Navigating Challenges of SAFe Adoption in Horizontally Sliced Agile Teams

Rakhi Jain
Colorado Technical University

Jeffrey Butler
Colorado Technical University

The Scaled Agile Framework (SAFe) enables IT organizations to implement agile practices on a larger scale, extending beyond conventional team-level agility. IT organizations that adapt SAFe practices across multiple teams often encounter significant challenges in coordination and integration. Agile practices can be opted in SAFe using horizontal or vertical slicing strategies. However, the horizontal slicing can impede time to market and hinder the operations of cross-functional teams. In contrast, vertical slicing prompts alignment, synchronization, and integration in product development. This article navigates the challenges associated with adopting horizontal slicing in the SAFe model. The authors have presented the insights favoring vertical slicing as a more effective approach.

Keywords: Scaled Agile Framework (SAFe), horizontal slicing, vertical slicing, agile development, agile process improvement

SCALED AGILE FRAMEWORK (SAFe)

The Scaled Agile Framework (SAFe) enables IT organizations to implement agile practices on a larger scale, extending beyond conventional team-level agility. SAFe promotes agile practices across large programs by organizing multiple agile teams to work collaboratively towards a shared objective ("SAFe 5.0 Framework", n.d.). Large organizations have adopted the SAFe model to reap the benefits of competitive advantage, increased productivity, and improved quality (Razzak et al., 2018). SAFe facilitates the development and management of complex applications and products that are challenging to manage using standard agile practices (Jain & Butler, 2024).

The SAFe framework, developed by Dean Leffingwell in 2010, has undergone continual evolution and is now in its latest iteration, version 6.0. Scaled. SAFe was initially implemented with three focus areas: portfolio, program, and team (Leffingwell, 2010). The primary motivation behind the development of the SAFe methodology was to address the evolving needs of customers. Today, SAFe is widely adopted among IT organizations for its ability to scale agile practices effectively. Within the SAFe framework, teams are more effectively coordinated and aligned to deliver consistent, targeted value (Jain & Butler, 2024). The framework outlines the roles and responsibilities of team members and stakeholders in a clear and organized manner to achieve value-driven outcomes (Atlassian, n.d.). The Agile Release Train is a SAFe term that refers to multiple agile teams collaborating to achieve a common objective ("SAFe 5.0 Framework", n.d.).

AGILE RELEASE TRAIN (ART)

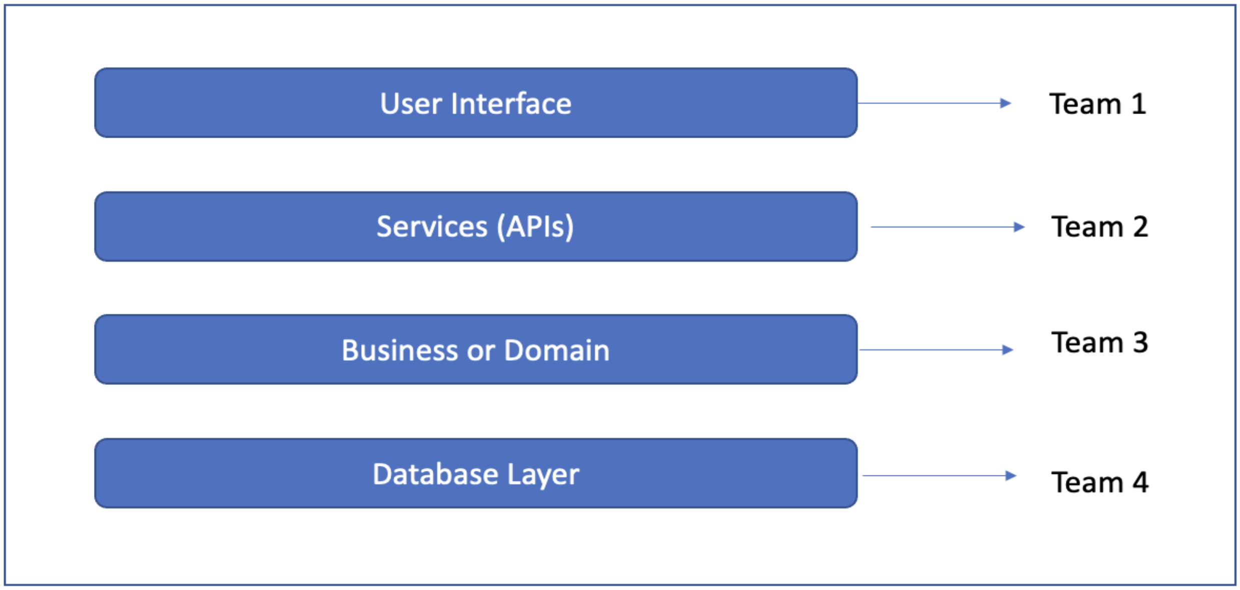
The Agile Release Train (ART) is a combination of self-organizing agile teams working together to deliver shared objectives within a predefined time increment. Each team in an ART is a scrum team that includes a product owner, a scrum master, and a few developers, along with quality assurance team members (Ciancarini et al., 2022). An ART is the heart or core of a scaled agile framework (SAFe) implementation at the program level in any organization and can hold up to 150 team members in total ("SAFe 5.0 Framework", n.d.). All agile teams within a single Agile Release Train (ART) are aligned by sharing a common mission and vision through a single program backlog that covers specific business or product areas (Insfran et al., 2022). Agile teams in an Agile Release Train (ART) are required to deliver a valuable, system-level solution at the end of each iteration. The iterations in ART are pre-defined time durations and usually last for one, two, or three weeks ("SAFe 5.0 Framework", n.d.).

