Contents

[Agenda 1](#_Toc82938790)

[The added value of Scaling Agile 1](#_Toc82938791)

[Time to Market 2](#_Toc82938792)

[Modern Issues for Delivery Organizations in Corporate Structures 4](#_Toc82938793)

[Scaling Agile Learning Path 4](#_Toc82938794)

[About Agile 4](#_Toc82938795)

[Agile Manifesto (Platform) 4](#_Toc82938796)

[Agile Philosophy 6](#_Toc82938797)

[Agile 6](#_Toc82938798)

[Issue with Agile at scale 8](#_Toc82938799)

# Agenda

Get the benefits of an agile scaling frameworks & its fundamentals

1. Value of Scaling Agile
2. Foundational Values & Principles
3. Processes, Roles & Responsibilities

# The added value of Scaling Agile



**Moved to…**

**De-Centralized Structure**

**Organizational Structure**

* Scale agile has new Organizational model that have classical **Organizational structure, which moves to decentralized model** where almost all the autonomy & decision-making power is given back to the teams building the products in organization
* Framework provides
  + The processes
  + Roles & Responsibilities needed to all the teams work together & let us create large, complex, and integrated products
* Agile scale has been around for quite a while, and the benefits of an organization going through the transformation are become clear

## Time to Market



**Engagement**

Research has shown time & time again that engagement in agile organization is just higher

**Quality**

Product Quality is going up because of the inherent quality structures that are present within a framework

**Productivity**

Increase in productivity because of core principles is,, the teams stick together for as long as possible so that productivity is increases over time

**Time to Market**

Faster time to market, because the framework does away with lot of dependencies, that teams have in large organization

* When autonomy rests on a team level, people become more engaged, which does wonders for a turnover rate increase in our organization
* Now we need to understand, how agile at scale achieves this is one of the fundamental questions

# Modern Issues for Delivery Organizations in Corporate Structures

## Scaling Agile Learning Path

1. **Scaling Agile – Getting Started**

Introduce the added value of an agile scaling framework with introduction of

* + All the processes
  + Roles & responsibilities present in the framework
  1. **Executing a Team Iteration**
     1. Focuses on all the concept that happen on team level when working in an organization, that applies agile at scale
  2. **Executing Program Increment**
     1. Useful for everybody working in an organization that executes agile at scale.
     2. Also focuses on all the alignment that goes on between teams on a program level

1. **Running a Large Solution at Scale (Advanced Level)**
   1. Focuses on all the alignment that needs to occur when you have an organization that is working on some of largest and most complex products, where the audience are the managers and the coaches within an agile organization
2. **Managing the Delivery Portfolio**
   1. Focuses on all the best practices of managing portfolio of different agile solutions, and coaches and the executive level
3. **Advanced Topics in Scaling Agile**
   1. Aimed at the executive & coach level and focuses on the most complex elements that are present in an agile scaling framework

# About Agile

## Agile Manifesto (Platform)

**Agile Manifesto**

* + **Values**
  + **Principles**

Content of the Agile manifesto caught on and a powerful community was forming that come up with all sorts of different….

* A couple of software programmers came together in Utah & Created Agile Manifesto
* In this manifesto, they came up with a couple of simple but extremely powerful and thought-out values and principles for optimizing the way software should be created

**Content of the Agile Manifesto**

* + The content of the Agile Manifesto caught on and powerful community was forming which come up with
    1. All sorts of different tools
    2. Techniques
    3. Methods
    4. Philosophies

