EMERGING SOFTWARE ENGINEERING TRENDS

- By F.M.Inamdar

- **Technology Evolution** : Rather that saying Technology Evolution as in programming language such as Java , Python, ML its about the technology practices evolution. Which is to be done in software engineering.
- In the context of **Technology Practices Evolution in** software engineering the technology evolution is to be understood as :
- 1. Agile
- 2. DevOps
- Because of advent of these two technology , **Technology Practices has been evolved.**

- For Ex. In early stage of software development – Integration is used to be Big Bang Integration & there use to be state process integration & change management & Integration.
- But now we have continuous integration - & change management is embedded over there & its continues job.
- Means its Continuous Function rather than Process Oriented Function.

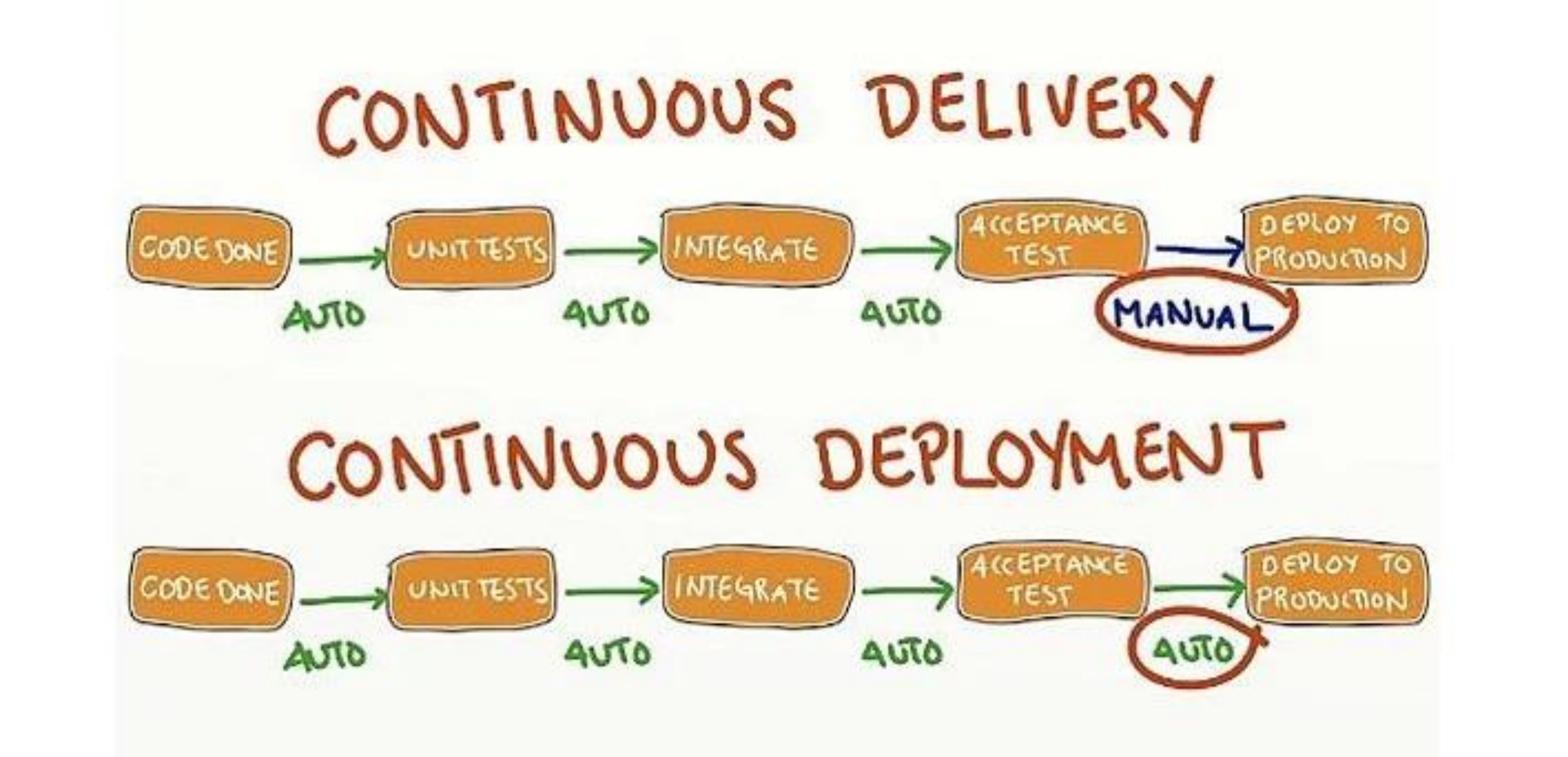
- Big Bang Integration : there is to be state process for integration as well as change management.

- Because of advent of these two technology practices evolution we have Continuous Integration & Change Management has been embedded once y

- In Software evolution it consist of
- 1. Continuous Integration (CI)
- 2. Continuous Delivery(CD)

CONTINUOUS INTEGRATION

- Continuous Integration (CI) is a development practice where developers integrate code into a shared repository frequently, preferably several times a day. Each integration can then be verified by an automated build and automated tests. While automated testing is not strictly part of CI it is typically implied.
-


CONTINUOUS DELIVERY
CODE DONE
AUTO
UNIT TESTS
AUTO
INTEGRATE
AUTO
ACCEPTANCE TEST
MANUAL
DEPLOY TO PRODUCTION
CONTINUOUS DEPLOYMENT
CODE DONE
AUTO
UNIT TESTS
AUTO
INTEGRATE
AUTO
ACCEPTANCE TEST
AUTO
DEPLOY TO PRODUCTION

CONTINUOUS DELIVERY

- Is the practice of keeping your codebase deployable at any point. Beyond making sure your application passes automated tests it has to have all the configuration necessary to push it into production.
- Many teams then do push changes that pass the automated tests into a test or production environment immediately to ensure a fast development loop.
-

COLLABORATIVE DEVELOPMENT

- Major issue in this :
- In early they use pyramid kind of structure for –
- 1. Authority
- 2 Information
- That means all authority & all information is with project manager & people who follow project manager have little information & authority.

- Now trend is different- All information & authority is with whole team.

