Table of Contents

How Do We Write Good Features? 3
Audience / Consumers of Features 3
Feature Elements 3
Feature Development Process 4
Want to Know More? 5

How Do We Write Good Features?

SAFe describes a hierarchy of artifacts that describe systems behavior: Epics, Features and Stories.

A feature is a description of a desired service of behavior of a system that addresses one or more user needs. The development of business features is the responsibility of the Product Manager. Assistance is available from train level business analysts and teams. A feature should be sized such that it could be completed within a Program Increment.

Enabler Features are meant to enable and support the development of business initiatives. These are generally created by system architects.

Features and enablers are prioritized and sized using Cost of Delay and WSJF techniques; and are the primary input to Program Increment Planning. A feature is only used through the end of the PI in which it is completed.

Audience / Consumers of Features

- Product Managers: The Product Managers associated with each train review features during their cost of delay discussions to evaluate the relative value and other cost of delay components.
- Release Train Engineer: The RTE develops an understanding of the work that is needed to help guide decision making during PI execution.
- Systems Architect: The system architects evaluate the feature and develop an estimated effort that is used to calculate the WSJF score.
- Product Owners: The Product Owner is the primary recipient of the feature from the assigned agile team. They use the features as a primary input to developing related stories with the agile team. They also use their understanding of the prioritization of the features to make decisions on the scheduling of team work.
- Agile Teams: The agile teams review features during PI planning and during sprint planning and work with the Product Owner on user stories as they plan and commit to related work.

Feature Elements

The goal is to have sufficient detail in the feature to enable the agile team to break it down into stories. The following information is typically captured for a feature as well as for recommendations for capturing and working with features:

- Title: The title briefly conveys the intended business purpose or outcome, using business terms. The title should not specify a particular technology or solution (unless its inclusion is necessary for understanding). It should complete the sentence “I wish the system would …” and should start with a verb. An example is: Provide 2017 SBC Template. Prefixes can be used if they aid in sorting and

grouping of features within an epic. Use brackets [] to distinguish prefixes when used.
- Description: This provides more detail about the business purpose of the work, within the constraints of the scope of the feature. One format is: The purpose of this feature is to <insert high level business requirement> so that… <business outcome or benefit>. Additional business requirement details and benefits should follow within the description.
- Benefits hypothesis: Describe the value expected from the business from this feature (not the entire epic). This should be in the form of a hypothesis - we think we want this, but we don't necessarily know this. It should include both tangible and intangible outcomes. If the value is only pertinent in conjunction with other features – that should be articulated. If available: include pertinent quantitative information about number of users impacted, dollars saved, FTE savings, etc. This is the place to reference, or provide a link to, any pertinent ROI or tangible benefits from the epic business case, if applicable.
- Acceptance Criteria: Describes how we know the feature is done; it describes the expected result in a way that can be verified. This field is often referenced to reflect the scope of the desired end result of the feature. It is used by system architects and agile teams in their estimating.
- Epic / Portfolio Item: This field ties the feature to the associated epic which is often a way of describing the funding source (e.g. project).

Additional considerations to include in the description field:

- Critical dates: Dates may be driven by compliance, contracts or other business events. Clarify between ‘must have’ dates and other types of deadlines.
- Risk reduction and Opportunity Enablement: What else does this feature do for the business? Does it reduce the risk of this or a future delivery? What else does this feature do for the business? Is there value in the information we would generate when we work this feature? Will the feature open up new business opportunities?
- Dependencies: Include any known dependencies: this can include other features, enablers or vendors
- Line(s) of business impacted
- Not all features will be perfect: Judgement is needed to evaluate what is ‘good enough’.
- If the feature requests updates to an existing process, program or interface – include any detail you have, such as interface number. However, this is not required.
- If the feature is an upgrade – indicate what environments are impacted (likely to need system architect involvement)
- Clarify whether the intent is to migrate all the way to production or some other integrated environment.
- Indicate if training or knowledge transfer is part of the acceptance criteria
- Determine if the end user area wants to perform User Acceptance Testing (UAT) before a feature is considered done.
- Unordered List ItemCan the feature be released on its own, or will it be bundled with others for UAT and/or deployment?

Feature Development Process

There is a cadence to feature development that aligns with the program increment schedule. Here is a

view of the order you will typically see:

1. Feature definition: The goal is to have drafts prepared by week a certain week of the previous PI
    1. New features
    2. Consider Likely carryover features: Reviewed and updated as needed
    3. Consider features in the backlog (not in current PI)
2. COD sessions: After availability of the features
3. Cross Train Review of features
4. Preview of features with product owner
5. RTE and Product Manager: Agree on features to bring into PI planning
6. PI Planning: Features may be changed during PI Planning
7. Follow-up to ensure that features reflect objective commitments
8. If the team committed to a portion of the feature – the feature should be updated, and a new feature created (in the backlog) to reflect the remaining functionality.
9. After the feature is completed by the team(s), it is closed.

Want to Know More?

- How Do We Improve How We Write Requirements and Goals?
- http://www.scaledagileframework.com/features-and-capabilities/

FAQ, Features, Good, SAFe, Scaled

From:
https://www.hanssamios.com/dokuwiki/ - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:
https://www.hanssamios.com/dokuwiki/how_do_we_write_good_features

Last update: **2025/07/29 12:37**