**Most notable of those were the scrum method and Kanban method**



## Agile Philosophy



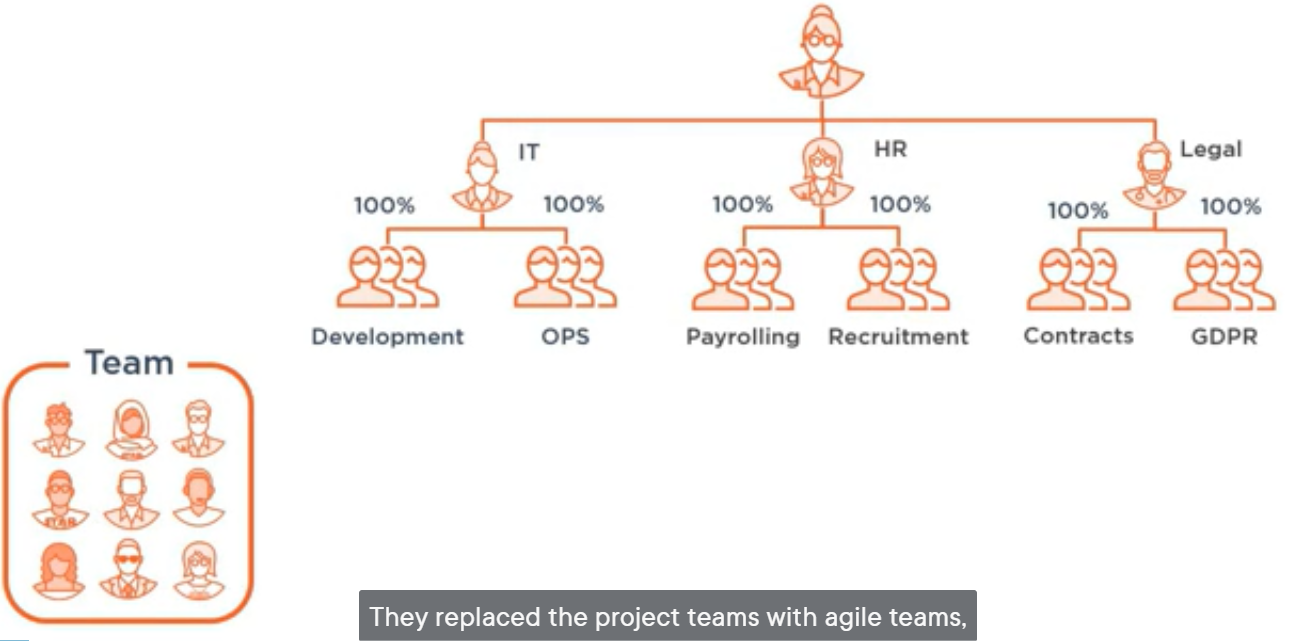
* Based on a couple of things
  + There is the aspect of not taking the organizational structure as starting point, but the team doing the work
  + We take the small team that s all the expertise is needed to create a product end to end

## Agile

1. Easy to embrace the changes
2. Excellent for small products and organizations!
3. The basics are
   1. **Long-lived teams**
      1. That stay together and are stable
   2. **Autonomous teams**
      1. That authority to make decisions
   3. **Independent teams**
      1. conclude the work from end to end
   4. **Frequent Releases**
      1. So value created early and it’s easy to embrace changes
   5. **Close collaboration**
      1. So we know exactly, what customer wants because they are the part of the development process
4. **Time to Market**
   1. Small teams and small organizations, has a very positive effect on the time to market, so that gets way smaller in an agile environment
5. **Productivity**
   1. Productivity goes up
6. **Quality**
   1. The quality of the products increases immensely (Hugely)
7. **Engagement**
   1. Of both the development team and the customer is way higher in an agile environment

* Agile is very effective way of creating software within small teams and for small solutions
* As agile grew, large organization also wanted to adopt this way of working
* But this way of working has a couple of disadvantages, when applied at scale in large organizations

