



ARCHITECTING PROJECT MANAGEMENT

for Enterprise Agility...

July 14 to 16, 2016,

NIMHANS Convention Centre, Bengaluru

Build an Agile Organization not a group of Agile Teams

Organization Design for Enterprise Agility

Paper ID: PMIBC_16_1_006

Author: Mr. Ranan Bhattacharya

CONTENTS

Abstract.....	3
Introduction	3
Details of the paper.....	6
A. Problem.....	6
A.1 How teams work now.....	6
A.2 Lack of systems thinking.....	7
B. Solution.....	8
B.1 Business value based decision making.....	8
B.2 Introduce portfolio, program and team level planning	9
B.3 Change team structure to be driven by program level 'Definition of Done'	11
Conclusion	13
References	15
Annexure	15

ABSTRACT

The benefits of evolutionary iterative development have led to most of the software development organizations adopting Agile. A recent Forrester Research survey reports that 45 percent of software professionals now use Agile methods, and many others use portions of the method.

In reality, however, many enterprises are facing the challenge of not producing the expected outcomes even after adopting Agile. The key reason for this is the failure to optimize the flow of the whole system.

During Agile adoption programs, organizations go through some Agile training and then Agile coaches start working with teams to help them adopt the new methodologies. Over a period of time, the teams become more efficient in implementing Agile methodologies and graduate to becoming self-organized individual entities.

This internal progression is however, not entirely supported by customer value addition. Surprisingly, many customers complain that they still do not get the outcome that they want although the organizations have bunch of highly matured Agile teams. The problem is lack of strategic thinking at the System level. Teams still tend to work in silos and hence overall value integration gets delayed. This happens due to varying priorities of related Features or Stories for different teams which are not aligned to the needs of the end user.

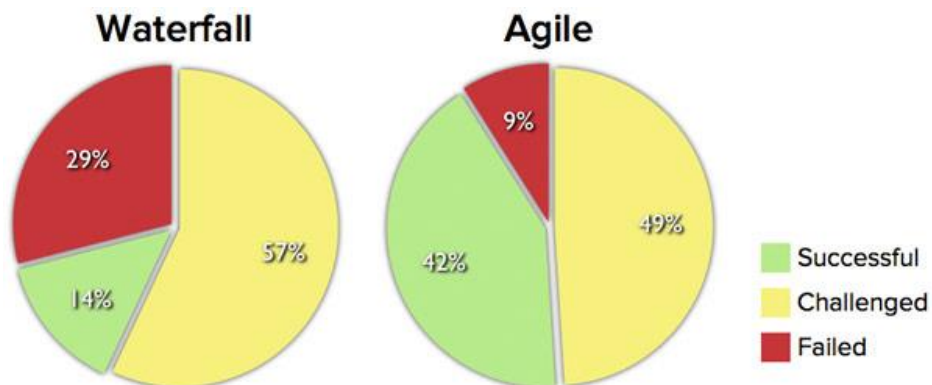
The solution lies in building an Agile ecosystem as a whole, rather than creating a group of teams which do not share a common goal. The Engineering, Product Management and Infrastructure teams should define their common goal based on a Value Stream. Prime importance should be given to drive every deliverable based on value delivered to the customer. It must be emphasised throughout the value delivery chain that individual team successes do not guarantee the best delivery to the customer.

INTRODUCTION

Agile principles talk about integrated cross functional teams working together to deliver value to customer early and often. While closely integrated Agile teams can achieve the best by adopting Agile, a multitude of teams too focussed on using Agile principles to improve only their teams' efficiency will bring in limited success.

During any transformation, people need to appreciate a system focussed view to find out the interdependence of teams, groups and processes. Any action taken by an individual person or team will have effect on the other parts of the system. It's important to build a framework taking into consideration the different interdependent parts of a system so that the actions of all the teams are meant to deliver a common value to the customer.

