Does DAD Know Best, Is it Better to do LeSS or Just be SAFe? Adapting Scaling Agile Practices into the Enterprise

Aashish Vaidya
aashish.vaidya@cambiahealth.com

Abstract

Organizations, large and small, that are experimenting and succeeding in using agile practices at team level, face their next challenge in scaling these practices across the enterprise. This challenge can come from expanding pilot programs from handful of teams to more teams; or, it comes from agile teams working with non-agile parts of the organization. The process inevitably creates confusion as teams employing different methods interface with each other. Introduction of agile practices challenges existing structures and practices, and brings forth questions regarding traditional roles, responsibilities and expectations. Should an organization continue to hone team level practices or should it try to extend agile practices to other parts of the organization? Are there large scale practices that work in easing the interaction between agile teams, and other non-agile, semi-agile business units? While scaling, can an organization get too process heavy and risk losing the original intent of transitioning to agile practices?

There are primarily three scaling frameworks that try to address scaling agile practices: Disciplined Agile Delivery (DAD), Large-Scale Scrum (LeSS), and Scaled Agile Framework (SAFe). Each of these frameworks draws from variety of agile and lean practices. However, an organization’s context matters most in deciding whether to embrace particular framework or only select practices to obtain desired results. In most cases, an organization has to make informed, pragmatic choices and experiment with various practices in order to address its specific needs without losing sight of why it embarked on an agile transition and transformation journey in the first place.

Cambia Health Solutions has for over three years rolled out Scrum and other agile practices across more than 40 development oriented teams. We also keep apprise of scaling models and newer practices to better understand our own implementation. For Cambia, scaling practices, articulated in scaling frameworks, such as enterprise-wide synchronized Sprints, multi-program Quarterly Release Planning, Scrum of Scrums and Communities of Practices, have been effective in organizing our work. Many of these practices can prove useful not only to larger enterprises like Cambia, but also to smaller ones, who are looking to improve and refine their adaption of agile practices.

Biography

Aashish Vaidya is a Staff Consultant at Cambia Health Solutions, a not-for-profit, health solutions company. He works on large portfolio projects and serves as an internal Agile and Quality coach. He is a founding member of Cambia’s Enterprise Transition Community, Agile Best Practices Exchange and other Communities of Practice. Aashish has over 20+ years of industry experience working in various leadership and management positions for companies such as Compaq, Intel, Kronos Incorporation. Aashish has been a presenter and a panelist for Technology Association of Oregon, AgilePDX, RoseCity SPIN and PNSQC. An agile practitioner for over 7 years, he holds CSP, CSM and SAFe certifications. He is a co-author of 2 articles on Agile Atlas on Scrum common practices and maintains an infrequent blog on Agile and Management topics at agilesutra.wordpress.com. Aashish has a Bachelor of Science degree in aerospace engineering from Texas A&M University.
1. Introduction

Almost three years ago, the IT department at Cambia Health Solutions transitioned its Software Development Life Cycle (SDLC) from traditional waterfall methods to agile practices. Cambia Health Solutions is a not-for-profit total health solution company. The Cambia IT department is a shared services group, which provides software development, integration and other IT services to Cambia’s two major divisions: Health Insurance Services (HIS) and Direct Health Solutions (DHS). Health Insurance Services is Cambia’s mainline, regulated health insurance business. Whereas, DHS is comprised of various companies that provide affiliated healthcare solutions. Just like many health care companies, Cambia is facing rapidly changing market forces including meeting the challenges of ever increasing health care costs, and once in a generation reforms in the form of the Affordable Care Act (ACA). Cambia added two new corporate values of agility and innovation as a reminder of these challenges. Transitioning to agile practices was in a direct response to prepare Cambia for meeting market challenges and achieving its business objectives.

All of Cambia’s development teams, as well as a few IT operational and other shared services teams, have adopted Scrum and Kanban for their SDLC about three years ago. Our experiences in leading transition efforts and team level practices were outlined in a 2012 PNSQC paper (Vaidya and others, 2012). Along with Scrum and Kanban team practices, we have employed various other practices that extend beyond teams and help us interact on larger, more complex projects. These practices also help agile teams interface with non-Agile units of the organization.

2. Agile Scaling Frameworks

At Cambia, when we changed our SDLC to agile approach, we adopted Scrum as our primary agile method. Many agile methods provide specific practices that inform team level practices. However, Agile methods generally do not provide specific guidance for wider organizational practices such as the front end process of software development such as business case approval, budgeting, project onboarding or the back end of interacting with partners and vendors, having the product ready for deployment to users, and project benefits realization.

For instance, Scrum is the best known and most used agile method in the industry. As defined by Scrum Alliance, Scrum specifies a minimal set of practices. It calls for three roles on a Scrum team – the Product Owner (PO), a ScrumMaster (SM) and a cross-functional team. It specifies each team to manage its work through four artifacts: a Product Backlog, a Sprint Backlog, a Product Increment and Definition of Done. In order to achieve its goals, Scrum specifies five activities for teams: Backlog Refinement, Sprint Planning, Daily Scrum, Sprint Reviews and Sprint Retrospectives (Agile Atlas). Beyond these roles, artifacts and activities, however, Scrum makes no specific recommendations in scaling beyond the team level. It expects an engaged leadership to provide broader vision and goals, and empower self-organizing agile teams. It expects practices to emerge naturally that will help organization continue its journey towards better agility.

