

Software Engineering

A Faculty of Engineering Course: CSEN 406

The Software Development Life Cycle (SDLC)

2

Dr. Iman Awaad

iman.awaad@giu-uni.de



Software Engineering

A Faculty of Engineering Course: CSEN 303

WS 2025/26 Lecture Information

Lecturer

Dr. Iman Awaad

iman.awaad@giu-uni.de

Office hours: Sundays, 10:15 - 11:15



Thursdays, 12:00 - 13:00



Acknowledgments

The slides are **heavily** based on the **slides** by **Prof. Dr. John Zaki**.

They are also **heavily** based on the slides and textbook by **Ian Somerville**.

Their contribution is gratefully acknowledged.

Any additional sources are referenced.

Software Development Life Cycle (SDLC)

- Methodologies
 - Waterfall
 - Spiral
 - V-Model
 - Agile
 - Sprint, Retrospective Sprint, Review, Standup Meetings, Scrum, Master Product, Owner, Product Backlog, Sprint Backlog, MVP, Agile Team Formation
 - Extreme Programming

What methodologies exist for the planning, design, development, testing, and deployment of high-quality software?

What are “best practices” when developing a software product?

How do we track software versions?

What is SDLC?

....is a **structured process** that defines the **phases** involved in **planning, designing, developing, testing, and deploying** high-quality software.

What is SDLC?

...is a **structured process** that defines the **phases** involved in **planning**, **designing**, **developing**, **testing**, and **deploying** high-quality software.

- **Systematic approach** to develop **high quality** software that meets the **necessary business requirements**
- **Phases** of software (S/W) development
- Each phase has its own **process & deliverables**
- A **cycle** of processes...

Benefits of SDLC?

iterations

facilitate communication

right product

minimize cost

high quality

minimize risk

discrete phases

transparency

improve efficiency

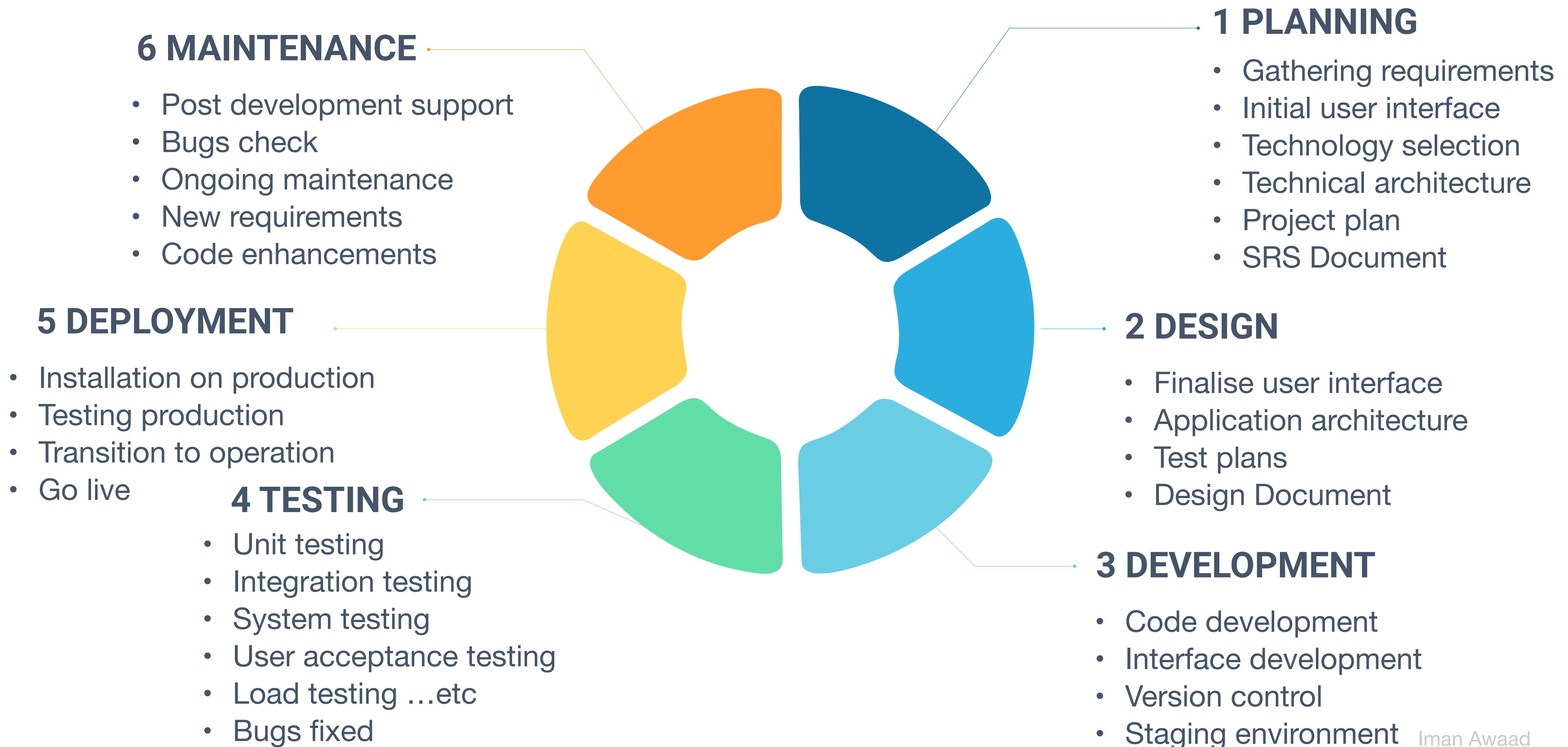
Phases of SDLC...

...is a **structured process** that defines the phases involved in **planning, designing, developing, testing, and deploying** high-quality software.

+ **maintenance**

Phases of SDLC

domain-independent & follow the complete software lifecycle



Software Methodologies

There is **no one-size-fits-all** methodology. How to manage the development process may depend on several factors, such as:

- **size** of the team — larger teams may need more fine-grained management than smaller teams
- associated software **risks** and **costs** (robots interacting with people pose different risks than extraterrestrial exploration robots)
- **availability of users** during the development process (frequent user feedback will shape the development differently than feedback only at specific milestones)...

Software Methodologies

There is **no one-size-fits-all** methodology. How to manage the development process may depend on several factors, such as:

- **size of the team** — larger teams may need more fine-grained management than smaller teams
- associated software **risks** and **costs** (robots interacting with people pose different risks than extraterrestrial exploration robots)
- **availability of users** during the development process (frequent user feedback will shape the development differently than feedback only at specific milestones)...

