

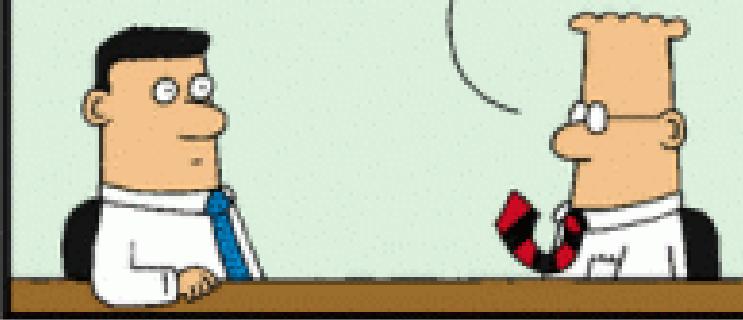
JIT

MY COMPANY IS
MOVING TO A "JUST
IN TIME" INVENTORY
STRATEGY. YOU'LL
DELIVER WHEN WE
NEED IT.



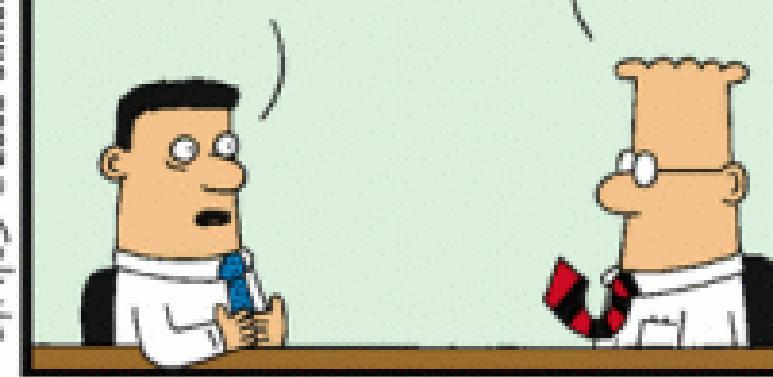
www.dilbert.com scottadams@sol.com

SO...YOUR SUCCESS
DEPENDS ON MY
COMPANY DOING
WHAT IT PROMISES?
YOU HAVE MY DEEPEST
SYMPATHY.



3/14/03 © 2003 United Feature Syndicate, Inc.

I FEEL
A SHARP,
STABBING
PAIN IN
MY CHEST.
AND
SO IT
BEGINS.

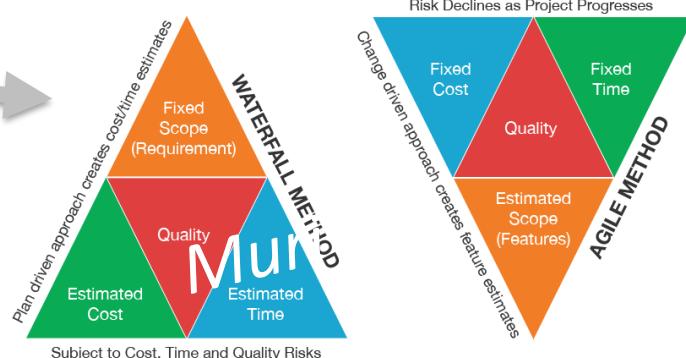


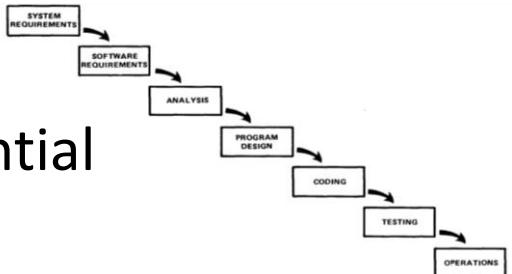
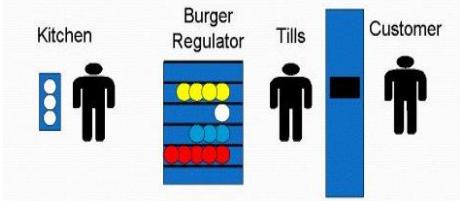
Goals



RECAP

- Initiation
 - PMBOK
 - PRINCE2
- Planning
 - Scope (WBS)
 - Time (Gantt, CPM, PERT)
 - Budget (EVA)
 - Performance (KPIs)



- Execution
 - **Traditional:** Planning essential (Waterfall)
 - 
 - **Lean:** eliminate waste, pull system, maximise value to customer (e.g. MVP)
 - 
 - 
 - 
 - **Agile:** embrace change, incremental delivery, collaboration, trust (e.g. Scrum)
 - 



This session is being recorded.



WARWICK

CS352 Project Management for Computer Scientists

7. Lean and Agile

Part 2



interact at:

warwick.ac.uk/pm4cs/7

Dr Ian Saunders

Scrum Quiz!



warwick.ac.uk/pm4cs/7



Who is responsible to measure the Project's performance?

- 1 The Scrum Master
- 2 The Project Manager
- 3 The Product Owner
- 4 Anyone on the team



The Product Backlog should be ordered on the basis of?

- 1 Size of the items being delivered
- 2 Value of the items being delivered
- 3 The complexity of the items being delivered
- 4 The risk associated with the items



What are the main responsibilities of a Scrum Master?

Vote for up to 3 choices

- 1 Manging the team to ensure progress stays on track
- 2 Facilitating meetings as and when requested
- 3 Help the Product Owner order the Product Backlog
- 4 Consulting the Development Team and Product Owner
- 5 Bridging the Gap between the Team and the Customer
- 6 Removing Impediments



What is the significance of determining a Definition of Done?

Vote for up to 3 choices

- 1 Determines the objective behind each sprint
- 2 Determines the number of tasks being completed for each Sprint Backlog item
- 3 Increases Transparency
- 4 Increments delivered are more effective and potentially releasable
- 5 Develops a common understanding amongst all the team members



What is done during a Sprint Review Meeting?

Vote for up to 2 choices

- 1 Demo of the Increment
- 2 The team discusses the improvements that can be applied for the upcoming sprints
- 3 Present the project's performance to the Stakeholders
- 4 Inspect progress towards the Sprint Goal
- 5 Discuss the architectural and technical aspects of the project



Leaderboard

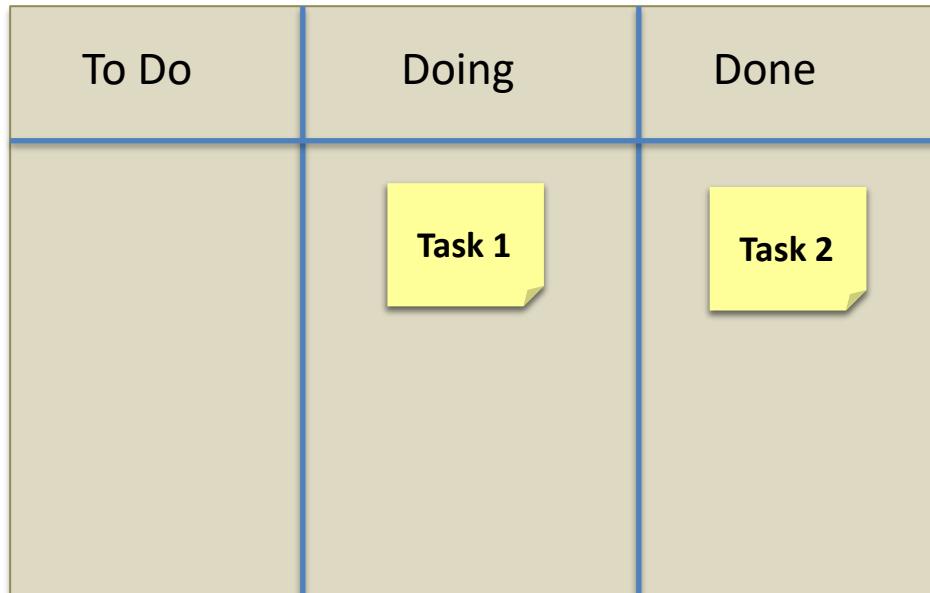


Position	Participants	Score
1	John Doe, Jane Smith	100
2	David Johnson, Emily Davis	95
3	Michael Brown, Sarah Green	90
4	Christopher White, Natalie Blue	85
5	Robert Black, Lucy Grey	80

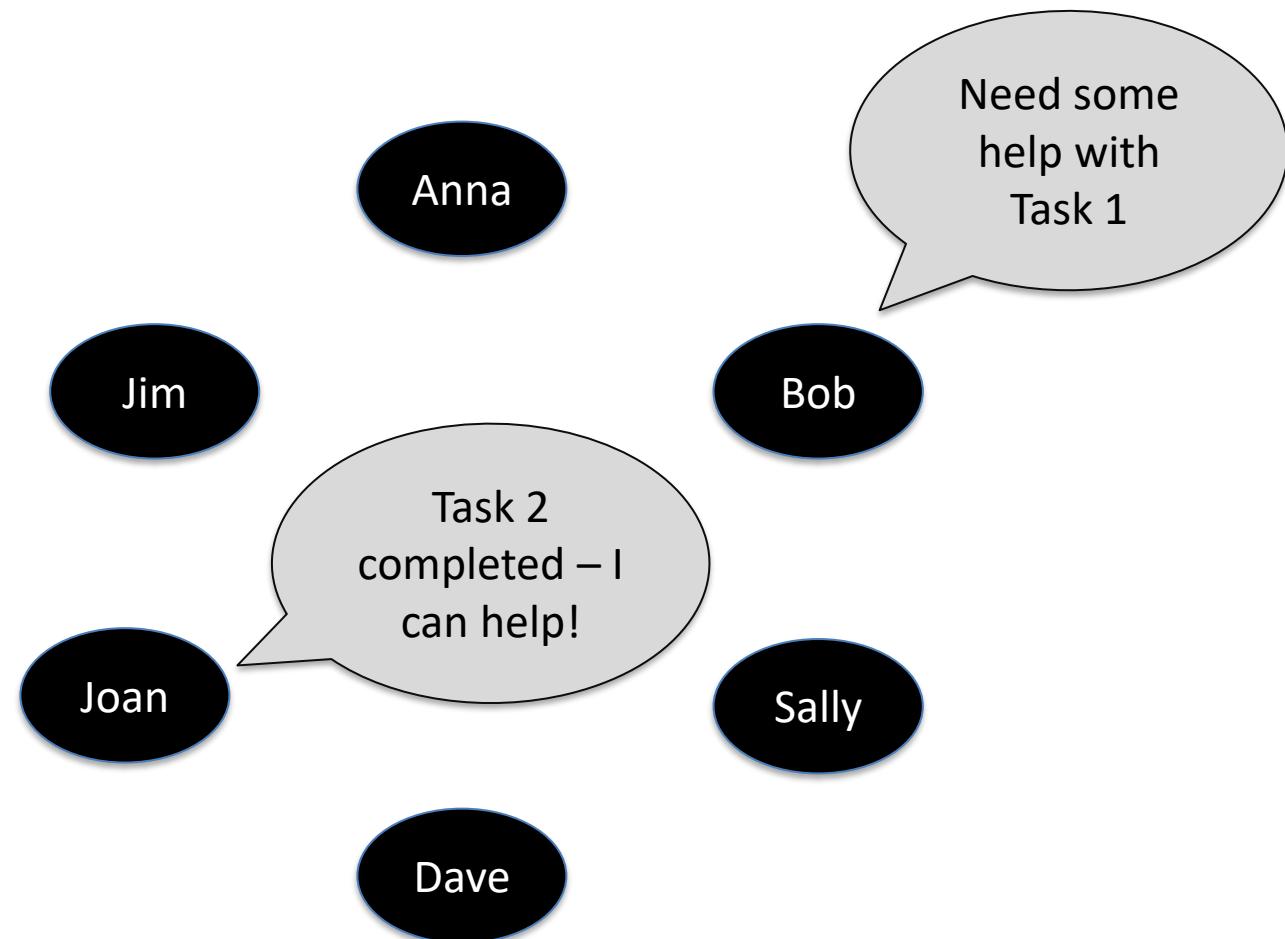
Total Participants: 0

Average Score: 0

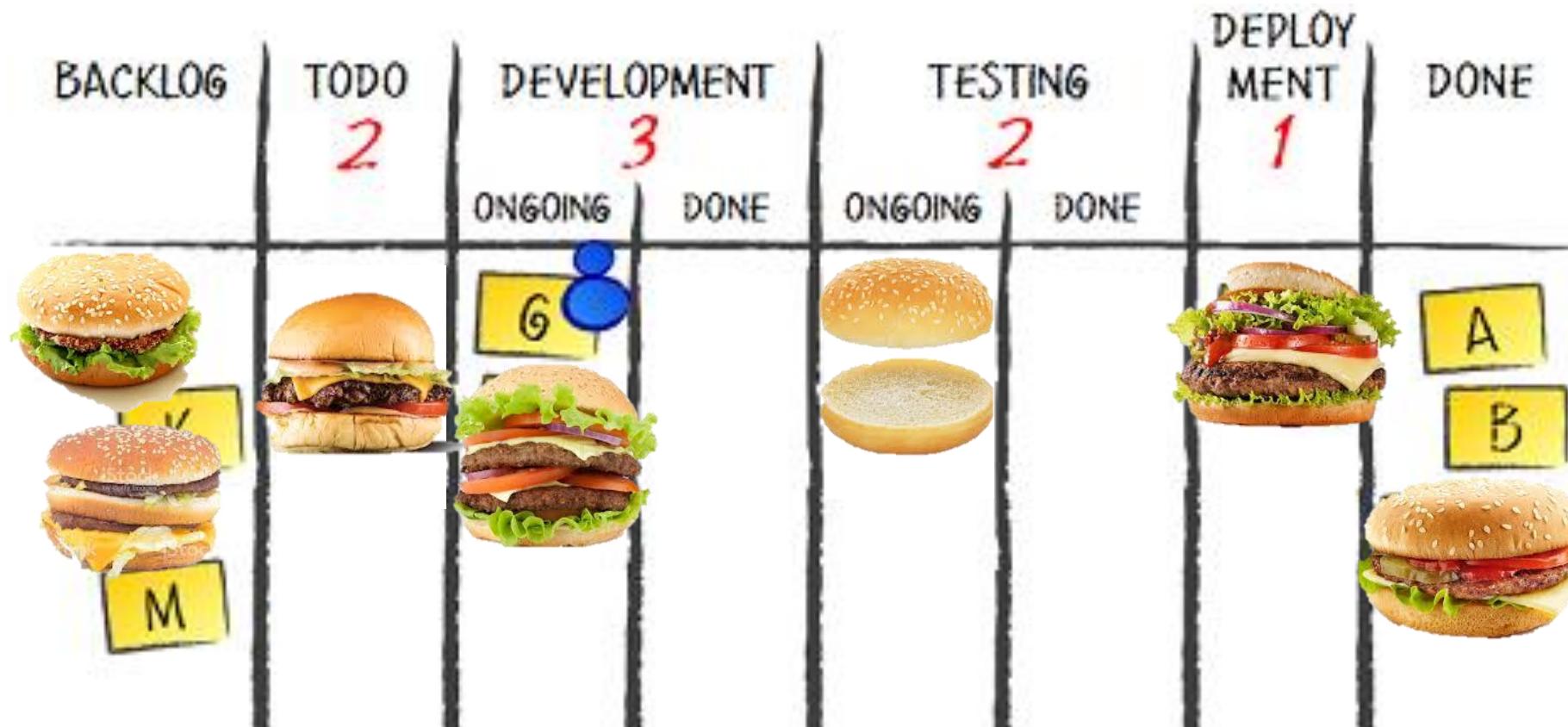
Scrum / Daily Stand Up



A Kanban Board!



Kanban Board



Today

Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles
Blockers and Idleness
Kanban Examples



Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile



Part 2: Guest Lecture

Toyota Lean “House”

Goal: Highest Quality, Lowest Cost, Shortest Lead Time

Just-in-Time

Continuous Flow
Takt Time
Pull System

Jidoka

Stop and notify
of abnormalities
Separate human
work and
machine work

Heijunka

Standardized
Work

Kaizen

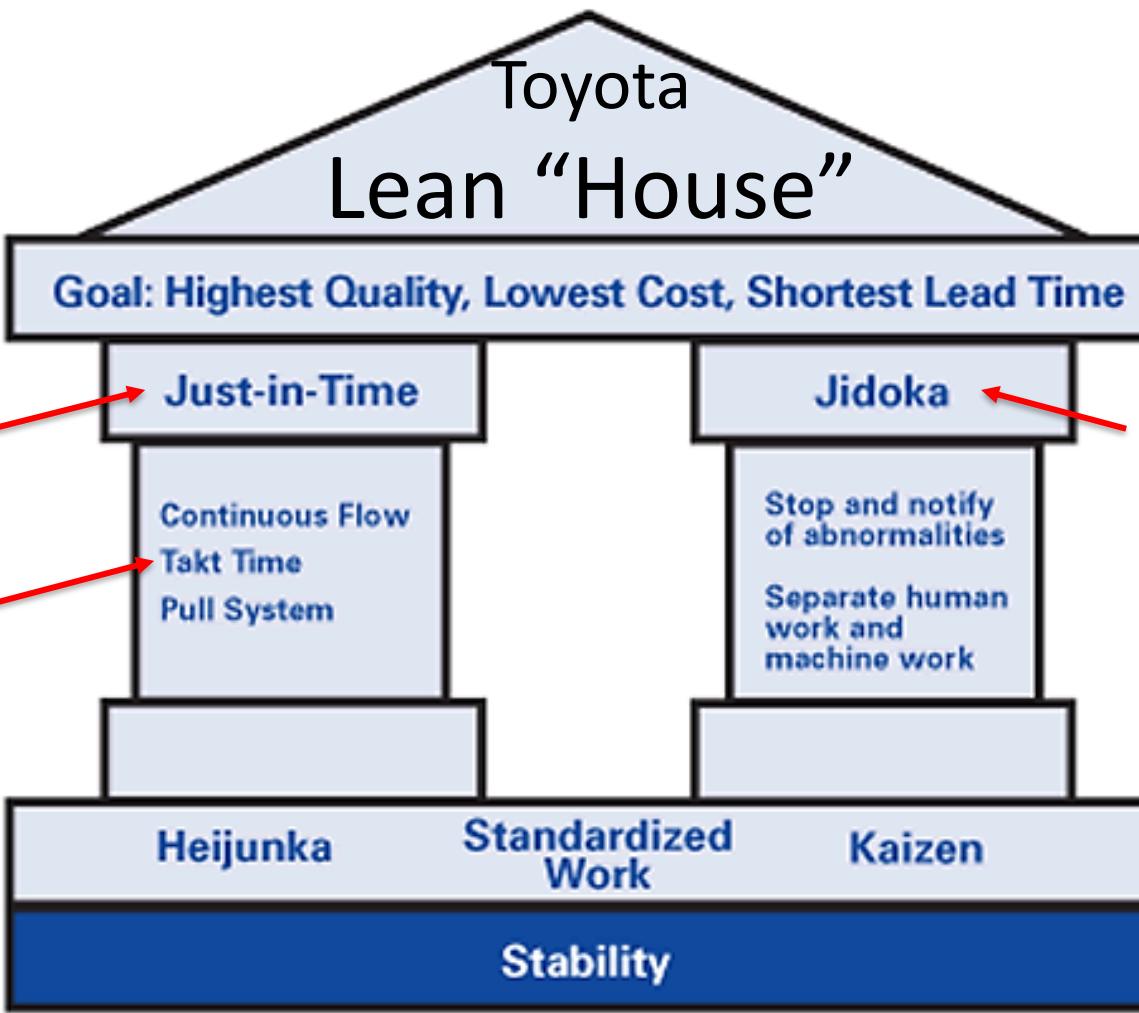
Stability

Toyota Lean “House”

