



VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

# SWE 2029 - Agile Development Process

## Module - VII

### ADVANCED CONCEPTS & CASE STUDIES

*Dr. Rajesh M*

*School of Computer Science and Engineering*

*Vellore Institute of Technology [VIT]*

*Chennai, India.*

Module	Topics	L Hrs
VII	<p><b>ADVANCED CONCEPTS &amp; CASE STUDIES:</b></p> <p>Scrum and Large Projects – Distributed Scrum – Agile Adoption - A case study of a scrum project, Scrum Success Stories</p>	5

# **Scrum and Large Projects**

# Scrum and Large Projects

- ❖ When we deal with large projects generally **involving four or more Scrum Teams**, some **additional processes** may be required to address the *additional coordination* and *synchronization efforts*.

**Some reasons additional processes would be needed for large projects are as follows:**

- ❖ *Increased interaction* and *dependencies among Scrum Teams*, as complexity increases for a large project
- ❖ *Need for collaboration* in a team of Product Owners
- ❖ *Need to manage conflicts, resolve issues*, and *set priorities* among all the Scrum Teams
- ❖ *Requirement for specialization* as some Scrum Teams may require specialized resources for specific tasks — and these particular skill sets are not needed on all Scrum Teams.

# Scrum and Large Projects

- ❖ Necessity to ***define certain guidelines and standards*** that should be adhered to by all Scrum Teams (e.g., security standards within a company or legal and governmental guidelines for specific industries). These may need to be defined by the Scrum Guidance Body.
- ❖ ***Requirement to set up an environment***, or working area, for the large project, which would then be used by all Scrum Teams
- ❖ ***Need for coordinating the outputs from several Scrum Teams*** to create a project release for a large project.
- ❖ The definition of a large project may ***depend on the company*** and the ***complexity of projects undertaken***. The key criterion for a project being large versus small is to have multiple Scrum Masters and/or Product Owners.

# **Scrum and Large Projects**

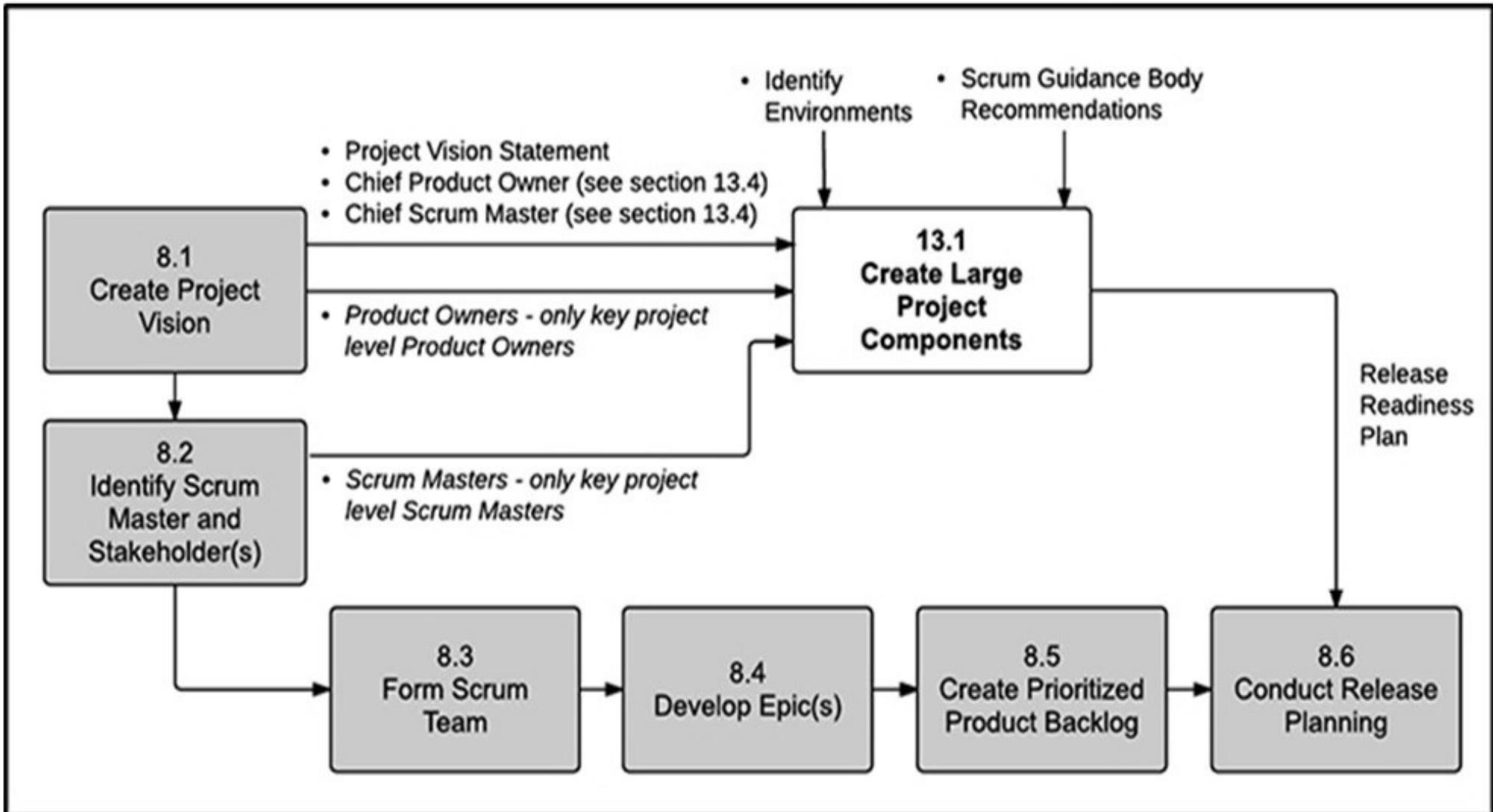
## **Scaling Scrum for Large Projects**

# Scrum and Large Projects

**Scaling Scrum for Large Projects** consists of following **processes**:

- ❖ **1. Create Large Project Components**—This process defines **how the multiple Product Owners work together and how the multiple Scrum Teams work together**. Also common components and common and specialized resources are identified.
- ❖ Following figure shows all the **relationship of the Create Large Project Components process** to the fundamental Scrum processes in the Initiate phase.

# Scrum and Large Projects

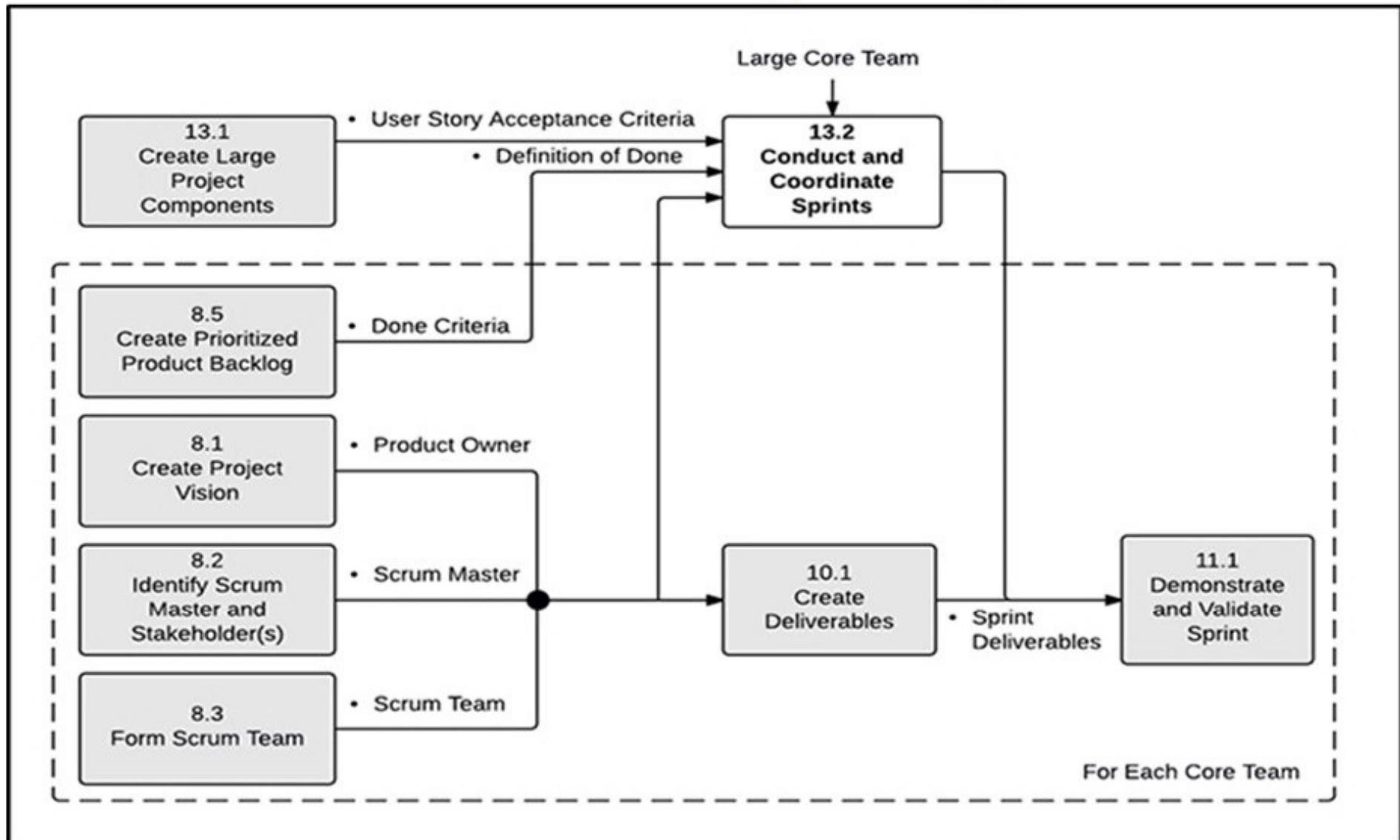


# Scrum and Large Projects

**Scaling Scrum for Large Projects** consists of following **processes**:

- ❖ **2. Conduct and Coordinate Sprints**—This process is usually only relevant for large projects and addresses specific aspects that should be considered during each Sprint. If required, **Scrum of Scrums Meetings are conducted** to coordinate efforts between multiple Scrum Teams.
- ❖ Following figure shows all the relationship of the Conduct and Coordinate Sprints process to the fundamental Scrum processes.