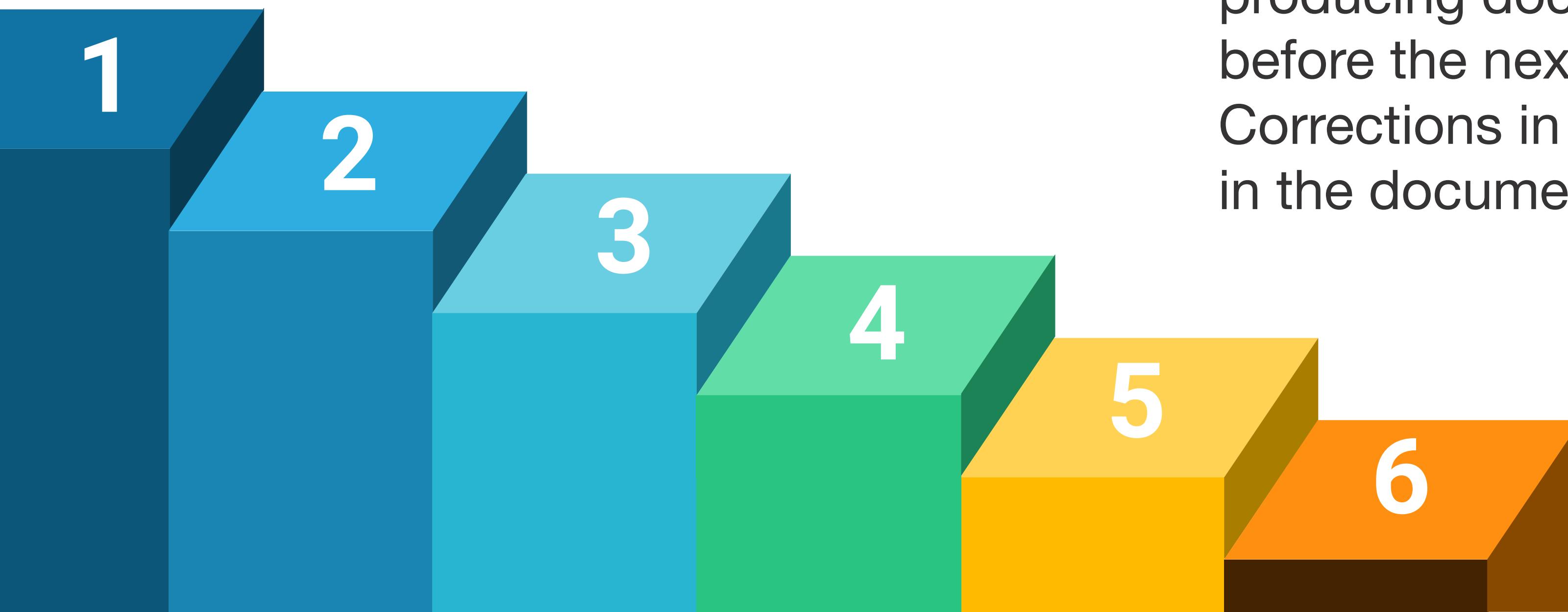
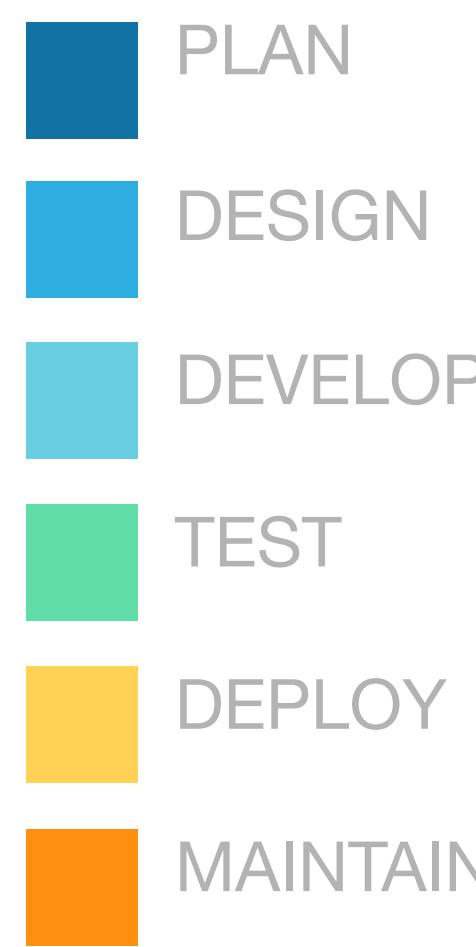
Waterfall model
...sequential

Spiral model
...risk-driven

V-Model
...incremental

Agile approach
...flexible,
collaborative

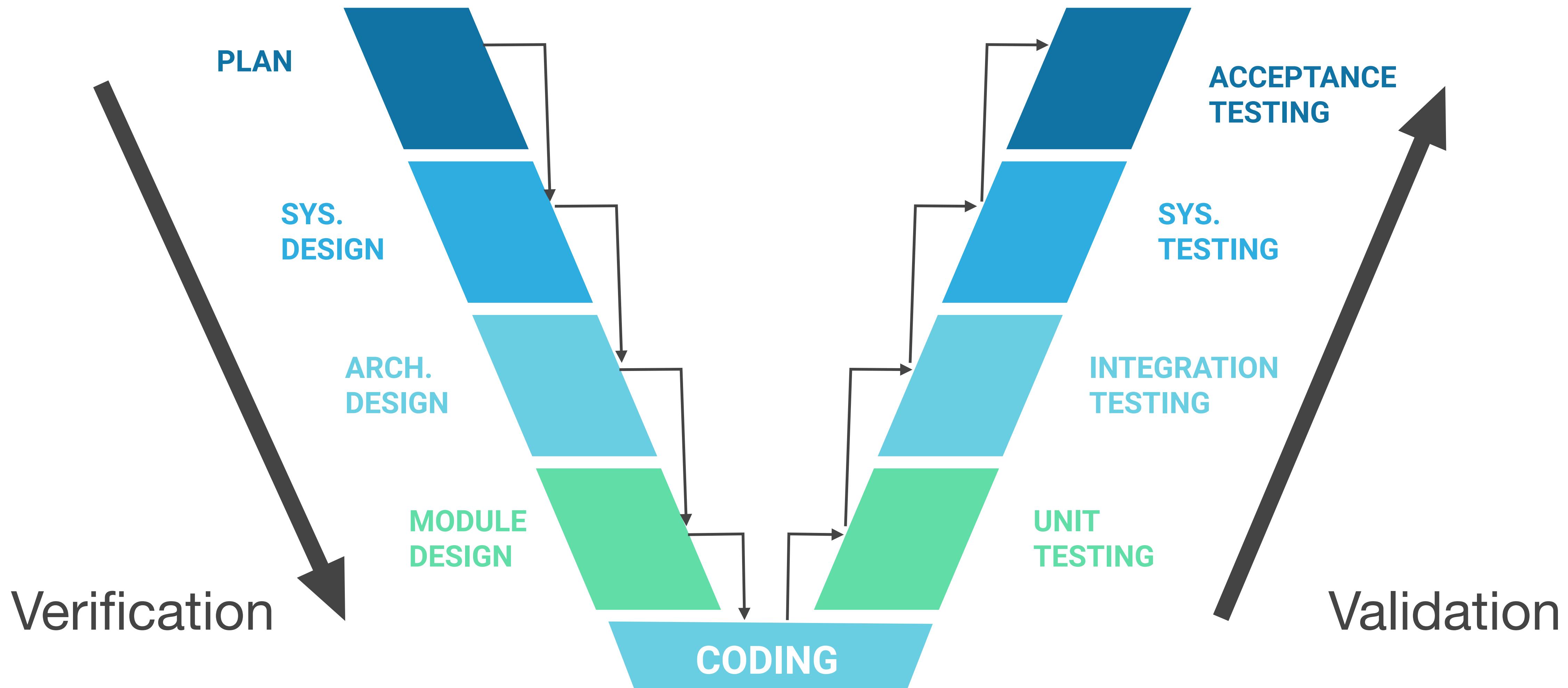
Waterfall method



- Development **proceeds to the next step only when the current step is completed**
- But later stages may notice issues with the previous steps, which then requires going a step back and reworking certain aspects
- The completion of each stage is followed by producing documentation that is approved before the next stage of the process can start. Corrections in previous steps require changes in the documentation as well.

