Agile

Prepared by Veda Shyamala

Contents

SDLC
WATERFALL MODEL
Overview
Phases
Challenges
Revision
AGILE
Process
Overview
Iterative & Incremental
Development phases
Values
Principles
Work Process
Devops
Benefits
Agile vs Waterfall
Revision
Quiz1:Agile

SCRUM			
Overview	**SCRUM TEAM**	**ARTIFACTS**	**SCRUM EVENTS**
Framework	Accountabilities	Commitments	Description
Empirical Process	***Product Owner***	Product Backlog	***Sprint***
Complex	PBM	***Product Goal***	***Sprint Planning***
Approach Benefits	***Developers***	PB Refinement	***Daily Scrum***
Pillars of Empiricism	***Scrum master***	Themes & Epics	***Sprint Review***
Values	Scrum master service	***Sprint Backlog***	Review Mechanics
Essentials	competencies	Sprint Goal	***Sprint retrospective***
Process	Revision	***Increment***	Revision
Revision	Quiz3:Scrum Team	Definition of Done	Quiz5:Scrum Events
Quiz2:Scrum		Revision	Summary of Scrum
		Quiz4: Artifacts	Benefits of Scrum

SAFe	SAFe & Scrum	When to use SAFe	SAFe Principles

What is SDLC

A process followed in software projects for creating high-quality software

It focusses on the various phases like planning, creating, testing and deploying

Each Phase consists of different tasks to be performed

Types of SDLC – Waterfall and Agile software development

Waterfall model

What is Waterfall model

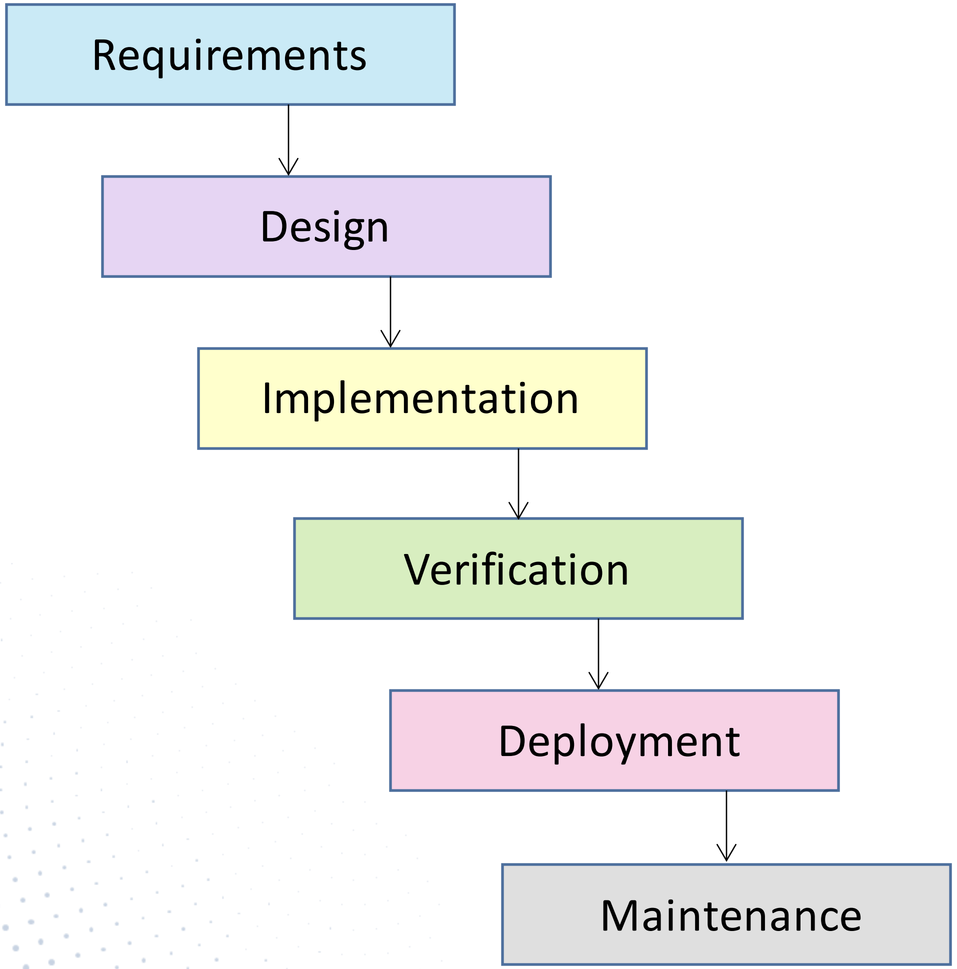
- It is the earliest SDLC approach that was used for software development

- It illustrates the software development process in a linear sequential flow

- Sequential model with top-down approach and very clear picture of final product

- Suited for smaller projects where requirements are well defined

Waterfall model phases



- Serialized process in consecutive order
- When requirements are not changing frequently
- Elaborate documentation at various phases
- Pre-planned and executed using formal change control

Challenges of Waterfall model

Not desirable for complex project	Client feedback only after product is built
Minimum client intervention	Testing only after development phase
Documentation at every phase	Makes changes difficult
Delay in phases	Working model delivered at later phases

Revision : Waterfall model

- Linear sequential flow of all phases in consecutive order
- Suitable for smaller projects where requirements are not changing
- Changes difficult and minimum client intervention
- More time taken for delivery since delay in phases

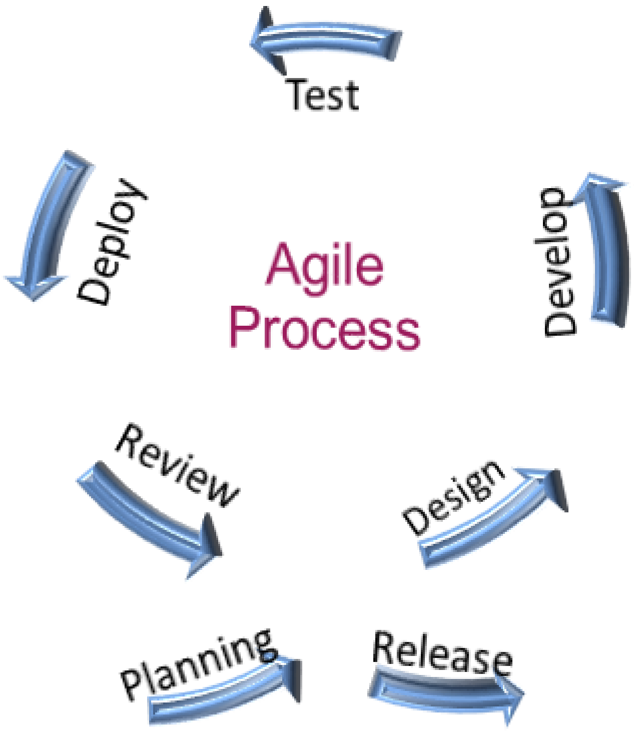
Agile

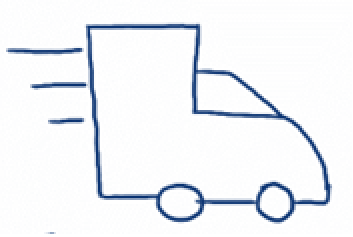
ACCELERATE NEXT

Agile Process

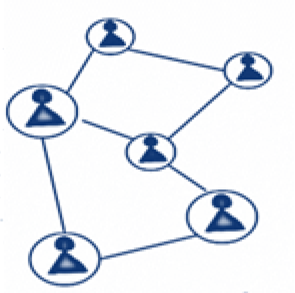
Faster ,Better & Happier

Respond to change

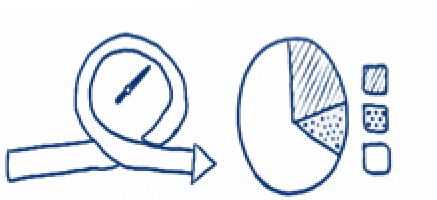




Faster delivery



Customer Collaboration



Iterative & Incremental

Overview: What is Agile

- It is a software development approach under which requirements and solutions evolve through collaborative effort of self-organizing and cross functional teams and their customer / end user encouraging flexible response to change

- ***Self-organizing*** : choose how best to accomplish their work, rather than being directed by others outside the team

- ***Cross functional*** : have all competencies needed to accomplish the work without depending on others not part of the team

