Scaling Scrum Across Modern Enterprises

Implement Scrum and Lean-Agile techniques across complex products, portfolios, and programs in large organizations

Cecil Rupp

Foreword by Manjit Singh (President and CEO, Agilious LLC)

Packt>
www.packt.com

About the Author

Cecil 'Gary' Rupp brings more than thirty years of executive-level experience in information technology (IT), spanning IT and management consulting, professional services, program and project management, and sales and marketing. Mr. Rupp has directly managed more than thirty enterprise-class IT programs and projects, with the last ten years focused almost exclusively on managing large enterprise health and government IT Programs.

His professional experience includes supporting large software and systems development projects and promoting and delivering IT solutions involving computer-aided systems engineering (CASE), middleware, portal, and software development tools. A graduate of the **U.S. Naval Academy**, Gary also has an MBA in Finance. Gary is also the author of the **Building our Digital World (BODW)** series of books on software and systems development practices.

Professional certifications

- Scaled-Agile, Inc. certified SAFe® 5 Program Consultant (SPC) - ID No. 24488227-0897
- Large-Scale Scrum (LeSS) Certified LeSS Practitioner
- Scrum Alliance Certified Scrum Product Owner (CSPO) - ID No. 650996
- Scrum Alliance Certified Scrum Master (CSM) – ID No. 650996
- PMI certified Project Management Professional (PMP) - PMP Number: 243644

Scaling Scrum and Lean-Agile Practices

Methods to scale agility help multiple small teams collaborate to develop, deliver, and support large products, while simultaneously improving business agility across an organization.

The origins of Agile

REQUIREMENTS
DESIGN
IMPLEMENTATION
TESTING
MAINTENANCE
WATERFALL
DEVELOPMENT
MODEL

The values and principles of Agile came out of the software development community, specifically from a group of software engineers who worked with so-called lightweight development practices. Their overarching goal was to eliminate the negative aspects of the traditional software development model.

The traditional model was project-based, plan-driven, and followed a linear-sequential software development life cycle (SDLC) approach. A graphical depiction of the traditional model displays a stepwise approach to development through a linear-sequential process, as a single development cycle, thus leading to the name *Waterfall* approach.

There are multiple problems with the waterfall model. For example, the linear-sequential process needlessly extended the length of the overall development and project delivery cycles. The project-based nature of the traditional model is predicated on an idea that customers know upfront what their needs and priorities are for the entire project period, which is seldom the case.

Given its plan-driven nature, which drives product delivery schedules and budgets, project teams are reluctant to change the original plans and schedules even when customer priorities and needs change. Moreover, the late-stage testing approach makes it a complicated, time-consuming, and costly process to find and fix bugs.

Lightweight but powerful

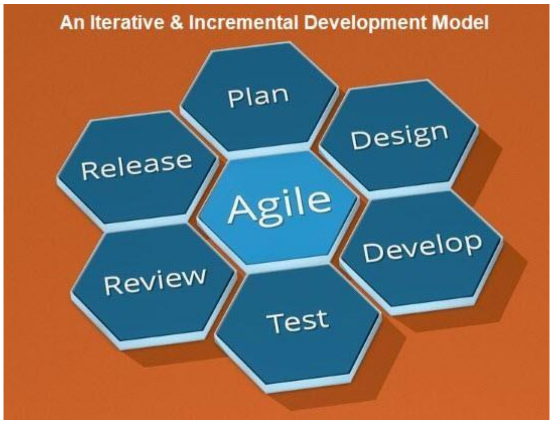
In contrast, the lightweight software development methodologies delivered updates in a continuous and customer-centric manner, adding incremental value rapidly through frequent releases. The iterative development approach enables more frequent testing on smaller sections of code, minimizing the impacts of bugs and defects while improving overall development throughput and efficiencies.

The lightweight approaches generally install small, autonomous, and fully self-contained teams that minimize communications, work integration, and dependency issues associated with managing larger teams or organizations. Instead of excessive documentation and reports, the lightweight methodologies provided visibility on priorities and progress through publicly displayed charts and lists, such as product backlogs, burndown and burnup charts, velocity charts, and Kanban boards.

All of this works well in relatively small product development activities. Among the lightweight methodologies, Scrum emerged as the leader in agile-based software development at the small team level. It retains its leadership position in that space to this day.

In any system, the number of potential connections (PC) between the nodes (n) that make up a system expands exponentially, per the equation $PC = n \ast$ (n-1) / 2. Network density is the ratio of actual connections to potential connections. ND = AC / PC. For the purpose of this article, organizations consisting of multiple agile teams are real-world examples of systems & people are the nodes.

But it wasn't long before organizations attempted to employ the values and principles of Agile on more extensive product development requirements, and across both *value creation* (i.e., development) and *value delivery* (i.e., customer-facing and support) activities.



The empiricism and small team models within the Scrum Framework are scalable, as teams of teams. But large organizations represent complex systems that required additional ways of thinking to deal with network density issues that make communications, integrations, and dependencies much more challenging to manage.

The question then becomes, how to do it?

Scaled Scrum and Lean-Agile Strategies

As it turns out, several strategies evolved to enable agility across large product organizations and also to support enterprise-wide business agility initiatives. Some scaling strategies deal more strictly with large software development projects. Others incorporate systems thinking and lean development practices to implement business agility across all value creation and value delivery activities on an enterprise scale.