Goal: Highest Quality, Lowest Cost, Shortest Lead Time

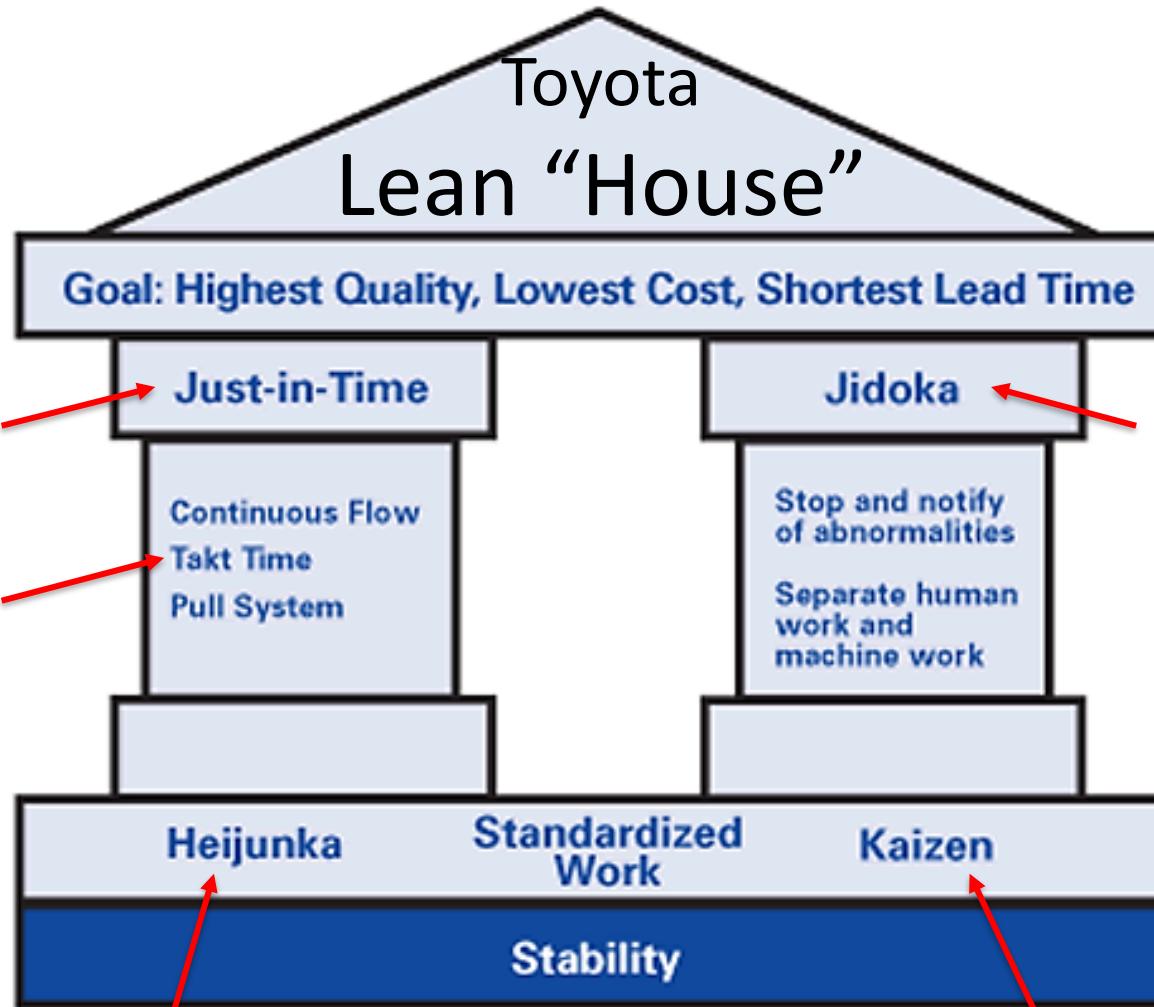
Just In Time

- Keep value flowing
- Meet rate of demand
 - “Pull” not “Plan”



Intelligent Automation

- Stop on faults
- Automate as much as possible
- Intelligent intervention



Just In Time

- Keep value flowing
- Meet rate of demand
 - “Pull” not “Plan”

Production levelling
(Small Batches)
Reduces Mura (unevenness)

Intelligent Automation

- Stop on faults
- Automate as much as possible
- Intelligent intervention

Continuous Improvement
Reduces Muri, Muda & Mura

PUSH

Just-In-Case

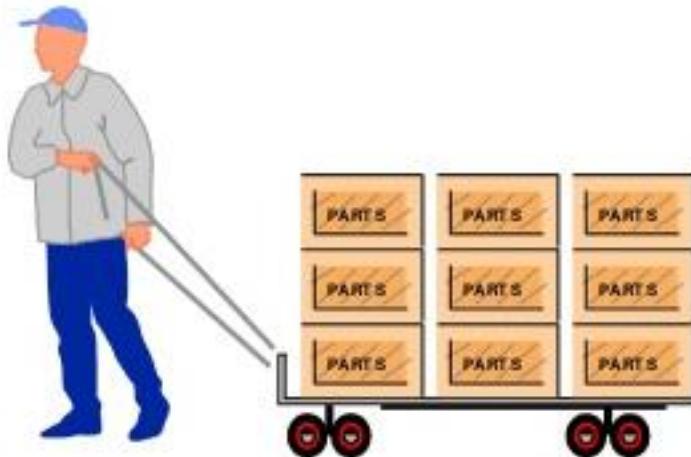
Make all we can
just in case.



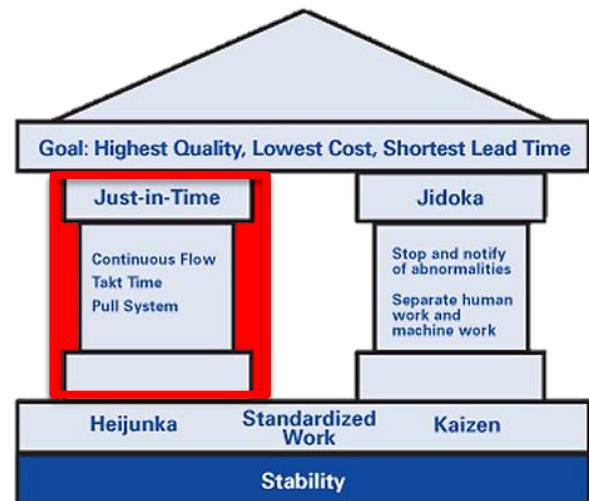
- Production Approximation
- Anticipated Usage's
- Large Lots
- High Inventories
- Waste
- Management by Firefighting
- Poor Communication

Just-In-Time

Make what's needed
when we need it



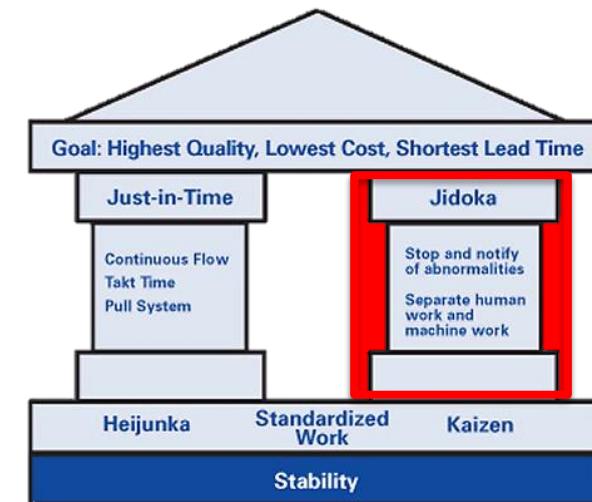
- Production Precision
- Actual Consumption
- Small Lots
- Low Inventories
- Waste Reduction
- Management by Sight
- Better Communication



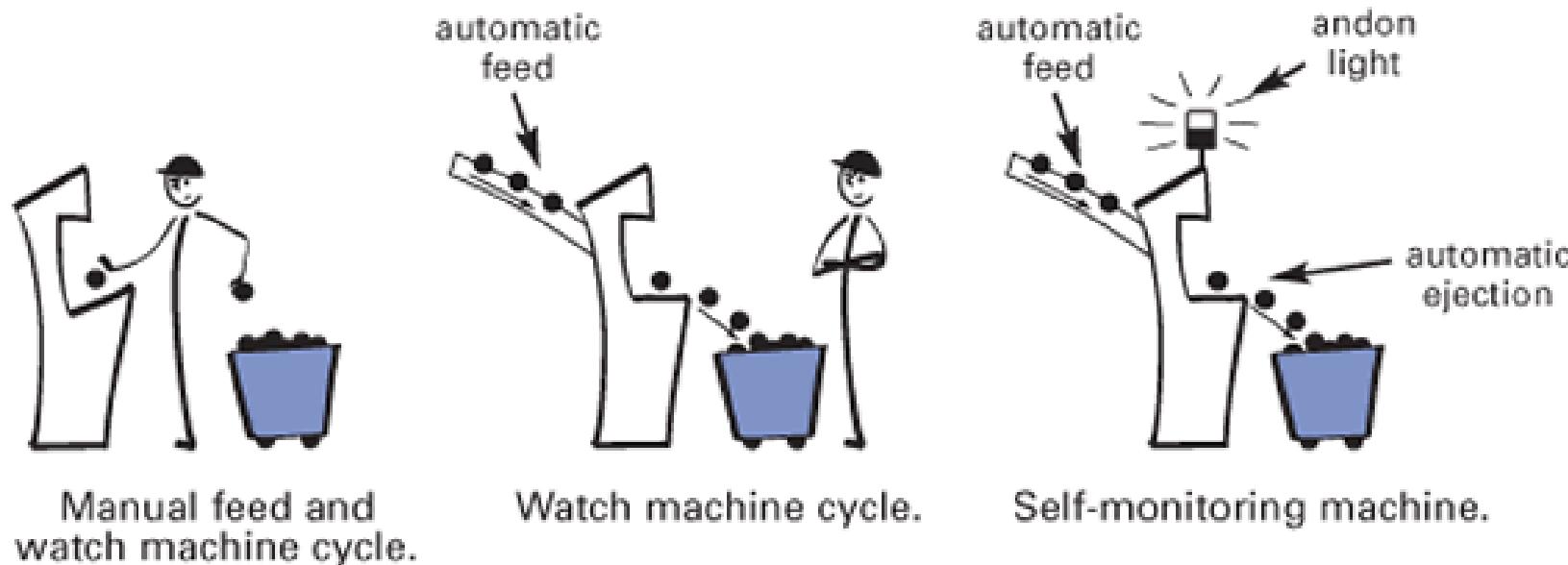
PULL

Jidoka

Intelligent Automation



The Evolution toward Jidoka



Automation

+

the ability to **self-monitor** and **stop** production in case of a problem.

Today

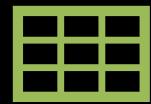
Part 1: Lecture



Lean

Toyota's "Lean House"

**Lean Thinking:
maximising flow of value**



Kanban

Kanban Software Development Principles
Blockers and Idleness
Kanban Examples



Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile



Part 2: Guest Lecture



5 Pillars of Lean Thinking

1. Identify Value

- Value is created by the producer
- Value can only be defined by the ultimate customer

2. Map The Value Stream

- The complete life-cycle of a product: from raw materials to customer use
- Some steps create value unambiguously
- Some create no value but currently unavoidable (Type 1 Muda)
- Others should be eliminated immediately (Type 2 Muda)

3. Create Flow

- Value stream keeps moving forwards, with each step in sync

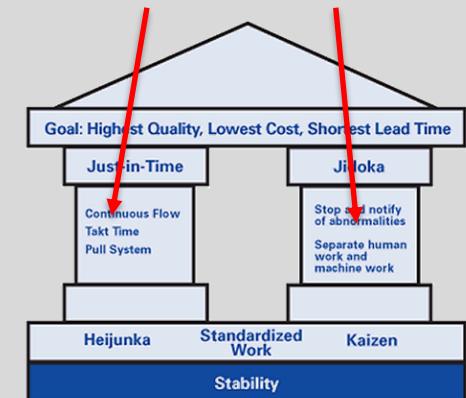
4. Establish Pull

- Don't make things ahead of time, make them just in time

5. Seek Perfection

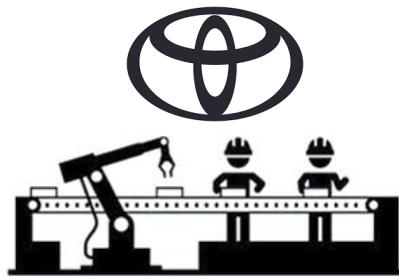
- Repeat, making continuous improvement, seeking perfection

Not to be confused with these pillars





Pillars vs Principles



5 Lean Thinking Pillars

Womack and Jones (1995)

1. Identify **Value**
2. Map **The Value Stream**
3. Create **Flow**
4. Establish **Pull**
5. Seek **Perfection**

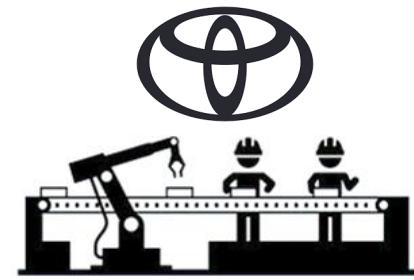
7 Lean Manufacturing Principles

Toyota (2001)

(from last week)

1. Eliminate waste
2. Amplify learning
3. Defer Commitment
4. Deliver Fast
5. Empower the team
6. Build Integrity in
7. See the whole

Pillars vs Principles



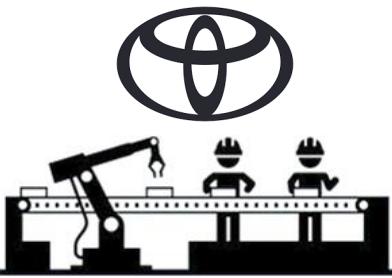
7 Lean Manufacturing Principles

Toyota (2001)

1. Eliminate waste
 2. Amplify learning
 3. Defer Commitment
 4. Deliver Fast
 5. Empower the team
 6. Build Integrity in
(mistakes disrupt flow)
 7. See the whole



Increase flow of value



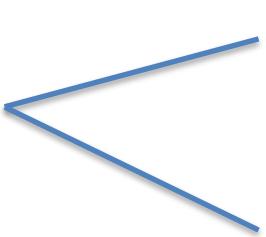
Pillars vs Principles

7 Lean Manufacturing Principles

Toyota (2001)



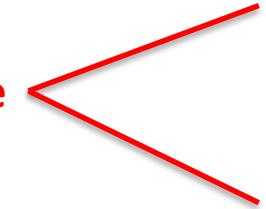
Reduce wastefulness
(increase **value**)



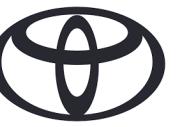
1. Eliminate waste
2. Amplify learning
3. Defer Commitment

(JIT)

Increase **flow of value**



4. Deliver Fast
5. Empower the team
6. Build Integrity in
7. See the whole



Pillars vs Principles

7 Lean Manufacturing Principles

Toyota (2001)

Reduce wastefulness
(increase **value**)

Increase **flow of value**

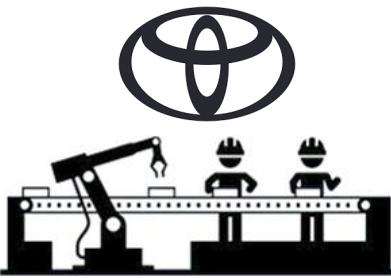
1. Eliminate waste
2. Amplify learning
3. Defer Commitment
4. Deliver Fast
5. Empower the team
6. Build Integrity in
7. See the whole

Continuous improvement
(to seek perfection)





Pillars vs Principles



Lean Thinking

Womack and Jones (1995)

1. Identify **Value**
2. Map **The Value Stream**
3. Create **Flow**
4. Establish **Pull**
5. Seek **Perfection**



7 Lean Manufacturing Principles

Toyota (2001)

1. Eliminate waste
2. Amplify learning
3. Defer Commitment
4. Deliver Fast
5. Empower the team
6. Build Integrity in
7. See the whole

to continuously increase flow of value through system

Today

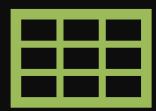
Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software Development Principles

Blockers and Idleness

Kanban Examples



Doing Agile

Conditions for Agile

Scrum vs Kanban

Agile Variants

Scaling Agile



Part 2: Guest Lecture

Kanban Software Development

“WIP-limited pull system”

- WIP
 - Work in Progress
 - Forces team/individual to only work on a limited number of **tasks**
 - reduces **overburden** and reduces overhead of **task-switching** (which reduce productivity and are therefore **wasteful**)
 - “*It’s leaner to finish all of one thing than get two things to 50%!*”
- Pull-system:
 - Start new work only when there is **demand** for it
 - No to-do list, no work overload, complete one thing then pull the next
 - Team focuses on **prioritising**, not estimating / planning
 - “*It’s leaner to focus on completing only what is required!*”

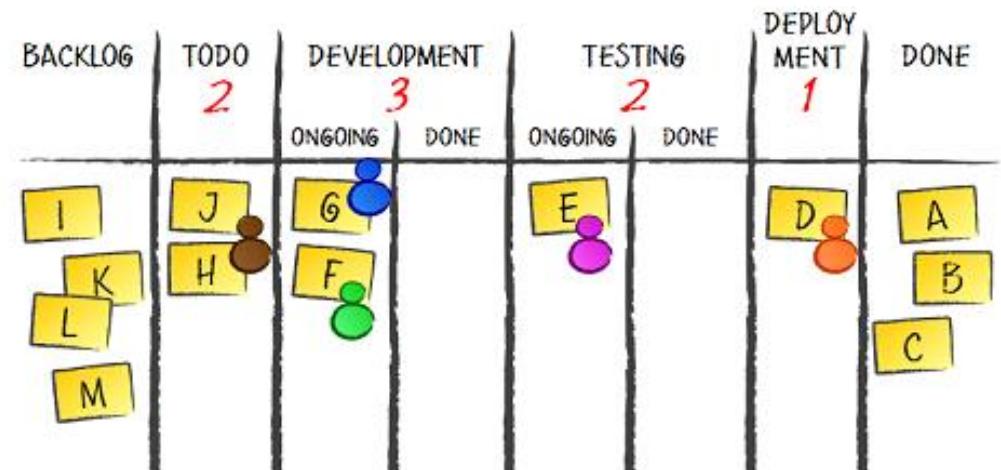
Kanban Methodology

Migration to Kanban:

1. Start with what you do now
2. Agree to pursue incremental, evolutionary change
3. Respect current roles, responsibilities & job titles
4. Encourage acts of leadership at all levels

Core Properties of Effective Kanban:

1. Visualize workflow
2. Limit Work In Progress
3. Manage Flow
4. Make Process Policies Explicit
5. Improve Collaboratively



Kanban Workflow for Software Development

Continuous Delivery

No fixed length iterations

Kanban board

- Visualise work and workflow.

Pull system

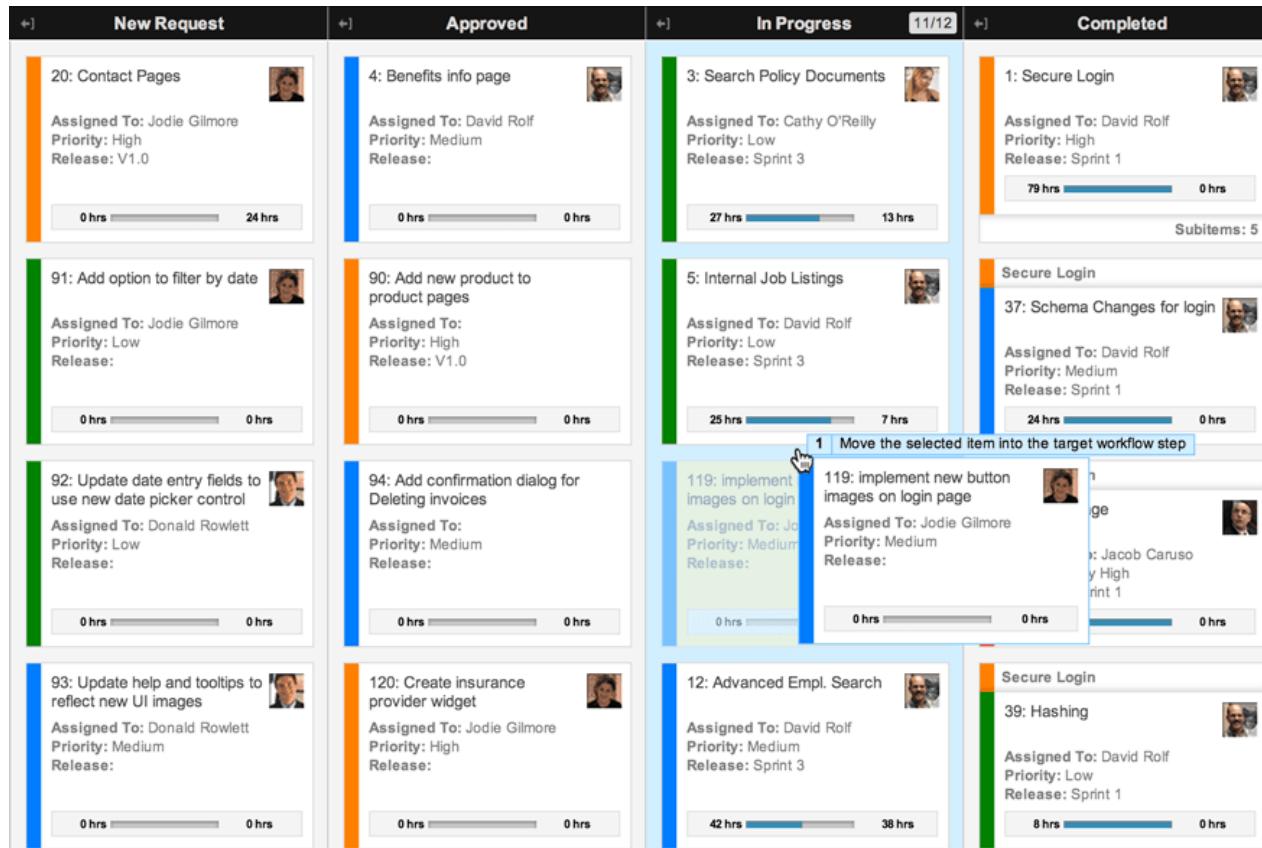
- Pull highest priority item from backlog.
- Avoids overburden.

