

Chapter 8. Scaled Agile Framework

How does the Quality with Agile process scale in an enterprise environment?

In this chapter, I discuss how the Quality with Agile process may be integrated with an enterprise-wide agile development process, specifically the Scaled Agile Framework (SAFe).

At the time of writing, SAFe is the most popular method for expanding agile development (such as Scrum and Kanban) to include multiple development teams on the same project. SAFe expands on the Scrum process framework to encompass multiple development teams that may be working on the same or multiple related products. Its purpose is to fulfill the vision and mission of a single enterprise.

-
- ♦ I am not an expert and I have minimal hands on experience with SAFe. The content of this chapter is primarily gathered from my understanding the SAFe web site. Many of the statements in this chapter are derived from this understanding, and not explicitly stated on the web site. I make no claim that this understanding complies with the intent of the SAFe authors.
-

8.1 SAFe Overview

In this section, I provide a very brief overview of the Scaled Agile Framework. For a full description of SAFe, see <https://www.scaledagileframework.com/>

SAFe is defined at 4 levels of abstraction; portfolio, large solution, program and team levels. Levels may be combined or omitted to obtain 4 SAFe configurations. The appropriate configuration is dependent on the size of your development organization.

Essential configuration— Combines the program and team levels to define a process to be used when multiple development teams are working on the same product.

Large Solution configuration – Adds the large solution level to the essential configuration to define a process to be used when multiple products or services are delivered as a solution.

Portfolio configuration – Adds the portfolio level to the essential configuration to define a process that is used when development is driven at the enterprise level, by vision and mission statements.

Full configuration— Combines all levels to define a process for the complete enterprise from vision mission through solution delivery, which includes multiple products developed by multiple Scrum (or Kanban) teams.

Figure 61 shows the 4 configurations and their relationships to the 4 levels.

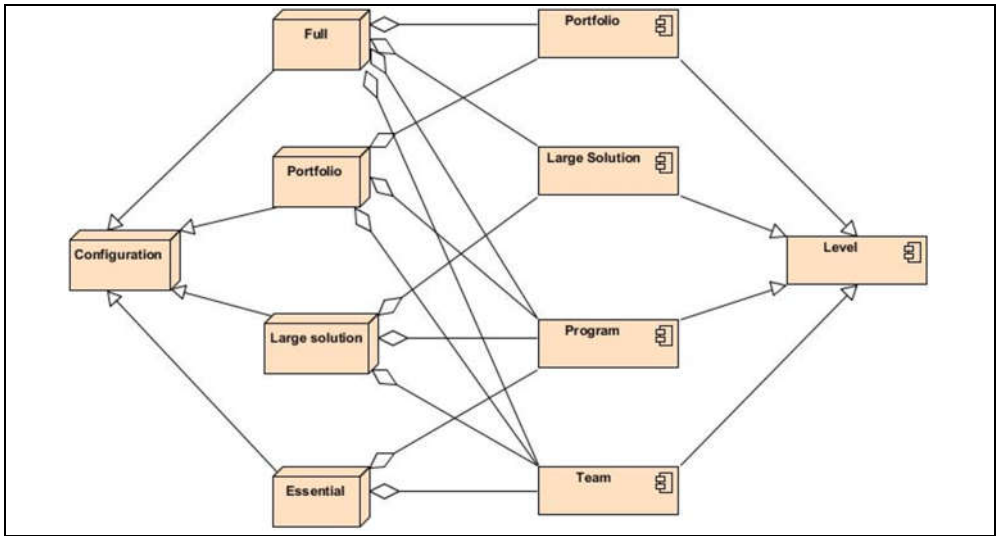


Figure 61 – SAFe Configurations

Full, Portfolio, Large Solution and Essential are the 4 types of SAFe configuration. Team, Program, Large Solution and Portfolio are the 4 levels of the process.

The full configuration contains the portfolio, large solution, program and team levels of the process.

The portfolio configuration contains the portfolio, program and team levels of the process.

The large solution configuration contains the large solution, program and team levels of the process.

The essential configuration contains the program and team levels of the process.

8.1.1 SAFe Backlogs

Each level of SAFe includes a backlog and work items that are managed within that backlog. Figure 62 shows the connection between backlog items at all 4 levels.

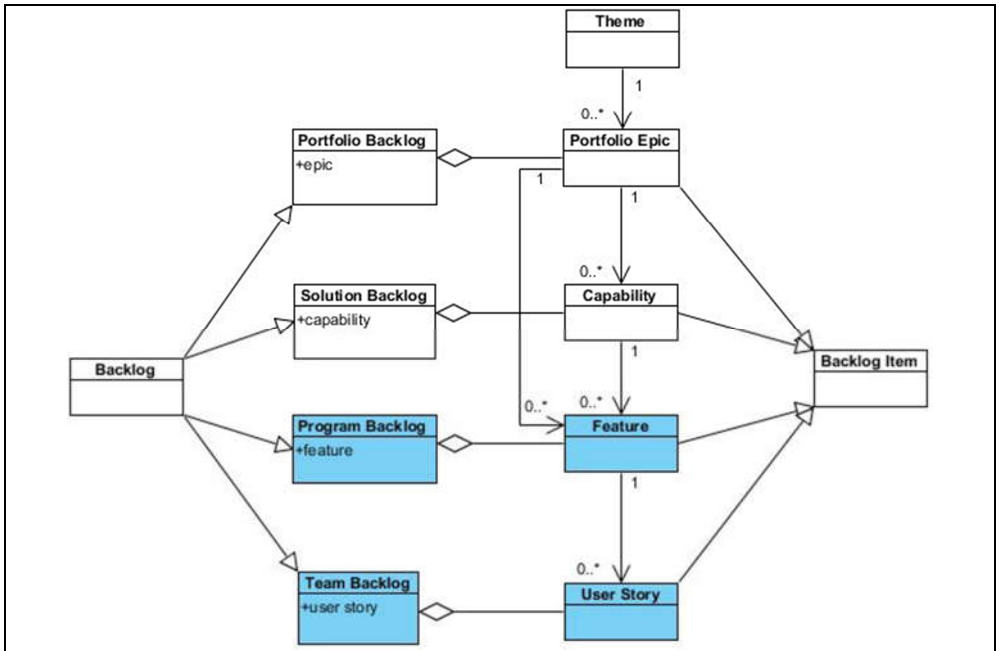


Figure 62 – SAFe Backlog Items

- ◆ Items with a white background do not capture technology. They may be solved using computers or they could become a manual process. A dark background (blue) indicates that these items capture system requirements (they are not manual).

The highest level artifact is a theme. A theme is a business objective or enterprise strategy. Themes are captured with portfolio epics in the portfolio backlog.

A portfolio epic (not to be confused with a user story type epic) is an enterprise initiative that requires budget approval. Portfolio epics are broken down into capabilities at the solution level, or features at the program level.

- ◆ Using the portfolio configuration, epics create features. Using the full configuration, epics create capabilities.

The solution backlog is used to manage capability work items. A capability describes a solution business need that may be a combination of manual and system operations.

A capability is broken down into features, each of which is implemented by a product in the solution.

The program backlog is used to manage features for a specific product. A feature is an epic sized user story that captures an automated objective that the product will fulfill.

A feature may be implemented by many development teams. The feature is broken down into user stories that are assigned to a specific development team.

-
- ♦ **Note that development teams are organized by product. It is assumed that a development team does not work on multiple products at the same time.**
-

These user stories are managed within a team backlog (similar to the product backlog described in the Quality with Agile process). In this description of SAFe, it is assumed (for the sake of simplicity) that user stories in the team backlog are always sprint sized.

-
- ♦ **There is no rule that states that the user stories in the team backlog have to be sprint sized. Features in the program backlog could be broken down into epic sized user stories and then assigned to the team backlog. In this situation, some sort of user story decomposition will need to be added to the team backlog activity.**
-

As with the product backlog, the SAFe backlogs contain a list of prioritized items, and a group of items that have not been prioritized, reviewed or approved.

Each SAFe level defines its own actors, activities and artifacts. These are described in the following paragraphs.

8.2 Portfolio Level

The portfolio level creates epics that are managed by a Kanban workflow. The result is a series of enterprise level steps for realizing a solution that delivers value to the business.

See <https://www.scaledagileframework.com/portfolio-level/> for a detailed description of the SAFe portfolio level.

8.2.1 Roles

The people involved at the portfolio level play the following roles:

- Enterprise Architect – identifies epics that describe technology used to realize a solution.
- Epic owner – equivalent to the business analyst at the enterprise level, this role is responsible for detailing and managing the content of an epic.
- Lean portfolio management – this team is equivalent to the product owner at the enterprise level, in that they identify themes, and they prioritize and organize epics in the backlog.

8.2.2 Artifacts

The following items are produced at the portfolio level:

- Epic – is a description of a solution that satisfies business strategies. A SAFe portfolio epic is defined at a much higher level of abstraction to that of a typical Scrum epic. The portfolio epic takes the form show in Figure 63.

For	<customer>
Who	<performs some activity>
The	<solution>
Is A	<a solution description>
That	<the value provided to the customer>
Unlike	<other potential solutions>
Our Solution	<why our solution is better than other solutions>

Figure 63 – Portfolio Epic Template

- Portfolio Backlog – is a prioritized repository for enterprise level epics.
- Theme – similar to a business need, a theme is a business strategy that is an input to the portfolio level.

8.3 Large Solution Level

The large solution is concerned with delivering multiple products to make a solution. The large solution level deploys solution builds to the customer. Large solutions do not require a portfolio backlog.

See <https://www.scaledagileframework.com/large-solution-level/> for a description of the SAFe large solution level.

8.3.1 Roles

The people involved at the large solution level play the following roles:

- Customer – is the recipient of the solution. Customers may be external or internal to the organization.
- Solution Management – is responsible for managing the solution backlog. They are similar to the product owner at the solution level.
- Solution Architect – is responsible for creating and maintaining solution user stories. They are similar to the business analyst at the solution level.
- Solution Train Engineer – manages the deployment of solutions comprising several products in a release. They are similar to the deployment manager at the solution level.

8.3.2 Artifacts

The following items are produced at the large solution level:

- Capability – is a description of the behavior of a solution. Capabilities may be implemented across many systems. A capability is a combination of

manual operations and multiple product features. The capability is broken down into features that are implemented by products that are part of the solution. Capabilities are managed in the solution backlog.

- Enabler – is a type of user story that supports the development of a solution. Enablers are used to perform exploration, architecture, infrastructure, compliance or any support to development that do not describe a capability of the solution. Enablers are managed in the solution backlog and prioritized alongside capabilities.
- Solution – is a service that is delivered to a customer in the form of products and services. A solution satisfies the capabilities in the solution backlog.
- Solution Backlog – is a repository for solution capabilities. SAFe suggests that the solution backlog contains items that are for a complete solution (no items are assigned to products or development teams).

8.4 Program Level

The program level of the process is concerned with delivering a product to the customer. The product is a combination of incremental builds that are developed by multiple development teams. An agile release train delivers product releases to customers. The agile release train performs integration of multiple incremental builds into a product release and deploys that release to the customer

The program level includes planned program increments, and product reviews. A product review is similar to a Sprint Review activity, but the review is of a product build that is produced by the agile release train.

The program level also includes the Deploy Build activity when the process has no large solution level (i.e. if you are using the portfolio or essential SAFe configuration).

-
- ♦ **When using the full or large solution configuration, the large solution level deploys product builds to the customer and the Deploy Build activity is not part of the program level.**
-

See <https://www.scaledagileframework.com/program-level/> for a description of the SAFe program level.

8.4.1 Roles

The people involved at the program level play the following roles:

- Business Owner –Is a stakeholder of the product. They are responsible for ensuring that the product satisfies the business needs. Business owners are customers at the program level.
- Product Management – Is responsible for managing the program backlog. They are the equivalent to product owners at the program level. The

difference between SAFe and Scrum is that there are many teams developing the stories in the program backlog. There may be many product owners prioritizing user stories in the product backlog. A product owner may be assigned to one or more development teams.

- Release Train Engineer – Is responsible for releasing a product that is the combination of several builds. The release train engineer plays a similar to the deployment manager. However whereas the deployment manager deploys a single build, at the program level a release is a combination of several product builds from different development teams.
- System Architect – Is responsible for the architecture of the product. All development teams working on the product use this architecture. The system architect is equivalent to the Quality with Agile solution architect role.

8.4.2 Artifacts

The following items are produced at the program level:

- Feature – A feature is a product user story that may be implemented by several development teams. A feature may be thought of as an epic in the program backlog. Similar to a Scrum epic, a feature is broken into user stories and those user stories are pulled into development team product backlogs.
- Program Backlog – Is a repository for product requirements at the feature level. SAFe suggests that when a feature is broken into user stories that those user stories are assigned to a development team. In this manner, the program backlog contains items that are for a single product. Items in the program backlog are not assigned to a development team.
- Release – Is a customer deliverable that is the integration of several incremental builds. This is similar to the release in the Quality with Agile process.

8.5 Team Level

The team level is identical to Scrum (or Kanban), with the exception that the product backlog is renamed the team backlog. The team backlog contains user stories that are derived from product feature stories in the program backlog.

Features in SAFe may be treated as the equivalent to epics in the Quality with Agile process. I.e. the feature is broken into user stories and those user stories are assigned to a team backlog.

I am going to assume that features in the program backlog are analyzed and broken into use cases. As user stories are created from the analysis of these use

cases, they are put in the program backlog. Only when they are assigned to a development team are they put into the team backlog.

In this manner, the team backlog only contains user stories. It does not contain epics.

When user stories are assigned to development teams, they are in an unprioritized state. The product owner still needs to approve and prioritize these user stories as they would for a normal Scrum product backlog.

User stories are created as a result of analyzing features. If the analysis is performed as described in the Elicit Business Needs activity, then these user stories are sprint sized. (As a result, the team backlog no longer needs to manage epic sized user stories.)

This has the effect that epics normally managed in a product backlog are now replaced by the features in the program backlog.

Figure 64 shows the structure of the team and program backlogs.

Program Backlog	
Prioritized Features	Prioritized User Stories
features	unassigned user stories
...	
Unprioritized Features	Unprioritized User Stories
features	unassigned user stories
...	
Team Backlog	
Prioritized User Stories	
assigned user stories	
...	
Unprioritized User Stories	
assigned user stories	
...	

Figure 64 – Program Backlog

The program backlog contains prioritized and unprioritized features and child user stories. The team backlog only contains prioritized and unprioritized user stories. (There are no epics in either backlog.)

In addition to user stories that are derived from features, both backlogs may include user stories that are enablers, are generated by the development team or are the result of defects.

Figure 65 shows how the team backlog may be combined with the sprint backlog to make a single team repository for user stories.

Team Backlog
Sprint
user stories
...
Prioritized Items
user stories
...
Unprioritized Items
user stories
...

Figure 65 – SAFe Team Backlog

In an alternative team backlog, user stories are prioritized by:

- Those in the current sprint.
- Those that have been prioritized for development.
- Those that have not been prioritized.

The team level focuses on development of a sprint. The activities performed at the team level are, Plan Sprint, Develop Sprint, Learn Lessons From Sprint and Hold Daily Scrum.

-
- ♦ **The Review Sprint activity is performed at the program and solution levels as product and solution demos.**
-

See <https://www.scaledagileframework.com/team-level/> for a description of the team level.

8.5.1 Roles

The roles at the team level are the same as those defined for Scrum (Scrum master, product owner and development team).

8.5.2 Artifacts

The artifacts of the team level are the same as those defined for Scrum (user story, backlog and incremental build).

8.6 SAFe Activities

Figure 66 shows the essential activities that are derived from the SAFe framework. These are organized by SAFe level (Portfolio, Solution and Program).

-
- ♦ **The team level activities are identical to Scrum.**
-

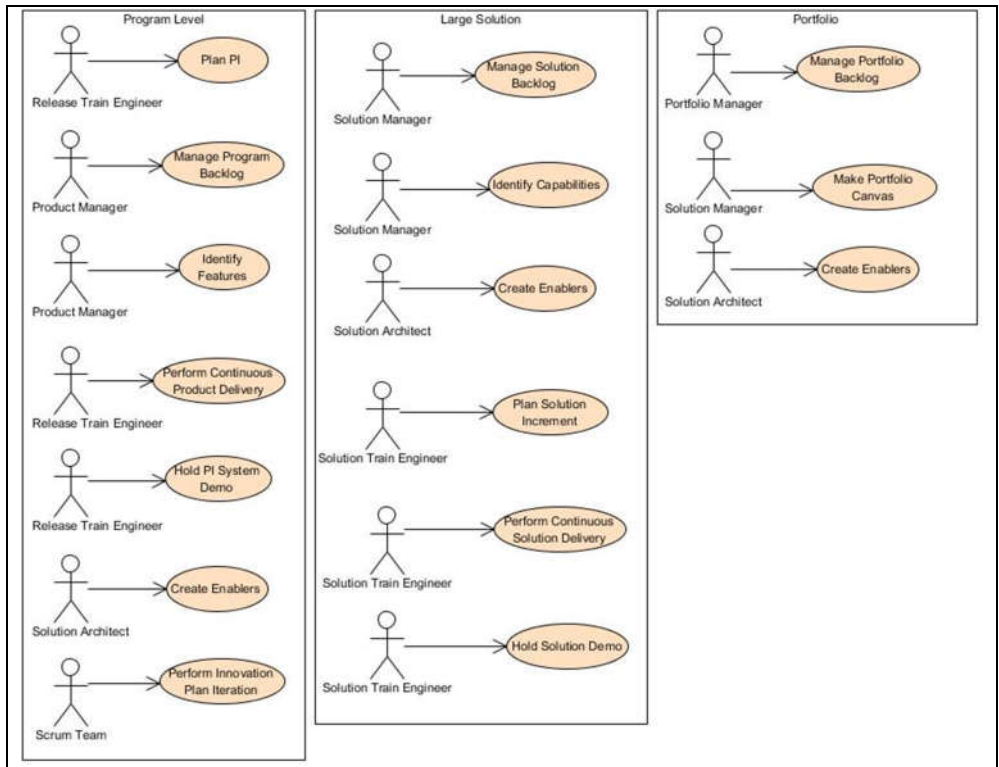


Figure 66 – Overview of SAFe Activities

In the following sections, SAFe activities are described in terms of their actors and how the activity is modified when the Quality with Agile process is applied to SAFe.

-
- ♦ To clarify, the activity descriptions are my understanding from the information available on the SAFe web site. I make no claim that this is 100% accurate.
-

8.6.1 Program Level

The program level activities are; Plan PI, Manage Program Backlog, Modify Features, Perform Continuous Product Delivery, Hold PI System Demo, Create Enablers and Perform Innovation Plan Iteration.

8.6.1.1 Plan PI (Product Increment)

Responsible actor – Release Train Engineer

This is a meeting between all developers and members of the product release train. Its purpose is to set objectives and prioritize features for the next release.

Applying the Quality with Agile process to this activity - It is similar to the Plan Sprint activity. The difference is that features are prioritized in the program backlog, as opposed to user stories being prioritized in a product backlog. There

are several development teams involved in this meeting, (unlike the sprint planning meeting where only 1 development team is involved). This meeting should be attended by all project team roles, including the solution architect, UI designer, business analyst, quality assurance and the writers.

8.6.1.2 Manage Program Backlog

Responsible actor – Product Manager

Similar to the Groom Backlog activity, this activity is concerned with cleaning and maintaining the program backlog. Features and user stories are prioritized. Conflicting and redundant items are resolved or removed from the program backlog. Changes to the program backlog may impact the various team backlogs.

Applying the Quality with Agile process to this activity – Grooming the program backlog includes the business analyst, product owner and quality assurance as secondary actors.

8.6.1.3 Identify Features

Responsible – Product Manager

This activity is similar to the Elicit Business Needs activity, but it applies to features at the program level. This activity is about identifying and creating features for the program backlog, and analyzing features to create user stories.

Applying the Quality with Agile process to this activity – Consider the program backlog feature to be equivalent to an epic in a traditional Scrum product backlog. Then the work performed by the business analyst is identical to that of the Elicit Business Needs activity. The responsible actor changes from the product manager to the business analyst. The business analyst elicits the feature details from the business owners and customers. They break down the feature into its component use cases. Use cases are broken down into user stories and acceptance criteria.

8.6.1.4 Perform Continuous Product Delivery

Responsible actor – Release Train Engineer

The release train engineer ensures that deployment of the product occurs according to a release schedule. The release train is an activity for continuously integrating builds from development teams.

Applying the Quality with Agile process to this activity – This activity replaces the Deploy Build activity. The deployment manager plays the role of the release train engineer. Instead of deploying incremental builds from a single development team, the deployment manager integrates builds from several development teams.

Quality assurance validates that the resulting product from this integration matches the test cases that were written for the product features.

If there is no large solution level, the writer will document user instructions for each release from the train.

The output from this activity is a deployed product release including an integrated software build, test results and if necessary, user instructions for using the product.

8.6.1.5 Hold PI (Product Increment) System Demo

Responsible actor – Release Train Engineer

A product increment is the result of integrating several builds from multiple development teams. The product increment satisfies the vision and objectives of the product increment planning meeting. The release train engineer is responsible for demonstrating the product increment to all interested stakeholders.

Applying the Quality with Agile process to this activity - This activity is equivalent to the Hold Sprint Review activity. The difference is that the demonstrated product the combination of builds from several development teams.

The release train engineer is a new role. All other actors are the same as described in section 7.4.8 - Hold Sprint Review.

8.6.1.6 Create Enablers

Responsible actor – Solution Architect

Enablers are types of user story that describe research, architecture, infrastructure and compliance work efforts. Enablers are generated at any level of SAFe and are generally created by architects and systems engineers. However, any SAFe role may have reason to generate a compliance or research enabler type user story.

Applying the Quality with Agile process to this activity – The Maintain Architecture activity generates similar architecture and infrastructure type user stories, and the solution architect is also the primary actor. The Maintain Architecture activity can be extended to include compliance and research type enablers. Any SAFe role can be a supporting actor.

Quality assurance generates test cases from enablers, at the solution, program and team levels.

The business analyst should understand the impacts that enabler type requirements have on the user stories. The business analyst may even be made responsible for creating enablers at the solution, program and team levels.

8.6.1.7 Perform Innovation Plan Iteration

Responsible actor – Scrum Team

This activity occurs once per program iteration. It is concerned with continuous collaboration and integration of development team builds. Members of all development teams that contribute to the product are involved to some extent in this activity.

Applying the Quality with Agile process to this activity - This is an additional activity that can easily be added to the process. The actors and work performed are as described by SAFe.

8.6.2 Large Solution Level

The large solution level activities are Plan Solution Increment, Manage Solution Backlog, Identify Capabilities, Create Enablers, Perform Continuous Solution Delivery and Hold Solution Demo.

8.6.2.1 Plan Solution Increment

Responsible actor – Solution Train Engineer

The solution train engineer ensures that deployment of a solution occurs according to a release schedule. The release train is an activity that continuously integrates builds from development teams.

Applying the Quality with Agile process to this activity –this is similar to the Plan Sprint activity. The development team, business analyst and product owners will be involved as secondary actors. The solution train engineer is a new role that ensures the success of the meeting.

8.6.2.2 Manage Solution Backlog

Responsible actor – Solution Manager

Similar to the Groom Backlog activity, this activity is about cleaning and maintaining priorities of features in the solution backlog.

Applying the Quality with Agile process to this activity – Grooming the solution backlog should involve the business analyst, product owner and quality assurance as secondary actors.

8.6.2.3 Identify Capabilities

Responsible actor – Solution Manager

This is equivalent to the Elicit Business Needs activity, but at the solution level. This activity is about identifying and creating capabilities for the solution backlog.

Applying the Quality with Agile process to this activity - A capability is the equivalent of an epic. It describes a business activity that may be automated by several products (some steps may be performed manually). The work performed by the business analyst in capturing capabilities is similar to that in the Elicit Business Needs activity. The business analyst elicits the requirements for the capability from the customers and analyzes the capability to create features. Features are the equivalent to system use cases. During analysis of the business use case business steps are performed manually or they are assigned to a product. So when an action in the business activity diagram is assigned to a system use case, the product that will implement that use case is also identified.

Consider the business to system mapping diagram in Figure 24. Actions that are identified as automated (dark blue background) are also assigned to the product that will implement their functionality.

Figure 67 shows that the systems have been added to this diagram to indicate which product will implement the use case.

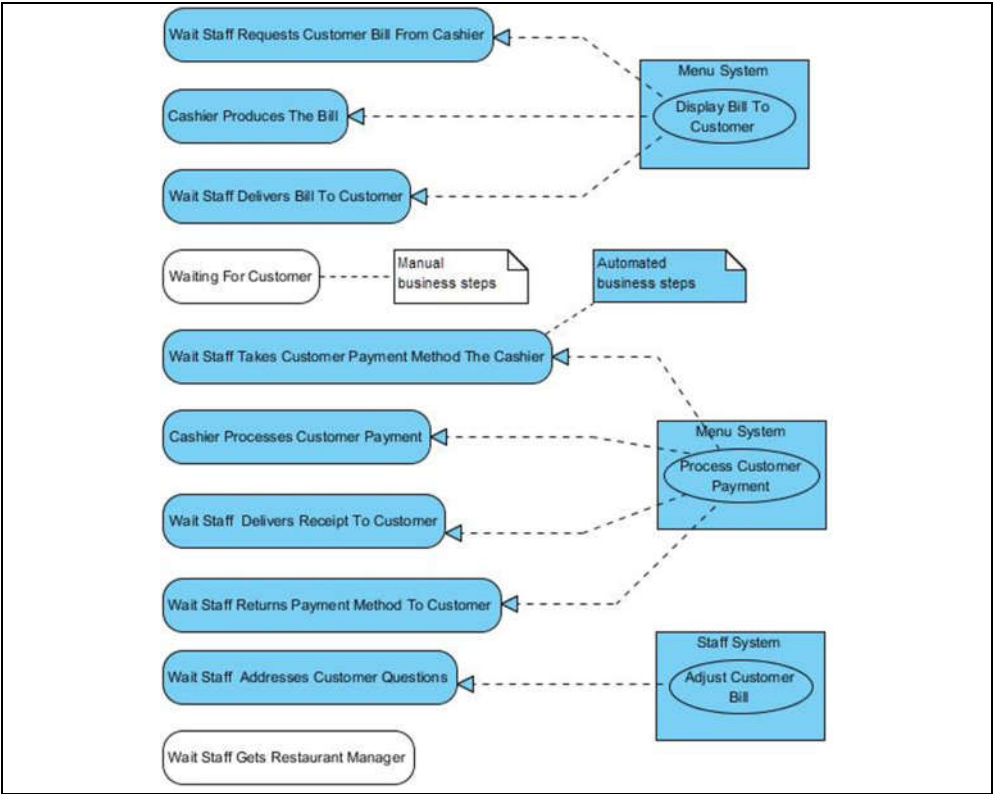


Figure 67 - Business To Multiple System Mapping Diagram

Assigning use cases to products requires knowledge of the system architecture. When the business analyst analyzes a capability, the solution architect needs to

assess the impacts to system architectures. Interfaces between systems are updated, and the system architect may have to redesign the architecture in order to accommodate new interfaces. Quality assurance uses capabilities to generate solution level test cases.

8.6.2.4 Create Enablers

This is the same as program level activity of the same name, in section 8.6.1.6.

8.6.2.5 Perform Continuous Solution Delivery

Responsible actor – Solution Train Engineer

The solution train engineer ensures that deployment of an integrated set of products and their instructions for use occurs according to a solution release schedule. The solution release train is an activity that continuously integrates products from development teams. These products are deployed to customers according to the solution release train schedule.

Applying the Quality with Agile process to this activity – This activity replaces the Deploy Build activity. However, it is more complicated because several products may be deployed in a single release. A solution release may change the steps in the business process. As a result, the user instruction manuals are even more important to a solution than a single product release.

The deployment manager may play the role of the solution train engineer and be responsible for the Perform Continuous Solution Delivery activity. In addition, quality assurance will validate the solution and write solution test cases. The writer documents user manuals for the solution as well as the individual products included in the solution.

8.6.2.6 Hold Solution Demo

Responsible actor – Solution Train Engineer

A solution increment is the result of integrating several product increments from multiple development teams. The solution increment satisfies the objectives of all product increment planning meetings for all products in the solution. The solution train engineer is responsible for demonstrating a solution increment to all interested stakeholders.

-
- ♦ **There is no equivalent to the PI planning meeting at the large solution level. Therefore, I assume that all product increment planning meetings occur at the same time, and their combined objectives are the objectives for a solution increment.**
-

Applying the Quality with Agile process to this activity - This activity is equivalent to the Hold Sprint Review activity, except that it demonstrates a solution that is the result of the integration of several products, delivered by several development teams.

The solution train engineer is a new role, but all other actors are the same as described in section 7.4.8 – Review Sprint activity.

8.6.3 Portfolio Level

The portfolio level activities are, Manage Portfolio Backlog, Make Portfolio Canvas and Create Enablers.

8.6.3.1 Manage Portfolio Backlog

Responsible actor – Portfolio Manager

Similar to the Groom Backlog activity, this activity is about cleaning and maintaining the priority of epics in the portfolio backlog.

Applying the Quality with Agile process to this activity - it is similar to the Groom Backlog activity. The business analyst and product owner may be asked to provide support, but the portfolio manager remains the primary actor.

8.6.3.2 Make Portfolio Canvas

Responsible actor – Solution Manager

The portfolio canvas is a visual chart for describing a business infrastructure, customers and finances. For more information see Business Model Canvas in Wikipedia or visit <https://www.scaledagileframework.com/portfolio-canvas/>

-
- ♦ I admit to never having used a business model canvas, so its description is out of scope for this book.
-

Applying the Quality with Agile process to this activity – there is no equivalent activity. This activity can be added without any change to the Quality with Agile process. As a minimum, I would expect that the business analyst and product owner provide support to this activity. The business analyst may even be expected to perform this activity (since it is described in the Business Analysts Body Of Knowledge).

8.6.3.3 Create Enablers

This is the same as program level activity of the same name, in section 8.6.1.6.

8.7 How The Quality with Agile Process Maps To SAFe

This section provides an overview of how the Quality with Agile maps to SAFe. Each of the artifacts, roles and activities in the Quality with Agile process are listed and their relationship to SAFe is described.

With the exception of the epic user story and the product backlog, the components that are defined by the Scrum framework are the same for both SAFe and Quality with Agile.

Other artifacts, roles and activities defined by the Scrum framework are not described in this section.

This section assumes SAFe is unchanged from the process described on the SAFe website, except where described below.

8.7.1 Artifacts

The Quality with Agile process introduces the following artifacts into SAFe.

8.7.1.1 Acceptance Criteria

In addition to acceptance criteria at the user story level, SAFe identifies acceptance criteria for features and capabilities.

Since I do not cover testing at the epic level in the Quality with Agile process, I am not going to attempt to describe acceptance criteria for capabilities and features. Instead, you can find out more about these test artifacts at <https://www.scaledagileframework.com/agile-testing/>

8.7.1.2 Backlogs

SAFe introduces several backlogs in addition to Scrum's product and sprint backlogs. The portfolio, solution and program backlogs follow the Kanban process as described by SAFe.

The team backlog contains user stories that have already been sized for a sprint and assigned to the product. (This is because the Elicit Business Needs process analyzes items in the solution or program backlog, not the team backlog.) Epic sized user stories that are assigned to a product can be managed in the program backlog. Here they are analyzed, broken into user stories and assigned to the team backlogs. Figure 68 shows the relationship between the backlogs.

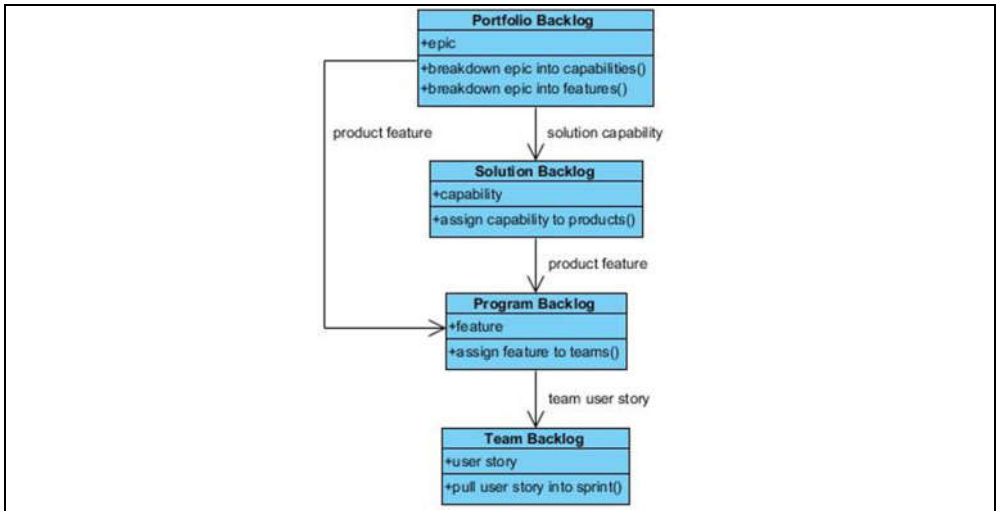


Figure 68 – SAFe Backlog Structure

The portfolio backlog contains portfolio epics. These are broken into solution capabilities. (If there is no solution level in the process portfolio epics are broken into product features.)

Solution capabilities are added to the solution backlog. These are analyzed by the business analyst and broken into features.

A feature is assigned to a product and added to the program backlog for that product. The business analyst analyzes features to break them down into user stories. These user stories are assigned to a development team.

User stories are added to the appropriate team backlog. From there they are prioritized and taken into team sprints.

-
- ♦ **Assignment of user stories from the program backlog to the team backlog should be done just-in-time for sprint planning. In this manner, if a sprint backlog is too full to take a high priority user story, it may be assigned to another development team that has available sprint capacity.**
-

Enabler (and team generated) user stories may be added to the appropriate team backlog at any time. Similarly, capabilities and features can be added to the appropriate backlog by solution and product management at any time.

Traceability is from portfolio epics to capabilities and features, from capabilities to features, and from features to user stories. This traceability needs to be maintained throughout the backlogs. If there is a change to a portfolio item, this may have a ripple effect on items in all backlogs through to the team backlog.

-
- ♦ **If updated, traceability may be maintained from capabilities through to user stories in the model, instead of the backlogs.**
-

8.7.1.3 Defects

It is unclear how SAFe handles defects that are discovered in a solution that has been released to the customer. As with Scrum, I assume that the Scrum team defines its own method for handling defects. Therefore, the method used by the Quality with Agile process for handling defects works just as well when applied to SAFe.

Defects may be detected by testing at any SAFe level; portfolio, solution, program or product.

- Portfolio Level – I assume that defects found with portfolio epics are fixed by editing the epic in the portfolio backlog.
- Solution Level – Defects may be generated as a result of testing the solution. It is assumed that these are added to the program backlog for the appropriate product.
- Program Level – Defects may be generated as a result of testing a product. These are added to the program backlog.
- Team Level – Defects are assigned to the team backlog in the same manner as any regular user story

The team that is responsible for handling the backlog at the appropriate level will manage the defect as if it is a regular backlog item.

-
- ♦ **For future discussion: I question if a team product backlog is actually needed in SAFe. User stories can be derived from features and managed in the program backlog, (the same as user stories and epics are managed in the product backlog). User stories may be assigned and re-assigned to development teams, while they are in the program backlog. Development teams could pull user stories directly from the program backlog into their sprint backlog during sprint planning.**
-

8.7.1.4 Business Use Case

Business use cases are captured at the program or solution level. As a result of analyzing a capability, a business use case may cover multiple products. At the program level, a business use case is for a single product. The business use case may be developed by multiple Scrum teams.

Mapping business use case actions to use cases is slightly different when analyzing a business use cases that encompasses multiple products. When identifying that a business step is to be automated, you now identify which product will satisfy that action performed by that step. If the action is automated by several products then the business process step is split into separate actions for each product.

In this manner, the analysis of a business use case produces system use cases, where each system use case applies to a single product.

8.7.1.5 Epic

The epic converts to a feature or capability in SAFe. These are captured in the solution or program backlog. They are traced from capabilities or portfolio sized epics.

-
- ♦ **I am assuming that the business analyst is knowledgeable enough to be able to break down features in the program backlog into user stories that are 1 sprint or less in size.**
-

As a result of epics being captured in the program or solution backlogs, the team backlogs will not contain any epics, only user stories.

8.7.1.6 Model

There is no mention of maintaining a model in SAFe (which is to be expected, since Scrum does not discuss models).

-
- ♦ **However, SAFe itself contains several models including a model of the requirements architecture.**
-

When applying the Quality with Agile process to SAFe, there is no change to the modeling guidelines. However, the structure of the model includes additional packages within the system view. The content of the Functional and Logical view is organized by product. A package is created for each product and use cases and classes are assigned to the appropriate product package.

Figure 69 demonstrates the structure of a model for a project using the SAFe Large Solution configuration.

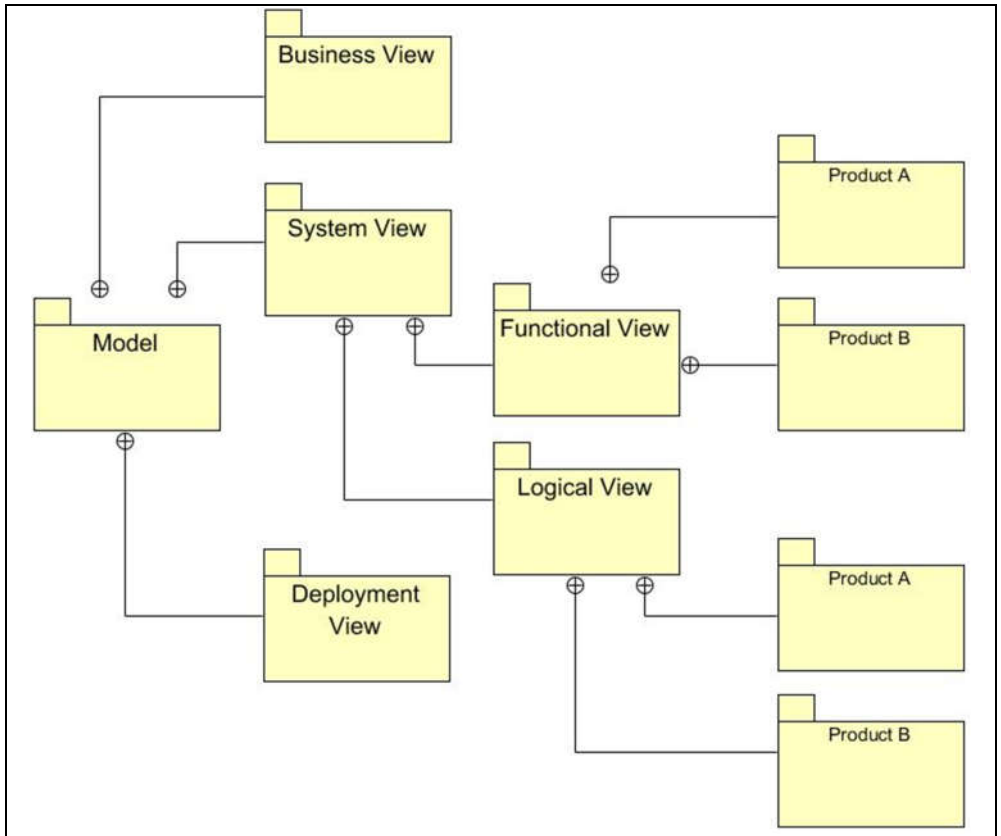


Figure 69 SAFE Model Structure

If using a large solution SAFe configuration, there is probably more than 1 business analyst on the project. The model is shared between all business analysts, with each given responsibility for certain product packages within the system view.

Separation of concerns within the business view is by business use case, with each business use case the responsibility of a single business analyst.

The model in the deployment view is the responsibility of the solution architect and it is shared by all users of the model. Any user of the model should be able to update the deployment view as the architecture for they are responsible, changes.

The alternative to the SAFe model structure is to create multiple models that are maintained by separate business analysts. Reasons why you may prefer several models are:

- Business analysts are using different modeling notations and model structures.
- The modeling tool does not support multiple users at the same time.

-
- ♦ **If your modeling tool does not support multiple users, you are probably using a diagramming tool, not a modeling tool.**
-

I highly recommend maintaining all product requirements in a single model. Having all products in one place discourages duplication, encourages consistency and helps with defining scope. In addition, there needs to be only 1 deployment package, which is shared by all business analysts.

Most modeling tools support multiple users by allowing packages within the model to be controlled as separate entities. These packages can be exported from the model, placed under configuration and version control and worked on as separate entities without impacting the work of other business analysts using the model. This allows branches of the model to be created. As with code branches, these model branches are merged back into the model at the appropriate time.

8.7.1.7 Need

Needs are addressed in SAFe as coming from various sources. Needs may be documented as capabilities at the large solution level or features at the program level. Working in a SAFe environment, as well as addressing the needs of the customer, the business analyst will elicit requirements from various internal stakeholders; including product management, business owners, architects, solution owners and epic owners.

The activity for eliciting business needs is the same as for the Quality with Agile process. The business analyst elicits stakeholder needs and writes features for the program backlog or capabilities for the solution backlog.

8.7.1.8 Product Release

Product deployment is an essential component of SAFe. It is part of the Agile Release Train (ART) activity. SAFe documentation goes into much more detail about deploying the product than this book does.

The major difference between what is described by the Deploy Build activity and SAFe, is that the ART deploys a product that is a combination of multiple build from different development teams. Although it is more complex, the ART activity is consistent with the intent of the Deploy Build activity.

-
- ♦ **If it turns out that Quality with Agile process is not consistent with the ART, then I am happy to alter the Deploy Build activity to make it so.**
-

More about deployment of multiple builds using ART can be read at <https://www.scaledagileframework.com/agile-release-train>.

8.7.1.9 System Use Case

Applying system use cases to SAFe does not change what is described in the Quality with Agile process. The model capturing the system use case is maintained at the program level. If using a large solution configuration, the functional view of the model contains system use cases for many products. In this situation, a package for each product is created within the functional view. Because a system use case is an autonomous artifact that impacts a single product, this arrangement of the system view allows system use cases to be organized by product (see Figure 69).

-
- ♦ **If a system use case happens to impact more than one product, then the business to system activity mapping needs to be revisited so that actions are mapped to no more than 1 product.**
-

8.7.1.10 Test Case

SAFe encourages best practices for testing at the development level. I assume that (as with Scrum) the development teams write the test cases for their own product builds. At the program level, test cases are written by the SAFe implementation team. At the solution level, I could find no description of testing.

After applying the Quality with Agile process to SAFe, a separate quality assurance team would be responsible for writing test cases at the solution level and test cases at the program level.

At the program level, test case steps are derived from the acceptance criteria for user stories that are taken into sprints. They are derived in the same manner as for a typical Scrum process.

At the solution level, test cases should be derived from capabilities in the solution backlog. These test cases are used to validate the integration of many products into a successful solution.

-
- ♦ **Deriving test cases from capabilities is outside the scope of this book.**
-

8.7.1.11 User Instructions

In SAFe, the activity for producing user instructions will be similar to the Quality with Agile process.

User instructions should be an output from the agile release train when there is no large solution configuration in the process.

At the large solution level, user instructions are released as part of the solution release train. In this situation, user instructions may be documented at both the product level and the solution level. Solution level documents will discuss how the products are combined to achieve a business objective.

8.7.1.12 User Interface Design

User interface design is an enterprise wide activity. UX standards and guidelines are applied to all products in a SAFe solution.

Individual screens are built for each product and implemented at the development level.

The user interface design activity is the same for SAFe as described in the Quality with Agile process.

8.7.2 Roles

The following roles can be added to SAFe and as a result of adopting the Quality with Agile process. This section summarizes how the roles are adapted for a SAFe configuration.

8.7.2.1 Business Analyst

In a SAFe environment, there are multiple business analysts (whereas in the Quality with Agile process a Scrum team is associated with 1 business analyst). As a business analyst on a SAFe project, you may find yourself sharing work on models, capabilities, features, with other business analysts. The business analyst is probably going to be writing user stories for multiple development teams. For example, a business need may be captured as a capability that produces user stories for more than 1 product. It may make sense for a single business analyst to create all the user stories for this single business need. They will therefore create features and user stories that are worked on by multiple development teams across multiple products.

-
- ◆ **In my experience, capabilities have always been decomposed into features and assigned to products prior to being assigned to a business analyst. This means that the business analyst only needs to be concerned with a single product. I do not recommend this practice.**
-

The business analyst role may be expanded to assist with writing epics and capabilities for the portfolio and large solution backlogs. Since features in the program backlog are derived from these backlog items, the business analyst can be expected to maintain traceability between features, capabilities and portfolio epics. (Traceability may be maintained in the model.)

Since Scrum epics are captured in SAFe as features in a program level backlog, the business analyst will maintain a model that encompasses multiple products. The model will clearly define the scope of each product and the interfaces between these products. As a result, the model will become much more complex in terms of its architecture and the deployment diagrams much more essential to development. In addition, if the model is shared with other business analysts a configuration management process and tools may be needed for the model.

The business analyst will contribute to the release train and should be expected to make a considerable contribution to the program increment meetings.

The BA may have to write acceptance criteria for capabilities at the solution level that encompass multiple products. This is in addition to product level acceptance criteria.

In summary, the business analyst contributes to the large solution program and team levels of SAFe. The business analyst may even be asked to contribute to the portfolio backlog by providing effort and product breakdown estimates for an enterprise level initiative. The business analyst is no longer a solo role for the project, and may have to work with several other business analysts, multiple product owners and multiple development teams.

8.7.2.2 Deployment Manager

This role is equivalent to the Release Train Engineer (RTE) (or Solution Train Engineer (STE) at the solution level). The RTE (or STE) is responsible for ensuring the smooth delivery of a product (or solution) at a predefined release schedule.

8.7.2.3 Product Owner

The product owner is elevated to the Program level where the product owner is responsible for the features (equivalent to Scrum epics) in the program backlog. Their role with Scrum development does not change, in that they still need to ensure that user stories are being pulled by the development team in the correct order. However, the product owner may be responsible for a backlog whose user stories are used by several SAFe development teams.

The product owner will contribute to product increment planning and may also be asked to contribute to capabilities in the solution backlog. They should probably be required to approve both a solution and product prior to their release.

-
- ♦ **Since there are no epic sized user stories in the team backlog, the business analyst can take sole responsibility for user stories at the development team level.**
-

For more on the product owner role with SAFe, see <https://www.scaledagileframework.com/product-owner/>

8.7.2.4 Quality Assurance team

As with Scrum, SAFe assumes that testing of user stories is performed during a sprint. Testers are part of the development teams (or release teams). Although SAFe encourages best practices for quality (such as test driven development), it does not encourage a separate quality assurance team that oversees all aspects of product delivery.

The Quality with Agile process identifies an externally controlled quality assurance team that contributes to almost every activity. When adding a separate quality assurance organization to a SAFe process, quality assurance will be involved with all activities in SAFe that contribute towards producing artifacts for a release. This includes the ART, the SRT, the program increment meetings and all reviews of program and solution backlog items. This work is in addition to the activities performed by quality assurance in the Quality with Agile process.

Because every SAFe configuration includes a program level, it is not necessary for quality assurance to validate builds at the team level. Test cases validate the solution or product at the large solution and program levels. At the program level, quality assurance validates a product against its features before it is released to a customer or to the solution release train. At the solution level, quality assurance validates that the complete solution satisfies its capabilities before being released to the customer.

The quality assurance team should not only be independent of the development team, but also removed from the program, solution levels.

-
- ♦ **I have no opinion as to whether quality assurance should be under the direction of management at the portfolio level.**
-

Quality assurance should be involved as a secondary actor at all levels of the SAFe process. At the portfolio level, quality assurance will need to understand the backlog items. At the solution, program or team level quality assurance reviews items and contributes items to the backlogs.

8.7.2.5 Solution Architect

This is the role is the same as that identified by SAFe of the same name.

See <https://www.scaledagileframework.com/system-and-solution-architect-engineering/> for a description of the SAFe solution architect role.

8.7.2.6 UX Designer

SAFe introduces the Lean UX guidelines, but does not identify a specific role for UX design. By applying the Quality with Agile process to SAFe, a consistent user experience would be designed at the solution level, no matter how many products there are at the program level. If there is no large solution level to your SAFe process, then enterprise wide shared UX design resources operate at the program level. The UX designer role ensures that all user interfaces in the ART (and the SRT) are consistent with company standards and best practices.

8.7.2.7 Writer

As with user instructions and documentation in general, this role is not discussed in SAFe. The writer still uses user stories and the output product

release in order to document user instructions. With SAFe, the difference to the writer is that the user stories are from several product backlogs, the writer may document several products and there may be additional documentation at the solution level.

8.7.3 Activities

The following sections give a brief overview of changes to each of the Quality with Agile activities, when they are applied to a scaled agile framework process.

8.7.3.1 Elicit Business Needs

Elicitation occurs at the solution or program levels.

If SAFe includes a large solution configuration, then business needs contribute to capabilities in the solution backlog. Capabilities are analyzed to produce features that are assigned to products.

(See section 8.6.2.3, for more information about analyzing capabilities.)

At the program level, features in the program backlog are analyzed as if they are regular epics. The resulting user stories are assigned to development teams. A single feature may produce user stories for more than one development team.

-
- ♦ **An example is the situation when a feature is to be supported by more than 1 operating system. Different development teams implement the feature for each operating system.**
-

(See section 8.6.1.3 for more information about analyzing features.)

The process described for eliciting business needs is the same for SAFe as it is for the Quality with Agile process. However there the following differences in project structure may need to be taken into account:

- Business activity can be assigned to several products.
- System use cases are captured as features in a program backlog.
- The team backlog does not need to manage epic sized user stories, since these can be captured as features in the program backlog.
- User stories include an additional 'team' attribute.

8.7.3.2 Groom Backlog

The groom backlog activity does not change in a SAFe environment. However, the same activity is applied to multiple backlogs: the portfolio backlog, solution backlog, program backlog and team backlog.

Changing an item in the portfolio, solution or program backlog will have implications on items that are traced from that item in other backlogs. (Figure 68 shows how items are traced from one backlog to another.) For this reason it

probably makes sense to regularly groom all backlogs in 1 effort, starting with the highest level backlog first and the team backlog last.

The program and team backlogs may be groomed more frequently, since they include user stories created by the development team. These items do not have to be traced from capabilities or features.

8.7.3.3 Maintain Requirements

The guidelines for maintaining requirements are the same for SAFe as for the Quality with Agile process.

Complexity is introduced because multiple business analysts may be working on the same model. The model can be portioned into packages such that each package is assigned to a single business analyst. Packages may be exported from the model so that the contents can be worked on without interfering with dependencies to other model elements.

Assuming that a single system use case is developed by the same Scrum team, (and that Scrum team includes exactly 1 business analyst) the model may be partitioned into packages that contain system use cases assigned to that team's business analyst.

-
- ♦ **A system use case contains a set of closely related actions that provide benefit to a product release. As such, I highly recommend that a system use case is wholly developed by the same Scrum team.**
-

The logical view is partitioned into systems. Each system package contains the logical view for a single product. If multiple business analysts are working with the same product, they will need to coordinate their efforts. Model configuration management tools that provide 'check-in' and 'check-out' capabilities are very useful when sharing a model with other business analysts.

8.7.3.4 Design Architecture

In SAFe, the Design Architecture activity produces enabler user stories at the solution, program and team levels. See section 8.6.1.3 for more information about enabler user stories.

8.7.3.5 Define User Experience

A user interface is generally implemented by a single development team. Therefore, the Define User Interface activity should be the same for a SAFe development effort.

8.7.3.6 Plan Sprint

This activity is the same in both SAFe and QAP.

8.7.3.7 Develop Sprint

This activity is the same in both SAFe and QAP.

8.7.3.8 Review Sprint

This activity is the same in both SAFe and QAP.

8.7.3.9 Learn Lessons From Sprint

This activity is the same in both SAFe and QAP.

8.7.3.10 Test Build

In SAFe quality assurance testing occurs at the program and solution levels. At the program level, the Test Build activity is the same as described in section 7.4.10.

Solution level quality assurance testing is performed against the capability backlog items. Capabilities are captured in the model as business use cases. Testing business use cases is out of scope for this book.

8.7.3.11 Deploy Build

The Deploy Build activity is replaced by the SAFe activities describing the product release train and the solution release train; Perform Continuous Product Delivery at the program level and Perform Continuous Solution Delivery at the solution level.

8.7.3.12 Document Release

At the solution level, all products that are part of a release are documented with instructions for the users. Release notes are produced for each product. Additional documentation may include instructions completion of a business process using multiple systems.

If there is no solution level then the Document Release activity is the same as described in QAP. Documents are released at the program level. Since each scrum team contributes to a single product, user instructions are not required at the team level.

8.7.3.13 Hold Daily Scrum

This activity is the same for both SAFe and QAP.

8.8 Summary

The Quality with Agile process scales well with SAFe by contributing at the appropriate levels. The roles that are identified in the Quality with Agile process perform the same activities and deliver the same artifacts. Some of these artifacts are already included in SAFe; others are new to SAFe. Some activities are a little more complex, because there may be multiple actors playing the

same role on the project. Most Quality with Agile roles provide additional support to SAFe activities and may take responsibility for some its deliverables.

Afterthought

Concerning the relationship between the program and team backlogs:

Figure 64 shows a program backlog containing only features and a team backlog containing only user stories. This implies that all user stories must be assigned to a team. Your organization may find it useful to create user stories without having a team assignment. These user stories are maintained in the program backlog and can be still prioritized without having an assigned development team. In this case, the program backlog resembles the Scrum product backlog. It contains both prioritized and unprioritized features and user stories.

Figure 70 shows this alternative program backlog structure.

Program Backlog	
Prioritized Features	Prioritized User Stories
features	user stories
...	
Unprioritized Features	Unprioritized User Stories
features	user stories
...	

Figure 70 – Alternative Program Backlog

Like the Scrum program backlog, features are separated from user stories. Features are initially unprioritized which means that their child user stories will also be unprioritized.

As a feature is prioritized, its child user stories may be prioritized. Prioritized user stories are not assigned to a development team (although it may be useful to identify the best candidate development team for the user story).

This method allows the development team to pull their user stories from the program backlog, as opposed to being assigned user stories. Once a user story is assigned to a team backlog, it is automatically removed from the program backlog.