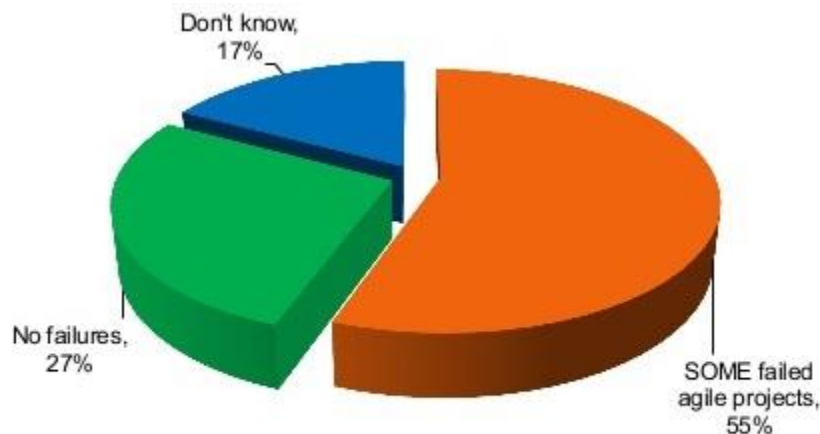
A study done by PM Solutions in 2011 shows that 37% of all the IT projects do not deliver any value and 80% of the projects cost more than they returned value. Another report by the Standish Group shows that there is definitely a visible trend of improvement in the success rate of the projects in the Organization where teams have adopted Agile.



Source: The CHAOS Manifesto, The Standish Group, 2012.

Figure 1

Although the overall trend looks positive, a State of IT Union Survey done in September 2012 found out that 55% of the Agile projects are not delivering the value they aimed for.



Implication: Agile adoption isn't always easy. However, the majority organizations are experiencing success with agile.

Figure 2: DDJ Sept 2012 State of IT Union Survey

This is a clear indication that the way IT programs are run worldwide is still flawed. On further analysis it was found that one of the main reasons for failure is the organization structure. In majority of the Agile organization, teams

work in silos and their goal is focussed on making the team more and more efficient. There are various reasons as mentioned in Figure 3, that cause silos in organizations like incentive systems, task allocation systems and sub-cultures formed based on common skill-set. Agile talks about building cross-functional teams to ensure collaboration between teams on a regular basis.



Figure 3: Problems with Silos

Various scaling approaches are being experimented with industry wide which includes Scaled Agile Framework (SAFe), Disciplined Agile Delivery (DAD) and Large Scale Scrum (LeSS). While these approaches provide great ideas about portfolio or program level thinking and implementation, organizations adopting the new framework find it difficult to implement it fast.

This paper will provide a simpler approach to Agile Integration which will not need complete re-structuring of the organization. It is hoped that this will encourage people to think a little differently and bring in smaller structural changes which will be much less time consuming and easier to implement.

DETAILS OF THE PAPER

A. Problem

A.1 How teams work now

With continuous pressure for faster return on investments, software development organizations are embracing Agile to ensure value is delivered faster. Although teams adopt various frameworks and methodologies, they still struggle to reduce the lead time between the time an idea is conceptualized and the time when the customer receives the functionality for use. Figure 4 below explains the core problem for this delay.

