CitiusTech

accelerating innovation in healthcare

Smart Scrum

Session 1

Version 1.0

December 2019

CitiusTech has prepared the content contained in this document based on information and knowledge that it reasonably believes to be reliable. Any recipient may rely on the contents of this document at its own risk and CitiusTech shall not be responsible for any error and/or omission in the preparation of this document. The use of any third party reference should not be regarded as an indication of an endorsement, an affiliation or the existence of any other kind of relationship between CitiusTech and such third party

Team Introduction

- Name of the Participant
- Years of Experience
- Current Project working on
- Prior experience with Scrum/Agile

Agenda

- **Introduction**
- Understand Scrum
- Adaptation to Scrum
- Scrum Framework

Course Objectives

- This 6 hours (3 Days) course is to lay a solid foundation for individuals or teams to begin effectively using Scrum immediately
- These sessions will provide information and practices that you can use to ensure that Scrum is understood and implemented effectively
- This course covers Scrum basics, including the framework, mechanics, and roles of Scrum
- It helps in the following area:
  - How to be an effective Scrum Team Member?
  - Improve teamwork
  - Understand how Agile helps deliver regular value to business
  - Improve engineering practices to deliver high-quality product

Course Content

Session1	Session2	Session3
Introduction	Scrum Roles	Mind Reading (Game)
Understand Scrum	Scrum Ceremonies	Scrum Estimates
Adaptation to Scrum	Scrum Artifacts	Estimation Ball (Game)
Values Advocacy (Game)		

Agenda

- Introduction
- **Understand Scrum**
- Adaptation to Scrum
- Scrum Framework

Understand Scrum

- Introduction to Agile
- Scrum Objective
- Scrum Principles

Introduction to Agile (1/4)

What is Agile?

- Agile is the ability to create and respond to change. Agile software development processes are faster with respect to delivery and customer feedback
  - Shorter development cycles
  - Active customer involvement helps for better output
  - Deliveries are goal based instead of plan based
  - Changes in requirement based on business priority can be catered

Why Agile?

- Traditional software development processes are too heavyweight or cumbersome
  - Traditional software development is too rigid
  - Difficulty with incomplete or changing requirements
  - Longer development cycles give delayed outcomes
  - Receives feedback late in cycle
  - Customer involvement is at initial phase and at the end of cycle

Introduction to Agile (2/4)

Agile Manifesto

Individuals and Interactions	over	Process and Tools
Working Software	over	Comprehensive Documentation
Customer Collaboration	over	Contract Negotiation
Responding to Change	over	Following a Plan

In Scrum focus, while there is value in the items on the right, we value the items on the left more.

Introduction to Agile (3/4)

Below are the 12 key principles

- **Early and Continuous Delivery of Valuable Software -** Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- **Embrace Change** - Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- **Frequent Delivery** - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- **Business and Developers Together** - Business people and developers must work together daily throughout the project.
- **Motivated Individuals** - Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done.
- **Face-to-Face Communication** - The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Introduction to Agile (4/4)

Key Principles

- **Working Software** - Working software is the primary measure of progress.
- **Constant Pace** - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- **Technical Excellence** - Continuous attention to technical excellence and good design enhances agility.
- **Simplicity** - Simplicity–the art of maximizing the amount of work not done–is essential.
- **Self-Organizing Teams** - The best architectures, requirements, and designs emerge from self-organizing teams.
- **Regular Reflection and Adjustment -** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Game

Agile Values Advocacy Game

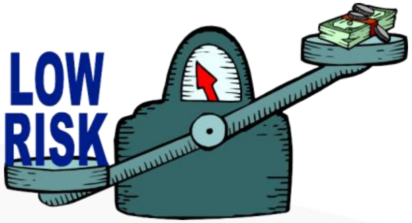
- **Moto**
  - Meaningful discussion about the values in the Agile manifesto
  - Emphasis the importance of both, the left and the right side of the value statements

