Requirements Management applied in an agile Project Management environment

AUTHOR

Franco (Frank) Curtolo, *CEng*
Principal Consultant, Program Planning Professionals Ltd, Frank.Curtolo@pcubed.com

CATEGORISATION

Accessibility – Practitioner

Application – Good Practice

Topic – Agile Systems Engineering; Project Management

ABSTRACT

This paper looks at how the Requirements Management process of Systems Engineering may benefit from some of the aspects derived from developing systems within an agile Project Management environment, using as example, the Scaled Agile Framework® of © Scaled Agile, Inc. The aim is to see if some of these agile techniques can enhance the Systems Engineering approach.

Systems Engineering (SE) is well suited to develop large, complex products[1] in a project environment where the requirements can be defined and fixed upfront. In fact, SE relies on an initial, well-defined End User's need specified in for example, a User Requirements Specification (URS), to establish an initial requirements baseline which is used to drive the rest of the development process. However not all development projects happen that way. Sometimes the End User's need is not clear, or cannot be well defined upfront, thus not allowing it to be captured in a fixed requirements baseline. Furthermore, the project environment may need to accommodate rapid change, both in the End Users' need and in the technologies available to satisfy this need.

In such a project environment, an agile development approach, as described in the Scaled Agile Framework® (SAFe®), may be more suitable since it allows for incremental delivery of the product and change to requirements. This can accommodate undefined End User needs, rapid change in requirements and allow for the introduction of innovation during the development and implementation stages.

The purpose of this paper is to describe how some of the techniques of the Scaled Agile Framework®, could be applied to enhance the Requirements Management process of Systems Engineering.

INTRODUCTION

With the need for well-defined requirements upfront; the Requirements Management (RM) process of SE has received a lot of attention. It has become a well-established and well defined, standardised process. Perhaps the best examples of this can be found in: -

- **ISO/IEC/IEEE 15288:2015** - *Systems and software engineering — System life cycle processes* [1],
- **ISO/IEC 29148:2011** - *Systems and software engineering — Life cycle processes — Requirements engineering* [2].

By performing processes like Business or Mission Analysis; Stakeholder Needs and Requirements Definition; and System Requirements Definition, the RM process ensures that the End User's needs are comprehensively analysed and well defined at the start of product development.

[1] The term product in this article, is used in the universal context of the ISO 9001:2000 definition to include both goods and services. i.e. a product as the outcome of a project.

RM in the SE context can be summarised using the Vee model taken from the **SYSTEMS ENGINEERING HANDBOOK** [3]. This depicts a sequential development life cycle (most suited to hardware products) with development and definition occurring top down (left leg of the Vee) through the System hierarchical levels and implementation and integration following a bottom up sequence (right leg of the Vee) from Components, into Subsystems and System levels. The main attributes of this model with respect to requirements are:

Decomposition

The Decomposition or allocation of requirements from the highest system hierarchical level to lower level system elements occurs top down along the left leg of the Vee. System requirements are broken done into Subsystem and Component requirements which act as the inputs for the development of these lower level system elements. Requirements baselines are established at each system hierarchical level.

Traceability

With the breakdown of requirements, Traceability between Lower level system elements requirements and Upper level system element requirements must be maintained so that changes in requirements and changes in the configuration of the system elements, can be managed using the Configuration Management process. This is achieved by means of defined development baselines in which requires are fixed.

Verification and Validation

At each level of the system hierarchy the realized (or implemented) solution (right leg of the Vee) is checked to confirm that it meets its specified requirements (in the left leg of the Vee). This is termed Verification when demonstrating conformance to specification (i.e. is the system *built right*) and Validation when the complete system performance is demonstrated to satisfy the End Users' need (i.e. is the *right system* built). Validation takes place with the system is in its operational environment.

Requirements Baselines

At each system hierarchical level, the System, Subsystem and Component requirements are captured in artefacts (called specifications) which are used to define the system development baselines, and act as: -

- input to the next level of system development or project phase, during design and definition,
- criteria against which the realised system solution is measured, during integration, Verification and Validation.
- mechanisms for managing the configuration of the product during development and implementation and for maintaining traceability of requirements.