WIP Limits

- Multitasking kills efficiency.

Flow management

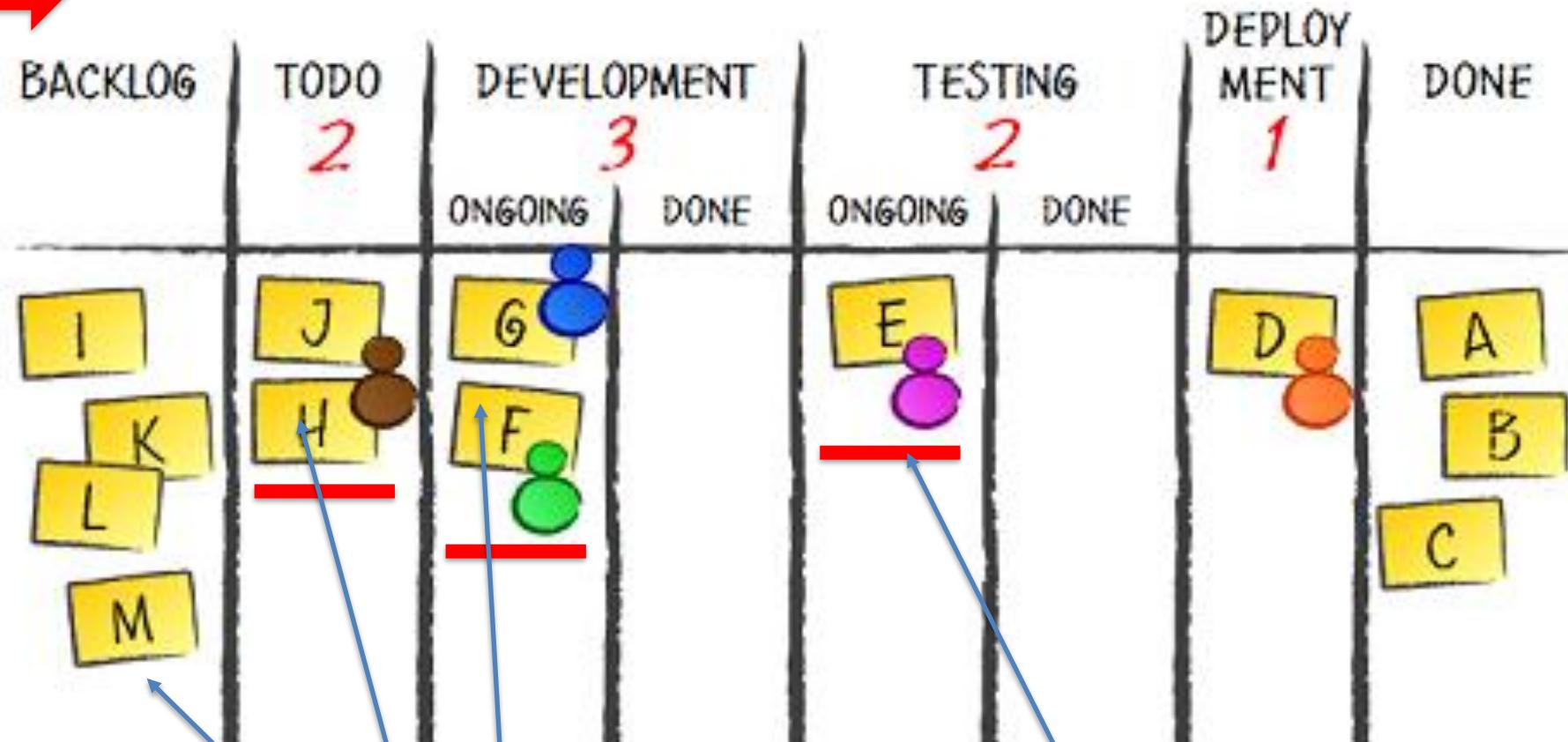
- Identify blockers and dependencies.
- Feedback and experimental evolution.



Flow Direction



Kanban Board



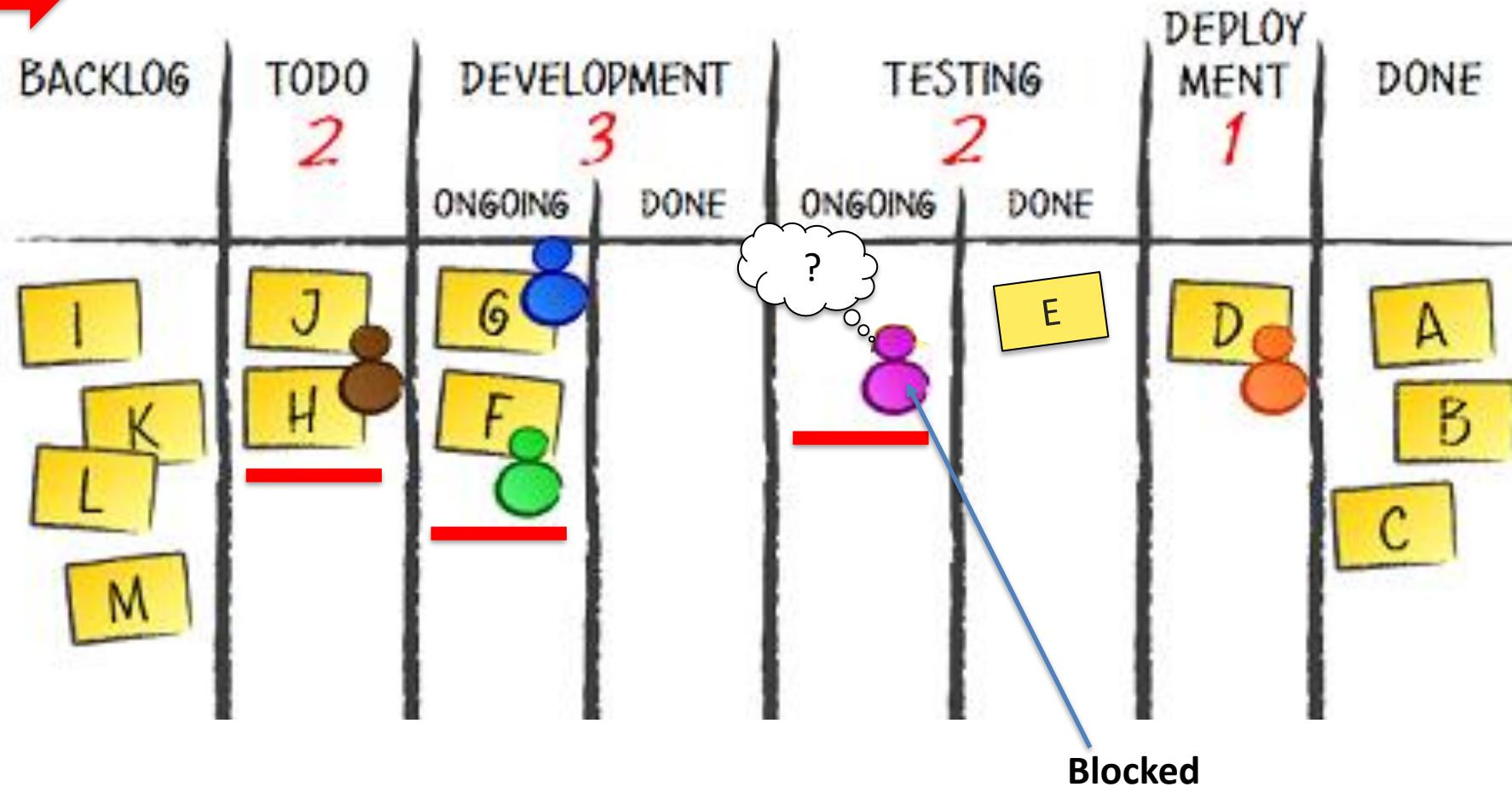
Work not PUSHED (or assigned)
It is PULLED from a backlog just in time

WIP limits
Multitasking kills efficiency
Workload is strictly limited

Flow Direction



Kanban Board



Q: How is this maximising the flow of value?

Today

Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles

Blockers and Idleness

Kanban Examples



Doing Agile

Conditions for Agile

Scrum vs Kanban

Agile Variants

Scaling Agile



Part 2: Guest Lecture

“Blocked” and “Idle” in a “Pull” system

“I can’t finish my work because I’m **blocked** waiting on someone else. Shouldn’t I just start on a new ticket?”

“Testing is **simpler** than development. The testers are **idle** for half of the time waiting for new work on their backlog!”

“Testing is **harder** than development. The testers can’t keep up with the incoming work, so the developers might as well be **idle** for half the time!”

Toyota Lean “House”

Goal: Highest Quality, Lowest Cost, Shortest Lead Time

Just In Time

- Keep value flowing
- Meet rate of demand
 - “Pull” not “Plan”

Just-in-Time

Continuous Flow
Takt Time
Pull System

Jidoka

Stop and notify
of abnormalities
Separate human
work and
machine work

Intelligent Automation

- Stop on faults
- Automate as much as possible
- Intelligent intervention

Heijunka

Standardized Work

Kaizen

Stability

Production levelling

(Small Batches)

Reduces Mura (unevenness)

Continuous Improvement

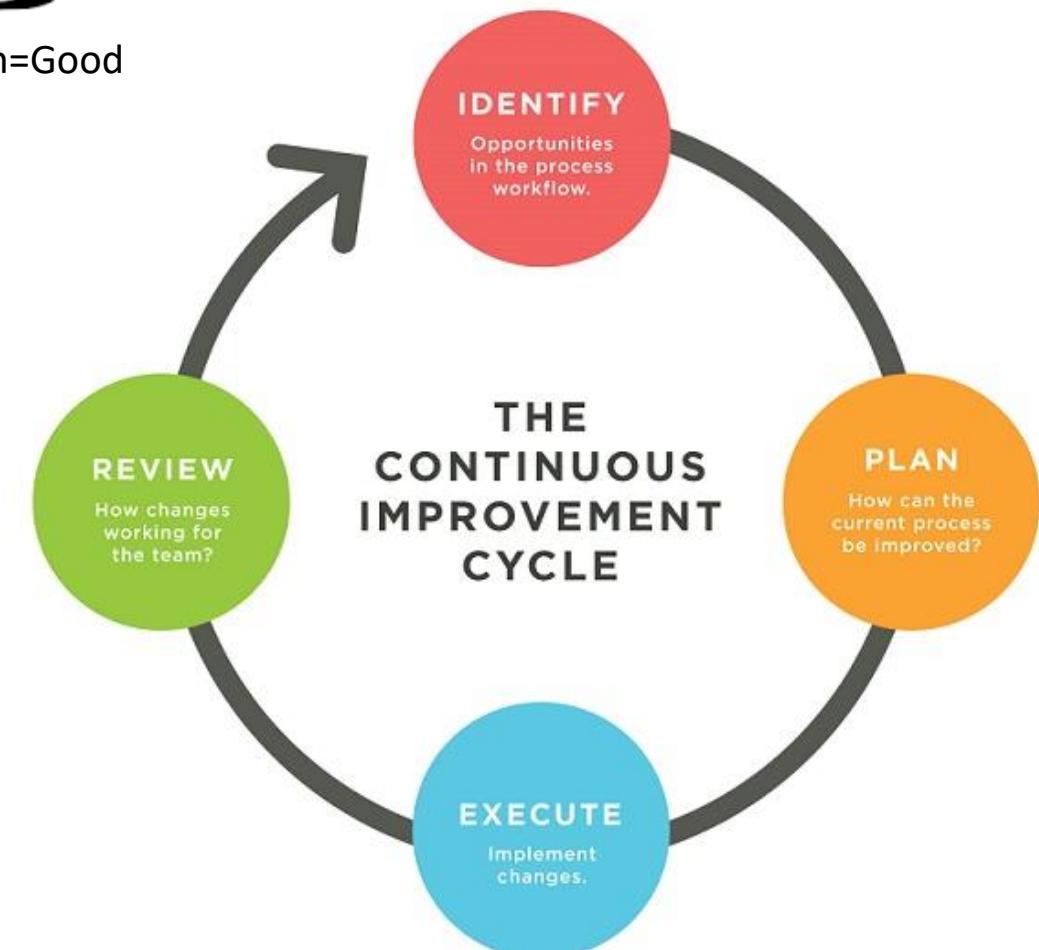
Reduces Muri, Muda & Mura

Kaizen

改善

Kai=Change Zen=Good

- *Continuous improvement*
- Regular, incremental improvements:
 - To all functions / processes (of the business)
 - From employees at all levels
- **Philosophy:**
 - a culture where everyone is actively engaged in suggesting and implementing improvements.
- **Strategy:**
 - Trust, respect and empower people. Events focused on improving the company, involving teams of employees at all levels



Heijunka

平準化

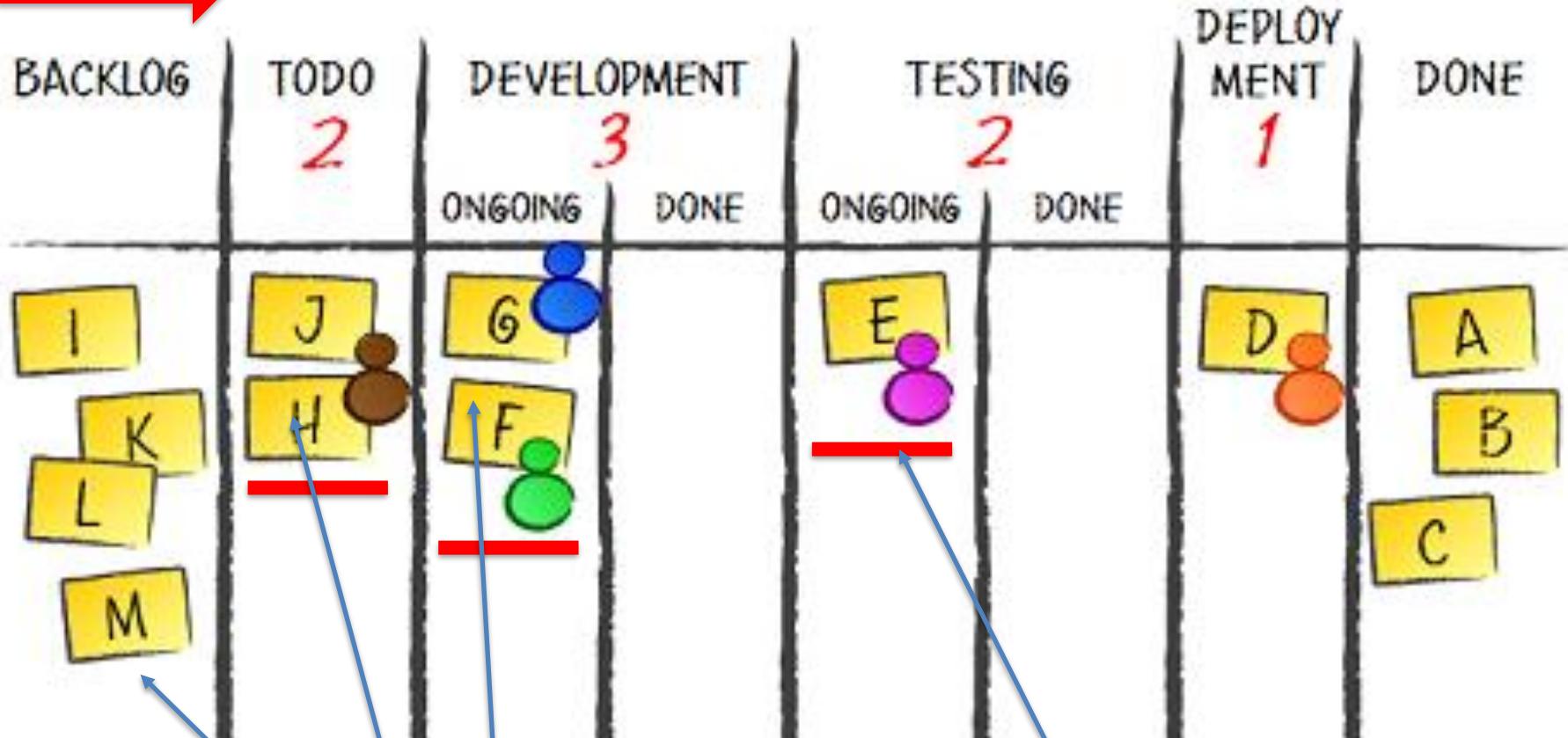
- *Production levelling*
- Unevenness (Mura) leads to waste (Muda)
- Smaller, more regular batches achieves:
 - Steady flow to the next stage
 - Greater predictability
 - Flexibility to changing demand

HEIJUNKA BY PRODUCT TYPE	
PRODUCT	CUSTOMER DEMAND PER WEEK
MODEL A plain t-shirt	5 green t-shirt icons representing weekly demand for Model A.
MODEL B t-shirt with pocket	3 green t-shirt icons representing weekly demand for Model B.
MODEL C t-shirt with v-neck	2 green t-shirt icons representing weekly demand for Model C.
MODEL D t-shirt with v-neck and pocket	2 green t-shirt icons representing weekly demand for Model D.
PRODUCTION SEQUENCE	
MASS PRODUCER	A sequence of 20 green t-shirt icons labeled A through D, representing a mass producer's production sequence.
LEAN PRODUCER	A sequence of 20 green t-shirt icons labeled A through D, representing a lean producer's production sequence.

(Note that this example does not address Heijunka by production quantity)

 **Heijunka**
Steady Flow

Kanban Board



 **Kaizen**

Adapt, change,
Continuously improve
Maximise flow of value

Work not PUSHED (or assigned)
It is PULLED from a backlog just in time

WIP limits

Workload is strictly limited
When blocked, collaborate

“Blocked” in a “Pull” system

“I can’t finish my work because I’m **blocked** waiting on someone else. Shouldn’t I just start on a new ticket?”

1. Taking on work to improve **your** output efficiency is **not** in the **team’s** best interest.
 - It won’t maximise **value** through the system.
 - Priority is **completion** of the **highest priority work**.
2. Being “blocked” forces team to collaborate:
 - Forced to ask: how can we solve the bottleneck?
 - Work as a team to get the flow going again
 - Improve the process (**Kaizen**) to achieve more **even-flow** (**Heijunka**)

“Idle” time in a “Pull” system

“Testing is **simpler** than development. The testers are **idle** for half of the time waiting for new work on their backlog!”

“Testing is **harder** than development. The testers can’t keep up with the incoming work, so the developers might as well be **idle** for half the time!”

- It’s a pull system - no requirement to be in sync!
- Each team focuses on getting value through their flow. Yet:
 - Kanban doesn’t magically solve resource utilization!
 - Teams need to adapt to improve end-to-end flow efficiency
- When “idle”, a bottleneck has been detected:
 - Improve the whole workflow
 - Reorganise the teams
 - Forces us to ask: “Do you really need separate developers / testers?”

Today

Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles
Blockers and Idleness

Kanban Examples

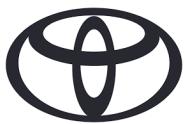


Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile



Part 2: Guest Lecture



</>



7 Lean Principles

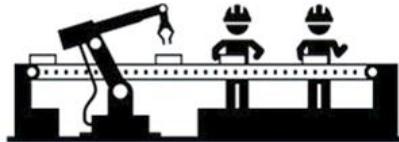
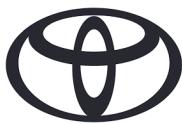
Lean Manufacturing

Toyota (2001)

1. Eliminate waste
2. Amplify learning
3. Defer Commitment
4. Deliver Fast
5. Empower the team
6. Build integrity in
7. See the whole

Software Development

?