Agile Benefits for Business

Quicker ROI

Lower Total Cost

Respond to Change



Reduce Risk

Faster Time to Market

Stakeholder Relations

Agile Benefits for Developers

Teaming

A Sense of Done

Quality Work

Rhythm

Visible Work

Create
Feedback
Improve
Validate

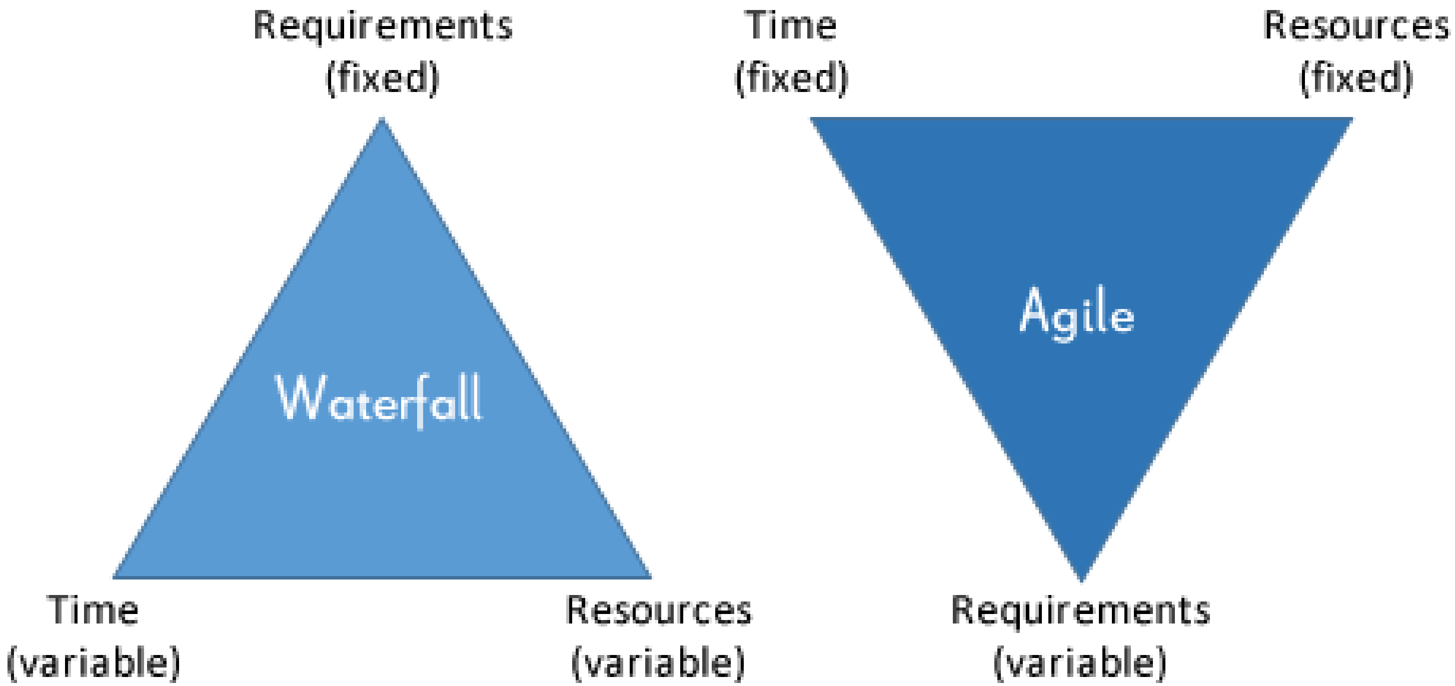
Comparing Waterfall & Agile

Old Approach

Create a project plan, then measure success by **adherence to plan.**

New Approach

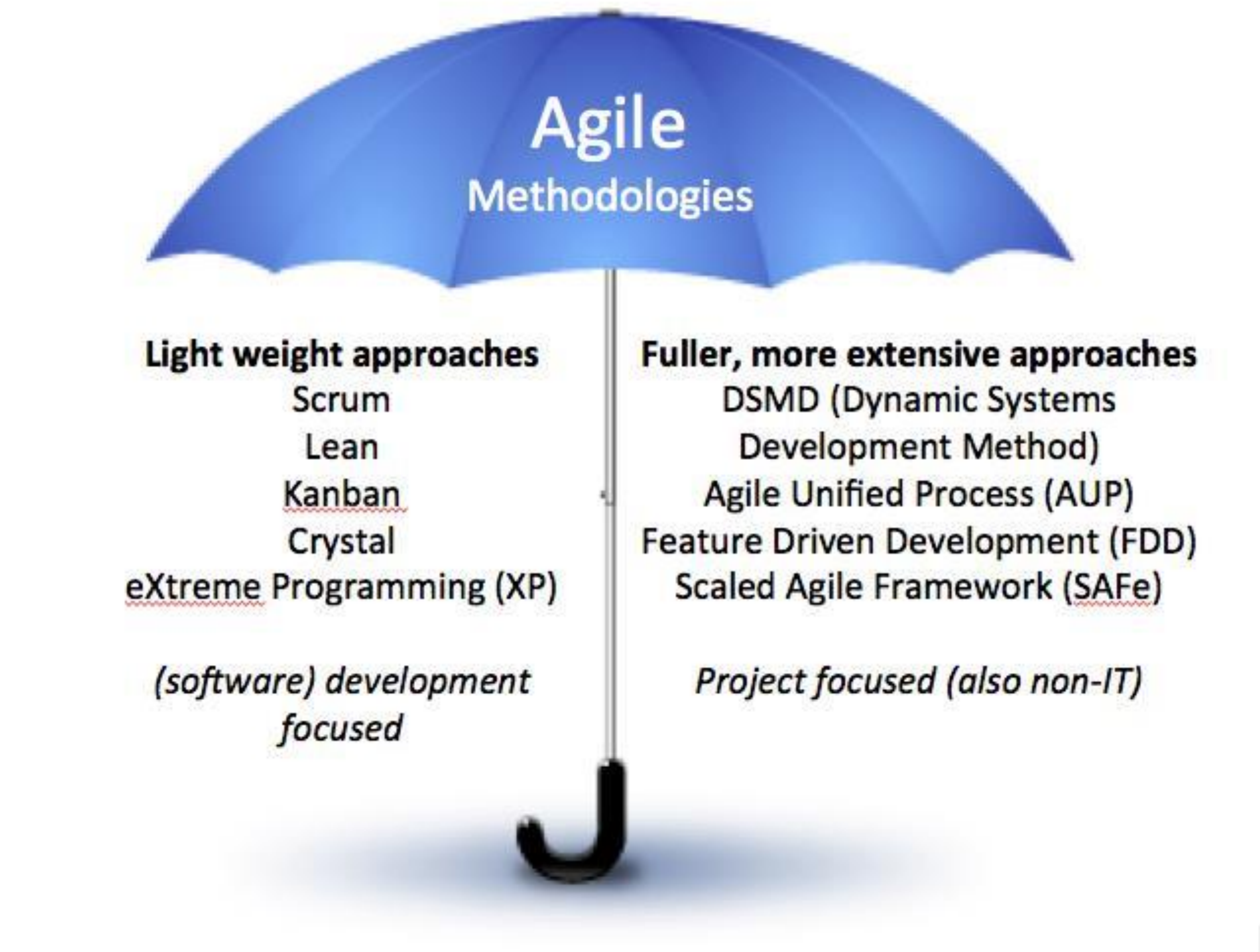
Maximize value and quality within specified project constraints.



Comparing Waterfall & Agile

	Scrum	Traditional Project Management
Emphasis is on	People	Processes
Documentation	Minimal—only as required	Comprehensive
Process style	Iterative	Linear
Upfront planning	Low	High
Prioritization of Requirements	Based on business value and regularly updated	Fixed in the Project Plan
Customer involvement	High throughout the project	Varies depending on the project lifecycle
Leadership	Collaborative, Servant Leadership	Command and control
Change	Updates to Productized Product Backlog	Formal Change Management System
Quality assurance	Customer centric	Process centric
Organization	Self-organized	Managed
Management style	Decentralized	Centralized

Agile Methodologies



Light weight approaches

Scrum
Lean
Kanban
Crystal
eXtreme Programming (XP)

(software) development focused

Fuller, more extensive approaches

DSMD (Dynamic Systems Development Method)
Agile Unified Process (AUP)
Feature Driven Development (FDD)
Scaled Agile Framework (SAFe)

Project focused (also non-IT)