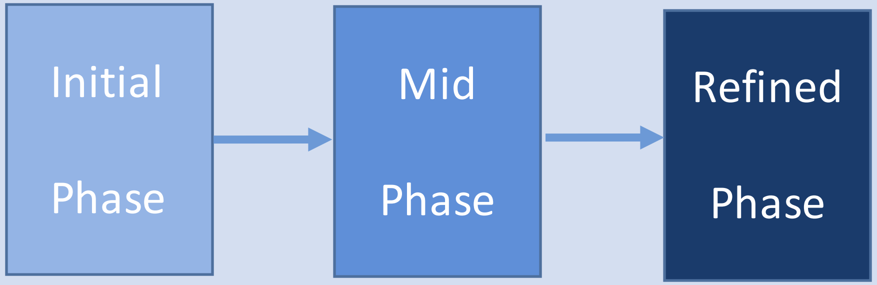
- The ability to create and respond to change in order to succeed in an uncertain and turbulent environment

- Both development and testing activities are concurrent

- **various agile *frameworks* -** Scrum.. Extreme programming.. Lean..Kanban..Crystal

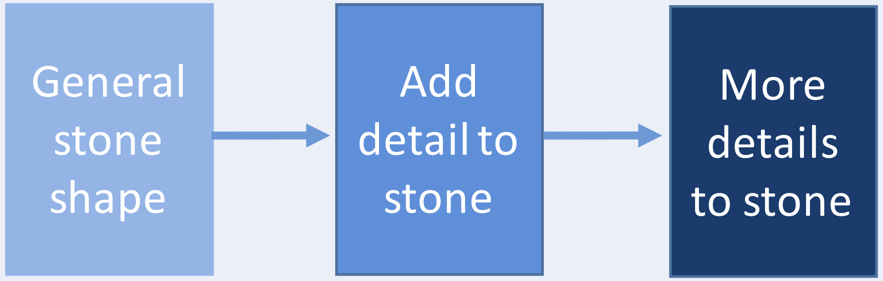
Iterative and Incremental Process

Iterative process



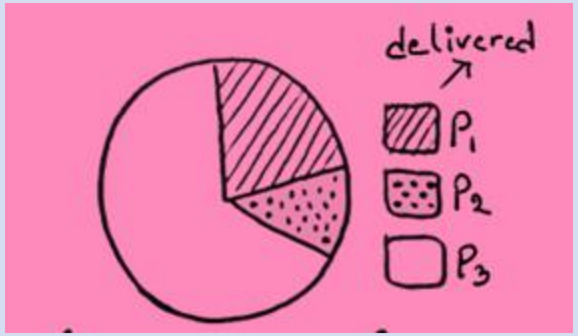
Initial Phase → Mid Phase → Refined Phase

