

# Session 2

## Agile software development

Dr. Bharati Wukkadada

# Agenda for the day

- Agile Development:
  - Why agile
  - What is Agility
  - Agile process,
  - Challenges in Adopting Agile Methods
  - Agile process models,
  - Extreme programming overview,
  - Scrum Overview,



## Agile means “move quickly”

---

The Agile methodology is a practice that encourages continuous development and testing throughout the software development lifecycle of a project. Unlike the waterfall methodology, the Agile methodology

---

Agile methodologies attempt to produce the proper product through small cross-functional self-organizing teams that produce small pieces of functionality on a regular basis, allowing for frequent customer input and course correction as needed.

---

Agile tries to address the issues that traditional "waterfall" methodologies of delivering huge products over extended periods of time encounter, such as client requirements changing frequently and resulting in the delivery of incorrect products. allows for parallel development and testing.

# Why agile

# Agile software development

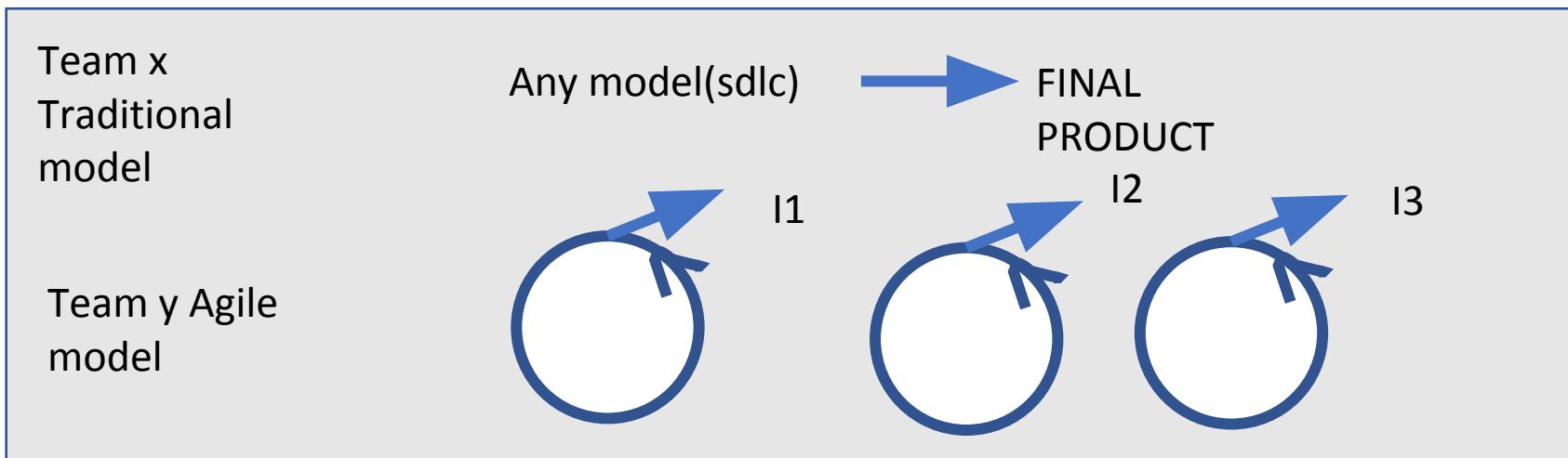
---

***“Thinking outside the box”  
for the 21st century, or the  
key to success? Let’s get to  
the bottom of agile project  
management.***

---

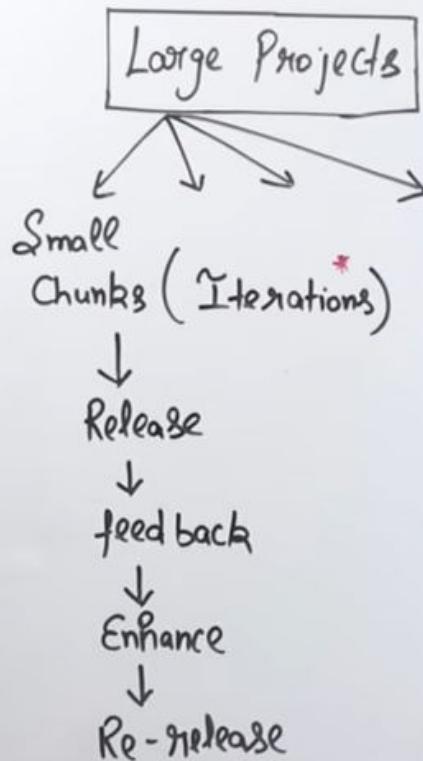
In 2001 , Kent Beck and 16 other noted s/w developers, writers, and consultants signed the “manifesto for the agile software development”. It stated:

- Agile is a philosophy to rapidly deploy an application in much more organised way
- Agile model changes can easily incorporate- quick appropriate response to changes incorporate the changes in development eg



# Basic model

"Agile" (Move Quickly)



- Advantages:
- 1) Frequent Delivery
  - 2) Face to face communication with client
  - 3) Changes
  - 4) Time
- Disadvantage:
- 1) Less documentation
  - 2) Maintenance Problem

# AGILE METHODOLOGY

— ● ● ● ● —  
**The Agile Process**



# What is it?

- Agile s/w engg combines a philosophy and a set of development guidelines.
- The philosophy encourages
  - cust satisfaction and early incremental delivery of s/w,
  - small highly motivated project teams
  - informal methods,
  - minimal s/w engg work products and
  - overall development simplicity.
- The development guidelines stress delivery over analysis and design and active and continuous communication between developers and cust.

## Who does it?

- s/w engineers and other project stakeholders ( **managers, customers , end users** ) work together on a agile team- a team that is self-organizing and in control of its own destiny.
- An agile team **fosters communication** and **collaboration** among all who serve on it

## Why it is imp?

- Modern business environment is computer-based and s/w products is fast-paced and ever-changing.
- agile is reasonable to conventional
- It has been demonstrated to deliver successful systems quickly

# Agile Manifesto

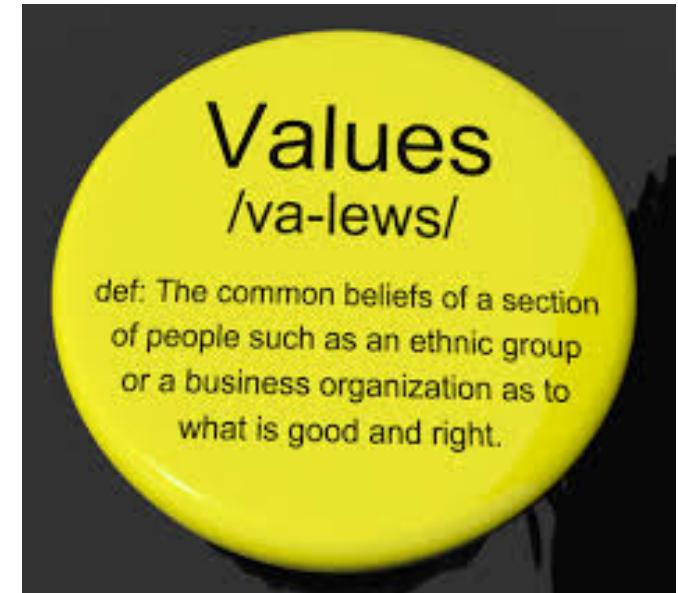
a written statement declaring publicly the intentions, motives, or views of its issuer.  
How to use manifesto in a sentence

- The **Agile Manifesto** is a document that sets out the key values and principles behind the **Agile** philosophy and serves to help development teams work more efficiently and sustainably. Known officially as 'The **Manifesto for Agile Software Development**', the **manifesto** detailing **4 Values and 12 Principles**.

# The Agile Manifesto 4 values

- **Individuals and interactions**
  - Over processes and tools
- **Working software**
  - Over comprehensive documentation
- **Customer collaboration**
  - Over contract negotiation
- **Responding to change**
  - Over following a plan

**Values** are basic and fundamental beliefs that guide or motivate **Values** in a narrow sense is that which is good, desirable, or worthwhile. **Values** are the motive behind purposeful action



We are covering better ways of developing s/w by doing it and helping others do it. Though this work we have come to value:

- Individuals and interactions over processes and tools
- Working s/w over comprehensive documentation
- Cust collaboration over contract negotiation
- Responding to change over following a plan

1



Individuals and interactions



Processes and Tools.

2



Working Software



Comprehensive Documentation

3



Customer Collaboration



Contract Negotiation

4



Responding to a change



Following a plan

## **12 principles of Agile**

The 12 principles of Agile, according to the Agile Alliance, are as follows:

1. Satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements.
3. Deliver working software frequently.
4. Work together daily throughout the project.
5. Build projects around motivated individuals who are supported and trusted to get the job done.
6. Use face-to-face conversation whenever possible.
7. Working software is the primary measure of progress.
8. Maintain a constant pace indefinitely.
9. Give constant attention to technical excellence and good design.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. Reflect on how to become more effective, then tune and adjust accordingly at regular intervals.

# principles



# 4 Key Agile Values

Individuals & Interactions  
over processes and tools



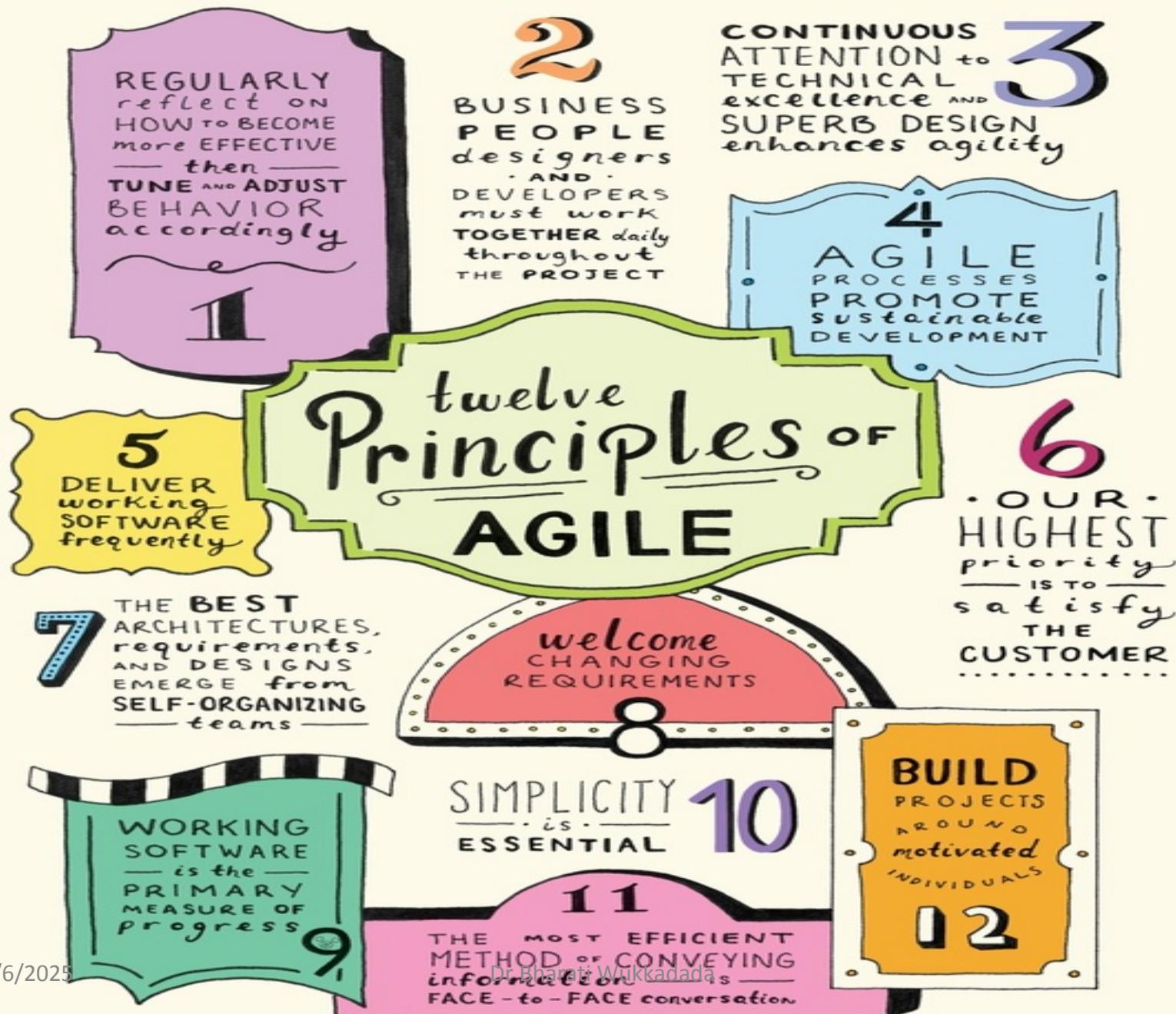
Customer Collaboration  
over contract negotiation

Working Software  
over comprehensive documentation



Responding to Change  
over following a plan





# Benefits of Agile



Transparency



Predictable Costs and  
Schedule



Allows for Change



Improves Quality

## The definition of Agile project management

- Agile project management is an iterative development methodology that values
- **human communication**
- **feedback,**
- **adapting to change, and**
- **producing working results.**

- Agile is **iterative**, meaning that it is done in pieces **SPRINTS**, with each sprint building and improving off the lessons from the previous sprint.

This is where that term Scrum comes into play. **Scrum** methodology is a workflow framework made up of sprints and reviews used to promote Agile project management.

- Unlike Scrum, which can be distilled to a step-by-step process, Agile is an **approach and a mindset**. It's not a textbook, or a list of instructions, or a certification. In fact, trying to turn Agile into a black-and-white template goes against everything that Agile is. It would be like trying to give someone a step-by-step plan on how to be “cool,” or play jazz.