- 1. Self Organizing
- 2. Cross Functional Team
- 3. Global Software Development

- Challenges
  - Big Size of software : Large enterprise software need to develop.
  - Bigger the organization
  - Virtual Team

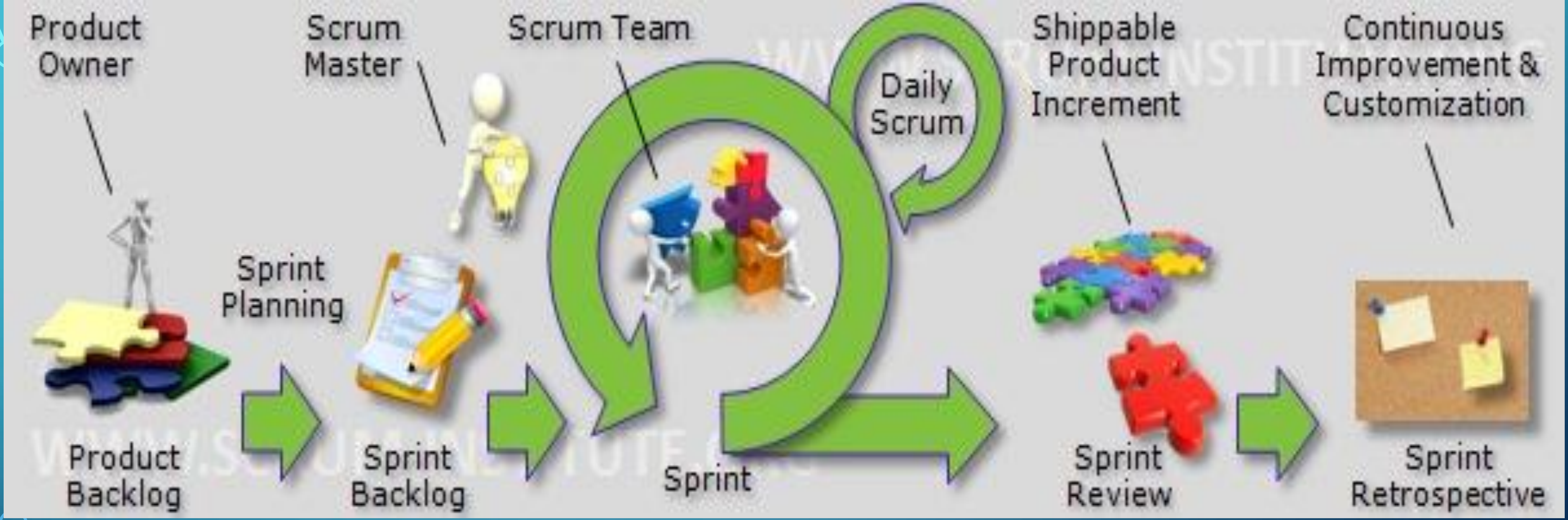
- Problems :
  - Time Difference
  - Communication
  - Cultural Differences
  - Trust

SCALING AGILE TO MANY TEAMS – ENTERPRIZE DEVELOPMENT

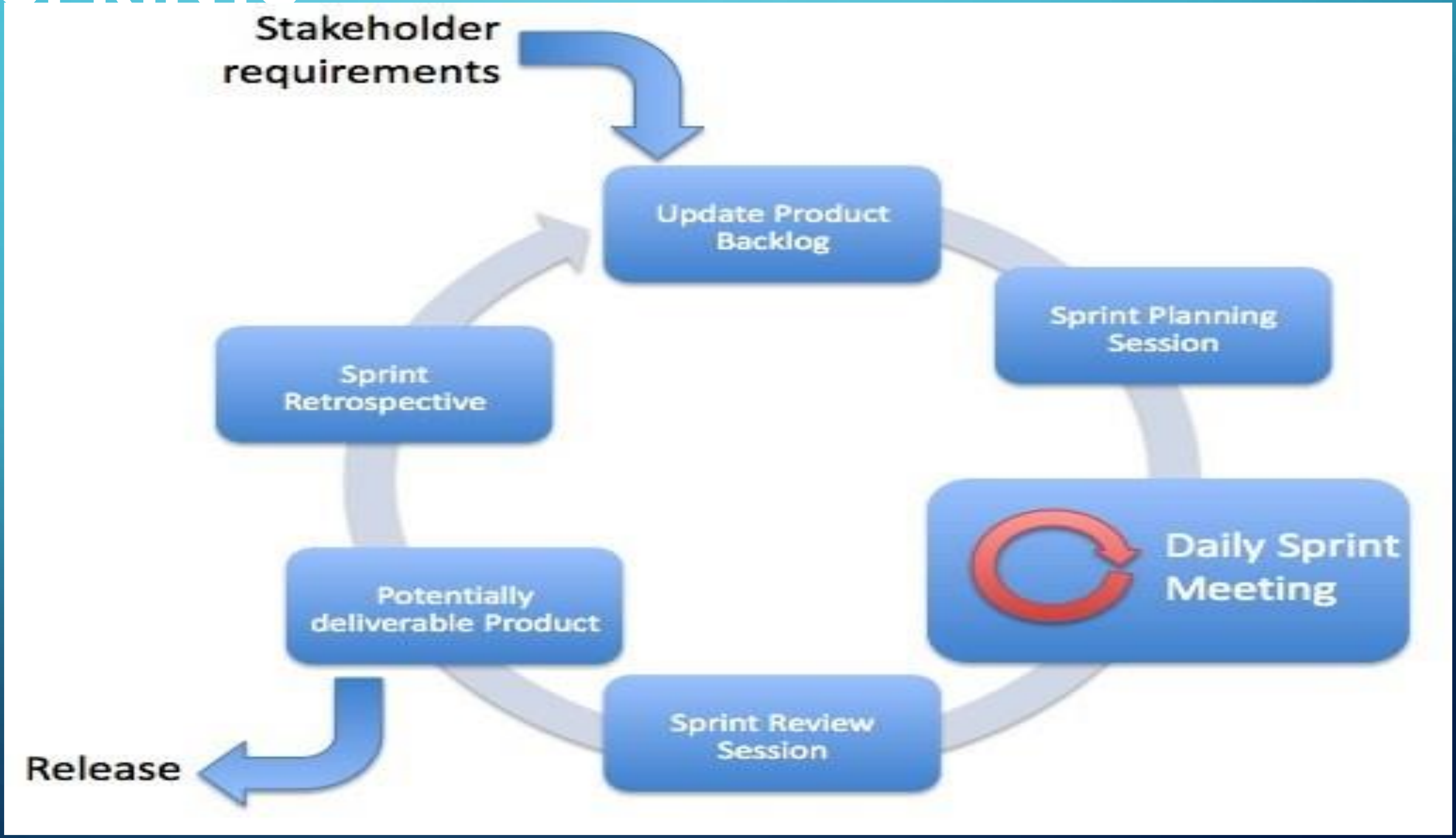
- 1. Scrum of Scrum :