Ex: Sculpting of bird



General stone shape → Add detail to stone → More details to stone

Incremental Process



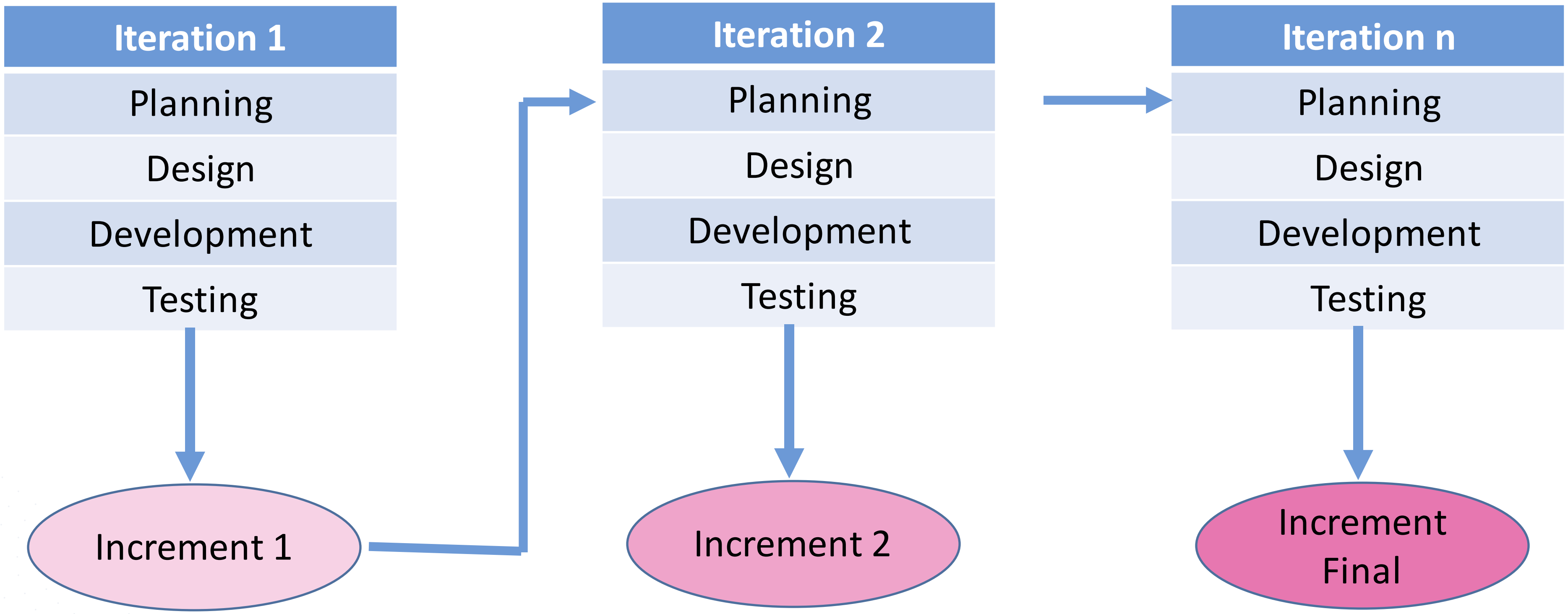
Ex: Sculpting of bird

Eyes ▨

Beak ▒

Head □

Iterative and Incremental Development



Values of Agile

1

Individuals and Interactions

------------ over ------------

Processes and Tools

2

Working software

------------ over ------------

Comprehensive Documentation

3

Customer collaboration

------------ over ------------

Contract Negotiation

4

Responding to change

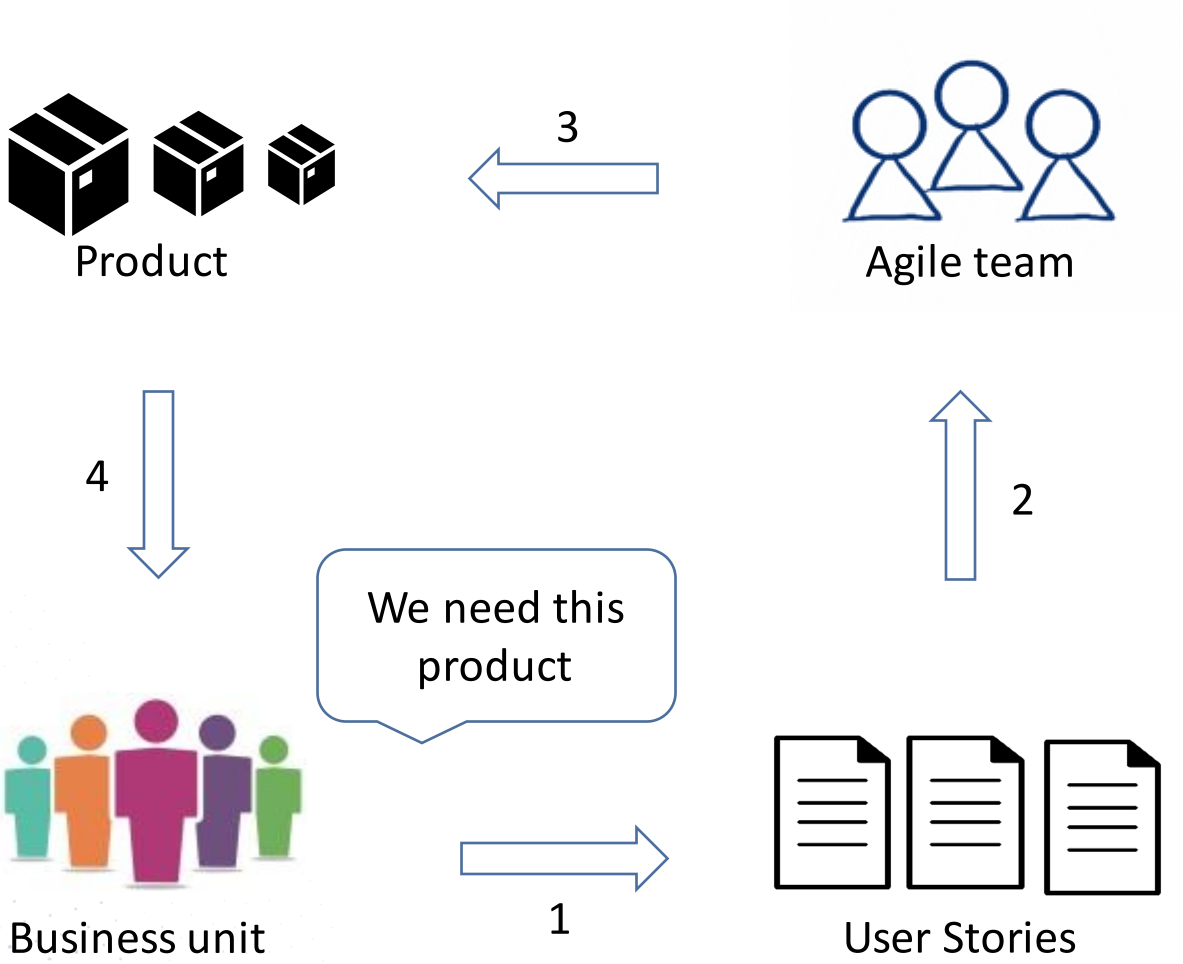
------------ over ------------

Following a Plan

Agile principles

Customer satisfaction	Welcome change	Deliver frequently	Working together
Motivated team	Face-to-face	Working software	Constant pace
Good design	Simplicity	Self-organization	Reflect and Adjust

How Agile Process Works



- Business Units ask for Product
- Provide user stories as requirements
- Agile team orders and estimates the stories
- Team Self-organized and cross-functional
- Follow Iterative and incremental approach
- Smaller build done and delivered
- Feedback taken from stakeholders
- Adjustments and refinements done
- Process continuous until final product done

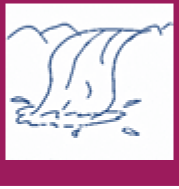
How Agile combines with Devops to contribute

- Both DevOps and agile are modern software development practices aimed at providing a framework to produce a part of a product, a launch, or a release
- DevOps addresses gaps and conflicts between the Developers and IT Operations
- Agile addresses gaps and conflicts between the customer and developers
- The most important thing to know about DevOps versus agile is that they are not mutually exclusive
- Both DevOps and agile offer a structure and framework that can speed software delivery

Benefits of Agile

Plan with minimum requirements	Welcome to changes at any stage
Incremental and iterative approach	Scale-up projects easy
Frequent Inspection and adaptation	Working software delivered frequently
Customer interactions and feedback	High-performance and production

Agile vs Waterfall

AGILE 	WATERFALL 
Supports changing Requirements	End Product is Defined
Iteration and Incremental	Fixed and separate phases
Product Mindset	Project Mindset
Feedback Loop	No feedback until testing
Collaboration of Team	Teams work in turns
Transparency	Lack of transparency
Unclear requirements	Fixed Requirements
Easy to incorporate changes	Difficult to enable changes
Minimal documentation	Extensive Documentation

Revision : Agile

- Smaller builds delivered
- Iterative and Incremental
- Self-organization and cross-functional
- Respond to change
- Customer collaboration and feedback
- Faster delivery
- Suitable for complex projects

Quiz1 : Agile

1.How is Agile planning different from the Waterfall model planning?

A. Agile planning is done only once

B. Agile planning is non iterative

C. Agile planning places emphasis on the plan

D. Agile planning places emphasis on planning and is iterative

2. In Agile projects, we plan to "learn as we go" because…

A. It is more fun for the developers

B. Many projects are evolutionary, and a better solution emerges this way

C. It is boring to try to analyze everything at the beginning of a project

D. It gives a good reason for late delivery

3. **An Agile team …**

A. Is self-organizing, so needs no specific skills

B. Collaborates and supports team members

C. Ensures that weak members of the team are thrown out of the team

D. Ensures blame is allocated fairly

Scrum

What is Scrum

- Framework for implementing Agile developed by Ken Schwaber and Jeff Sutherland
- Founded on Empiricism and Lean thinking
- For addressing complex adaptive problems
- While creatively and productively developing iteratively
- Incorporating customer feedback on the working software
- Delivering products of highest possible value incrementally

Why Scrum is a framework

- It is not complete - Provides structure and direction to do something without being too detailed or rigid. Only the parts required to implement scrum theory are defined

- Built upon collective intelligence - It gives general purpose ideas to accomplish some specific purposes(events) or items(backlog) for specific uses rather than provide people with detailed instructions, the rules of scrum guide their relationships and interactions

- No predefined rules - Scrum wraps around existing practices or renders them unnecessary.

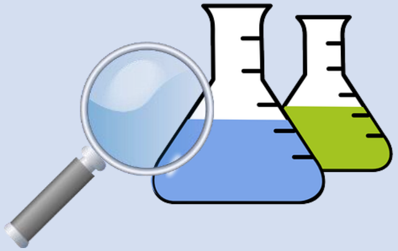
- Various processes and techniques are employed - It makes visible the relative efficacy of current management, environment and work techniques, so that improvements can be made