An organization can operate multiple ARTs to deliver the various work components. Typically, one ART can comprise between 5 to 12 teams. Multiple ARTs within an organization can work independently or coordinate their efforts to deliver their work. However, the teams in one ART should work in tight collaboration to deliver the shared objectives. The success of SAFe implementation in an organization depends on how well these teams have been carved out, which need to operate under one ART. The carving or slicing of large-scale projects into smaller subsets or teams is required for better management, maintenance, and tracking. Organizations are opting for either horizontal or vertical slicing strategies to structure ARTs for delivering value across their value streams.

ART's HORIZONTAL SLICING

Horizontal or functional sliced teams are organized or carved out by identifying the technical expertise in the SAFe model. For instance, user interface (UI), database, middleware, and infrastructure teams can be part of one ART when teams are sliced horizontally.

FIGURE 1
HORIZONTAL SLICING IN AN ART



"Figure 1" illustrates the ART setup when horizontal functional areas organize teams. Each functional layer, including user interface, services, domain, and database, is represented by an individual team. Each layer of the application is developed independently by a separate team within an Agile Release Train (ART).

Benefits of Horizontal Slicing

The key benefit of using a horizontally sliced team in an ART is to maintain a specialized focus at the ART level. ART leadership can reuse the core capabilities in the deliveries. Horizontal teams can deliver fully functional parts of an application even when the overall development system is not yet complete (Saltz et al., 2022). However, due to dependencies on other teams, the time required to complete the work often increases.

Challenges of Horizontal Slicing

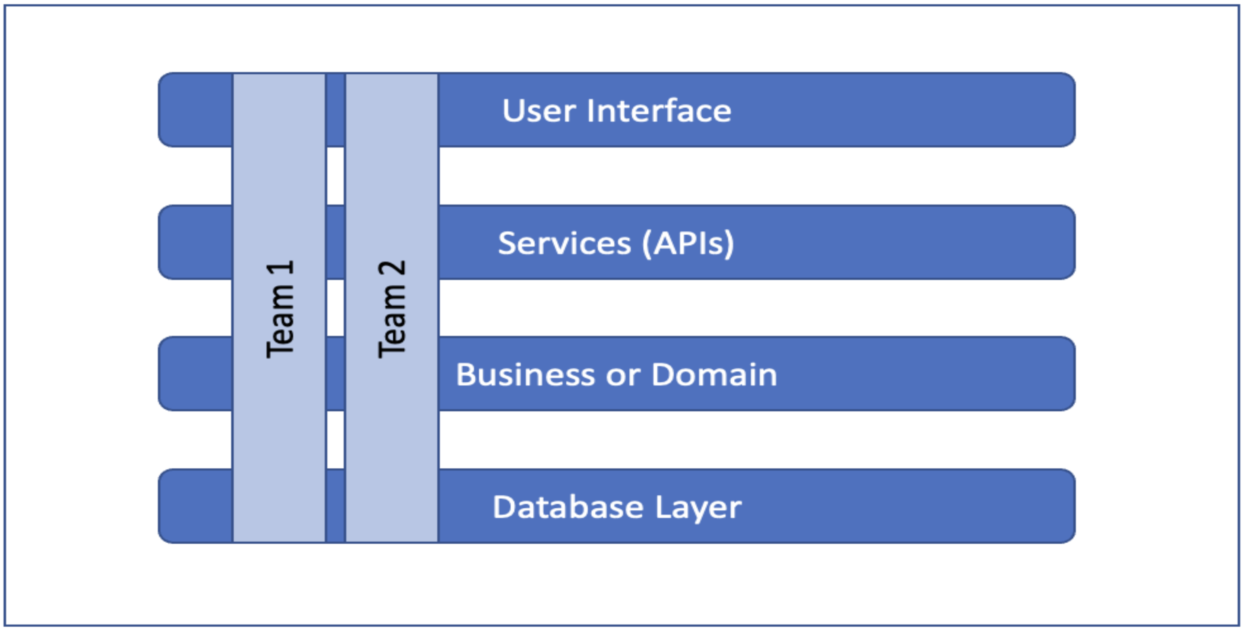
Teams organized through horizontal slicing often encounter challenges with delivering products and value to customers. Significant coordination is required to maintain the scheduled and independent deliveries at the ART level in this setup, due to unmanaged dependencies. Product deliveries are held at risk due to the late integration of components with other agile teams in ART. Agile teams may deliver incomplete and intermediate work to customers at the end of each iteration; however, such deliverables may not be considered progress toward business value (Saltz et al., 2022). Agile teams enhance and refine their technological skills by adopting work practices that utilize horizontal slicing (Alter & Browne, 2005). Testing various components and production deployments can be challenging due to the individual technical layers of the product. Horizontal slices may optimize specialization but deliver no user value until all layers are complete. Teams have observed the risk of rework after integration and deliveries to the customer in this model (Saltz et al., 2022).

