# Solutions to development of model-based standards [using STEP as an example] – **Brandon/Marion**

**Content Moved to Integrated Doc**

## Adoption of Agile Framework [II.A, II.B] - **Brandon**

### What and How to use Issue Management [backlog]

### What and How to use Increment Planning and Agile Release Trains

**SOLUTIONS**

According to survey results of a study by the Institute of Electrical and Electronic Engineers “a majority of respondents' organizational units are using agile and/or lean methods (58%). Furthermore, lean appears as a new player, being used by 24% of respondents, mainly in combination with agile (21%).” These statistics reinforce the increased development rate of industry towards a trend of rapid incremental development as they strive towards enabling the digital threads for their enterprises.

Many development teams still use traditional methods to create their products. These traditional methods drive the teams to long phases of requirements documentation, product development, integration, review, and publication. Many organizations have adopted agile as a means to shorten the development cycle and provide a usable product to the users faster. These organizations have realized XX benefits. (Rico, David F., 2008)

Agile, itself, is not a new concept. There are many examples of projects using agile concepts such as rapid application development, prototyping, and many others. However, since the creation in 2001 of the Agile Manifesto, there have been many related implementations and development of new methods. The manifesto describes 12 principles – but there are three that hit home for the development of model-based standards. The first is “Deliver working software frequently.” The second is, “Working software is the primary measure of progress." And the third is, “At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behavior accordingly.” Agile Manifesto (Beck et al., 2001) Note, the term “software” can be replaced with any product such as “data models” or published data standards. Agile methods include [but not limited to] practices such as Extreme Programming, Scrum, KANBAN, Backlog Management, and Continuous Delivery.

In addition to the specific methods used, there are some overarching frameworks that help tie them all together to help large organizations implement at different scales. These frameworks include Scaled Agile [SAFe], Disciplined Agile Delivery [DAD] and Large-scale Scrum [LeSS]. While some have criticized SAFe as being too prescriptive, it has seen double the implementations by industry over LeSS and DAD. (KnowledgeHut, 2018)

The FULL SAFe framework by Scaled Agile provides the most comprehensive configuration for deployment. (Scaled Agile, 2018a) Each project team must analyze their needs and identify which component(s) of the framework that will enable them to meet their goals. Scaled Agile has documented case studies that bring real business results, including happier, more motivated employees, faster time-to-market, increase in productivity, and defect reductions. (Scaled Agile, 2018b)

While SAFe provides many tools to implement agile – this paper will discuss only a few that can bring benefit to the development teams of model-based standards: Backlog Management, Program Increment Planning and Agile Release Trains.

***Backlog Management***

Having a backlog isn’t the same as managing the backlog. In the course of STEP development, most teams use a system such as Bugzilla to store all the issues. Teams will assign, at bulk, issues to the next milestone and perform a quick reassessment few times during the length of the project. Steps a team can take to actively manage the backlog is to establish, and make it a priority, a Product Owner/Manager Role. The person in this role will be primarily responsible for why, when and what of the product that the development team will deliver. Each team should have a person designated in this role and actively manages the backlog by reprioritizing, adjusting, grooming and adding to the backlog. This will prevent the backlog from getting too big or out of date. It will also provide reliable work that is ready for the team to assign to a sprint. Many tools exist to manage backlogs in an agile framework, such as Atlassian’s JIRA, Micrsoft’s TFS, VersionOne or PivotalTracker.

***Agile Release Trains***

Using the Scaled Agile definition and framework, an Agile Release Train [ART] is used to group agile teams that operate to develop and deliver “one or more solutions in a value stream.”(“Agile Release Train – Scaled Agile Framework,” n.d.). The ART is a virtual organization that breakdowns the existing silos for development, testing, and publication. The ART is lead by a Release Train Engineer [RTE] but has other important roles such as a Product Manager, System Arch, Business owners/Customers.

For the development of Model-Based Standards like ISO 10303 Application Protocols, an Agile Release Train can be used to create/revise an edition of an AP, such as a new edition to AP242. Then an agile team can be created for the different domains that will deliver capabilities, such as Electrical Wire Harness [EWH], Product Manufacturing Information [PMI] or Additive Manufacturing [AM]. These agile teams would be each have a Scrum Master, Product Owner and a set of developers. 

An ART can address one of the biggest problems with multiple teams. That is a regular integration. Recently integration issues with AP242 e2 teams could have been avoided with synchronized and integrated development iterations. Each team may adopt agile, but can and sometimes do operate with different velocities and do not sprint together. The ART addresses that problem by employing systems thinking and applying an operating cadence and synchronization that enables all the teams to sprint together while integrating. There is not limit to how many trains can operate together. The concurrent development of AP242e2, AP239e3 and AP243e1 could be managed as multiple trains.

***Program Increment Planning***

Using the Scaled Agile framework, an ART delivers value in a fixed timebox called a Program Increment. The planning that increment is critical to the synchronization of the teams on the train. This synchronization will facilitate planning and limit work in progress.

The RTE, and team, will decide on the amount of iterations [sometimes called sprints] that will be performed in the Increment. All agile teams will follow the same schedule and operate harmoniously. At the beginning of each Increment, all of the teams will have a planning event where they decide their velocity, estimate and plan the work packages. For voluntary teams or teams with resources that are only available part-time, this planning event is critical to establishing the team resource availability and velocity. There are many estimation techniques. The team must avoid detailed analysis and estimation and instead adopt a method like Planning Poker, T-Shirt Sizes, Dot Voting or something similar where the process is quick and relative.(“7 Agile Estimation Techniques – beyond Planning PokerâAMIS Oracle and Java Blog,” n.d.)

After each iteration there is a Plan, Do, Check, Adjust [PDCA] activity where the team can make changes to the plan. At the end of the Increment a product is available to the customer. This aligns back to the principles of Agile from the Manifesto. For Model-Based Standards this could be a draft standard or schema of the AP that is made available for testing in an Implementor Forum.

**REFERENCES:**

7 Agile Estimation Techniques – beyond Planning PokerâAMIS Oracle and Java Blog. (n.d.). Retrieved July 19, 2019, from https://technology.amis.nl/2016/03/23/8-agile-estimation-techniques-beyond-planning-poker/

Agile Release Train – Scaled Agile Framework. (n.d.). Retrieved July 19, 2019, from /agile-release-train/

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved July 18, 2019, from https://agilemanifesto.org/

KnowledgeHut. (2018, May 11). LeSS Vs SAFe: Which Certification Should You Choose And Why? Retrieved July 18, 2019, from https://www.knowledgehut.com/blog/agile/less-vs-safe-which-certification-should-you-choose-and-why

Rico, David F. (2008). *What if the ROI of Agile vs Traditional Methods*.

Scaled Agile. (2018a). *Introducing SAFe 4.6*.

Scaled Agile. (2018b, November). *SAFe 4.6 IntroductionâOverview of the Scaled Agile Framework for Lean Enterprises*.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**BLUE = Content**

**GREEN = Content linked to MBSD or STEP**

**RED = References**