</>



7 Lean Principles

Lean Manufacturing

Toyota (2001)

1. Eliminate waste
2. Amplify learning
3. Defer Commitment
4. Deliver Fast
5. Empower the team
6. Build integrity in
7. See the whole

Lean Software Development

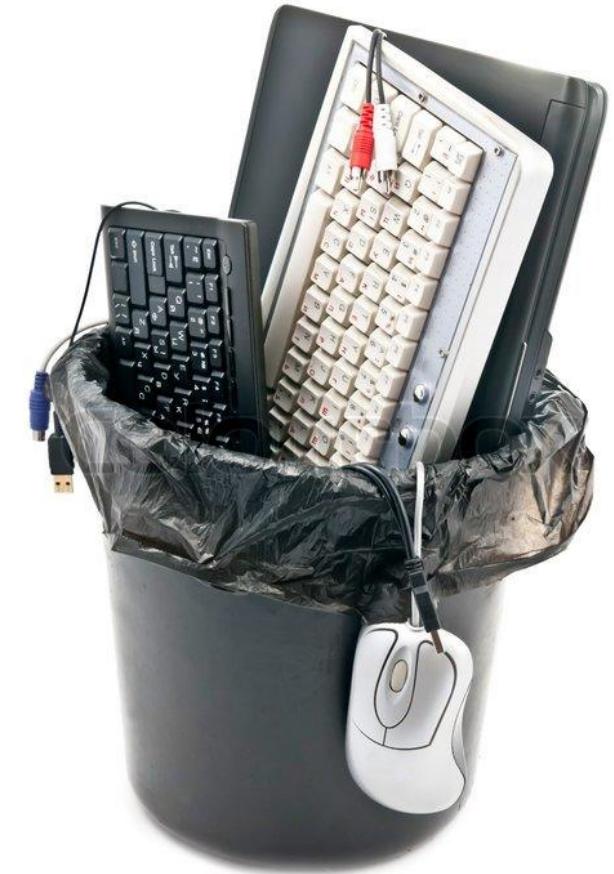
Poppenieck (2003)

1. Eliminate Waste
2. Create Knowledge
3. Defer Commitment
4. Deliver Fast
5. Respect People
6. Build Quality In
7. Optimise The Whole

Software Development Wastes

Industry research revealed these software development wastes:

- Building the wrong feature or product
- Mismanaging the backlog
- Rework
- Unnecessarily complex solutions
- Extraneous cognitive load
- Psychological distress
- Waiting/multitasking
- Knowledge loss
- Ineffective communication

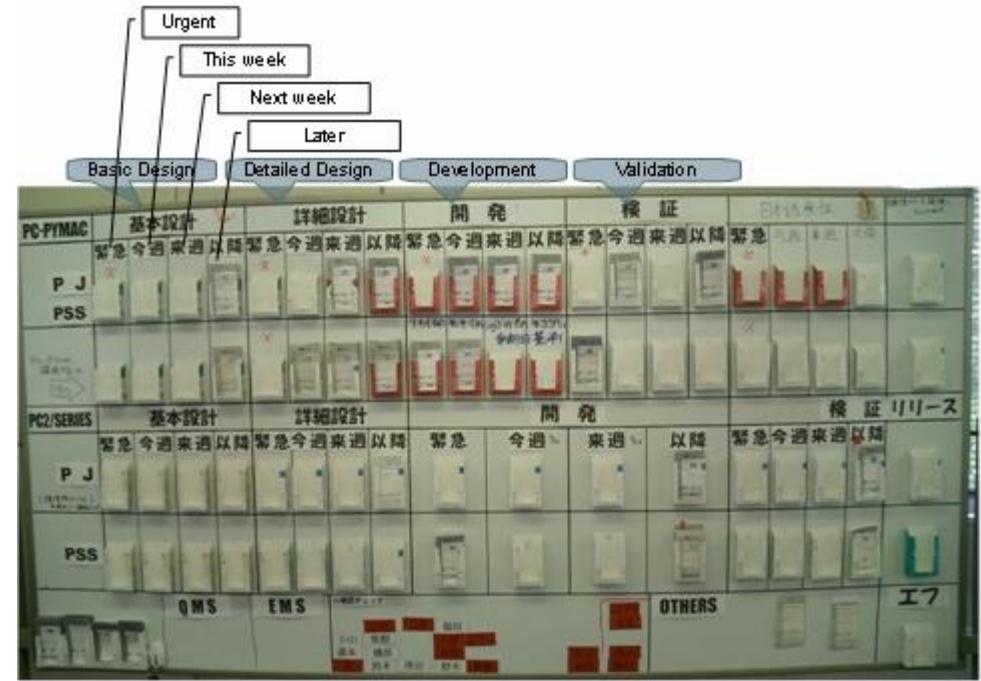


Software Development Waste (2017)

https://www.researchgate.net/publication/313360479_Software_Development_Waste

Software as a Production Line

- Software development is not a production or a manufacturing activity [1].
- Software engineers create different things every time, whereas manufacturing produces same things over and over again [2]
- How to design a Kanban board for software projects?



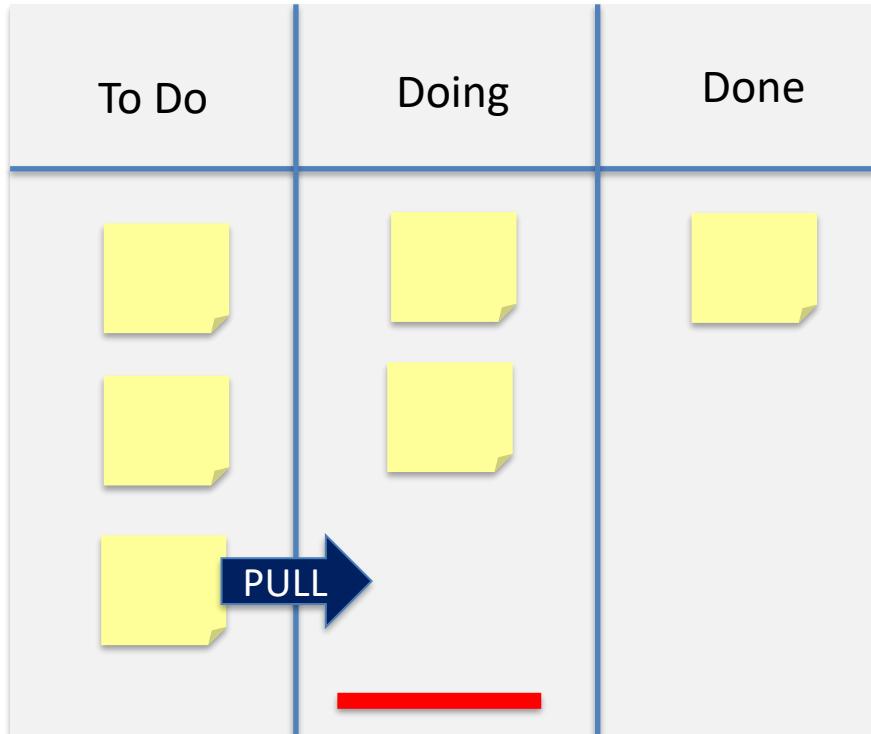
Kanbans are physical cards that are used to signal information from upstream to downstream

[1] Jack W. Reeves, "What is Software Design?" C++ Journal, 1992

[2] <https://www.infoq.com/articles/hiranabe-lean-agile-kanban/>

Kanban Example 1

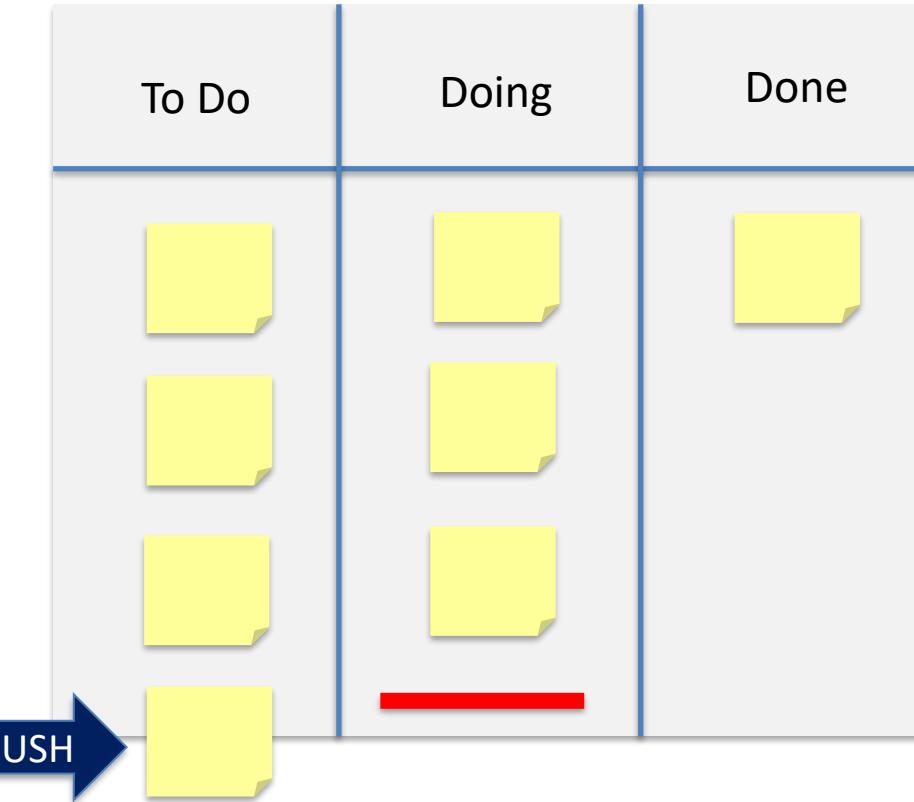
Typical of an **agile** team



- Limits WIP
- PULL from To Do list
- Helps visualise flow

Kanban Example 1

Typical of an **agile** team

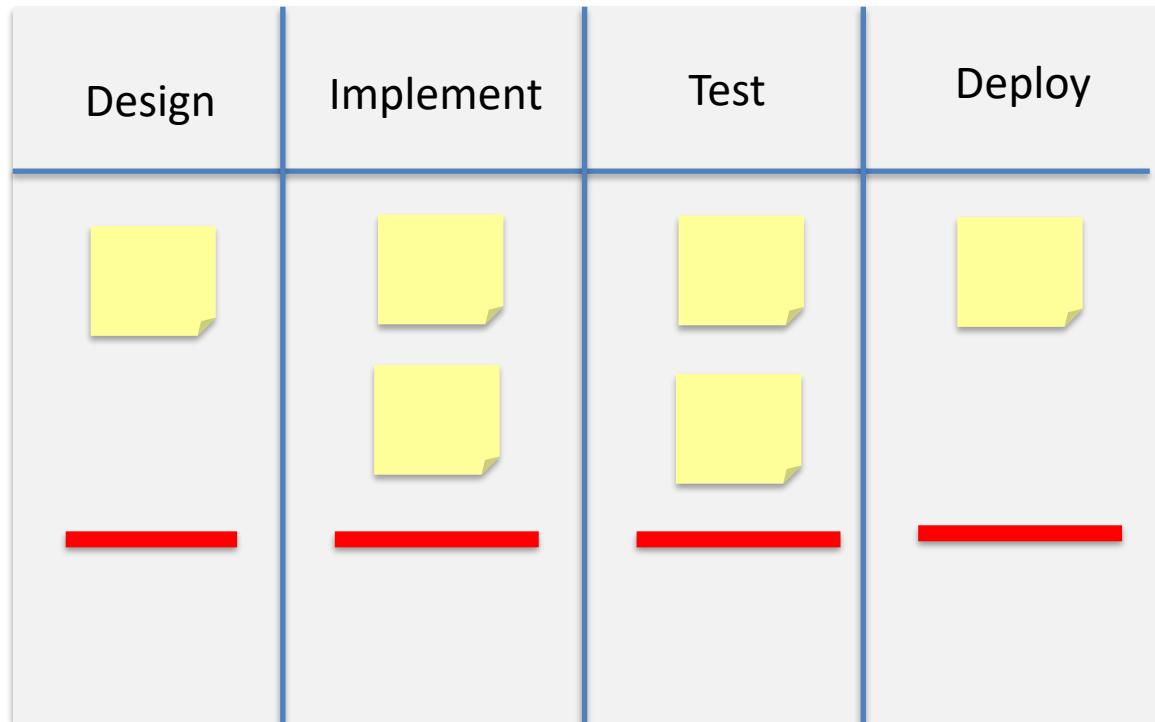


- Limits WIP
- PULL from To Do list
- Helps visualise flow

- Work PUSHED onto unbounded “To Do list” (backlog)
- No mapping of *processes* (flow of work)
- Hard to achieve a rhythm

Kanban Example 2

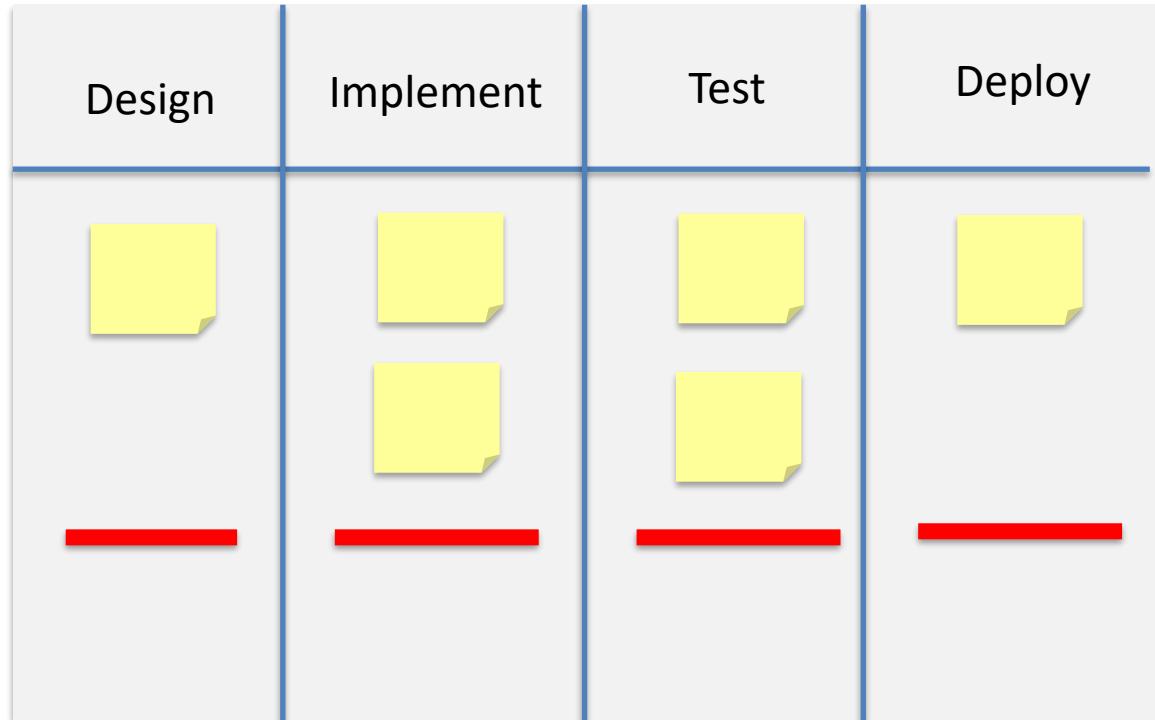
For a **waterfall-based** workflow



- WIP limit encourages completion
- Helps visualise flow / status
- Maps flow of work (*processes*)

Kanban Example 2

For a **waterfall-based** workflow

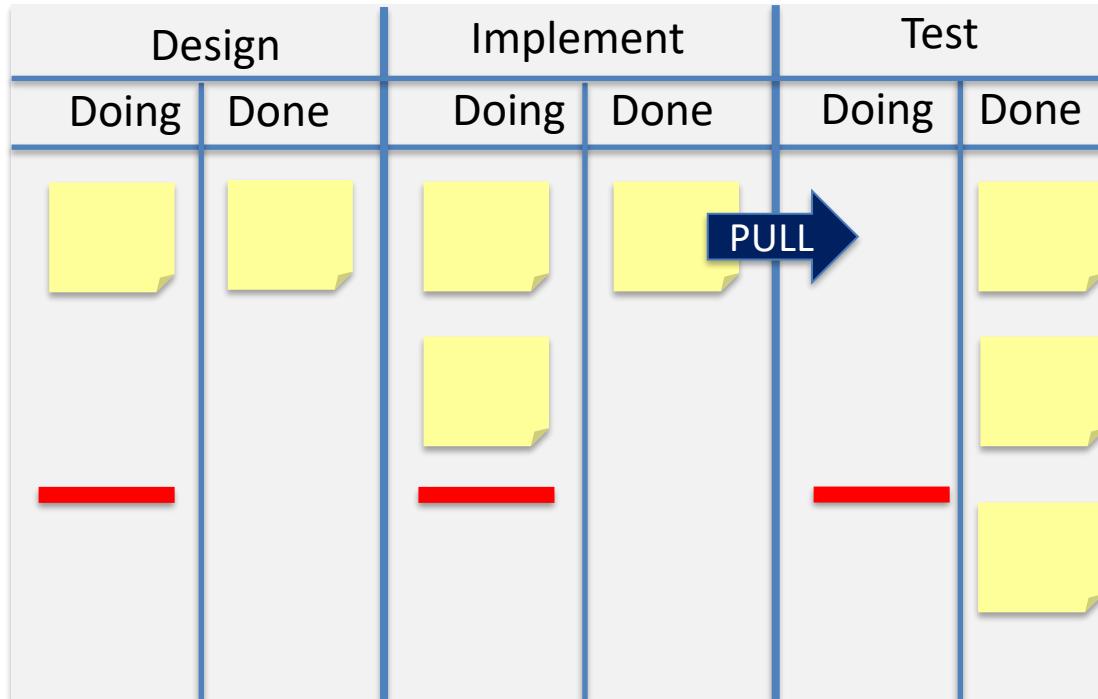


- WIP limit encourages completion
- Helps visualise flow
- Maps flow of work (*processes*)

- Who moves the cards?
- Is it *push* or *pull*?
- No downstream *signalling*

Kanban Example 3

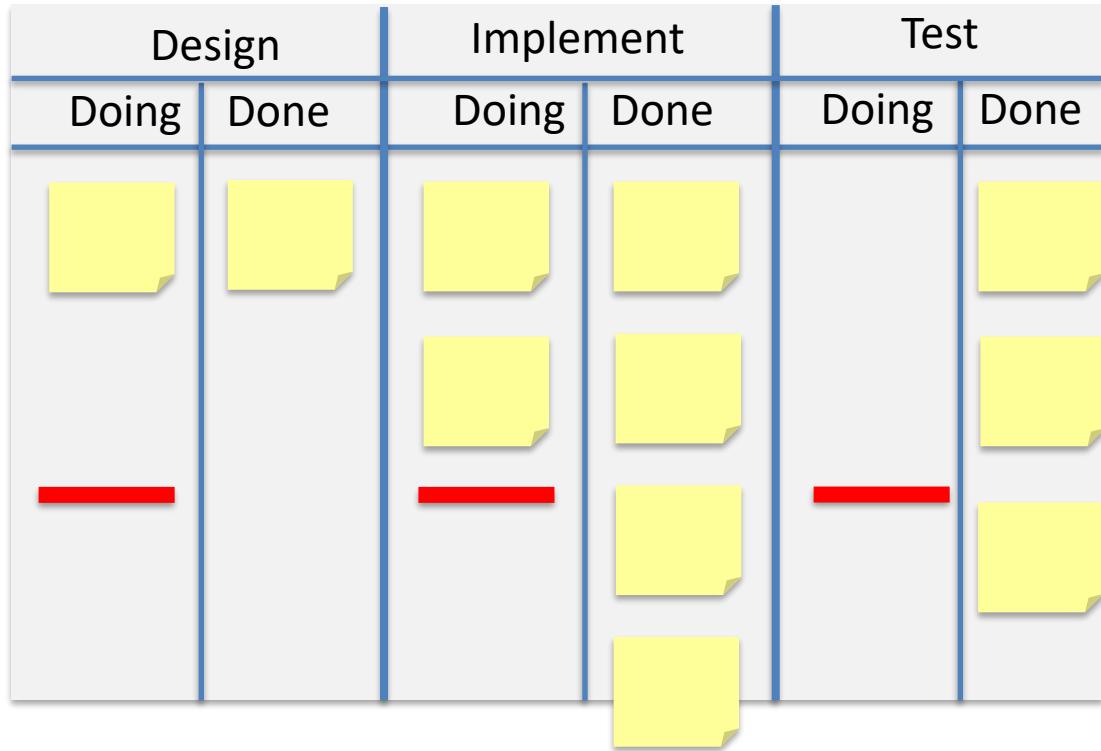
For a **waterfall-based** workflow



- WIP limit encourages completion
- Helps visualise flow
- Maps flow of work (*processes*)
- PULL from upstream
- Work done *on demand* (no over-production)

Kanban Example 3

For a **waterfall-based** workflow

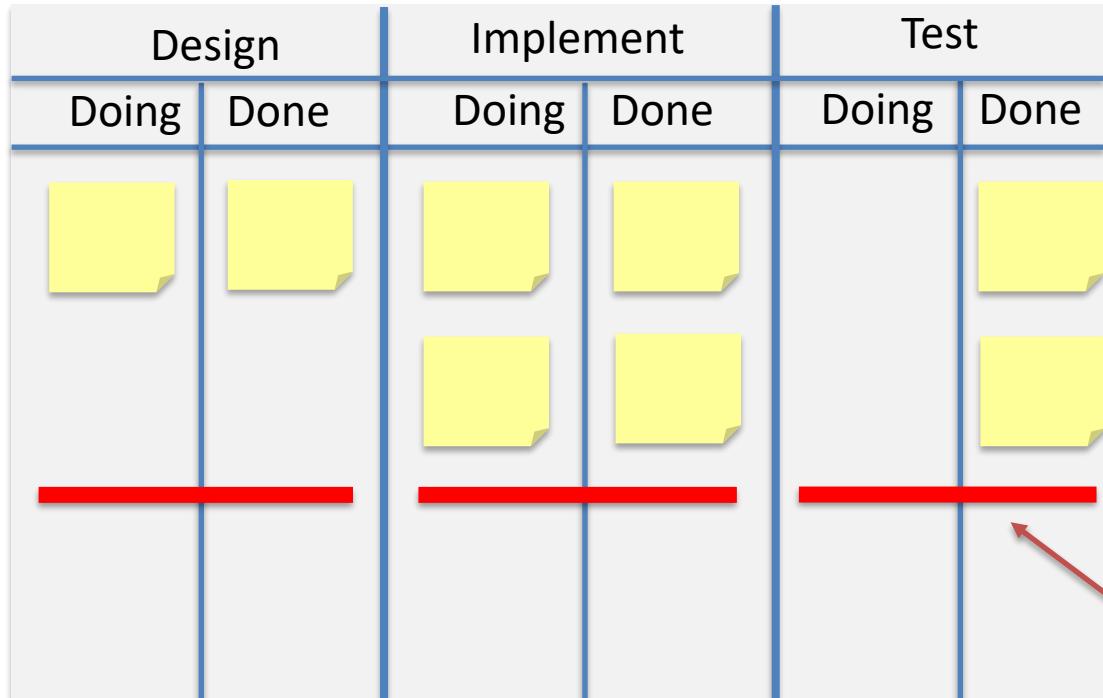


- WIP limit encourages completion
- Helps visualise flow
- Maps flow of work (*processes*)
- PULL from upstream
- Work done *on demand* (no over-production)

- No *upstream* signalling
- Prioritisation?
- Load balancing?