- . Agile project management is all about **efficient communication** over documentation, convoluted email chains, or excessive meetings. According to **the 12 principles behind the agile manifesto**: “The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation.**” If you can communicate something with a 10-second conversation instead of an email, you should.
- . Agile is about **producing tangible, working results** after each iteration. According to the 12 principles, “**Working software** is the primary measure of progress.” To compare Agile to the editorial process—you deliver a rough draft, then revise based on your editor’s suggestions. You’re not delivering the entire piece all at once on the day it goes to press.

# What is Agility???

- You should implement Agile project management at your business **if you want to be successful**. If your business is not using Agile, you're in the increasing minority, and you're **falling behind as a result**.
- According to Project Management institute **more than 70% of organizations** have incorporated an Agile approach, and Agile projects **are 28% more successful** than traditional projects.

- Agile -better equipped **to help implement Agile practices** at your organization, and recognize situations that could be improved with a dash of Agile( like fixing the weekly meeting-planning meeting )
- We'll also look at **three real-world examples** of Agile project management in action.



# Real-life examples of Agile project management

---

## The build-your-own meal

- Everyone is familiar with the build-your-own-meal trend in fast, casual dining. At restaurants such as Chipotle or Subway, an employee puts your meal together as you give feedback.
- More cheese? Less cheese? Different bread? Guacamole? No guacamole? No problem.
- Every step of the way, your food project manager checks in with you to make sure your food project is still on track. The end result is a delicious meal that was improved during each step thanks to constant face-to-face collaboration.

## 2. The Apple Genius Bar

- Looking past the pretentiousness of the name, the Apple Genius Bar is a great, real-world example of Agile project management in action.
- When you come in with your busted iPhone or iPad, you don't have to fill out a bunch of forms or wait in a series of lines (it's more like a waiting gathering). It's a world apart from your last experience at the DMV.
- What makes the Genius Bar an Agile process is the focus on communication. The associate you deal with asks you questions and takes notes. In other words: "individuals and interactions over processes and tools."
- You may be saying, "But Apple uses processes and tools, like the iPad they take notes on."
- Yes, but the conversation between humans comes first.

The Genius Bar is a technical support service provided by Apple Inc. inside Apple Stores to support the use of its products and services. The locations provide concierge-style, face-to-face support for customers from "Geniuses" who are specially trained and certified by Apple, with multiple levels of certification depending on the products serviced.

### 3. Baseball

- You may think that this is a stretch but stick with me here.
- A baseball manager has to be an Agile project manager to succeed. Every season is a major project made up of 162 games plus playoffs, and each game is an iteration of that project.
- Imagine if a baseball manager put the same players in the same positions, batting in the same order for all 162 games despite injuries, poor performance, or bad match-ups. That manager would probably not be very successful (even [Lou Piniella with the 2001 Seattle Mariners](#) needed to make adjustments here and there).
- In fact, Agile is all over baseball. Infield Scrum meetings at the pitcher's mound, phone calls to the bullpen (*not* emails), a concrete result (win or loss) at the end of every iteration (game).
- The next time you watch a baseball game, think about how your small-business team could operate more like a

**Ask question to any one???**

## **How Agile is your team?**

- If you just remember that Agile project management is about human-to-human communication, adapting to changing conditions, and producing working results, you'll be on the right track.
- But Agile project management is, by definition, ever evolving and changing. Ask 10 different project managers to define Agile and you'll probably get 10 distinct answers.

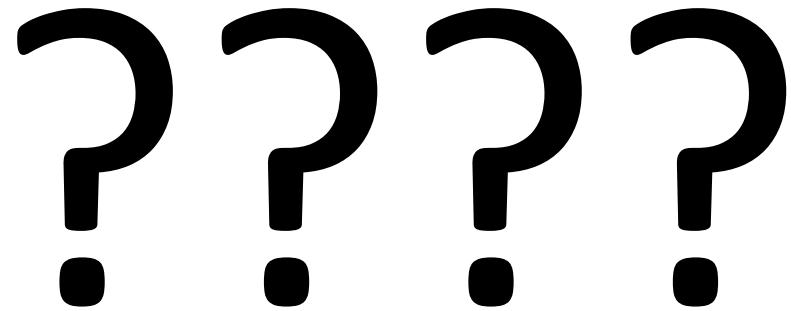
# Agile Vs Traditional Methods

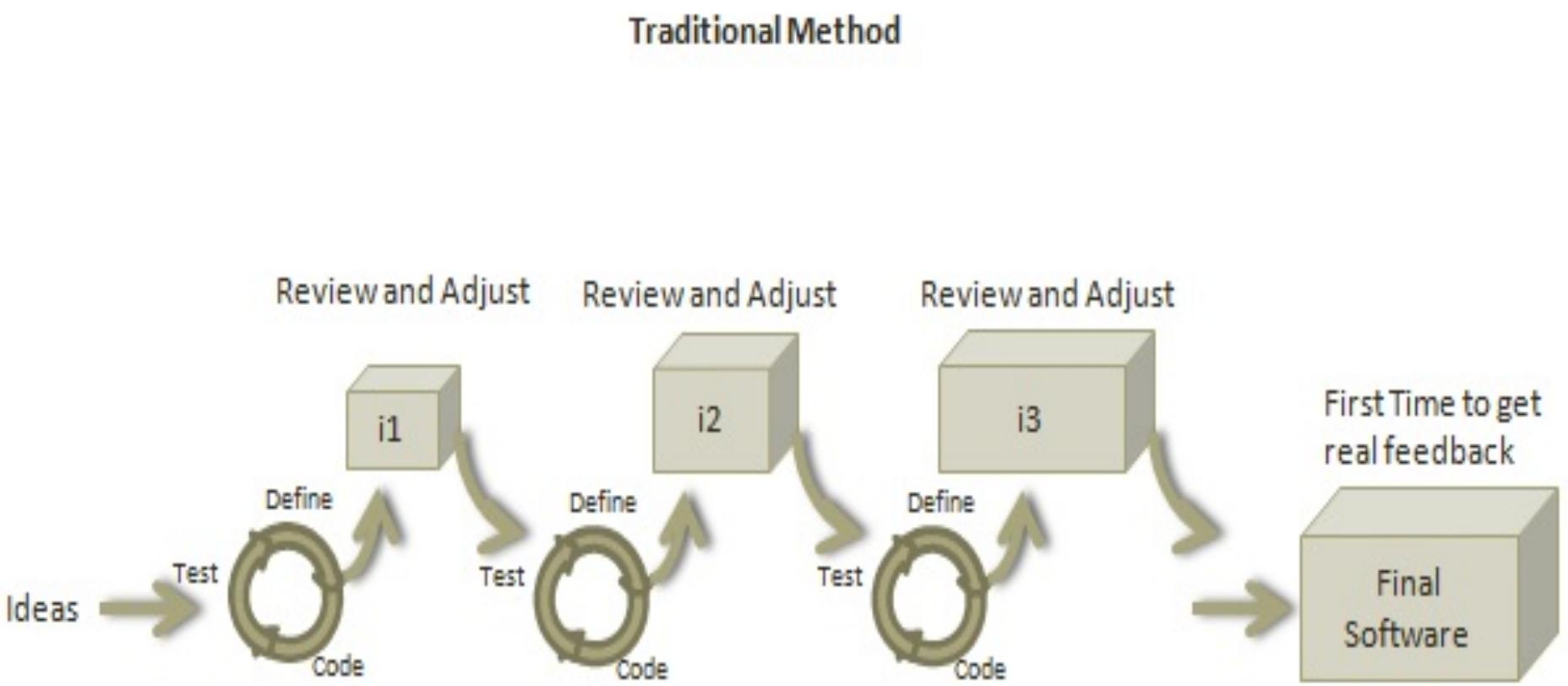
	Traditional Method	Agile Method
<b>Changes</b>	Resisted and Controlled	Welcomed and Adapted to
<b>Contracts</b>	Binding	Flexible
<b>Customer Interaction</b>	During selected milestones	Continuous
<b>Delivery</b>	One-time	Iterative and Incremental
<b>Orientation</b>	Process	People
<b>Process</b>	Heavyweight	Lightweight
<b>Requirements</b> <b>Architecture &amp; Design</b>	Defined Upfront	Evolves
<b>Roles</b>	More	Less
<b>Success Criteria</b>	Conformance to the Requirements	Business value delivered to the customer

# Agile vs Iterative vs Waterfall – {Planning}

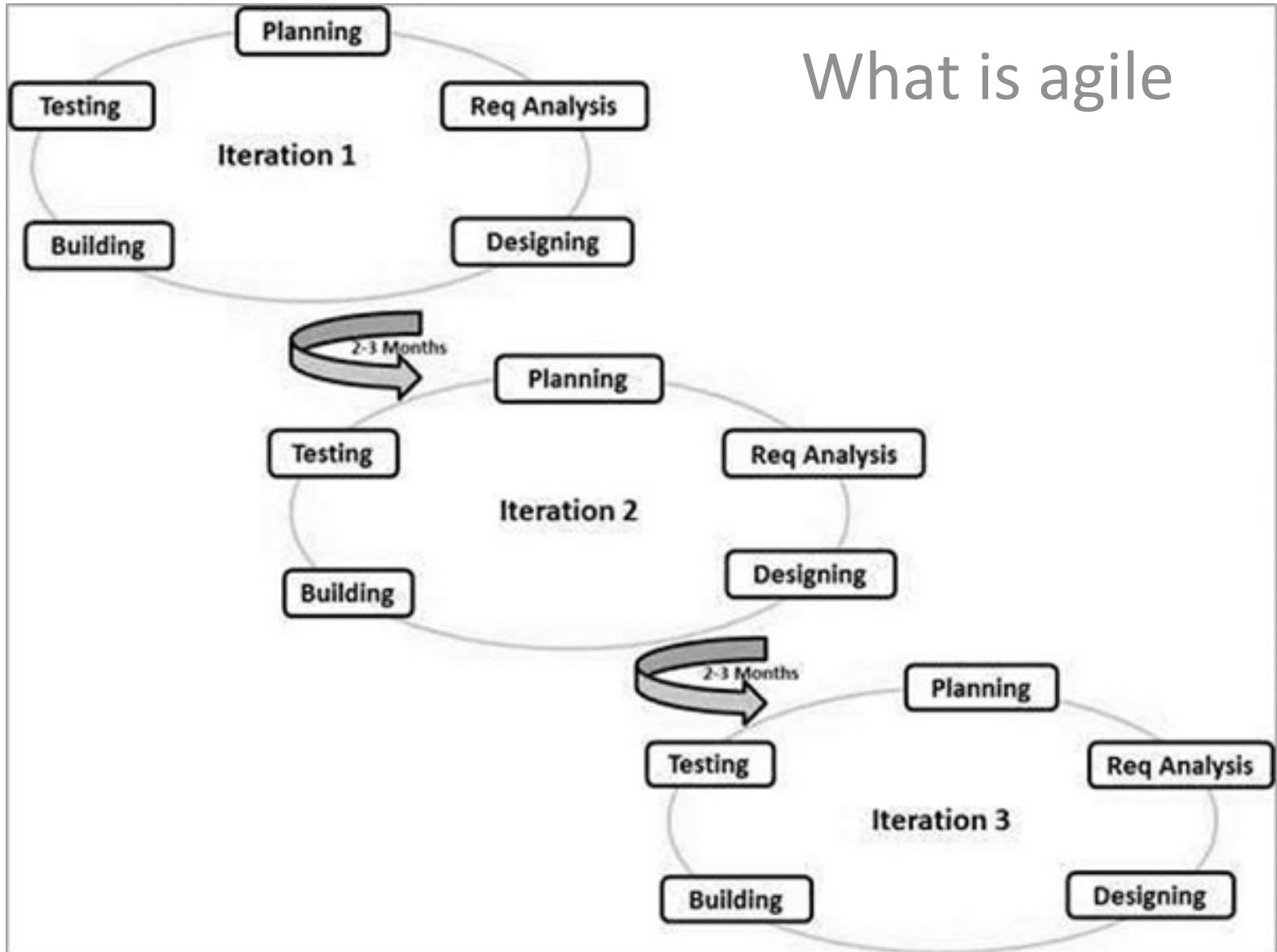
Practice	Waterfall	Iterative (hybrid)	Agile
<b>Goals</b>	Goals are defined for each phase by defining entry and exit criteria. Delivering artifacts by phase wise.	Completion of analysis & design for a set of features by one team followed by completion of code & test by another team.	Completing the feature/story in all aspects within a sprint. Delivering the shippable product.
<b>Definition of Ready</b>	Analysis and design should be completed for all stories before programming	Analysis and design should be completed for a 'set of stories' before programming	Stories that are not subjected to change, that can be completed (analysis to demo) within the sprint will be considered for the sprint.
<b>Scoping</b>	Product Owner (PO) decides project scope	Project Manager (PM) decides scope for iteration in consultation with Product Owner	Team decides the sprint scope as per the capacity availability & product backlog prioritization