APPROACH

Scaled Agile for large scale development

The Scaled Agile Framework® (SAFe®) is a knowledge base for implementing Lean-Agile development [4]. It is based on applying an Agile Mindset which is described (by Susan McIntosh) as *'the set of attitudes supporting an agile working environment. These include respect, collaboration, improvement and learning cycles, pride in ownership, focus on delivering value, and the ability to adapt to change'*. It is based on three bodies of knowledge, namely, Agile Development; Systems Thinking; and Lean Product Development.

SAFe® Requirements Model

To support bringing the benefits of Lean and Agile development to large scale, complex systems, SAFe® also has a requirements model. This model, taken from the Scaled Agile Framework® version 4.5 in reference [4], is shown in Figure 1 below.

The Scaled Agile Framework® requirements model is applied 'iteratively' in small, continuous development cycles, in contrast to the sequential development life cycle of the SE Vee model of reference [3] described in the Introduction.

In SAFe® the 'requirements' are contained in the Epics, Capabilities, Features and Stories, instead of the specifications of the SE requirements model.

An Epic is a container for a Solution development initiative large enough to require analysis, the definition of a Minimum Viable Product (MVP), and financial approval prior to implementation. Implementation occurs over multiple Program Increments (PIs) and follows the Lean start-up cycle of 'build-measure-learn'.

A Capability is a higher-level solution behaviour that typically spans multiple Agile Release Trains. Capabilities are sized and split into multiple Features.

A Feature is a service that fulfils a stakeholder need. Each Feature is described by a phrase, benefit hypothesis and acceptance criteria, and is sized or split as necessary to be delivered by a single Agile Release Train (ART) in a Program Increment (PI), iterative cycle.