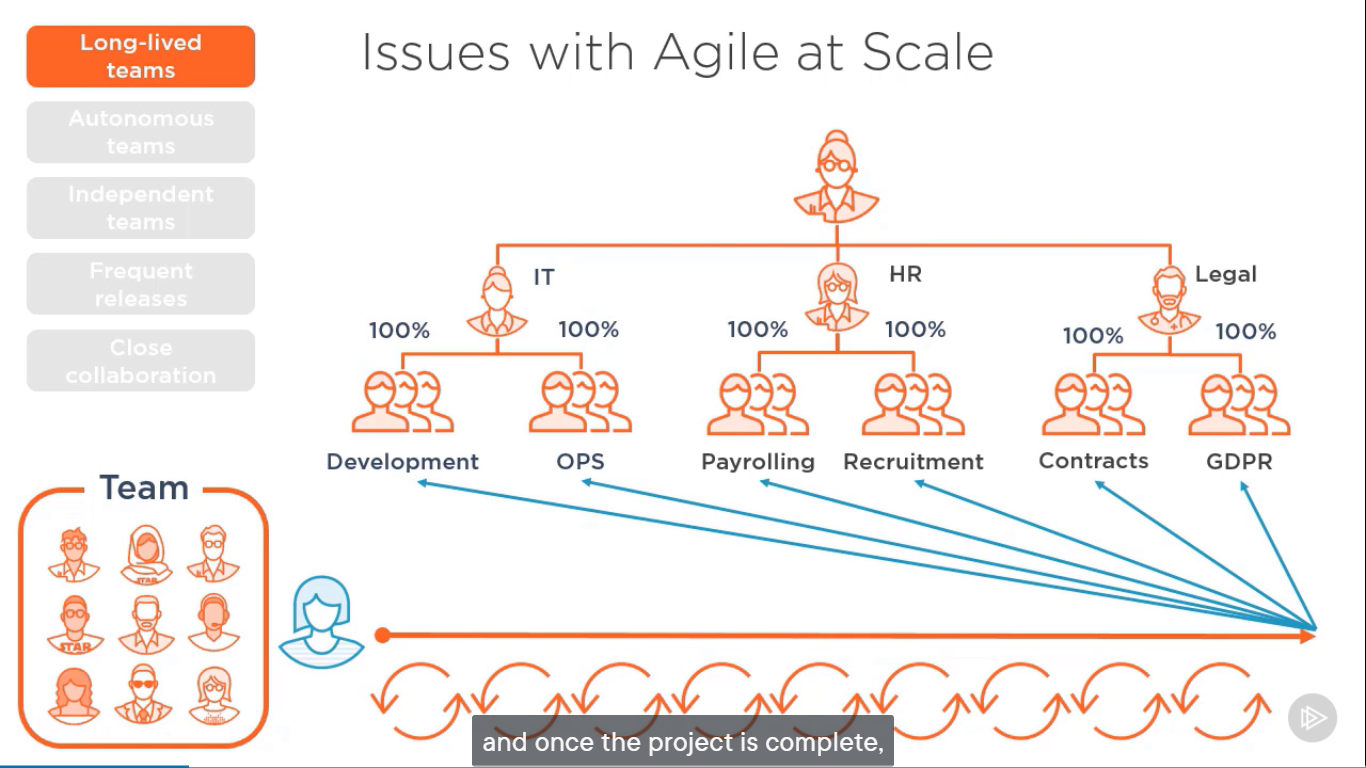
## Issue with Agile at scale



* + Executive in big organization knows what agile was producing great results
  + They do
    1. Replace the project team to agile teams, which look likes above diagram
    2. But it’s creates difficulties with big complex products

**If the product is big and complex**

* + To use agile at its fullest potential, we need
    1. **Long-lived teams**



* + - 1. Organization’s set-up projects with a finite goal
      2. The product exists to create a product or service
      3. Once product is complete, the project team is disbanded, and everybody returns home to their original departments
      4. First aspect, long-live teams, does not really play out well in large organizations that use agile n a team level
      5. As it’s nobody fault, as its just a way that organizations are setup and the managers must make things as efficient as possible within their own silos, not within the projects
      6. So we can say that its not very efficient when working at scale
    1. **Autonomous teams**
    2. **Independent teams**
    3. **Frequent releases**
    4. **Close collaboration**