Kanban Example 3

For a **waterfall-based** workflow

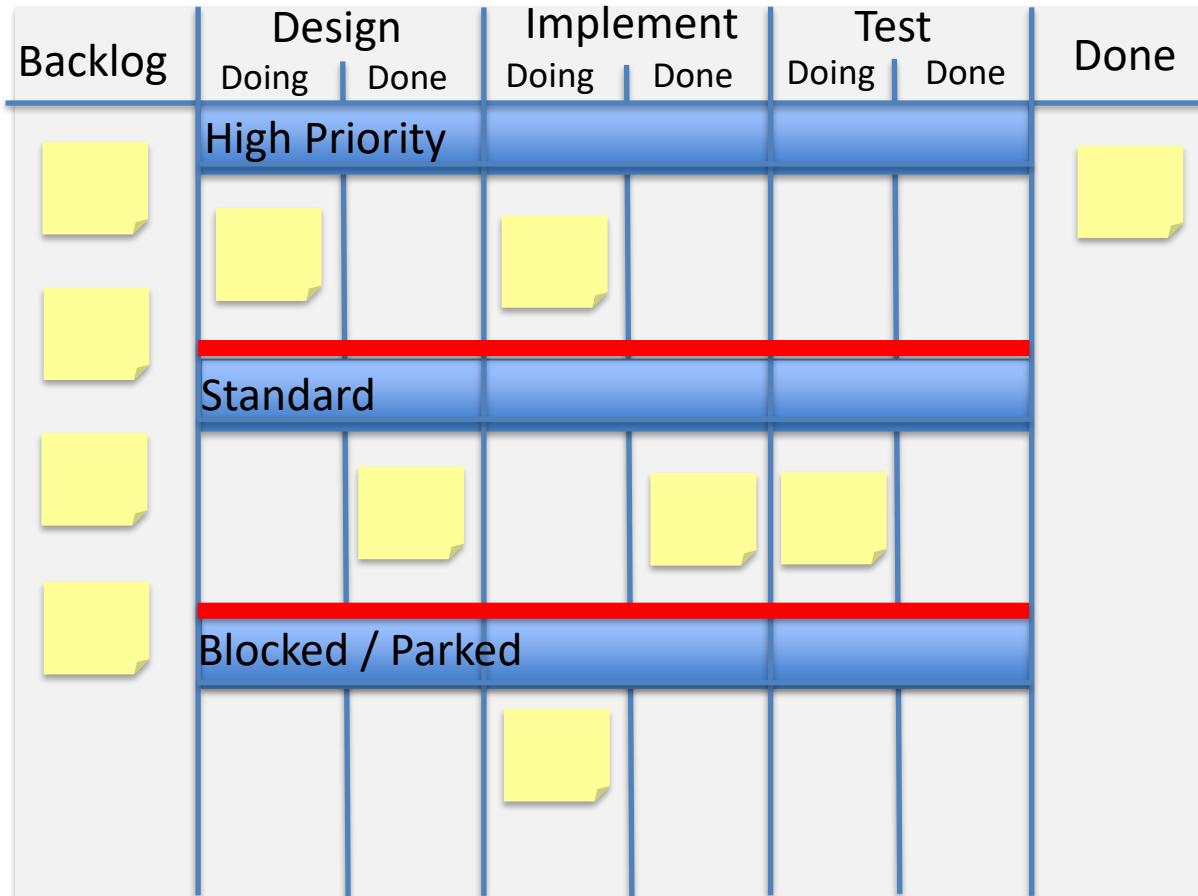


- WIP limit encourages completion
- Helps visualise flow
- Maps flow of work (*processes*)
- PULL from upstream
- Work done *on demand* (no over-production)

- upstream signalling
- Prioritisation?
- Load balancing?

Kanban Example 4

With swimlanes for prioritised work



- Helps visualise flow
- Maps flow of work (*processes*)
- Work can be prioritised and expedited
- Limit WIP for signalling

Today

Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles
Blockers and Idleness
Kanban Examples



Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile



Part 2: Guest Lecture

Quiz

Should you use Agile?



warwick.ac.uk/pm4cs/7

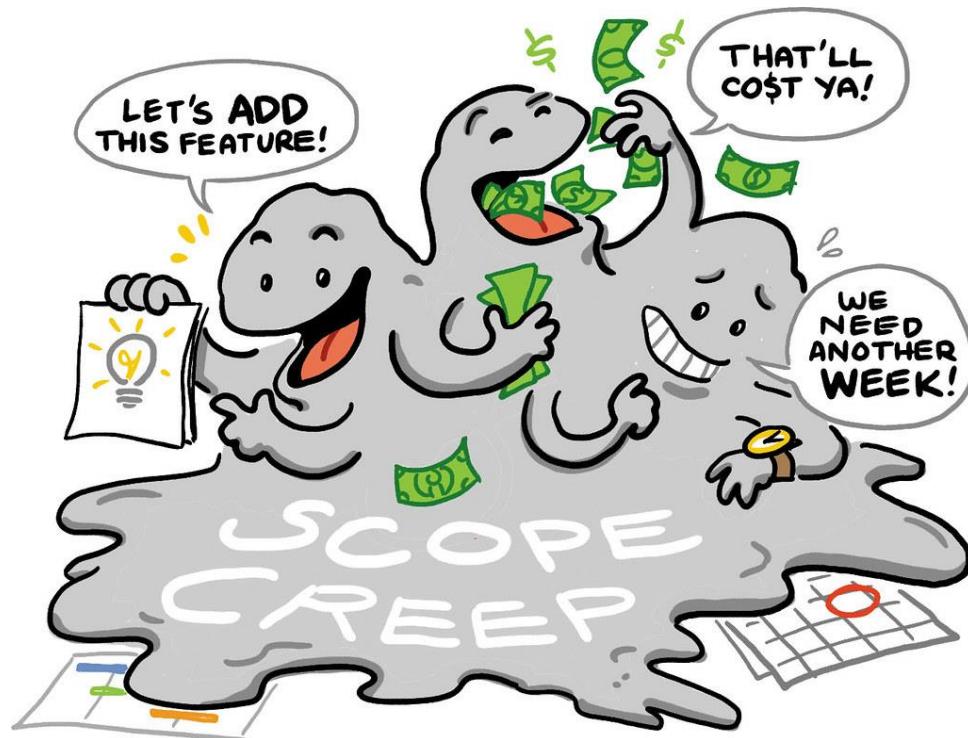
Your customer is a law firm. Money is no problem, but they are too busy for regular meetings - email is preferred. They just want you to get on with the work. They trust you to do a good job.

Should you use agile?

- 1 Use Agile
- 2 Don't Use Agile



Customer is a startup. Must keep costs as low as possible. Unclear about viability of idea. Concerned about inevitable scope creep with each iteration, resulting in high costs.



Should you use agile?

- 1 Use Agile
- 2 Don't Use Agile



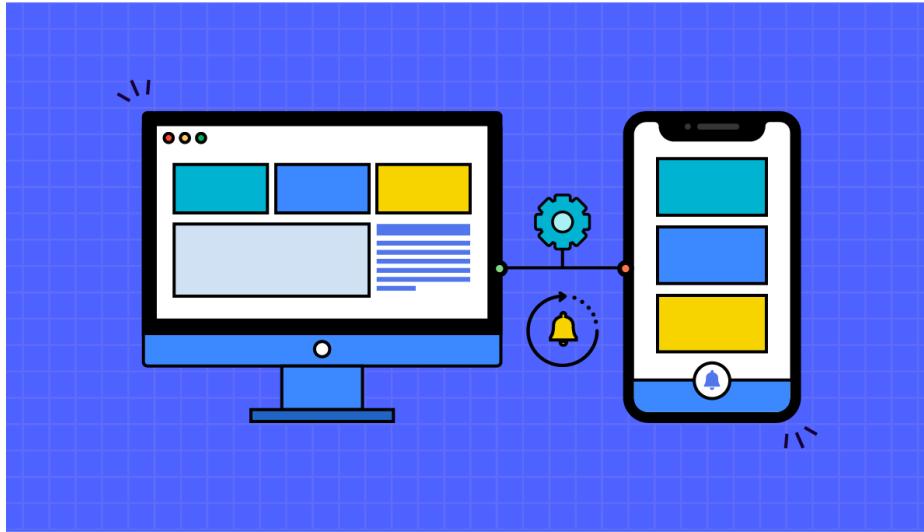
Boeing have asked you to develop a new aeroplane docking system that allows electric planes to be recharged in the air. Large budget, clear goals. Mistakes are inevitable but will provide valuable learning.

Should you use agile?



1 Use Agile

2 Don't Use Agile



Cloning website to a mobile app. Design has already been done by another team as this is part of a bigger project. External factors set budget and deadline.

Should you use agile?

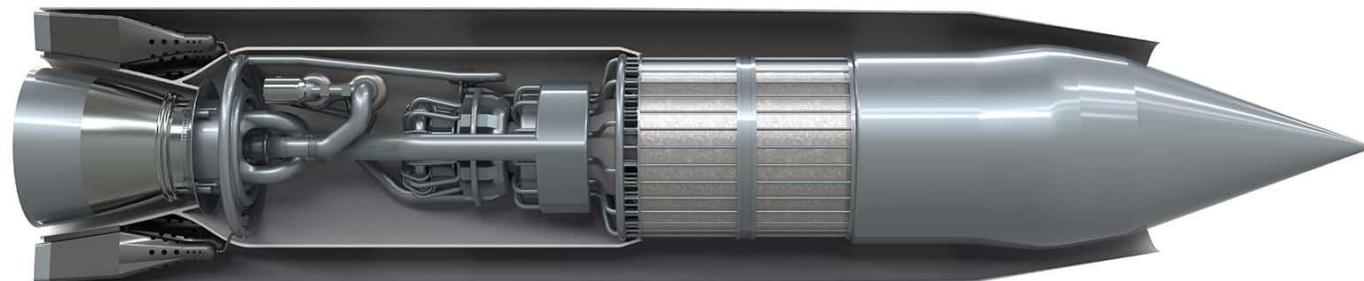
- 1 Use Agile
- 2 Don't Use Agile



You are building the cooling component of a rocket engine. Hard to test in isolation. Flexible on budget but not on deadline. Late changes will be very problematic.

Should you use agile?

- 1 Use Agile
- 2 Don't Use Agile





Leaderboard

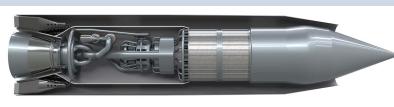


Position	Participants	Score

Total Participants: 0

Average Score: 0

Conditions for Agile

Conditions	Favourable	Unfavourable
Customer Involvement 	Close collaboration and rapid feedback are feasible . Customers know better what they want as the process progresses.	Requirements are clear at the outset and will remain stable. Customers are unavailable for constant collaboration.
Market Environment 	Customer preferences and solution options change frequently .	Market conditions are stable and predictable.
Impact of Interim Mistakes 	They provide valuable learning.	They may be catastrophic.
Innovation Type 	Problems are complex, solutions are unknown , and the scope isn't clearly defined . Product specifications may change . Creative breakthroughs and time to market are important. Cross-functional collaboration is vital.	Similar work has been done before , and innovators believe the solutions are clear. Detailed specifications and work plans can be forecast with confidence and should be adhered to. Problems can be solved sequentially in functional silos.
Modularity of Work 	Incremental developments have value, and customers can use them. Work can be broken into parts and conducted in rapid, iterative cycles. Late changes are manageable.	Customers cannot start testing parts of the product until everything is complete . Late changes are impossible or expensive .

Today

Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles
Blockers and Idleness
Kanban Examples



Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile



Part 2: Guest Lecture

Scrum vs Kanban

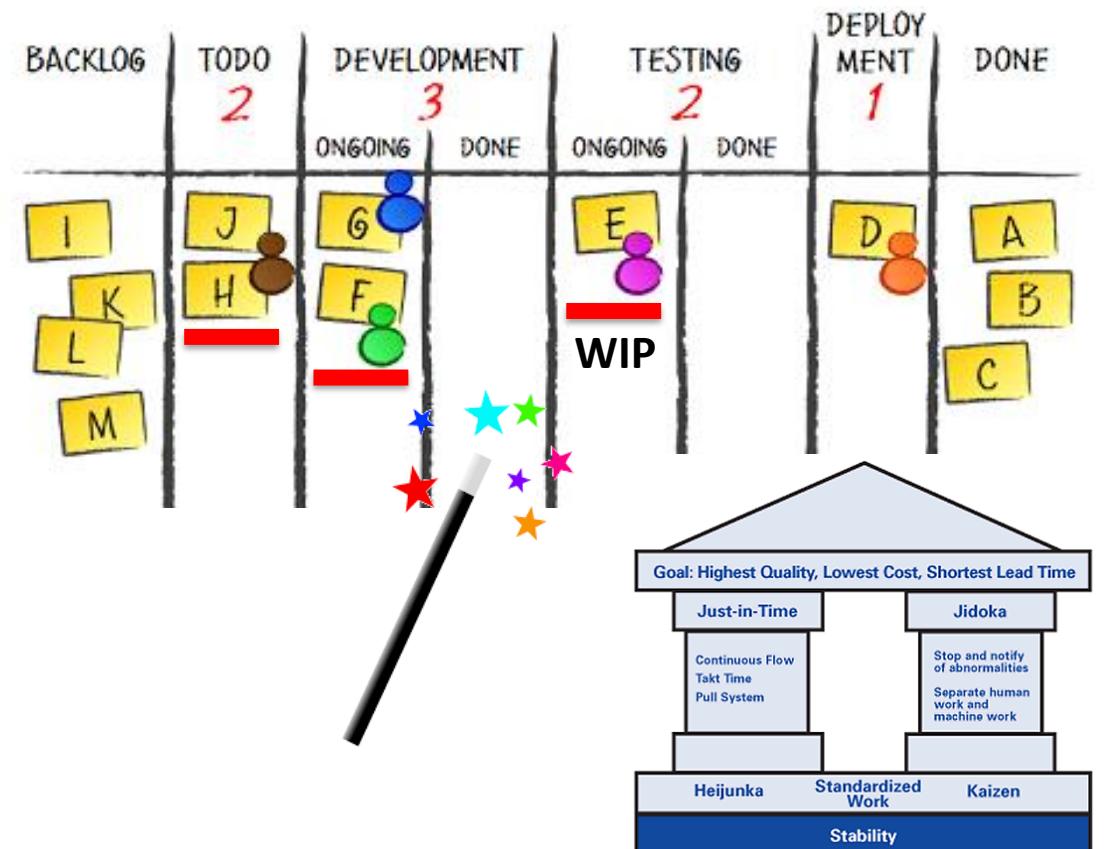
Scrum



79

(iterations, servant leader, empowered team)

Kanban



Both: (communication, end-user)

(continuous flow, pull system, lean)

	Scrum	Kanban
Iterations	Timeboxed iterations prescribed.	Continuous delivery, typically event-driven.
	Can have separate cadences for planning, release, and process improvement.	Timeboxing optional.
Commitment	Team commits to a specific amount of work for this iteration.	Commitment optional.
Scope alteration	Cannot add items to ongoing iteration.	Can add new items whenever capacity is available
Visualisation	Optional. Often a Kanban board	A Kanban board
Persistence	Kanban board is reset between each sprint	Kanban board is persistent
Work size	Items must be broken down so they can be completed within 1 sprint.	No particular item size is prescribed (but smaller is leaner)
WIP Limits	WIP limited indirectly (per sprint)	WIP limited directly (per workflow state)
Estimation	Estimation prescribed	Estimation optional
Prioritisation	Prescribes a prioritized product backlog	Prioritization is optional.
Roles	Prescribes 3 roles (PO/SM/Team)	Doesn't prescribe any roles
Ownership	A sprint backlog is owned by one specific team	Kanban board may be shared by multiple teams or individuals
Cross-functionality	Cross-functional teams prescribed. Specialist teams allowed.	Cross-functional teams optional.
Planning tools	Burndown chart prescribed	No particular type of diagram is prescribed
Default metric	Velocity (for planning and process improvement).	Lead time (for planning and process improvement).

When to use Scrum / Kanban?

Scrum

- the work is in large vague chunks that you can then break down into smaller chunks
- the work forms part of an even bigger series of long-term goals (“releases” or “horizons”)
- regular fixed timeboxes let you measure your rate of progress
- fixed timeboxes and a rate of progress mean you can plan towards the big long-term goals.