ART's VERTICAL SLICING

Vertical-sliced teams are organized or carved out by identifying the end-to-end value delivery in the SAFe model. This is a preferred model in Scaled Agile practice where teams deliver the work in small increments, also known as iterations, by cutting through all layers of the targeted application. Vertical teams in ART are aligned by product or feature and are responsible for delivering a thin, end-to-end, completed, and usable product. Vertical-sliced teams span across different applications and architectural layers, e.g., user interface, services, and database, and can deliver customer value continuously with each iteration.

In vertical slicing, the system layer is divided into multiple components, with an agile team assigned to each element. However, in the case of horizontal slicing, an agile team is represented by a system layer.

FIGURE 2
VERTICAL SLICING IN AN ART



"Figure 2" illustrates the ART setup when teams are organized based on the vertical slicing strategy. Each team will have the representation from various application layers, e.g., user interface, services, domain, and database.

Benefits of Vertical Slicing

The key benefit of using a vertically sliced team in an ART is to maintain clear ownership of product deliveries and improve stakeholder management (Burmeister et al., 2019). Teams can also reap the benefits of faster, small-batch, and continuous deliveries by employing short iterations when using vertically sliced teams. The application is developed using a top-down approach, and each component can be tested independently. Testing of the products or applications is relatively easy in vertically sliced teams as compared to horizontally sliced teams. Teams can receive faster feedback on a well-integrated, quality-oriented, and working product from customers after each sprint or iteration. Cross-team dependencies are reduced because end-to-end functionality is delivered within the team, as opposed to being sliced horizontally. Agile teams' synchronization is more effortless in vertically sliced teams, which results in the achievement of aggressive deadlines. Teams can enforce the early integration of various components, a core objective of the Scaled Agile Framework (SAFe) or Agile Release Trains (ART) deliveries.

VERTICAL SLICING IMPLEMENTATION

ART teams strive to deliver quickly when the vertical slice method is selected for the team formation (Saltz et al., 2022). This section provides a sequence of steps that can be followed in any ART to implement the vertical slicing strategy.

Define Product Roadmap and Features

When defining the structural setup of any ART, the first step is to identify the product and technology roadmaps, along with key deliverables, in the form of high-level requirements. Product roadmaps are referenced in business plans, which in turn include high-level requirements (Lal & Clear, 2021). These high-level requirements are derived from the organization's tangible objectives (Gollhardt et al., 2022).

These high-level requirements need to be grouped further according to the layers of enterprise architecture (Röglinger et al., 2016). Project or program leadership should also establish a technology roadmap or landscape to understand the upcoming technological changes within the organization, based on the enterprise's vision, strategy, and objectives (McKeen & Smith, 2006).

Form Cross-Functional Teams

Once the product roadmap is finalized, ART needs to be divided into smaller, vertically sliced teams. These vertical teams should be formed with team members from each of the system layers, capable of delivering end-to-end functionality without handing over the development work to other teams that specialize in specific development skills. Each agile team should include developers, quality assurance team members, a Scrum Master, and a Product Owner (Ciancarini et al., 2022). The formation of a vertical slice ensures that the necessary expertise and required skills are present within the team.

Synchronized Cadences and Integration

Soon after the formation of agile teams, a regular and synchronized cadence needs to be established across all teams within the given Agile Release Train (ART) for progressive integration. The regular cadence includes defining the scrum of scrums, iteration durations, and demonstrating iteration results through system demos (Horlach et al., 2019). All agile teams within one Agile Release Train (ART) will participate in a Program Increment (PI) planning event to plan for the upcoming quarter, based on the product and technology roadmap (“SAFe 5.0 Framework”, n.d.). Apart from these synchronized cadences across teams, each team should establish its iteration planning, daily stand-ups, and refinement sessions regularly (“SAFe 5.0 Framework”, n.d.). Teams need to develop continuous integration pipelines to check in the implemented code in a shared repository. Setting up a synchronized cadence will help accelerate ART deliveries and minimize dependencies between teams. ART teams will be able to integrate the components more quickly and will conduct regular interface testing.