Empirical process control theory

Empiricism: Progress based on observation of reality and not on plans

Learn as we progress based on

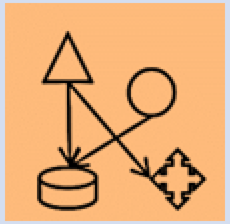
- Observation
- Explanation



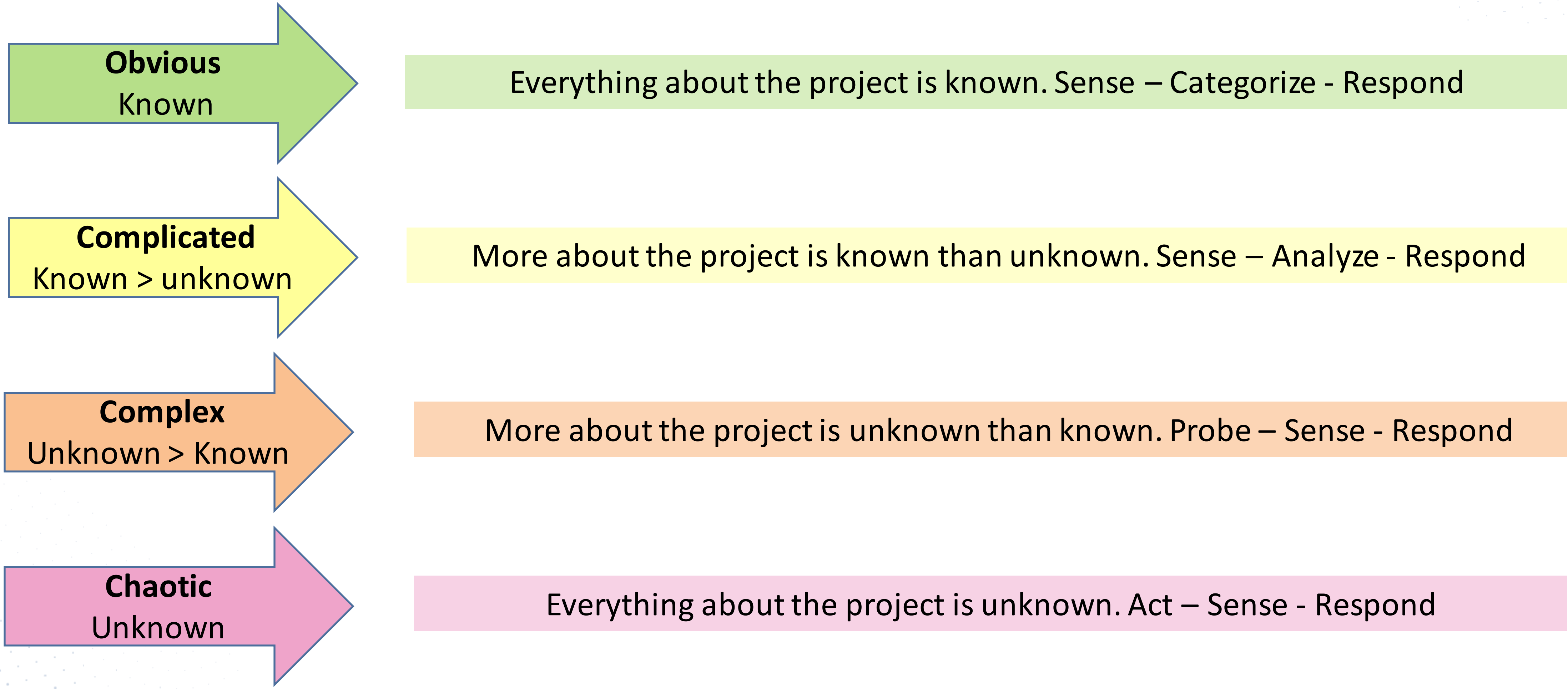
Working in a

- Fact – based
- Evidence – based
- Experience - based

Embracing and expecting change



Complex adaptive problems



Obvious
Known

Everything about the project is known. Sense – Categorize - Respond

Complicated
Known > unknown

More about the project is known than unknown. Sense – Analyze - Respond

Complex
Unknown > Known

More about the project is unknown than known. Probe – Sense - Respond

Chaotic
Unknown

Everything about the project is unknown. Act – Sense - Respond

Benefits of Iterative and Incremental approach

- Maximizing opportunities for feedback
- Optimize predictability and control risk
- Generating working software quickly and early during SDLC
- More flexible, less costly to change scope and requirements
- Easier to test, debug and identify errors
- Easier to manage and handle risk

Pillars of Empiricism

Transparency	Inspection	Adaptation
▪ Open work culture ▪ Common language shared ▪ Done definition shared	▪ Scrum Artifacts ▪ Sprint Goal progress	▪ Adjustment ▪ Continuous improvement

Scrum Values

Commitment

“People personally commit to achieving the goals of the Scrum Team.”

Focus

“Everyone focuses on the work of the Sprint and the goals of the Scrum Team.”

Openness

“The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work.”

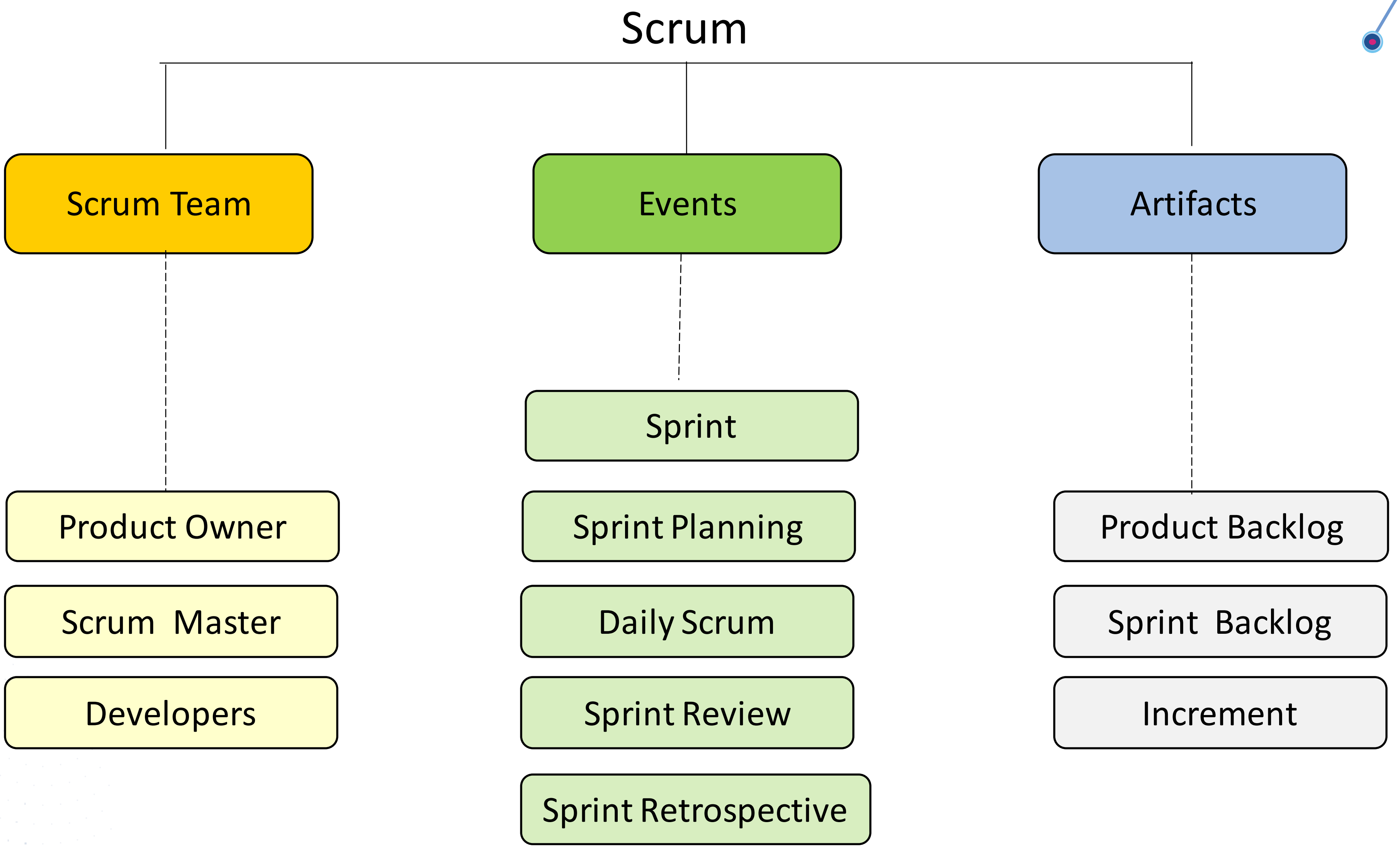
Respect

“Scrum Team members respect each other to be capable, independent people.”

Courage

“Scrum team members have the courage to do the right thing and work on tough problems.”

Scrum (11 essentials)