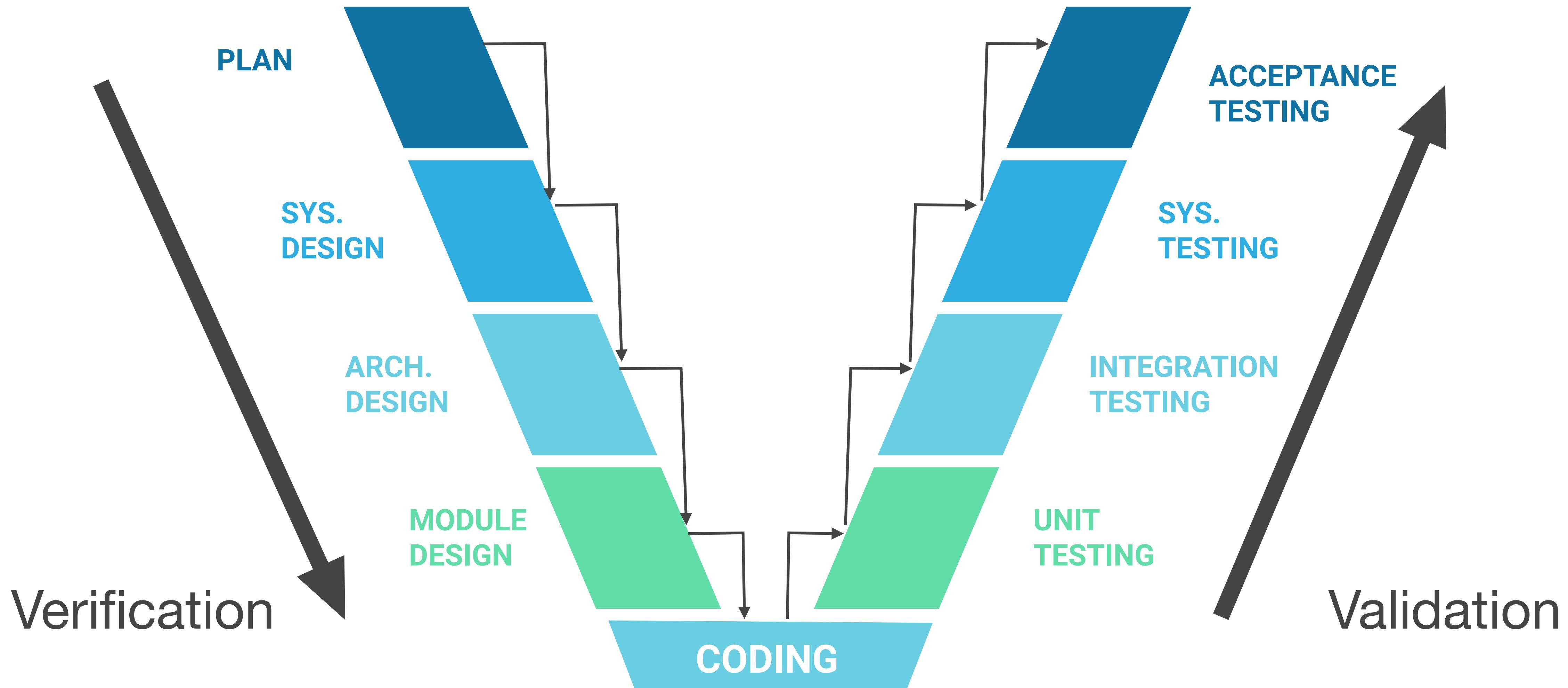
...a plan-driven process

V-Shape method



V-Shape method

How is this different than the waterfall method?!



Take a minute...

A product manager is working with a customer to prioritize requirements??

- Design
- Planning
- Development

Take a minute...

The QA team assesses the code released to the staging platform against use cases. They discover one requirement that has not been addressed...

- Maintenance
- Deployment
- Testing

Take a minute...

The software engineers release the code into the production environment

- Deployment
- Development
- Maintenance

Can we do better than plan-based development?

- Software company managers need to understand:
 - **how much it costs** to develop a software product
 - **how long it will take**
 - **when the product can be brought to market**
- **Plan-driven development**: provides this info via **long-term development plans** that identify **deliverables**: items the team will deliver and when these will be delivered.
- **Plans change!** Long-term plans are unreliable...

Agile method

- Software products must be **brought to market quickly** ...rapid software development and delivery!
- **Virtually all** software products are now developed using an **agile approach!**
- Numerous ‘agile methods’ developed: all based on **incremental development and delivery**
- No ‘best’ agile method/technique
- Depends on who is using the technique, the development team and the type of product being developed...
- Product development focuses on **software features** (feature does something for the user)
- Start by prioritizing the features: most important features implemented first
- Only define the details of the feature being implemented in an increment
- That feature is then implemented and delivered
- Users or surrogate users can try it out and provide feedback to the development team. You then go on to define and implement the next feature of the system.

Agile method

“We are uncovering better ways of developing software by doing it and helping others do it.