FINDINGS

The authors have analyzed both vertical and horizontal slicing strategies in SAFe development. They have found that both are valuable techniques for software development cycles; however, the selection of a specific strategy depends on the project's requirements. Horizontal slicing works well for functionalities that consist of multiple layers, as it enables each layer to be developed separately. This approach is particularly beneficial when building complex features with numerous interdependent components (Saltz et al., 2022). Vertical slicing, in contrast, is advantageous when a project needs a fully functional system at each stage of development. It’s beneficial for applications with distinct components that can be developed and tested as complete, end-to-end slices (Saltz et al., 2022). There may be instances when a project requires a hybrid approach, combining both horizontal and vertical slicing methods. In the SAFe model for the initial business understanding phases, teams may want to use horizontal slicing. For end-to-end product development across all system layers, teams should use the vertical slicing strategy (Saltz et al., 2022).

CONCLUSION

Vertical slicing is a critical practice for implementing the SAFe projects with higher success rates. ARTs in SAFe can organize small agile teams that are responsible for the end-to-end system layer work to avoid coordination and delay challenges. Vertical slicing in SAFe accelerates delivery cycles by enabling early integration at the component or feature level. This approach aligns teams and stakeholders around complete features, enabling early system integration and feedback. IT organizations can adjust more rapidly to change, deliver critical features within business constraints, and increase overall agility by adapting vertical slicing.

REFERENCES

Alter, S., & Browne, G. (2005). A Broad View of Systems Analysis and Design: Implications for Research. *Communications of the Association for Information Systems*, *16*. https://doi.org/10.17705/1CAIS.01650

Atlassian, J. (n.d.). *What is SAFe?* Retrieved from www.atlassian.com

Burmeister, F., Drews, P., & Schirmer, I. (2019). An ecosystem architecture meta-model for Supporting ultra-large-scale digital transformations. *AMCIS 2019 Proceedings*, *3*. Retrieved from https://aisel.aisnet.org/amcis2019/enterprise_systems/enterprise_systems/3

Ciancarini, P., Kruglov, A., Pedrycz, W., Salikhov, D., & Succi, G. (2022). Issues in the adoption of The scaled Agile framework. *2022 IEEE/ACM 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 175–184. https://doi.org/10.1109/icse-seip55303.2022.9794087

Gollhardt, T., Hermann, A., Cordes, A., Barann, B., & Kruse, P. (2022). Design of a software tool Supporting orientation in the context of digital transformation. *Proceedings of the Annual Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2022.592

Horlach, B., Schirmer, I., & Drews, P. (2019). Agile Portfolio Management: Design Goals And Principles. In *Proceedings of the 27th European Conference on Information Systems (ECIS),* Stockholm & Uppsala, Sweden, ISBN 978-1-7336325-0-8.

Insfran, E., González, F., Abrahão, S., Fernández, M., Barry, C., Lang, M., . . . Schneider, C. (2022). *Advances in information systems development: Crossing boundaries between development and operations in information systems.* Springer Nature.

Jain, R., & Butler, J. (2024). Team-level reforms for a scaled Agile framework in outsourced Setting. *The Pinnacle: A Journal by Scholar-Practitioners*, *2*(2). https://doi.org/10.61643/c22958

Lal, R., & Clear, T. (2021). Three Levels of Agile Planning in a Software Vendor Environment. *ACIS 2021 Proceedings*, *48*. Retrieved from https://aisel.aisnet.org/acis2021/48

Leffingwell, D. (2010). *Agile software requirements: Lean requirements practices for teams, programs, and the enterprise.* Addison-Wesley Professional.

McKeen, J.D., & Smith, H.A. (2006). Developments in practice XXIII: Creating and evolving a Technology roadmap. *Communications of the Association for Information Systems*, *18*. https://doi.org/10.17705/1cais.01821

Razzak, M.A., Richardson, I., Noll, J., Canna, C.N., & Beecham, S. (2018). Scaling Agile across The global organization. *Proceedings of the 13th International Conference on Global Software Engineering*, 121–130. https://doi.org/10.1145/3196369.3196373

Röglinger, M., Bolsinger, M., Haeckel, B., & Walter, M. (2016). How to Structure a Business Transformation Projects: The Case of Infineon's Finance IT Roadmap. *Journal of Information Technology Theory and Application (JITTA)*, *17*(2).

SAFe 5.0 Framework. (n.d.). *Scaled Agile Framework.* Retrieved from https://www.scaledagileframework.com/

Saltz, J., Sutherland, A., & Hotz, N. (2022). Achieving lean data science agility via data-driven Scrum. *Proceedings of the Annual Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2022.876