Scrum

Scrum Team

- Product Owner
- Scrum Master
- Developers

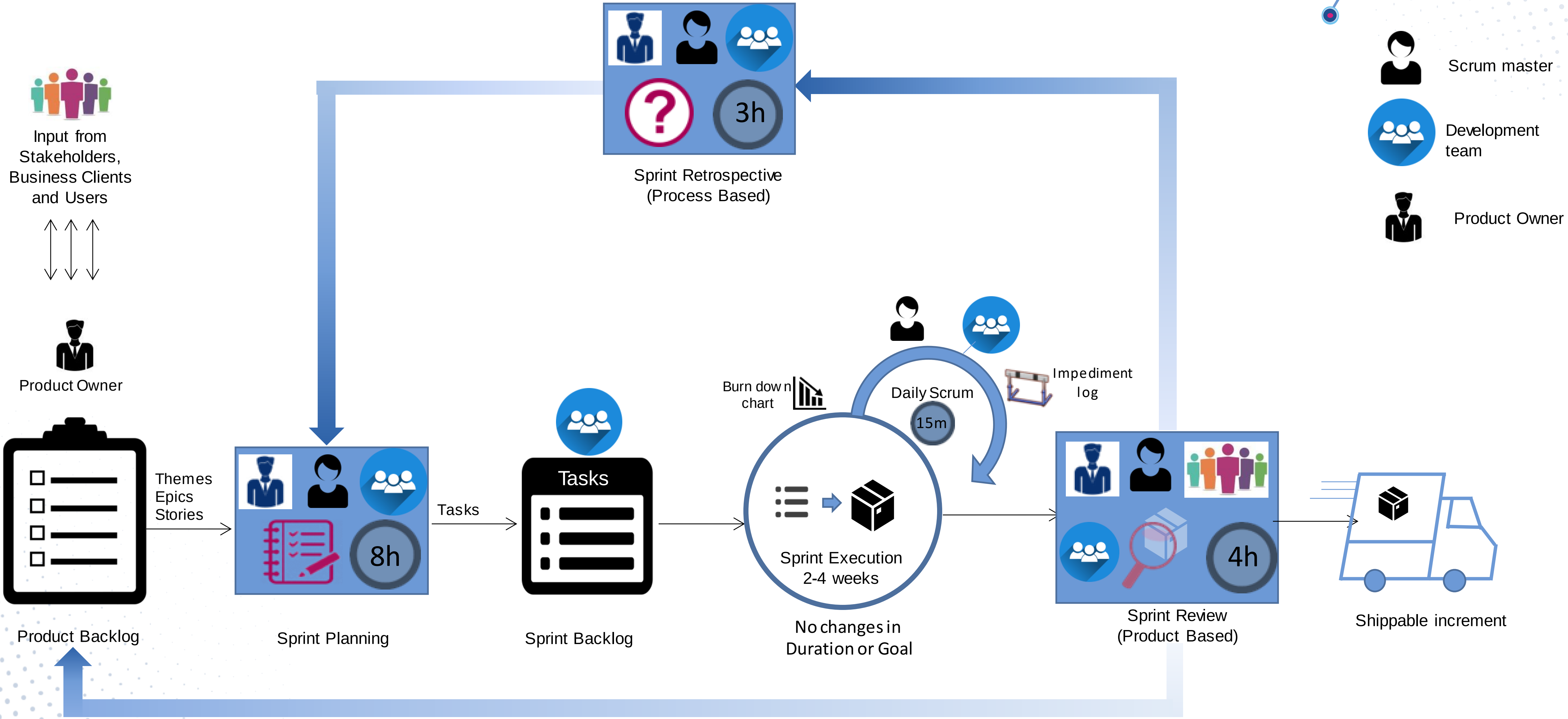
Events

- Sprint
- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Artifacts

- Product Backlog
- Sprint Backlog
- Increment

Scrum Process



Input from Stakeholders, Business Clients and Users

Product Owner

Product Backlog

Themes
Epics
Stories

Sprint Planning

8h

Tasks

Tasks

Sprint Backlog

Sprint Retrospective
(Process Based)

3h

Burn down chart

Daily Scrum

15m

Impediment log

Sprint Execution
2-4 weeks

No changes in Duration or Goal

Sprint Review
(Product Based)

4h

Shippable increment

Scrum master

Development team

Product Owner

ACCELERATE NEXT

Revision : Scrum

- Framework as it is incomplete with only defining the parts required
- Various processes and techniques employed
- Founded on empiricism and lean thinking
- Knowledge comes from observation and experience
- Employs iterative and incremental approach

Quiz2 : Scrum

1. Which statement best describes Scrum?
A) A defined and predictive process that conforms to the principles of Scientific Management.
B) A complete methodology that defines how to develop software.
C) A framework for creating complex products in complex environments.

2. An organization has decided to adopt Scrum, but management wants to change the terminology to fit with terminology already used. What will likely happen if this is done?
A) Without a new vocabulary as a reminder of the change, very little change may actually happen.
B) The organization may not understand what has changed with Scrum and the benefits of Scrum may be lost.
C) Management may feel less anxious.
D) All of the above.

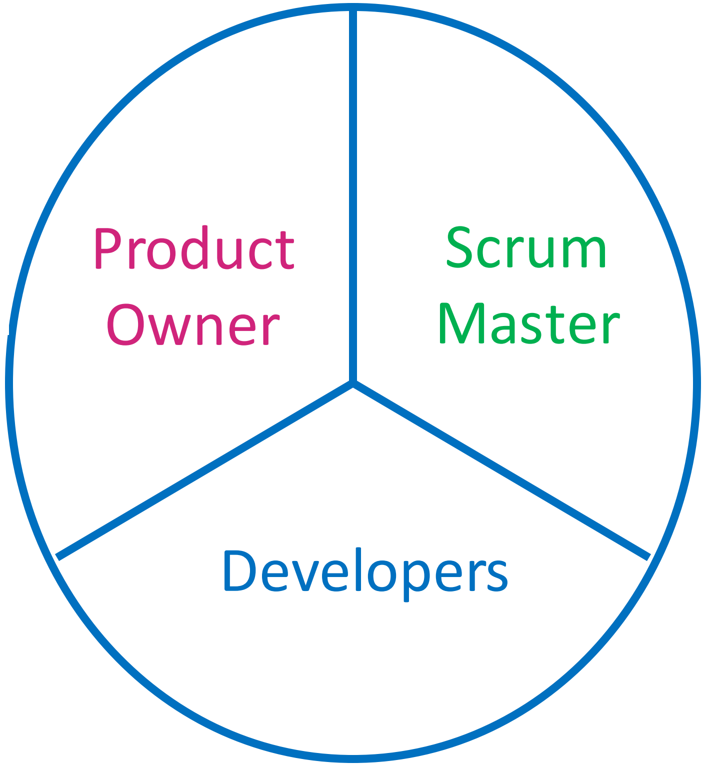
3. What is the role of Management in Scrum?
A) Identify and remove people that are not working hard enough.
B) Monitor the progress of the Developers on the Scrum Team.
C) Support the Product Owner with insights and information into high value product and system capabilities. Support the Scrum Master to cause organizational change that fosters empiricism, self-management, bottom-up intelligence, and intelligent product delivery.

Scrum Team

ACCELERATE NEXT

Scrum Team Accountabilities

Maximizing the value of the product managing the PB



Promote and support Scrum removing impediments

Create quality increments which are potentially releasable

Accountability of Product Owner

Ideally have profit and loss accountability for the product

- Sole person responsible for managing the product backlog
- Maximizes the value of the product
- Chooses what and when to release
- Represents the stakeholders and customers to the development team
- Product Backlog management

Product Backlog Management

Accountability of Product Owner
Developing and explicitly communicating the Product Goal
Creating and clearly communicating Product Backlog Items
Ordering items in PB to best achieve goals
Ensuring the PB is visible, transparent and clear

Accountability of Developers

Self-organizing and Cross-functional

- Create a plan for the Sprint, the sprint Backlog
- Build the product increment
- Estimates the product backlog
- Sprint backlog changes and responsibility
- Size of the team between 3 and 9
- Adapting their plan each day toward the Sprint goal
- No sub-teams or titles other than Developer

Accountability of Scrum master

Facilitator, Personifies Agility

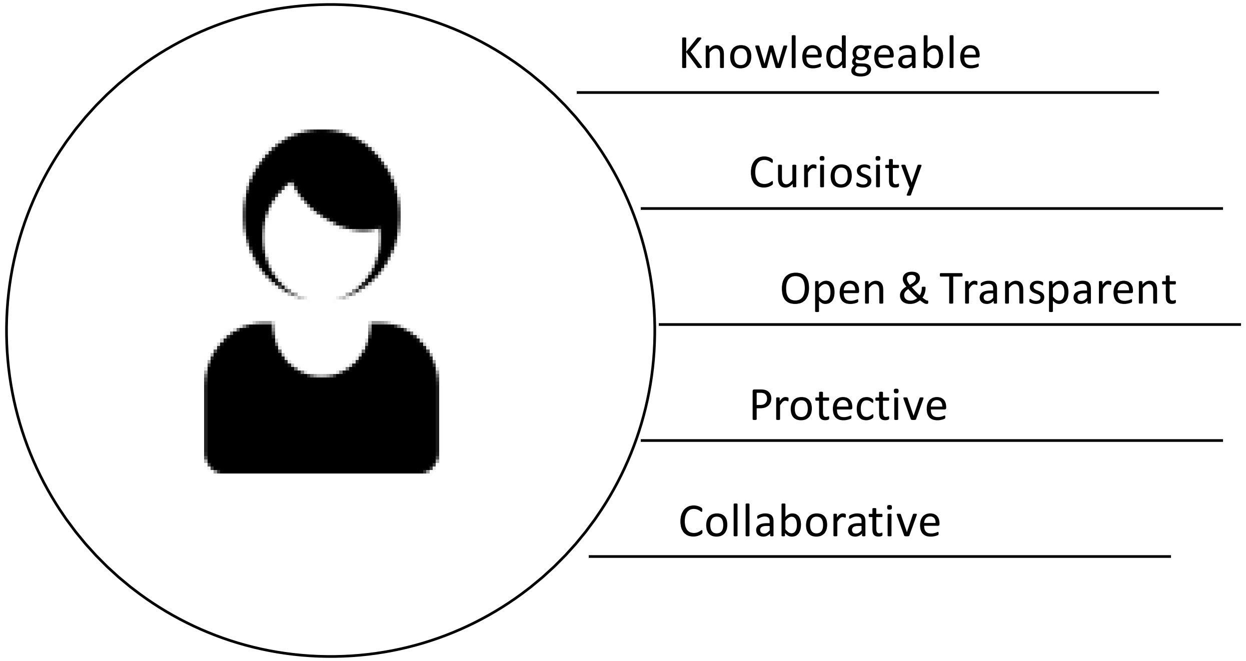
- Ensures Scrum is understood and enacted
- Protecting the team from outside interruptions and distractions
- Promotes and supports Scrum removing impediments
- Provides guidance and support for the Scrum team and organization
- Works with Team to increase transparency of Artifacts