Through this work **we have come to value:**

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

Kent Beck

Mike Beedle

Arie van Bennekum

Alistair Cockburn

Ward Cunningham

Martin Fowler

James Grenning

Jim Highsmith

Andrew Hunt

Ron Jeffries

Jon Kern

Brian Marick

Robert C. Martin

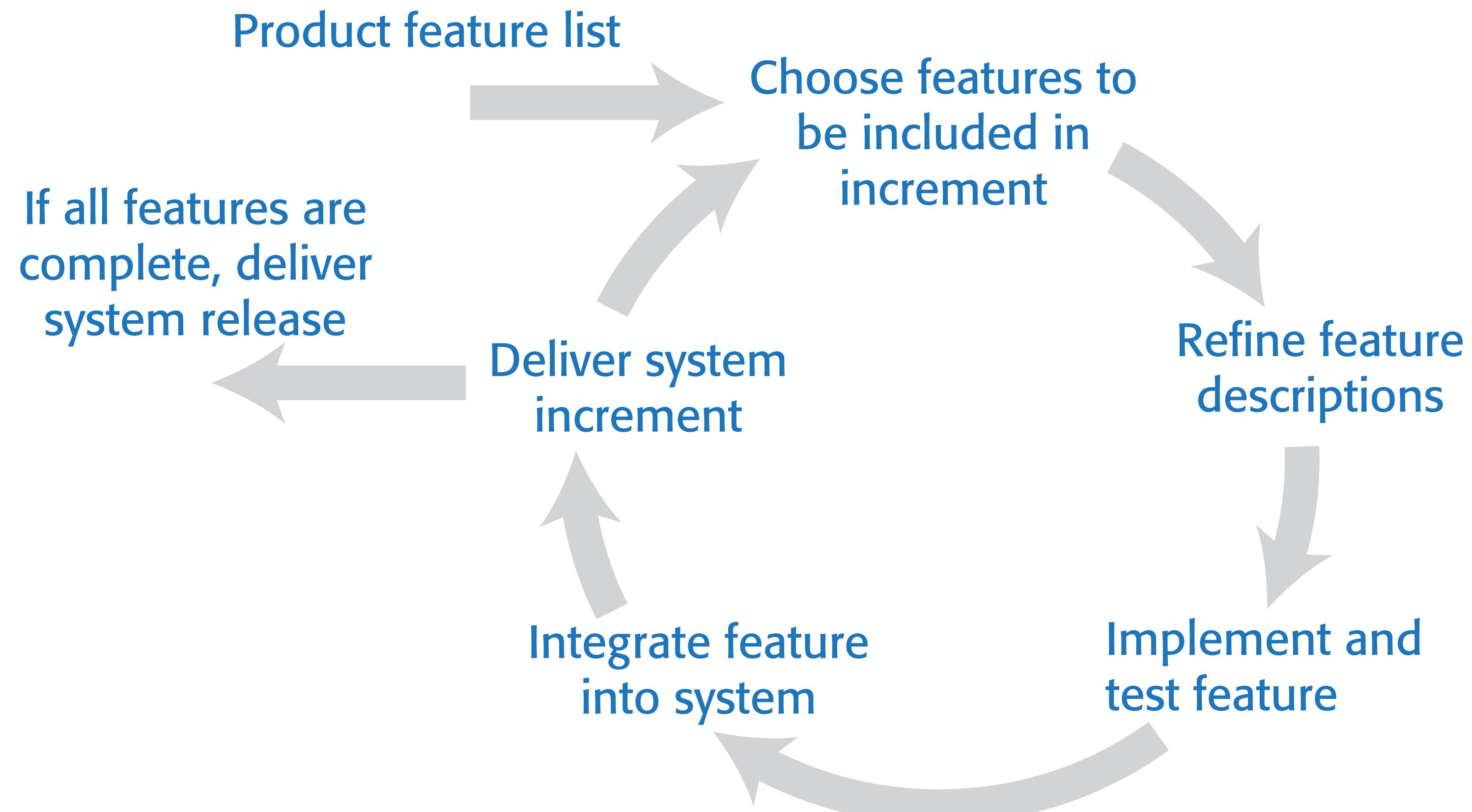
Steve Mellor

Ken Schwaber

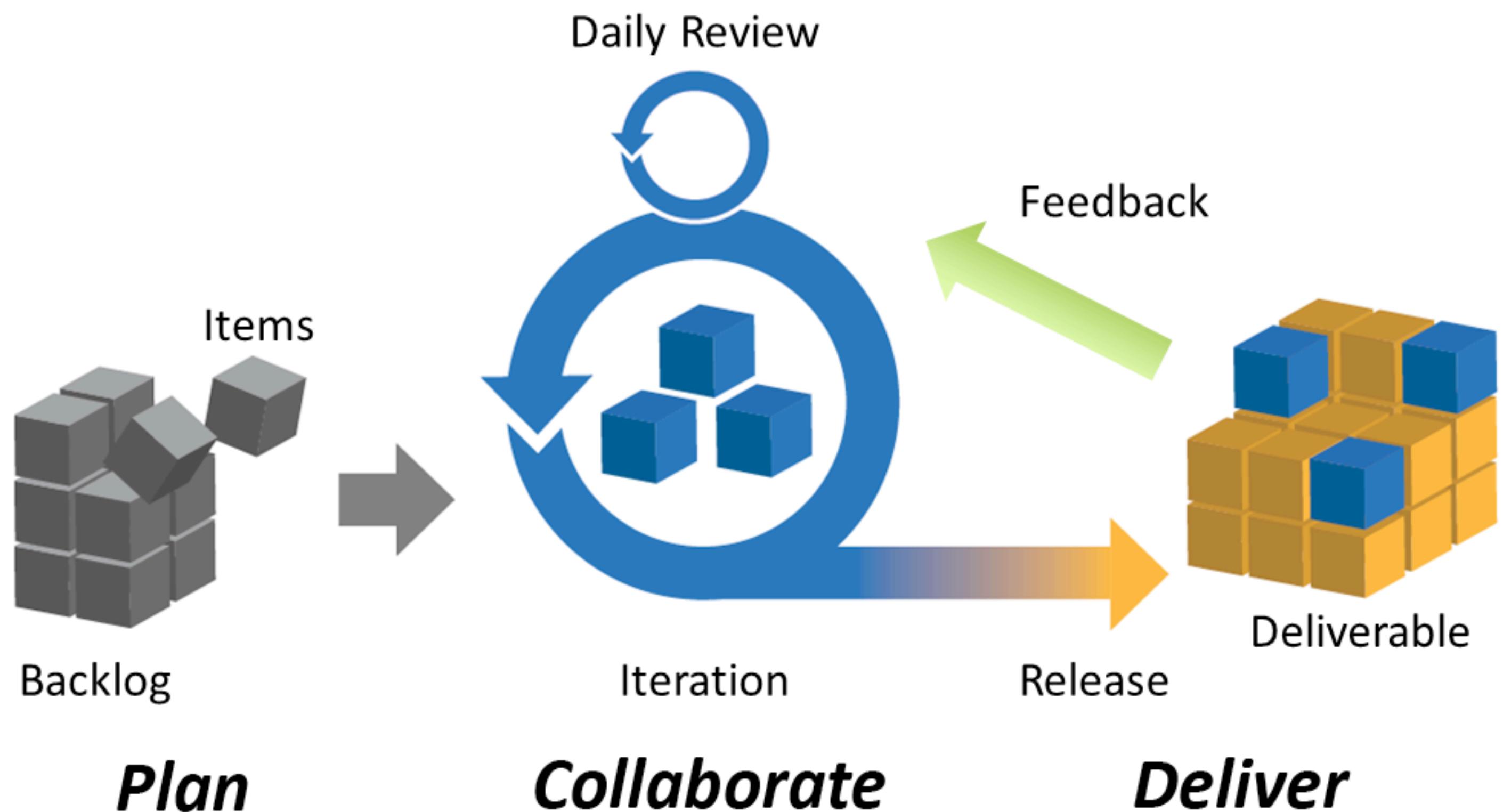
Jeff Sutherland

Dave Thomas

Agile method



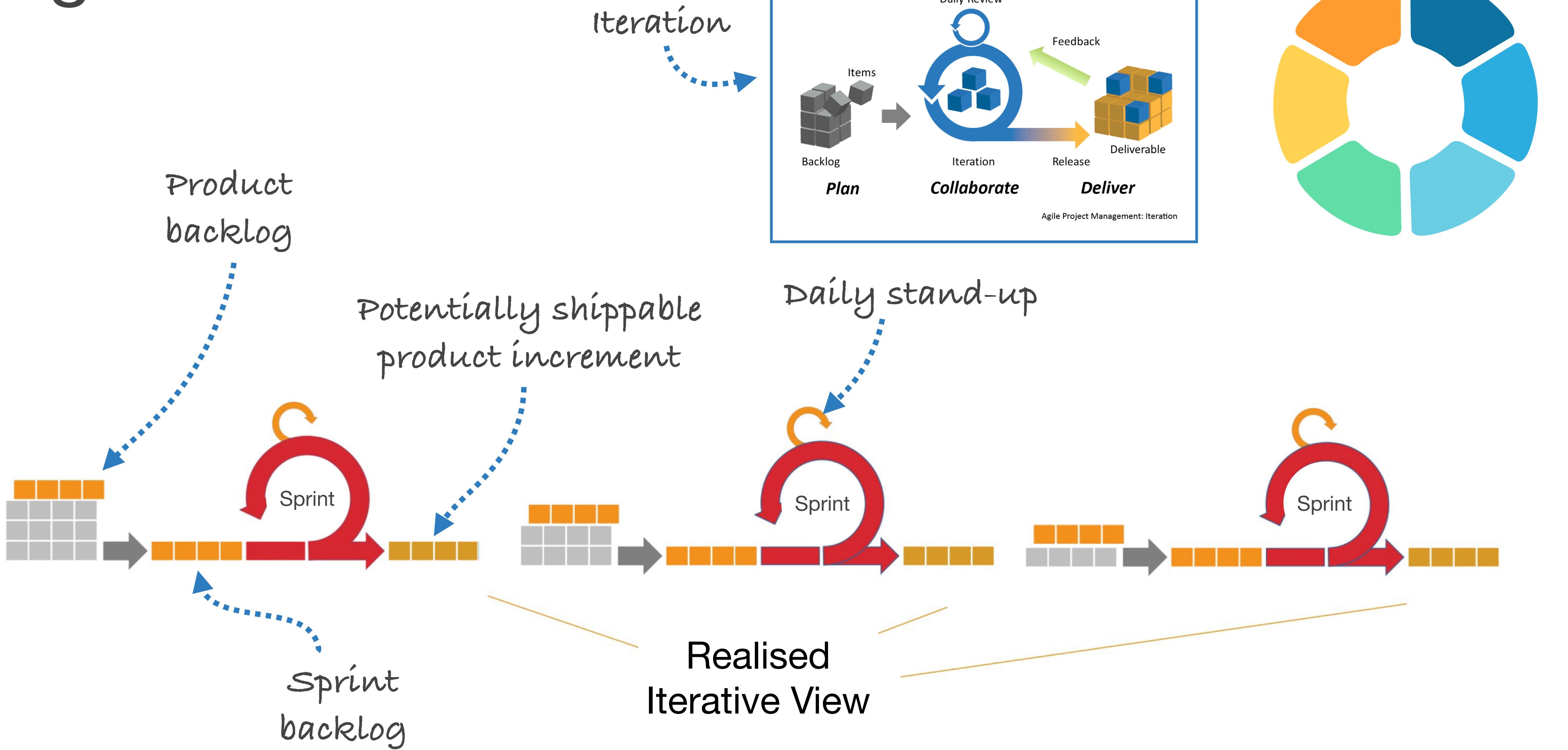
Agile method



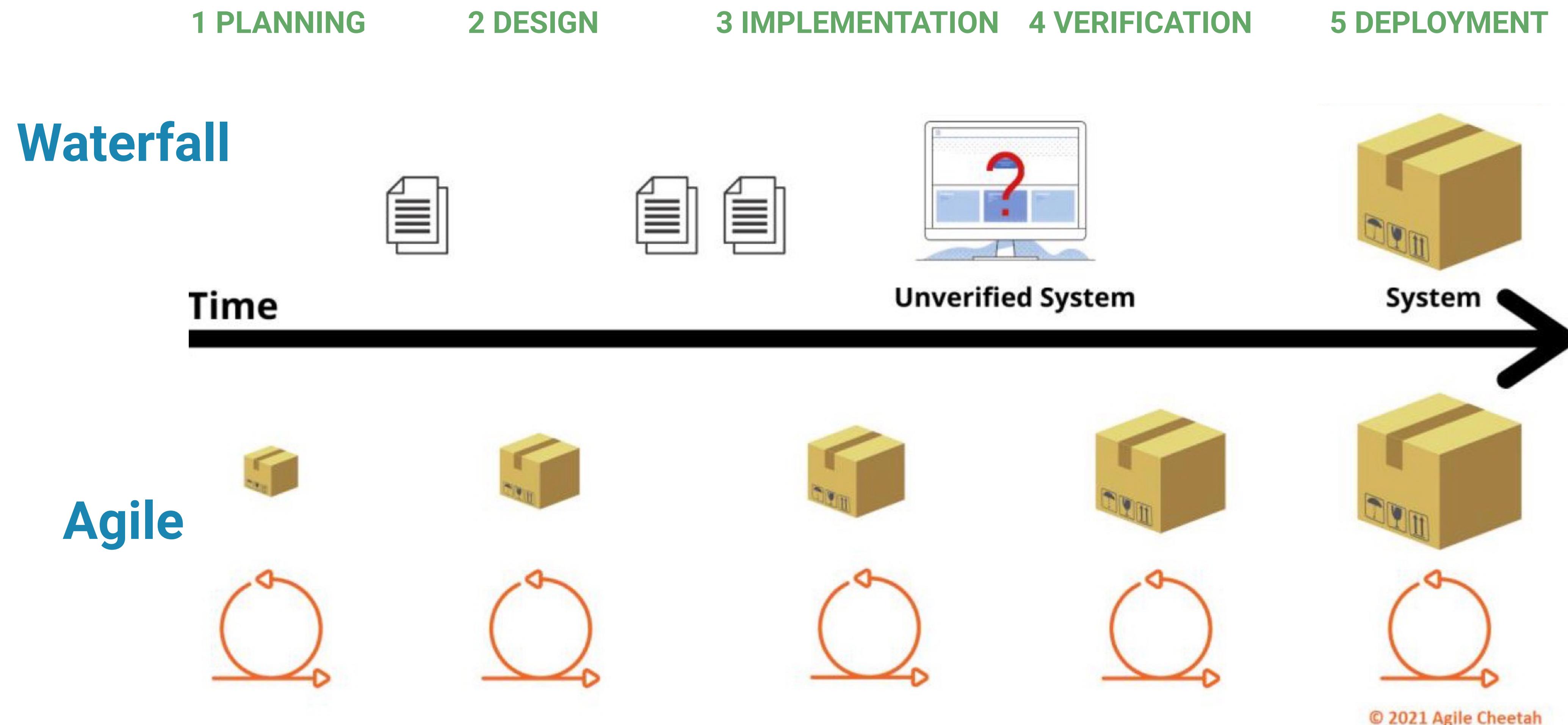
Agile Project Management: Iteration



Agile method



How do they compare re: product shipping?



When is one better when?

Take a minute...