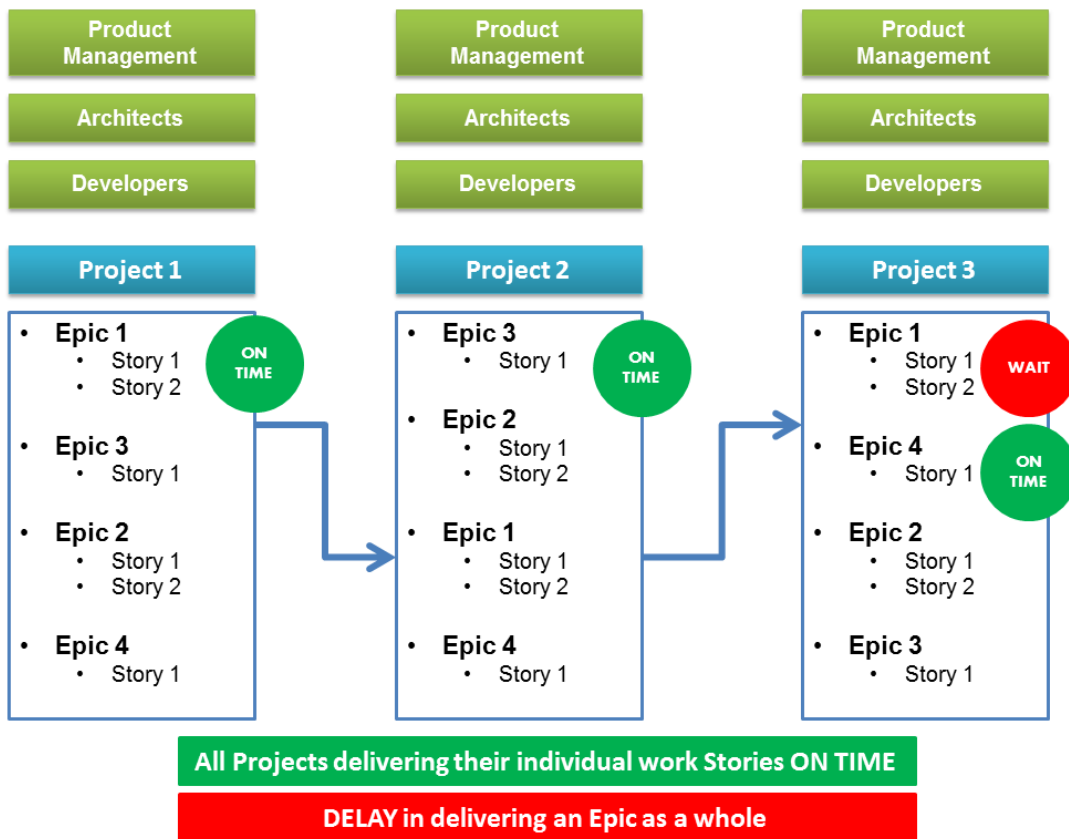


Figure 4

The largest portfolio item in a project hierarchy is Theme (Figure 6). Themes are further split into large blocks of work like Epics and Stories that are assigned to different Project teams. In majority of the cases, the understanding of completion of an Epic or Story is limited to completion of all the linked tasks for a particular team. But completion of a Story does not guarantee delivery of the complete value (Epic) to the customer. Completion of such initiatives depends on completion of all the interdependent Stories across teams. Unfortunately, the partially completed part

of the Epic developed by a project team has to wait for the availability of portions assigned to other project teams for further integration. Since other project teams may be busy working on another part of another Epic, the end user's waiting gets elongated to get the requirement delivered completely.

The teams in this structure are specialist groups with dedicated Product Owners, Architects, Developers and Operation team member who are always focused on the team's delivery as per schedule rather than focussing on building the complete working solution that the end user is expecting.

In a large organization, usually there is always competition between teams for success and recognition. Competing priorities are detrimental for accomplishing the goals of the organization. Every team wants to be successful even at the cost of overall failure. Organizational cultures may encourage collaboration over isolation. Reward system geared towards recognising individual excellence may undermine efforts towards team success.

A.2 Lack of systems thinking

“A system must be managed. It will not manage itself. Left to themselves, components become selfish, competitive, independent profit centers, and thus destroy the system. The secret is cooperation between components toward the aim of the organization.”—W. Edwards Deming,

- Optimizing some components in a system does not bring efficiency rather creates bottlenecks.
- Any system's efficiency is as fast as its slowest part; faster the parts are integrated, risk of failure gets reduced
- Not identifying the dependencies between various components of a system is the reason for delay in most of the IT projects.