A shortlist of leading scaled Scrum and Lean-Agile practices include the following:

- **Scrum-of-Scrums** – the original Scrum scaling strategy as a team of teams
- Scrum-At-Scale – An extension to the Scrum Guide that scales the basic Scrum of Scrums concepts enterprise-wide and across business domains with minimum viable bureaucracy (MVB) via scale-free architectures
- **The Nexus Framework** – the software developer's extension to the Scrum Guide that implements Network Integration Teams (NIT) to manage cross-team dependency, integration, and synchronization issues on multiteam product development efforts
- **Large Scale Scrum (LeSS)** – Another scaled-Scrum approach, with two Scrum scaling frameworks, that helps coordinate the activities of multiple teams, around *features* (LeSS Framework) and *requirements areas* (LeSS Huge Framework), working in collaboration to develop large and complex software-enabled products
- **Disciplined Agile (DA)** – A Lean-Agile approach to development that provides six product development lifecycles, numerous process guides, and hundreds of potentially useful techniques that allow teams to choose their preferred *Way of Working* based on their unique business and organizational needs and situations
- **Scaled Agile Framework® (SAFe®)** – With four configurations, a Lean-Agile approach for large organizations working on large-scale product development

efforts that can leverage their economies of scale as strengths to provide greater efficiencies and yet incorporate Lean-Agile practices to enable business agility on an enterprise scale

At the time of this writing, the **Scaled-Agile Approach (SAFe)** is the leader among the scaled-agile approaches. But *business agility* is a relatively new field, and time will tell which practices stay at the top of the leaderboard.

Scaling to achieve business agility

Our modern digital era drives today's competitive landscape, and organizations need to be agile at all levels of their businesses to compete effectively. Therefore, despite its early origins within the software industry, the values and principles of agile must apply to all value stream activities across an enterprise. In short, this means all aspects of a business must have the ability to respond quickly with frequent improvements that deliver customer-centric value at the lowest possible cost.

At one level, this shouldn't be a surprise. Software and computing systems add useful functionality to products while also making it easier and quicker to add new features. Those two capabilities alone change the competitive landscape dramatically. However, if we only apply agile practices to software development activities, the rest of the organization will lag and fail to deliver the new capabilities to their markets in a timely and cost-efficient manner. So, we must expand our thinking about applying Agile's values and principles on a larger scale to enable enterprise-wide business agility.

Business agility is the ability of an organization to rapidly evaluate alternatives and competitively respond to changes driving both a business and its industry and do so from a customer-centric and value-added perspective. Installing business agility practices across an organization helps it respond appropriately to competitive drivers while simultaneously keeping a customer-centric approach across the enterprise. In short, business entities must evolve all their value creation and delivery strategies, taking a systems-level view, to stay competitive in our fast-paced and digital-enabled world.

Organizations that make ad hoc or opportunistic changes, without looking at the systemic impacts, fall into a trap called *local optimizations*. The organization can expend much time, effort, and money and not achieve any significant gains. Instead, the enterprise must take a holistic view to optimize and streamline all its value streams. Systems thinking is the essence behind lean development, which is an essential component of any serious attempt at becoming agile on an enterprise scale.

Modern Scrum and Lean-Agile practices aim to achieve enterprise-wide agility by allowing synchronization and coordination between all the value stream activities across an enterprise to produce high-quality and profitable products. The same concepts equally apply to non-profits and government agencies that seek to deliver high-quality and customer-centric services at the lowest possible costs.

No one should think it's an easy task to obtain business agility on an enterprise scale. It takes executive level commitments, strategy, structural changes, a lot of trial and error, and leadership. Ultimately, success breeds success, and organizational culture evolves incrementally to support the most effective business strategies.

It's impracticable to provide much more detail on these agile scaling practices within a single article. Those readers who want to dive deeper can read my book titled **Scaling Scrum Across Modern Enterprises**: *Implement Scrum and Lean-Agile techniques across complex products, portfolios, and programs in large organizations*. The book is available via Amazon.com and PACKT Publishing.

References

1. Ambler, S., Lines, Mark. (2020) Choose Your Wow! A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working. Newtown Square, PA The Project Management Institute, Inc.
2. Beck, K. et al. (2001). Manifesto for Agile Software Development. http://agilemanifesto.org/
3. Corporate. (2020) SAFe® for Lean Enterprises 5.0. Retrieved from: https://www.scaledAgileframework.com/
4. Larman, C. and Vodde, B., Large-Scale Scrum: More with Less (Addison-Wesley Signature Series (Cohn)), Pearson Education, Inc., Boston, MA (2017)
5. Schwaber, K. (2018) The Nexus™ Guide. The Definitive Guide to scaling Scrum with Nexus: The Rules of the Game. Scrum.org. Retrieved from https://www.Scrum.org/resources/nexus-guide.
6. Schwaber, K. Sutherland, J. (November 2017) The Scrum Guide™ The Definitive Guide to Scrum: The Rules of the Game. https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum- Guide-US.pdf
7. Sutherland, J. (March 2020). The Scrum At Scale® Guide. The Definitive Guide to the Scrum@Scale Framework. Version 2.0 https://scrumatscale.scruminc.com/scrum-at-scale-guide- online/