Scrum Master Service

To Team	To Product Owner	To Organization
Coaching in self-management and cross functionality	Finding techniques for effective Product Goal definition and PBM	Leading and coaching organization in Scrum adoption
Helping on creating high-value increments that meet the definition of done	Helping team understand clear and concise PBI	Planning and advising Scrum implementation within the organization
Removal of impediments to Scrum Team's progress	Helping establish empirical product planning for complex environment	Helping employees and stakeholders understand and enact empirical approach
Ensuring all events take place and kept within the time box	Facilitating stakeholder collaboration as requested or needed	Removing Barriers between stakeholders and scrum teams

ACCELERATE NEXT

Competencies of Successful Scrum Master



Knowledgeable

Curiosity

Open & Transparent

Protective

Collaborative

Revision: scrum team

- Product owner orders the stories into product backlog
- Developers estimate and prepare sprint backlog
- Selection of work turned into increment of value during a sprint
- Results are inspected and adapted for the next sprint
- Repeat

Quiz3: Scrum Team

1. Who is responsible for prioritizing the product backlog?
A) Product Owner
B) Project Manager
C)Lead Developer
D) Tester

2. Who should know the most about the progress toward a release, and be able to explain the alternatives most clearly?
A) The Project Manager
B) The Scrum Master
C) The Developers
D) The Product Owner

3. What are two ways a Scrum Master serves to enable effective Scrum Teams?
A) By starting and ending the meetings at the proper time.
B) By facilitating Developer decision-making.
C)By removing impediments that hinder the Scrum Team.
D)By keeping high value features high in the Product Backlog.

Artifacts

Artifacts & their Commitments

Artifacts	Description	Commitments
Product Backlog	Holds the requirements for the Product Managed by Product Owner	Product Goal
Sprint Backlog	Holds all the work for current Sprint Managed by Developers	Sprint Goal
Increment	Stepping stone toward the Product Goal Working addition to the product	Definition of done

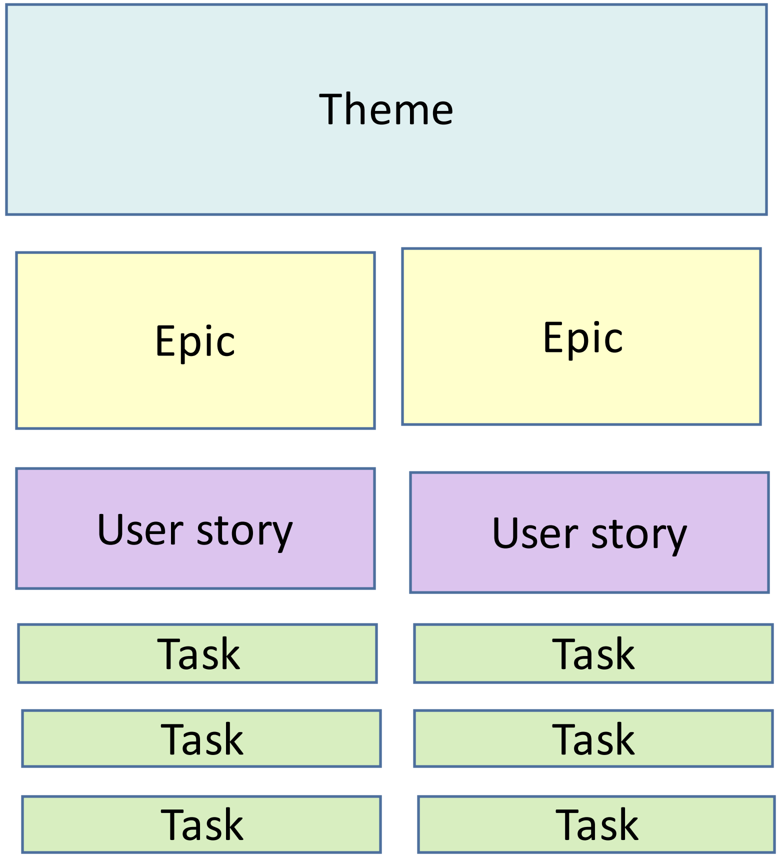
Theme, Epics, stories and tasks

Theme: Large focus areas that span the organization and collections of epics that drive toward a common goal.

Epic: Epics are usually broad in scope, lacking in details, and are meant to be split into multiple, smaller stories before they can be worked on. Epic is usually regarded as the ‘top tier’ or a work hierarchy.

User story: It is simply the list of items that need to be done within a project.

Task: Small incremental units of work that are required to complete a story



Artifact: Product Backlog

- Transparent unit of deliverable work
- Only one Product Backlog for multiple scrum teams and one product
- Higher order of PBI are more clear
- Expression and ordering done by Product Owner
- Constantly revisited and re-prioritized during Product Backlog Refinement
- Estimates done by Developers
- Commitment: Product Goal

Commitment of Product Backlog: Product Goal

- Measurable stepping stones towards product vision. They are achievable and measurable
- Describes future state of the product and Long term objective for the scrum team
- Market driven, not Product Backlog driven.
- As each Increment is produced, the Product incrementally moves toward the Product Goal
- Product Goals are a result of responding to *emerging* market conditions
- The Product Goal is made transparent in the Product Backlog

Product Backlog Refinement

- Adding detail, estimates and order to items of Product Backlog
- Planning the Product backlog to an actionable level of detail
- Consumes no more than 10% of development team capacity
- Larger requirements broken into smaller tasks
- Product Owner and developers collaborate on details
- Factors causing Product backlog changes

Product Backlog Estimation

- Assessment of how much delivery of each user story will take

- Estimates by using more abstract metrics

- Planning poker/ Scrum poker

- A typical deck has cards showing the Fibonacci sequence

Artifact: Sprint Backlog

- Consists of the plan for delivery of Increment
- Set of Product Backlog Items selected for the Sprint
- Updated throughout the Sprint
- Owned and managed by Developers
- At least one high priority improvement from sprint retrospective
- Commitment: Sprint Goal

Commitment of Sprint Backlog: Sprint Goal

- An objective to be met in the sprint through the implementation of Product Backlog
- Created during sprint planning where selected PBI’s deliver one coherent function
- Guidance to developers on why the increment is built
- Allows flexibility in delivering the increment
- Scrum team craft the Sprint goal and is fixed throughout the sprint
- Developers start designing system and work needed to convert Product Backlog into increment

Artifact: Increment

- Stepping stone towards the product goal
- Each Increment is additive to all the previous increments
- Multiple increments may be created within a sprint
- Increment will be delivered by the developers
- Product owner decides about the release
- Commitment: Definition of Done

Commitment of Increment : Definition of Done

- Guides developers of how many PBI can be selected during sprint planning
- Ensures artifact transparency and shared understanding of work completed
- Increment born when Product Backlog item meets definition of done
- Multiple scrum teams mutually define and comply with same definition of done
- Used to access when work is complete on the product increment
- Each increment is additive to all prior increments

Revision : Artifacts

- Represent work and future goals
- Provide transparency
- Opportunities for inspection and adaptation
- Product Backlog: what to do
- Sprint backlog: What to take on now + how
- Increment : what gets done

Quiz4 : Artifacts

1. How much work must the Developers complete for each Product Backlog item they select for a Sprint?
A) Analysis, design, programming, testing and documentation.
B) As much as it can fit into the Sprint.
C) All development work and at least some testing.
D) Enough so that each Product Backlog item they select meets the Definition of Done.