Functional teams need always to be aware of the overall flow of value as delivering of value, as opposed to speed of delivery by an individual team, is the primary goal. . There has to be a trade-off between the speed and value depending on how much the speed is affecting the delivery of value.

It's very common to come across people who think their piece of work is the most important. Others can have their budgets cut or schedules changed but not theirs. It's very much part of human nature to be driven by own specialism and concentrate on improving that specialism more and more. Unfortunately, optimising one part of the systems or all the parts individually can never be effective in delivering a goal.

An American car company announced incentives for sales people to sell cars to reduce backlog in their inventory. They forgot to instruct the manufacturing team to stop further production till inventory was reduced. The incentive scheme led to an increase in sales orders which in turn sent a message to the manufacturing teams to produce more. This of course led to a bigger disaster – a further increased inventory.

Lean talks about Muda(waste), Mura(uneven demand) and Muri(overburdening the system). The terms individually are not so important but the concept of how they are related is very important. Uneven production by one unit leads to more bottlenecks and wastages rather than increasing overall efficiency.

B. Solution

B.1 Business value based decision making

All the decisions taken at any level should be aimed to deliver what the business needs. It is important to understand that a program or project exists as long as the business has a requirement. Hence a project's success is entirely dependent on understanding the business need correctly and delivering it when the business needs it.

In order to execute a project successfully, key focus should be on the following areas

- Understand the Business need
- Set the project's priorities based on the need of the Business and not based on the need of the project team
- Continuously involve business for getting early feedback
- Get executive sponsorship to deliver value to the customer

Although technology is an integral part of most of the organizations today, there is still an underlying disconnect between business and engineering. Many engineering organizations still feel that they do not need to understand much of 'Business'. Their job is limited to deliver whatever they are asked to deliver. But as engineering team consumes a significant part of cost and time, it must focus and use its resources towards the organization's overall value delivery to the customer.

'People don't want a quarter-inch drill, they want a quarter-inch hole' – Theodore Levitt

Customers care only about the value delivered to them. Product organizations have tendency to focus more on the deliverables while in reality, customers do not care about. On an iPhone, for example, the touch screen is an attribute. This attribute was designed to make the phone very easy-to-use. Although majority of the customers enjoyed this feature, the elderly customers struggled to use the touch-screens. Binatone later came up with 'Big Button Phone' with brighter display for the elderly customers. Their target customers found this phone much easier to use compared to touch screens. So 'Touch Screen' is not what customers are really looking for, they are looking for 'Ease-of-use'. In this case, if a designer restricted her thought around developing and enhancing touch-screen only, she would have missed the need to address a separate customer base.

Product management should always work with engineering team to ensure that the engineering team works on the items which have the highest **Cost of Delay(CoD)**. Cost of delay helps in setting up the priority of the work items

from the Business user's perspective. CoD is a measure to calculate the loss incurred by not delivering a value on time. Refer to Annexure for detailed understanding of Cost of Delay and ways to calculate the same.

Tesco, one of the largest retailers in the world is the best example of how to drive an organization's vision based on end-to-end value delivered to their customers. Tesco adopted a concept called The BIG 6 to ensure all colleagues in Tesco are enabled, no matter where they work, to take the right actions to help transform Tesco's business and ultimately deliver better for customers. The Big 6 is a balanced scorecard approach which is an industry recognised way of measuring performance. It enables the business to focus on what's important right now as well as on long term goals. The Big 6 balances what the business needs to do for customers, colleagues, suppliers and communities, with what needs to be done to improve its financial health. These six measures are key to the organisation and it makes things simpler for colleagues. The Big 6 will also tell the business quickly if what it's doing is working.