What are the pros and cons
of Agile v Waterfall methods?

Take a minute...

What does the following terminology mean?

Sprint

Retrospective

Sprint Review

Standup Meetings

Scrum Master

Product Owner

Product Backlog

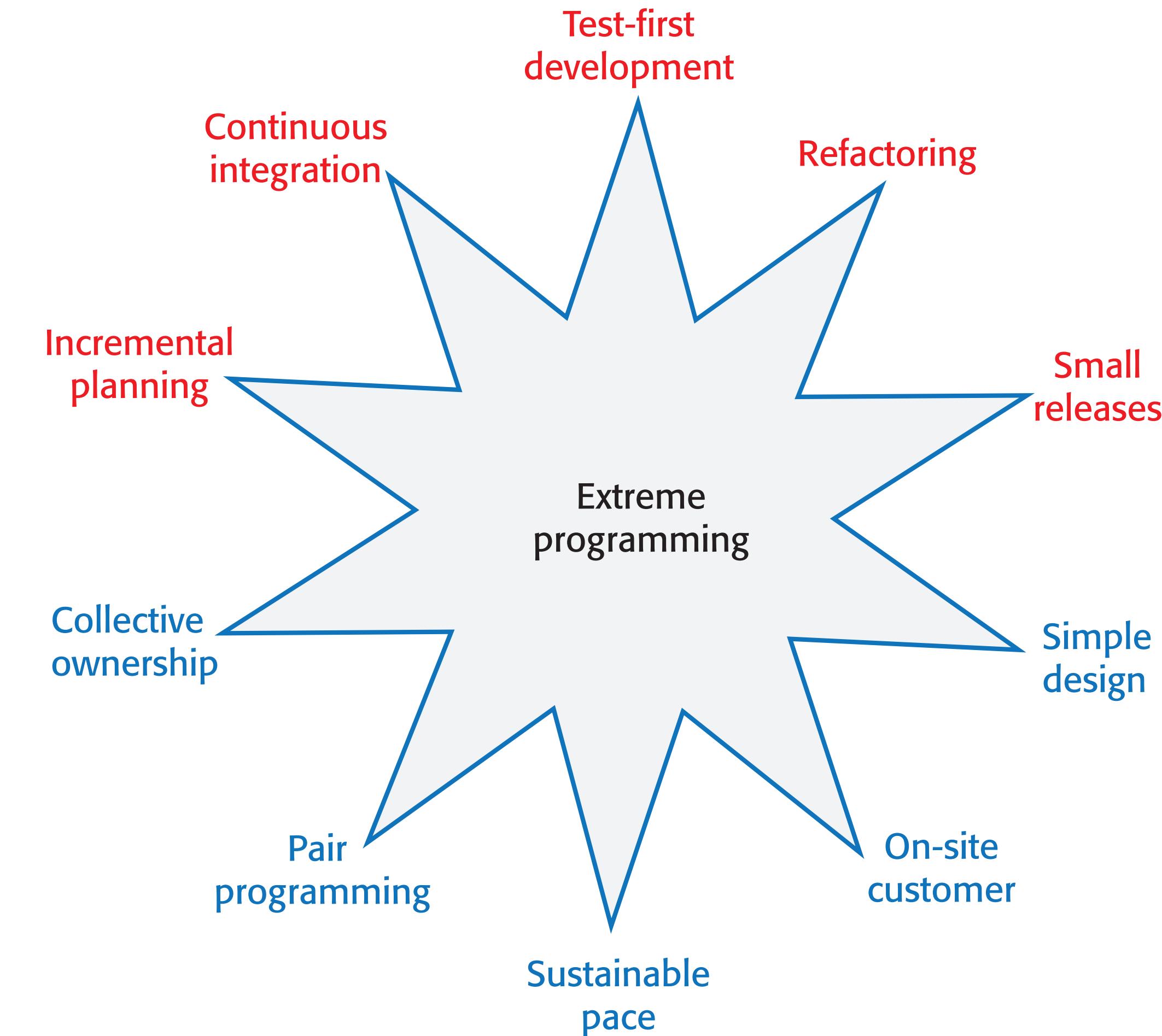
Sprint Backlog

MVP

Extreme Programming (XP)

(Kent Beck, 1998: pushing good practice, e.g. iterative development, to 'extreme' levels)

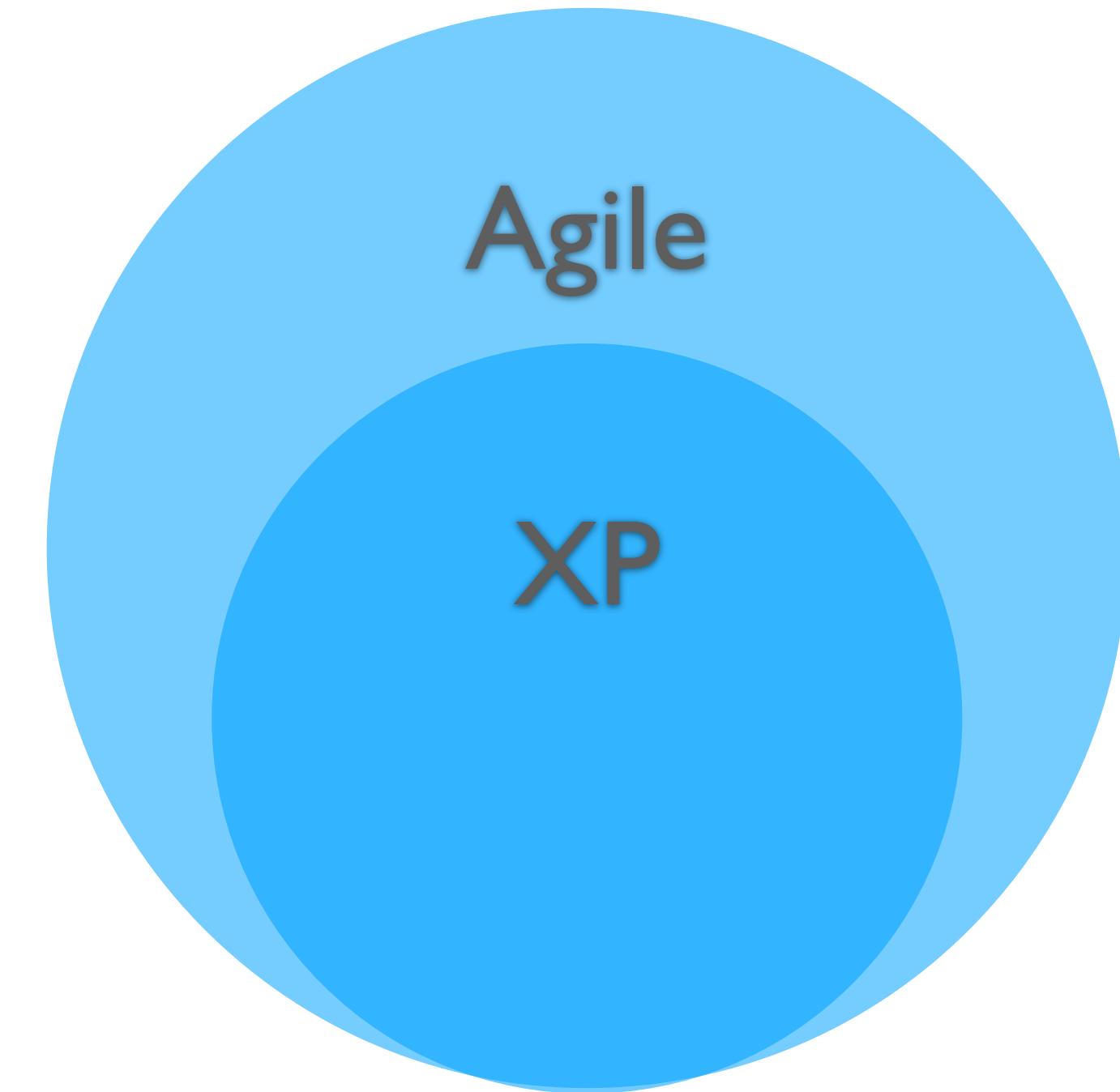
- **Most influential** concept to change software development culture!
- Focused on 12 “new” development techniques for rapid, incremental software development, change & delivery...
- Today, some are widely used; others are less popular
- Most **widely-used** techniques:



Extreme Programming (XP)

(Kent Beck, 1998: pushing good practice, e.g. iterative development, to 'extreme' levels)

- **Most influential** concept to change software development culture!
- Focused on 12 “new” development techniques for rapid, incremental software development, change & delivery...
- Today, some are widely used; others are less popular
- Most **widely-used** techniques:



Extreme Programming (XP)

Incremental planning/user stories

- No ‘grand plan’
- What needs to be implemented (requirements) in each increment via discussions with a **customer representative**
- Requirements are written as “**user stories**”: included in a **release** determined by the **time available** and their **relative priority**

Small releases

- **Minimal useful set of functionality** that provides business value is **developed first**

- Frequent releases that add functionality to the previous release.

Test-driven development

- Developers **write the tests first!** This clarifies what it should do and that there is **always** a ‘**tested**’ version of the code available!
- An **automated unit test** framework is used to run the tests **after every change**. New code should **not ‘break’** code that has already been implemented!

Extreme Programming (XP)

Continuous integration

- As soon as the work on a task is complete, it is **integrated** into the whole system and a **new version** of the system is created
- **All unit tests** from all developers are **run automatically** and **must** be successful before the new version of the system is accepted!

Refactoring

- ...**improving** the **structure**, **readability**, **efficiency** and **security** of a program.

- All developers **must refactor** code as soon as possible code improvements are found. This keeps code **simple** and **maintainable**.

What is Scrum?

...is an **agile method** that provides a
framework for **agile project organization**
and **planning**.

Scrum Terminology

Product [software product being developed by the Scrum team]

Product owner [Team member responsible for identifying product features & attributes. They review work done and help to test the product]

Product backlog [To-do list of items e.g. bugs, features and product improvements that the Scrum team have not yet completed]

Development team [Small, self-organising team of 5 to 8 people responsible for developing the product]

Sprint [short period, usually 2 to 4 weeks, when a **product increment** is developed]

Scrum [Daily team meeting where progress is reviewed and work to be done that day as discussed and agreed]

ScrumMaster [Team coach who guides the team in the effective use of Scrum]

Potentially shippable product increment [Output of a sprint which should be of high enough quality to be deployed for customer use]

Velocity [An estimate of how much work a team can do in a single sprint]

Take a minute...

What does the following terminology mean?

Sprint

Retrospective

Sprint Review

Standup Meetings

Scrum Master

Product Owner

Product Backlog

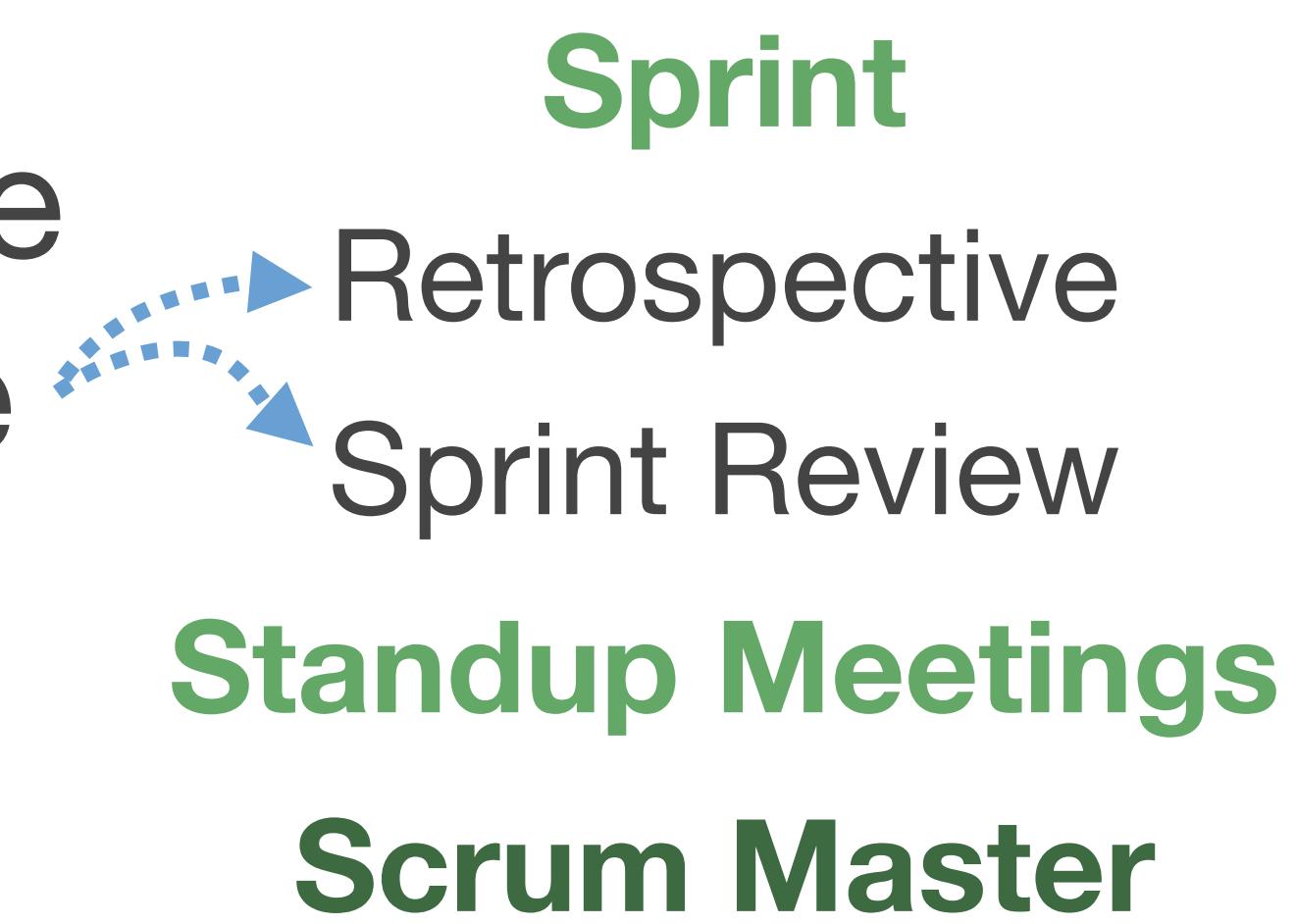
Sprint Backlog

MVP

Take a minute...

What does the following terminology mean?

What is the difference between



Review
inspects the **outcome**
(the product)

...includes **external stakeholders**;

...aims to adapt the
product and plan future
work

Retrospective
...inspects the **process**
(how the team worked)

...is **for the internal Scrum Team only**

...aims to adapt the
team's processes

Take a minute...

What does the following terminology mean?