What is Agile

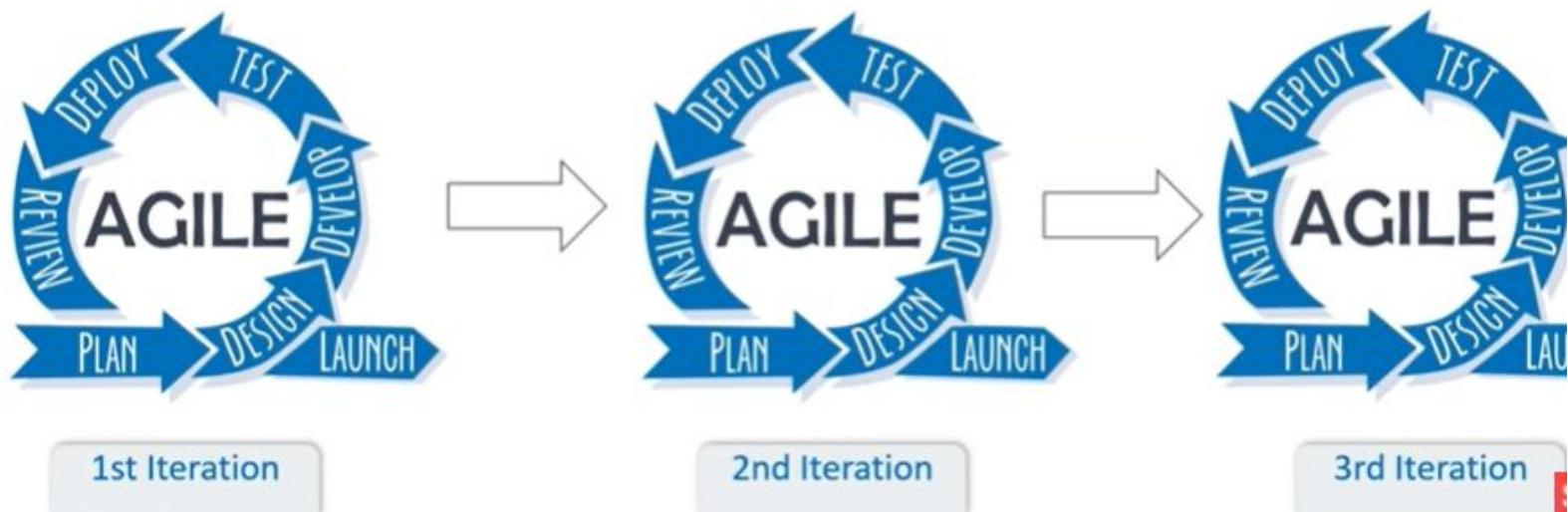




# What is agile



# What is agile



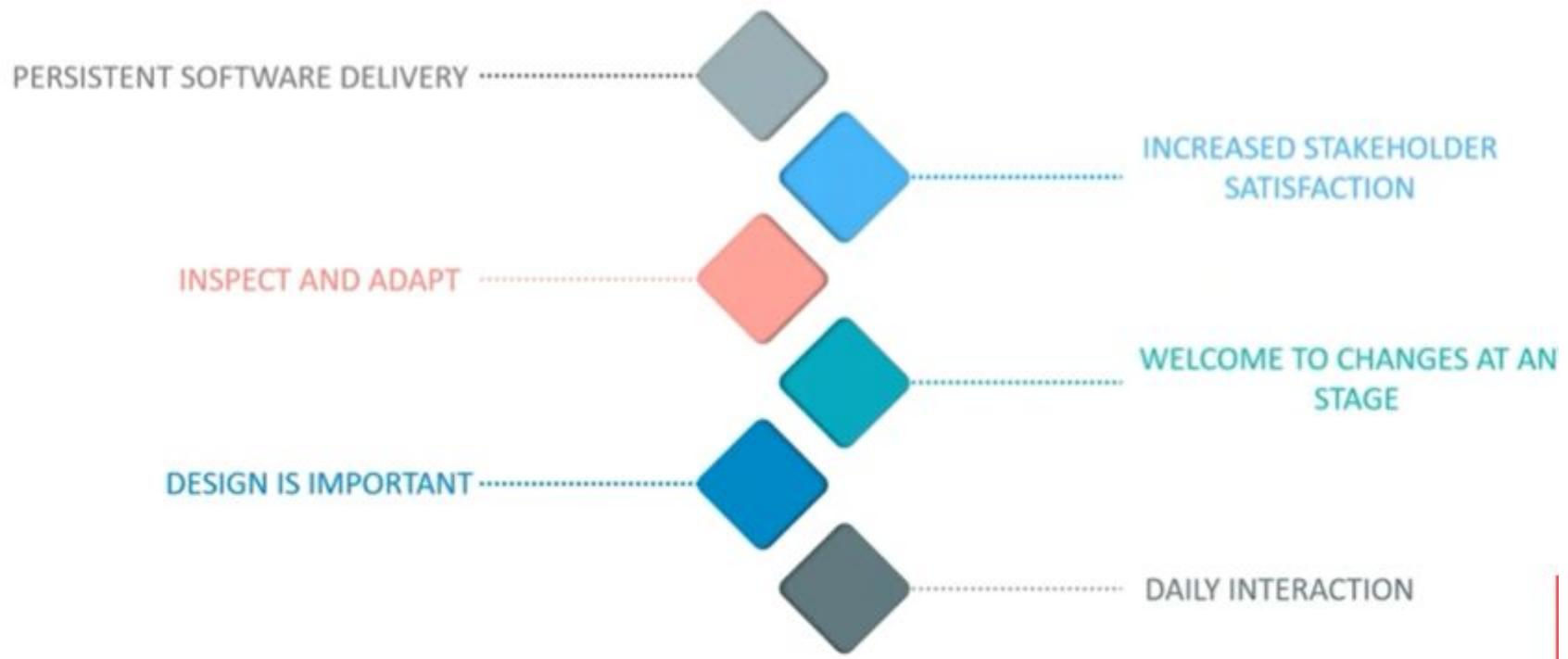
# Agile - Characteristics

- Iterative/incremental and Ready to Evolve
- Face-to-face Communication
- Feedback Loop

# Agile Success Factors

- Freedom to change
- Energized team
- Commo to customer
- Collaboration
- Attention to quality
- Incrementalism
- Automation

# Advantages of Agile



# How to implement agile



**How to implement  
Agile?**

# Practices of agile

---



Daily standup



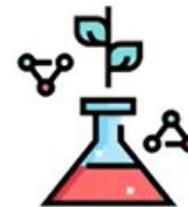
Sprints



Splitting user stories



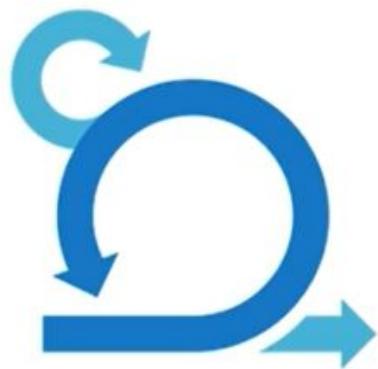
Pair programming



Test-driven development

# Agile frameworks

---



Scrum



Lean



Kanban

# · Key Components of Agile Development Methodologies

## User stories

These are short descriptions of what a feature should do, written from the user's point of view. They help break down the work into smaller pieces and answer the [type of user], I want [some action] so that [some benefit] question.

## Epics

These are like big storylines, with smaller user stories that connect. They're too huge to finish in one sprint.

- **Product backlog**

This ever-changing to-do list holds all the features, user stories, and tasks. It's organized by what's most important right now, and as new information comes in or priorities change, the backlog shifts around.

- **Product owner**

It is like the bridge between the dev team and the customer. They're the go-to person for figuring out what the customer needs. They figure out the priorities to make the path forward clear.

- **Sprints (time-boxed iterations)**

These are short, focused work periods, usually lasting 2-4 weeks. The team works to complete a set of tasks from their backlog during the sprint, intending to have a usable product by the end.

- **Daily stand-ups (daily scrums)**

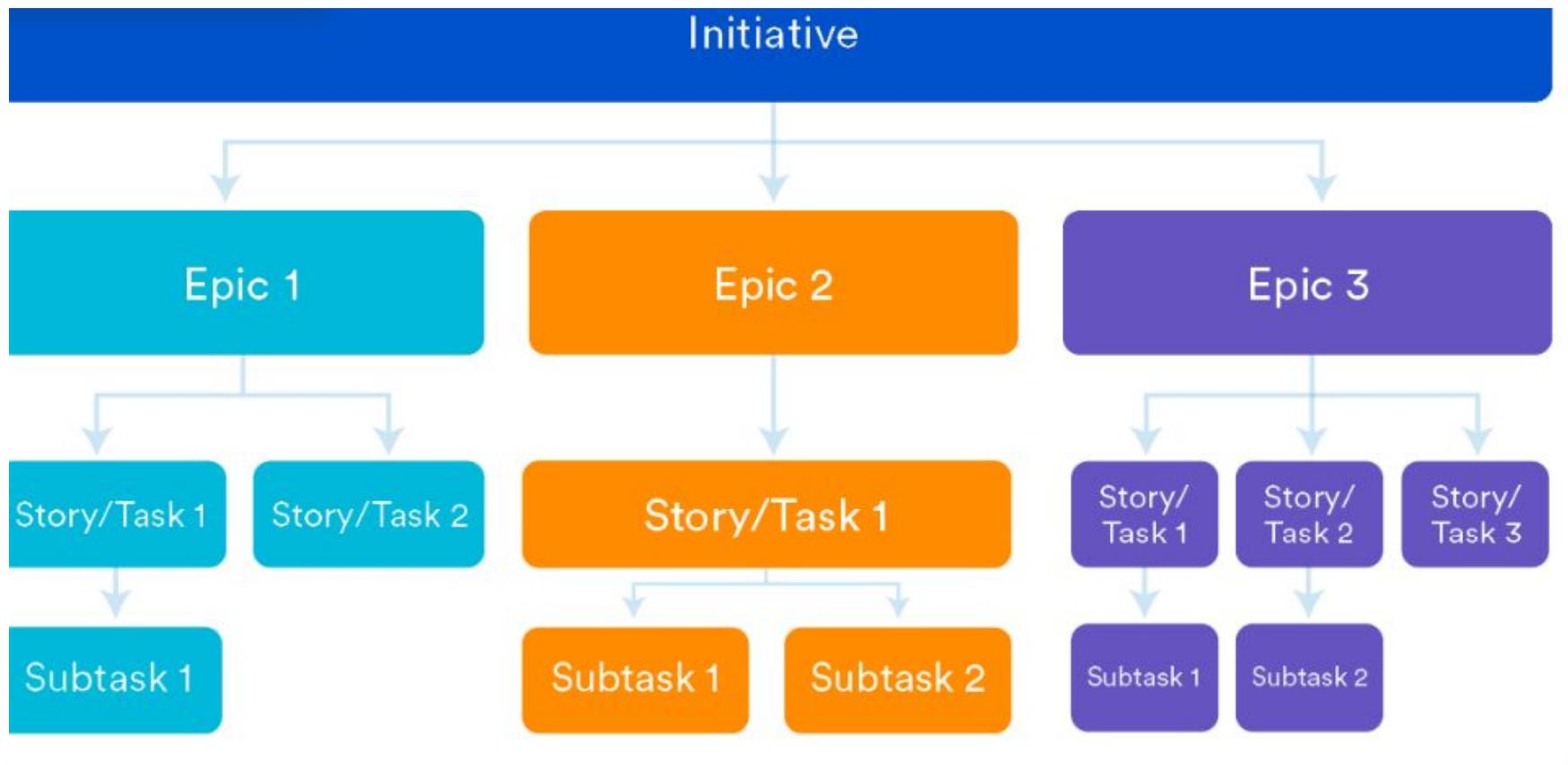
These are quick daily check-ins with the team to see where everyone's at. Team members share what they did and are doing now and if anything is blocking them from moving forward.

- **Retrospectives**

These are reflective meetings after a sprint ends. The team looks back on what went well and what didn't, coming up with ways to improve for the next sprint. It's an opportunity to fine-tune the strategy based on what they just experienced.

- **Scrum master or facilitator**

It is like the team's guide, ensuring everyone follows the Agile rules and that things run smoothly. They clear any roadblocks and keep the workflow efficient.



## The Scrum Framework

The Scrum framework is a simple, agile way to manage projects that helps teams work together to create high-quality products quickly. Here are the basic components and principles of Scrum, explained in simple terms.

### Scrum Team

- **Product Owner:** Decides what the team should work on and makes sure it meets business and customer needs.
- **Scrum Master:** Helps the team follow Scrum rules and removes any obstacles they face.
- **Development Team:** A group of people with different skills who work together to build the product.

### Artifacts:

- **Product Backlog:** A list of everything that needs to be done for the product, organized by priority.
- **Sprint Backlog:** A list of tasks the team plans to complete during the current sprint.
- **Increment:** The finished work at the end of a sprint that could be released to users.