Figure 5: Tesco's Big 6

B.2 Introduce portfolio, program and team level planning

When an organization has multiple teams working in different parts to finally produce a working solution, coordination between teams to achieve focussed value delivery goals becomes challenging. The most common scenario is that teams forget about the value chain and instead optimize themselves. The unintended consequence of this is that teams will not be in sync with each other until the need of integration becomes unavoidable. During that period teams might have developed parts which are not compatible with each other or teams might have duplicated effort.

Like a chariot, a central force is required to pull all the horses to channelize their energy to a common direction. A bunch of super performing horses can jeopardize everything if they start running in different direction. Similarly in a complex product development environment it is very important to prepare a centralized plan which will align all the different parts towards a common direction.

A portfolio based strategic model best addresses this problem where the organization defines the strategy at a centralized level but execution happens in a decentralized manner. A strategic theme at the top represents a goal over a longer period of time. The lower level of hierarchies like Epics and Stories represent more discrete items with market value. The Portfolio, Program and Projects define the value stream which ensures delivery of value from an idea or concept. Figure 6 explains the strategic model which defines the sequence of activities in a system consisting of various specialized teams.

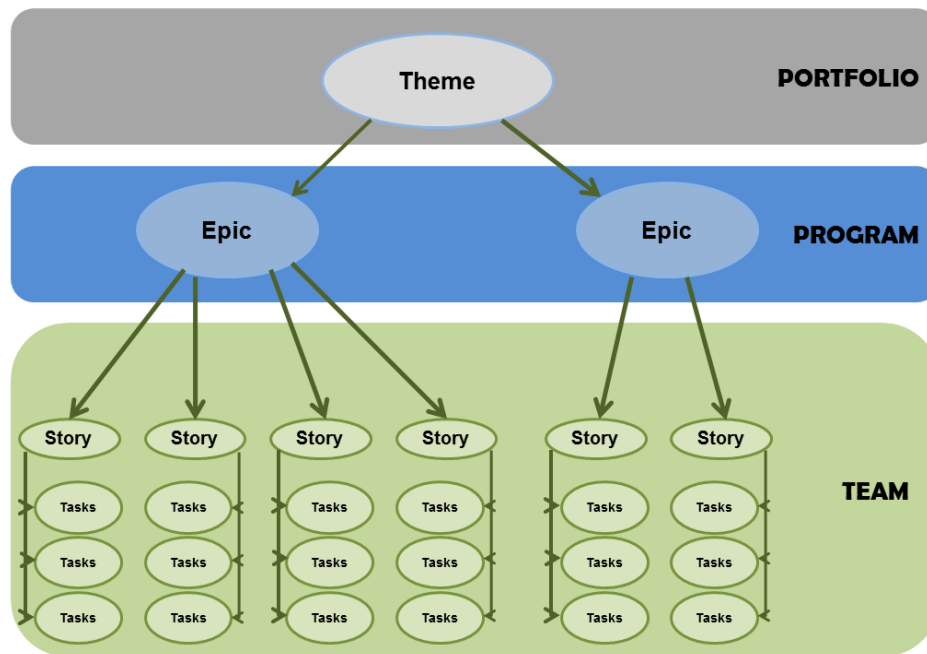


Figure 6

Theme – This is the top level objective or vision for a group or division in an organization

Epic – A group of related stories that defines a high level functionality needed to deliver the Theme

Story – An Independent, Negotiable, Valuable, Estimable, Small and Testable (INVEST) piece of work

Tasks – Lowest level of unit of work for the teams to work on

Program Manager is the central role for governance and management for a Theme and the related value stream to deliver the theme. The detailed responsibilities of the Program manager will be discussed in section B.3

B.3 Change team structure to be driven by program level ‘Definition of Done’

Earlier we have seen the problem of creating independent project teams which work in silos and concentrate on a team’s Definition of Done. Now think of a situation in which these project teams come together and work collectively as a program to deliver a complete working solution. In this structure, each functional group allocates the needed people to one project. These team members form the cross-functional project team to produce working software iteratively.

In this scenario, Product Management, Engineering and DevOps are always in sync about the common business goal and maintain same priorities for the interdependent deliverables.

“Business people and developers must work together daily throughout the project.” – Agile Manifesto

The following illustration (Figure 6) depicts this structure and shows how a set of integrated skills within a team could provide business value.

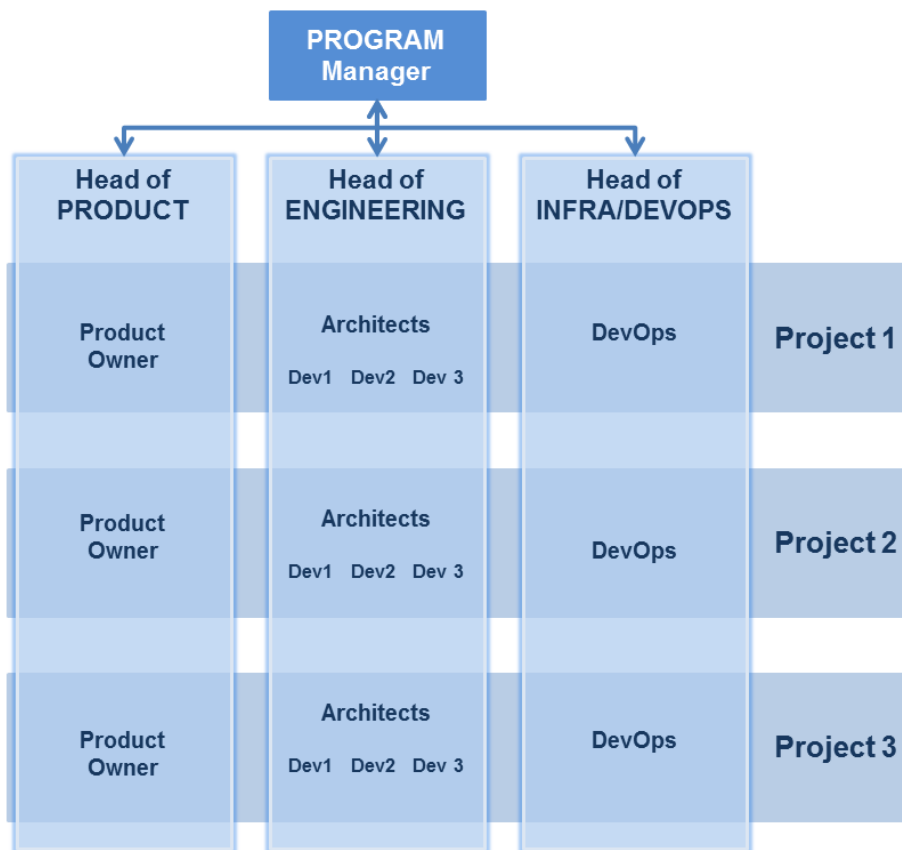


Figure 7

The Agile Program management should not follow the practices of conventional PMO world. Instead of incorporating a centralized and controlled environment, the **Agile Program Managers** should focus on decentralization of decision making. The conventional PMO practices were based on centralized decision making and detailed up-front project planning. In this model, the Agile Program Managers operate more on rolling wave planning; lightweight Epic based business cases and stays focussed on continuous value flow. Their core responsibility will be to define the value streams consisting of Product Owners, Architects, Developers and DevOps people. This will ensure that all the needed teams are in sync to develop a solution which will deliver the desired value.

Architects, Product Owners and the DevOps engineers should be involved at the program level from the beginning. It is counter-productive when these people work in isolation at the project or team level. The organization structure should support cross-functional teams. Each project team responsible for contributing towards the Program level initiatives should consist of representatives from Product Management, Solution Architecture, DevOps and Engineering teams. All the members of a specialized team like Product Management or Solution Architecture should collaborate regularly with other team members of their own specialized team to identify the dependencies and integration points with other project teams. Lack of collaboration between these people from the beginning causes huge integration issue due to lack of understanding of the requirement and incompatible architecture.