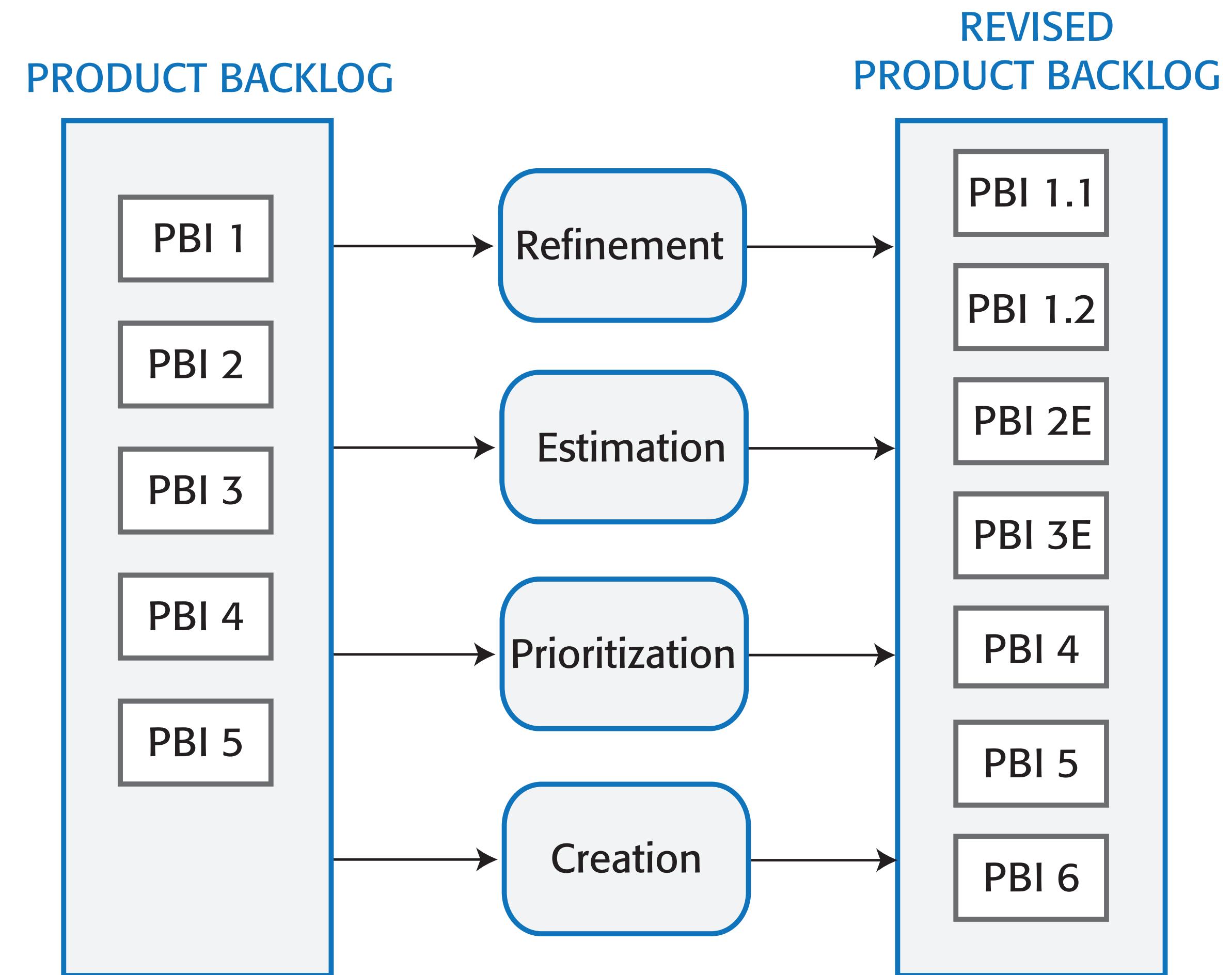
What is the
difference
between

Sprint
Retrospective
Sprint Review
Standup Meetings
Scrum Master

Product Owner
Product Backlog
Sprint Backlog
MVP
What is

Product Backlog Items (PBI)

1. As a teacher, I want to be able to configure the group of tools that are available to individual classes. (feature)
2. As a parent, I want to be able to view my childrens' work and the assessments made by their teachers. (feature)
3. As a teacher of young children, I want a pictorial interface for children with limited reading ability. (user request)
4. Establish criteria for the assessment of open source software that might be used as a basis for parts of this system. (development activity)
5. Refactor user interface code to improve understandability and performance. (engineering improvement)
6. Implement encryption for all personal user data. (engineering improvement)



PBI states: **Ready for consideration** **Ready for refinement** **Ready for implementation**

Iman Awaad

Product Backlog Items (PBI)

Refinement

Existing PBIs are analysed and refined to create more detailed PBIs. This may lead to the creation of new product backlog items

Estimation

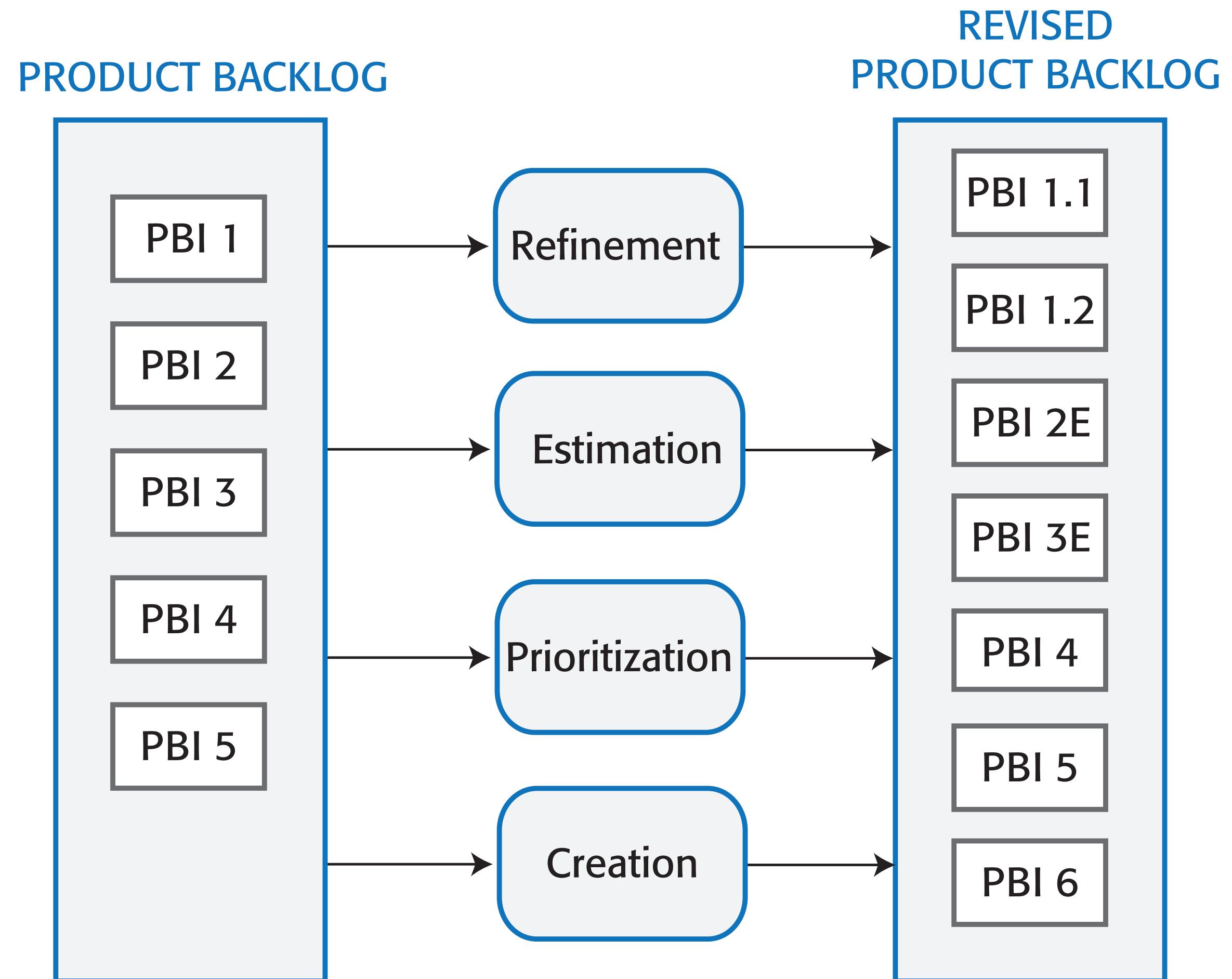
The team estimate the amount of work required to implement a PBI and add this assessment to each analysed PBI (in person-hours or person-days)

Creation

New items are added to the backlog. These may be new features suggested by the product manager, required feature changes, engineering improvements, or process activities such as the assessment of development tools that might be used

Prioritization

The product backlog items are reordered to take new information and changed circumstances into account