Similarly, Kanban is even less specific in its required practices. In most Kanban implementation, generally two practices are specified. One is a visualization of workflow, generally through a Kanban board. Another practice is imposing a work-in-progress limit for every step in the workflow. David Anderson who articulated the Kanban method adds three other practices: measure and manage flow, make process policies explicit and use of models to recognize improvement opportunities (Anderson, 2010, Loc 619 of 4729). Just like Scrum, Kanban relies on self-organized teams and highly involved leadership in devising other practices that help the adoption of Agile into the organization.

Three years back, Cambia’s Enterprise Transition Community was tasked to help rolling out agile practices to the organization. Part of the transition effort involved not just addressing team level practices, but also aligning 40+ agile teams to work together with rest of the organization that still used traditional methods. At the time, our primary scaling framework reference model was Scaled Agile Framework (SAFe). Since then we have also referenced two other scaling frameworks – Disciplined Agile Delivery (DAD) and Large Scale Scrum (LeSS). All three of these frameworks try to address...
practices beyond the team level. These frameworks use combination of practices based on various Agile and Lean concepts. Many proponents claim that an organization must adopt a scaling framework in its entirety to reap the benefits. At Cambia, instead, we have taken a more pragmatic approach of adopting, blending, and modifying practices that suit our context, and help us meet our business objectives. We reference these scaling frameworks to understand our current implementation and examine whether adoption of additional practices will help us make continuous improvements and solve specific issues.

Before I outline Cambia’s scaling practices beyond the team level, let us review these three scaling frameworks and their approaches to roles, processes, and other salient features. This will provide some context on practices we follow at Cambia as compared to those advocated by the scaling frameworks. To provide a frame of reference, I will also compare and contrast the key features against standard Scrum. Also, note that these frameworks are continually updating their practices and as such, the information below may have changed since the time of this writing.

### 2.1. Disciplined Agile Delivery

Disciplined Agile Delivery is developed by Scott Ambler and is an attempt to “fill in the process gaps that Scrum purposely ignores.” DAD is a “hybrid approach which extends Scrum with proven strategies from Agile Modeling (AM), Extreme Programming (XP), Unified Process (UP), Kanban, Lean Software Development, Outside In Development (OID) and several other methods” (Ambler, 2013). DAD teams “focus on producing repeatable results, such as delivering high-quality software...[but] do not strive to follow repeatable processes” (Ambler, 2012).

#### 2.1.1 Roles

The DAD framework starts with Product Owner and primary Team Members roles, just like Scrum. In addition, DAD introduces a role called Team Lead, which is akin to ScrumMaster in Scrum and an Architecture Owner role. There are other secondary roles like Specialist, Independent Tester, Domain Expert, Technical Expert and Integrator. These secondary roles are introduced to address scaling issues and can be deployed on a temporary basis. The Product Owner and primary Team Members roles are similar to Scrum (Disciplined Agile Delivery).

#### 2.1.2 Practices

The DAD framework employs four distinct lifecycles and expects an organization to employ these lifecycles to suit its needs (Ambler, 2013).

The first lifecycle, called Agile/Basic, illustrated in Diagram 1, has three phases called the Inception, Construction and Transition.

The Inception phase is meant for “lightweight visioning activities to properly frame the project”, meant to be brief for one or more short iterations where initial requirements and Release Plan are worked out. This may include initial modeling and architectural visioning. The Construction phase follows the contours of typical Scrum iterations. Similarly, the Transition phase could be one or more short iterations and is used for deploying the solution into production. Daily stand-ups, iteration planning, review and retrospective remain the same as Scrum. In DAD, a work item list is slightly different from a Scrum Product Backlog, since it includes not just requirements and defects, but also, “other non-functionality oriented work such as training, vacations, and assisting other teams” (Disciplined Agile Delivery).
Diagram 1: Agile/Basic Lifecycle (Credit: Disciplined Agile Consortium)

The second lifecycle, called Advanced/Lean, as depicted in Diagram 2, is closer to Kanban in its Construction phase. Just like the Agile/Basic lifecycle, the Inception phase is used to stock a work item pool. This work item pools are organized along business value, fixed delivery date, expedited or some other intangible categories. Planning, retrospectives, demos, stand-ups and other activities are performed as needed rather than following a scheduled cadence as followed in the Agile/Basic lifecycle. This includes deployment readiness during the Transition phase. Some upfront architectural modeling and visioning similar to Agile/Basic lifecycle is done during the Inception phase.

Diagram 2: Advanced Lean Lifecycle (Credit: Disciplined Agile Consortium)

The third lifecycle is called Continuous Delivery lifecycle, which is a “leaner” version of Advanced/Lean lifecycle, illustrated in Diagram 3, below. It does away with an explicit Inception phase and has very short Transition period. In this lifecycle, a “product is shopped into production or the marketplace on a very regular basis…as often as daily,” but weekly and monthly is common too (Disciplined Agile Delivery).
Diagram 3: Continuous Delivery Lifecycle (Credit: Disciplined Agile Consortium)

The last or the fourth lifecycle is called the Exploratory lifecycle, shown in Diagram 4, which is meant for “agile or lean teams that find themselves in startup or research situations where their stakeholders have an idea for a new product but they do not yet understand what is actually needed by their user base.” This lifecycle consists of six activities such as envisioning, prototyping, deploying, observe and measuring and based on the feedback canceling or productizing the idea (Disciplined Agile Delivery).