Roles in the new team structure

A **Head of Product's** involvement is crucial during the process of setting up priorities at the Program level. Priorities of work should be set based on Cost of Delay and not based on any subjective means of assessment. With inputs from System Architects or Head of Engineering, the Head of Product has to develop the Product Backlog. He will have a group of Product Owners from different teams working with him to set priorities of work items at the team level. The **Product Owners** are the proxy customers for any team. They should guide the development teams to deliver what is valuable and meaningful for them. They are responsible for ensuring that the priorities of the initiatives are set in the right context of the end users. The Product Owners from different Projects or Teams should always work together to ensure that the linked items are assigned same priorities across teams.

The **Head of Engineering** is responsible for the complete development of the solution. The Solution Architects along with him have to be involved with the Product Management to design the architecture of the solution. Many a time, the priorities of the work items change due to technical dependencies and challenges. It is always important to find out such dependencies and plan to address them early in the cycle. Similarly, it's key to understand and create work items related to Non-Functional requirements (NFR). Product Owners and Architects should be responsible for identifying and deciding the complete list of NFRs.

The **Architects** should not only be decision makers but should be part of the engineering team who will work with the developers to build the solution. Architects are significantly more effective when they are part of the program

team. They should work closely to have a common understanding of the requirements and thus build right architecture and strategy.

In today's world **DevOps** has become an integral part of the engineering teams. It's very important that the teams have dedicated involvement of DevOps engineers in the team right from the beginning. Lot of times organizations tend to ignore the need to operations when they prepare the strategy and implementation plan. Any Agile Program should setup the right engineering processes from the beginning to achieve highest level of efficiency. Lack of automation and too much dependency on manual intervention during deployment and testing generate bottlenecks and increases wastage. In larger organizations still we may see Infrastructure and DevOps as a horizontal service. Ideally the Definition of Done of teams must include DevOps related aspects like validation of code, automated integration testing, continuous deployment etc.

CONCLUSION

Dan Ariely, author of *Predictably Irrational* suggests that people make irrational decisions under the influence of emotion. It is important for an organization to improve decision making abilities in emotion provoking situations. The sense of ownership sometimes makes people emotionally driven to perceive the value of a process or work to be much higher than actually they are. It is important for any organization to create an environment where people give more value towards overall achievement rather than own achievement.

There are already various frameworks which people are talking about to create a perfect Agile organization. Most of these frameworks are complex to develop and time consuming as they involve re-organizing the company structures. All of them suggest of creating new roles to take care of various portfolio and program level responsibilities.

The recent "State of Agile Report" by VersionOne indicates that 72% of the organizations still use Scrum of Scrums for implementing Agile at scale for the simplicity of implementation.

SCALING AGILE

Scaling Methods and Approaches

The majority of respondents continue to use Scrum/Scrum of Scrums to help scale agile within their organizations, but SAFe saw the largest increase from 19% in 2014 to 27% in 2015 making it the second most prevalent scaling method cited by respondents.

