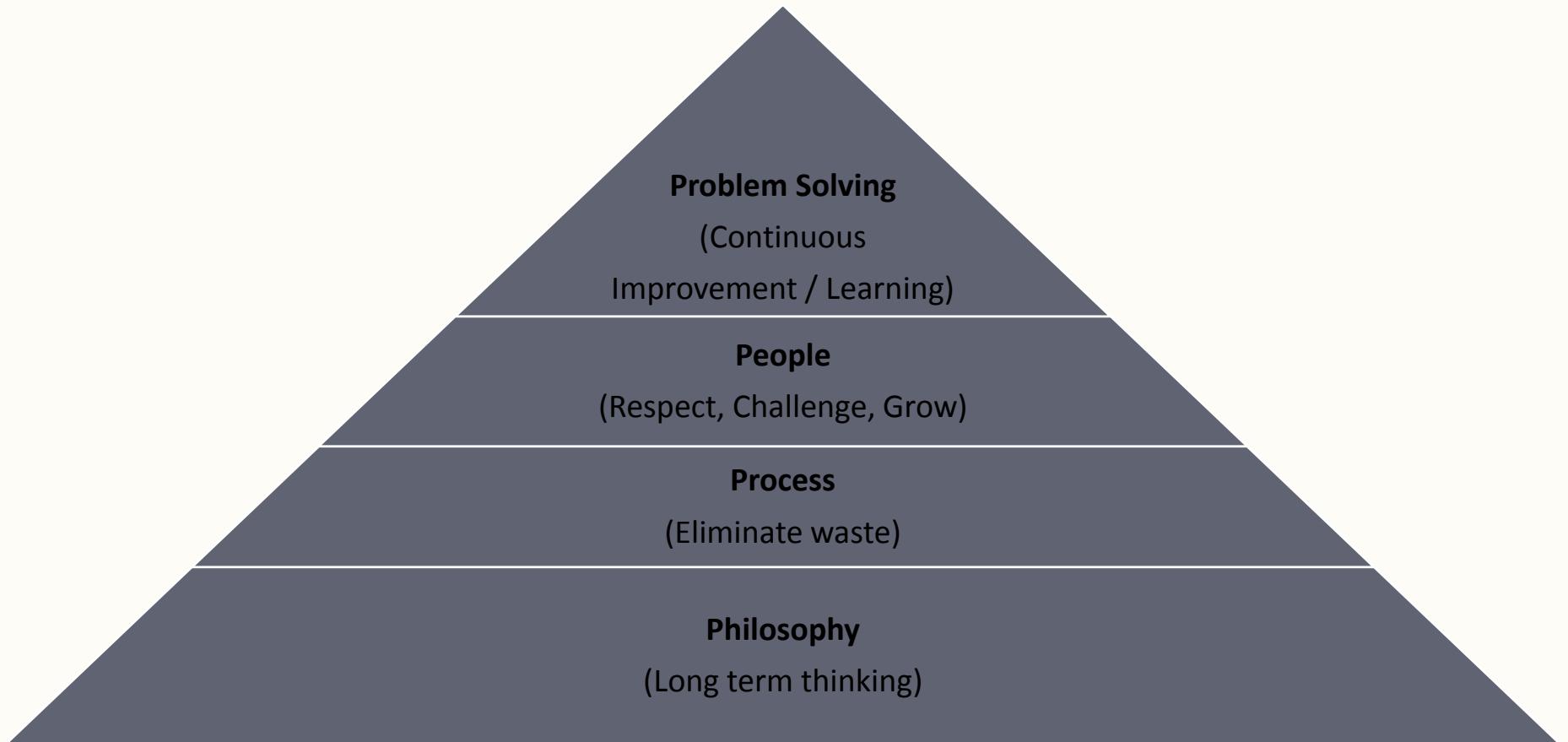
- Focus on the flow of the product through the total process
- Quick setups so each machine could make small volumes of many part numbers
- Pull between all steps where continuous flow is possible



Lean manufacturing - Principles



Continuous improvement culture



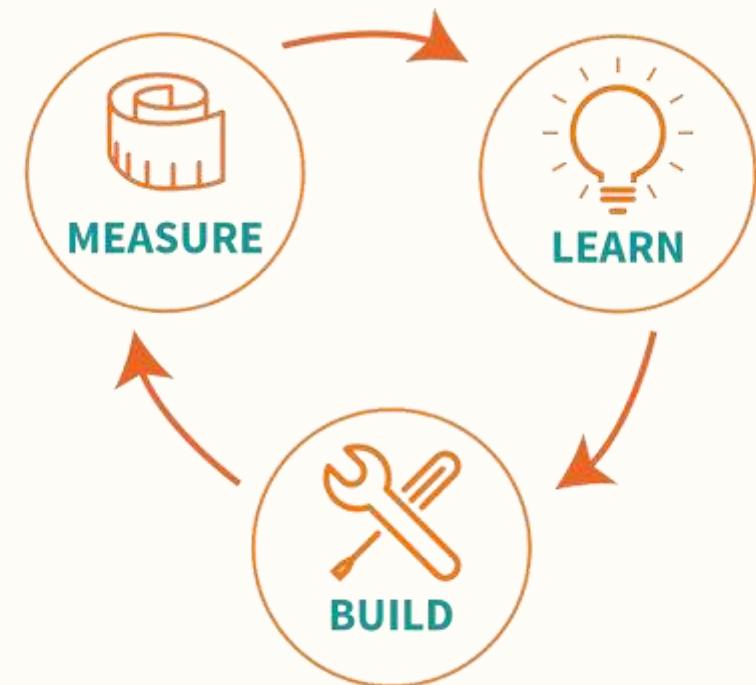
Kanban - Definition

Kanban is a **change management approach** that employs a **WIP limited pull system**



Kanban - History

- Kanban in software development emerged for knowledge work in 2005
- Kanban has roots in Lean



Kanban - Adoption



Kanban is adopted for teams where:

- Their work is hard to plan
- They need to be flexible and responsive

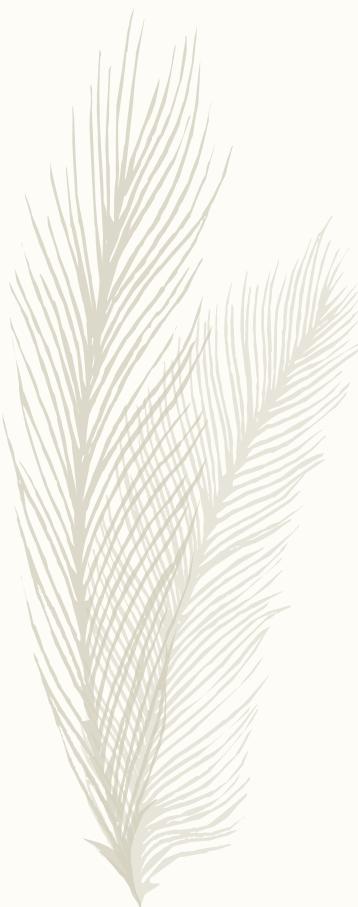
Kanban can be used in:

- Agile
- Waterfall

Kanban - Principles

- Start with what you do now
- Agree to pursue incremental, evolutionary change
- Initially, respect current roles, responsibilities & job titles





Kanban - Practices

- Visualize workflow
- Limit Work In Progress
- Manage Flow
- Make Process Policies Explicit
- Improve Collaboratively



Visualisation

Advantages:

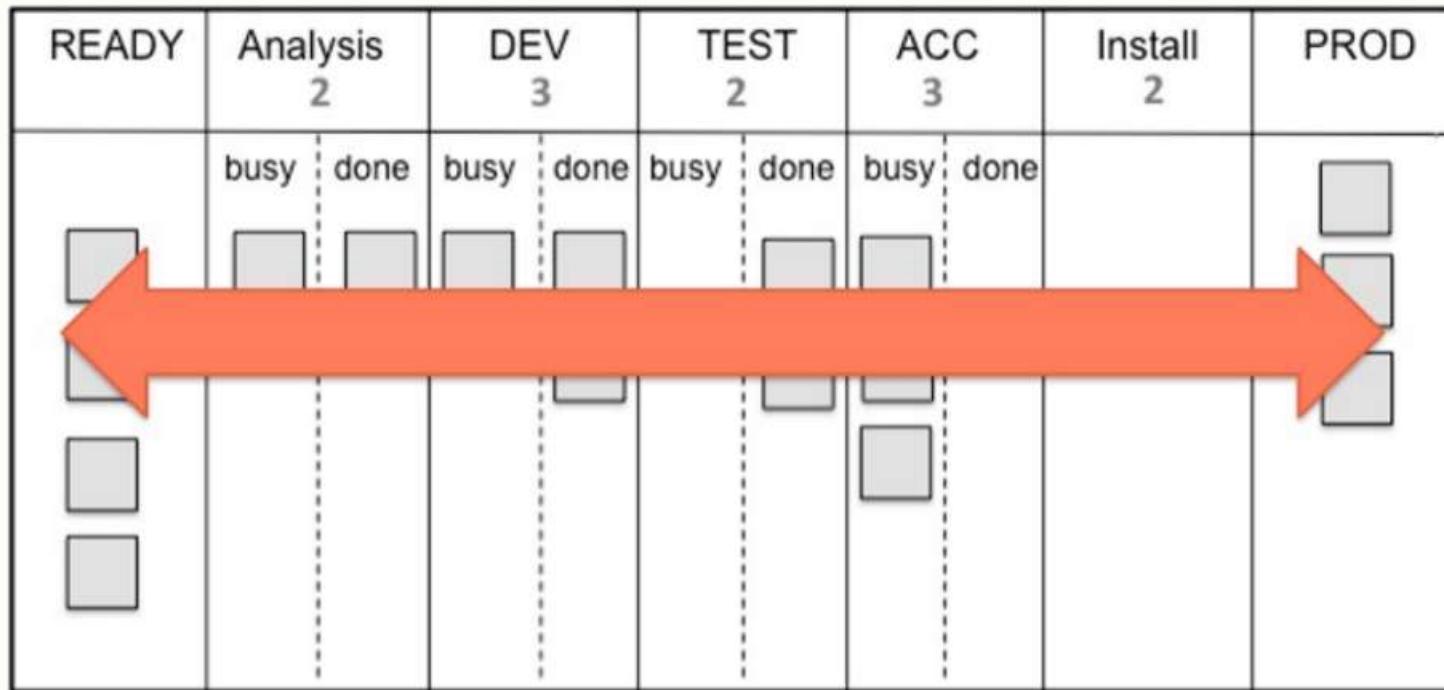
- effective collaboration
- continuous improvement

Information radiators

- Big, readable
- Always visible
- Easily maintainable
- Clean, only relevant information

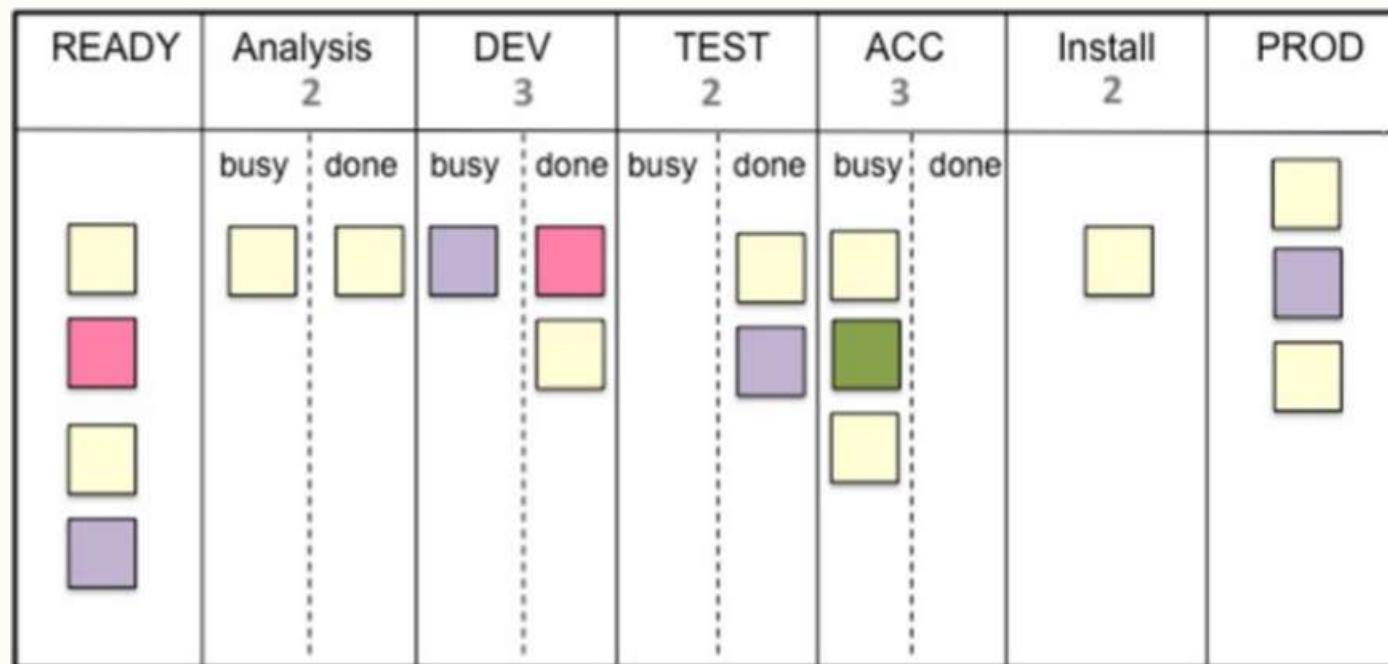
Kanban board

1) From demand until delivery



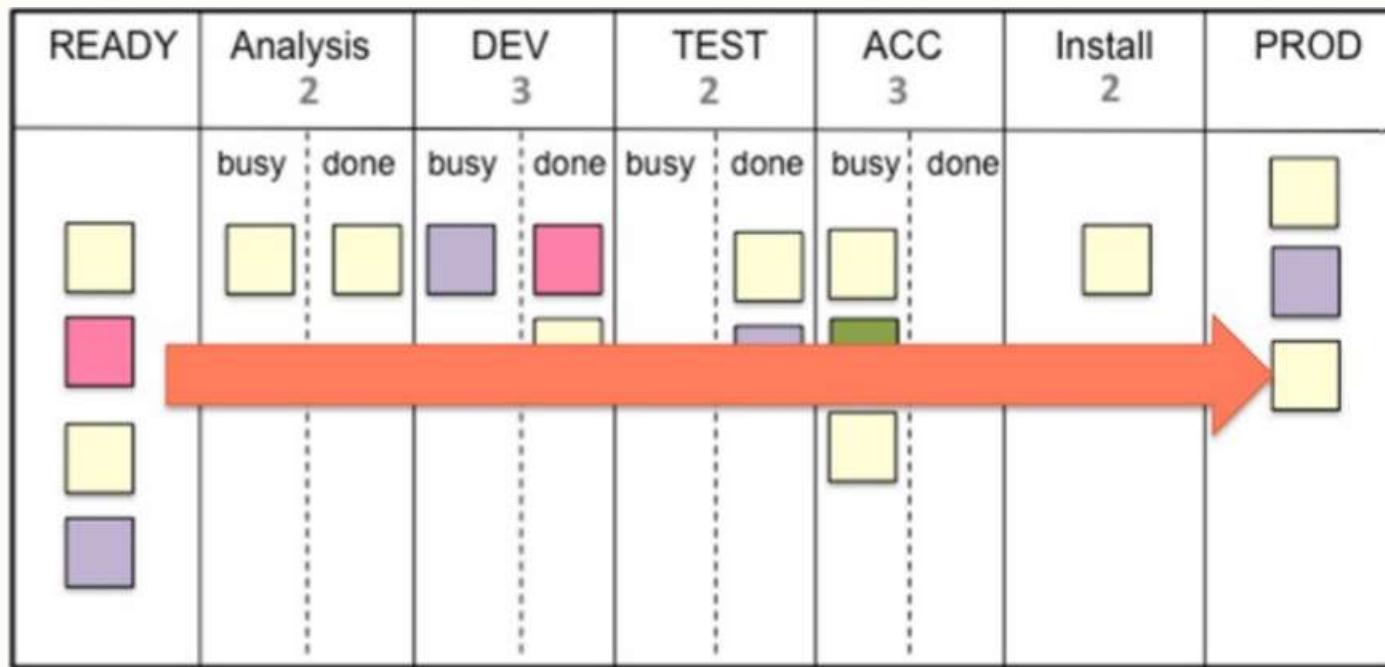
Kanban board

2) Different work item types



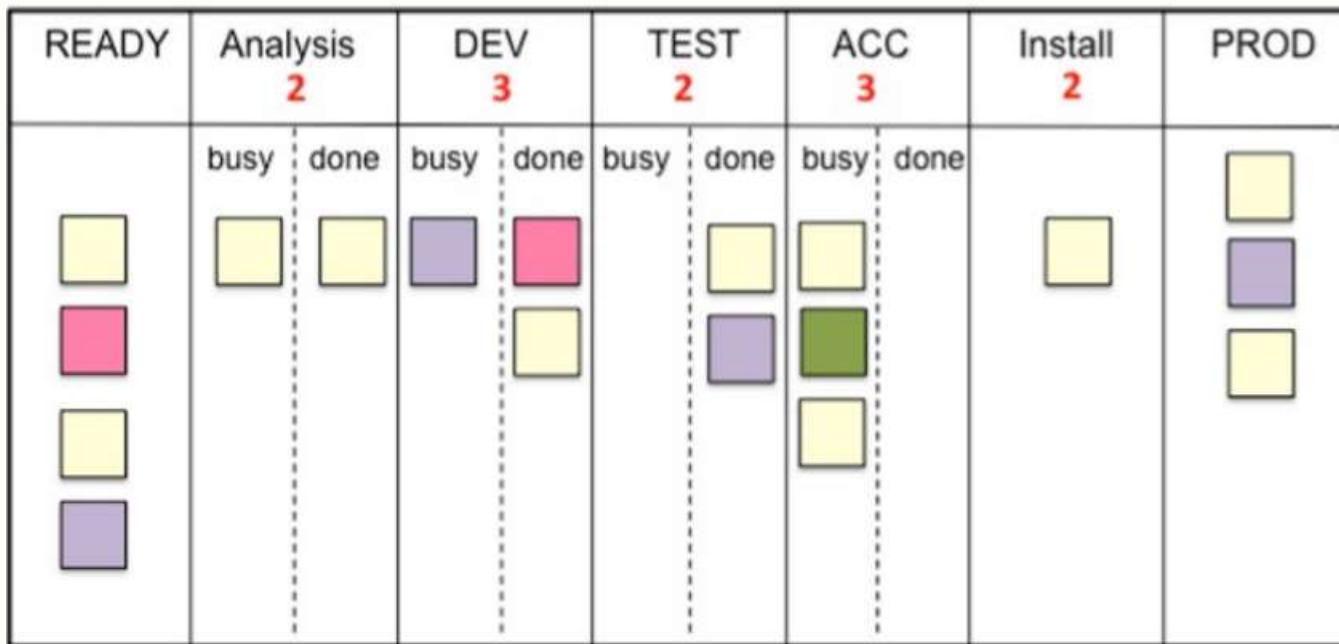
Kanban board

3) Continuous flow



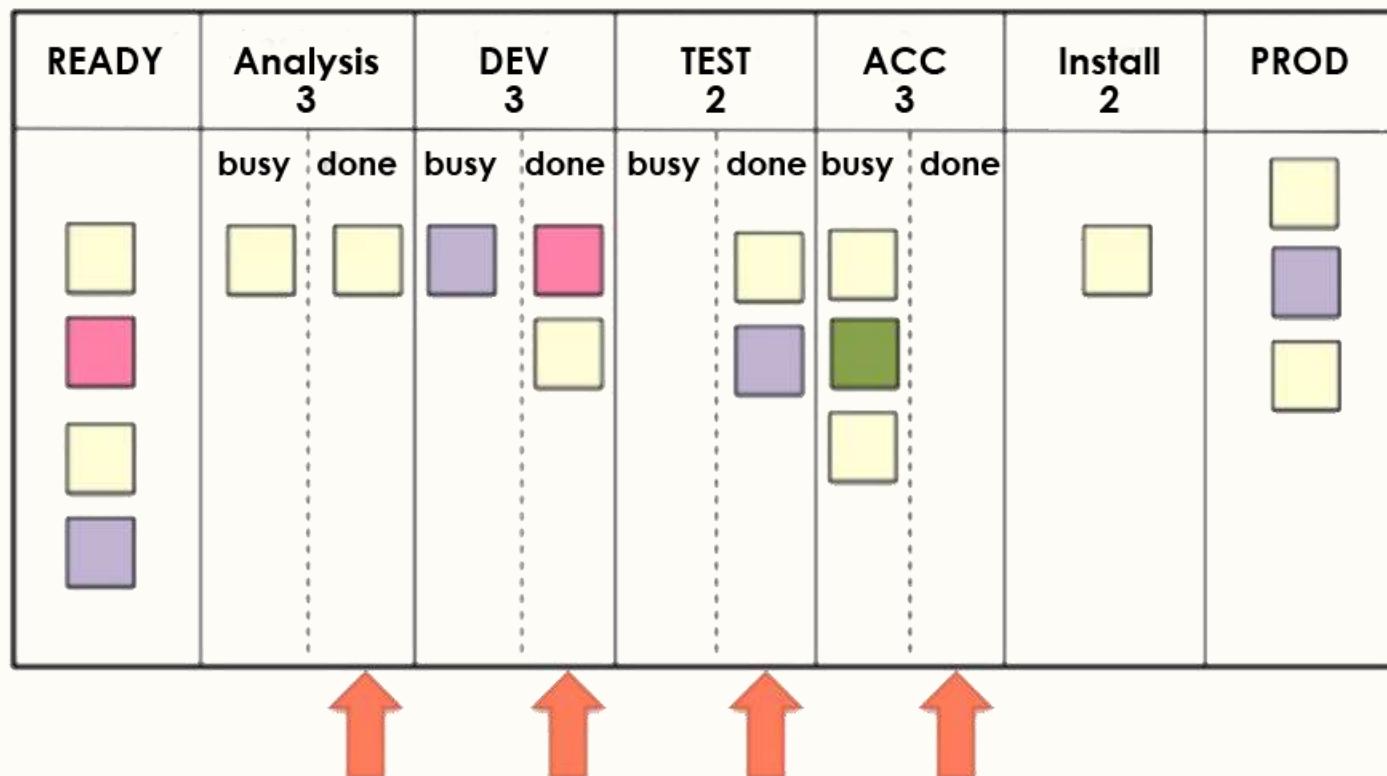
Kanban board

4) Work in Progress limits



Kanban board

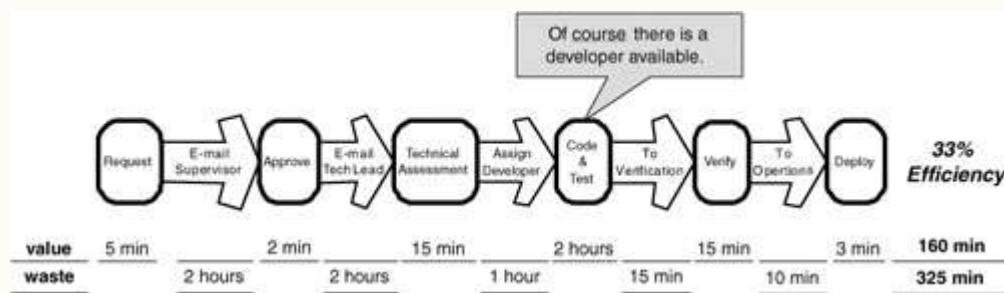
5) Queues



Kanban board

How to create a Kanban board?

- Create a value stream map



- Draw the Kanban board
- Change over time



Limit Work In Progress (WIP)

Lowering WIP = lowering cycle time

- Work items move faster through the process
- Leads to faster feedback

Increasing WIP = higher cycle time

- Risk of changing business conditions, work items decrease in value
- More context switching => lower efficiency
- More delays => extra work
- More overhead
- Lower quality

Limit Work In Progress (WIP)

Keeping everybody busy at 100% utilization rate automatically leads to **multitasking** which increase cycle time.

WIP limits prevent over-utilization and regulates a healthy flow.

Slack time is necessary to keep your organization healthy

- Time for continuous improvement
- Cleaning up
- Learn
- Experiment



Limit Work In Progress (WIP)



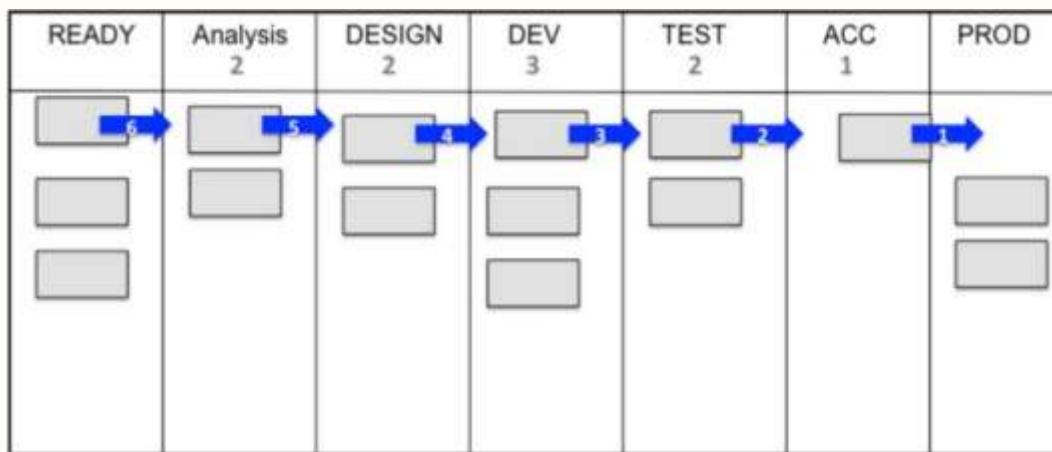
How to find **good WIP limits**?

- Lower is better
- Start comfortably - nr of people working in the workflow phase + 1
- Regularly assess and try to lower WIP limits

Limit Work In Progress (WIP)

Consequences of WIP limits

- PULL

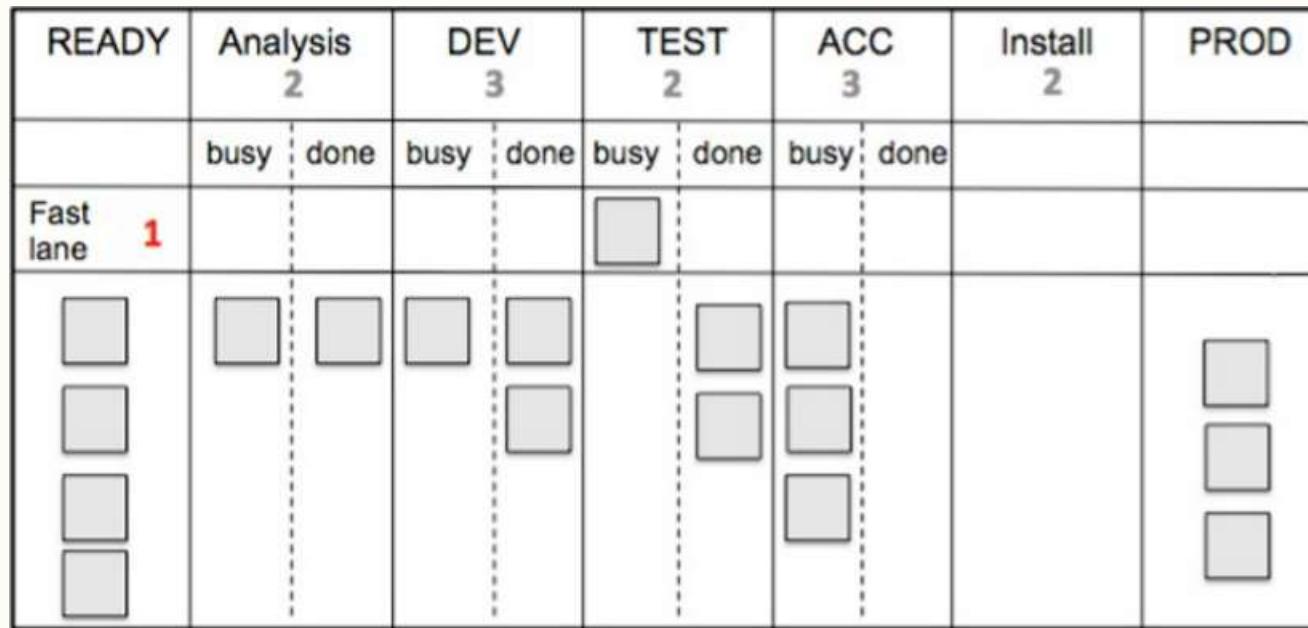


- Stop starting, start finishing
- Bottlenecks will appear
- Collaboration

Limit Work In Progress (WIP)

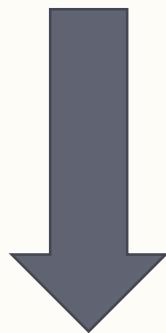
How to deal with emergencies?

Use a **Fast lane** (with WIP limit)

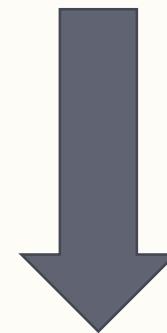


Manage Flow

Flow = Each piece of work moves from one value adding step to the next without waiting time or batch work



Just-In-Time



Pull system with WIP limits



Manage Flow

1. Actively manage **bottlenecks**

5 focusing steps of the Theory of Constraints

- (1) Identify the system's constraints.
- (2) Decide how to Exploit the system's constraints.
- (3) Subordinate everything else to the above decision.
- (4) Elevate the system's constraints.
- (5) Repeat

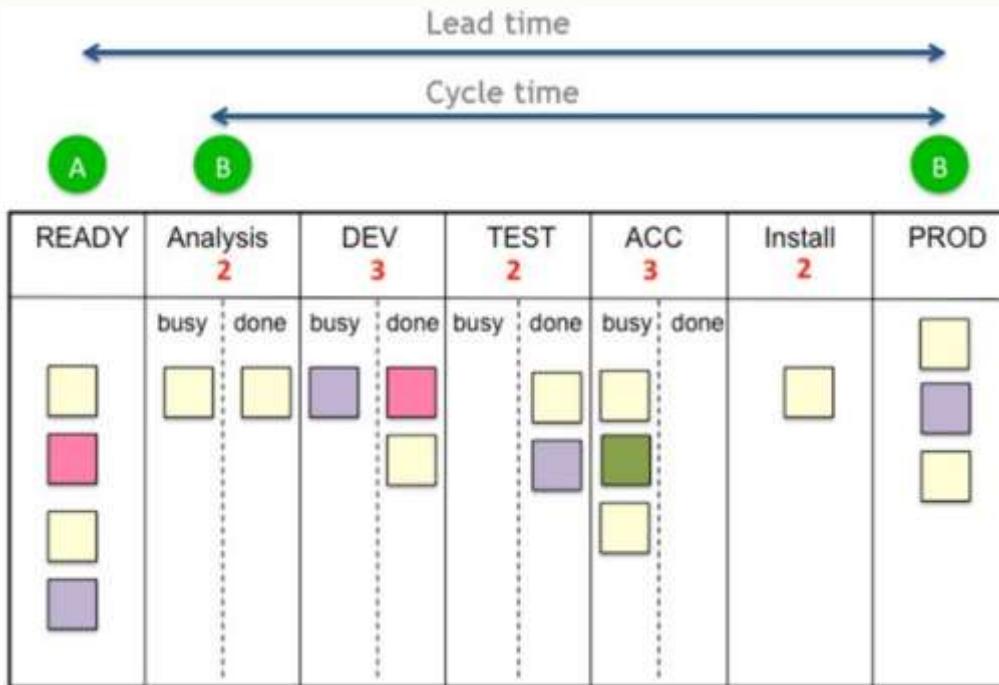


Manage Flow

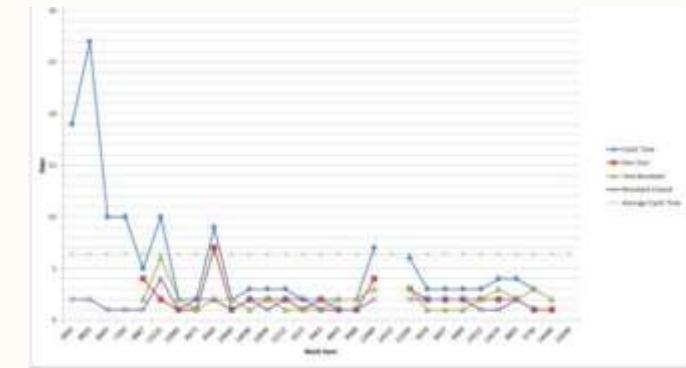
2. Eliminate **waste**
 - Rework
 - Waiting time
 - Blockers
 - Task switching
 - Knowledge isolation

Manage Flow

3. Measure lead & cycle time of each work item.



Cumulative Flow Diagram



Cycle time distribution

Continuous Improvement

- Daily standup meeting
- Operations review (process control chart & lead and cycle measurement)
- Coaching
- Retrospective



Continuous Delivery



In a pull system each feature moves individually through the system and CAN be released individually.

BUT sometimes it makes sense to release in group (versions).

Tools

- Electronic Kanban tools – ex. Leankit Kanban
- JIRA
- Trello



Kanban Statistics