Diagram 4: Exploratory Lifecycle (Credit: Disciplined Agile Consortium)
2.2. Large Scale Scrum (LeSS)

Craig Larman and Bas Vodde articulated Large Scale Scrum “to apply Scrum to very large, multisite, and offshore product development.” They define LeSS to apply to a median implementation that covers, “around 800 people on one product at 5 sites, with about 15 million lines of source code” (Larman & Vodde, 2013). LeSS framework specifies organizational changes, which isn’t directly addressed in standard Scrum. LeSS specifies cross-functional, cross-component, end-to-end feature teams through the elimination of traditional team lead and project manager roles. In an article outlining a case study at J.P. Morgan, the feature teams each had a “blend of domain, technical and functional skills.” (Larman & Winn, 2014)

The LeSS framework recommends two frameworks, one that covers up to ten Scrum teams (called LeSS framework-1) and another that has more than ten teams (called LeSS framework-2). This suggests that framework-1 should be used for up to 100 people, since it also recommends a team size of up to ten team members. In framework-2, scaling is accomplished using sets of framework-1 groups.

2.2.1 Roles

In LeSS framework-1, a single Product Owner is common to all ten teams, whereas, in framework-2, an Area Product Owner role is introduced that covers a specific product area. An Area PO covers 3+ Scrum teams and specializes on one requirement area (Larman & Vodde, 2013). No other special roles are specified compared to standard Scrum.

2.2.2 Practices in Framework 1

Sprint Planning meeting changes in framework-1, illustrating in Diagram 5. For Sprint Planning, each of the Scrum teams sends two members per team (as opposed to entire Scrum team participating in standard Scrum) plus the one overall Product Owner to decide on which chunk of Product Backlog items to work on. Product Owner arbitrates between teams when a backlog item is in contention. Similarly, Sprint Review changes to a single meeting for all Scrum teams, but is limited to two team members per each Scrum team (Larman and Vodde, 2013).

Diagram 5: LeSS framework-1 (Larman and Vodde, 2013)
In addition to these changes, LeSS introduces three more practices: Inter-team coordination meeting, a Joint Light Product Backlog Refinement and Joint Retrospective meeting. The Inter-team coordination
meeting can be held frequently during the week and it can use various format, including an “Open Space, Town Hall Meeting, Multi-Team Daily Scrum or Scrum of Scrums, to increase information sharing and coordination.” The Joint Light Product Backlog Refinement has a maximum duration not to exceed 5% of the Sprint duration. It is also restricted to two team representatives. In this meeting, the team looks to refine product backlog items for upcoming Sprint. Lastly, a Joint Retrospective is added, which is attended by team Scrum Masters and one representative from each team. They jointly “identify and plan improvement experiments for the overall product or organization” (Larman and Vodde, 2013).

LeSS framework-1 also has an optional practice called In-Sprint Item Inspection where the teams “informally seek out early feedback from the PO or other stakeholders on finished Product Backlog items as soon as possible during the Sprint.”

2.2.3 Practices in Framework 2

All framework-1 meetings continue for each of the product area as outlined in framework-1 practices. In framework-2, as shown in Diagram 5, a new Pre-Sprint Product Team Meeting is introduced. This meeting is held prior to Area Sprint Planning meeting outlined in framework-2 practices, with all Area POs and the overall Product Owner, in order to focus on “product-level rather than area-level optimization”. (Larman and Vodde, 2013) In addition, an overall Sprint Review meeting at the product level is introduced which only focuses “on a subset of interest to the overall PO or to many Area POs.” Lastly, an Overall Sprint Retrospective meeting at the product level is introduced. This meeting “happens earlier in the subsequent Sprint, after area-level Joint Retrospectives.” (Larman and Vodde, 2013).

Diagram 6: LeSS Framework-2 (Larman and Vodde, 2013)
2.3. Scaled Agile Framework (SAFe)

The Scaled Agile Framework is created by Dean Leffingwell along with other collaborators, is illustrated in Diagram 7. SAFe framework articulates three levels of organization: Team, Program and Portfolio. Each level has its own activities and all levels are tied together. SAFe incorporates agile and lean practices at all three levels (Leffingwell, 2011, loc 1261-1315). SAFe provides team and program size patterns, which can then be used for scaling across larger organization. SAFe specifies standard Scrum team size consisting of five-nine team members. A program is defined as consisting of persistent five-twelve agile teams or 50-125 individuals that are dedicated to the program and capable of delivering business capability or value.

![Diagram 7: Scaled Agile Framework Big Picture (Scaled Agile Framework)](image)

2.3.1 Roles

At the team level, an agile team continues to resemble a typical Scrum team with some variations. A team has a ScrumMaster, which could “be part-time role for a team member (25-50%), or a single ScrumMaster may be shared across 2-3 teams” (Scaled Agile Framework). The team, referred to as ScrumXP, continues to have Product Owner and a team of five-nine team members, just like standard Scrum. A ScrumXP team can be a specialized component team and does not have to be broadly cross-functional or a feature team. The collection of ScrumXP teams coordinates with each other to develop and deliver cohesive end-user value. However, a ScrumXP team must have capability to design, build and test its own work.