### Events:

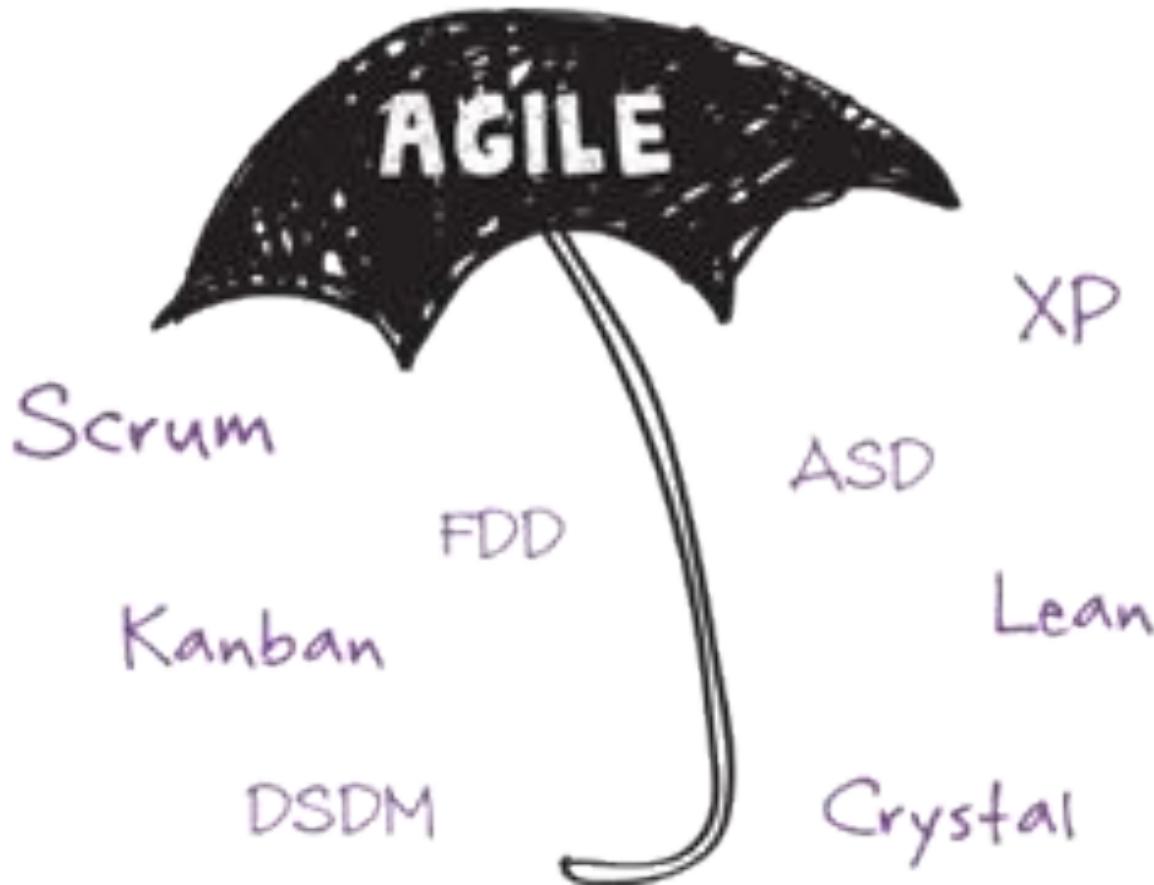
- **Sprint:** A short, fixed period (usually 1-4 weeks) where the team works on specific tasks from the sprint backlog.
- **Sprint Planning:** A meeting at the start of the sprint where the team decides what to work on.
- **Daily Scrum:** A quick daily meeting where the team checks in on progress and plans the day's work.
- **Sprint Review:** A meeting at the end of the sprint where the team shows what they've accomplished and gets feedback.
- **Sprint Retrospective:** A meeting at the end of the sprint where the team discusses what went well and what could be improved for the next sprint.

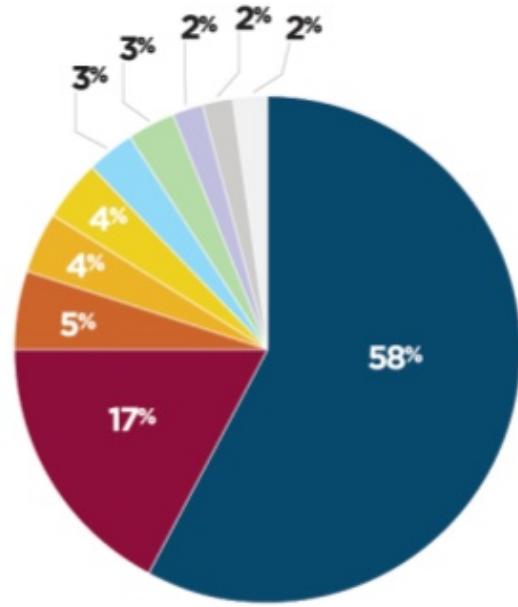
## Key Terminologies of Scrum

Here are some of key terminologies of Scrum:

1. **Product Backlog:** The product backlog is known to be the prioritized list of fixes as well as features that is included in the product's roadmap.
2. **Sprint:** Sprint is known as a **time-box** event which typically lasts from one week to four weeks, in this phase a product increment or iteration occurs.
3. **Development Team:** The development team is a group of individuals who are professional in their field and are responsible for product delivery.
4. **Daily Scrum:** Daily scrum is a 15 minute daily meeting used by the development team to integrate activities and to create a plan for the next 24 hours of development.
5. **Sprint Review:** The sprint review is held at the end of the sprint in which the team presents all the work that is completed to their stakeholders and the stakeholders give back their feedback.
6. **Sprint Retrospective:** The sprint retrospective is a meeting concluded at the end of each sprint so that the team can discuss what went well and what could be improved as well as how to make those improvements.

# Agile Methodologies





## AGILE METHODOLOGY MOST CLOSELY FOLLOWED

Scrum or Scrum variants were by far the most common agile methodologies employed.

- SCRUM
- SCRUM/XP HYBRID
- CUSTOM HYBRID
- OTHER
- EXTREME PROGRAMMING (XP)
- DON'T KNOW
- SCRUMBAN
- LEAN
- FEATURE DRIVEN DEVELOPMENT (FDD)
- AGILEUP

# SCRUM

# scrum

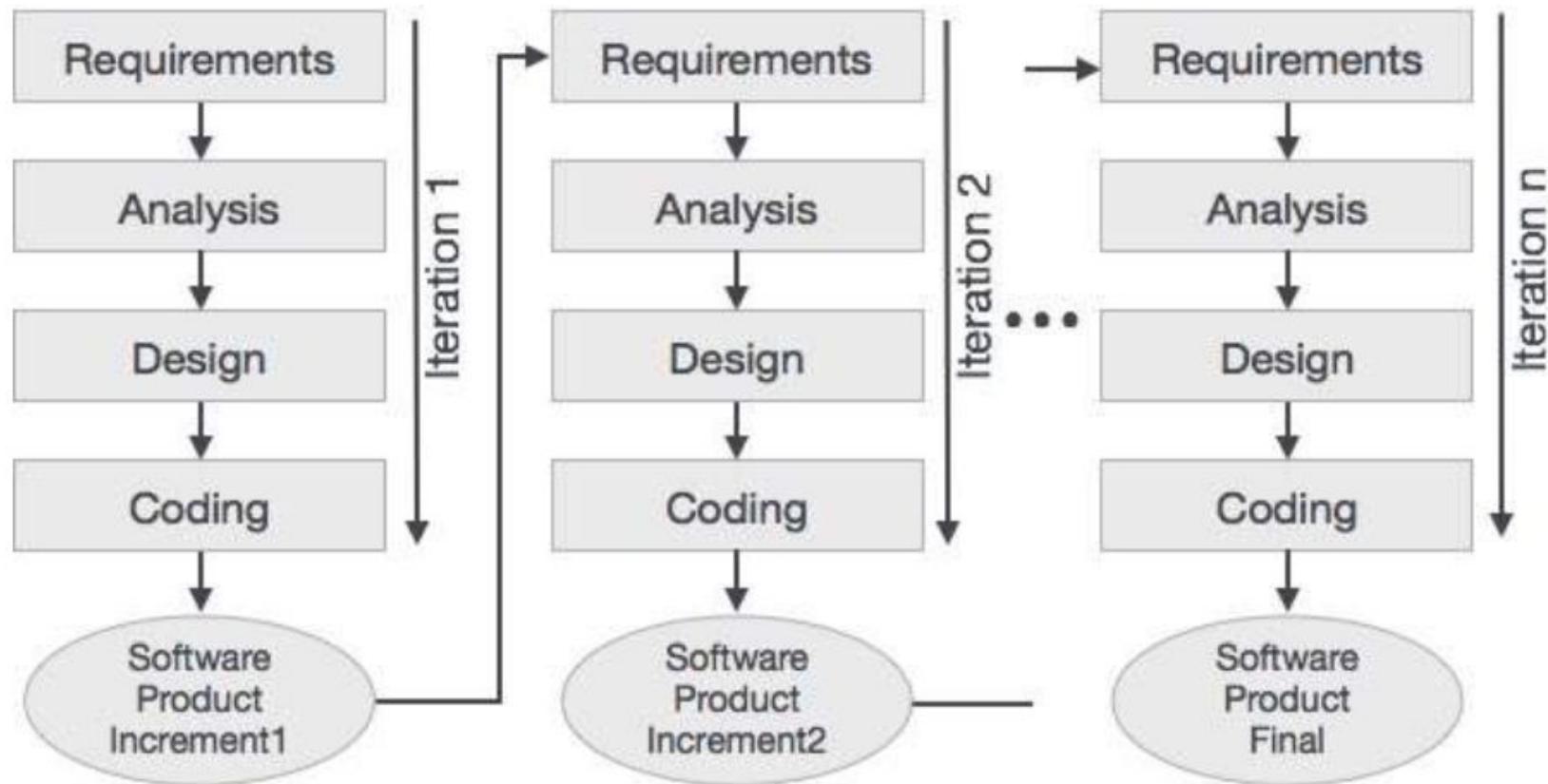
It is the most popular agile framework, which concentrates particularly on how to manage tasks within a team-based development environment. Scrum uses iterative and incremental development model, with shorter duration of iterations. Scrum is relatively simple to implement and focuses on quick and frequent deliveries.

# SCRUM



- One of the most popular agile methodology.
- Scrum is a lightweight, iterative and incremental framework.
- Scrum breaks down the development phases into stages or cycles called “sprints”.
- The development time for each sprint is maximized and dedicated, thereby managing only one sprint at a time.
- Scrum Team has scrum master and product owner with constant communications on the daily basis.
- Keywords: Backlog, Sprint, Daily Scrum, Scrum master, Product owner,

# Iterative Incremental Model



# Iterative Incremental Model

In the iterative incremental model, the development starts with a limited number of finalized and prioritized requirements.

The deliverable is a working increment of the product. A set of activities ranging from requirements to code development is called an iteration. Based on the functionality of the increment and any or all of the new, modified, pending requirements, the next lot of requirements is given to the subsequent iteration.

The outcome of the subsequent iteration is an enhanced working increment of the product. This is repeated till the product accomplishes the required functionalities.

# What is scrum

- Better way of building products
- Frame work
- Lightweight, simple to understand and difficult to master
- Scrum is everywhere
  - 90% estimated agile teams use scrum
  - 12m+ estimated using scrum daily
  - Practiced everywhere
  - **one scrum guide**
- Scrum team

# The Agile Scrum Framework at a Glance

Inputs from Executives,  
Team, Stakeholders,  
Customers, Users



Product Owner



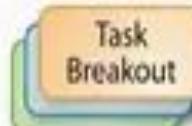
Product Backlog



The Team

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting



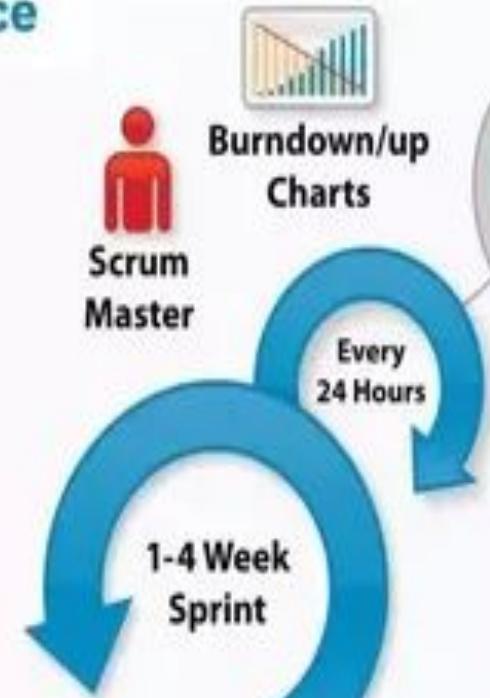
Sprint Backlog

Scrum Master

Burndown/up Charts

Every 24 Hours

1-4 Week Sprint



Sprint end date and team deliverable do not change



Sprint Review



Finished Work

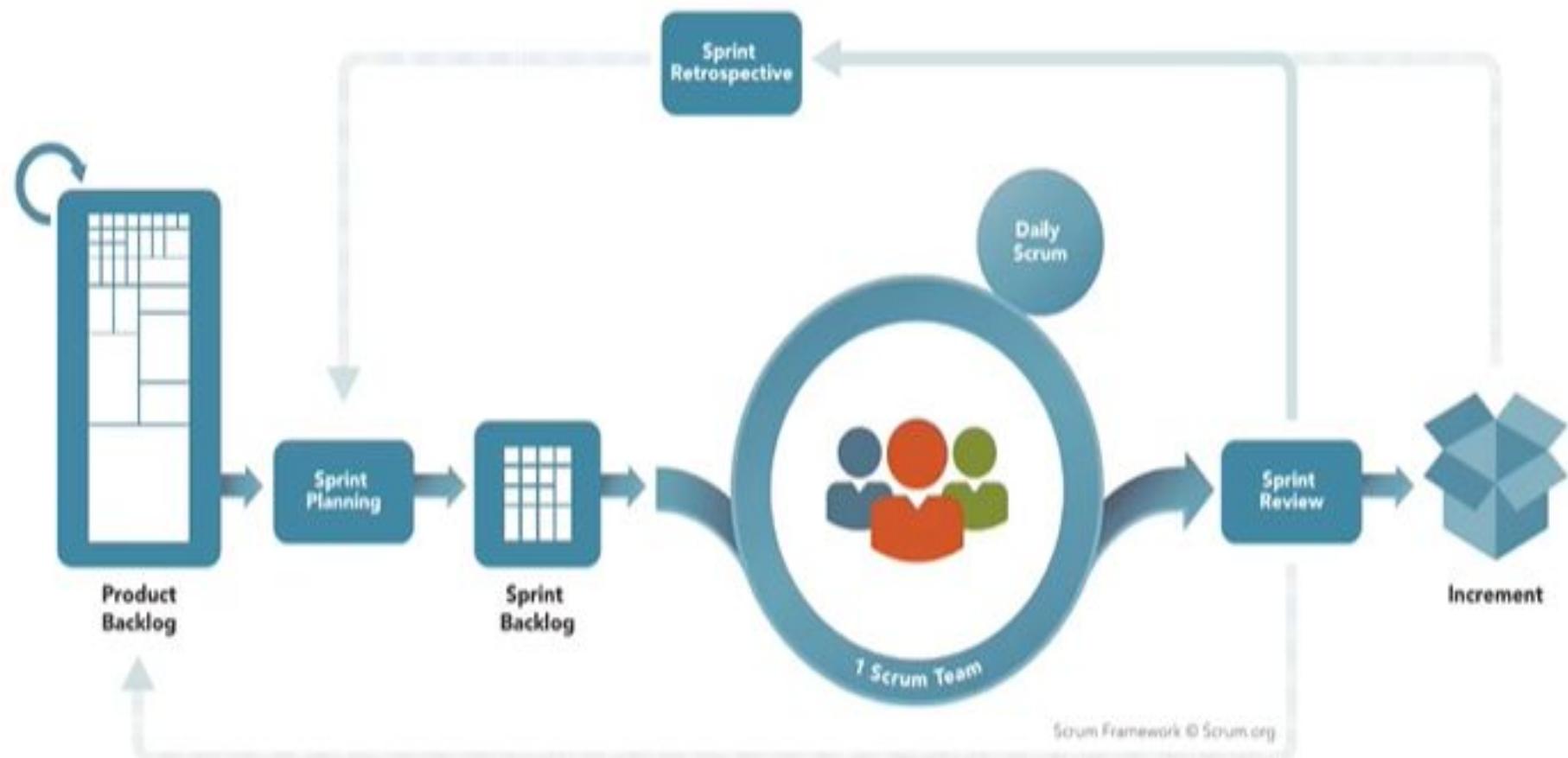


Sprint Retrospective



AGILE FOR ALL  
Making Agile a Reality®

# SCRUM FRAMEWORK



- To ensure that Agile Inspect & Adapt processes are leveraged properly which includes :

1. Daily stand-ups,
2. Planned meetings,
3. Demo,
4. Review,
5. Retrospective Meetings, and
6. To facilitate team meetings and decision-making process.