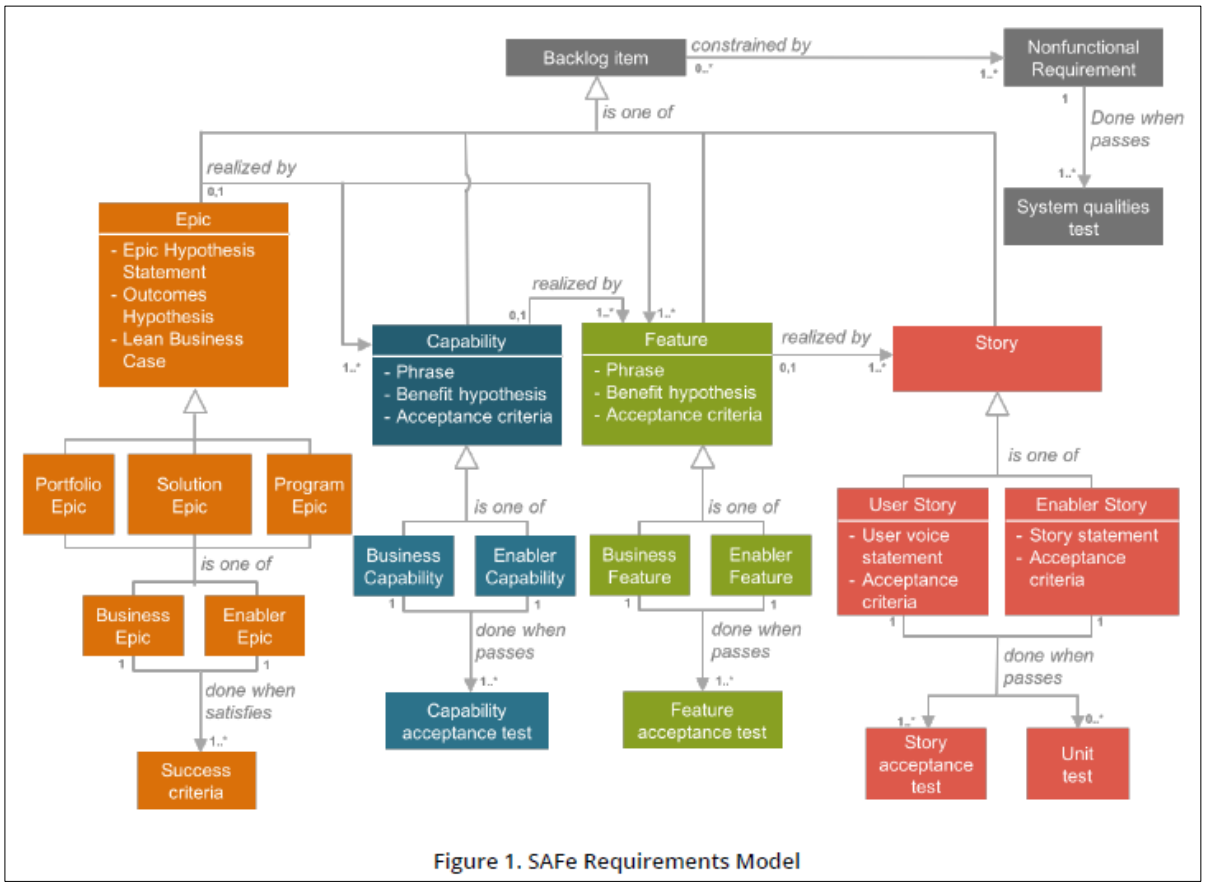

Figure 1. SAFe Requirements Model

Figure 1 SAFe Requirements Model, taken from Ref [4]

Stories are short descriptions of a small piece of desired functionality, written in the End User's language of a user-voice statement and acceptance criteria. Agile Teams implement small, vertical slices of system functionality and are sized so they can be completed by the Team in a single Iteration.

The above artefacts replace the requirements specifications used in SE with new paradigms based on Lean-Agile development.

The SAFe® requirements model also has hierarchical structure providing the decomposition and traceability of 'requirements' from Epics and Capabilities at Business Level (the SAFe® Portfolio and Large Solution Levels) to Features and Stories at implementation or in SAFe® language, Program and Team Levels. A broad and generic mapping of the hierarchical levels and the type of 'requirements' artefacts used in Systems Engineering and SAFe® is shown in Table 1 below:

SYSTEMS ENGINEERING		SCALED AGILE FRAMEWORK	
LEVEL	DOCUMENT	DOCUMENT	LEVEL
System of Systems	Stakeholder/User Requirements Specification (StRS/URS)	Epics	Portfolio
System	System/Software Requirements Specification (SyRS/SRS)	Capabilities	Large Solution
Subsystem	Subsystem Specification (SSS)	Features	Program
Component	Component Specification	Stories	Team

Table 1 Systems Engineering and Scaled Agile hierarchy and requirements artefacts

Since SE and its RM process is used in large system development, it seems appropriate to look at SAFe® and its requirements model for possible features of an Agile Mindset which could enhance the SE RM process. The features of SAFe® most appropriate for System development occur primarily at the Large Solution and Program Levels, with the development of System-of-Systems aligning with the SAFe® Portfolio Level and Team level aligning mostly with Component level development.

The SAFe® Lean-Agile Principles taken from [4] which are most applicable to RM are:

- Principle #1 - Take an economic view
- Principle #2 - Apply systems thinking
- Principle #3 - Assume variability; preserve options
- Principle #4 - Build incrementally with fast integrated learning cycles
- Principle #5 - Base milestones on objective evaluation of working systems
- Principle #7 - Apply cadence, synchronise with cross-domain planning
- Principle #8 - Decentralized decision-making

These principles are contained in the SAFe® techniques, taken from [4], and described in the next section.

RESULTS

So, what can RM learn from applying SAFe®, an Agile Mindset and the Lean-Agile Development of products?

Continuous Delivery Pipeline

The Continuous Delivery Pipeline represents the workflows, activities, and automation needed to provide a continuous release of value to the End User. The pipeline consists of four elements: Continuous Exploration (CE), Continuous Integration (CI), Continuous Deployment (CD), and Release on Demand. The first three elements of the pipeline work together to support delivery of small batches of new functionality [or parts of the product], which are then released in accordance with demand.

Perhaps the first, and most obvious benefit of a Continuous Delivery Pipeline when it comes to RM, is that the End User gets to see if the product is satisfying its requirements, early, and on a continuous basis. In other words, is the Supplier providing value and supplying what *is of value* to the End User. If not, it provides benefit also to the Supplier who can adjust what it delivers incrementally, and so saving effort and resources associated with End User's non-acceptance and rework of large parts or the whole product, at the end.

This early confirmation of meeting requirements is especially important in regulated industries such as Nuclear, Aerospace, Pharmaceutical and Financial, where a Regulator can reject a product due to regulatory non-compliance, irrespective of whether the End User accepts that the product does not fully meet its desired values. Incremental, continuous delivery also allows for incremental, continuous regulatory review and acceptance eliminating the need for a bow wave of work load on the regulatory authorities to review and accept at the end of development or implementation, all in one go.