At the Program level, several new roles and teams are created. A Product Manager role serves “as the ‘content authority’ for the release train and is responsible for defining the prioritizing the Program Backlog, and working with Product Owners to optimize Feature delivery” (Scaled Agile Framework). Product Manager directs the work of Product Owners at the team levels. A dedicated System Architect role is present to perform some up-front architecture and guide the emergent architecture for all program
teams. Just as a Product Manager is a “chief Product Owner” for the program, a Release Train Engineer role serves as a “chief ScrumMaster”. The Release Train Engineer “facilitates program level processes and program execution, escalates impediments, manages risk, and helps drive program-level continuous improvement” (Scaled Agile Framework). A User Experience or UX designer provides “cross-program design guidance so as to provide a consistent user experience across the components and systems of the larger solution.”

SAFe also has additional program level teams, apart from the individual roles mentioned above. A Business Owner team comprising of three-five stakeholders have “the ultimate fiduciary, governance, efficacy and ROI responsibility for the value delivered by a specific release train” (Scaled Agile Framework). A Release Management Team (RMT) is generally manages across one or many products lines, the responsibility for scheduling, managing and governing of synchronized releases (Scaled Agile Framework). A DevOps team “provides tighter integration of development and operations” and maintains deployment readiness for the program. A System Team is “responsible for providing assistance in building and using the development environment infrastructure – including Continuous Integration, build environments, testing platforms and Test Automation frameworks – as well as integrating code from Agile Teams, performing end-to-end system testing, and demonstrating solutions to stakeholders at each iteration” (Scaled Agile Framework).

At Portfolio level, there is a Program Portfolio Management team that “represents the highest-level fiduciary (investment and return) and content authority (what gets built)....” This team comprises of “business managers and executives who understand the enterprise business strategy, technology, and financial constraints....” This team has the responsibility of providing portfolio vision, strategic and investment funding and overall portfolio governance (Scaled Agile Framework).

### 2.3.2 Practices

At the team level, SAFe specifies a blend of Scrum and Extreme Programming (XP) practices. The most significant departure from Scrum is that SAFe specifies practices surrounding code quality or agile software engineering practices mainly derived from XP. The code practices include Agile Architecture, Continuous Integration, Test-First, Code Refactoring, Pair Work, and Collective Code Ownership. The other significant change from Scrum is that SAFe does not expect teams will produce Potentially Shippable Increment (PSI) every Sprint, but rather over a quarterly cadence. At the program level provides features, which the teams deconstruct and size to fit into iterations.

SAFe employs an Agile Release Train (ART), at the program level, in order to develop large-scale systems. An ART comprises of four-two weeks iterations followed by a three-week HIP (Hardening, Innovation and Planning) iteration. Teams are dedicated to the ART and they all synchronously develop and release a PSI on the same quarterly cadence. A system team generally operates iteration behind the individual ART teams. The system team integrates code from various ART teams and performs “end-to-end and system performance testing” of features, either manually or through test automation. The system team also helps stage a “system Sprint demo”, where teams showcase the whole system to the stakeholders. At the end of every quarter, during the HIP iteration, all teams within the ART plan the next PSI during an all-hands Release Planning meeting. During execution of a PSI, the ART also organizes a twice a week Scrum of Scrum for all the dedicated ART teams.

At the Portfolio level, SAFe introduces concepts of investment themes and values streams that align the ARTs at the program level. A value stream is defined as a “long-lived series of system definition, development and deployment process steps used to build and deploy systems that provide a continuous flow of value to the business, customer or end user” (Scaled Agile Framework). In SAFe parlance, a value stream is realized through an Agile Release Train at the program level. Investment themes “reflect how the portfolio allocates budget to the release trains that implement the portfolio business strategy” (Scaled Agile Framework). These themes in turn act as a funnel, seeding a portfolio backlog with business or technical epics. There are two types of Epics in SAFe – Business and Architectural. Business Epics are “large, typically cross-cutting customer-facing initiatives that encapsulate the new development necessary to realize certain business benefits” (Scaled Agile Framework). And Architectural
3. Program and Portfolio Level Practices at Cambia

Agile methods such as Scrum, Extreme Programming (XP), Kanban, offer practices that generally start with the team. Many of the Agile methods expect that through self-organization, truly cross-functional, cross-component teams will be able to deliver business value quickly on a cadence, with high quality. They expect these teams to respond to changes and work at a sustainable pace over a long duration. So, for companies that have transitioned to agile practices for their SDLC but still largely employ traditional PDLC and portfolio management techniques, these methods don’t provide much concrete guidance. In many smaller organizations, where formal structures and processes do not exist, or where structures are less hierarchical, adoption of agile practices can happen quickly and with relative ease. However, in larger organizations where there are Enterprise Program Management Office or Portfolio Management structures, introduction of practices at team levels leads to inevitable issues with existing norms, and processes.