A **daily stand-up** meeting is an **opportunity for the project team to discuss a project's progress at a high level**. These meetings last 15 minutes and allow each contributor to report on their accomplishments since the last stand-up meeting.



# What is Scrum process?

## Scrum Roles



Product Owner



Scrum Master



Scrum Team  
(max - 9)

## Key Artifacts

### Product Backlog

- Requirements – user stories
- Desired work
- Prioritized by Product Owner
- Anybody can add to it

### Sprint Goal

- Summary of focused work in sprint
- Declared by Product Owner
- Accepted by team

### Sprint Backlog

- Team signs up for work of their own – **work never assigned**
- Owned/managed by the team
- Estimated work remaining is updated daily

### Blocks List

- List of blocks & unmade decisions
- Owned by Scrum Master
- Updated daily

### Burndown chart

- Effort spent over period
- Stories/ features completed

## Ceremonies

### Sprint planning

- Hosted by Scrum Master
- Pick highest priority items in Product backlog and the team turns the items into Sprint Backlog
- Estimate sprint backlog in hours
- Work breakdown
- Declare Sprint Goal

### Daily Scrum

- Hosted by Scrum Master
- 15 minutes - same time every day
- Not for problem solving
- 3 questions in meeting: (1) What did you do? (2) What will you do? (3) What's in your way?
- Team updates sprint backlog

### Sprint Review

- Hosted by Scrum Master – 2-4 hours
- Accomplishments
- Whole team participants
- Take form of demo for features

### Sprint Retrospective

- Hosted by SM – 15-30 minutes
- Discuss on “Start doing”, “Continue doing” and “Stop doing”

## Process

### Product Backlog

### Sprint - 2-4 weeks

#### Sprint Planning

#### Sprint Goal

#### Sprint Backlog

#### Blocks

#### Product

#### Daily Scrum

#### Daily Work

#### Sprint Review/ Retrospective

### Product Backlog

- **Product owner(po):** inputs from executives, teams, stakeholders , users
- **Proudct backlog(pb):** ranked list of what is required features, stories etc[ priotorised list ]
- **Sprint:** means a time box to fulfil
- **Sprint planning(Sp):** team will meet and selects which is high priority and start at top as much as it can commit to deliver by end of sprint
- **Sprint backlog(sb):** task breakout like
  - Sprint1: feature 1, feature 2
  - Sprint 2: feature 4
  - Sprint 3: feature 3
- **Daily Scrum (ds):** a daily scrum meeting(10-15 min)

- **Finished Work /increment:** potentially shipped product for implement (pspi)
- **Sprint review (sr):** reviewed by cust and other team members
- **Sprint Retrospective:** hosted by scrum master-15-30 min , discuss on “start doing”, “continue doing” and “stop doing” (eg end of the cricket match)

# A Typical Sprint Retrospective Model

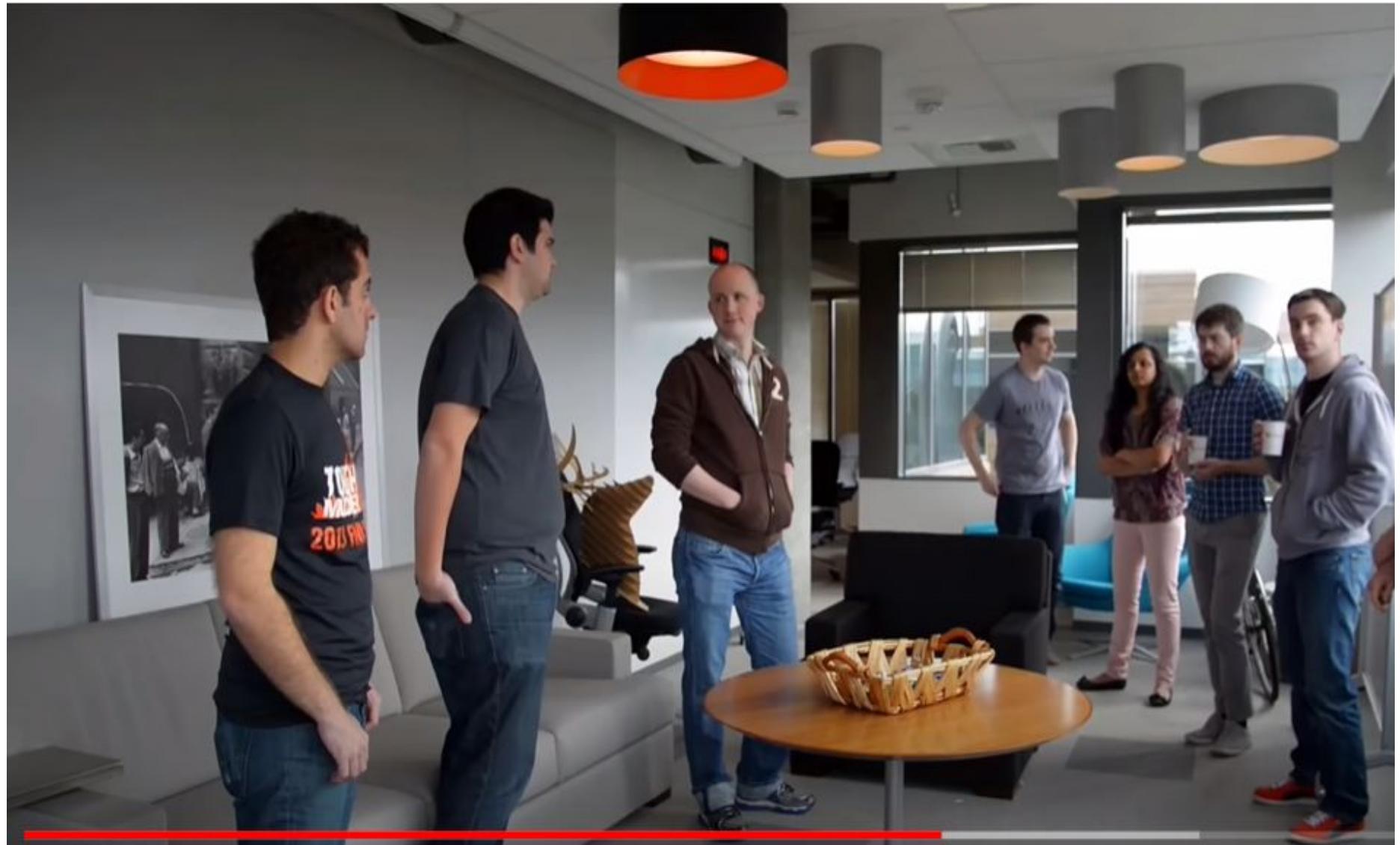
What worked well?

What could be improved?

What will we commit to  
doing in the next Sprint?

*Scrum Team members  
make actionable  
commitments*

# 10min meeting



# Documents of scrum

- 1. Product Backlog**
- 2. Sprint Backlog**
- 3. Increment**

# Product Backlog

VS

# Sprint Backlog

## Product Backlog

Userstory#1

Userstory#2

Userstory#3

Userstory#4

Userstory#5

Userstory#6

## DEEP

Technique

- \* Detailed
- \* Estimated
- \* Emergent
- \* Prioritized

## Sprint Backlog

Sprint #1

Userstory#1

Userstory#2

Userstory#3

Sprint #2

Userstory#4

Sprint #3

# What are roles in Scrum?



Product Owner (PO) is a client's representation

- ✚ Define features of product
- ✚ Decide Release Date and content
- ✚ Prioritize features according to market value
- ✚ Be responsible for the profitability of product
- ✚ Accept or reject work item result



Scrum Master represents Management

- ✚ Enacting Scrum values
- ✚ Ensure team's productivity
- ✚ Prioritize features according to market value
- ✚ Corporate across all roles and functions
- ✚ Shield team from external interferences



The Team

- ✚ 5-9 members
- ✚ Including: developers, testers, designers,...
- ✚ Full time
- ✚ Work Self-organizing
- ✚ Membership should be changed each sprint

## A practice for prioritizing features (or User Story)

Priority = Max (5 (or 6) x priority that assigned by PO, sum (priority that assigned by team members))

12

# Product Owner

- To define the requirements and prioritize their values.
- To determine the release date and contents.
- To take an active role in iteration planning and release planning meetings.
- To ensure that team is working on the most valued requirement.
- To represent the voice of the customer.

# Scrum master

To enable **close co-operation between all roles and functions.**

To remove any blocks.

To shield the team from any disturbances.

To work with the organization to track the progress and processes of the company.

## Scrum Master Responsibilities

- \* Clearing Obstacles
- \* Establishing an Environment where the team can be effective
- \* Addressing team dynamics
- \* Ensuring a good relationship between the team and product owner as well as others outside the team
- \* Protecting the team from outside interruptions and distractions.

# The Scrum Board

Story	To Do	In Progress	To Verify	Done

<https://www.youtube.com/watch?v=oTZd2vo3FQU>

Scrum Explained in Hindi | Software Engineering and Project Management Course

# Lean

# Lean

- **Management philosophy** inspired by Toyota system practices and results is characterized by a process structure where there is an attempt to **minimize risk and waste** while maximizing customer value. Lean is the foundation of Agile and can be perfectly applied across multiple business areas.

- **The objective of Lean methods**
- In fact, Lean came much earlier than the Agile Manifesto. It emerged in postwar Japan in automobile factories that wanted to be more productive.
- It turns out that Lean projects are quite effective if they incorporate Agile concepts into their execution. After all, **Lean means lean, without excess or waste**, something that meets all that the Agile methodologies propose.
- There are a number of Lean tools (such as the 5S and just-in-time, 6sigma) that should be used to combat the 8 wastes that bring inefficiency and reduce productivity in companies.
- See what these 8 wastes are:



- 8 wastes are:

- 1. Wait**
- 2. Defect**
- 3. Transport**
- 4. Movement**
- 5. Excess Inventor**
- 6. Excess production**
- 7. Unnecessary processing**
- 8. Unused talent**



#### DEFECTS

Waste from a product or service failure to meet customer expectations



#### OVERPRODUCTION

Waste from making more product than customers demand



#### WAITING

Waste from time spent waiting for the next process step to occur



#### UNUSED TALENT

Wastes due to underutilization of people's talents, skills, and knowledge



#### TRANSPORTATION

Wasted time, resources, and costs when unnecessarily moving products and materials