Agenda

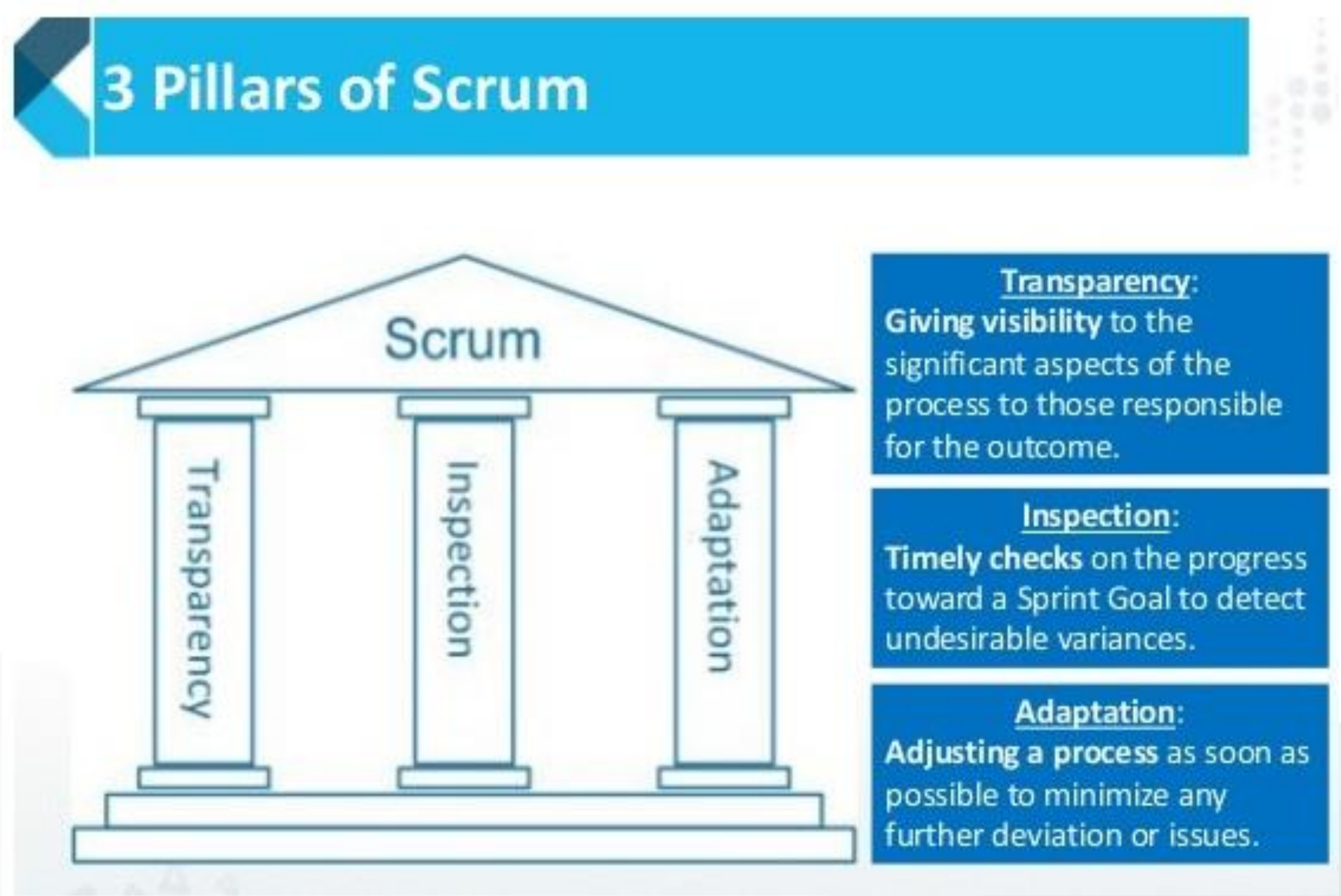
- Introduction
- **Understand Scrum**
- Adaptation to Scrum
- Scrum Framework

What is Scrum?

- Scrum is not an acronym. It is a name taken from the sport of Rugby, where everyone in the team acts together to move the ball down the field
- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Scrum makes this possible using sprints where a team can deliver a product feature in an iteration (Typically 10 days).
- Every two weeks to a month anyone can view real working software and decide to release it as is or continue to enhance it for another sprint
- Producing product incrementally and iteratively reduces the risk and enhance visibility

Video: Scrum

3 Pillars of Scrum



3 Pillars of Scrum

Transparency:
Giving visibility to the significant aspects of the process to those responsible for the outcome.

Inspection:
Timely checks on the progress toward a Sprint Goal to detect undesirable variances.

Adaptation:
Adjusting a process as soon as possible to minimize any further deviation or issues.

Scrum Values

- **Commitment**
  - Be willing to commit to a goal. Scrum provides people all the authority necessary to meet their commitments.
- **Focus**
  - Do your job. Focus all of your efforts and skills on doing the work you've committed to doing.
- **Openness**
  - Scrum keeps everything about a project visible to everyone.
- **Respect**
  - It is important to respect the different people who comprise a team.
- **Courage**
  - Have the course to commit, to act, to be open, and to expect respect.

Commitment
Courage
Focus
Scrum
Respect
Openness

Companies Using Scrum

Microsoft®

Google™

YAHOO!®

NOKIA

JOHN DEERE

® LexisNexis®

SIEMENS

Salesforce

EA ELECTRONIC ARTS™

intuit®

Sabre

océ

nielsen

HIGHMOON STUDIOS®

First American Real Estate Corp.

CitiusTech

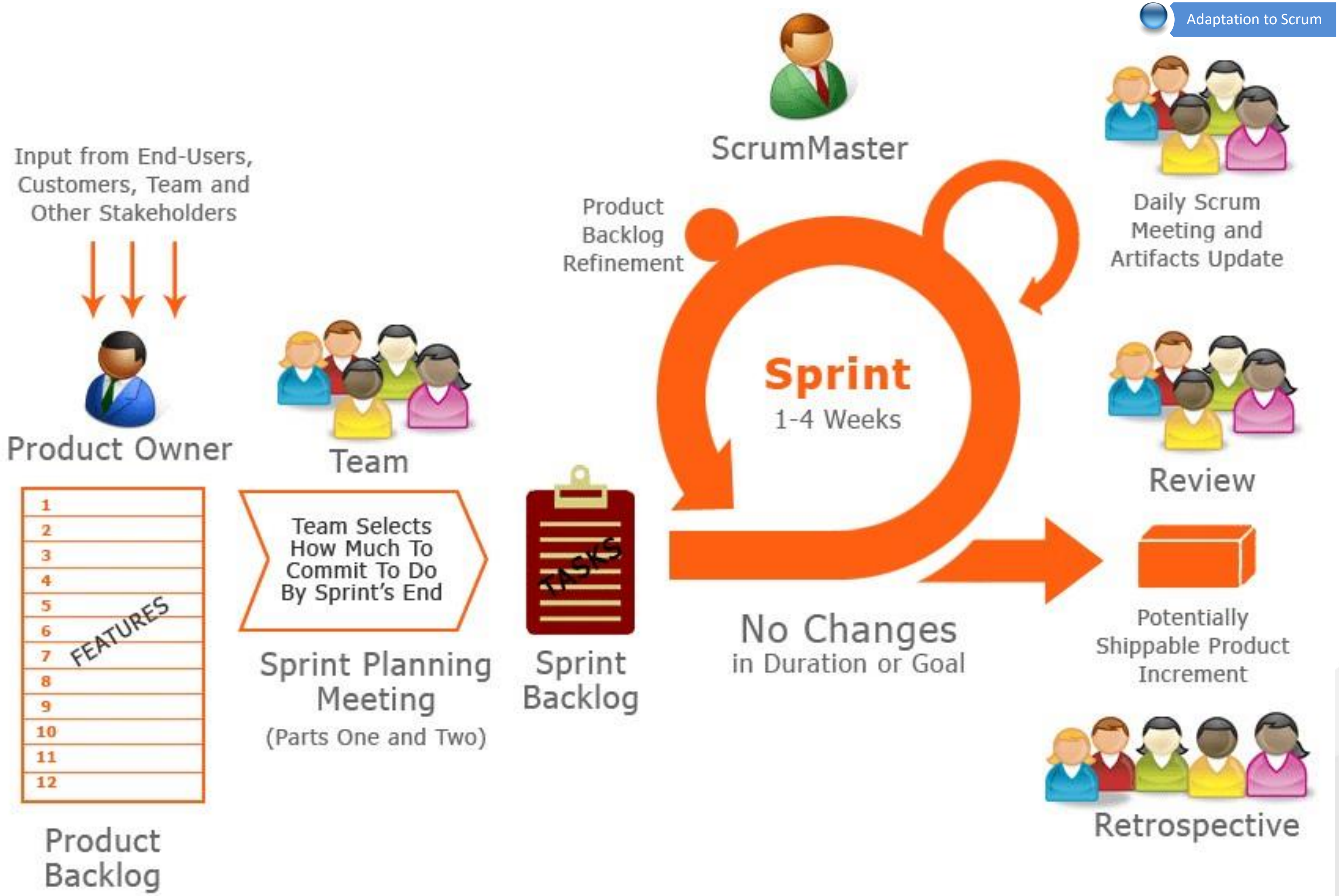
Agenda

- Introduction
- Understand Scrum
- **Adaptation to Scrum**
- Scrum Framework

Adaptation to Scrum

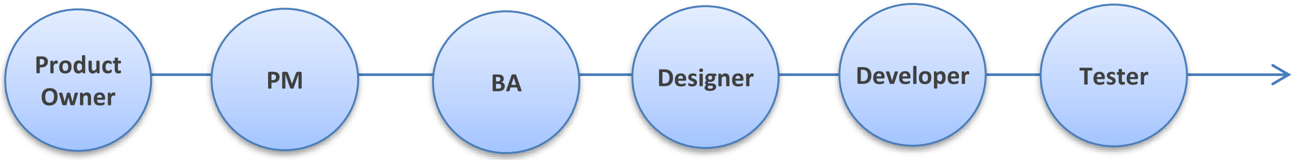
- How Scrum Works?
- Key Differences
- Myths Around Agile Development

Scrum – How does it work?


Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
Team
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Product Backlog
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
TASKS
Sprint Backlog
Product Backlog Refinement
ScrumMaster
Sprint
1-4 Weeks
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective

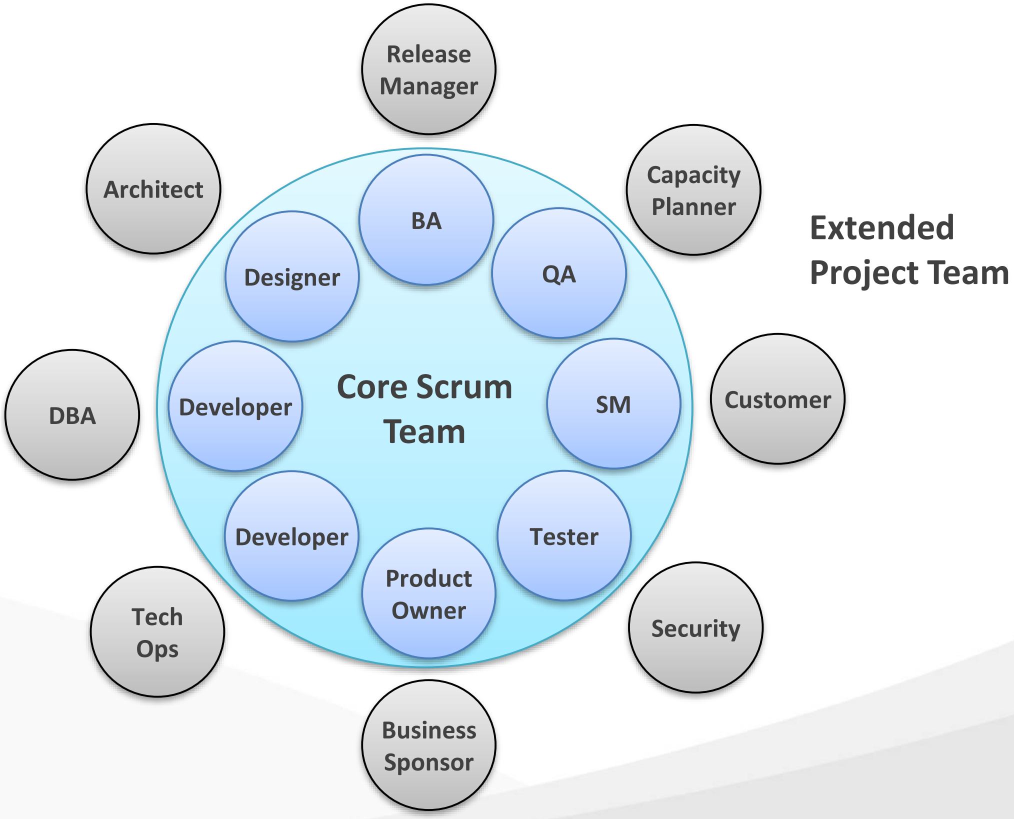
Team Structure

Traditional structure



Integrated Agile Team

The Core Project Team ideally consists of **5-9 (7 plus or minus 2)** members



Where Scrum is ‘Not Effective’ and ‘Effective’ Scrum?