Jim Highsmith, writing for the Agile Alliance, anticipated these culture issues at the initial adoption of the Agile Manifesto over a decade ago:

*But while the Manifesto provides some specific ideas, there is a deeper theme that....people who held a set of compatible values, a set of values based on trust and respect for each other and promoting organizational models based on people, collaboration, and building the types of organizational communities in which we would want to work. At the core, I believe Agile Methodologists are really about “mushy” stuff about delivering good products to customers by operating in an environment that does more than talk about “people as our most important asset” but actually “acts” as if people were the most important, and lose the word “asset”. So in the final analysis, the meteoric rise of interest in and sometimes tremendous criticism of Agile Methodologies is about the mushy stuff of values and culture (Highsmith, 2001).*

For transitioning or transforming organizations such as Cambia, Agile practices at team levels inevitably leads to larger questions about organizational design, command-and-control hierarchical management, HR and management practices and overall company culture. The three scaling frameworks described above provide approaches that attempt to address some of the issues that an organization faces and offer solutions to address these gaps. However, each framework provides some benefits, but they have shortcomings as well.

For example, DAD creates four distinct lifecycles, each of which an organization can adapt to fit its context. However, it also specifies an overly complicated work items pool, which an organization can address in much simpler ways. Further, DAD introduces several temporary secondary roles, which can hamper agile teams from becoming cohesive, self-organizing units capable of producing end-to-end features.

LeSS starts where Scrum leaves off when it comes to scaling agile practices in large organization. However, in the process, it makes recommendations that are problematic, like having a single Product Owner for up to ten teams. For a transitional organization, the business units are generally not accustomed to interacting with software development from the inception of the project to production delivery. With a Product Owner responsible for so many teams, in turn shortchanges the agile teams on the much-needed interaction with their business users and partners on a regular basis, throughout the project cycle.

Similarly, SAFe organizes its practices into three levels: team, program and portfolio, which is quite useful for larger organization. At a team level, it embraces certain XP practices, which standard Scrum does not. However, the framework has myriad of issues, including being overtly process heavy.
At Cambia, we have not embraced any particular scaling framework on a wholesale basis. None of the scaling frameworks completely aligns with our business needs. Instead, we have chosen few practices to help meet our needs. We also look for opportunities to refine and experiment with new approaches, make continuous adjustments as we obtain more information about the effects of these changes on the organization. This has allowed us to realize incremental benefits without putting the organization through excessive change management churn.

At a high level, Diagram 8 shows Cambia’s approach to scaled Agile and program level organization.

Diagram 8: High Level View of Cambia’s Scaled Agile Approach

Multiple projects and multiple agile teams stay aligned and coordinated with the use of synchronized quarterly planning and Sprint schedules (see Diagram 9 for visual representation of the process).

Diagram 9: Cambia Synchronized Quarterly and Sprint Process for Program and Team Alignment

Below is a discussion of some of these practices and how they are currently practiced at Cambia. Wherever, appropriate, I will highlight practices that the scaling frameworks specify as well.
3.1. Synchronized Enterprise Sprint Schedule

With the change in SDLC to agile approach, we broke away from traditional IT functional silos and re-organized into cross-functional agile teams, with representation from at least development, testing and analysis functions. However, most of the teams are not cross-feature oriented, neither are they dedicated to a single portfolio project. The predominance of components teams over features teams, also means that most of our teams have dependencies and need a way for them to stay coordinated and synchronized. Using an enterprise-wide synchronized Sprint schedule and Sprint duration is one good way to address this need. Since our transition, our Sprint duration at three weeks for all teams.

SAFe and LeSS recommend synchronized sprint as a scaling pattern. Mike Cohn also mentions synchronized sprint as an effective way to coordinate and synchronize several teams (Cohn, 2009, 343-345). Although Cohn does not stipulate that, each team needs to have the same Sprint duration.

At Cambia, synchronized enterprise sprints have helped team stay coordinated on work that requires multiple teams. Since all teams are in Sprint Planning on the same day, the Product Owners are able to coordinate dependencies, and if necessary, re-order their respective Sprint Backlogs. In addition, as Cambia does not deploy specialized project or program level Systems team as SAFe recommends, synchronized sprint schedule helps in reducing risks of untested or unintegrated code for longer periods. This in turn helps reduce hardening time before production deployment.

Since Cambia has offices in four states, many agile teams are distributed over multiple locations, having Synchronized enterprise Sprint schedule also helps in planning and budgeting for occasional co-location of these distributed teams for face-to-face Sprint planning or other major planning events. This has led to better planning sessions, and in many cases it has helped in making distributed teams more cohesive.

3.2. Scrum of Scrums Meetings

Synchronized Sprints allows Cambia agile teams to coordinate and synchronize their work during sprint planning. However, we also needed a way to stay aligned during the Sprints as well track and monitor project progress. In order to address this issue, we use Scrum of Scrum meetings. This practice is recommended in SAFe as well as LeSS. Cambia utilizes various types of Scrum of Scrums depending on the need of the teams and projects:

- **Project Scrum of Scrums** – Each portfolio project generally runs a Project Scrum of Scrums. This meeting invites representatives from Agile teams that have work for that particular project. Build and deployment engineers, operations and other business units are also represented on this meeting.

- **Testing-Specific Project Scrum of Scrums**: Many projects also organize testing specific Scrum of Scrums. During this meeting, testing concerns that are not critical enough to be raised during regular Scrum of Scrums, but are important to Agile teams and to User Acceptance Testing (UAT) group, can be discussed. It also serves as a way for teams to coordinate their system integration and testing work. Generally, a team member with QA expertise from each of the Agile team attends this meeting, along with UAT representatives and test environment coordinators. This meeting also helps in coordinating integration and testing work with vendors or external partners.