#### INVENTORY

Wastes resulting from excess products and materials that aren't processed



#### MOTION

Wasted time and effort related to unnecessary movements by people



#### EXTRA-PROCESSING

Wastes related to more work or higher quality than is required

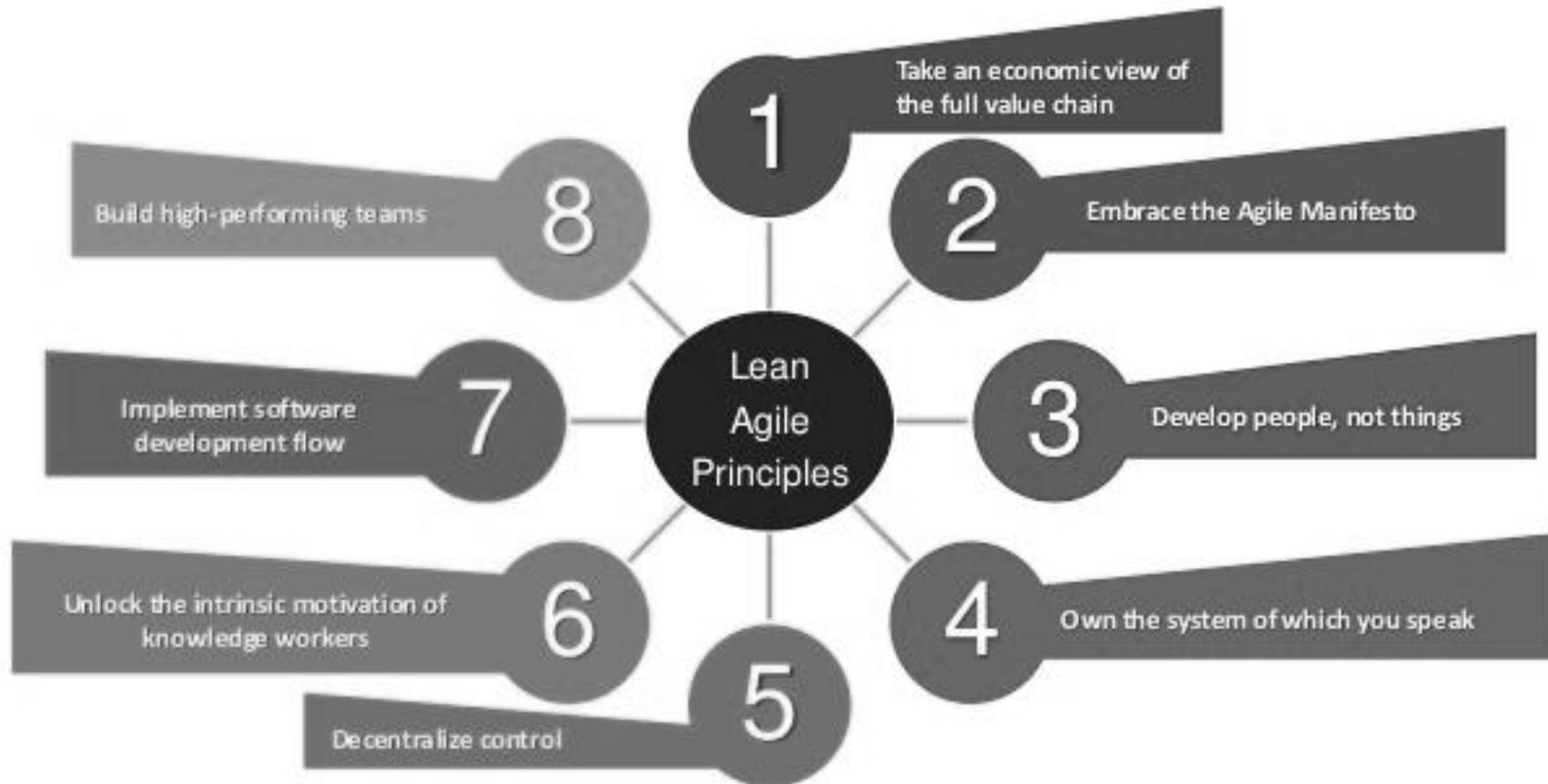
# Lean

- The main principles of Lean methodology include:
  - Eliminating Waste
  - Amplifying Learning
  - Deciding as Late as Possible
  - Delivering as Fast as Possible
  - Empowering the Team
  - Building Integrity In
  - Seeing the Whole



SUBSCRIBE  
e! ▶

# 8 Principles of Lean Agile Leadership



# Kanban

# kanban

- Kanban is a popular framework used to implement agile and DevOps software development. It requires real-time communication of capacity and full transparency of work. Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.
-

- Kanban story
- Kanban is enormously prominent among today's agile and DevOps software teams, but the kanban methodology of work dates back more than 50 years. In the late 1940s Toyota began optimizing its engineering processes based on the same model that supermarkets were using to stock their shelves.
- Supermarkets stock just enough product to meet consumer demand, a practice that optimizes the flow between the supermarket and the consumer. Because inventory levels match consumption patterns, the supermarket gains significant efficiency in inventory management by decreasing the amount of excess stock it must hold at any given time. Meanwhile, the supermarket can still ensure that the given product a consumer needs is always in stock.

- When Toyota applied this same system to its factory floors, the goal was to better align their massive inventory levels with the actual consumption of materials. To communicate capacity levels in real-time on the factory floor (and to suppliers), workers would pass a card, or "kanban", between teams. When a bin of materials being used on the production line was emptied, a kanban was passed to the warehouse describing what material was needed, the exact amount of this material, and so on. The warehouse would have a new bin of this material waiting, which they would then send to the factory floor, and in turn send their own kanban to the supplier. The supplier would also have a bin of this particular material waiting, which it would ship to the warehouse. While the signaling technology of this process has evolved since the 1940s, this same "just in time" (or JIT) manufacturing process is still at the heart of it.

## **Kanban is based on 3 basic principles:**

1. Visualize what you do today (**workflow**)
2. Limit the amount of work in progress (WIP)
3. Enhance flow:

# Kanban boards

The work of all kanban teams revolves around a [kanban board](#), a tool used to visualize work and optimize the flow of the work among the team. While physical boards are popular among some teams, virtual boards are a crucial feature in any agile software development tool for their traceability, easier collaboration, and accessibility from multiple locations.

Regardless of whether a team's board is physical or digital, their function is to ensure the team's work is visualized, their workflow is standardized, and all blockers and dependencies are immediately identified and resolved. A basic kanban board has a three-step workflow: To Do, In Progress, and Done. However, depending on a team's size, structure, and objectives, the workflow can be mapped to meet the unique process of any particular team.

The kanban methodology relies upon full transparency of work and real-time communication of capacity, therefore the kanban board should be seen as the single source of truth for the team's work.

# Team Kanban Board

QUICK FILTERS: Critical partners Only my partners Recently updated

1 To do

+ TIS-28

↑ Research options  
to travel to Pluto



4 In progress

+ TIS-25

↑ Engage Jupiter  
Express for travel



3 Code review Max 2

+ TIS-27

↑ Engage Saturn  
Resort as PTP



1 Done

+ TIS-23

↑ Engage JetShuttle  
SpaceWays for  
travel

Release



+ TIS-25

↑ Add Deimos Tours  
as a travel partner



+ TIS-20

↑ Engage Saturn  
Lines for group tours



+ TIS-24

↑ Sign Contract for  
SunSpot Tours



06 Aug to 14 Sept

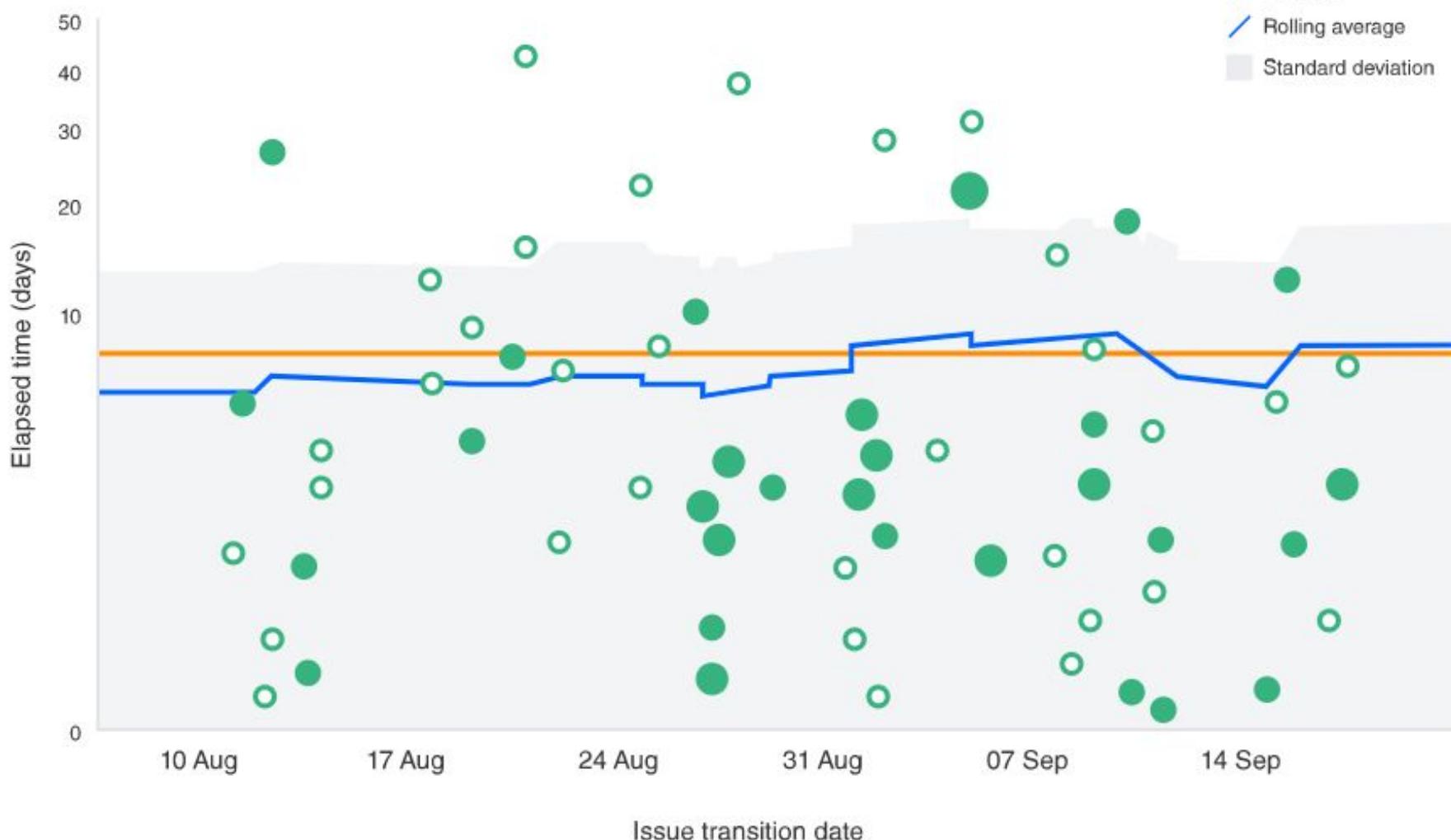
1w 13h 17m average

3d median

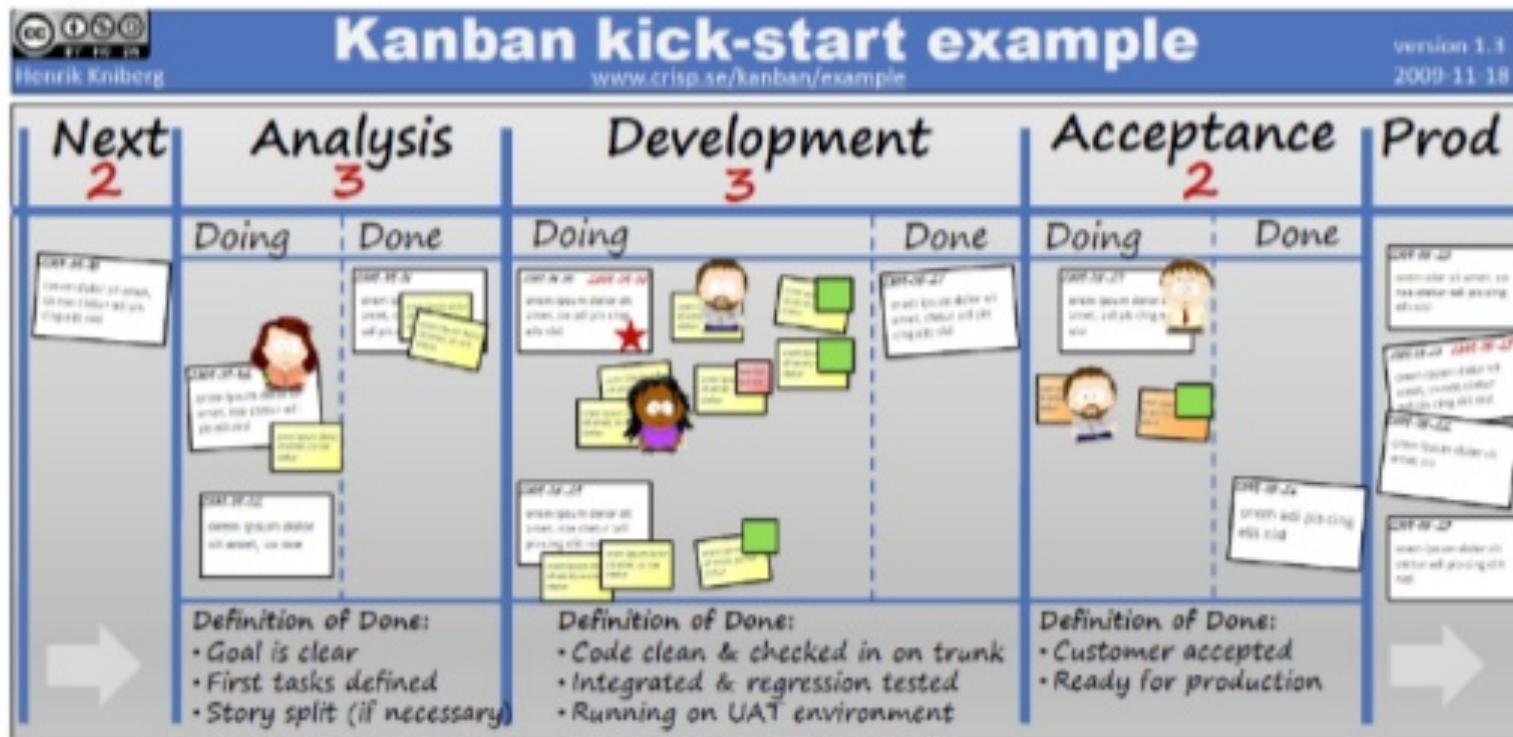
<1m min time

13w 6d 23h max time

240 issues



# Sample Kanban Board



### Feature / story

Date when added to board

2009-08-20 2009-09-30

(description)



Hard deadline (if applicable)