A second, and perhaps less obvious benefit is that incremental, continuous delivery, enables the introduction of innovation by allowing changes to the requirements of the lower level system elements and thus introduction of alternative solutions. This can add value to the overall product by implementing newer technology solutions to the lower level system elements, to meet the overall End User need.

Release on Demand

Releasing parts of the realised product on Demand, or on a suitable cadence, offers the End User an opportunity to see how the value of the development is happening. The cadence interval can be made the regular monthly or quarterly project review meeting, or other suitable regular interval on large projects if the rate of progress is not as fast as on smaller product development projects.

One could also use the concept of 'Release but with Features turned off' to distinguish between interim releases of the product which are 'For Information Only' and actual contractually agreed deliverables specified on Contract Documents Record Lists (CDRLs) of the project. In this way, it can be made clear to the End User which deliverables are linked to contractual terms (payment and timescales) and which are not.

Minimum Marketable Feature and Minimum Viable Product

A Minimum Marketable Feature (MMF) or a Minimum Viable Product (MVP) represent the minimum effort necessary to sufficiently validate or invalidate the stated (i.e. specified) benefit or value which a Capability (or group of Features), will achieve. This provides a means for getting feedback as early as possible to test ideas and make decisions more quickly. It is supported by a Lean business case to state the costs involved, technology and architectural enablement and infrastructure required to deliver it.

The concept of a Minimum Marketable Feature (MMF) or a Minimum Viable Product (MVP) can prove useful in ringfencing a set of Key End User requirements by identifying the requirements associated with a particular MMF or MVP. This in turn can be used to fix a set of requirements for incremental development of parts of a system, in a modular fashion. It also provides a mechanism for an End User to identify which requirements can or cannot be subject to change by the Supplier and to be used to introduce innovation to parts of the system during its development.

Cadence and Program Increment Planning

The Program Increment (PI) planning technique is a cadence-based, face-to-face event that serves as the heartbeat of the Agile Release Train (ART), aligning all the teams on the ART to a shared mission and vision. PI planning has a standard agenda that includes a presentation of business context and vision followed by team planning breakouts—where the teams create their Iteration plans and objectives for the upcoming PI.

System development, especially large-scale systems have long development stages and often value cannot be seen for long periods of time. Using cadence to provide continuous feedback of progress, at regular intervals to the End User, helps demonstrating achievement of value. For large projects at the Large Solution layer, SAFe® introduces the concept of Pre-PI and Post-PI planning sessions to integrate several ART's and Suppliers and engage with Owners and End Users. This is especially useful for large projects which have long development timescale, and the value can only really be demonstrated in a completed system.

The regular PI planning event could also offer an ideal opportunity to review and confirm End User requirements and to align plans and project objectives for the next step in the development stage. The face-to-face environment would increase communication between the Suppliers' teams developing the product to meet these requirements and the End User who has specified these requirements. Thereby providing a mechanism for requirements clarification and requirements modification to increase value through introduction of innovative solutions.

System and Solution Demos

The System Demos and Solution Demos events offer the ideal forum for reviewing and confirming if a system (or part thereof) meets its requirements. Often the complete system is required to demonstrate system performance and for system Validation. With modern techniques such as Virtual Reality and Augmented Reality a 'virtual' system can be used to demonstrate to the End User how the actual system will perform.

Other means such as simulations, modelling and rapid prototyping (3D printing) can also provide means for demonstrating conformance to requirements before the actual system gets implemented. Model Based Systems Engineering is rapidly gaining ground because of this ability to Verify and Validate before system implementation, i.e. to V&V during the definition and development stage in the left leg of the Vee model.

System and Solution Demos also offer an opportunity to communicate with the End User on potential changes to initial requirements which could be innovative changes which increased value. This is particularly attractive if it can be done on a virtual (i.e. not yet realised) system or part thereof.

Although using virtual systems and models for V&V has its limitations since one is not measuring the actual, realised system or parts, it can provide a very useful means not only to predict acceptance of the system but also to study 'what-ifs' should the requirements of the system be changed. With these tools, such evaluations can be done 'off-line' without impacting the development of the system itself.

Inspect and Adapt workshop

The SAFe® concept of an Inspection & Adapt workshop held at the end of each Program Increment and consisting of PI System Demo; Quantitative Measurement and Retrospective and problem-solving Workshops, offers an ideal event to also review the requirements of the product. Not only to demonstrate that the requirements are met by means of the System Demo or Solution Demo, and Quantitative Measurement but also to change requirements to introduce innovation or additional value. Since attendance at the I&A, wherever possible, includes all the people involved in developing the system it can provide for quick and complete communication of such requirements changes.