# Scrum and Large Projects

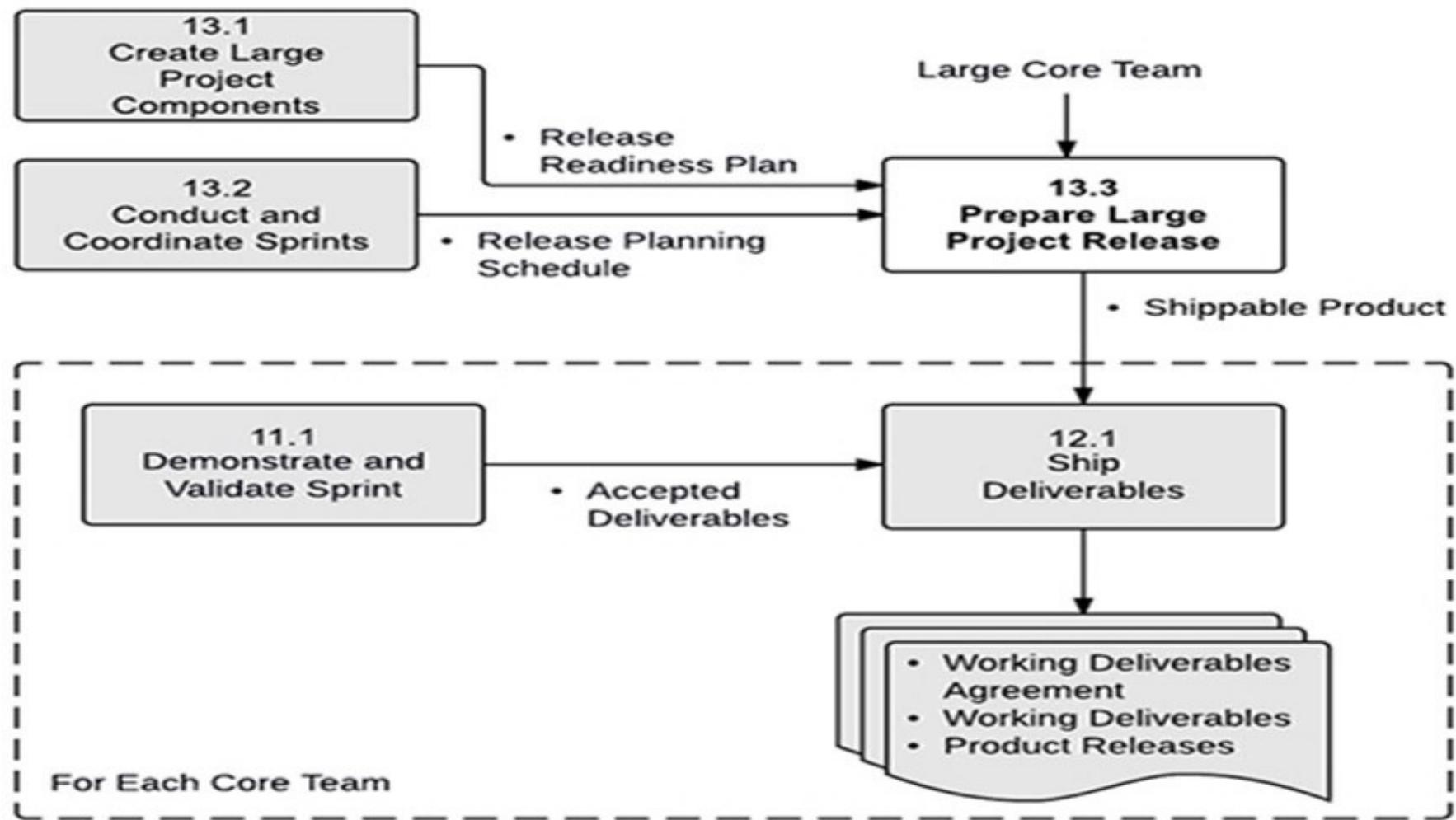


# Scrum and Large Projects

**Scaling Scrum for Large Projects** consists of following processes:

- ❖ **3. Prepare Large Project Release**—In some large projects it may make business sense to do a **Sprint prior to a release** in order to **prepare for releasing the product** (to be decided by the project team based on business needs). This process addresses such a preparation Sprint.
- ❖ Following figure shows the relationships of the Prepare Large Project Release process to the fundamental Scrum processes.

# Scrum and Large Projects



# **Scrum and Large Projects - Approaches**

- **Large-Scale Scrum(LeSS)**
- **Scrum of Scrum(SoS)**
- **SAFe**

# Scrum and Large Projects

- ❖ There is a **lot of confusion** and some fairly polarized opinions about **scaling Agile and Scrum for large, complex projects involving multiple teams**.
- ❖ Some people think that it can be done simply by adding a *Scrum-of-Scrums* approach to **provide a mechanism to coordinate the efforts of multiple teams**.
- ❖ A more comprehensive approach for **integrating the efforts of multiple development teams** is *Large-Scale Scrum (LeSS)*.
- ❖ A *Scrum-of-Scrums* approach is a *loosely-coupled approach* that **only provides for basic coordination of the work between teams** – each team still operates fairly independently.
- ❖ *LeSS* is a much more *tightly-coupled approach* that goes **beyond the very basic level of coordination of work** that the Scrum-of-Scrums approach provides.

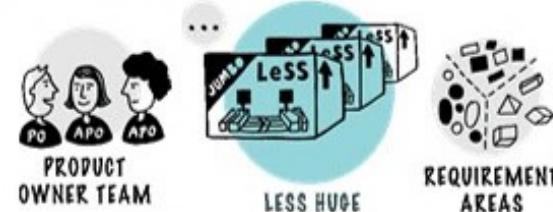
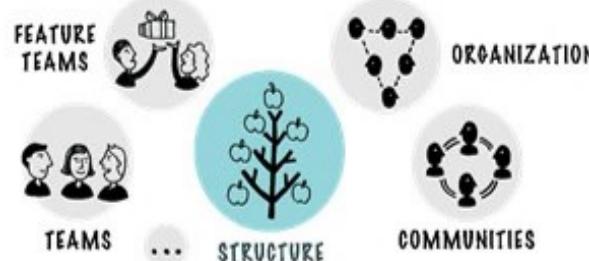
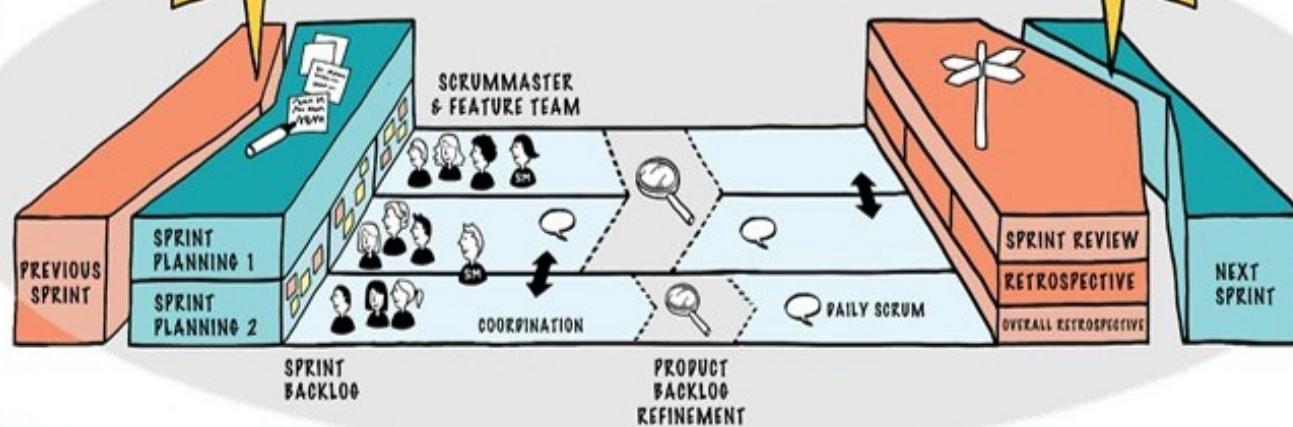
# Scrum and Large Projects

## LeSS:

- ❖ LeSS is a **lightweight (agile) framework** for scaling Scrum to more than one team.
- ❖ It was extracted out of the experiences of Bas Vodde and Craig Larman while Scaling Agile development in many different types of companies, products and industries over the last ten years.
- ❖ There are several [case studies](<https://less.works/case-studies/index.html>) available and an upcoming book describing [LeSS in detail](<http://www.amazon.com/Large-Scale-Scrum-More-Craig-Larman/dp/0321985710>).

# Scrum and Large Projects

LeSS BOOK CHAPTER 2:  
INTRODUCTION



CC BY-NC-ND

# Scrum and Large Projects

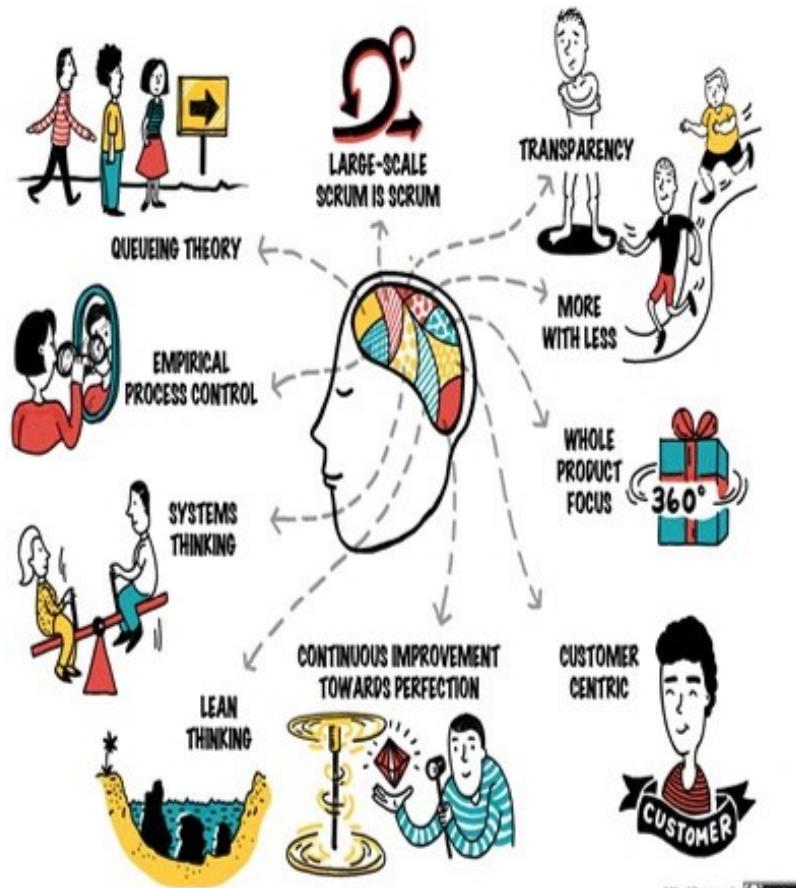
- ❖ LeSS consists of the **LeSS Principles**, the **Framework**, the **Guides** and a **set of experiments**.
- ❖ The LeSS framework is divided into two frameworks: ***basic LeSS for 2-8 teams and LeSS Huge for 8+ teams***. All of these are also available on the [less.works website](<http://less.works>).

# Scrum and Large Projects

- ❖ LeSS is different with other scaling frameworks in the sense that it provides a very **minimalistic framework** that enables empiricism on a large-scale which *enables the teams and organization to inspect-adapt their implementation based on their experiences and context.*
- ❖ LeSS is based on the idea that *providing too much rules, roles, artifacts* and asking the organization to tailor it down is a fundamentally flawed approach and instead scaling frameworks should be minimalistic and allowing organizations to fill them in.