- **Organization Scrum of Scrum** – This meeting is usually organized for “like” teams to share any common concerns. For example, at Cambia, several Agile teams work on Electronic Data Interchange (EDI) processes. These EDI teams meet to discuss common issues, concerns, and development approaches for their domain.

The frequency of these meetings varies. Typically, each project Scrum of Scrums meets at least twice a week. Project Scrum of Scrums has improved interaction between the agile teams, operations teams and other business units. With the use of project dashboards, most projects also use the Scrum of Scrums meeting to monitor the progress of feature development, surface any issues regarding integration of code, deployment readiness, vendor and partner coordination and other needs arising out of inter-team interaction, on a set cadence. Other major benefit of various Scrum of Scrums is a marked reduction of
effort on the part of the Project Management Office (PMO) in collecting project status from each individual
teams (as it used to happen before), as well as less overhead and distractions for the agile teams.
Similarly, both the Project and Testing-specific Scrum of Scrums has allowed to better alignment on
systems integration and testing between agile teams. These meetings have provided a way for the UAT
testers to monitor the progress of feature development, engage in early testing and refine their test plans
for the formal user acceptance testing.

3.3. Hardening Sprints

As, many Cambia teams are component teams, there is not enough Agile maturity or fluency to produce
potentially shippable code at the end of every Sprint. Moreover many Product Owners and teams are
not adept at deconstructing features, and stories into vertical, thin, business value oriented, requirement
slices. To address this issue, we use hardening Sprints to account for any remaining work that could not
be completed during the Sprints, to make potentially shippable code in product-ready solution.

Hardening Sprint before the end of quarterly Release process is a feature of SAFe framework. Hardening
Sprint at Cambia typically comprised of systems performance testing, formal user acceptance tests and
other end-to-end testing including final integration work with external vendors and partners. The number
of Hardening Sprints is largely dependent on the project size, complexity and number of releases each
project has planned to complete its delivery. For a typical project which generally plans quarterly
releases, a Sprint (and sometimes even less time) is scheduled for hardening. Projects that have longer
release cycles and those that require complicated vendor coordination, or require integration with outside
entities usually plan higher number of Hardening Sprints. By shifting systems integration work as part of
feature complete definition, we are find many teams have to reserve less time for hardening.

3.4. Multi-team, Multi-Project Quarterly Planning Meeting

As Scrum only specifies two levels of planning – Sprint and Daily Scrums, use of Scrum alone was not
sufficient for Cambia to understand the overall progress of our portfolio implementation during the course
of the year. Due to various constraints including meeting regulatory compliance, for many projects and
teams just-in-time planning is not always sufficient. For many portfolio projects that have mandated
deadlines, Cambia also needed a way to predict our work demand and understand our resource capacity.

Over two years ago, Cambia started organizing Quarterly Planning events to start addressing some of
these needs. The initial Quarterly Planning meeting was attended by a handful of teams who were
contributing to a federally mandated compliance project called ICD-10 (WHO approved 10th revision of the
International Classification of Diseases). Centers for Medicare and Medicaid (CMS) at that time had
mandated all health care organizations in the US to implement ICD-10 by October 2012 (since then the
deadline has moved twice). Every quarter since, Cambia has planned a multi-team, multi-project,
quarterly planning meeting to align our work over the mid-term horizon, and to get insight into potential
deadline and resource contentions.

This practice is similar to LeSS recommended framework-2 Pre-Sprint Product Backlog meeting, in intent.
However, it is differs significantly. It is perhaps closer to SAFe specified Release Planning meeting, with
couple of major differences. First, the meeting is not an all teams, all-hands Release Planning meeting
as SAFe specifies it. Generally, two to three representatives from each team (Scrum Master, Product
Owner and a key Team Member) attend this meeting. Apart from team level participation, project
managers, select representatives from IT operations and members from key business units, also attend
this meeting. The second difference is that as Cambia teams are not organized along SAFe-like Value
Streams, we do not perform a rigorous planning of a Potentially Shippable Increment (PSI) or planning of
Agile Release Trains.

For the Quarterly Planning meetings, the teams define their quarterly team backlog with project specific
deliverables, non-project backlog items in form of enhancements, product support and other maintenance
work. Many teams also set aside team capacity for training or other specific continuous improvement
items. Although, many teams and POs find it challenging to maintain a healthy balance between all the competing project priorities and ongoing improvements.

The general format of this meeting is:

1) At team level, team members refine their quarterly backlog of known work with the PO.
2) Team representative attend the quarterly planning meeting. During the first part of the meeting, teams meet, discuss, and coordinate work. This may lead to re-ordering of their quarterly backlogs including identification of additional work, or coordinating systems integration and testing.
3) Teams identify questions, impediments or issues affecting their teams’ ability in committing to an achievable quarterly plan. This includes planning for any Hardening Sprints for project deliverables.
4) Teams report out their confidence level.
5) Projects collate information from teams’ backlogs and report out their confidence at a project level.