DevOps

DevOps, which is a combination of two words, *development* and *operations,* provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a Solution. Enterprises which apply SAFe®, implement DevOps to break down silos and empower each Agile Release Train (ART) and Solution Train to continuously deliver new Features to their End Users. Over time, the separation between development and operations is significantly reduced, and trains operate within an automated Continuous Delivery Pipeline. This mechanism seamlessly defines, implements, and delivers solution elements to the End User, without predefined sequential handoffs or excessive external production or operations support.

In applying SE in a waterfall project environment, there are distinct sequential hand-offs between the End User inputting its need (i.e. requirements) to the Supplier for development and implementation and then the Supplier handing back the developed product (i.e. solution) to the End User for acceptance testing, operations and maintenance. If these single hand-off events, which are often also interpreted by both parties as 'keep your hands off my part of the product lifecycle', can be broken down into smaller, continuous collaborative events, as in DevOps, it would lead to a smoother transition from the development and implementation stages, to the operations and maintenance stages. The Validation part of RM would happen continuously rather than just at the end of implementation, once the product enters it operational environment. The DevOps environment also lends itself to an easier management (identification, definition, control of change, demonstration of conformance) of requirements between those acquiring the system and those supplying it, during the development and implementation stages.

CONCULSION

The application of the SAFe® techniques mentioned above has influenced the SE Vee lifecycle model in the following ways:

- It closes the angle of the two legs of the Vee thereby reducing the system development time and so providing a quicker time to market the product.
- It promotes the iteration of requirements allocation and requirements verification at each system level. i.e. several smaller iterations of Design-Implement-Test rather than just a single big one.
- It promotes a continuous demonstration of satisfying the End Users' need during development and allows for early response when this is not achieved.
- It promotes a staged approach in fixing requirements i.e. several Vees instead of only one which allow for the introduction of innovation and for situations where the complete End Users' need is not defined upfront
- It promotes a progressive hand-over from the Development and Implementation to the Operations and Support life cycle stages, i.e. a staged handover along the right leg of the Vee.

When it comes to Requirements Management, Systems Engineers often like to quote a famous software engineer, Edward V. Berard, who said: -

> *'Walking on water and developing software from a specification are easy if both are frozen'*

SAFe® and Lean-Agile Development say: -

> *'Yes, that's true, but one needn't freeze the whole lake upfront, all in one go; continue freezing only that part needed to take the next steps forward.'*

In that way one can continuously adjust one's direction (i.e. be agile) and in so doing take the most effective and efficient (i.e. Lean) approach to manage the requirements. That is what Agile Requirements Management is all about, and in applying it one achieves two very important values of the Agile Manifesto [5]:

- Customer collaboration, and
- Responding to change

Obviously, the extent to which these SAFe® and Lean-Agile Development techniques can be applied to the development and implementation of systems depends on the specific type of product or project. Software lends itself more to this than Hardware but applying an Agile Mindset is relevant to all.

ACKNOWLEDGEMENTS

To my colleagues and agilista project managers at Pcubed – Ben Beavers and Rhodri Cavi, who enlightened me with their extensive knowledge of Agile practices and SAFe® training.

REFERENCES

1. **ISO/IEC/IEEE 15288:2015** - *Systems and software engineering — System life cycle processes*
2. **ISO/IEC 29148:2011** - *Systems and software engineering — Life cycle processes — Requirements engineering*
3. **SYSTEMS ENGINEERING HANDBOOK** – *A Guide for System Life Cycle Processes and Activities – 4th Ed* International Council on Systems Engineering (INCOSE)
4. http://www.scaledagileframework.com/
5. http://agilemanifesto.org/

ABOUT THE AUTHOR

Franco (Frank) Curtolo CEng, MSc, MINCOSE, is a Principal Consultant at Program Planning Professionals Ltd, specialising in integrating Systems Engineering and Project/Program Management. He has over 35 years' experience in the roles of both Senior Systems Engineer and Senior Project Manager, in engineering systems and facilities and managing international, high technology, multi discipline, development and EPCM projects, in the Nuclear (Reactor & Fuel), Defence (Navy & Aerospace), Science (Radio Astronomy) and Architecture-Engineering-Construction fields.

* … * … *