2. How a Product Backlog should be ordered?
A)Based on the size of the items
B)Based on the risk of the items
C)Based on the float of the items
D)Based on the value of the items

3.Who turns the product backlog into incremental pieces of functionality
A)Product Owner
B)Scrum Master
C)Scrum Team

Scrum events

Scrum Events

Events	Description
Sprint	From : Product Backlog
	To: Increment
Sprint Planning	From: Product Goal, Product Backlog
	To: Forecast, Sprint Backlog, Sprint Goal
Daily Scrum	From: Daily Progress, Tasks, Impediments
	To: Updated daily plan & Sprint Backlog, Impediment log
Sprint Review	From: Increment
	To: Updated product Backlog
Sprint Retrospective	From: Past Sprint
	To: Improvements for next sprint

Scrum Event: Sprint

- Time-boxed Container for all other scrum events
- Fixed length event of one month or less
- New sprint starts after conclusion of previous sprint
- No changes are made that would endanger Sprint Goal
- Starts with Sprint Planning and ends with Sprint Retrospective
- Scope may be clarified/re-negotiated between Product owner & developers
- Only product owner can cancel Sprint when Sprint Goal becomes obsolete

Scrum Event: Sprint Planning

- Time-boxed event <=8 hrs. for planning current Sprint work by Scrum Team
- Product Backlog is inspected and PBI selected for Sprint Backlog creation
- Sprint Goal set for the sprint which describes why it is building the increment
- Addresses why the sprint is valuable
- What can be delivered this sprint is discussed
- Consists of how the work needed to deliver be achieved
- Collaborative work of whole scrum team
- Scrum Team can invite technical or domain advice from outside team

Scrum Event: Daily Scrum

- Daily 15 mins Time-boxed event for developers
- Held at same place and time
- Key inspect and adapt meeting
- Create a plan for next 24 hrs.
- Adapt and adjust the Sprint Backlog , identify impediments
- Share commitments and create focus

Scrum event: Sprint Review

- Four hrs. time-boxed event
- Scrum Team presents the results to Key stakeholders
- Developers discuss work done and Impediments
- Product Backlog is updated upon new insights
- Review of timeline, budget, potential capabilities
- Stakeholders provide feedback and collaborate
- Progress towards the Product Goal is discussed

Mechanics of Sprint Review

Product Owner shares	Developers share	Everyone
What was done	Actual increment of software	Provide and hear feedback
What wasn’t done	What happened in sprint	Inspect and adapt opportunity
State of the Product Backlog	How problems were addressed	Collaborate on what to do next
Projections of likely release targets	Effect on the increment	Review on potential use of product

Scrum team and stake holders attend

Product Owner explains done and not been done

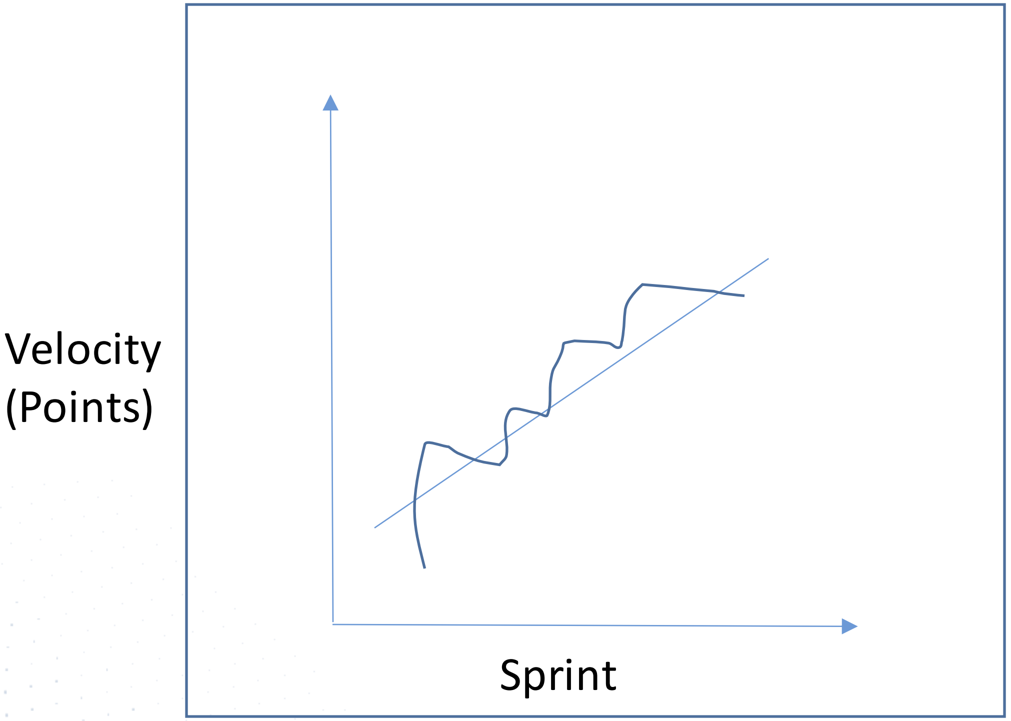
Product Based Event to inspect, increment and adapt

Revised Product Backlog for next sprint is the output

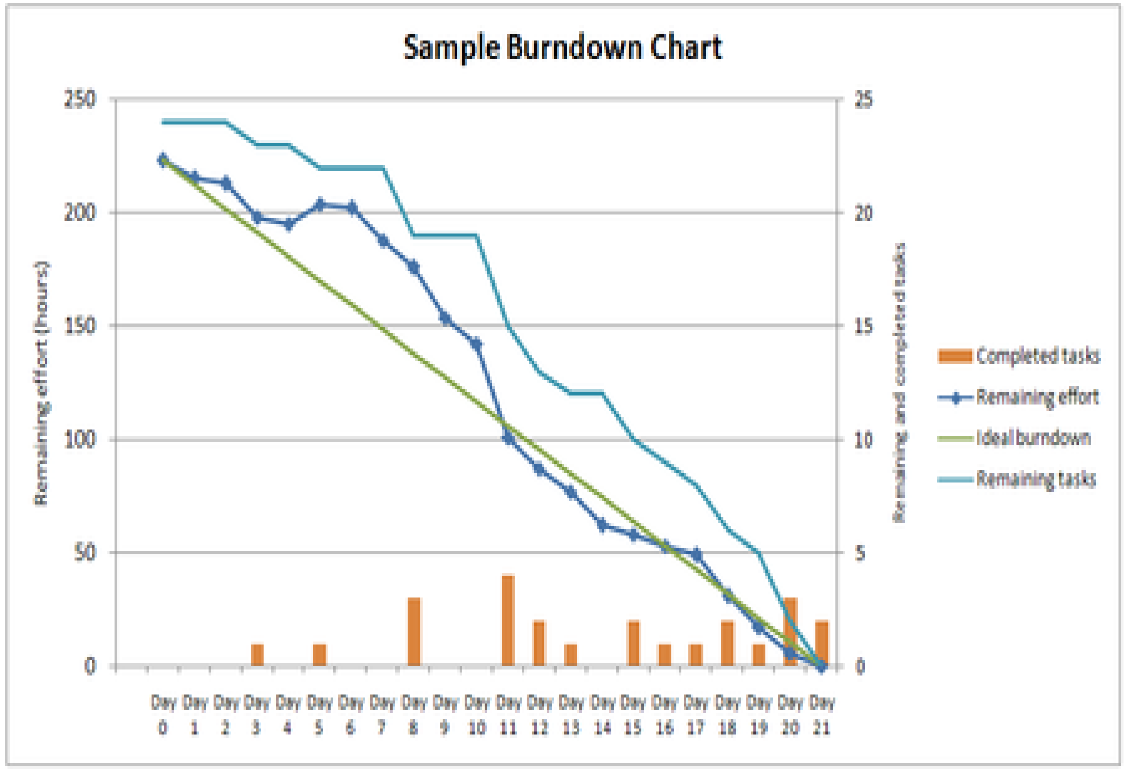
Scrum event: Sprint Retrospective

- Process based time-boxed event to 3 hrs.
- Plan ways to increase quality and effectiveness
- Inspect how the last sprint went with regards to people, relationships, processes and tools
- Identify and order the major items that went well and potential improvements
- Create a plan for implementing improvements(product quality)
- Improvements identified for next sprint implementation and definition of done updated

Metrics



Velocity
(How many points team can produce in each sprint)



Sprint Burndown chart
(Tracks Progress)

Revision : Scrum Events

- Sprint : container for all events
- All events are time-boxed
- Fixed sprint duration and goal
- Requirements and solutions evolve
- Events : opportunity to inspect and adapt
- Shippable Increment

Quiz5 : Scrum Events

1. When is a Sprint over?
A) When all the tasks are completed.
B) When the timebox expires.
C) When all Product Backlog items meet their Definition of Done.
D) When the Product Owner says it is done.

2. What does it mean to say that an event has a timebox?
A) The event must take at least a minimum amount of time.
B) The event must happen at a set time.
C) The event must happen by a given time.
D) The event can take no more than a maximum amount of time.

3. Which statement best describes the Sprint Review?
A) It is a mechanism to control Developer activity during a Sprint.
B) It is when the Scrum Team and stakeholders inspect the outcome of a Sprint and figure out what to do next.
C) It is a demo at the end of the Sprint for everyone in the organization to check on the work done.

Summary of Scrum

- Scrum implements empiricism in software development
- Every Scrum role(3) has a clear accountability
- The Scrum artifacts (3) provide transparent information
- All Scrum events(5) serve inspection , adaptation and transparency
- Iterative and increment way of delivering working software
- Feedback incorporated frequently to ensure right business value delivered

Benefits of Scrum

- Deliver smaller features to customers with higher efficiency and speed. so greater customer satisfaction
- Large projects are divided into easily manageable sprints.
- Greater ability to incorporate changes as they occur
- Team collaboration and valuable work delivered
- Early detection and Faster problem solving limiting complexity of problems
- Continuous software delivery & successive refinements lead to building better products
- Ensure higher success rates for releases & speed software delivery

SAFe - Scaled Agile Framework

- It is framework with a set of organization and workflow patterns for implementing agile practices at enterprise scale

- Originally called ‘Agile enterprise big picture, SAFe was released by Dean Leffingwell and Drew Jemilo in 2011. The latest version is SAFe 5.0

- SAFe promotes alignment, collaboration, and delivery across large numbers of agile teams

- Designed to help businesses continuously and more efficiently deliver value on a regular and predictable schedule

- It has two dimensions

  Vertical Scaling: It is several independent teams, jointly planning, developing, integrating, testing and deploying a product based on an integrated product vision

  Horizontal Scaling: It is adopting a lean-agile mindset and agile concepts at functional level- such as HR, sales or finance with different other groups collaborating

Difference between SAFe and scrum

Scrum	SAFe
It is adopted by the agile teams	Adopted by enterprise as a whole(teams of agile teams)
Lightweight framework	comprehensive framework targeted to enable Business Agility of an enterprise
Deals with small collocated teams	Deals with big multi geography teams
Level of dependency between teams is low	High level of coordination and alignment across teams required
The middle management plays no role.	Program and Portfolio management are two important tiers of SAFe®.

When to use SAFe

- 1. Managing inter-team dependencies to deliver an Increment
- 2. From vision to product / aligning enterprise level with team level
- 3. Align Product decision makers and the Product Owner
- 4. Give top level of the organization a control mechanism
- 5. Syncing delivery / integrating increments
- 6. Improve productivity
- 7. Increase value delivery

SAFe lean-agile principles

SAFe is based on ten fundamental concepts that have evolved from Agile principles and methods, Lean product development, systems thinking, and observation of successful enterprises.

#1 Take an economic view

#2 Apply systems thinking

#3 Assume Variability; preserve options

#4 Build incrementally with fast, integrated learning cycles

#5 Base milestones on objective evaluation of working systems

#6 Visualize and limit Work in Process (WIP), reduce batch sizes, and manage queue lengths

#7 Apply cadence, synchronize with cross-domain planning

#8 Unlock the intrinsic motivation of knowledge workers

#9 Decentralize decision making

#10 Organize around value

JIRA

ACCELERATE NEXT

What is Jira

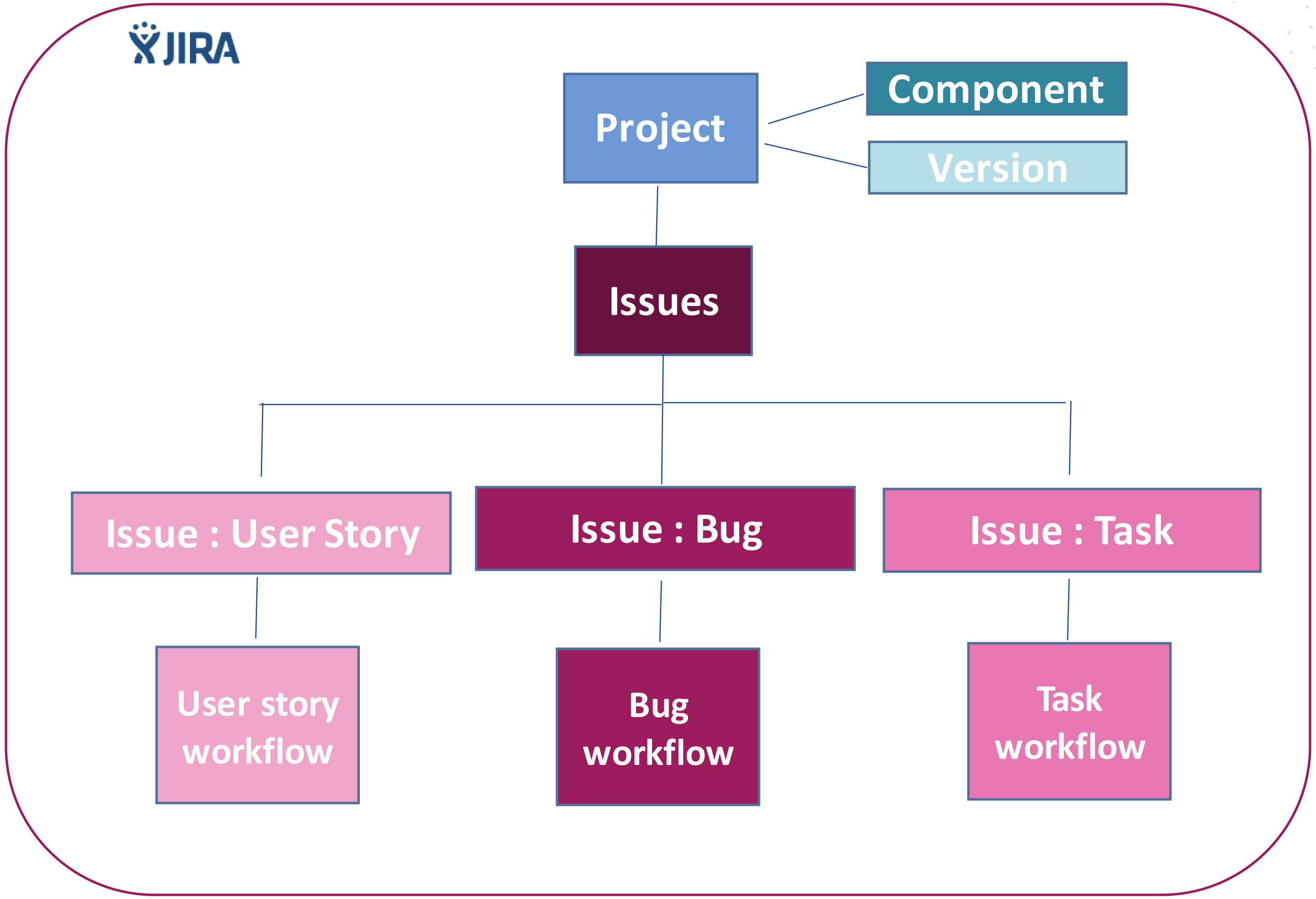
Issue and Project Tracking Software

Used by agile development teams to

- Plan
- Track and Manage projects
- Customize workflows
- Team collaboration
- Release software

Defect tracking/Project Management tool by Atlassian, Inc.

It is a platform-independent software



MAVERIC

ACCELERATE NEXT

GLOBAL LOCATIONS

Singapore | UK | US | UAE | Riyadh | Malaysia | Mexico

CORPORATE HEADQUARTERS

2nd Floor, Block-5,
DLF IT Park – SEZ,
1/124, Shivaji Gardens,
Mount Poonamalle Road,
Manapakkam, Chennai – 600 089
Tamil Nadu, India
www.maveric-systems.com