★ = priority

★★ = panic

Who is analyzing / testing right now

### Task / defect

= task = defect

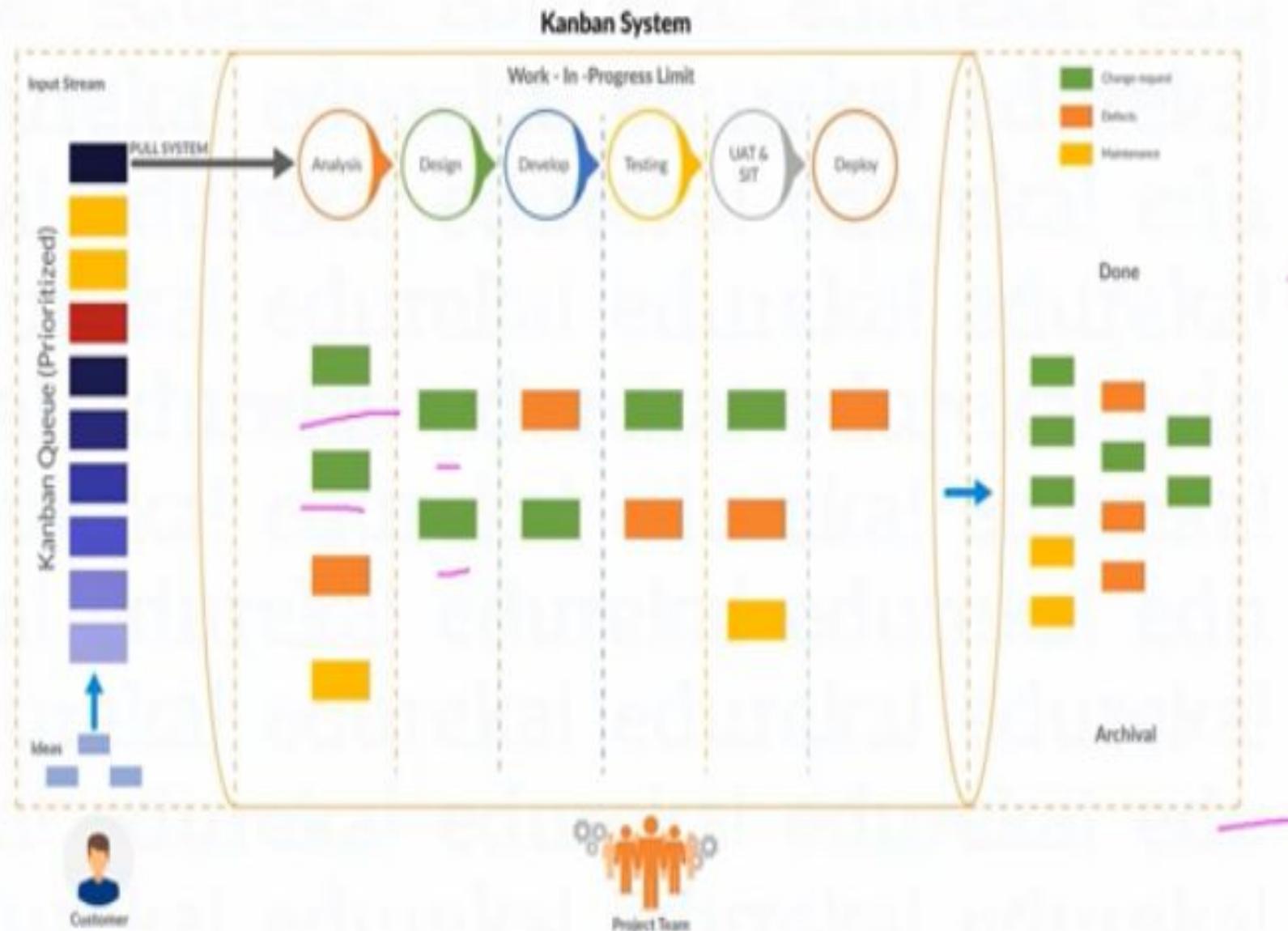
= completed

= blocked

= who is doing this right now

### What to pull first

1. Panic features ★★  
(should be swarmed and kept moving. Interrupt other work and break WIP limits as necessary)
2. Priority features ★
3. Hard deadline features (only if deadline is at risk)
4. Oldest features



# Dynamic Systems Development Method (DSDM)

# Dynamic Systems Development Method (DSDM)

**M** – *Must have requirements*

**S** – *Should have if at all possible*

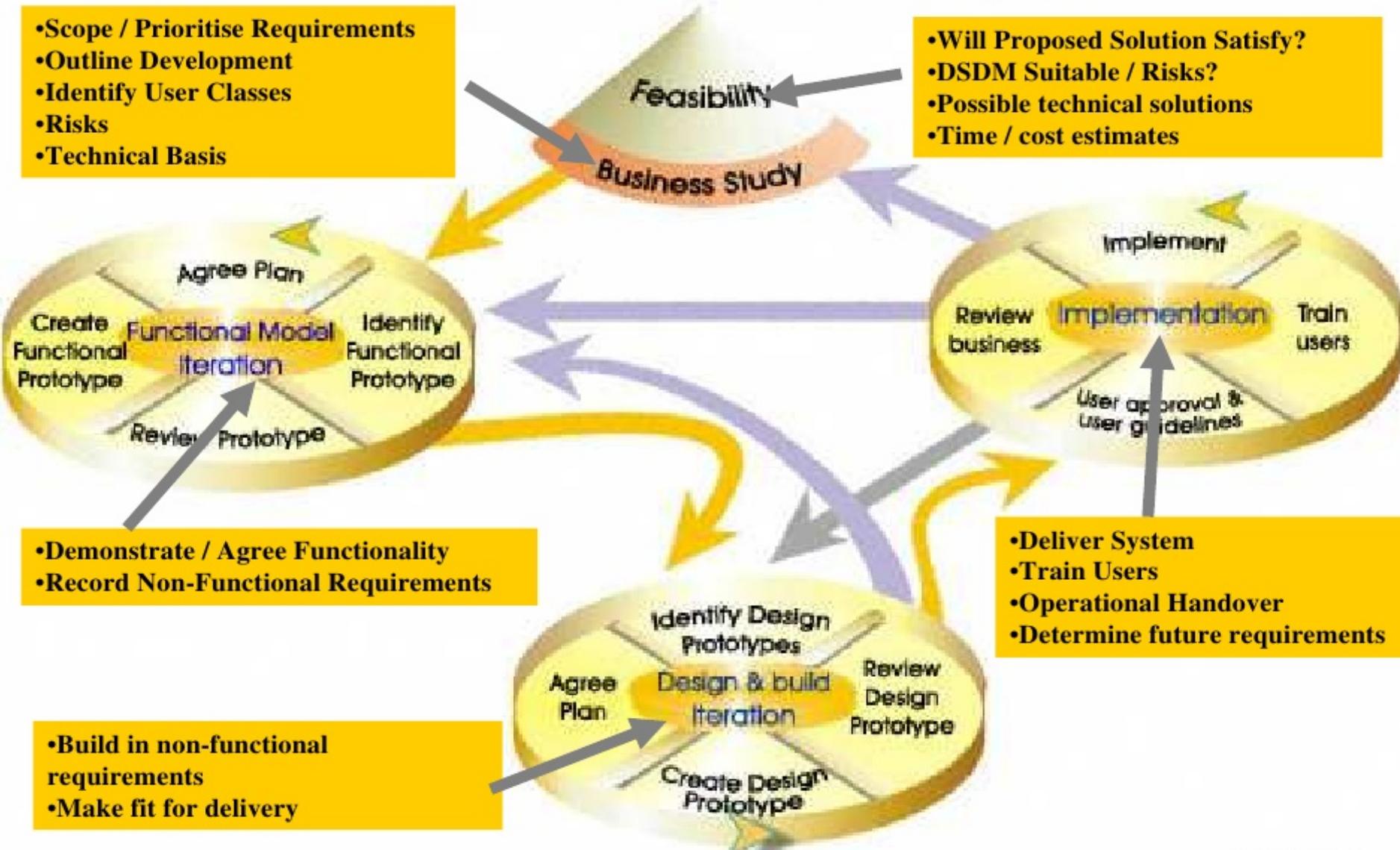
**C** – *Could have but not critical*

**W** – *Won ‘t have this time, but potentially later*

# The DSDM Lifecycle

- Scope / Prioritise Requirements
- Outline Development
- Identify User Classes
- Risks
- Technical Basis

- Will Proposed Solution Satisfy?
- DSDM Suitable / Risks?
- Possible technical solutions
- Time / cost estimates



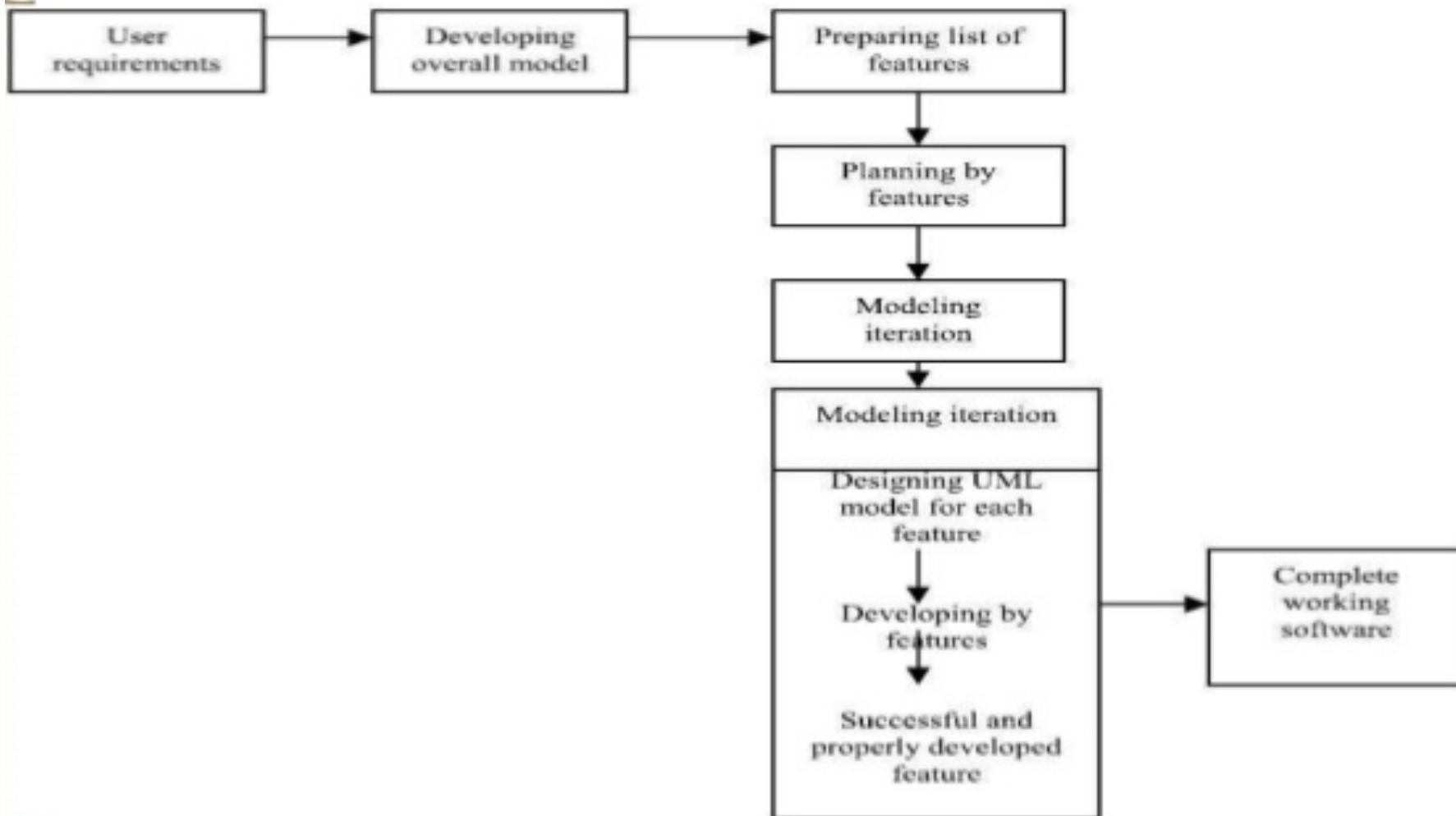
# Feature-Driven Development (FDD)

# Feature-Driven Development (FDD)

- Domain Object Modeling
- Developing by Feature
- Component/Class Ownership
- Feature Teams
- Inspections
- Configuration Management
- Regular Builds
- Visibility of progress and results

# Agile Process & Methodologies

## Feature Driven Development (FDD)



# crystal

- Crystal is an agile framework focusing on individuals and their interactions, as opposed to processes and tools. In other words, this framework is a direct outgrowth of one of the core values articulated in the Agile Manifesto.
- The Crystal agile framework is built **on two core beliefs:**
- Teams can find ways on their own to improve and optimize their workflows
- Every project is unique and always changing, which is why that project's team is best suited to determine how it will tackle the work

# **What are the Strengths and Weakness of Crystal?**

## **Crystal's strengths include:**

- Allows teams to work the way they deem most effective
- Facilitates direct team communication, transparency and accountability
- The adaptive approach lets teams respond well to changing requirements

## **Crystal's weaknesses include:**

- Lack of pre-defined plans can lead to scope creep
- Lack of documentation can lead to confusion

		Crystal Methodologies				
		Clear	Yellow	Orange	Red	Maroon
Criticality of the Project	Life (L)	L6	L20	L40	L80	L200
	Essential Money (E)	E6	E20	E40	E80	E200
	Discretionary Money (D)	D6	D20	D40	D80	D200
	Comfort (C)	C6	C20	C40	C80	C200
		1 to 6	7 to 20	21 to 40	41 to 80	81 to 200
Number of People involved in the Project						

# Extreme Programming (XP)

# Extreme Programming (XP)

- Xp is approach/methodology of the agile
- Will helps to incorporate the changes of the cust
- XP is a disciplined approach to delivering high-quality software quickly and continuously.
- It promotes high customer involvement
  - rapid feedback loops,
  - continuous testing,
  - continuous planning,
  - and close teamwork
  - to deliver working software at very frequent intervals, typically every 1-3 weeks.