Where Scrum is NOT effective?

- Lack of Customer Involvement
- Big sized team (> 8-10)
- When the budget and schedule is fixed
- Requirements changing drastically during the sprint
- Working with client that enforces their very formal approach on vendors
- Although not impossible, its hard to implement Scrum when all team members are not in the same location
- Small size projects

Where Scrum is effective?

- Project where scope is changing rapidly (for whatever reason)
- Fast feedback and burning visibility
- Team size is 5-7 members
- Requirements are not clearly defined
- Work delivered is incremental
- Control and management is empirical (practical)

Myths around Agile Development

- Agile doesn't allow documentation
- Agile methods do not scale
- Agile means no plan
- Agile is undisciplined
- Agile doesn't need up front design
- We're doing scrum so we don't need to do TDD, Refactoring Pair Programming, etc.
- One can learn Agile from a book
- Agile is a silver bullet solution to software engineering problems (There is NO silver bullet solution)

Scrum But's

- There are many adaptations or modifications of Scrum by many organizations
- These custom adaptations of "We use Scrum, but X so Y" pattern are called as **"Scrum but's"**
- **Scrum Master's** primary role is to ensure the Scrum principles are followed to ensure project's success
- Examples of **Scrum But's**:
  - If you ask a Team member whether they use Scrum and you get a respond, "Yes, we use Scrum – but our manager believes that we will fail if he doesn't tell us what to do, so we are told what to do and how to do it, so feel safe"
  - "We use Scrum, but Retrospectives are a waste of time, so we don't do them"
  - "We use Scrum, but we can't build a piece of functionality in a month, so our Sprints are 6 weeks long"

Kanban overview

- Kanban – an agile method for incremental improvements
- It is a pull system versus a push system, instead of, “You start left with the Backlog, and push your work items along through the in progress”
- It is a visual system for managing work as it moves through a process. Kanban visualizes both the process (the workflow) and the actual work passing through that process
- No defined velocity as there is no ‘start’ and ‘stop’ date defined
- Goal of Kanban is to identify potential bottlenecks in your process and fix them so work can flow through it cost-effectively at an optimal speed or throughput
- Identify ‘Class of service’ or ‘Swim lanes’ to have efficient tracking of efforts and distribution of work on the Kanban board

Kanban Fundamentals

Predictability over Throughput

- How long it takes something to get through, not how much gets through

Management via Policies over Management by Intervention

- Kanban provides self-directed team levers – don’t manage by exception

Continuous Improvement over maintaining the status quo

- Kanban provides a framework for adaptation and improvement

Helps to Identify Bottlenecks for removal

- Recognize interruptions to flow, and work to eliminate them

Increase Quality by Limiting Work in Progress

- Focus on a few items at a time, and do them well

Transparency

- No secrets, no surprises, maximize visibility

Swarming Behavior

- Everyone helps to remove impediments, blocked items are a big deal

Lean Programming

- Lean Thinking capitalizes on the intelligence of frontline workers, believing that they are the ones who should determine and continually improve the way they do their jobs.
- Mary and Tom Poppendieck have transferred principles and practices from the manufacturing environment to the software development environment
- Principles of Lean Thinking are as follows –
  - Eliminate Waste
  - Increase Feedback
  - Delay Commitment
  - Deliver Fast
  - Build Integrity In
  - Empower the Team
  - See the Whole
- The lean production metaphor is a good one for software development, if it is applied in keeping with the underlying spirit of lean thinking.

Resources

- **Online Discussion Groups**
  - Agile Project Management
  - Scrum Development
  - Agile Manifesto

- **User Groups**
  - Italian Agile Movement
  - Agile Alliance User Group List

- **Articles**
  - Introduction to Agile Methods
  - The New Methodology
  - Getting Started with Agile Delivery
  - So, How’s that Agile Initiative Doing?
  - Agile Project Management: Emergent Order through Visionary Leadership
  - The Lean-Agile PMO: Using Lean-Thinking to Accelerate Agile Delivery

Thank You

CitiusTech Markets

CitiusTech Services

CitiusTech Platforms

Accelerating Innovation

CitiusTech Contacts

Email ct-univerct@citiustech.com

www.citiustech.com