- Challenges
  - Co-ordination in multiple team

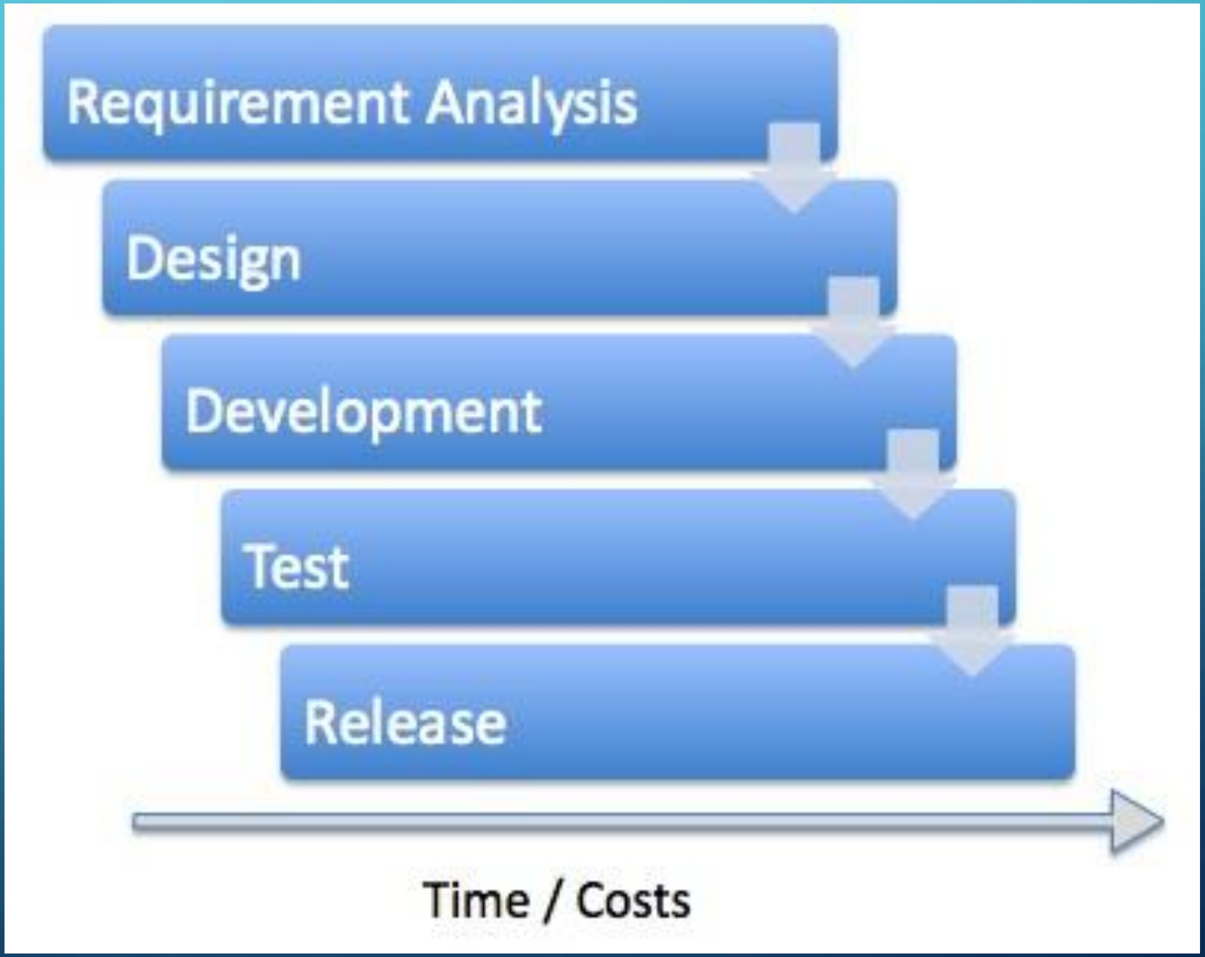
WHAT IS SCRUM?



A REAL WORLD EXAMPLE ACROSS VARIOUS SCRUM PHASES AND SPRINTS



PHASES IN THE CLASSICAL WATERFALL SOFTWARE DEVELOPMENT MODEL



Requirement Analysis

Design

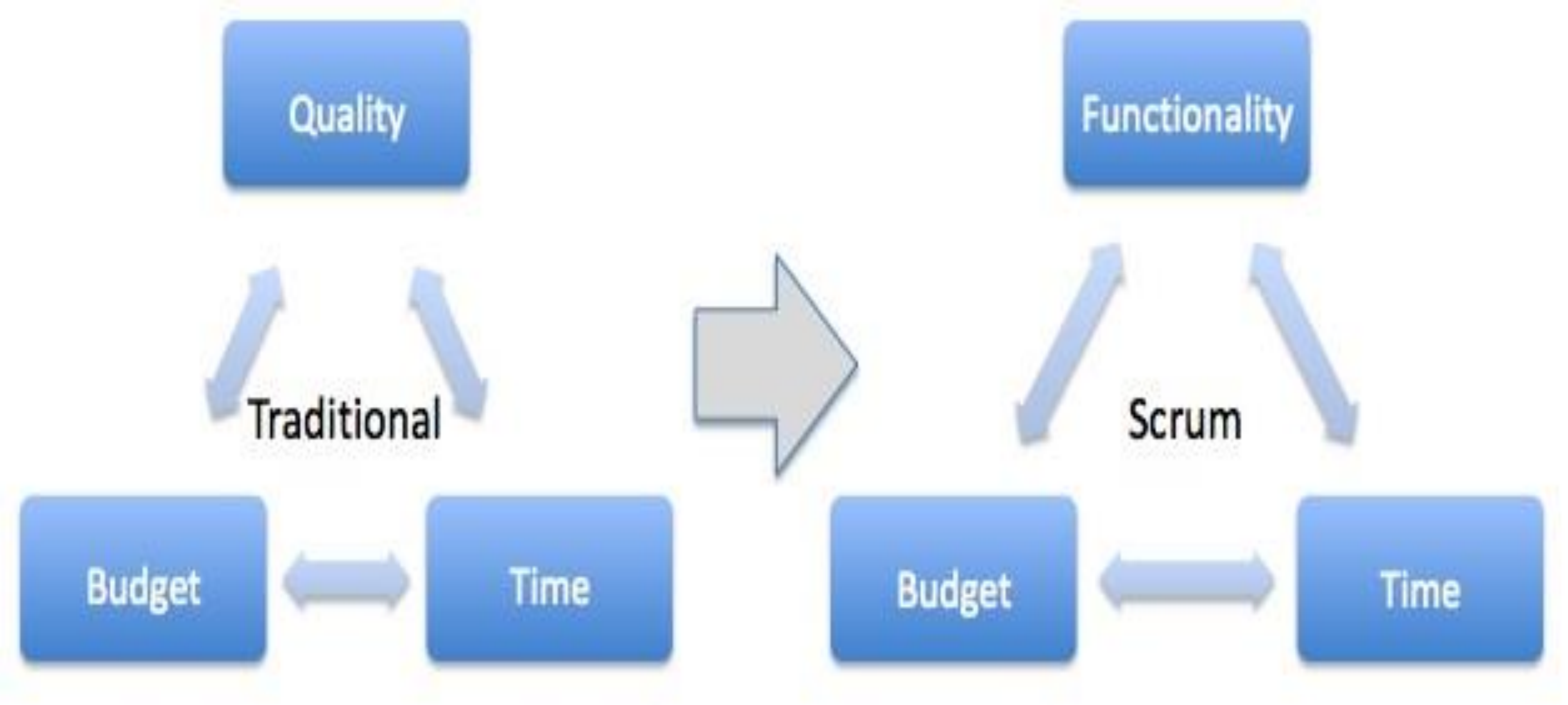
Development

Test

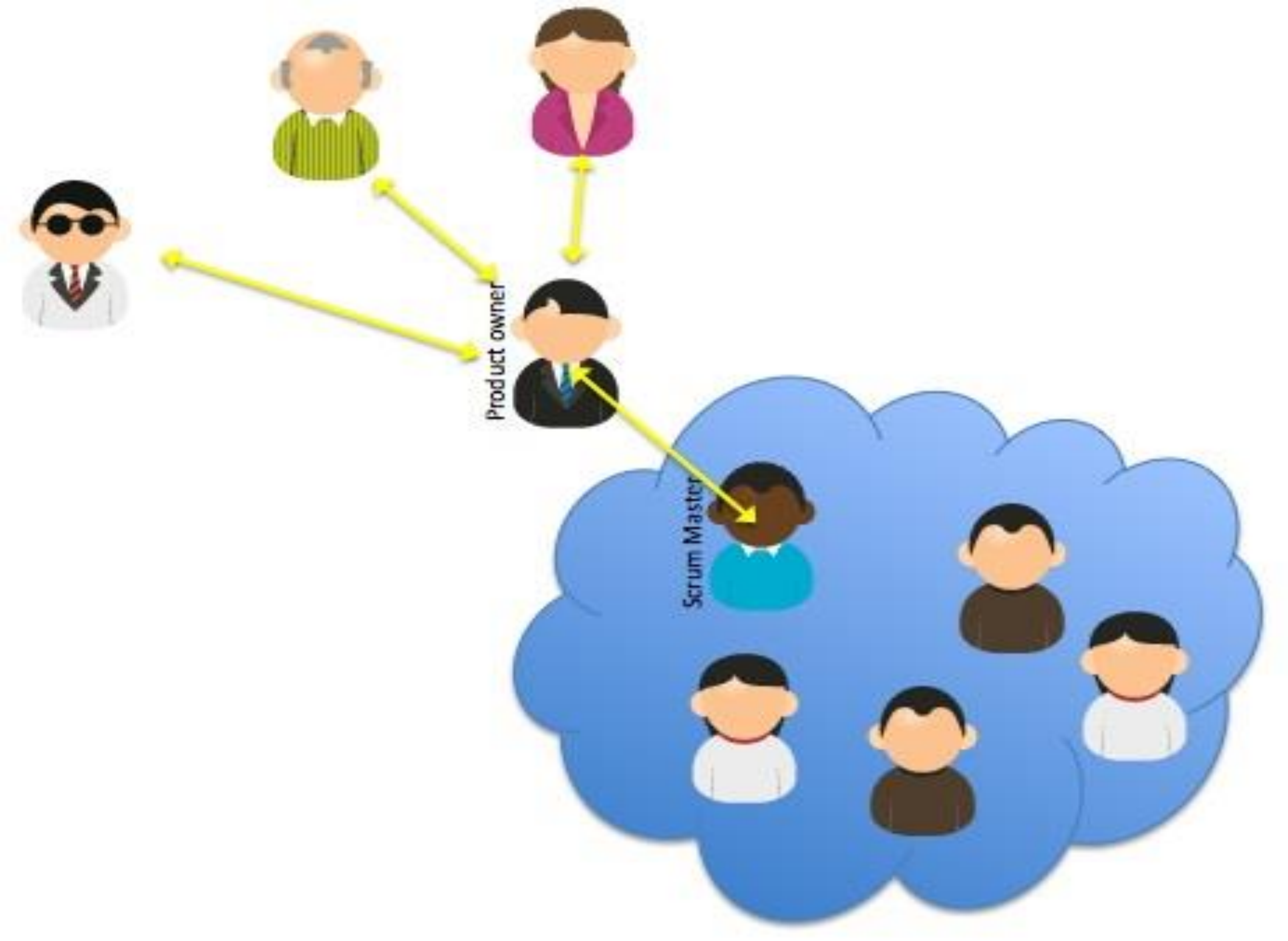
Release

Time / Costs

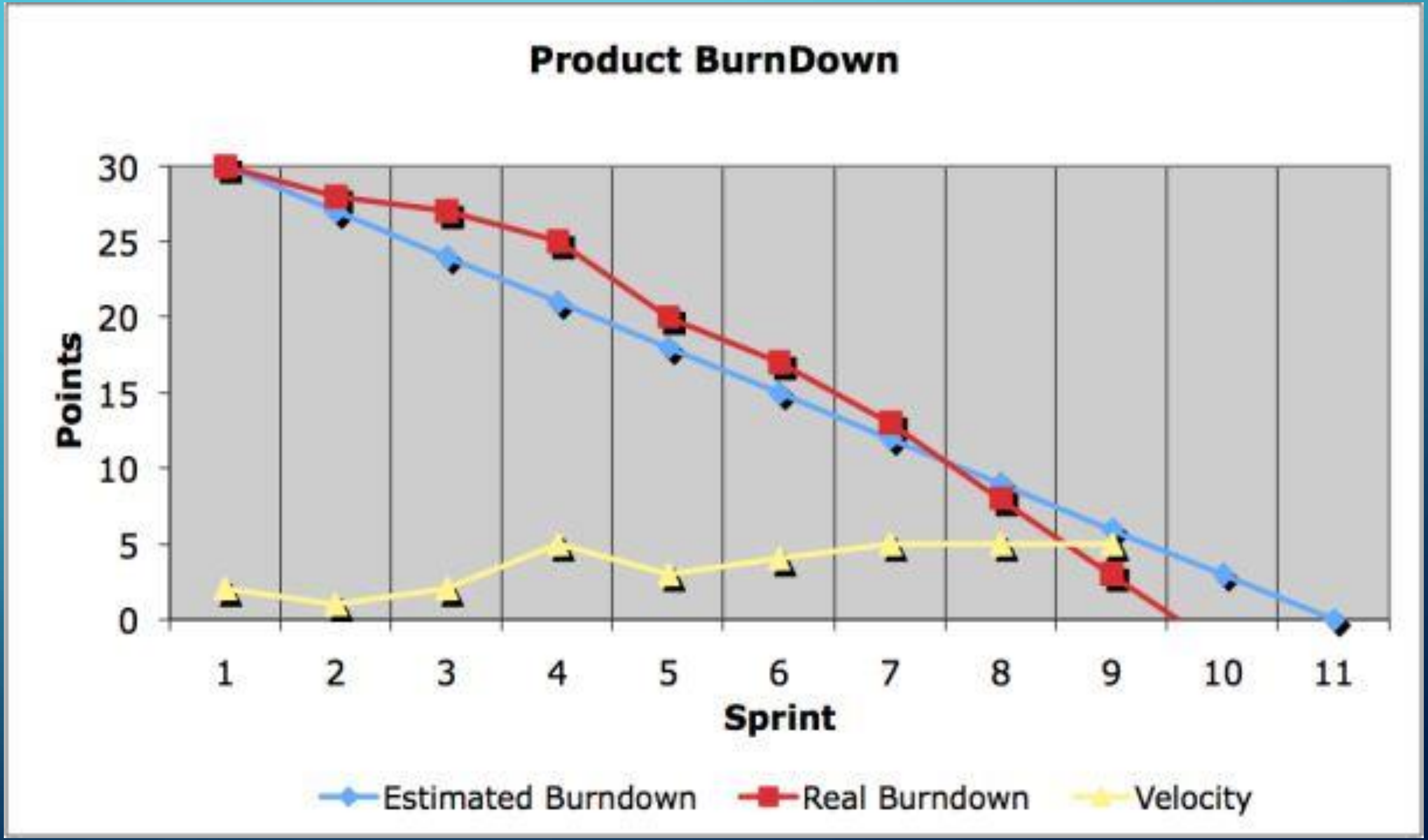
TRIANGLE OF PROJECT MANAGEMENT



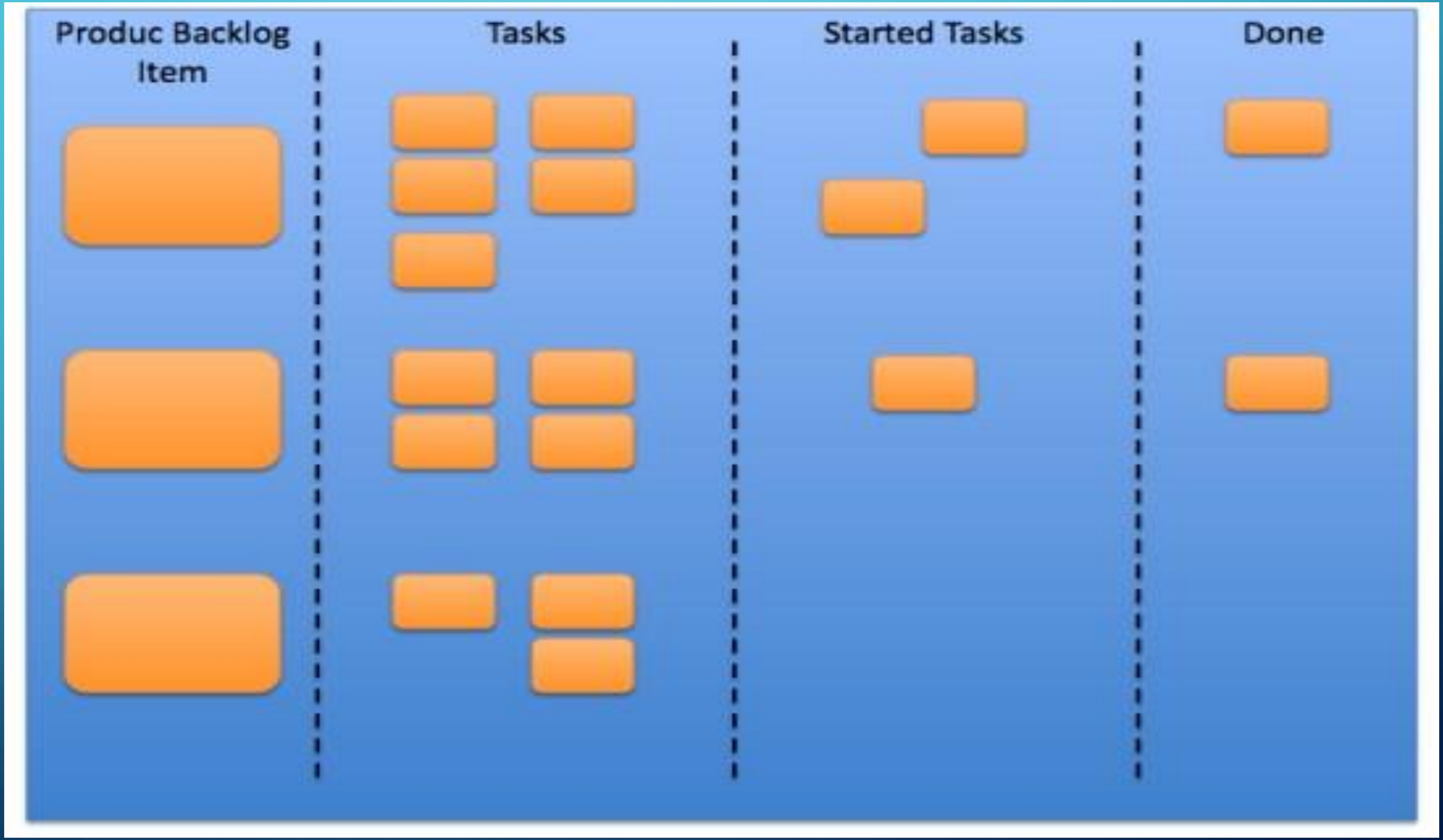
SCRUM ROLES & STAKEHOLDERS



SIMPLE BURNDOWN CHART



EXAMPLE SPRINT TASK BOARD



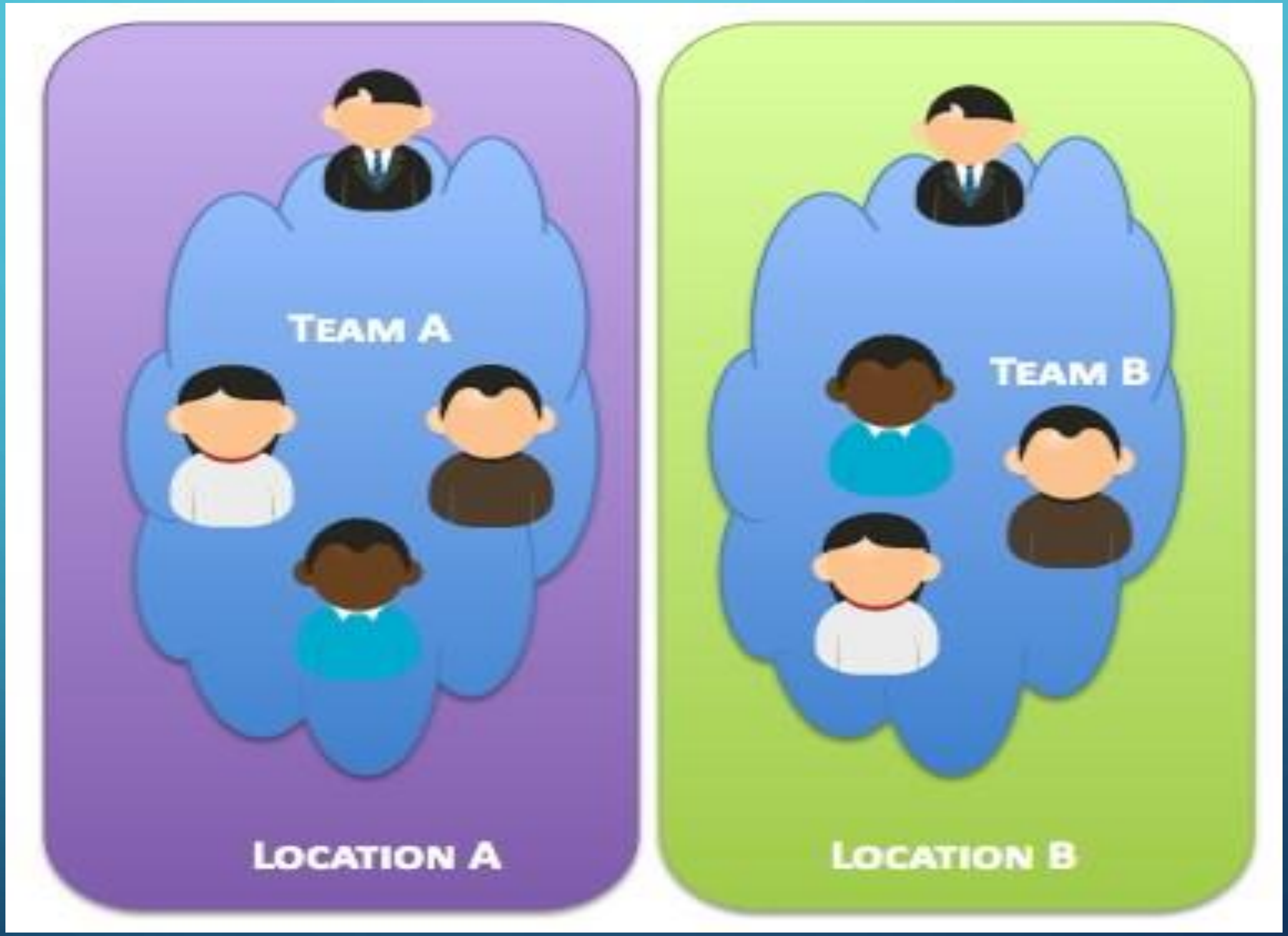
Produc Backlog Item

Tasks

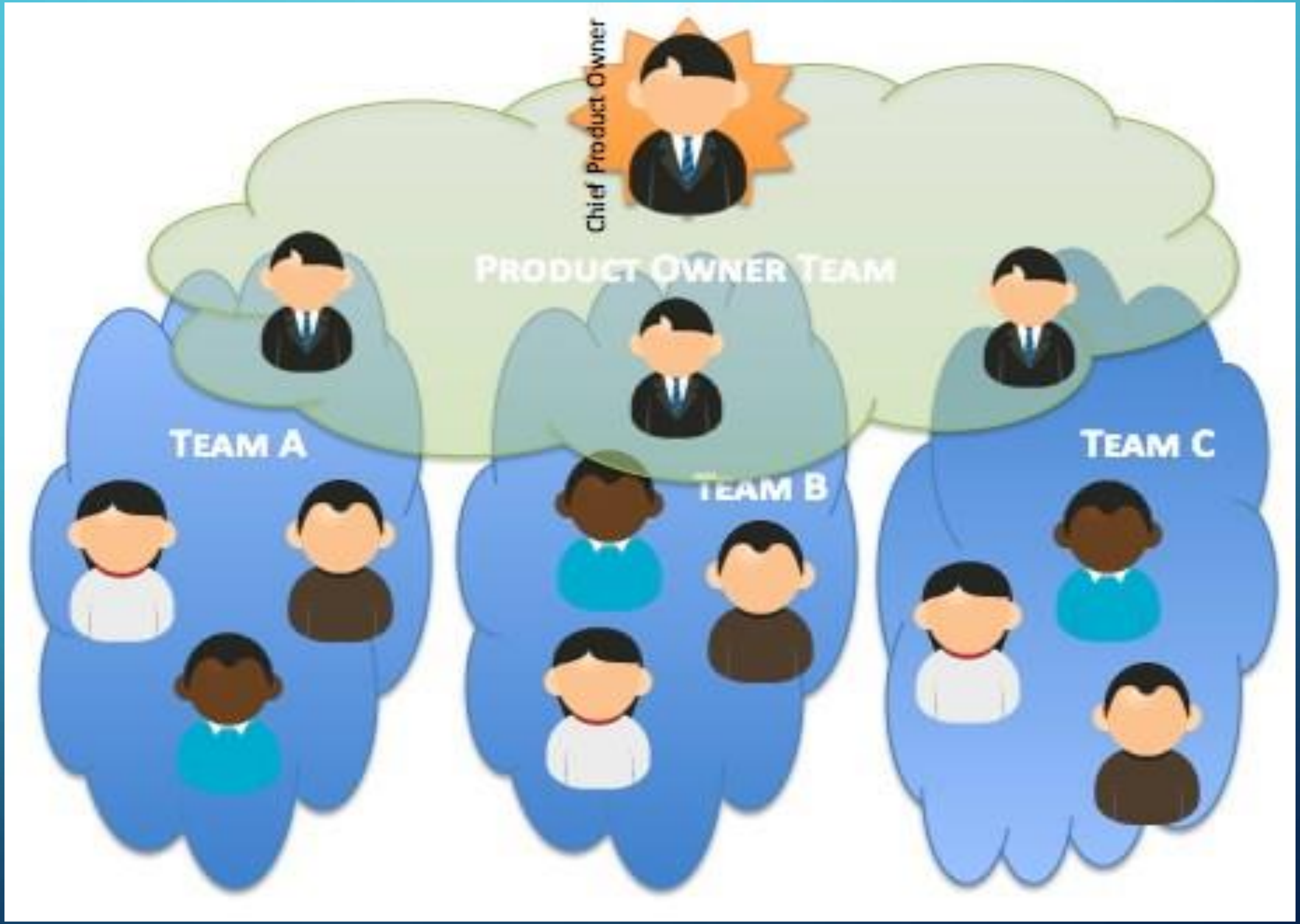
Started Tasks

Done

MULTIPLE TEAMS IN MULTIPLE LOCATIONS

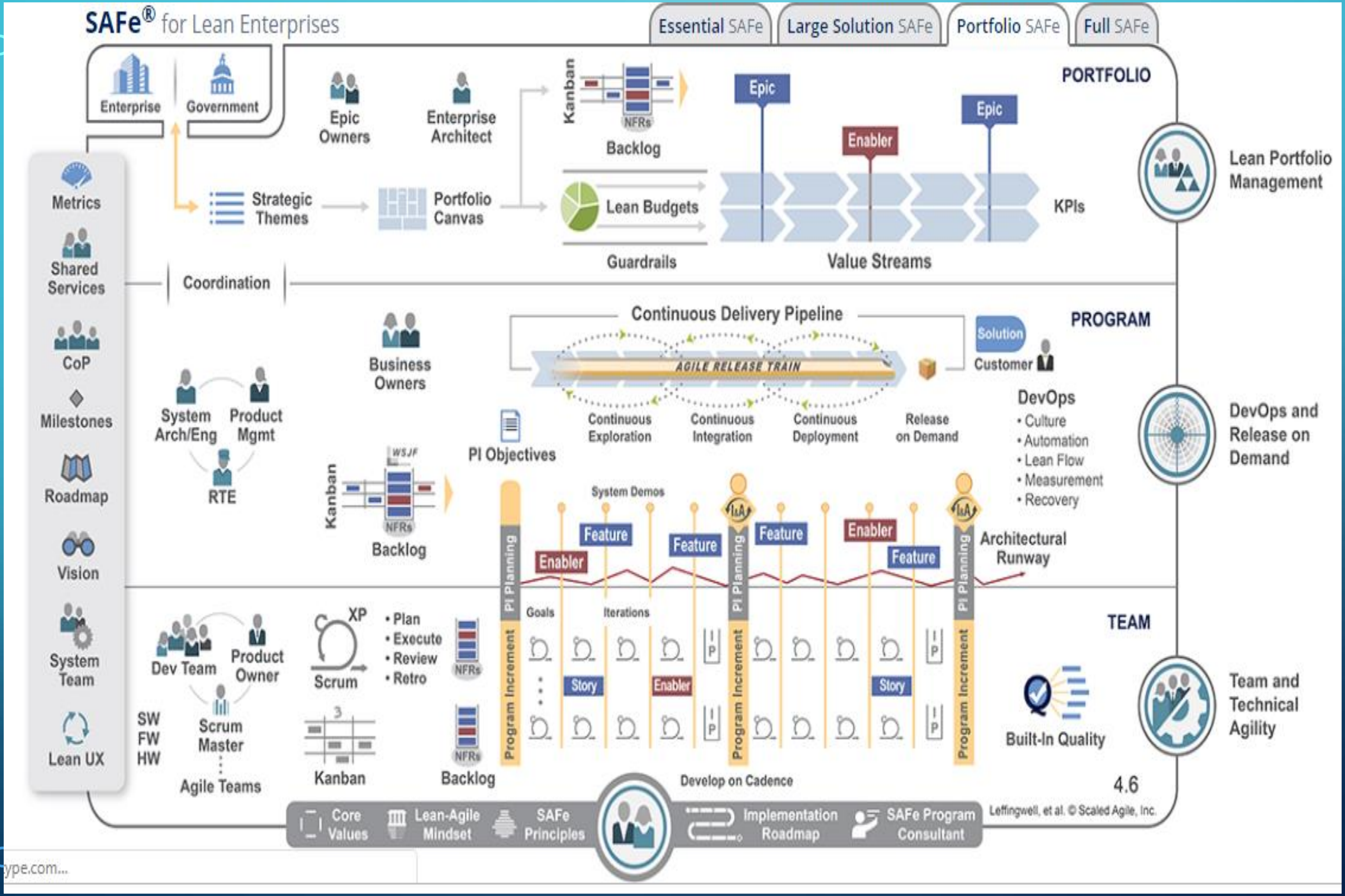


SCRUM PRODUCT OWNER TEAM



FRAMEWORKS FOR SCALING AGILE

- 1. SAFe : Scaled Agile Framework
- 2. Nexux
- 3. DAD : Disciplined Agiled Delivery
- 4. LeSS : Large Scaled Scrum


SAFe® for Lean Enterprises
Essential SAFe
Large Solution SAFe
Portfolio SAFe
Full SAFe
Enterprise
Government
Epic Owners
Enterprise Architect
Kanban
NFRs
Backlog
Epic
Enabler
Epic
PORTFOLIO
Strategic Themes
Portfolio Canvas
Lean Budgets
KPIs
Guardrails
Value Streams
Lean Portfolio Management
Metrics
Shared Services
Coordination
CoP
Milestones
Roadmap
Vision
System Team
Lean UX
Continuous Delivery Pipeline
Solution
Customer
PROGRAM
Business Owners
AGILE RELEASE TRAIN
Continuous Exploration
Continuous Integration
Continuous Deployment
Release on Demand
DevOps
• Culture
• Automation
• Lean Flow
• Measurement
• Recovery
DevOps and Release on Demand
System Arch/Eng
Product Mgmt
RTE
WSJF
Kanban
NFRs
Backlog
PI Objectives
System Demos
I&A
Feature
Enabler
Feature
Feature
Enabler
Feature
Architectural Runway
PI Planning
Program Increment
Goals
Iterations
TEAM
Dev Team
Product Owner
Scrum Master
SW
FW
HW
Agile Teams
XP
Scrum
• Plan
• Execute
• Review
• Retro
NFRs
Kanban
NFRs
Backlog
Story
Enabler
Story
IP
Built-In Quality
Team and Technical Agility
Develop on Cadence
4.6
Leffingwell, et al. © Scaled Agile, Inc.
Core Values
Lean-Agile Mindset
SAFe Principles
Implementation Roadmap
SAFe Program Consultant
ype.com...

- **Scaled Agile Framework (SAFe) empowers complex organizations to achieve the benefits of Lean-Agile software and systems development at scale.**
- SAFe is the world’s leading framework for scaling Agile across the enterprise. Used by hundreds of the world’s largest organizations, SAFe sustains and drives faster time-to-market, dramatic increases in productivity and quality, and improvement in employee engagement.
-

SAFe is designed to help businesses continuously and more efficiently

- An important key to success in supporting a Lean-Agile transformation is leadership engagement combined with education and training.

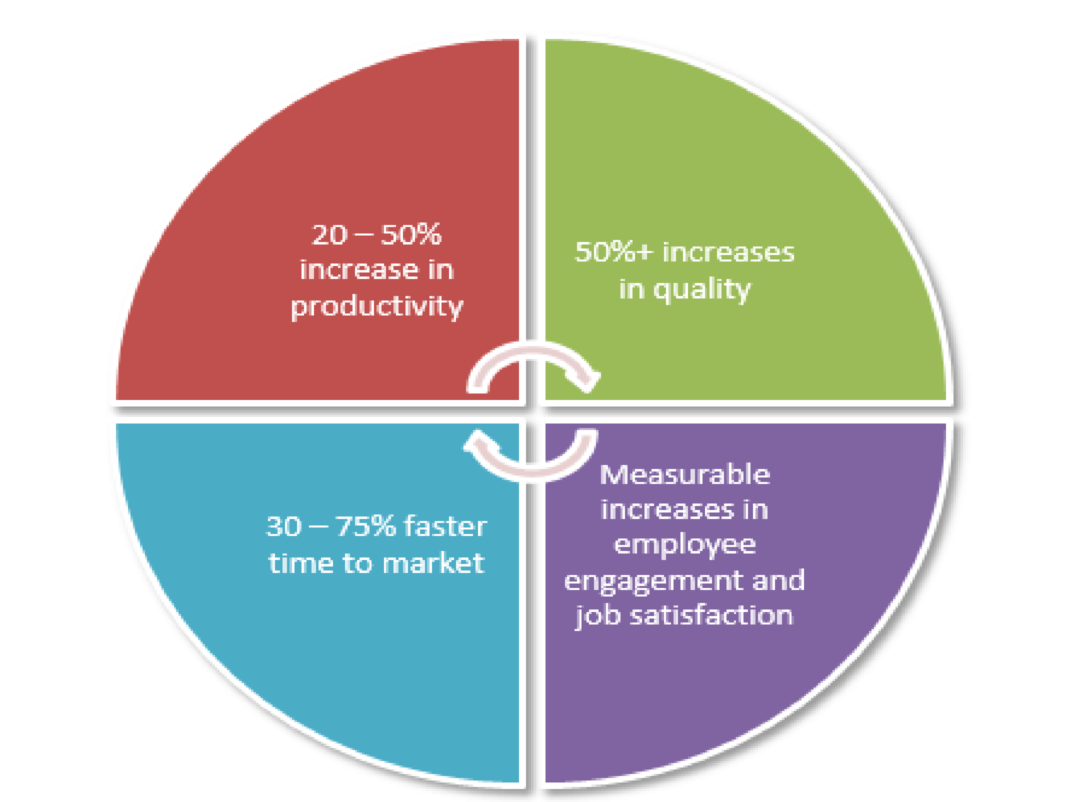
- Scaled Agile’s role-based curriculum helps enterprises unlock business results with SAFe.

- Scaled Agile Framework SAFe, is a freely available online knowledge base that allows you to apply lean-agile practices at the enterprise level. It provides a simple, lightweight experience for the software development team. The whole framework is divided into three segments
- **Team,**
- **Program and**
- **Portfolio.**

SAFe allows team for,

- Implementing Lean-Agile software and systems in enterprise level
- It's based on Lean and Agile principles.
- It gives detailed guidance for work at the enterprise Portfolio, Value Stream, Program, and Team.
- It's designed to meet the needs of all stakeholders within an organization.

WHY TO USE AGILE FRAMEWORK


20 – 50% increase in productivity
50%+ increases in quality
30 – 75% faster time to market
Measurable increases in employee engagement and job satisfaction

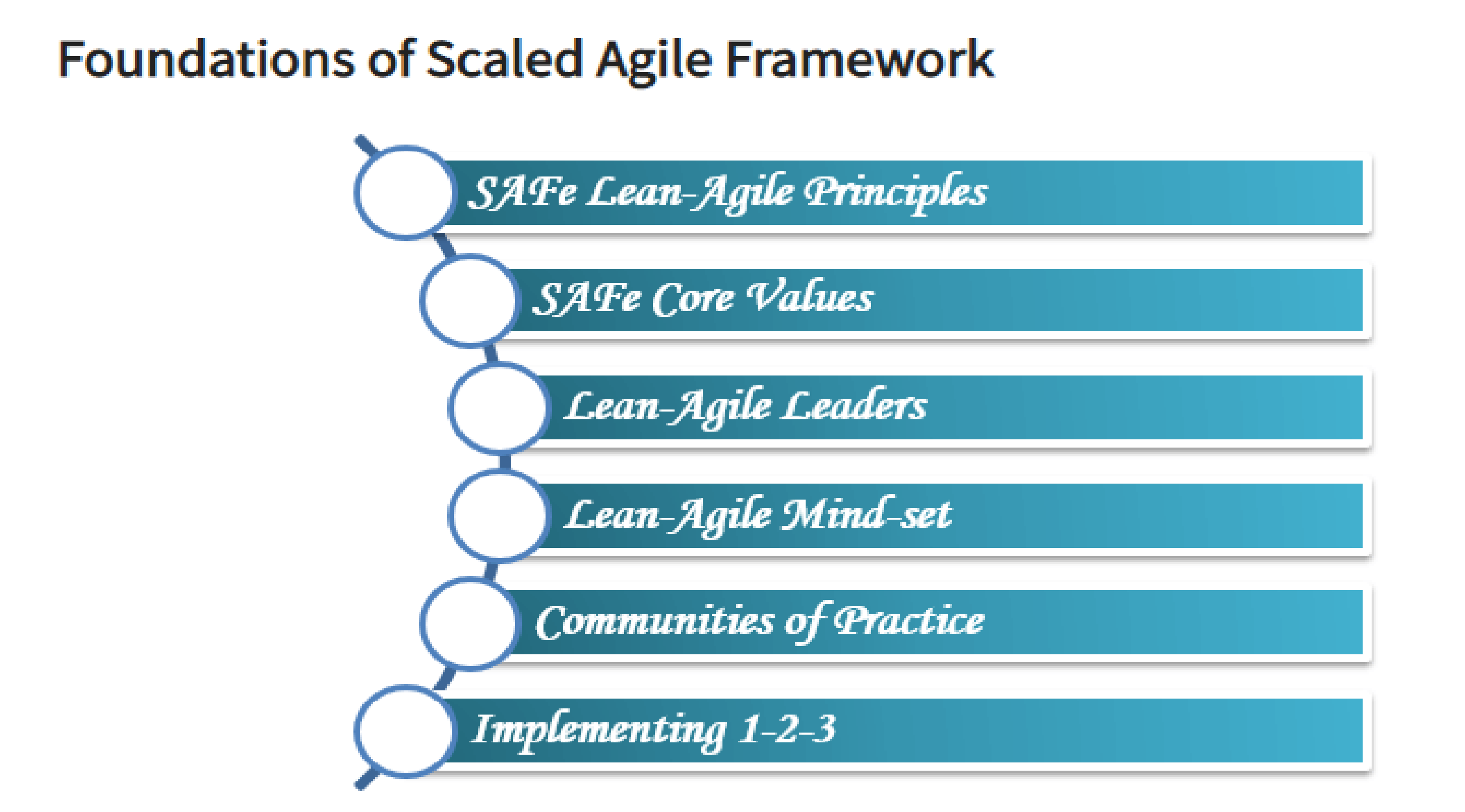
- It is simpler and lighter in weight, yet it expands to handle the needs of large value streams and complex system development. By implementing an Agile Framework, you will have following benefits,
- **Productivity increased** by **20 - 50%**
- **Quality** increased more than **50%**
- **Time to Market** is faster than **30 -75%**
- Increased **employee engagement** and **job satisfaction.**

WHEN TO USE SCALED AGILE FRAMEWORK

- When a team is interested to implement an agile approach consistently across larger, multi-team programs and portfolios.
- When multiple teams are running their own way of Agile implementation but regularly facing obstacles, delays, and failures.
- When teams want to work independently.
- When you want to scale Agile across the organization but not sure what new roles may be needed or what existing roles (i.e., management) need to change and how.
- When you have attempted to scale the Agile across your organization but struggling in alignment to achieve uniform or consistent strategy across business departments from portfolio to program and team levels.
- When an organization needs to improve its product development lead time and want to know how other companies have succeeded in scaling Agile with SAFe.
-

HOW DIFFERENT THAN OTHER AGILE PRACTICES

- It's publicly available and free to use.
- Available in a highly approachable and usable form.
- It's lightweight, practically proven results and specific to level.
- It constantly/regularly modifies/maintains most commonly used agile practices.
- Offers useful extensions to common agile practices.
- Grounds agile practices to an enterprise context.
- Offers complete picture of software development.
- Visibility or transparency is more on all the levels.
- Continues or regular feedback on quality and improvement.



Foundations of Scaled Agile Framework

- *SAFe Lean-Agile Principles*
- *SAFe Core Values*
- *Lean-Agile Leaders*
- *Lean-Agile Mind-set*
- *Communities of Practice*
- *Implementing 1-2-3*

SAFE LEAN-AGILE PRINCIPLES

- These basic principles and values for SAFe must be understood, exhibited and continued in order to get the desired results.
- Take an economic view
- Apply systems thinking
- Assume variability; preserve options
- Build incrementally with fast, integrated learning cycles
- Base milestones on an objective evaluation of working systems
- Visualize and limit WIP, reduce batch sizes and manage queue lengths
- Apply cadence, synchronize with cross-domain planning
- Unlock the intrinsic motivation of knowledge workers
- Decentralize decision-making