# XP 5 values

- The original XP recipe is based on five simple values “

Communication

- Onsite Customer, Pair Programming and Co-Location

Simplicity

- Simple Design and Code What's Needed

Feedback

- Small releases w/ Quick Feedback and TDD

Courage

- Collective Code Ownership and Refactoring

Respect

respect is required by all the stakeholders

TDD: Test-driven development

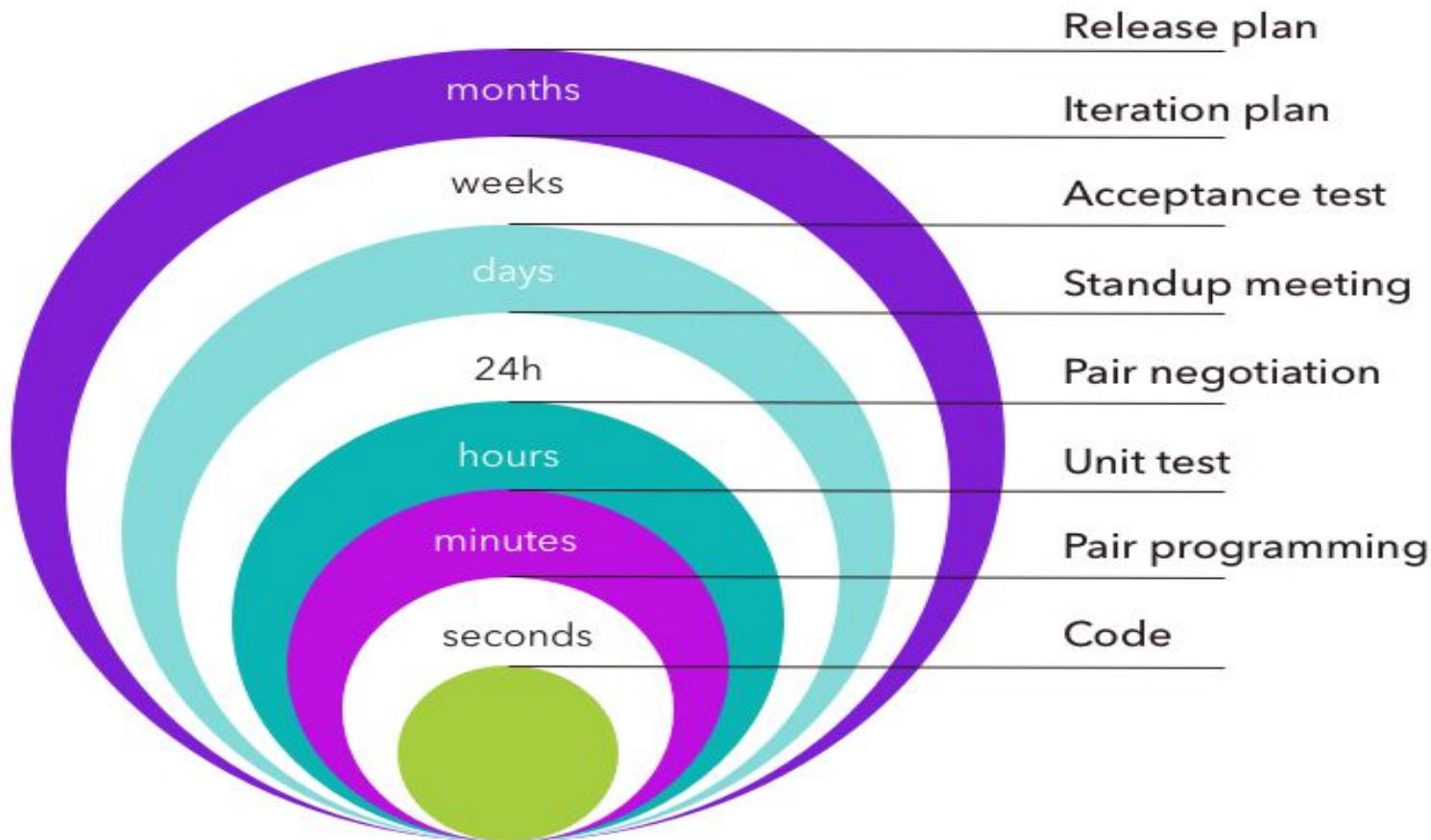
# 5 values

- **Communication:** primitive, foundation for all stakeholders frequently
- **Simplicity** developer to develop immediate requirement not future requirement, reduce complex
- **Feedback:** cust feedback of actual and expected features ie error margin and s/w team feedback on budget and cost time to time
- **Courage:** s/w team required courage and discipline
- **Respect:** s/w team required respect as well as all stakeholders

# **twelve supporting practices:**

- 1. Planning Game**
- 2. Small Releases**
- 3. Customer Acceptance Tests**
- 4. Simple Design**
- 5. Pair Programming**
- 6. Test-Driven Development**
- 7. Refactoring**
- 8. Continuous Integration**
- 9. Collective Code Ownership**
- 10. Coding Standards**
- 11. Metaphor**
- 12. Sustainable Pace**

# *XP Feedback Loops*



Source: <http://www.extremeprogramming.org>

objectstyle

# XP process

- planning:
  - user stories- cust writes stories
  - Values- add values to the stories
  - Cost-expert team examine/evaluate the story and cost-measured
  - 3 ways- commit delivery -1. all story immediate 2. highest value first 3. risky story first
  - Compute - project velocity
  - Cust – can change story/eliminate story/add story
- Design:
  - Principle of KIS- keep it simple
  - CRC –class responsibility collaborator cards- OOPs classes will be identify and organise
  - Spkie solution-technical investigation.
  - Refactoring technique of improving code without changing functionality.
- Coding and testing:
  - Pair-programming- two developers team together and work on one computer. The two people work together to design, code and test user stories.
  - Continuous integration

# Agile Model



Advantages



Disadvantages



Challenges in  
adapting agile

# Agile Model - advantages



Is a very realistic [accurate] approach to software development.



Promotes teamwork and cross training.



Functionality can be developed rapidly and demonstrated.



Resource requirements are minimum.

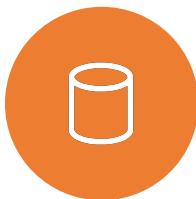


Suitable for fixed or changing requirements

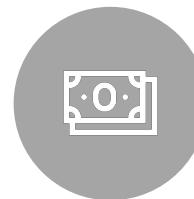


Delivers early partial working solutions.

## Agile Model - advantages



**Good model for environments that change steadily.**



**Minimal rules, documentation easily employed.**



**Enables concurrent development and delivery**



**Little or no planning required.**



**Easy to manage.**



**Gives flexibility to developers.**

# The disadvantages of the Agile Model

**Not suitable for handling complex dependencies.**

**More risk of sustainability, maintainability and extensibility.**

**An overall plan, an agile leader and agile PM practice is a must without which it will not work.**

**Strict delivery management and adjustments to meet the deadlines.**

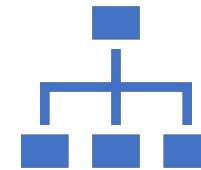
# The disadvantages of the Agile Model



**Depends heavily on customer interaction**

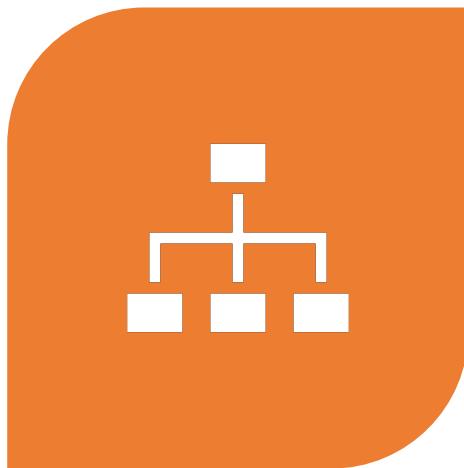


**There is a very high individual dependency, since there is minimum documentation generated.**

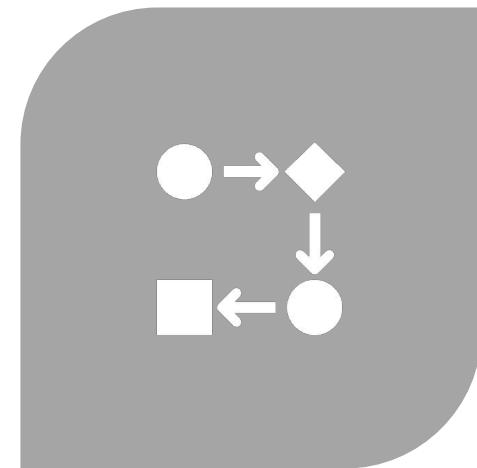


**Transfer of technology to new team members may be quite challenging due to lack of documentation.**

# Grady Booch speaks



**"PEOPLE ARE MORE  
IMPORTANT THAN ANY  
PROCESS."**



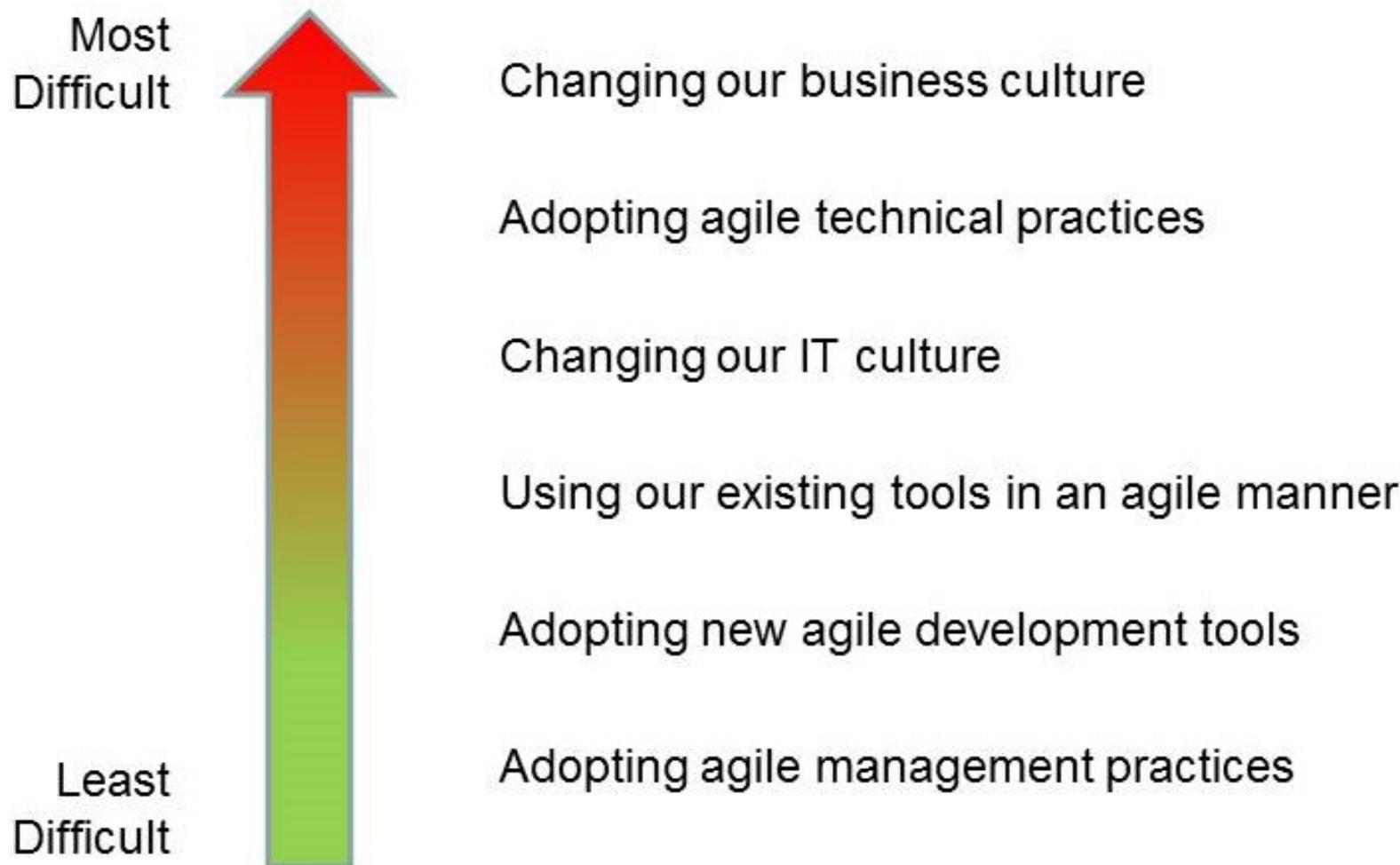
**GOOD PEOPLE WITH A GOOD  
PROCESS WILL OUTPERFORM  
GOOD PEOPLE WITH NO  
PROCESS EVERY TIME."**

# Strategies for Enterprise Agile Transformation

1. Accept the situation that you actually face
2. Stop looking for easy, prescriptive answers
3. Look beyond software development
4. Be prepared to scale agile tactically and strategically
5. Address people, process and tooling simultaneously – But focus on making your people awesome
6. Evolve enterprise IT teams in parallel with delivery teams
7. Adopt a comprehensive approach to coaching
8. Recognize that transformation is a journey, not a destination



# How difficult were the following issues to address during your Agile Adoption?



# Challenges in adopting agile methods

- 1. Quality, Cost, Time and Scope**
- 2. Ready to use' product [performance tested-data migration/setting users in live system/security/anti-hacking testing]**
- 3. Inability to 'design' for future requirements**
- 4. External and Internal Dependencies**
- 5. Need of experts in agile team**

# Watch this

[https://www.youtube.com/watch  
?v=WjwEh15M5Rw&t=229s](https://www.youtube.com/watch?v=WjwEh15M5Rw&t=229s)

# questions

- Explain briefly agile process
- Scrum overview and challenges in adopting agile methods
- What is extreme programming [EP]
- Lean six sigma

# Thank You!!!



# PRESENTATION TOPICS

- Scrum
- Lean
- Six sigma
- Kanban
- Dsdm(Dynamic system dev methods)
- Fdd
- Crystal
- Xp