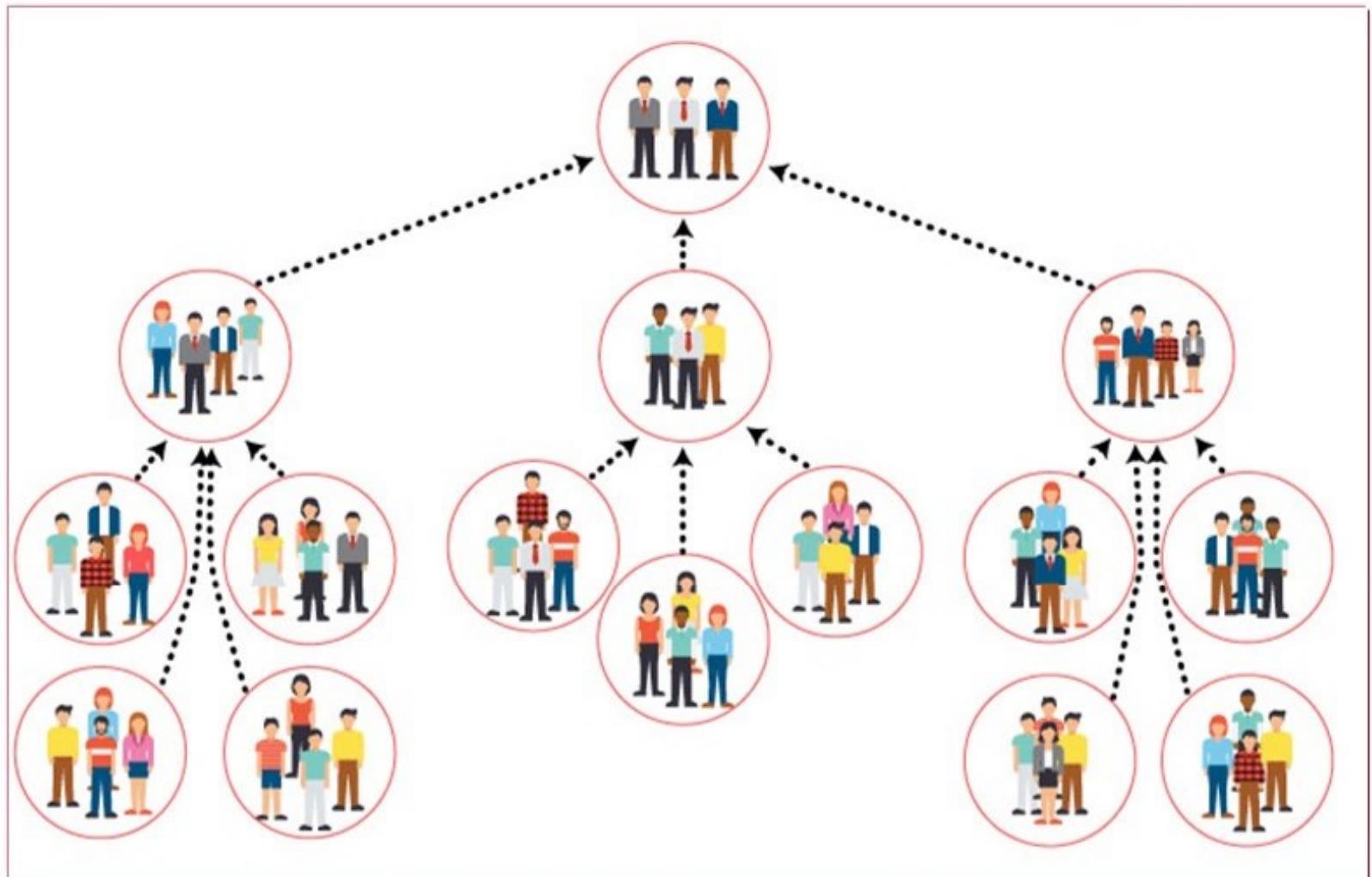
# LeSS - Principles

- ❖ LeSS defines 10 principles for applying the value, elements, and overall purpose of scrum across an enterprise. They help create more responsible teams with greater customer focus and collaboration. Here's the complete list:
- ❖ Large-Scale Scrum is scrum
- ❖ Empirical process control
- ❖ Transparency
- ❖ More with less
- ❖ Whole product focus
- ❖ Customer-centric
- ❖ Continuous improvement towards perfection
- ❖ Systems thinking
- ❖ Lean thinking
- ❖ Queuing theory



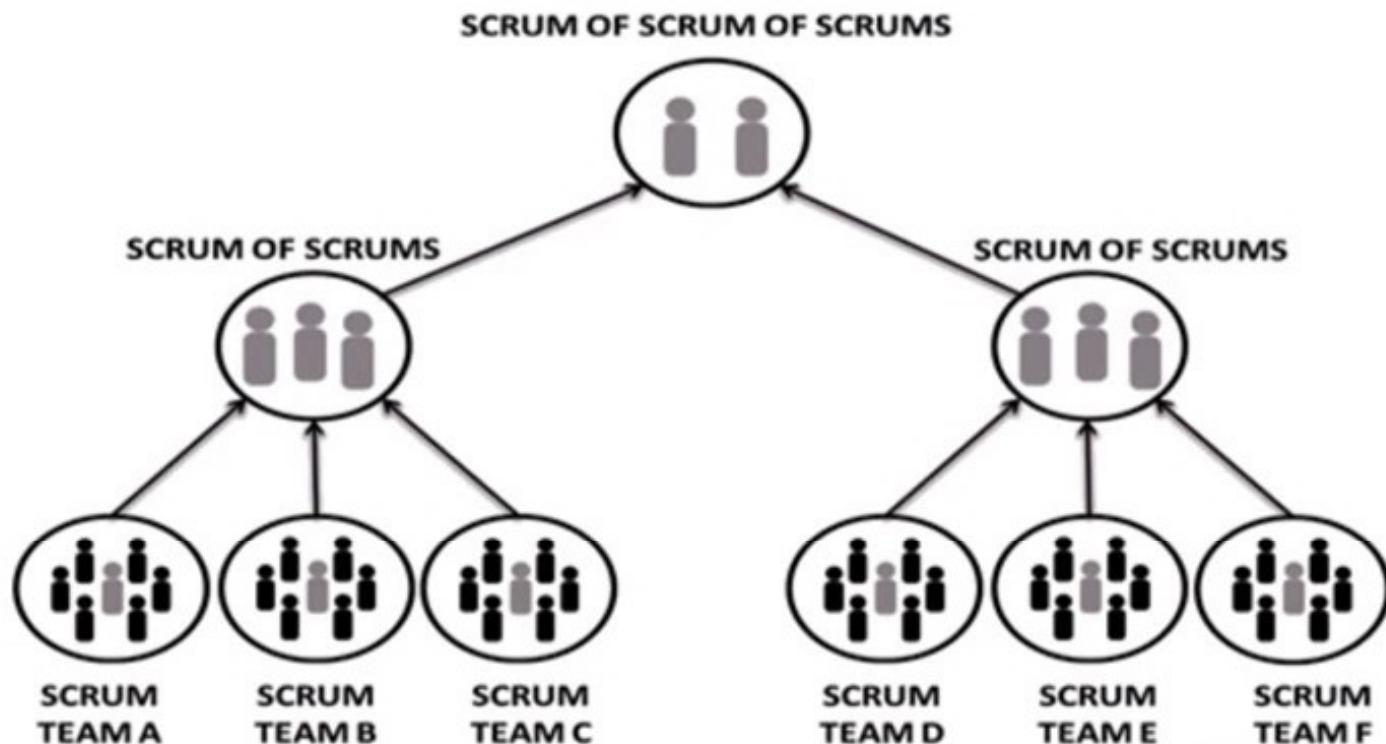
# Scrum and Large Projects

## Scrum of Scrums:



# Scrum and Large Projects

## Scrum of Scrums:



# Scrum and Large Projects

## Scrum of Scrum:

- ❖ A technique to scale Scrum up to *large groups (over a dozen people)*, consisting of dividing the groups into **Agile teams** of 5-10.
- ❖ Each daily scrum within a sub-team ends by designating one member as "**ambassador**" to participate in a daily meeting with ambassadors from other teams, called the **Scrum of Scrums**.
- ❖ Depending on the context, ambassadors may be **technical contributors**, or each team's **Scrum Master**, or even **managers of each team**.

# Scrum and Large Projects

## Scrum of Scrum:

- ❖ The Scrum of Scrum meeting is run very similarly to the Daily Scrum meeting each team holds on a daily basis but is not limited to a fifteen-minute timebox. At the Scrum of Scrums meeting, each team "ambassador" should respond to the following questions:
  - ❖ What has your team accomplished since our last meeting?
  - ❖ What problems occurred, if any, that negatively affected your team?
  - ❖ What does your team want to accomplish before we meet again?
  - ❖ What output from your team in future sprints, do you see as possibly interfering with the work of other teams?
  - ❖ Does your team see any interference problems coming from the work of other teams?

# Scrum and Large Projects

## Scrum of Scrum:

### outcome of a Scrum of Scrums:

- ❖ Scrum of Scrum meeting gives the clear picture and the path the team is moving towards the end goal, the team can identify the clear plan for the day.
- ❖ The team can identify the possible impediments that destruct in achieving sprint goal.
- ❖ Once the meeting is done, the team can work on the impediments in more detail and find solutions and end results in better productivity and a healthy environment throughout the project.

# Scrum and Large Projects

SAFe : “Scalable Agile Framework.”

# Scrum and Large Projects

## SAFe :

- ❖ SAFe stands for “**Scalable Agile Framework.**” It defines an approach for scaling Scrum to make it work for bigger enterprises that has much bigger teams working on the same product than what Scrum recommends.
- ❖ SAFe is an embellished acronym that stands for Scaled Agile Framework. The **embellishment is the lower-case “e” at the end that turns SAF into SAFe.**
- ❖ The additional “e” making SAFe “safe” is an excellent marketing consideration. SAF + “e” is safe to do. SAFe adds a **level of safety by lowering risk.**

# Scrum and Large Projects

## SAFe :

- ❖ 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 deliver value on a regular and predictable schedule.
- ❖ It provides a knowledge base of proven, integrated principles and practices to support enterprise agility.

# Scrum and Large Projects

## SAFe :

- ❖ Principles of SAFe
  
- ❖ Assume variability; preserve options.
- ❖ Build incrementally with fast integrated learning cycles.
- ❖ Base milestones on objective evaluation of working systems.
- ❖ Visualize and limit work-in-progress, reduce batch sizes, and manage queue lengths.

# Scrum and Large Projects

## SAFe :

### ❖ 3-Level of SAFe

The 3-Level SAFe is implemented at the following levels:

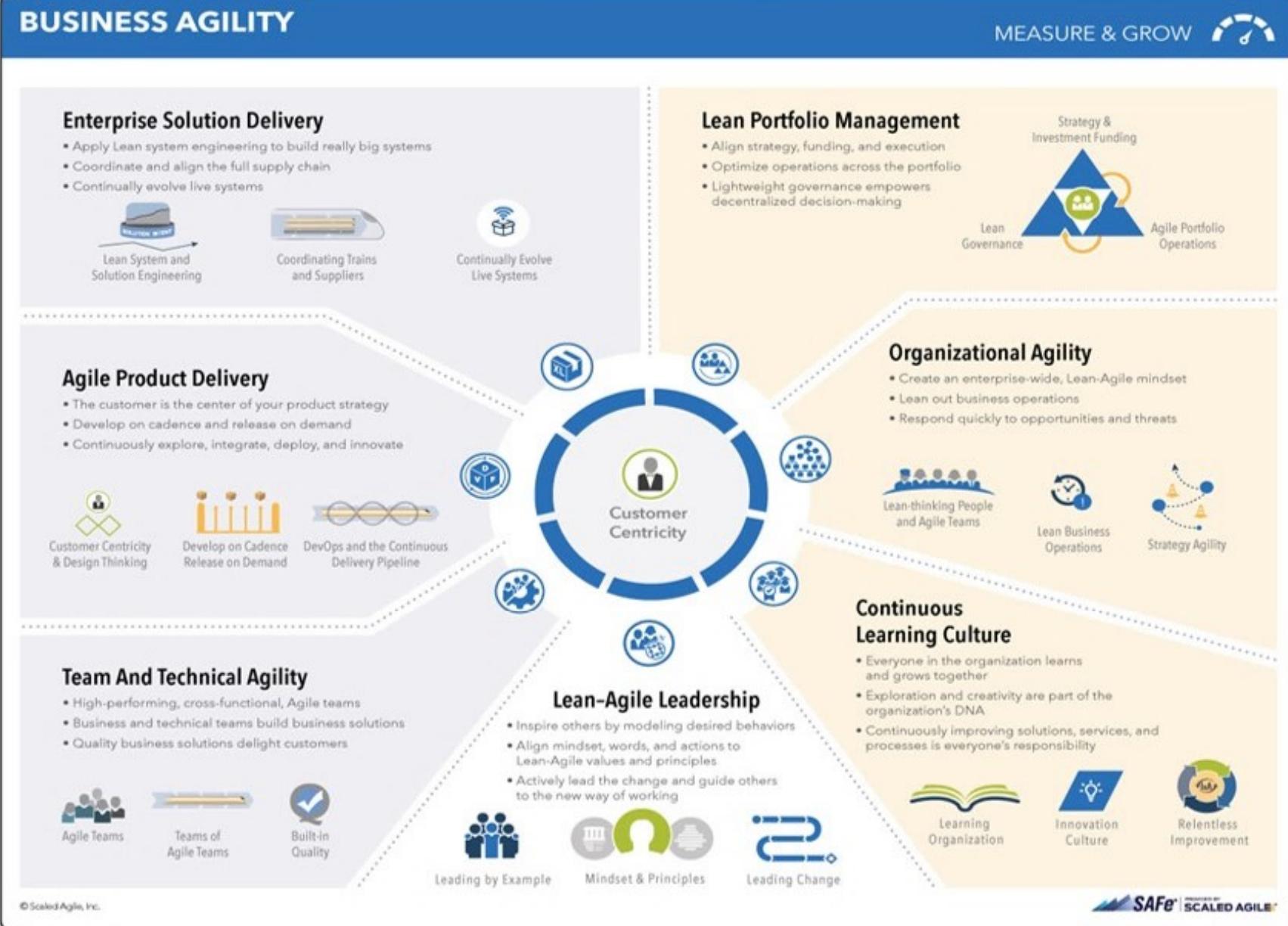
- ❖ team,
- ❖ program and
- ❖ portfolio.

## SAFe Portfolio:

Lean Portfolio Management describes how a SAFe portfolio is a collection of Value Streams for a specific business domain in an Enterprise. Each value stream delivers one or more Solutions that help the enterprise meet its business strategy.

# SAFe 5.0:

# Scrum and Large Projects



# Scrum and Large Projects

	Scrum-of-Scrums	LeSS
Coordination of Work	Formal Scrum-of Scrum's Meeting	Informal, "Just Talk"
Product Backlog Management (Single or Multiple Backlogs)	Not Specified	Single Product Backlog
Sprint Planning (Separate or Joint)	Not Specified	Joint
Sprint Review (Separate or Joint)	Not Specified	Joint
Allocation of Work (Component or Feature)	Not Specified	Feature

# Pitfalls of Agility on Large Projects

# **10 Pitfalls of Agility on Large Projects**

- ❖ **Pitfall #1: Planning a month or less ahead is not enough.**
  - ❖ Use rolling wave planning to create an evolving big picture.
- ❖ **Pitfall #2: Effective small teams need coordination to make an effective large organization.**
  - ❖ Combine bottom-up (scrum-style) and top-down (traditional) planning.
- ❖ **Pitfall #3: We can't afford to trust everyone on larger teams.**
  - ❖ Turn up the knob on transparency (especially time and quality data).

# **10 Pitfalls of Agility on Large Projects**

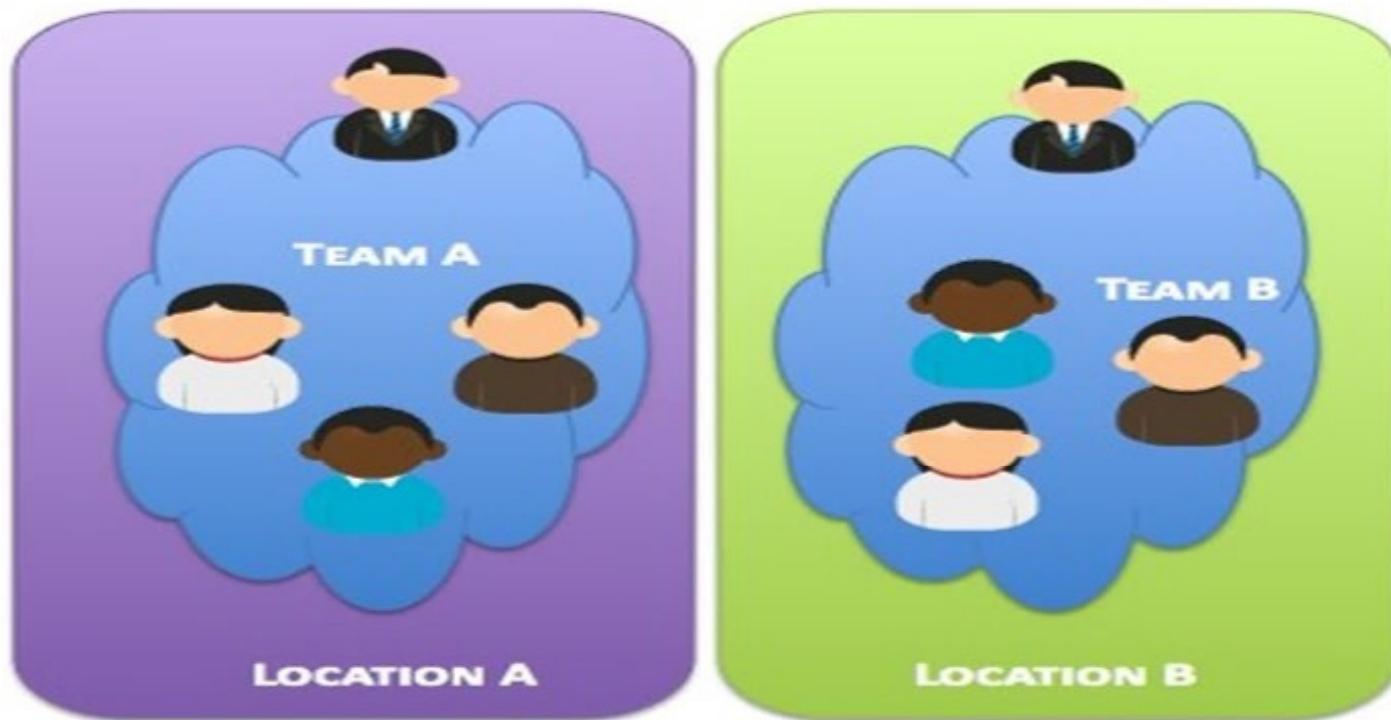
- ❖ **Pitfall #4: The customer doesn't want a release every month.**
  - ❖ Release early and often internally, with longer cycles for expanded audiences.
- ❖ **Pitfall #5: Hundreds of people can't check directly into "main" every day.**
  - ❖ Separate dependent sub-projects and use incremental integration with branches.
- ❖ **Pitfall #6: Not all activities are best handled by generalists.**
  - ❖ Apply lean techniques to more effectively handle specialization.
- ❖ **Pitfall #7: Our team/management expects to plan, and execution to plan.**
  - ❖ Making firm commitments to something we don't yet understand is counter-productive. As they come in, actuals have to trump estimates.

# 10 Pitfalls of Agility on Large Projects

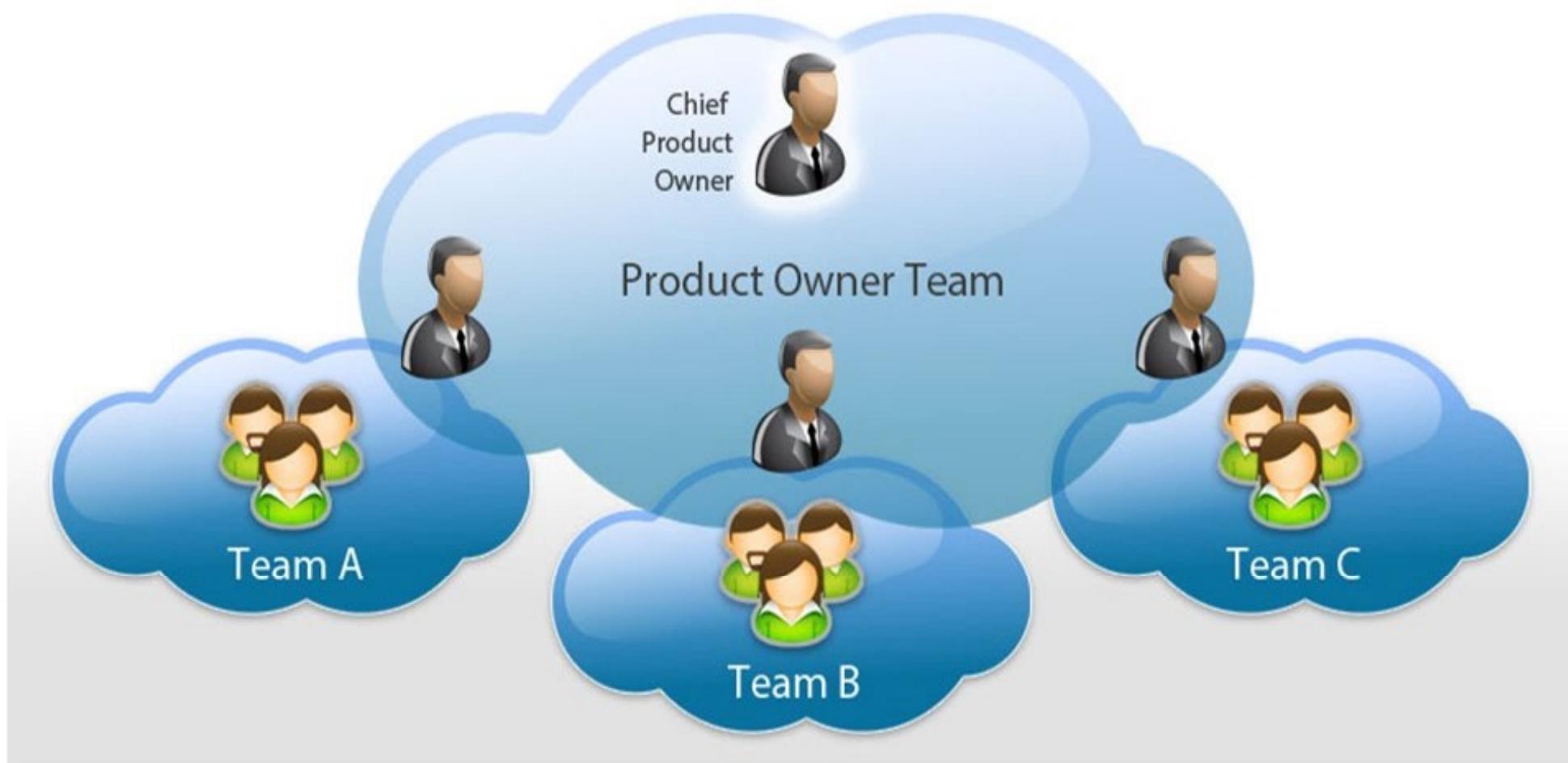
- ❖ **Pitfall #8: We are already in the dark. We need more documentation, not less.**
  - ❖ On large projects, there are usually reams of wasted documentation. But it may be that "just enough" documentation and status-taking is still a lot.
- ❖ **Pitfall #9: Large teams will reject big changes in how we work.**
  - ❖ Start with the way the team works today. Reflect and adapt towards agility.
- ❖ **Pitfall #10: Being agile on a large project is unrealistic and impossible to sustain.**
  - ❖ There is no surer strategy for large-scale failure than large projects without empowered teams, short cycles, strong feedback, and a culture which embraces change and adaptation. All we can do is have the patience, persistence, and thoughtfulness to always keep driving in the right direction.

# **Distributed Scrum**

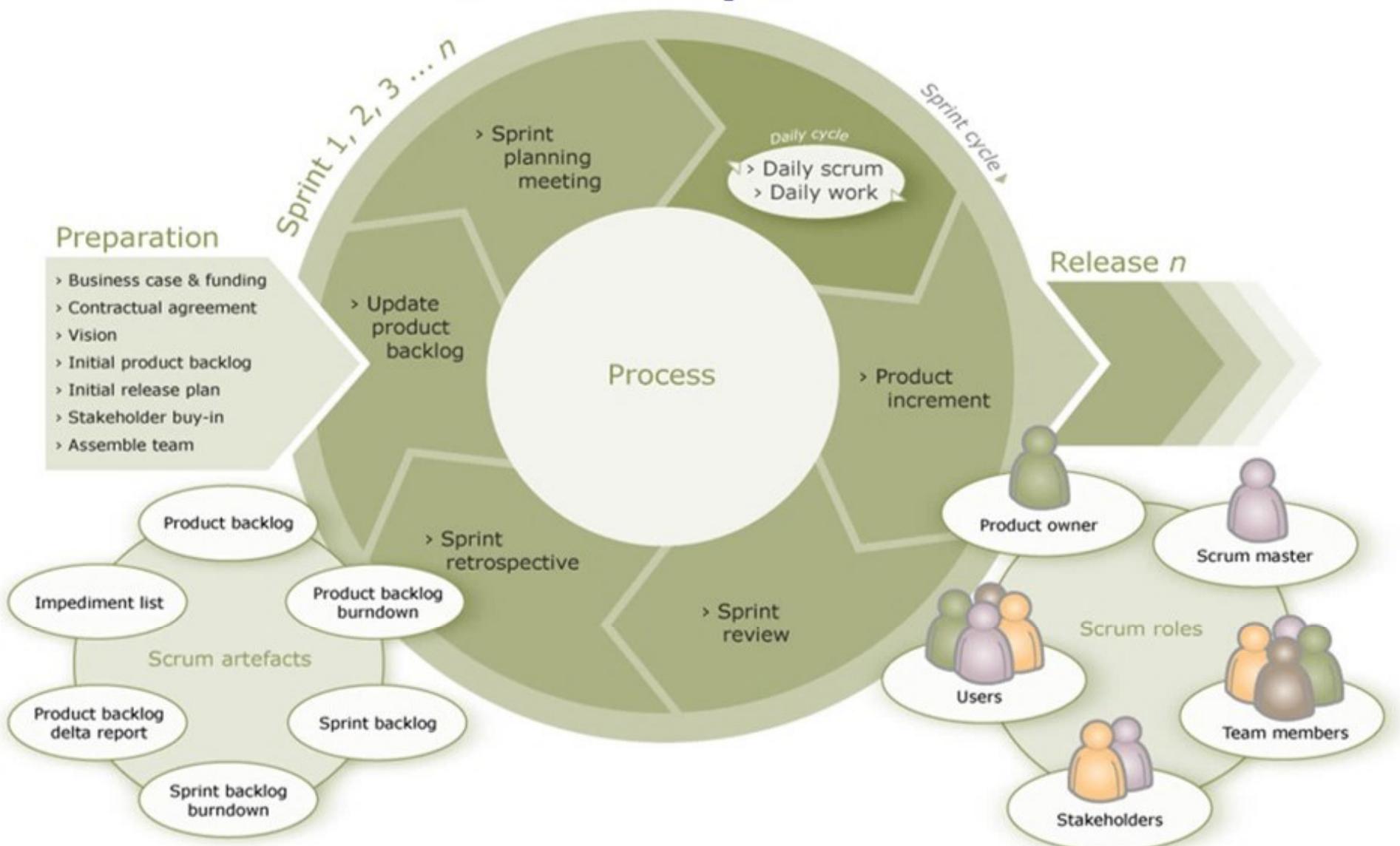
# Distributed Scrum



# Distributed Scrum



# Scrum Preparation



# Key Factors that influenced need for Scrum

- ❖ *Uncertainty* is inherent and inevitable in software development processes and products
- ❖ For a new software system, the *requirements will not be completely known* until after the users have used it
- ❖ It is *not possible to completely specify* an interactive system
- ❖ Ambiguous and changing requirements, combined with evolving tools and technologies make implementation strategies unpredictable

## **“All-at-once” model**

- ❖ “All-at-Once” model is a *single super-programmer creating and delivering an application from beginning to end.*
- ❖ *Concept has a fatal flaw* in that there are at most one or two individuals even in a large company that can execute this model.
- ❖ The next level of “All-at-Once” development *is handcuffing two programmers together.* Pair programming, an eXtreme Programming practice, is an implementation of this.
- ❖ This has been shown to *deliver better code* (usability, maintainability, flexibility, extendibility) faster than two developers working individually. The challenge is to achieve a similar productivity effect with more than two people.

# Scrum Highlights

- ❖ Scrum process is *characterized by 15 minute daily meetings* where each person answers three questions –
  - ❖ *what did you accomplish yesterday,*
  - ❖ *what will you do today, and*
  - ❖ *what impediments are getting in your way?*
- ❖ *Developers consider sets of possible solutions* and gradually narrow the set of possibilities to converge on a final solution.
- ❖ *Decisions on how and where to implement a feature is delayed* until the last possible moment.

# **Conversion of Scrum to “distributed” Scrum**

***Best practices for distributed Scrum consist of:***

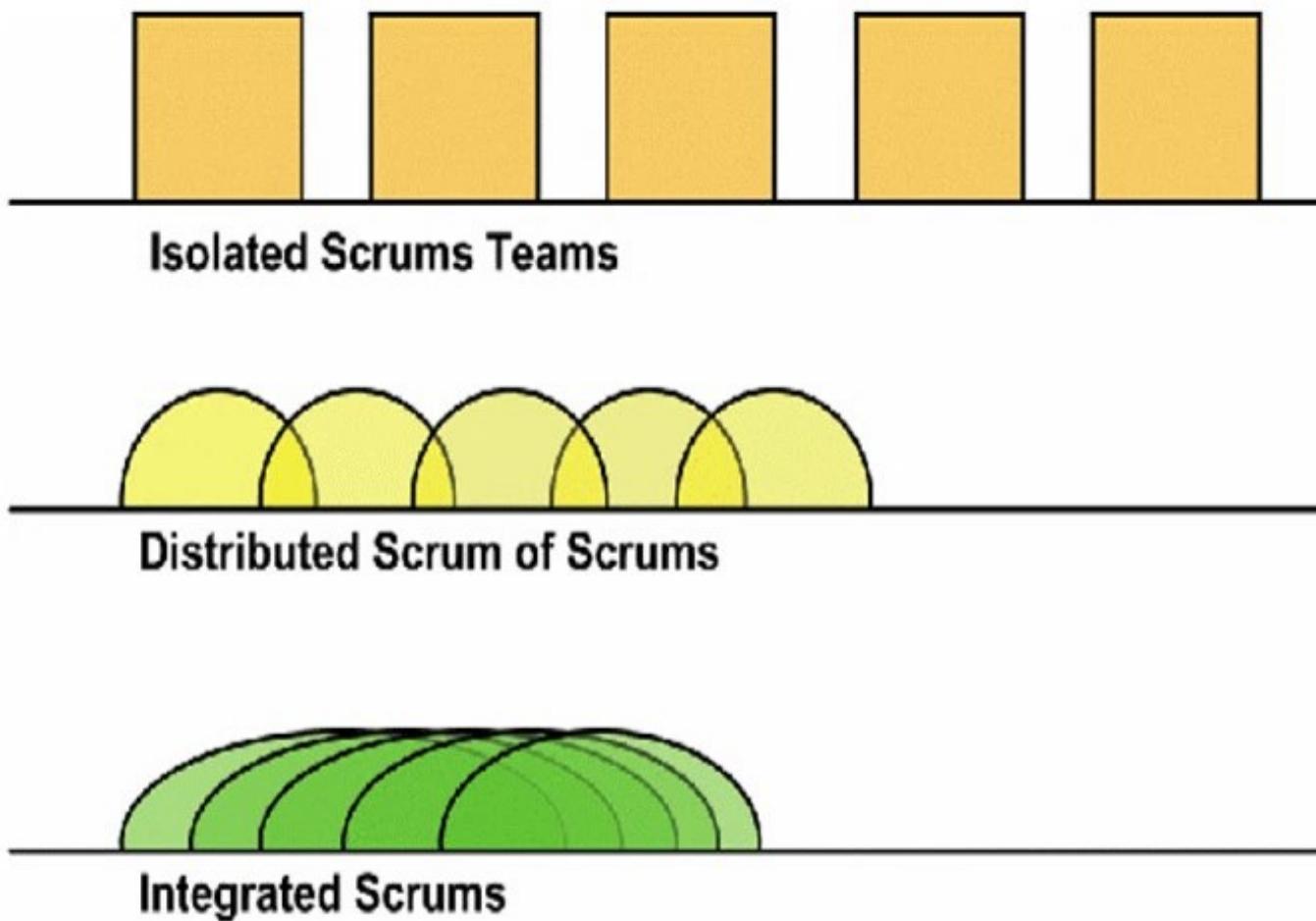
- ❖ ***daily Scrum team meetings*** of all developers from multiple sites,
- ❖ ***daily meetings of Product Owner team***
- ❖ ***hourly automated builds*** from one central repository,
- ❖ ***no distinction between developers at different sites*** on the same team,
- ❖ ***seamless integration of XP practices*** like pair programming with Scrum.

# **Distributed Team Models**

# Distributed Team Models

- ❖ **Isolated Scrums:**
  - ❖ Teams are *isolated across geographies*. In most cases *off-shore teams* are **not cross-functional** and **may not be using the Scrum process**.
- ❖ **Distributed Scrum of Scrums:**
  - ❖ Scrum teams are *isolated across geographies and integrated by a Scrum of Scrums* that meets regularly across geographies.
- ❖ **Totally Integrated Scrums:**
  - ❖ Scrum teams are *cross-functional* with members *distributed across geographies*.

# Distributed Team Models



# Components of Distributed Scrum

## Team formation

- ❖ This was achieved by *splitting teams across sites* and fine tuning daily Scrum meetings.

## Scrum Meetings

- ❖ *Each individual reports on what they did since the last meeting, what they intend to do next, and what impediments are blocking their progress.*

Teams found it necessary to **distribute answers to the three Scrum questions in writing before the Scrum meeting.**

# Problems with implementing Distributed Scrum

- ❖ Most outsourced development efforts use a degenerative form of the Isolated Scrums model where ***outsourced teams are not cross-functional and not Agile.***
- ❖ Typically, ***cross-cultural communication problems*** are compounded by differences in work style in the primary organization vs. the outsourced group. In the worst case, ***outsourced teams are not using Scrum*** and their ***productivity is typical of waterfall projects*** further delayed by cross-continent communications lag time.
- ❖ ***Outsourced teams not using Scrum*** will typically achieve less than half the velocity of a primary site using Scrum.

# Hidden Costs of Outsourcing

- ❖ The hidden costs of outsourcing are *significant, beginning with startup costs.*
- ❖ Under the *survey of 50 companies* and *found that 14% of outsourcing operations were failures*. In the remainder, *costs of transitioning to a new vendor often canceled* out anticipated savings from low labor costs.

# Hidden Costs of Outsourcing

- ❖ The average time from evaluating outsourcing to beginning of vendor performance was **18 months for small projects.**
- ❖ The MIT Sloan Management Review advises readers ***not to outsource critical IT functions.***

# **Top Issues in Distributed Development**

# Top Issues in Distributed Development

- ❖ **Strategic:**
  - ❖ *Difficult leveraging available resources*, best practices are often deemed proprietary, are time consuming and **difficult to maintain**.
- ❖ **Project and process management:**
  - ❖ *Difficulty synchronizing* work between distributed sites.
- ❖ **Communication:**
  - ❖ *Lack of effective communication* mechanisms.

# Top Issues in Distributed Development

- ❖ **Cultural:**
  - ❖ *Conflicting behaviors*, processes, and technologies.
- ❖ **Technical:**
  - ❖ *Incompatible data formats*, schemas, and standards.
- ❖ **Security:**
  - ❖ Ensuring electronic transmission *confidentiality and privacy*.

# Best Distributed Scrum Model

- ❖ *Best practice* recommended by the *Scrum Alliance* is a "*Distributed Scrum of Scrums model*".
- ❖ This model partitions work across cross-functional, isolated Scrum teams while eliminating most dependencies between teams.
- ❖ Scrum teams are linked by a Scrum-of-Scrums where ScrumMasters (team leaders/project managers) meet regularly across locations.
- ❖ *This encourages communication, cooperation, and cross-fertilization* and is appropriate for newcomers to Agile development.

## Sprints

- ❖ Sprints are *two weeks long* on the agile project. There is a *Sprint planning meeting similar to an XP release planning meeting* in which requirements from User Stories are broken down into development tasks.

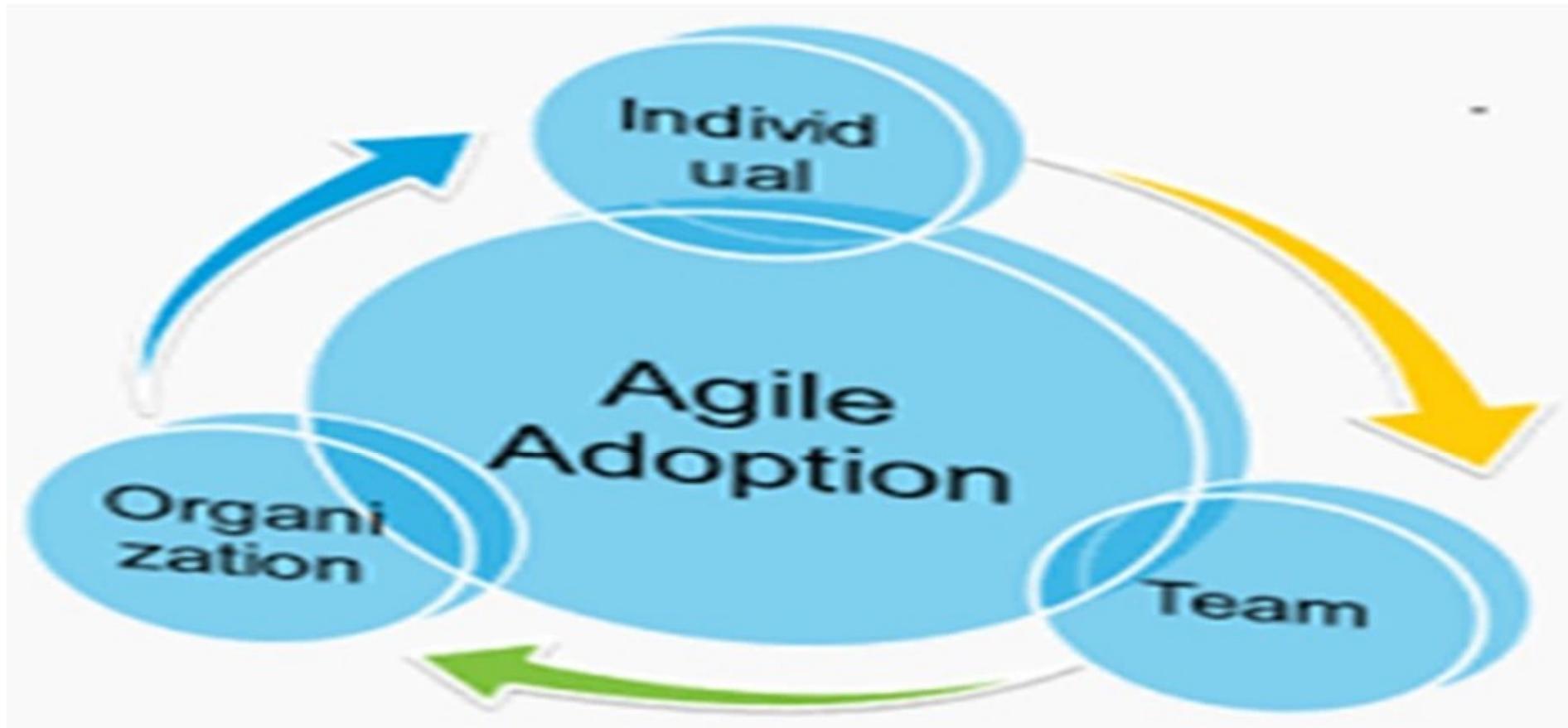
## Product Specification

- ❖ *Requirements are in the form of User Stories* used in many Scrum and XP implementations. Some of them are lengthy and detailed, others are not.

# Testing

- ❖ *Developers write “unit tests”.*
- ❖ The *Test team* and *Product Owners do “manual testing”.*
- ❖ An Automation Test team creates scripts for an automated testing tool.
- ❖ Stress testing is as needed.

# Agile Adoption



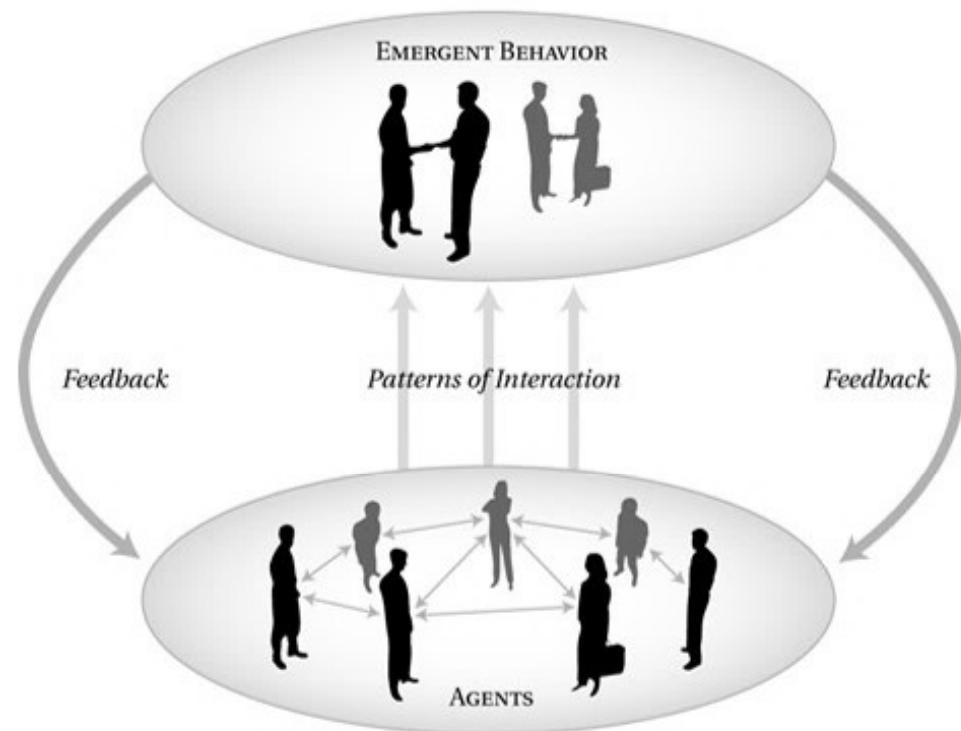
# Agile Adoption

- ❖ A chaordic project harmoniously blends characteristics of both chaos and order – *freedom and control, optimization and exploration, competition and cooperation.*

Agile projects can be seen as chaordic:

## Competition and Collaboration

- ❖ **Agents:** Individuals
- ❖ **Mental Models:** Vision and alignment
- ❖ **Groups:** Project teams



# Agile Adoption

## Emergence and Self-Organization

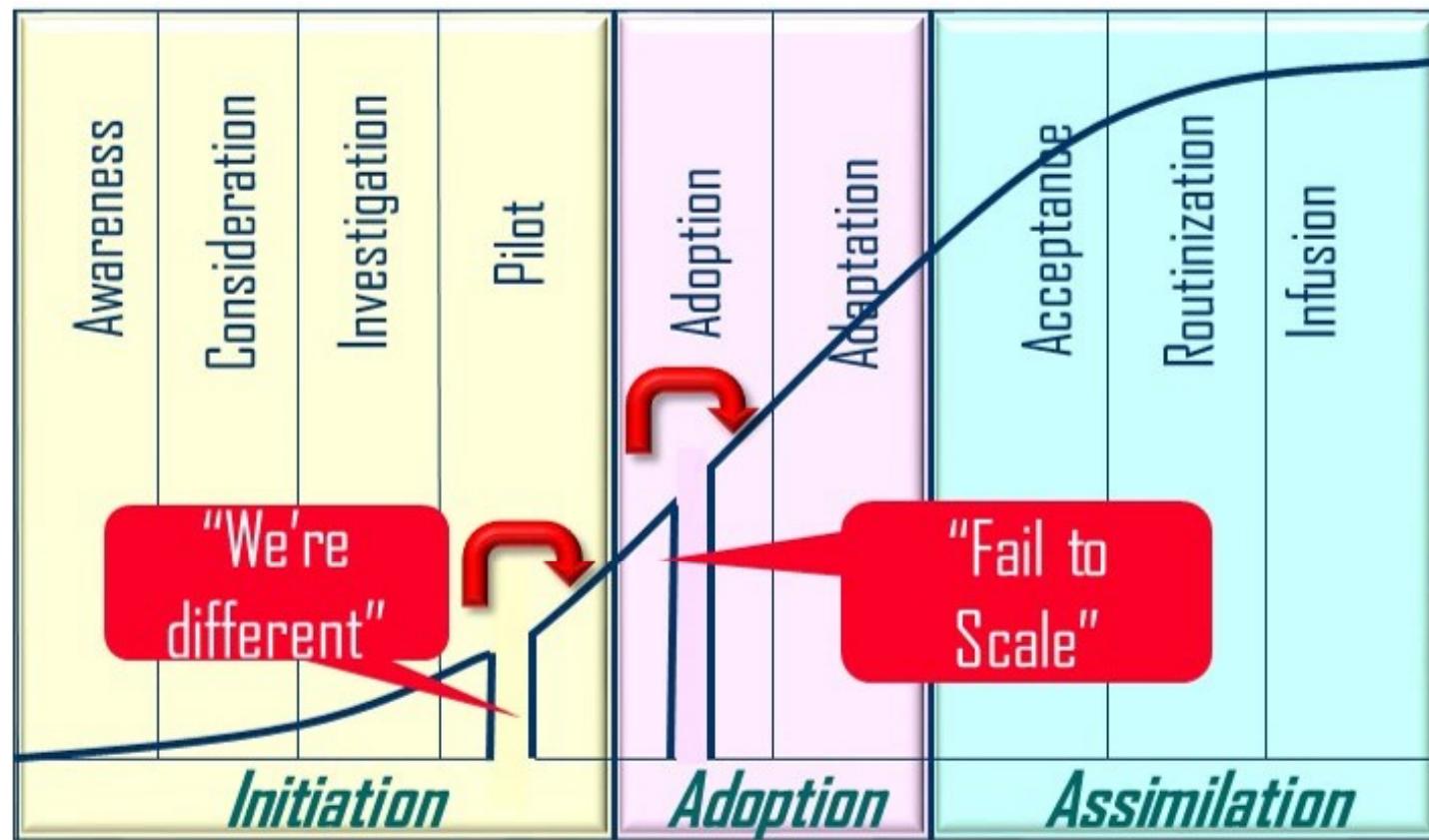
- ❖ **Interactions/Feedback:** Information exchange and relationships among individuals
- ❖ **Simple Rules:** XP/Scrum/FDD Practices

## Learning and Adaptation

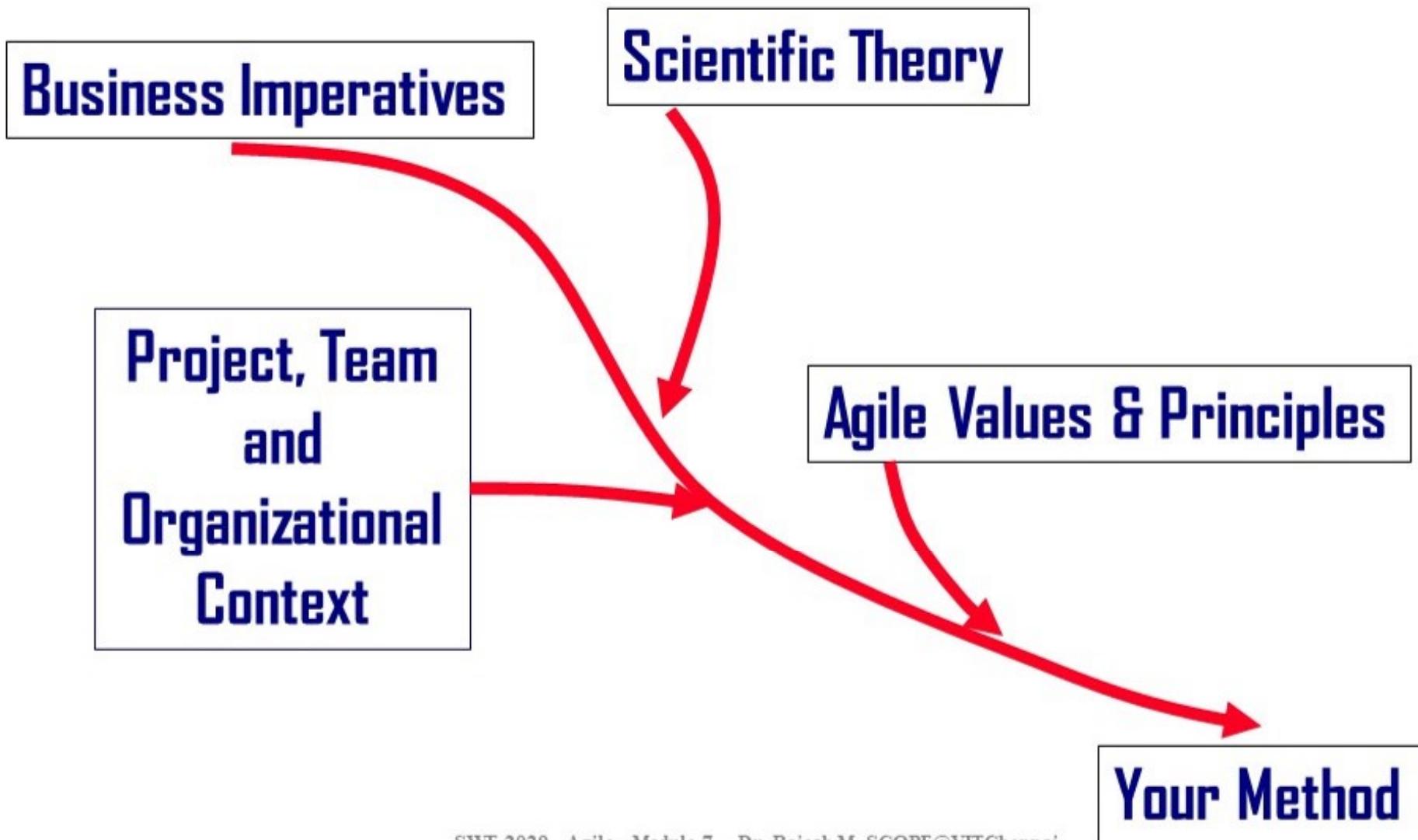
- ❖ **Learning:** Observation, monitoring, measurement and reflection
- ❖ **Adaptation:** Process changes, team adjustments
- ❖ **Environment:** Project environment

# The Agile Adoption Curve

*The Agile Adoption Curve and 'Chasms'*



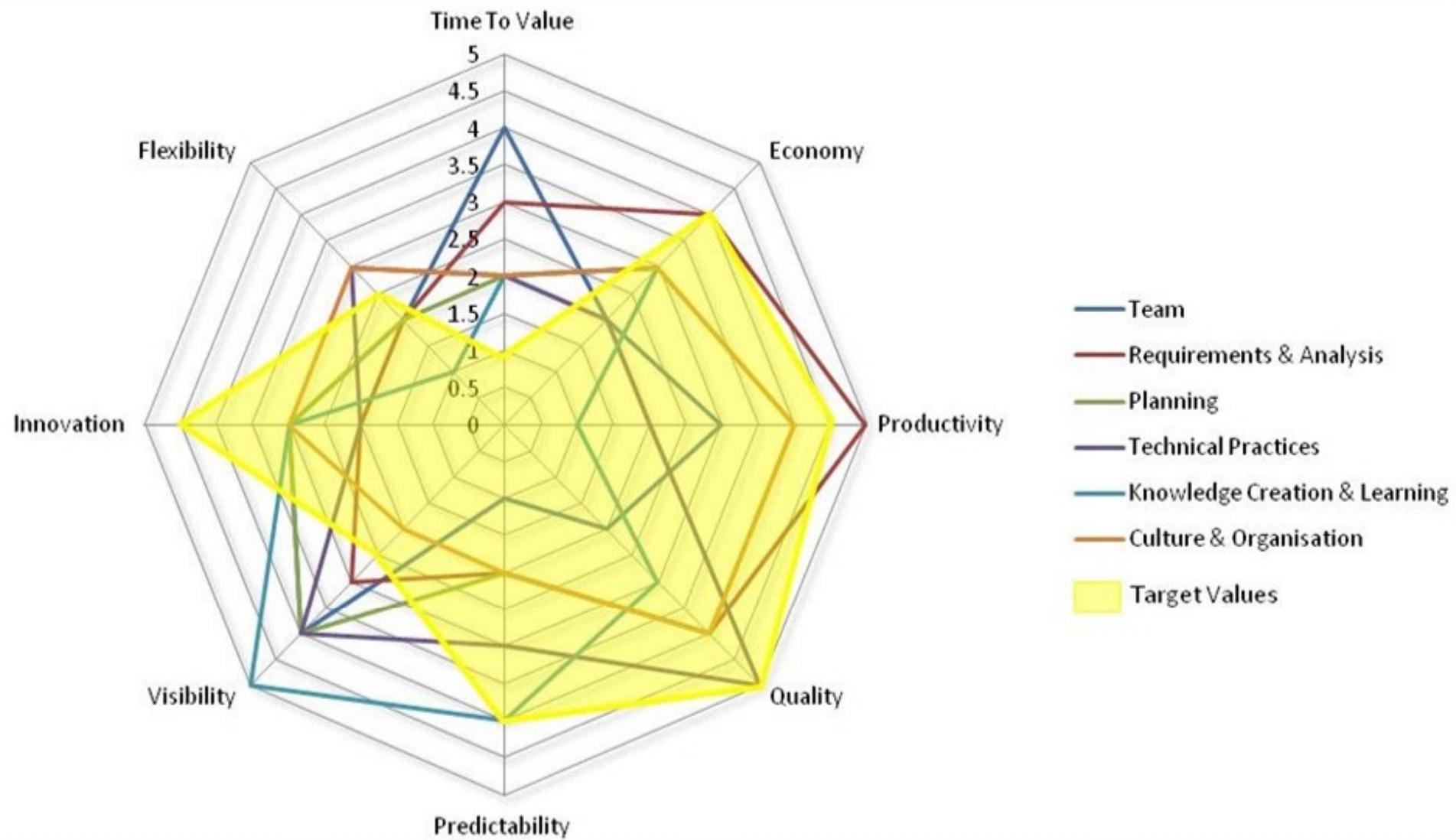
# It depends...



# Where am I?

<i>Assessment Dimensions</i>	Team	Requirements & Analysis	Planning	Technical Practices	Knowledge Creation & Learning	Culture & Organisation
<i>Business Drivers</i>						
Time To Value	4	3	2	2	2	2
Economy	2	4	3	2	3	3
Productivity	3	5	4	2	1	4
Quality	2	4	4	5	3	4
Predictability	1	2	2	3	4	2
Visibility	4	3	4	4	5	2
Innovation	3	2	3	2	3	3
Flexibility	2	2	2	3	1	3

# Where am I?





# **Case Study #1: Scrum vs Waterfall**

# Case Study #1: Scrum vs Waterfall



- ❖ A project that was done initially with a waterfall team and then re-implemented with a Scrum team.
- ❖ The waterfall team took 9 months with 60 people and generated 54,000 lines of code.
- ❖ It was re-implemented by a Scrum team of 4.5 people in 12 months. The resulting 50,803 lines of code had more functionality and higher quality.