Project management software systems

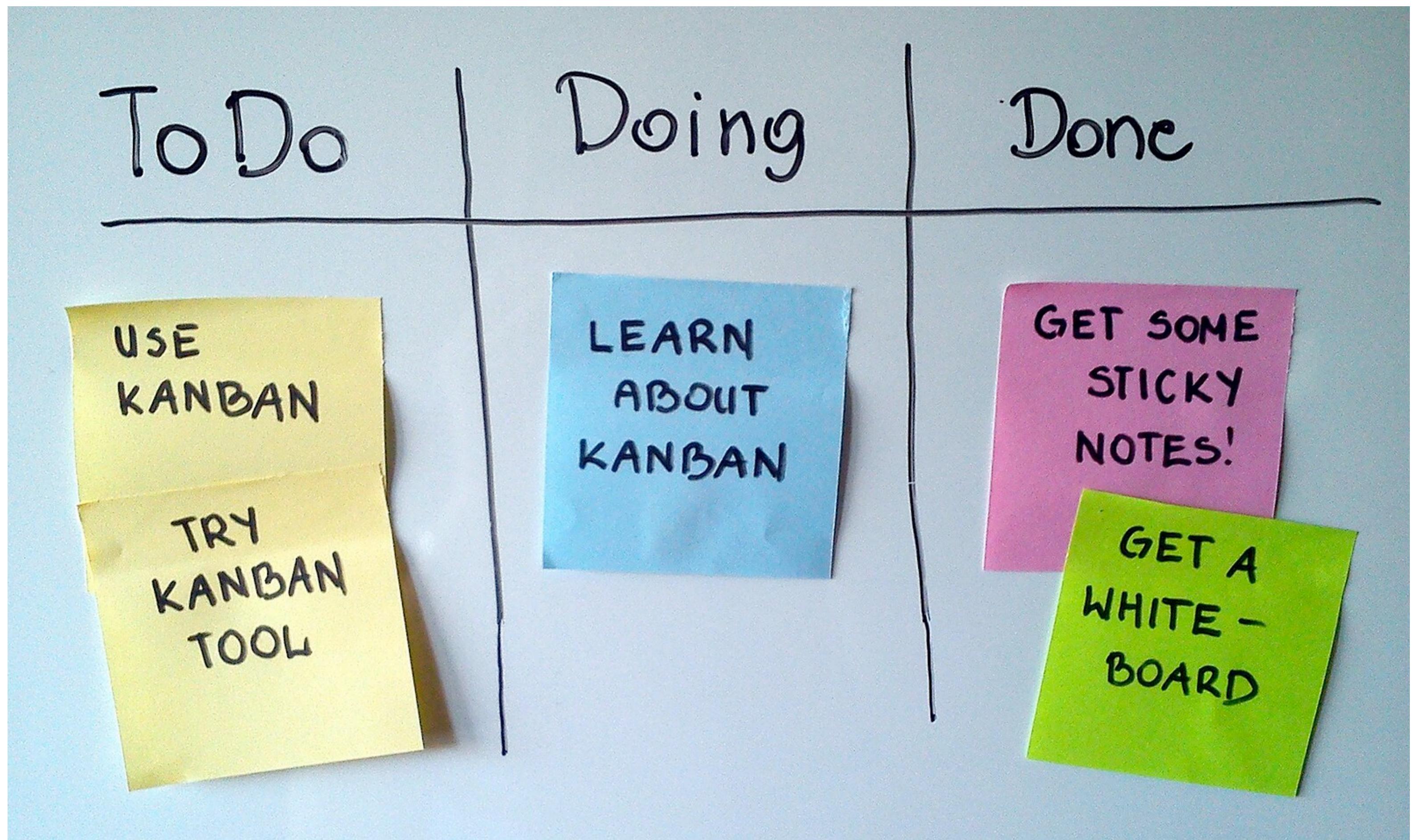
- In most product development companies, development teams need to **report on progress** to company management...
- A self-organizing team has to **appoint someone** to take on these responsibilities.
- Must maintain continuity of communication with people outside the group, **rotating** these activities around team members is **not a viable approach**.
- The developers of **Scrum** did not envisage that the **ScrumMaster** should also have project management responsibilities... but it happens!

ScrumMaster takes on project management!

- They know the work going on and are best-placed to provide accurate information and project plans and progress.

What is Kanban?

... a **visual workflow management method**, originating from the Toyota Production System (TPS), that **helps teams manage their work by visualizing tasks on a board**.



Take a minute...

What are the differences and similarities
among Extreme Programming (XP), Kanban, and Scrum?



REAL LIFE EXAMPLES

Dr. JOHN ZAKI

Philips

Philips is another firm that has adopted Agile principles. After numerous changes to management structure, the firm introduced several Agile coaches that went to deploy Scrum principles such as Scrum boards and breaking down teams into smaller ones. As a result of changes like this, teams could react to situations quicker, bureaucracy was removed, and it was ultimately easier for these smaller teams to take responsibility for their respective products.

VistaPrint

VistaPrint is the go-to marketing company for small businesses. The company performed some analysis of their existing waterfall methodology and found that teams were taking more than 60 days to move from the “ideation phase” to product delivery! The 60-day cycle only amounted to 40 hours of actual work!

The company looked further into why this was the case, it transpired that unclear decisions and long creative lead times were to blame, all of which resulted in feedback loops. Round and round it went!

VistaPrint decided to switch to Agile practices with a focus on decreasing project lead time. By introducing daily stand-ups, Kanban Boards and idea pipelines, [**VistaPrint**](#) was able to optimize their business processes and saw their Lead Time decrease from 40 days down to 15!

- The Key to Bosch's Sustained Success: Bosch's journey to embrace Agile Project Management commenced with CEO Volkmar Denner's epiphany that genuine business agility was essential to stay abreast of the ever-evolving market demands. The company initially experimented with a 'dual organization' model, where only certain groups within the organization would become agile. However, it soon became apparent that this approach wouldn't deliver the desired results.
- The Decision to Go All In: Recognizing the need for a more comprehensive change, Bosch implemented Agile Project Management across the entire organization, including the leadership team. This shift was a game-changer, allowing the company to break down silos, foster cross-functional collaboration, and create a more responsive and adaptable organization.
- Bosch's Agile Leap in Decision-Making: One of the most remarkable shifts at Bosch was the transition towards continuous planning and funding cycles. This enabled the company to adapt and respond to market conditions more effectively. Bosch's Agile principles have led to heightened efficiency and innovation across the organization by focusing on simplicity and prompt decision-making.
- Bosch's Agile Project Management Success Stories: Bosch's commitment to Agile Project Management has yielded impressive results in various business sectors. In their automotive group, development time was slashed in half, earning them Tesla's 'Excellent Development Partner Award.' Their home and garden division witnessed a surge in innovation, efficiency, and employee engagement through the formation of cross-functional teams. Moreover, their agricultural sector saw an impressive acceleration in innovation rates, rolling out ten new systems in a mere four weeks.

Software Versions

Version Number

How SW developers keep track of releases, updates, and bug fixes

Track Update

Mostly follow a semantic 4 parts numbering separated by a dot *e.g.* V3.1

Track Release

Two or more sets of numbers.
Sets are separated by a dot. *e.g.* V1.0

Track Bug Fix

Version found in the about section or help of any software *e.g.* V3.1.33

Software Versions

Version Number

How SW developers keep track of releases, updates, and bug fixes

Track Update

Mostly follow a semantic 4 parts numbering separated by a dot *e.g.* V3.1

Track Release

Two or more sets of numbers.
Sets are separated by a dot. *e.g.* V1.0

Track Bug Fix

Version found in the about section or help of any software *e.g.* V3.1.33

V3.1.33.6?

Build No., Build date, or less significant changes

Software Versions

Version Number

How SW developers keep track of releases, updates, and bug fixes

Compatibility

Older versions may not work well in newer versions

Track Update

Mostly follow a semantic 4 parts numbering separated by a dot e.g. V3.1

Troubleshooting compatibility issues using software versions

Track Release

Two or more sets of numbers.
Sets are separated by a dot. e.g. V1.0

Backward compatibility

...older software can work on new versions

Track Bug Fix

Version found in the about section or help of any software e.g. V3.1.33

V3.1.33.6?

Build No., Build date, or less significant changes

“If you only do what you can do,
you will never be more than you
are now”

— *Master Shifu*
in Kung Fu Panda III

Reproduced from:
< <https://wallpapercave.com/master-shifu-wallpapers> >