Quarterly Planning meeting has allowed teams and projects to understand potential gaps in getting feature and story definitions completed, recognize potential schedule risks, as well as recognize resourcing issues. The Quarterly Planning meeting has also provided us a way to perform backlog leveling between certain under- and over-allocated teams. The meeting provides insight for the PMO team to work with Product Owners in leveling and maintaining desired portfolio proportions between strategic and maintenance work allocation. However, it should be noted that based on our current maturity level, backlog and portfolio leveling is successful only on a limited basis.

3.5. Communities of Practice

Communities of Practice (CoP) are another important scaling practice that is widely used at Cambia. Communities of Practice are generally “a group of people who share a craft or profession.” A Center of Excellence (CoE) refers to a “team, a shared facility or an entity that provides leadership, evangelization, best practices, research, support and/or training for a focus area. At Cambia, CoPs are generally a combination of Communities of Practice and Centers of Excellence. They are called Best Practice Exchanges (BPEs). Cambia BPE normally doesn’t have a dedicated team that provides leadership and determines best practices. Instead, a volunteer core team generally forms as the nucleus of a CoP. The volunteer core team member generally dedicates about 10-20% of her time towards CoP activity. This core team engages in activities typical of a CoE, as well as helps with organizing learning and knowledge sharing events.

Two of the more active CoPs at Cambia are Agile Best Practice Exchange and Software Quality Best Practice Exchange. Last year, Agile Best Practice Exchange developed twenty best practice primers on agile practices. It also organized 36 Agile Overview Training, Agile Online Spotlight sessions and User Story writing workshops. It also organized two internal Open Space events. Within the Agile BPE, there is also an agile coaching and training community that provides coaching and mentoring help to teams and projects.

Similarly, Software Quality BPE organizes quarterly town halls, weekly online discussion events, bi-weekly quality practices meetings, QA certification learning forums, and practices regarding test automation.

Apart from these CoPs, there are other BPEs such as Secure Application Coding, Architecture, and Software Development. The activity level of each of these BPEs ebbs and flows, depending on the engagement of the core team and the community at-large as well as delivery pressures.

As we know, the Product Owners represent the “entrepreneurial” aspect of the company, whereas, role-based and function-based CoPs represent the craft or profession. Product Owners generally focus on meeting project delivery and time to market considerations. Cambia BPEs serve as a counter-balance by sharing and improving craft-based sustainable practices. BPEs work to enhance skills in team members...
which can continue to provide sufficient fuel to deliver high quality working software, on time, perpetually into the future. Both of these functions are needed and complement each other. LeSS explicitly calls for the use of CoPs. SAFe doesn’t directly specify the needs of CoPs. However, many Agilists recommend this practice including Mike Cohn (Cohn, 2009, 347-352). Henrik Kniberg documented a similar structure to Cambia comprising of Agile teams and CoPs at Spotify. At Spotify, agile teams are called Squads, and CoPs are called Guilds (Kniberg and Ivarsson, 2012).

3.6. Benefits

Many of the team and enterprise level practices have led to significant improvement in our delivery and quality practices. On the delivery front, our agile teams have achieved steady flow of business value. For example, vast majority of our teams generally have a 90+ percentage of commit versus accepted story ratio. Through the Quarterly Planning meetings, we have achieved good visibility one quarter out on gaps in our planning, risks, issues and capacity needs. The practices of last three years have led to significant improvement in our overall agility and adaptability. For example, in 2013, during the first year of Affordable Care Act implementation, Cambia was prepared with viable solution for each of the four state-based health exchanges that we operate in (Oregon, Washington, Utah and Idaho). This included working with challenging partners such as Federal Facilitated Marketplace and Cover Oregon, which had significant issues in their implementation and did not provide timely specifications and requirements to insurance carrier or engaged in any meaningful integration testing. Another federally mandated ICD-10 project has now gone through two delays. Largely due to the implementation of Scrum, and enterprise level scaling practices, Cambia has achieved a level of agility to meet these types of changing market needs. In the past, these type of changes introduced significant stress in the organization.

Similarly, we have managed to improve our quality practices, with performing systems integration and user acceptance testing much earlier in the project cycle. Vast majority of the projects at any given time manage to keep their defects backlog to under double digits. This is also a significant improvement, these backlogs used to be significantly higher, in the tens of defects. Another instance of improvement is the number of change requests (CRs) we introduce into production. For instance, in the first seven months of 2014, we deployed 466 CRs to production. Following these CRs, there were 23 CRs that “caused harm” (production incidents attributed and root caused to a specific change request). Over the same period in 2013, we had 22 CRs that caused harm with 365 CRs deployed. Thus, we introduced 27% more changes into production, while maintaining the number of production incidents. It should be noted that this improvements are correlational as opposed to causal, as we cannot attribute them to any specific practice improvement.

4. Challenges

There continue to be significant challenges for us in our agile transition and transformation process. To highlight a few:

- **Availability of a full-time Product Owner role on each of the agile teams**: The need for a Product Owner is a crucial need that continues to be a gap on some teams. Many agile teams represent several business areas, and many Product Owners are only available part-time to them as they perform other non-product management duties as well. In many instances, a Product Owner is either not empowered or is not comfortable representing other business areas. In other instances, even if a Product Owner is empowered to represent various business units, but they do not have ready access to subject matter experts (SMEs) in other business units.