# Case Study #1: Scrum vs Waterfall

- ❖ Capers Jones of Software Productivity Research has published extensive tables on average number of function points per lines of code for all major languages.
- ❖ Since the average lines of code per function point for Java is 53, we can estimate the number of function points in the Scrum application. The waterfall implementation is known to have fewer function points.



## **Case Study #2: Distributed Scrum vs World**

## Case Study #2: Distributed Scrum vs Waterfall



- ❖ **Distributed teams working on Horizon 8.0 generated 671,688 lines of code in 14.5 months with 56 people.**
- ❖ During this period they **radically refactored the code on two occasions** and **reduced the code based by 275,000**.
- ❖ They have not been penalized for refactoring as that is rarely done in large waterfall projects in the database from which Capers derived his numbers.
- ❖ They have also not been rewarded for refactoring even though reducing lines of code is viewed as important as adding new code on well run Agile projects.

Jones has also shown from his database of tens of thousands of projects that industry average productivity is **12.5 function points per developer/month** for a project of **900 function points** and that this drops to **3 for a project with 13000 function points.**

	SCRUM	Waterfall	SirsiDynix
<b>Person Months</b>	54	540	827
<b>Java LOC</b>	50,803	54000	671,688
<b>Function Points</b>	959	900	12673
<b>FP per dev/month</b>	17.8	2.0	15.3
<b>FP per dev/month (industry average)</b>	12.5	12.5	3

- ❖ This case study is a proof point for the argument that distributed teams and even outsourced teams can be as productive as a small collocated team.
- ❖ This requires excellent implementation of Scrum along with good engineering practices. The entire set of teams must function as a single team with one global build repository, one tracking and reporting tool, and daily meetings across geographies.
- ❖ Outsourced teams must be highly skilled Agile teams and project implementation must enforce geographic transparency with cross-functional teams at remote sites fully integrated with cross-functional teams at the primary site.

# **A case study of a scrum project**

# A case study of a scrum project

## Case Study – WDS Global:

- WDS Global successful project for over four years.
- It consisted of **one global platform** with XP teams in three locations – the **U.S., the U.K., and Singapore**.
- The **teams used** a core set of **Extreme Programming practices** – **test driven development, pairing, refactoring, and continuous integration**, in combination with the other distributed patterns.

# A case study of a scrum project

## Case Study – WDS Global:

- They made use of a **single integrated global code base** (no branching), and a **single continuous integration server**.
- There were **daily hand-offs between each location**.
- Even though the **teams operated on two-week iterations, weekly production releases** were carried out with **collaboration between Development and IT**.

# A case study of a scrum project

## Further Reference:

<http://www.scrumcasestudies.com/>

# **Scrum Success Stories**

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

- Just a few short years ago I began singing the praises of Agile to just about anyone who would give me the time of day. I would talk about the values associated with agility and dig deeply into the various Agile Manifesto principles that resonated with me the most. Having come from a fairly diverse background, I was convinced that Agile methods could be used in a multitude of business applications, not just software development. At the time, I was a Scrum practitioner and truly embraced the mind-set of continual inspection and adaptation. Unbeknownst to me at the time, I was indirectly sowing seeds that would one day provide the ideas to help take a family practice medical facility to an entirely new level. This is Part I in a series in which I'll share the approach, benefits, and ongoing practices associated with a medical practice that has truly embraced Agile.

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

- ❖ For those who may not be familiar, family practice medical facilities see a wide variety of medical conditions every day. Staff members (administrators, doctors, nurses, medical assistants, etc.) are constantly evolving in order to meet the needs of their patient population. The demands placed on these professionals are enormous, and they carry a heavy burden of social and moral responsibility. However, the burden isn't one that is carried by one person alone . . . nor would some say that it's even a burden to begin with. Below is the story of how one family practice, with a mind-set of continual evolution, started its own Agile transformation.

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

## The background

- ❖ The medical practice, located in the southeastern United States, had experienced tremendous success during the last five years. Initially founded by a small number of providers, the practice is geographically situated in a location that contains an excellent mix of young, middle-aged, and elderly patients. The patient population required that the practice develop and continually enhance an exemplary standard of care.

# Scrum Success Stories



## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)

<https://www.linkedin.com/in/ericsking>

### The background

- During the last five years, the practice had doubled in size on two different occasions. At any organization, any time the employee base doubles, procedures and processes need to be revisited. As the practice grew, various leaders understood the definitive need to focus on continual learning in order to provide patients with the best level of care. In addition, due to the geographic location chosen for the practice, leaders knew that they had to be "doing it better than everyone else" so that they could keep their competitive edge. Given these characteristics and with the support of multiple practice leaders, there were certain aspects of agility that the leaders immediately embraced and began to implement.

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

## The approach

- Many family practices emphasize maintaining and continually developing standards of care for doctors, nurses, and medical assistants. Regardless of where the team member received his or her initial medical training, constant refinement of medical skills was determined to be the most important item as the family practice created its initial product backlog. In addition, so that a strategic alignment could be created between the front office and the back office, two product owners were selected to represent and prioritize the needs of the practice. The product owners chosen were the family practice administrator and the lead physician for the practice.

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

### Team formation

- The practice expressed, from the beginning, a deep desire to create the best teams possible so that significant improvement could be made in the critical skills area. Initially, the practice considered forming Scrum teams of up to seven members before embarking on the first sprint.
- However, team members expressed a strong desire for much smaller teams so that continual improvements could be made without negatively affecting the normal work day and the patient "flow" within the office. Therefore a decision was made to create teams of three, based on both pairing and pairing. Below is a sample of the smaller team structure:

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

### Iteration 1 outcome

- At the end of the month, the family medical practice had a chance to showcase its skills and to discuss the learnings that took place among the various practitioners. With a definitive commitment from everyone, the results were remarkable and helped set the stage for further iterations.
- Several highlights were recognized by the organization. One example is that, in lieu of waiting until the end of the month, each of the seven teams demonstrated their clinical skills multiple times during the month. Respective team members would "demo" their clinical skills to at least one doctor in the practice. These demonstrations were conducted as part of the regular patient experience within the practice, which meant that clinical skill improvements were being demonstrated where they had the most value -- with actual customers. After each demo, doctors would provide immediate feedback to the individual who performed the demo as well as to his or her additional team members.

# Scrum Success Stories

## Success Story: Agility and Health Care

A Family Practice Transformation Based on Continual Learning and Adaptation

(14 January 2014 by Eric King)



<https://www.linkedin.com/in/ericsking>

- As highlighted in this short article, Agile isn't just about software development these days. Agile can certainly be about people, managerial, organizational, and executive development. There is a definitive need for agility in many facets of organizational structures, and I hope that the initial success seen by this family practice will give you ideas about how to expand Agile in your sphere of influence. Stay tuned for more Agile-based success stories in the health care space. The next article will review how this family practice incorporated aspects of the "Undercover Boss" and Agile in order to continue its incremental improvement journey. Until then. . . .

# **Scrum Success Stories**

**Success Story:**

**Transition from Agile-cum-Waterfall to Scrum :**

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ I was not a Certified ScrumMaster when this story began. I had about 13 years of experience in software services. I joined the Global Products team as a ScrumMaster, however. I attended the ScrumMaster training conducted by Pete Deemer at Bangalore and carried back a lot of enthusiasm for implementing Scrum at Global Products.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum

**to Scrum :** 03 January 2014 by Bharath Malaivaiyavur

- ❖ Global Products was the team that developed Internet products for our customer. There were about 50 people on the team in Chennai, and they were working as a single team. The customer manager managed the project in a time-and-materials contract and had been leading the team for the past three years toward developing the products via Waterfall but within Agile methods. The customer's senior management had been sold on the idea of implementing Scrum at Global Products as a pilot to extend the Agile philosophy across the account.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ I was the second of 5 ScrumMasters for the team. Upon my return to Chennai, I briefed my bosses about the training I had attended and shared with them the principles of Scrum, which I had learned from Pete. Eventually, on my insistence, the 50-member team was split into 5 smaller teams of about 9 to 10 members each, and I was identified as the Scrum Master for two teams named Bugatti and Mercedes.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum

: 03 January 2014 by Bharath Malaivaiyavur

- ❖ I worked with the project management office (PMO) team, which conducted the Scrum of Scrums to do the release planning and assignment of features to the Scrum teams. The sprint schedule was arrived at for all the teams.
- ❖ During the first sprint, I introduced my teams to sprint planning, estimation using planning poker, the Daily Scrum call, sprint burn-down, sprint retrospectives, etc. The teams started showing a keen interest in all these things, which had not been practiced previously. Initially, I had the backing of my bosses as well.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ The customer manager was not convinced that the new model would work and expressed his worries to his management and to my bosses as well. I had invited the customer manager to attend the daily Scrum meetings, although as an optional attendee. I thought this would dispel some concerns about the new way of working that the teams had adopted.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ Two sprints had passed; my team members were showing great enthusiasm and active participation in all the Scrum ceremonies. During the retrospective meetings, certain things became clear: that they were working more continuously than before, that they did not have to wait for the requirements after the sprint had begun, that they could actually collaborate more effectively among themselves, and that they could find ways to improve their work in a friendly setting.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- The challenges that remained included winning the customer's confidence; ensuring that defects were reduced on the sprint deliverables; and ensuring that the teams were able to interact more frequently with the business analysts, who were proxy product owners working in a different continent and time zone.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ During the third sprint, I had made sure the requirements discussion happened during the overlap hours. This had two benefits. One was the closer interaction between the teams that developed the Internet products and the business analysts who documented the proposed products. The second benefit was a reduction in the number of defects.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ The challenge that remained for the fourth sprint was winning the customer's confidence. I set up a discussion between the customer manager and my teams and asked the question, "What should we do differently to be able to satisfy the customer?" Different opinions were voiced. Finally, after due discussion, the teams agreed that in order to win customer confidence, they would have to complete the full feature by the end of this sprint and demonstrate it to the customer. This was not going to be an easy task. I checked the project plan and confirmed that this feature had been planned for completion by the end of the fifth sprint. This meant that the teams had set themselves the goal of completing this feature a whole sprint in advance.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum :

03 January 2014 by Bharath Malaivaiyavur

- ❖ We started the requirements discussion for the fourth sprint early, three days ahead of the start of that sprint. The teams signed off on the requirements on the scheduled sprint start date and decided that they would forgo afternoon coffee breaks for the sprint. They also decided to work weekends. On the penultimate day of the sprint, the teams announced that they had completed all the deliverables for the sprint, called the testers to validate the feature, and the whole feature was accepted by the business analysts -- at the end of the fourth sprint.

# Scrum Success Stories

## Success Story: Transition from Agile-cum-Waterfall to Scrum :

03 January 2014 by Bharath Malaivaiyavur

- ❖ The customers lauded the teams for making this huge contribution, and with such good quality. This was a huge morale booster for the teams. The members had not only embraced Scrum themselves but had also convinced avowed Waterfall champions that Agile using the Scrum framework was the way forward for development of Internet products.

# Scrum Success Stories

Success Story: Transition from Agile-cum-Waterfall to Scrum : 03 January 2014 by Bharath Malaivaiyavur

- ❖ Team members were feeling motivated and the customers were feeling more satisfied with the products. Managers breathed more easily. And that's how a cultural transformation took place at Global Products.

# Scrum Success Stories

## Further Studies:

- ❖ Please Visit :

<https://www.scrumalliance.org/success-stories>