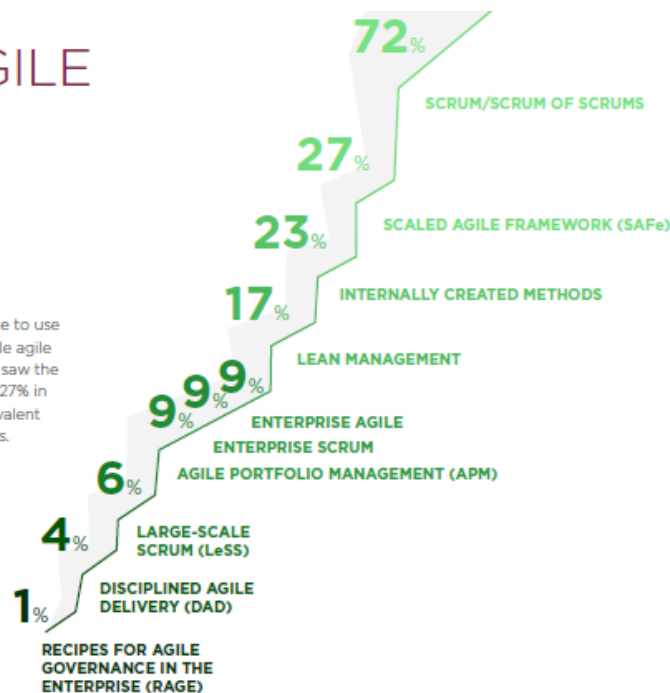


Figure 8: 10th State of Agile Report by VersionOne

The model described in this paper is simpler to implement Portfolio and Program level strategies using existing roles and structure. It is more about making changes in the alignments of the cross functional departments more closely. Moreover, it defines the need and mechanism of collaboration among the key representatives from different functional groups from the beginning.

Organizations may still struggle to implement this as it will require breaking the existing silos and forcing people to put Portfolio level goals above team level goals. Heads of Engineering may still argue that it is not their responsibility to attend the meetings where Portfolio Backlog will be created or Heads of Product may deny taking part in the discussion to decide a suitable architecture. They will be more inclined to continue in their comfort zone of doing things separately in their own silos and integrate it later.

In this fashion, programs will continue to fail to deliver early and optimize the whole flow. The suggestion will be to try this model to address the issues of delayed integration, wastage due to waiting and rework, uneven flow in the complete system and hence delayed delivery of value to the customer.

REFERENCES

1. Donald Reinertsen, "The principles of Product Development Flow", Celeritas, 2009
2. Peter Senge, The Fifth Discipline, Currency, 2006
3. www.scaledagileframework.com
4. www.scrumalliance.org
5. www.valueflowquality.com by Emergn

ANNEXURE

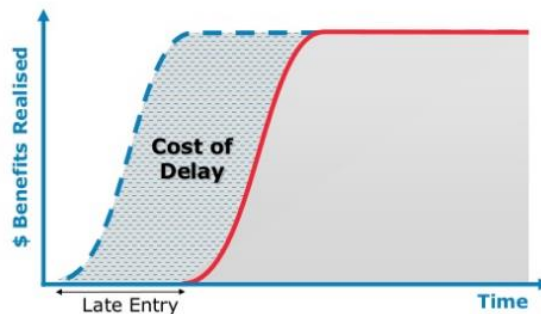
Calculating the Cost of Delay

$$\text{Cost of Delay} = \text{User-Business Value} + \text{Time Criticality} + \text{Risk Reduction and/or Oppty Enablement}$$

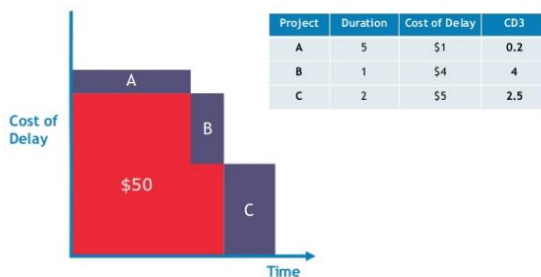
User-Business Value: Do our users prefer this over that? What is the revenue impact on our business? Is there a potential penalty or other negative impact if we delay?

Time Criticality: How does the user/business value decay over time? Is there a fixed deadline? Will they wait for us or move to another solution? Are there **Milestones** in the critical path impacted by this?

Risk Reduction-Opportunity Enablement Value: What else does this do for our business? Does it reduce the risk of this or a future delivery? Is there value in the information we will receive? Will this feature open up new business opportunities?



FIFO Queuing method
First In First Out



CD3 Queuing method
Cost of Delay Divided by Duration

