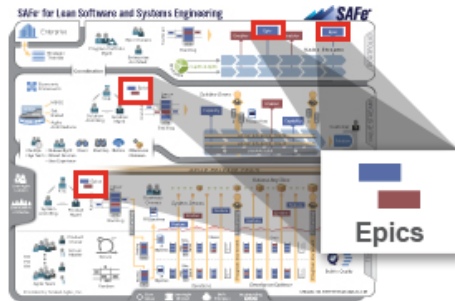


<https://twitter.com/ScaledAgile>
<https://www.linkedin.com/grps/Scaled-Agile-Framework-4189072?>

<https://www.youtube.com/user/scaledagile>
<http://www.slideshare.net/ScaledAgile>



(/)

|| *A system must be managed. It is management's job to direct the efforts of all components toward the aim of a system.*

—W. Edwards Deming

Epic Abstract

Epics are containers for significant initiatives that help guide value streams toward the larger aim of the portfolio. In so doing, they drive much of the economic value for the enterprise. Epics are large and typically crosscutting, crossing multiple Value Streams and Agile Release Trains (ARTs). They are investment intensive and far ranging in impact; formulation and analysis of cost, impact, and opportunity is a serious matter. To this end, Epics require a lightweight business case and financial approval before implementation. There are two kinds of epics: Business epics and Enabler epics, and they may appear at the Portfolio, Value Stream, and Program Levels.

Most of this article is devoted to describing the definition, articulation, and implementation of portfolio epics, as they are the largest and most impactful. Portfolio epics are developed and managed through the Portfolio Kanban, where they are processed through various states of maturity until they are either rejected or approved, in which case they move to the Portfolio Backlog to await implementation. This process assumes reasoned economic analysis of the business drivers behind the epic, its technical impact, and strategies for incremental implementation.

Value stream and program epics require similar treatment. They are also described in this article.

Details

Portfolio Epics

The largest *Epics*, portfolio business epics and Enabler Epics (/enablers/), capture the largest crosscutting initiatives that occur within a Portfolio (/portfolio-level/). Business epics directly deliver business value; enabler epics are used to evolve the Architectural Runway (/architectural-runway/) to support upcoming business epics.

These epics are initially captured in the Portfolio Kanban (/portfolio-kanban/) and move through this system under work-in-process (WIP) limits. This helps assure that those doing the work have the time necessary to conduct responsible analysis. Equally important, the Kanban system helps manage expectations for reasonable scoping and time frames for implementation of new business ideas. The final decision for actual implementation of each epic is subject to the authority of Program Portfolio Management (PPM) (/program-portfolio-management/). As resources become available, decision makers can select from the best of a number of possible business opportunities, as there will typically be multiple analyzed epics in the backlog at any time. Epics that are approved go to the Portfolio Backlog (/portfolio-backlog/), where they await implementation capacity.

Figure 1 provides an *Epic Value Statement* template that can be used to capture, organize, and communicate key information about an epic. This expression is developed in the Kanban *Review* state and is intended to provide just enough information to have a meaningful discussion about the proposed initiative.

Forward-Looking Position Statement	
For	<customers>
who	<do something>
the	<solution>
is a	<something – the "how">
that	<provides this value>
Unlike	<competitor, current solution, or non-existing solution>
our solution	<does something better – the "why">
Scope	
Success criteria:	• •
In scope:	• •
Out of scope:	•

	•
NFRs:	•
	•

Figure 1. Epic value statement template format

Analysis

Epics are the most significant initiatives within the portfolio, so they must be carefully analyzed before being committed to implementation. Epic Owners (/epic-owner/) take responsibility for this important task, while Enterprise Architects (/enterprise-architect/) shepherd enabler epics that support technical considerations for business epics. The most worthy epics from the backlog are passed to *analysis* when space becomes available in that queue. There, effort and economic impact are better defined, WSJF (/wsjf/) prioritization is refined, cost estimates are established, and a lightweight business case is developed. Analysis efforts can include:

- Workshops with business stakeholders to understand and describe business benefits of the business epic
- Workshops with Architects or Systems Engineering (/system-and-solution-architect-engineering/), from the Value Stream (/value-stream-level/) and Program Levels (/program-level/), and Agile Team (/agile-teams/) to understand implementation effort and impact on current solutions; other relevant SMEs can be involved
- Implementation of spikes, research, and exploration activities by the teams
- Developing concrete examples to resolve ambiguities (*specification by example*)
- Defining the *success criteria* for an epic

The result of the analysis phase is a lightweight business case that captures the results of the analysis, including a refined description, success criteria, estimates of implementation time and cost, and program impact. An example format is provided in Figure 2 below. The business case is then used by the appropriate authorities to make a go/no-go decision for the epic.

SCALED AGILE™			
Lightweight Business Case			
Epic Name: (Short name for the Epic)	Funnel Entry Date: (Date the Epic entered the funnel)	Epic Owner:	
Epic Description: (A description of the Epic; consider using the Epic Value Statement format: http://www.scaledagileframework.com/epics/)			
Success Criteria: (Describe how the success of the Epic will be measured, Availability increases from 93% to 99.9%, etc.)			
In Scope:	Out of Scope:		
• ...	• ...		
• ...	• ...		
• ...	• ...		

SCALED AGILE™		
Lightweight Business Case		
Analysis Summary: (Brief summary of the analysis that has been formed to create the business case.)		Go / No-Go: (Go, or No-Go recommendation)
Estimated Investment	Estimated Story Points: (Estimated story points for the Epic)	Estimated Monetary Cost: (Example: Estimated story points * cost per story point)
Weighted Shortest Job First: (WSJF relative to other Epics in the analysis queue)	Type of Return: (Market share, increased revenue, improved productivity, new markets served, etc.)	Estimated Revenue: (Revenue, Return on Investment, or other applicable financial metrics)

Sponsors: (List key business sponsors who will be supporting the initiative)		In-house or Outsourced Development: (Provide recommendations for where the Epic should be developed)	
Users and Markets Affected: (Describe the user community of the solution and any markets affected)		Estimated development Timeline	Start Date: (Estimated start date)
Products, Programs, Services Affected: (Identify products, programs, services, teams, departments, etc.)		Completion date: (Estimated calendar date or estimated number of PIs)	
Impact on Sales, Distribution, Deployment: (Describe any impact on how the product is sold, distributed, or deployed)		Incremental Implementation Strategy: (Epics are defined as a single whole, but each epic undergoes incremental implementation. Visit http://www.scaledagileframework.com/implementation-strategies-for-business-epics/ for details on potential strategies. Many parts of this guidance can be applied to Architectural Epics)	
Sequencing and Dependencies: (Describe any constraints for sequencing the Epic and identify any potential dependencies with other Epics)			
Milestones or Checkpoints: (Identify potential milestones or checkpoints for reevaluation of the Epic)			
Attachments: (Other supporting documentation, links to other data, feasibility studies, models, market analysis, etc., that were used in the creation of the business case)			
Other Notes and Comments: (Any additional miscellaneous information)			

Figure 2. Epic lightweight business case

Download Lightweight Business Case Template (<http://www.scaledagileframework.com/?wpdmact=process&did=MTA0LmhvdGxpbms=>)

Success Criteria

The epic value statement and lightweight business case both include *epic success criteria*, which can be used to validate that implementation is complete and successful. By their very nature, portfolio epics are large and tend to be abstract, so defining success criteria is a capstone to establishing a shared understanding among stakeholders of what the epic really implies for the business.

Be Careful about Absolute Success Criteria

However, epics are unique from Capabilities and Features (/features-and-capabilities/), as it is often not necessary to complete epics, and that is why teams have to be so careful with success criteria. Due to their scope and impact, it is sometimes the case that implementing some—but *not all*—of an epic achieves the bulk of the economic value, and in accordance with WSJF, there may be a point at which some other investment would produce a greater economic return than completing the epic as anticipated.

For an example from experience, we discovered that an epic of “migrate all applications to use JBoss Webserver” would have been better stated as “migrate all applications to Jboss webserver—*except those applications approaching end of life.*”

Measuring Progress

Since epics typically cut across value streams and Program Increments (/program-increment/), it may be difficult to assess how progress

is being made on the development of its capabilities or features. Therefore, SAFe provides a set of portfolio Metrics (/metrics/) that can be used to measure epic progress.

Incremental Implementation

Approved epics stay in the portfolio backlog until such time as there is implementation capacity available in one or more Value Streams (/value-streams/). While it's easy to think of epics as big, binary, monolithic blobs of work, they must be implemented incrementally to achieve the benefits of agility [2]. The Epic Owner or Enterprise Architect has the responsibility to work with the Product Managers (/product-and-solution-management/) and Architects or Systems Engineering to split the epics into value stream level and program level epics, or directly into capability or feature backlog items.

Splitting Epics

Implementing incrementally means that an epic must be split into smaller pieces that represent incremental value. Table 1 provides nine suggested methods for splitting epics, along with examples for each:

1. Solution / Subsystem / Component – Epics often affect multiple Solutions (/solution/), subsystems, or large components. In such cases, splitting by these aspects can be an effective implementation technique.	
<i>Multiple user profiles</i>	<i>... Multiple profiles in the opt-out website</i> <i>... Multiple profiles in the admin system</i>
2. Success Criteria – The epic success criteria often provide hints as to how to incrementally achieve the anticipated business value.	
<i>Implement new artifact in search results:</i> Locations Success criteria: a) Locations should provide additional filtering method when other disambiguation methods aren't useful b) Provide detailed location of a person	<i>... Provide state information in the search results</i> (criteria [a] is partially satisfied, as states alone already provide some good filtering capability) <i>... Implement compound location:</i> State and city (entire success criteria is satisfied)
3. Major Effort First – Sometimes an epic can be split into several parts, where most of the effort will go toward implementing the first one.	
<i>Implement single sign-on across all products in the suite</i>	<i>... Install PINGID protocol server and test with mock identity provider</i> <i>... Implement SSO management capability in our simplest product</i> <i>... Implement SSO in our most complex product</i> <i>... Proliferate as quickly as backlog capacity allows</i>
4. Simple/Complex – Capture that simplest version as its own epic, and then add additional program epics for all the variations and complexities.	
5. Variations in Data – Data variations and data sources are another source of scope, complexity, and implementation management.	
<i>Internationalize all end-user facing screens</i>	<i>... in Spanish</i> <i>... in Japanese</i> <i>... prioritize the rest by then-current market share</i>

6. Market Segment / Customer / Class of User – Segmenting the market or customer base is another way to split an epic. Do the one that has higher business impact first.	
<i>Implement opt-in functionality</i>	... For current partners ... For all major marketers
7. Defer Solution Qualities (NFRs) – Sometimes the initial implementation isn't all that hard and the major part of the effort is in making it fast—or reliable—or more precise—or more scalable, so it may be feasible to achieve the solution qualities (Nonfunctional Requirements (/nonfunctional-requirements/)) incrementally.	
8. Risk Reduction Opportunity Enablement – Given their scope, epics can be inherently risky; use risk analysis and do the riskiest parts first.	
<i>Implement filtering search results by complex user-defined expression</i>	... Implement negative filtering ... Implement complex filtering expressions with all logical operations
9. Use Case Scenarios – Use cases [1] can be used in Agile to capture complex user-to-solution or solution-to-solution interaction; split according to specific <i>scenarios</i> or <i>user goals</i> of the use case.	
<i>Transitive people search functionality</i>	(goal 1 ~) Find connection to a person (goal 2 ~) Find connection to a company (goal 3 ~) Distinguish strong and weak connections

Table 1. Methods for splitting epics

Approving the Investment in Epics

Epics play another important role in the system that constitutes SAFe. Specifically, epics, by their very definition, require financial approval from the portfolio stakeholders, even in the context of an approved value stream budget. The ability to even implement Lean-Agile budgeting—to fund the value streams directly and empower value stream stakeholders to allocate spend to the initiatives—is dependent on certain checks and balances in the system. One of these is that large-spend items still require visibility approval, even if the value stream budget can support them. Epics play a key role in that process as they consume significant amounts of resources and often represent changes in technical or business strategy. It behooves everyone—those empowering, and those empowered—to collaborate and approve such large-spend items. Refer to Budgets (/budgets/) for further discussion.

Program and Value Stream Epics

The above discussion describes the most impactful kind of epic, portfolio epics. Many of these epics generate value stream epics and, correspondingly, program epics as well. In addition, many large initiatives (epics) arise at the local value stream or program level. While largely a local concern, their financial, human, and other resource impacts are large enough in scope to warrant a lightweight business case and discussion and financial approval from PPM. That's what makes an epic an epic. However, the splitting, implementation, and management practices for value stream and program epics is primarily a local concern. Methods for managing the epics at these levels are discussed in the article on Program and Value Stream Kanban systems (/program-and-value-stream-kanbans/).

Learn More

[1] Leffingwell, Dean. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley, 2011.

[2] Yakyma, Alex. Implementation Strategies for Business Epics (/Implementation Strategies for Business Epics)

Last Update: 23 May 2016

The information on this page is © 2010-2016 Scaled Agile, Inc. and is protected by US and International copyright laws. Neither images nor text can be copied from this site without the express written permission of the copyright holder. Scaled Agile Framework and SAFe are registered trademarks of Scaled Agile, Inc. Please visit Permissions FAQs (<http://scaledagile.com/permission-faq/>) and contact us (<http://www.scaledagile.com/permissions-form/>) for permissions.

FRAMEWORK

[Downloads \(/posters/\)](#)

[Latest Updates \(/blog/\)](#)

[SAFe 3.0 \(http://v3.scaledagileframework.com\)](http://v3.scaledagileframework.com)

TRAINING

[Course Calendar \(http://www.scaledagileacademy.com/events/event_list.asp\)](http://www.scaledagileacademy.com/events/event_list.asp)

[About Certification \(http://www.scaledagileacademy.com/?page=WhichCertification\)](http://www.scaledagileacademy.com/?page=WhichCertification)

[Become a Trainer \(http://www.scaledagileacademy.com/?page=becomeatrainer\)](http://www.scaledagileacademy.com/?page=becomeatrainer)

PERMISSIONS

[Permission FAQ \(http://www.scaledagile.com/permission-faq/\)](http://www.scaledagile.com/permission-faq/)

[Permissions Form \(http://www.scaledagile.com/permissions-form/\)](http://www.scaledagile.com/permissions-form/)

Usage and Permissions (/usage-and-permissions/)

PARTNER

Becoming a Partner (<http://www.scaledagile.com/become-a-partner/>)

Partner Directory (<http://www.scaledagile.com/listingcategory/directory/>)

Partner Event Calendar (<http://www.scaledagile.com/event-list/>)

GET SOCIAL

Twitter (<https://twitter.com/ScaledAgile>)

Linkedin (<https://www.linkedin.com/grps/Scaled-Agile-Framework-4189072?>)

YouTube (<https://www.youtube.com/user/scaledagile>)

SlideShare (<http://www.slideshare.net/ScaledAgile>)

RECENT POSTS

New Essential SAFe guidance article (/new-essential-safe-guidance-article/)

Nov, 30th 2016

SAFe Website Updates and new Glossary (/safe-website-updates-and-new-glossary/)

Nov, 29th 2016

End of life for SAFe 3.0 website and courseware 12/30/16 (/eol-safe3/)

Nov, 28th 2016

SCALED AGILE, INC

CONTACT US

5480 Valmont Rd., Suite 100

Boulder, CO 80301 USA

Email: support@scaledagile.com (<mailto:support@scaledagile.com>)

Phone: +1.303.554-4367

BUSINESS HOURS