Kanban

- the work pops up in front of the team (i.e. event based)
- the work is in small discrete pieces
- there is no overall long-term objective or goal to reach
- planning is unimportant or irrelevant.
- where new work activities arrive continuously (e.g. a “support” environment)

Today

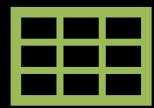
Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles
Blockers and Idleness
Kanban Examples



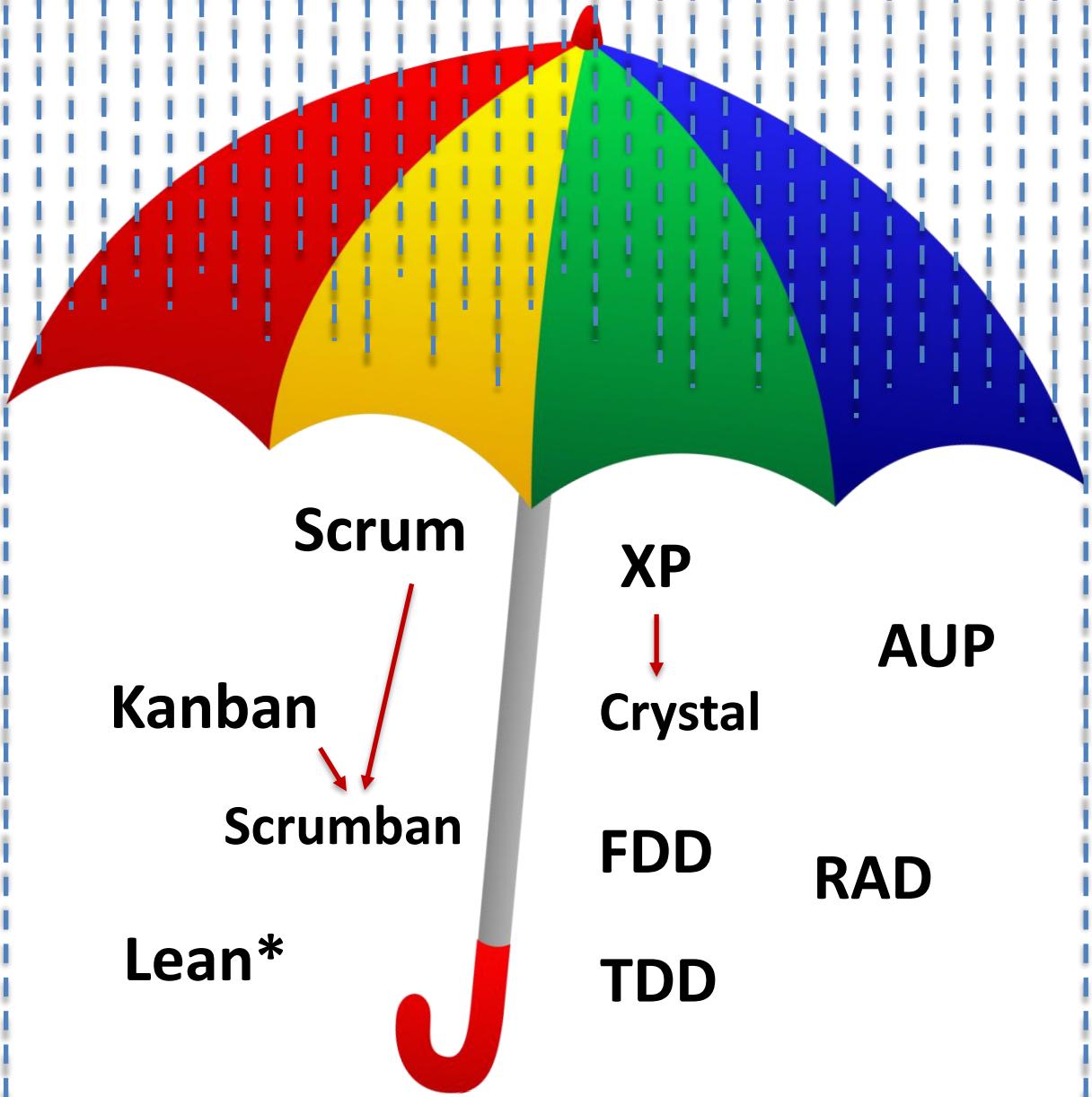
Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile

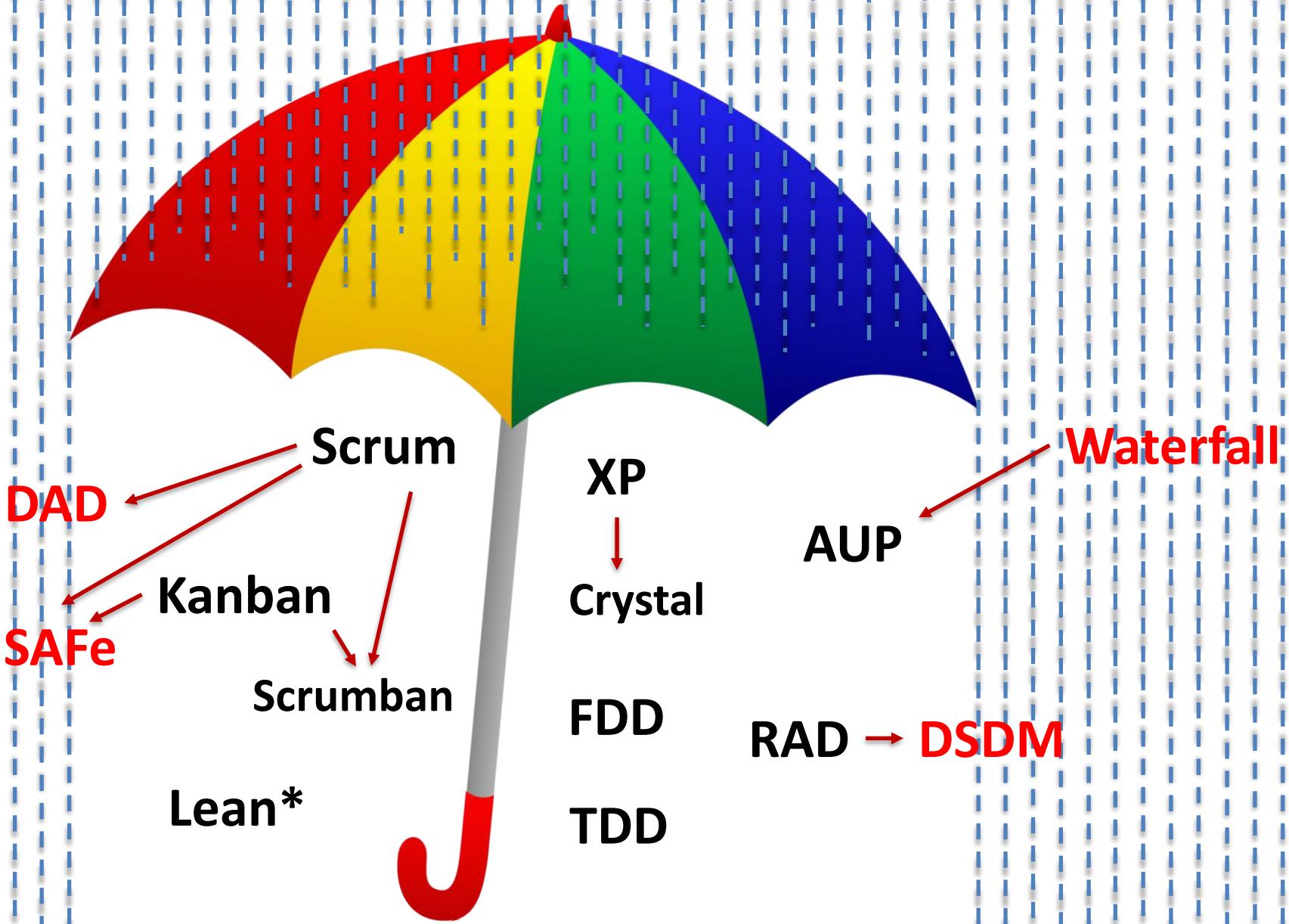


Part 2: Guest Lecture

Agile Approaches



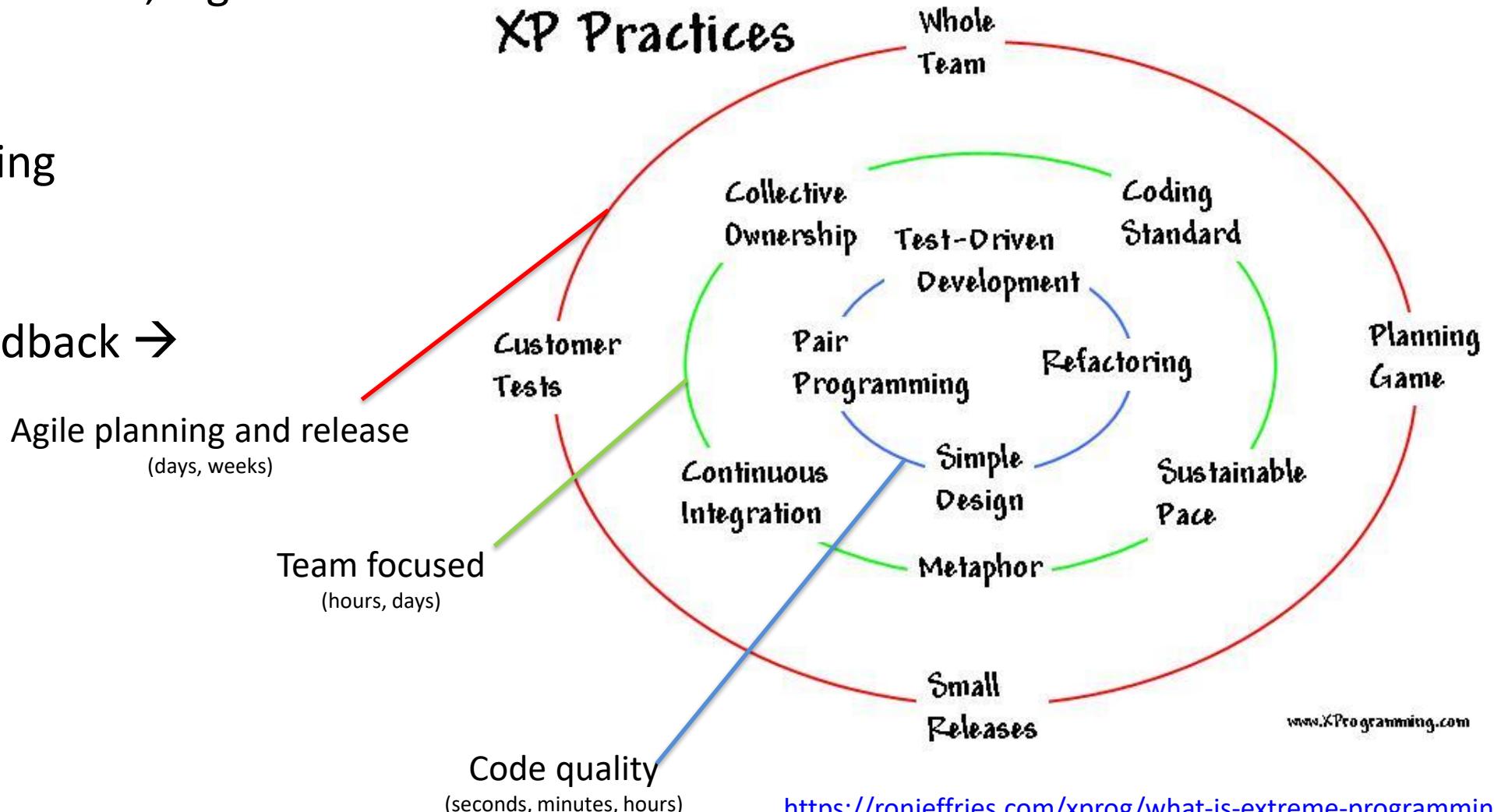
Agile Approaches



Extreme Programming

A series of **core practices**, e.g.

- Agile planning
- Collaboration
- Pair Programming
- Test oriented
- Quality code
- Continuous feedback →

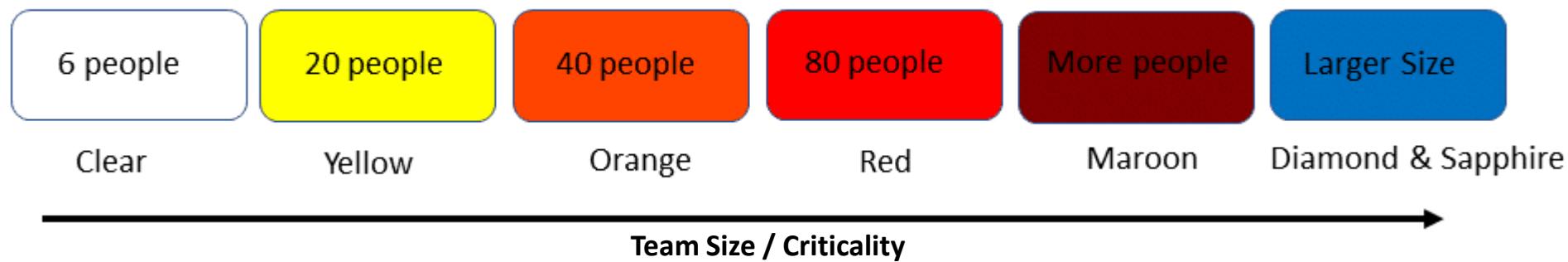


Crystal Method

“XP without discipline” -- creator Alistair Cockburn

*The Agile Manifesto values **Individuals and interactions** over processes and tools*

- A family of methods that focus on **Teamwork, Communication, Simplicity**
- **Lightweight** (more talking less documentation)
- **Adaptable** (Processes and tools adapted to fit **team size** and **criticality**)

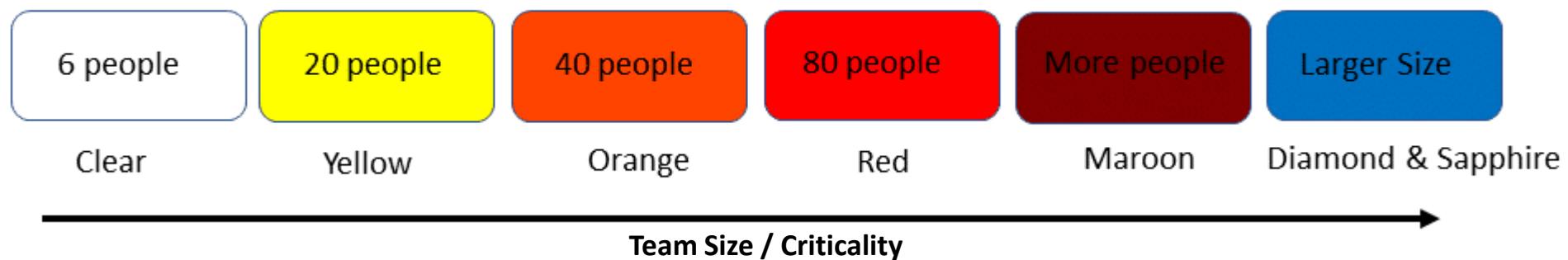


Crystal Method

“XP without discipline” -- creator Alistair Cockburn

*The Agile Manifesto values **Individuals and interactions** over processes and tools*

- A family of methods that focus on **Teamwork, Communication, Simplicity**
- **Lightweight** (more talking less documentation)
- **Adaptable** (Processes and tools adapted to fit **team size and criticality**)



- | | | | | |
|-------------------|---------------------|----------------------|---------------------------------|--------------------|
| • Lightweight | • Code areas | • Skill-based teams | • Adaptive teams | • Mission critical |
| • Self-organising | • Automated testing | • Quarterly releases | • Methods chosen to fit project | • Risk to life |

Some Other Agile Techniques

Rapid Application Development (RAD)

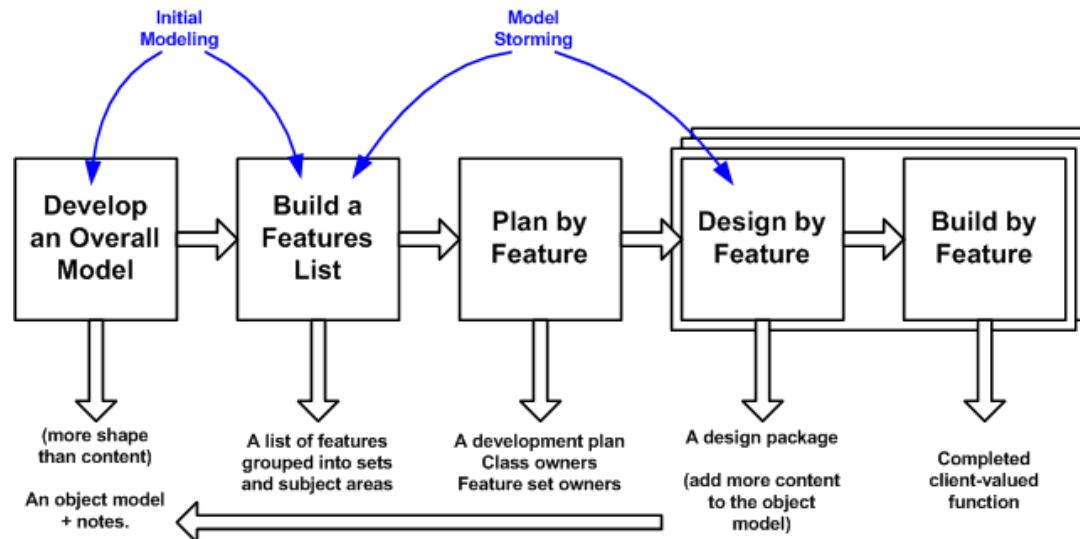
Particularly suited to UI building. Relies on **prototypes** instead of specifications. Importance of the *user design* phase to build a working model.

Test-Driven Development (TDD)

Requirements are turned into very specific test cases, based on use-cases, before writing the code. Once code passes test harness, then refactor. Particularly suited to improving and debugging legacy code.

Feature-Driven Development (FDD)

Based on client-value functionality.



Today

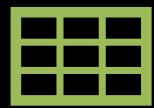
Part 1: Lecture



Lean

Toyota's "Lean House"

Lean Thinking:
maximising flow of value



Kanban

Kanban Software
Development Principles
Blockers and Idleness
Kanban Examples



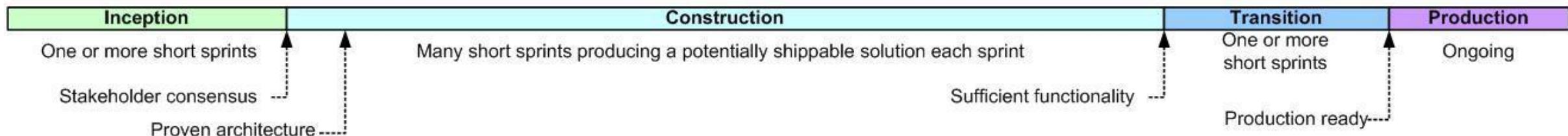
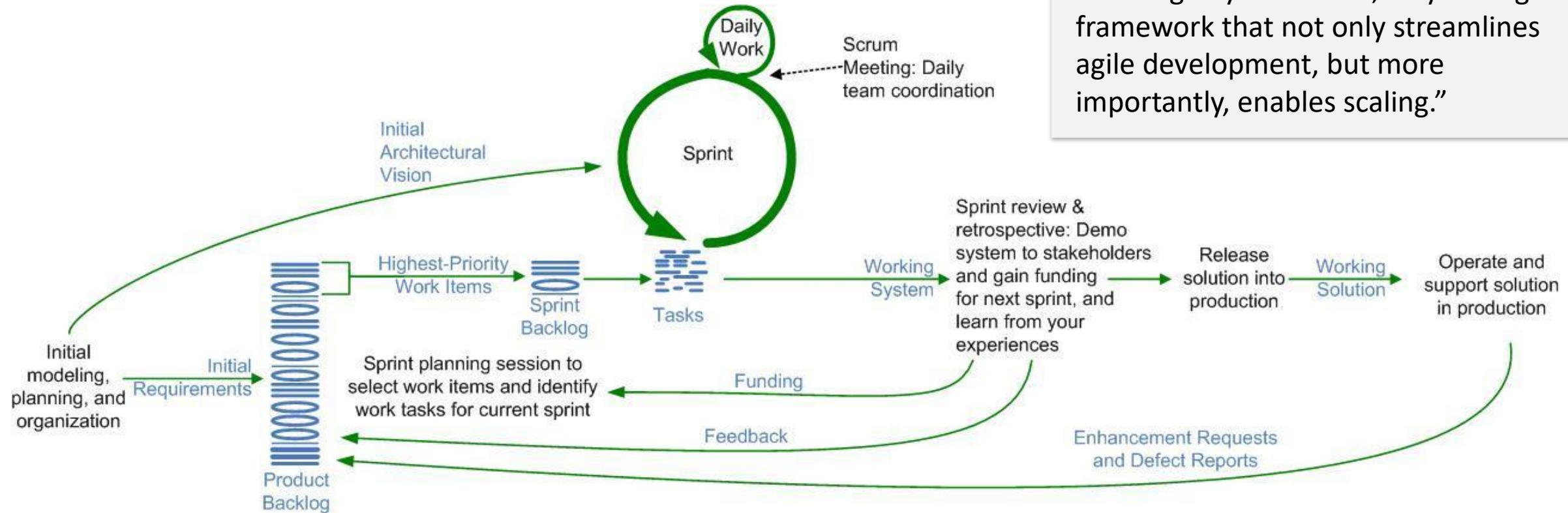
Doing Agile

Conditions for Agile
Scrum vs Kanban
Agile Variants
Scaling Agile



Part 2: Guest Lecture

Disciplined Agile Delivery

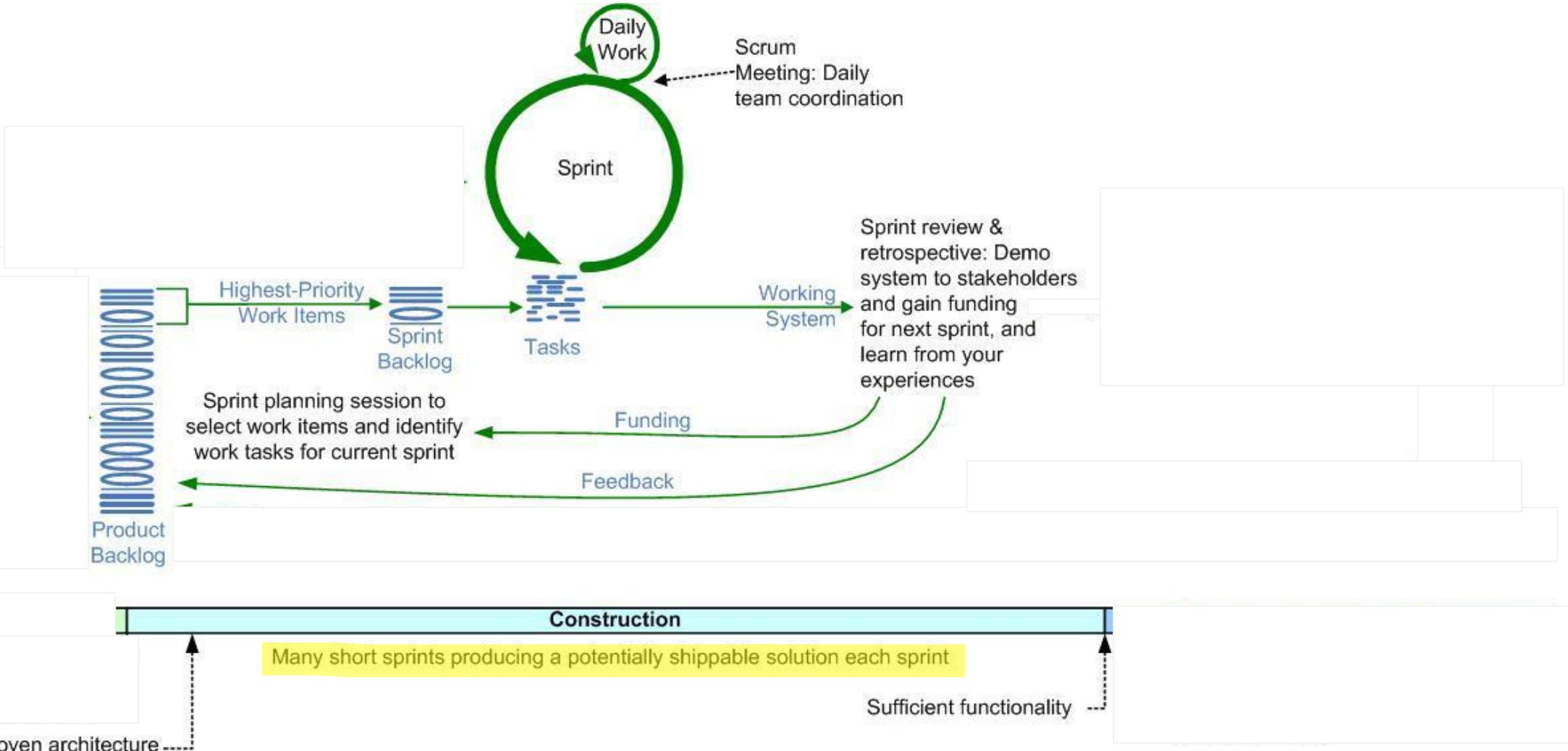


Copyright © 2010 IBM Corporation

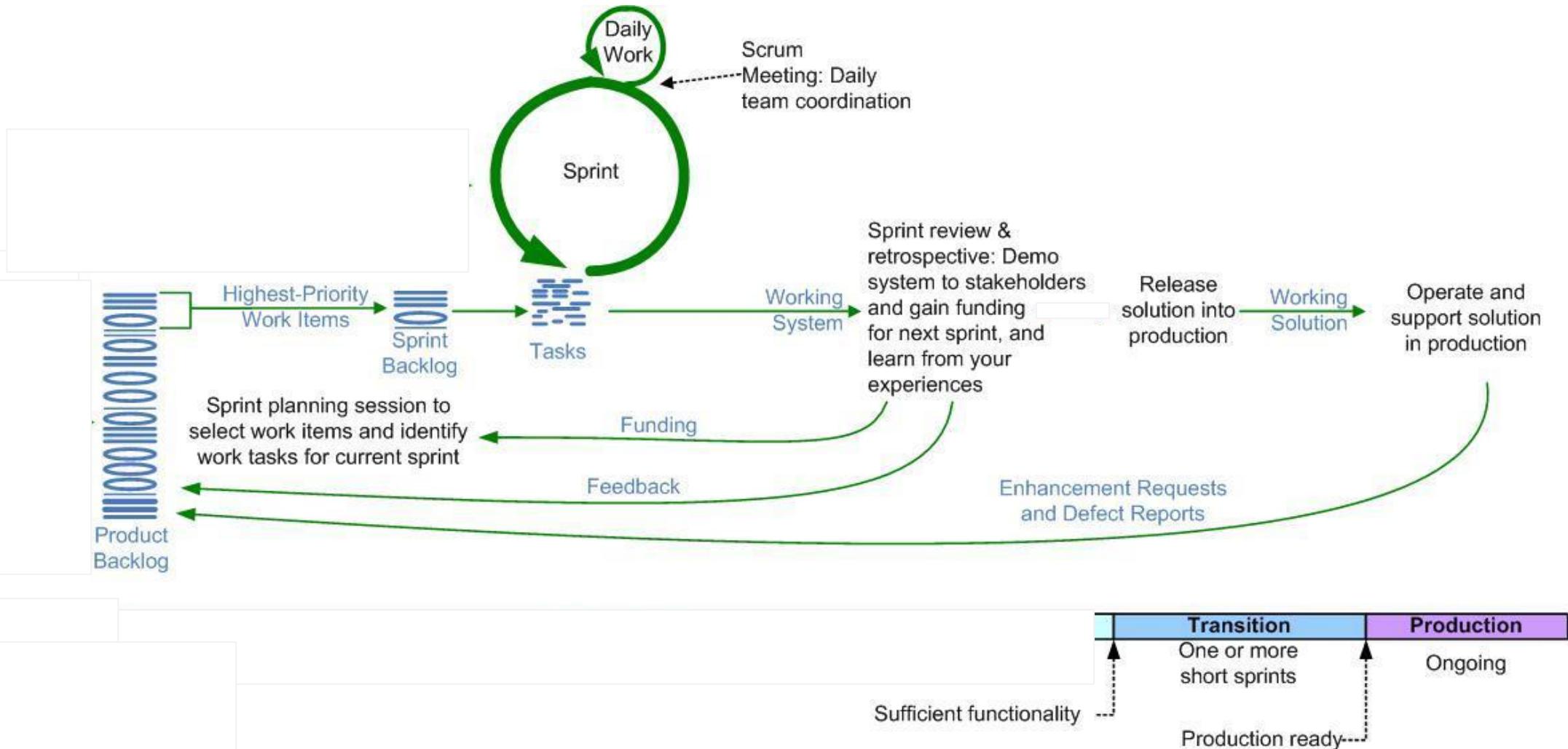
Disciplined Agile Delivery



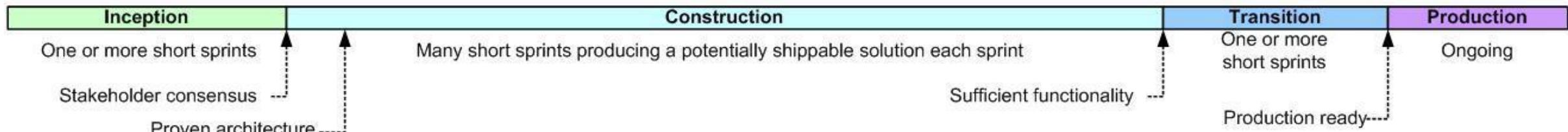
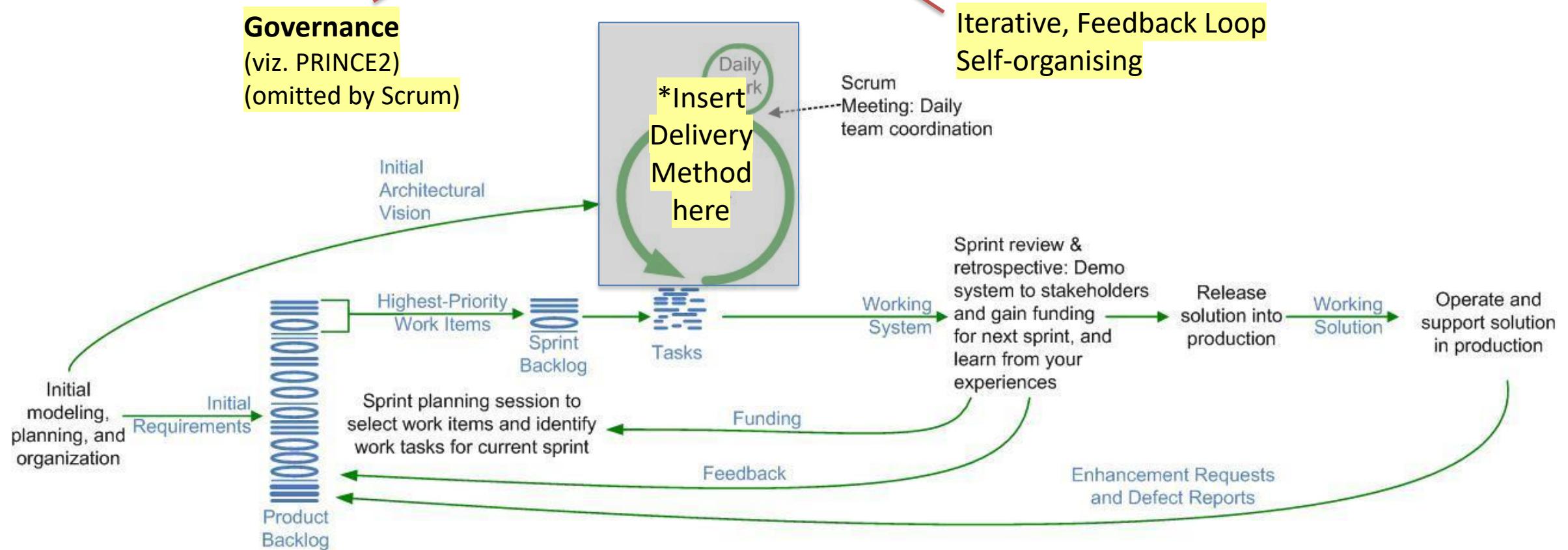
Disciplined Agile Delivery



Disciplined Agile Delivery



Disciplined Agile Delivery

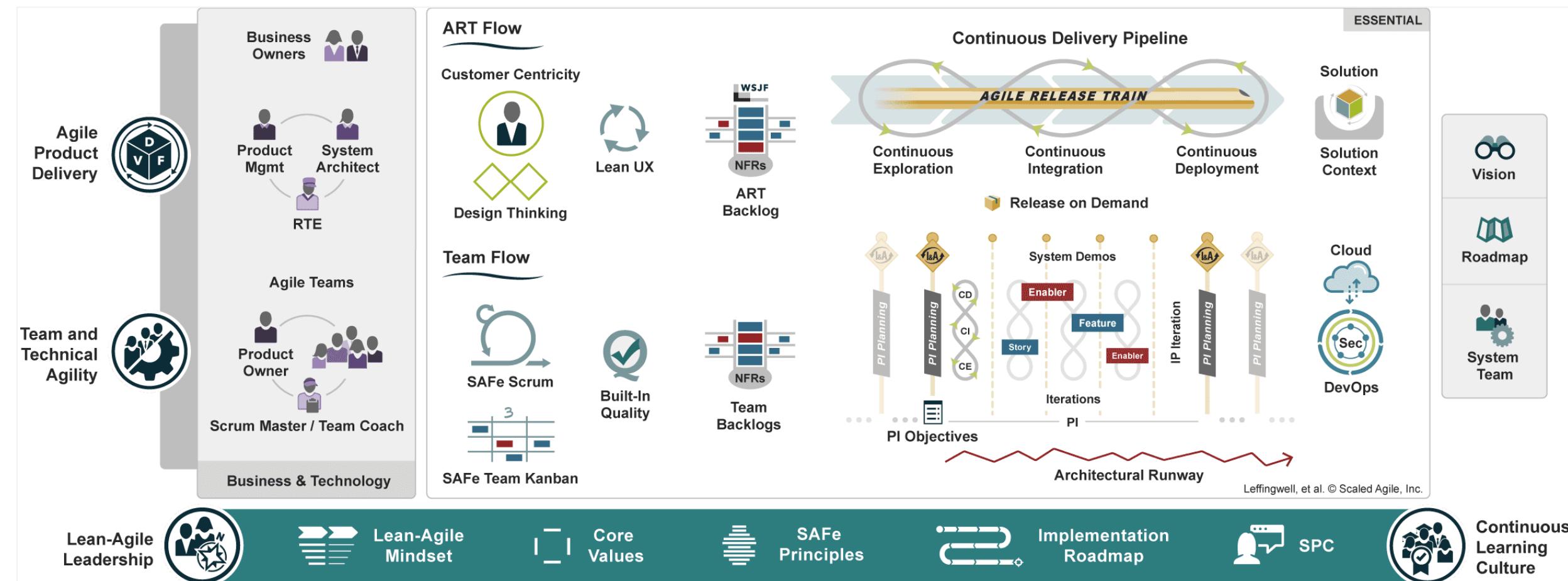


Copyright © 2010 IBM Corporation

The Scaled Agile Framework

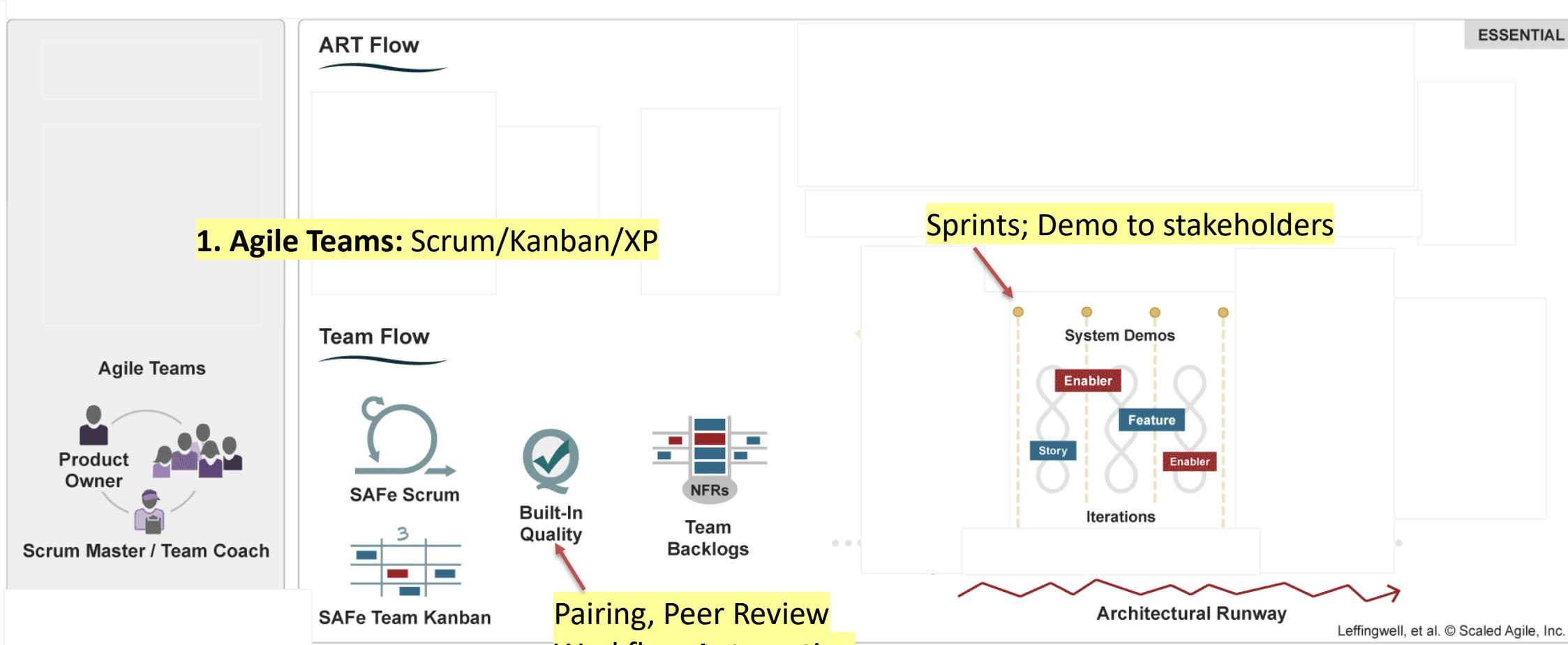
A set of organization and workflow patterns intended to guide enterprises in scaling lean and agile practices

<https://www.scaledagileframework.com>



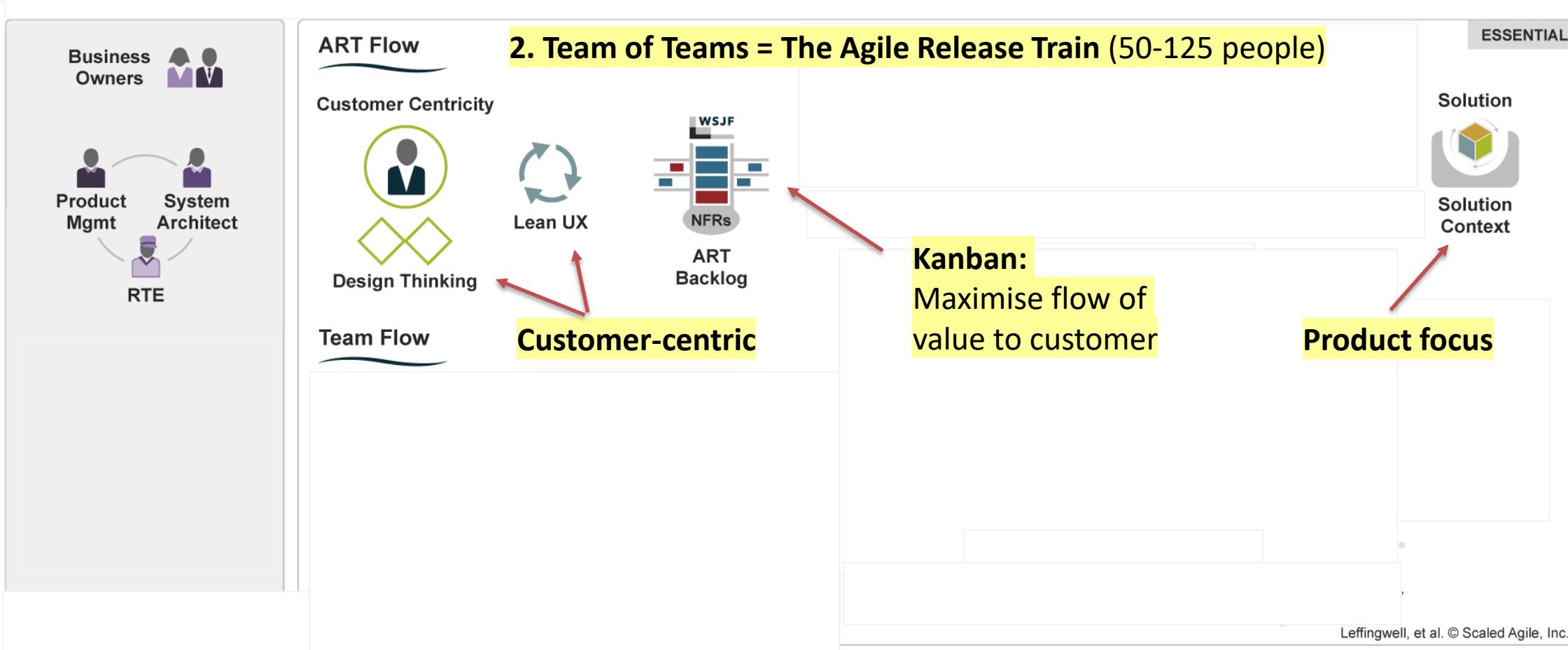
The Scaled Agile Framework

<https://www.scaledagileframework.com>



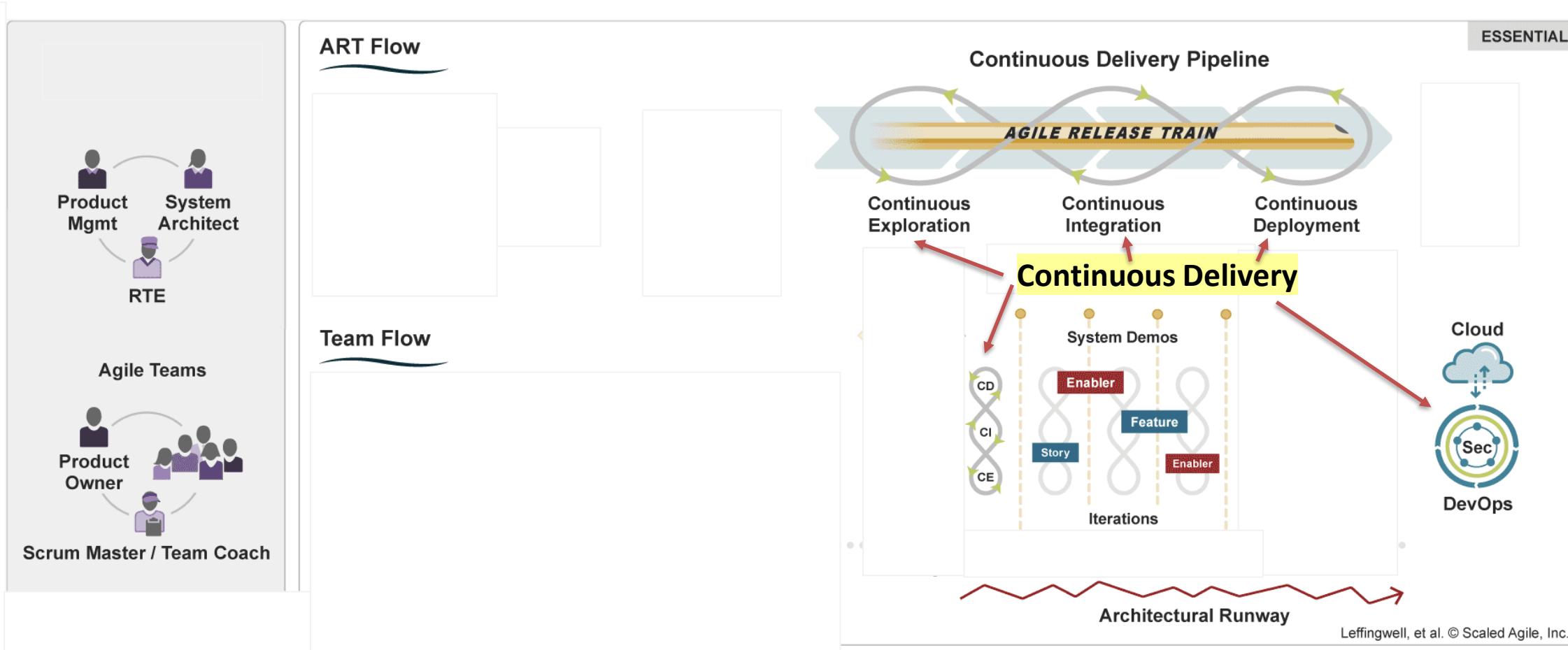
The Scaled Agile Framework

<https://www.scaledagileframework.com>



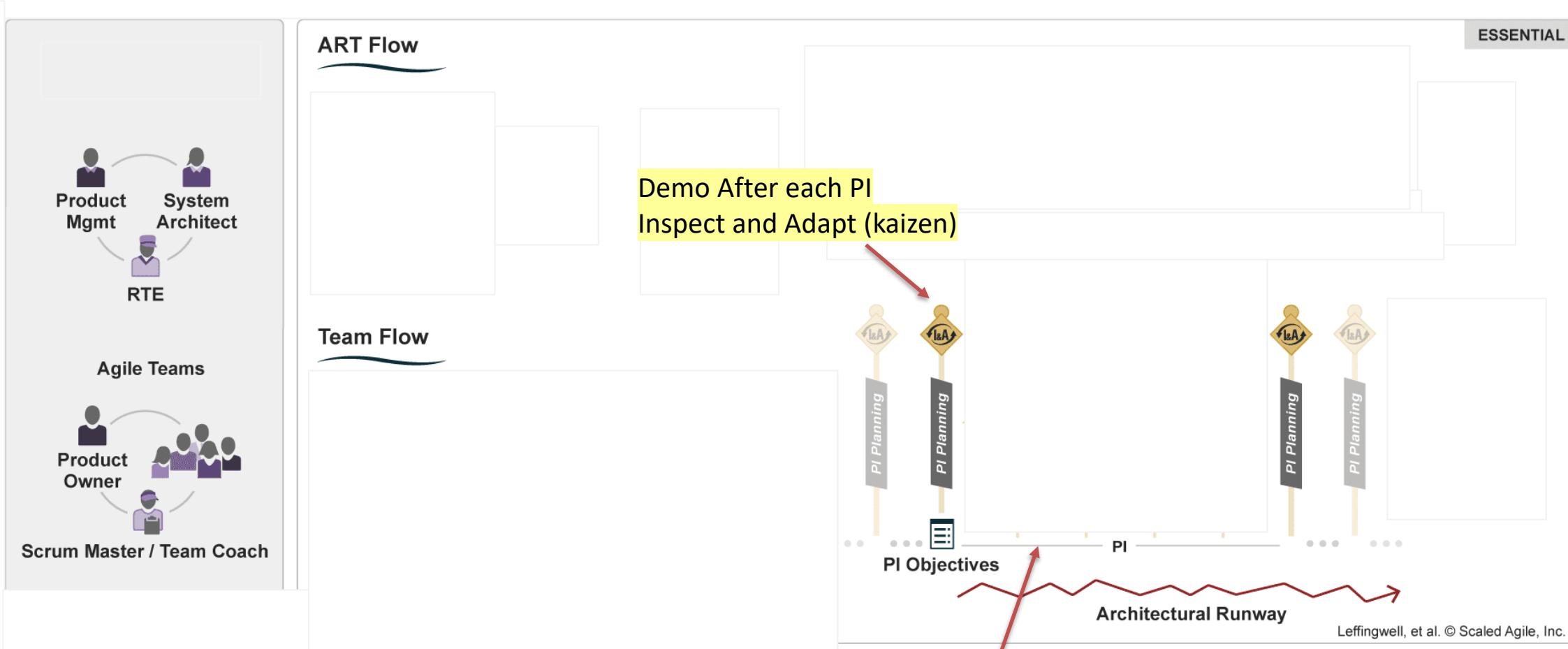
The Scaled Agile Framework

<https://www.scaledagileframework.com>



The Scaled Agile Framework

<https://www.scaledagileframework.com>

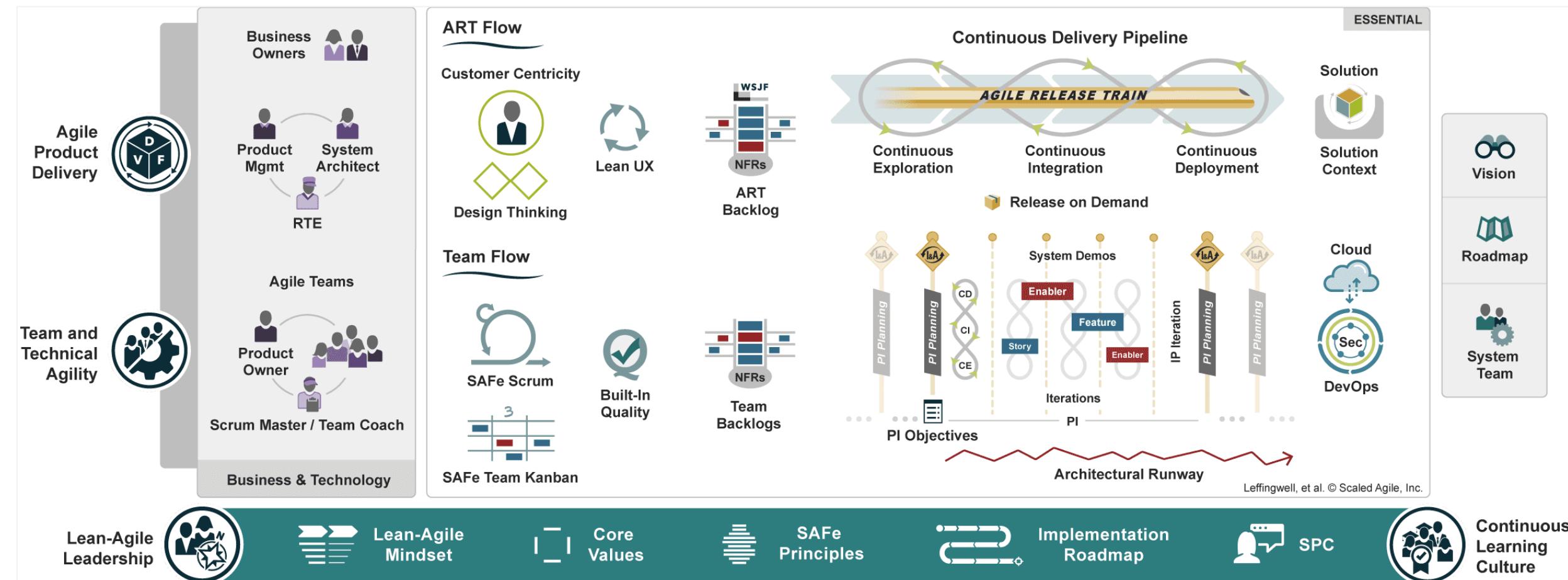


Planning: 8-12 week Planning Intervals (like “stages”)

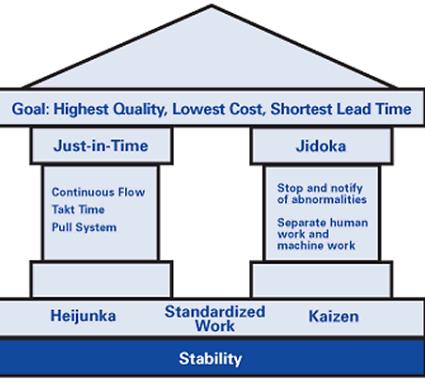
The Scaled Agile Framework

A set of organization and workflow patterns intended to guide enterprises in scaling lean and agile practices

<https://www.scaledagileframework.com>

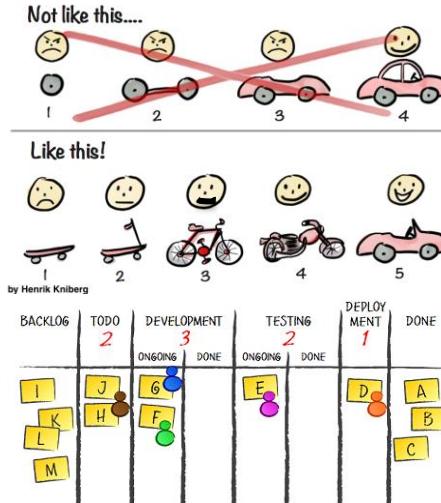


Summary



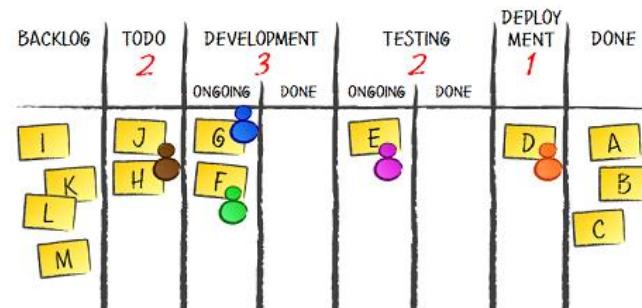
- **Lean**

- Lean house: JIT, Automation
- Eliminate waste
- **Maximise flow of value**
- e.g. MVP, Kanban



- **Agile**

- Focus on flexible scope, incremental delivery and trusting people
- e.g. Scrum, Kanban



- **Other Implementations:**

- Scrumban
- XP, Crystal, FDD, TDD
- Scaling Agile: DAD, SAFe



Agile

- Embrace **change**
- Short **cycles**
- Focus on **team**

Lean

- Focus on **value**
- Eliminate **waste**
- Limit work **queues**
- Improve **whole system**

Deliver **quickly**
Improve **quality**
Amplify **learning**
Continuously **improve**
Empower **people**

Guest Lecture



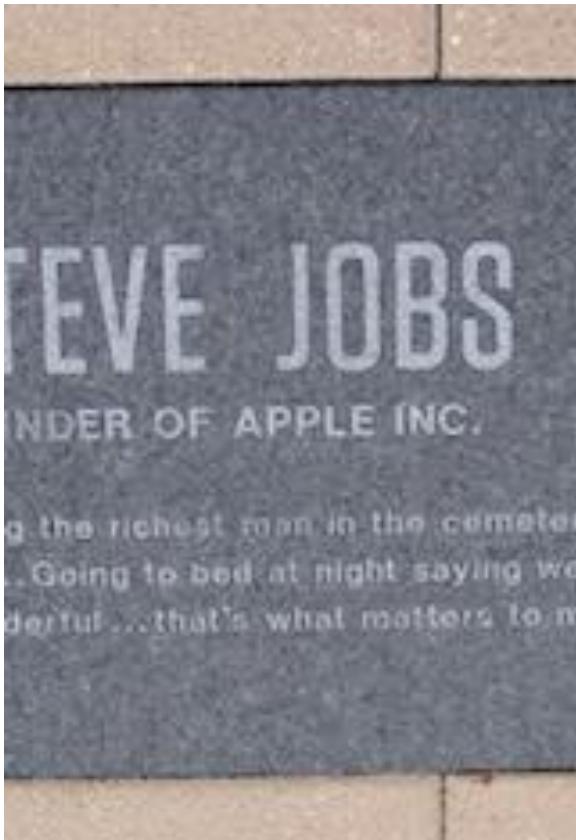
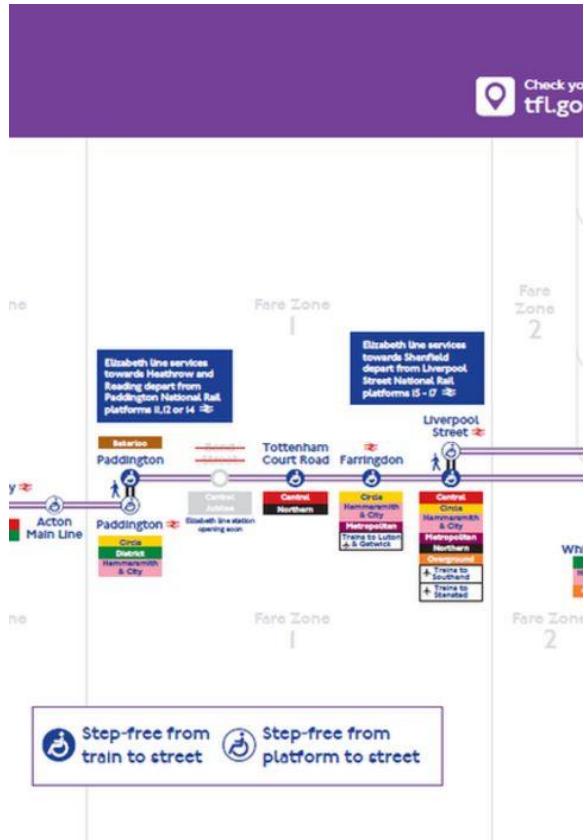
Catherine Attenborough

Programme manager, Product Division

KANTAR



Project Management is all around us





What do you think is the best recent tech invention?



Truth #1

Project Management is a Team Sport and the PM is the Captain



Truth #2

Agile is best for software development



Which of the statements below are true?

Vote for up to 5 choices

- 1. Agile is a silver bullet.**
- 2. Agile means no documentation.**
- 3. Agile means no planning**
- 4. Agile is undisciplined**
- 5. Team members get to do whatever they like**

The background of the slide features a scenic landscape with mountains and a forest silhouette at sunset. The sky is a warm orange and yellow, transitioning into a darker red and purple towards the horizon.

*Discipline is the
bridge between goals
and accomplishments.*

- JIM ROHN

Truth #3
Agile = discipline

Some examples of how we plan and manage our work at Kantar

Feature Kanban Board in ADO

Azure DevOps kantarware / KT-Platform-MRP Syndicated / Boards / Boards

As a stakeholder, you can access the backlog, task and Kanban boards, work items and manage approvals for Releases. [Learn more](#)

BrandNow

Board Analytics Epic/Feature Timeline Feature Timeline Epic Roadmap ...

Features

New On Hold 5/5 Requirement Elaboration 6/2 Architecture Des

1214634 Extraction of additional data from CINT State On Hold EDP MH milky-way postMVP ros-hnatyuk 22/25

1738515 Incremental Data Load for nfield Acquisition pipeline State Requirement E... Yana Yakob Jonathon 01 - Priority Data Team 0/1

1699822 Additional Bar chart representation for L2 and L3 pages - Current Performance State On Hold

1730346 [EDP/BrandNow]: Deliver Non-ABS survey changes (Familiarity, Buzz, Touchpoints) State

1701690 system backup/recovery State

Feature definition in ADO

FEATURE_1738515

1738515 Incremental Data Load for nField Acquisition pipeline

Yana Yakob 1 comment 01 - Priority Data Team

Status Requirement... Area KT-Platform-MRP Syndicated\BrandNow
Reason Moved to state... Iteration KT-Platform-MRP Syndicated\BrandNow

Description

Scope:
Data Load in nField Acquisition Pipeline is partially incremental due to nfield specifics. We can invest to see if better incremental approach can be achieved

Implement Incremental Data Load for nField Acquisition pipeline.
Currently data load for nField Acquisition pipeline is partially incremental even though historical data for the nField Acquisition pipeline is immutable thereby impacting processing time and storage costs. We can avoid unnecessary processing of historical data and thereby save processing time and storage needs by going for an full Incremental Data Load for the nField Acquisition Pipeline.

Planning

Program Increment PI Commitment 2023 Q4
T Shirt Estimation

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Outstanding Work Sprints

Development

+ Add link

Link an Azure Repos commit, pull request or branch to see the status of your development. You can also create a branch to get started.

Risk

Priority 2

Effort

Business Value

Related Work

Epic Kanban Board in Aha!

KANTAR Syndicated Brand Notebook Strategy Releases Ideas Features Roadmaps

Epics board Add epic Add release

Views Search epics (Ctrl+F)

Syndicated POC Syndicated MVP Syndicated Post MVP - Q3 Q4 2023 Q1 2024

SYNBRN-E-77 In Development [PRODUCT] Syndicated Enablement - MVP 8 features Epic: 1275389

SYNBRN-E-45 In Development Download 8 features Epic: 1133439

SYNBRN-E-55 Closed Epic: 1442637

SYNBRN-E-80 In Development [EDP] Platform Orchestrator MVP 4 features Epic: 1171278

SYNBRN-E-85 Closed [Prod] Provision Prod Environment 2 features Epic: 1442637

SYNBRN-E-136 New Tech Debt - Data 16 features Epic: 1751350

SYNBRN-E-116 In Development Incremental Data Load - MVP 0 features Epic: 1710692

SYNBRN-E-97 Requirement Elaboration Client specific filters (MVP) 1 feature Epic: 1577604

SYNBRN-E-119 Requirement Elaboration BrandNow Additional Visualisation (Bar Charts, Table Views)-Q4 0 features Epic: 1577604

SYNBRN-E-64 Closed Epic: 1442637

Epic definition in Aha!

KANTAR Syndicated Brand Notebook Strategy Releases

Epics board Add epic Add release

Views Search epics (Ctrl+F)

Epic SYNBRN-E-97

Client specific filters (MVP)

Key capabilities of this feature include:

- The ability of a Kantar user to:
 - Add a new filter(s) to a specific cell e.g. UK Banking
 - Define the variables each filter should be based on
 - Female AND 18-35 AND High income
 - Live in New York OR New Jersey OR Illinois
 - The ability to select which client organization can see which predefined filters. For example, filters added for NatWest in UK Banking, will only be visible to NatWest UK banking
 - The next time data is processed, the data pipeline will calculate TrendAI signals for the additional filters. This will also include
 - Filters will appear on the dashboard the next time the data is reprocessed in the current filter selector
 - Filters will appear on historical data as well, not just from the day they get added.
 - When the user selects the filter, the dashboard will show data for the specific filters
 - When a client does not renew, we have the ability to remove the filter

Architectural Considerations:

Significance testing of historical data: Currently significance testing is run in the EDP and stored in analysis tables. If we created a new filter, we won't have

OVERVIEW COMMENTS 2 TO-DOS RELATED 5 HISTORY



Questions

Week	Lectures		Seminars			Individual Report		
	Topic	Guest	Case Study	Exercises	Submission	Chapter	Submission	Marking
1	Specification			Specification				
2	Initiation		Selection		Pitch			
3	Scope / Time			Scope/Time				
4	PRINCE2	PRINCE2	Initiation			Ch.1 Initiation		
5	Budgeting			Budgeting				Self-assess
6	Lean/Agile 1	Waterfall / Agile	Planning			Ch.2 Planning		
7	Lean/Agile 2	Lean		Scrum/Kanban		Ch.1-2		
8	Risk	Risk / Finance	Monitoring				Review Ch.1-2	
9	Teamwork	Large Projects	Prepare Presentation	Risk		Ch.3 Execution		
10	Revision				Presentation	Ch.4 Monitoring	Ch.1-4	
11								
Term 2								Review Ch. 1-4

Interim Submission (Ch.1-2)

This one!

The screenshot shows a Moodle course interface. At the top, there's a navigation bar with the WARWICK logo, Dashboard, This Module, Help, More, and user icons for profile, settings, notifications, and student status. Below the navigation is a grid-based timeline or calendar view showing weeks 10, 11, and Term 2. Week 10 is highlighted in yellow and labeled 'Revision'. Week 11 has a grey box labeled 'Presentation'. Term 2 has a blue box labeled 'Presentation' and an orange box labeled 'Ch.4 Monitoring'. A small blue icon is visible on the left side of the main content area.

Individual Report - Submission and Peer Assessment

In addition to the team-based activities in this module, you will need to produce an **Individual Report**. You will work on this in your own time (independently of your team).

You will submit this report (including an interim submission) to Moodle. You are also required to mark the work of your peers and provide constructive feedback. See the [assignments](#) page for more details. You should also be aware of the [deadlines and penalties](#).

- [Self-Assessment \(Ch. 1\)](#)
- [How to mark the essay](#)
- [Interim Essay Submission and Peer Marking \(Ch.1-2\)](#)
- [Interim Essay Results \(Ch. 1-2\)](#)
- [Final Essay Submission and Peer Marking \(Ch.1-4\)](#)
- [Final Essay Results \(Ch. 1-4\)](#)

Construction Allusion