- **Organizing work through traditional projects means work allocation is not evenly distributed on agile teams**: We organize work and budget using traditional projects through a yearly cycle. So depending on the type of work needed for any giving year, many agile teams get capacity constrained. Adding capacity to existing teams or spinning up new teams requires lead-time, even if budget is available. The practice of adding or changing team members to existing team is not always effective as it leads to team reforming and storming. This tends to slow teams down exactly when they need to be more productive.

- **Absence of true feature teams and value streams**: Most Cambia teams are organized around technology stacks and are only “partial” feature teams. Despite the addition of enterprise-wide
practices, Cambia agile teams require a fair amount of coordination, alignment and synchronization. Lack of feature teams also means costly handoffs and heavier reliance on processes. We have achieved a fair amount of stability by dedicating team members to a team. However, we don’t have the next level of alignment of having dedicated teams to a program or value streams, as both LeSS and SAFe specify.

- **Slow uptick of adding agile engineering practices**: A fair amount of early efforts were concentrated on booting up agile teams, acclimating team members and affiliated team members on agile methods such as Scrum or Kanban. However, penetration of agile engineering practices such as Continuous Integration, Continuous Delivery, robust automated testing regiment to decrease reliance on Hardening Sprints, development in business oriented vertical stacks, have not been uniform. The good news is that through the use of CoPs, many teams are getting enabled and improving on these practices. In many cases, shoring up development practices and reducing existing technical debt are trumped due to time to market considerations, especially meeting mandated compliance needs.

- **Onboarding of strategic projects**: As we go through annual budget cycles, many projects get funded at the beginning of the calendar year. Thus, the first quarter becomes a significant strain on the agile teams as well as the entire organization as many projects require onboarding work. However, this year, Cambia’s PMO has been staging and staggering the onboarding of projects, which has led to some relief in this area.

5. **Planned Practices**

In our agile transition and transformation journey, we have noticed that there is a significant gap between the approval of a business case, which has a very high-level program deliverables and the type of information needed at the agile team levels. To bridge this gap, we are engaging in a two-tier planning process: at the program level and at the team level. As much of our business capability and requirements flow through programs, we have started maintaining work in features, a coarser grain of functionality, into program backlogs. These backlogs expressed in the form of features are shared with teams to be deconstructed into Sprint-able stories. Just as the team Product Backlogs is refined every quarter for the Quarterly Planning event, we are doing the same of refining features on a quarterly basis. We are hoping this will start laying the groundwork on adding the practice of specifying a PSI on a quarterly basis and launching SAFe type ARTs. We think this will be a good improvement over the current practice of large, all-encompassing product solutions, rather than defining and deploying more manageable, but useable chunks of the product backlog, early.

At Cambia, it is possible to have 20-25 projects onboarding or in delivery phases, simultaneously. We recognize that we have a complex process of overlaying these projects on 40+ agile teams. We know through various feedback mechanisms that teams find it overwhelming to deal with multiple projects (for some teams it can be as high as 4-6 projects), and juggle with enhancements, maintenance requests and drawdown of technical debt. One potential solution is to launch a SAFe value stream and dedicate teams to a single program or project. We have identified couple of persistent value streams that we expect to exist and funded on a multi-year basis. This SAFe pilot will allow us to learn and understand practices that may be suitable for our context. The pilot may also provide an impetus to collapse our program level into a handful of persistent value streams, rather than the current practice of launching multiple, traditional projects on an annual basis. We expect this may provide us an impetus to start moving away from projects and move towards enabling product and business capability based development approach.

6. **Conclusion**

Each of the three Agile scaling frameworks discussed here have their pluses and minuses. Each get some things right and each fall short on some of the agile values and principles. There has been a fair amount of discussion and consternation in the agile community on best way to scale agile practices into large organizations. These discussions usually tend to break into two camps. One camp maintains that organizations must accept transformative (and sometimes radical) changes to its organizational structure
to achieve the true goals of achieving true agility. This means changes to organizational design, to HR practices, moving away from Taylorian Scientific Management approach, away from command-and-control decision-making, and away from traditional project management techniques. The other camp tends to advocate a more gradual approach by introducing transitory practices, and realizing tangible increment benefits, which then in turn can generate much needed transformative changes.

However, Ron Jeffries (and many other Agilists) through his criticism of SAFe warns that these flawed scaling frameworks might actually end up doing more harm than good, as it may give an organization a false sense of security. A sense that by implementing these scaling practices, they have truly transformed the way they do business, but they may never achieve true agility, hyper-productivity, and a more innovative and truly engaged work force (Jeffries, 2014). This is a valid criticism and it represents a true risk for an organization, which is seeking to obtain a competitive edge in meeting the challenges of rapidly changing market conditions.

At Cambia, we continue to experiment with practices that help us meet our corporate goals and approach the level of agile fluency required by the organization. We need to continue to hone our agile transition journey, continue to use practices from existing agile methods as well look to scaling models for practices that can truly provide a competitive and innovative edge. Just like any other organization, Cambia will have to maintain focus on our original intent of adaption of agile practices so that we can thrive as an organization in a post-health reform era, maintain our not-for-profit ethos, and provide affordable and transformative health care services to our members.
References:


Cohn, Mike, 2009, Succeeding with Agile: Software Development Using Scrum, Addison Wesley Professional.


Additional Reference: