**NIST Advanced Manufacturing Series XXX-X**

**Agile for Model-Based-Standards Development**

Brandon Sapp

Melissa Harvey

Sylvere Krima

Marion Toussaint

This publication is available free of charge from:

https://doi.org/10.6028/NIST.AMS.XXX-X



**NIST Advanced Manufacturing Series XXX-X**

**Agile for Model-Based-Standards Development**

Brandon Sapp

The Boeing Company

*Business Architecture Integration/  
Industry Data Standards*

Melissa Harvey

The Boeing Company

*Business Architecture Integration/  
Industry Data Standards*

Sylvere Krima

Engisis, LLC

Bethesda, MD

*Office of XXXXXXXXXX*

*First Operating Unit*

Marion Toussaint

Systems Integration Division

Engineering Laboratory

This publication is available free of charge from:

https://doi.org/10.6028/NIST.AMS.XXX-X

September 2019



U.S. Department of Commerce

*Wilbur L. Ross, Jr., Secretary*

National Institute of Standards and Technology

*Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology*

Certain commercial entities, equipment, or materials may be identified in this

document in order to describe an experimental procedure or concept adequately.

Such identification is not intended to imply recommendation or endorsement by the

National Institute of Standards and Technology, nor is it intended to imply that the

entities, materials, or equipment are necessarily the best available for the purpose.

**National Institute of Standards and Technology Advanced Manufacturing Series XXX-XX**

**Natl. Inst. Stand. Technol. Adv. Man. Ser. XXX-XX, NNN pages (Month YYYY)**

**This publication is available free of charge from:   
https://doi.org/10.6028/NIST.AMS.XXX-X**

Abstract

Industry is undergoing a wide scale digital revolution as they strive towards enabling their digital enterprises. This paradigm shift from unstructured data sources and paper based artifacts to Digital Twins places key importance on the interoperability of the software applications and information systems involved. One of the main responses to this new paradigm is the use of neutral model based data standards. However as this digital strategy represents an increased development rate for industry it must be ensured that as their foundation standards are able to support rapid incremental development as well. Examination of the current standards development process points to two primary roadblock inhibiting this advancement: (1) inflated standards development time lengths and (2) quality escapements in the published standards. An analysis of the key contributing factors to these roadblocks and of available optimization opportunities has resulted in a recommendation to pursue the adoption of an agile framework and toolchain by standards development bodies. This proposed solution includes Scaled Agile [SAFe], Disciplined Agile Delivery [DAD] and Large-scale Scrum [LeSS] and offer a means to shorten the development cycle and provide a usable product to the industry faster.

Key words

model based standards development; agile

Table of Contents

[1. Introduction 1](#_Toc15302639)

[1.1. Statement of Industry 1](#_Toc15302640)

[1.2. Information Standards Support of Business Needs 1](#_Toc15302641)

[2. Issues in Current Development Lifecycle 1](#_Toc15302642)

[2.1. Development Time Length 1](#_Toc15302643)

[2.2. Quality/Completeness of Standard 2](#_Toc15302644)

[3. Solution Concepts [using STEP as an example] 3](#_Toc15302645)

[3.1. Adoption of Agile Framework 3](#_Toc15302646)

[3.1.1. Backlog Management 4](#_Toc15302647)

[3.1.2. Agile Release Trains 4](#_Toc15302648)

[3.1.3. Program Increment Planning 5](#_Toc15302649)

[3.2. Improved Tool-Chain 6](#_Toc15302650)

[3.2.1. Requirements Management and Traceability 6](#_Toc15302651)

[3.2.2. Solutions 6](#_Toc15302652)

[4. Benefits 7](#_Toc15302653)

[4.1. Benefits to MBS Developer 7](#_Toc15302654)

[4.2. Benefit to Industry/Enterprise 7](#_Toc15302655)

[5. Conclusion 8](#_Toc15302656)

[References 9](#_Toc15302657)

[Appendix A: Supplemental Materials 10](#_Toc15302658)

[Appendix B: Term Bank 10](#_Toc15302659)

[Appendix C: Change Log 10](#_Toc15302660)

List of Tables

[Table 1. Title. 1](#_Toc480895143)

List of Figures

[**Fig. 1.** This is caption text for Fig. [1]. 1](#_Toc15302575)

[**Fig. 2.** This is caption text for Fig. [2]. 2](#_Toc15302576)

[**Fig. 3.** This is caption text for Fig. [3]. 3](#_Toc15302577)

[**Fig. 4.** This is caption text for Fig. [4]. 5](#_Toc15302578)

[**Fig. 5.** This is caption text for Fig. [5]. 5](#_Toc15302579)

[**Fig. 6.** This is caption text for Fig. [6]. 7](#_Toc15302580)

# Introduction

TBD

## Statement of Industry

Digitization of knowledge, promoted by recent advances in the information and communication technology field, has led to a digital revolution of manufacturing. While paper-based information artifacts (such as 2D drawings) are being replaced by their digital twins, unstructured data sources (spreadsheets, text documents, email, …) are slowly being replaced by structured data models embedding different types of information (design, manufacturing, inspection, …). This digitization supported by a formal representation of the data is a key enabler of the Model-Based Enterprise paradigm, in which information management is digital and can be automated.

Through this new paradigm, manufacturing data processing can now leverage modern computing techniques and be made faster, consistent, and more accurate, offering a better insight and leading to a Smarter Manufacturing. This new approach requires that structured digital product data be shared and exchanged among numerous engineering and business software applications, and information systems [ref]. Through its entire lifecycle, a product generates an enormous amount of data in response to different processes (e.g., design, manufacturing, distribution) and needs (e.g., technical, commercial, regulatory). This data is often critical to every organization that plays a role in the product lifecycle. This is where the organizational contribution and value reside.

Due to the complexity of a product and its lifecycle, the interoperability of the software applications and information systems involved is key to support the organizational collaboration required to successfully design, manufacture and support a product. A lack of interoperability is not only costly in term of time and money [ref to NIST report] but can also impede this organizational collaboration and hinder future work and opportunities.

For Model-Based Enterprises to collectively and collaboratively achieve Smart Manufacturing and support the full (and often decade long) product lifecycle, an open and harmonized representation of the information must be used and shared through the supply chain. One of the main responses of this new product data management strategy is the use of product information standards. These information standards are a key to integrating, exchanging, and accurately interpreting the different product models and data produced during the product lifecycle across multi-disciplinary systems. The different IT systems consuming and generating product data need a common language to exchange information and understand each other. Information standards provide an agreed upon data format and definitions to exchange and share knowledge about a product and its lifecycle.

## Information Standards and How They Support Business Needs

### What are Standards

Today, organizations are more aware of the importance of the digital integration and exchange of information assets. Hence, information standards are developed to ensure an optimal interoperability and compatibility between information assets in order to exploit and process these assets in a consistent fashion. An information standard is a formal definition, agreed upon by a community of experts, on how to represent and process domain-specific information. This agreement can be represented as information models, a computer-interpretable representation. Thus, information from different providers can be represented, exchanged and integrated together, seamlessly. Information standards are used in many domains, such as healthcare, security, or manufacturing.

One of these standardization efforts is the ISO 10303 *Automation systems and integration – Product data representation and exchange* standard series, informally known as STEP[ref].  STEP started in the early 90s and is one of the information standards that enables product lifecycle collaboration. It includes a set of product data representations and implementation methods to exchange product data. Each of these product data representations is domain-specific and developed by industrial experts to enable standard-based product data exchange. To support its requirements, the STEP community developed its own data modelling language ISO 10303-11 *Part 11: Description methods: The EXPRESS language reference manual* and a file format to represent STEP data, called ISO 10303-21 *Part 21: Implementation methods: Clear text encoding of the exchange structure*. As XML emerged as a common file format for data exchange, ISO 10303-28 *Part 28: Implementation methods: XML representation of EXPRESS schemas and data, using XML schemas* was developed as another way to represent STEP data.

It aims to provide a complete and unambiguous description of manufacturing products, usable throughout their life cycle, regardless of the IT support used. The scope of STEP is much broader than other existing computer-aided design (CAD) data exchange formats. It is intended to handle a wide range of product types (electronic, mechanical, fiber composites, ships, architectural, process plant, furniture,...) and cover all life-cycle stages (design, analysis, planning, manufacture,...). [1]

The development of standard is a relatively long process and it includes a lot of people from different organizations working together. As mentioned before, the members of a standard WG work for different organizations and their contribution to its development is voluntary. Therefore, the resources available depend on the experts’ schedules and their organizations’ needs, which makes the standards development process long, irregular and difficult to plan. Moreover, some standards are complex due to their architecture and application domains. In the case of STEP, its development is one of the largest projects that ISO has ever carried out and six hundred people from many different countries have been involved for the last thirty-five years[1]. STEP is a product-centric standard that covers so many areas which has created a big and multi-disciplinary community and needs. STEP covers a lot of domains that need to collaborate and a lot of subjects that need to be combined. In course of its development, the STEP architecture has changed. The objective of this new modular architecture is “to enable the more efficient implementation and deployment of STEP standards without changing the fundamentals of the current technical architecture”[2], i.e. to create new modules by reusing, integrating and extending existing Application Protocols (APs). Thus, APs are “more interoperable, easier to understand and manage, and quicker to develop” [3].

### Who Uses Them

### Who Creates Them

Developing a standard involve different stakeholders. There are four major types of stakeholders : the Sponsor, who is the entity that creates a request for standardization; the Standards Development Organization (SDO) that supervises the standard development process and, supports the publication and the maintenance of the standard; the Standard Board, which consists of members of the SDO, reviews and approves standards projects before their publication; and finally the Working Group that is composed of domain experts, implementers, end users, standards experts and technical solution experts. The role of the WG is to support the development of the standard by writing the standard draft, working on the standard maintenance and developing products that implement the standard. For example, STEP is developed and maintained by ISO and more precisely, by the ISO technical committee TC184 that deals with automation systems and their integration, and the sub-committee SC4 that manages the industrial data. The STEP working groups are composed of experts from government agencies, the automotive industry, the aerospace industry and the nautical industry for instance. All of these experts work together to meet the specific needs of their industry.

### Development Lifecycle of Model-Based Standards [1.B.3]

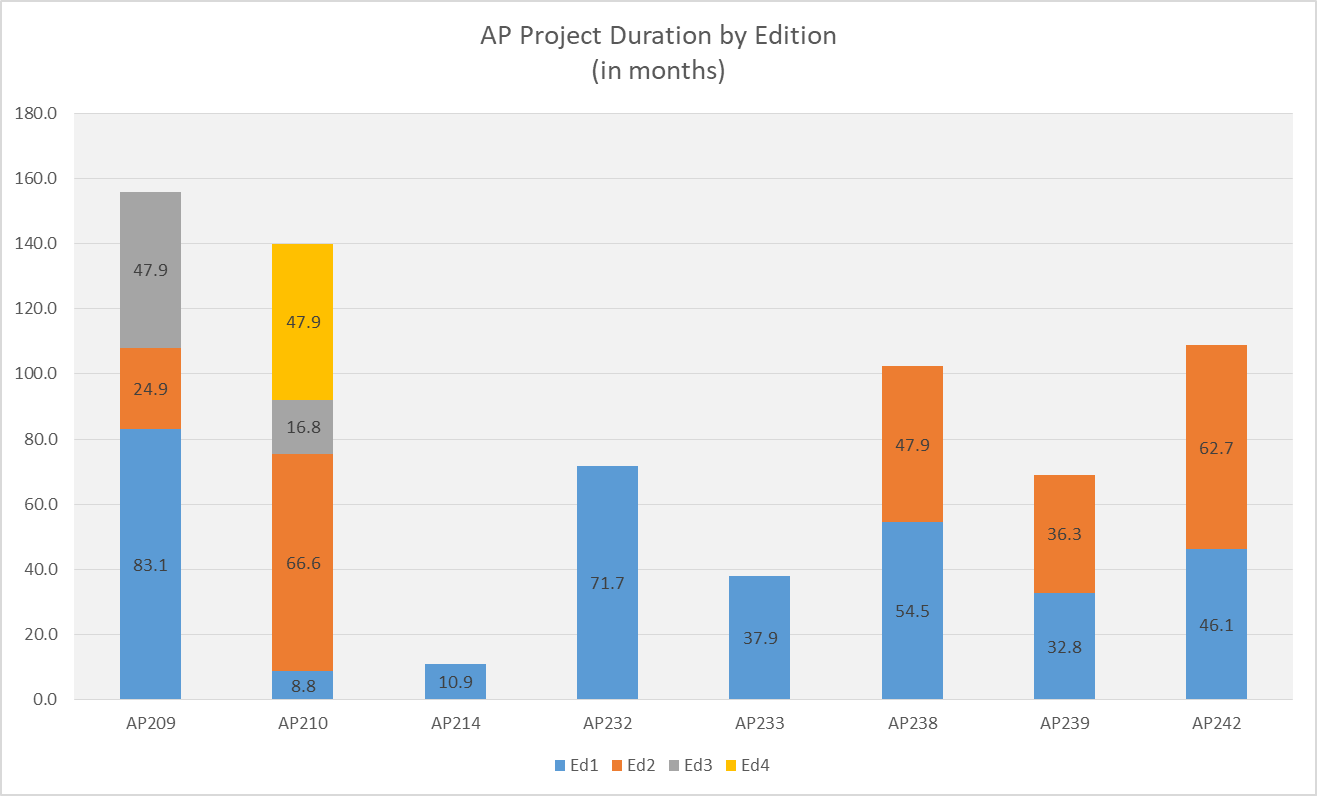
The model-based standard development process consists of multiple stages. Firstly, a proposal for the drafting of a standard is submitted by an individual or an entity, called Sponsor, to a Standards Development Organization (SDO). Then, a collaborative team of experts, called the Working Group (WG), is assembled. This WG works on the development of a committee draft. Once this draft finalized, it is reviewed, changed if necessary and approved first by the WG and then, by a balloting group created by the Sponsor. After that, the final draft is submitted to the SDO Board for final approval. Finally, the standard is published and maintained over the years. In parallel of the standard publication, members of the WG work on developing, testing and implementing tools, methods, and models to support the standard application.

## Information Standards Support of Business Needs

TBD

# Issues in Current Development Lifecycle

"We are implementing features and products and using technology that were not invented 18 months ago. No longer can we afford these large monolithic programs that go on for two to three years (24-36 months)." -Bronwyn Clere, Executive Director for Capital Planning & Delivery, at Telstra Corporation. An analysis of ISO project metrics on a sample across 8 standards and 16 edition publishs revealed the current average project duration is 43.5 months for the release of an edition of a standard. This calculates the time from stage 10.99 of New Project Approval to stage 60.60 International Standard Published. Additional analysis reveals this is an average project length growth of 4.5 months from edition 1 lifecycle lengths to that of edition 2 lifecycles.



**Fig. 1.** This is caption text for Fig. [1].

Average project duration by edition (data based on ISO.org project metrics). Note these metrics are based on a status range from stage 10.99 (new project approval) to 60.60 (international standard published). This is a truncated view and does not include the white paper authoring process which can add as much as an additional year to project duration.

This trend of project length growth for standards is at odds with the industry’s need for faster releases of incremental functionality. ISO’s 2016-2020 strategic directions include six tenants two of which are (1) “Develop high-quality standards through ISO's global membership”, by ensuring we effectively (2)“ Engage stakeholders and partners ”. In order to support industry demand and maximize the benefit of standards the quality and length of development of standards must be addressed.

## Development Time Length

There are two primary contributing factors causing an extended development cycle:

1. Failure to gain consensus [International Agreements]
2. Resource commitment/management.

In the course of STEP development once a new work item is approved or a defect raised, most teams use a system such as Bugzilla for coordination. Bugzilla was originally designed as a general purpose bug tacking and testing tool (this is from wiki, find a better source) however in STEP development it has been stretch beyond that scope. In existing STEP development application it is used as a requirements management, collaboration /consensus tracking, issue management, and task management system in addition to for version control of source models. This leads to much confusion, maintenance, and misuse of Bugzilla. The result is extended development cycles and many times rework during the committee stage 20.

Resource commitment for the performance of development work is also a primary obstacle. Factors that affect commitment of resources include:

1. Volunteer staff
2. Lack of developer resources (which are shared by multiple ISO projects)
3. Virtual distributed team

## Quality/Completeness of Standard

Another issue facing the current development lifecycle is that of quality/completeness concerns of the published standards. CAx-IF, which is a joint testing forum between AFNet, PDES, Inc. and prostep ivip tasked with testing CAx STEP translators quality, has continually reported issues of implement-ability of the standards. These issues fall into three areas and can be categorized as data quality issues, incomplete solutions, and overtly complexities/non implementable solutions. Metrics from Bugzilla show 420 bugs have been reported in these three categories by the CAx-IF against STEP to date.



**Fig. 2.** This is caption text for Fig. [2].

While measures have been taken combat complexity and streamline solutions, such as the transition from monolithic to modular architecture, the problem still persists.

There are two primary root causes contributing to the quality/completeness issues of standards:

1. Lack of knowledge
2. Lack of adequate toolchain

A robust knowledge of the STEP data model is necessary to ensure enhancements and defect resolutions have a complete end to end solution. The integrated nature of the elements of the STEP data model, ARMS, MIMs, and IRs, mandate that parallel changes take place in each of these elements to ensure continuity of the model. A lack of understanding of integration points leads to (1) inelegant solutions and (2) data quality issues. While quality checks exist to catch data quality issues these checks are not infallible. Other issues, such as the elegant or clumsy nature of solution concepts, require analysis by human interpretation and rely solely on the knowledge level of the human resource performing the work. Understanding of the development process and toolset are also factors that affect code quality. Results from a survey of the standards development team reports that 89% of developers incurred some amount of rework during their last publication project due to lack of knowledge of the data model, development process, and/or tool-chain. Of those 7% reported more than 50% of rework was required. For a view of the survey results in their entirety reference appendix B. See the below figure for survey results of perceived knowledge level by tool-chain application.

****

**Fig. 3.** This is caption text for Fig. [3].

The second primary root causes contributing to quality/completeness concerns is an inadequate tool-chain. The current tool-chain for model based standards is fragmented which is the result of it’s incremental development over the last 30 years to adopt fit for use tools to specific tasks. While this has allowed development to go forward it has also resulted in a disjointed tool-chain which relies heavily on process controls and manual integration to achieve end to end project management and development tasks. This leads to a temperamental development process, poor collaboration and version control, and manually introduced errors.

# Solution Concepts [using STEP as an example]

TBD

## Adoption of Agile Framework

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.



**Fig. 4.** This is caption text for Fig. [4].

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.



**Fig. 5.** This is caption text for Fig. [5].

### 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 Implementer Forum.

## Improved Tool-Chain

### Requirements Management and Traceability

TBD Marion

### Solutions

One of the most important principles of agile, and specifically the Scaled Agile’s SAFe framework, is Continuous Delivery. Continuous Delivery can be considered as three independent yet related phases: Continuous Exploration, Continuous Integration and Continuous Deployment.

#### *Continuous Exploration*

Continuous Exploration [CE], as defined by Scaled Agile, is a “process that fosters innovation and builds alignment on what should be built.” (“Continuous Exploration – Scaled Agile Framework,” 2018) CE is when the customers and team members express new ideas, refined and prioritized in the backlog. The final alignment comes during the PI Planning event.

Some ISO 10303 STEP standards [e.g., AP242, AP209, AP210] are developed in a second-generation version control system called CVS. Recently, the development community has migrated to a third-generation version control system called GIT and integrates with KANBANs and advanced communications tools like ChatOps. This can help agile teams rapidly explore new ideas, validating their ability to integrate while not disrupting the production system or branch line. GIT Branching is a key enabler as the previous generation of source code management did not provide collaboration or development areas.

#### *Continuous Integration*

Continuous Integration [CI], as defined by Scaled Agile, is a process of developing and integrating in a continuous flow. This will include tasks such as developing, testing, integrating and validating in an environment before production release. (“Continuous Integration – Scaled Agile Framework,” 2018)

CI is made possible with software development best practices that include version control, automated testing, and build automation. There are many choices in the industry such as Bitbucket/Bamboo, Jenkins, AWS CodePipeline, and Gitlab.

To take advantage of the CI capabilities, the development environment must move to a third-generation version control system based on GIT technology. The new tools will allow for continuous exploration as well as continuous integration via the decentralized and distributed architecture, commit before merge capabilities and integrated quality controls. A CI capability will allow standards developers to receive immediate feedback on the pass/failure of their commits by hooking in tools like EXPRESS Engine, JSDAI Compiles, Python scripts or ANT Builds. Immediate feedback will allow developers to fix the issue in the current iteration and not pass it to the end of the flow for someone else to address [in which case the resource may have moved on and not available]. Another feature of some of the CI tools, and really a requirement, is the connection to other issue/task management systems. Jobs can be triggered by lifecycle promotion of the issue/task and feedback so that everyone on the team can have a clear picture of status of the project deliverables. NOTE: The ISO 10303 Extended Architecture already makes use of GIT capabilities but has not developed a continuous integration pipeline for quality and integration automation.



**Fig. 6.** This is caption text for Fig. [6].

#### *Continuous Deployment*

Again, Scaled Agile provides a framework for Continous Deployment [CD]. In this stage, the product is deployed, verified, monitored and setup for responding to issues. (“Continuous Deployment – Scaled Agile Framework,” 2018) There are many tools for this such as Jenkins, AWS CodeDeploy and Bamboo.

Standards development teams can use these technology to automate the deployment of standards to implementer forums, formal ISO Balloting and Publication processes and systems. If Continuous Integration is fully utilized then the deployment or publication can be performed quickly by automation. Perhaps a better term for this activity in the context of a data standard is Continuous Publication.

# Benefits

TBD IDK

## Benefits to MBS Developer

The Model-Based Standard’s developer is one of the biggest beneficiaries of the solutions identified in this paper. Agile methods and continuous deployment paired with enhanced requirements traceability will bring many benefits to the developer, including:

* Immediate feedback loop to detect and fix issues early
* Increase transparency and visibility to other developers and team members
* Avoid “integration hell”
* Improve quality and testability

The feedback loop is probably the most important aspect to a developer. A NIST study on the Impacts of Inadequate Infrastructure for Software Testing shows that 45% of errors are found in the integration stage of development. A bug introduced in by bad requirements or poor coding can take double the cost to fix in the integration stage and triple the cost in the testing stage. If the bug continues to production, then it can cost up to six times to fix then in the stage it was introduced. (Tassey, Gregory, 2002).

## Benefit to Industry/Enterprise

TBD Sylvere

# Conclusion

TBD IDK

# References

1. 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/
2. Agile Release Train – Scaled Agile Framework. (n.d.). Retrieved July 19, 2019, from /agile-release-train/
3. 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/
4. 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
5. Rico, David F. (2008). What if the ROI of Agile vs Traditional Methods.
6. Scaled Agile. (2018a). Introducing SAFe 4.6.
7. Scaled Agile. (2018b, November). SAFe 4.6 Introductionâ”Overview of the Scaled Agile Framework for Lean Enterprises.
8. Benmoshe, I. (n.d.). How to calculate the ROI of Continuous Delivery. 18.
9. Hilton, M., Tunnell, T., Huang, K., Marinov, D., & Dig, D. (2016). Usage, costs, and benefits of continuous integration in open-source projects. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering - ASE 2016, 426–437. https://doi.org/10.1145/2970276.2970358
10. Tassey, Gregory. (2002, May). The Economic Impacts of Inadequate Infrastructure for Software Testing. NIST.
11. Continuous Deployment – Scaled Agile Framework. (2018). Retrieved July 19, 2019, from https://www.scaledagileframework.com/continuous-deployment/
12. Continuous Exploration – Scaled Agile Framework. (2018). Retrieved July 19, 2019, from https://www.scaledagileframework.com/continuous-exploration/
13. Continuous Integration – Scaled Agile Framework. (2018). Retrieved July 19, 2019, from https://www.scaledagileframework.com/continuous-integration/
14. <https://ieeexplore.ieee.org/abstract/document/6475408>
15. <https://www.iso.org/files/live/sites/isoorg/files/archive/pdf/en/iso_strategy_2016-2020.pdf>
16. <https://www.iso.org/files/live/sites/isoorg/files/developing_standards/docs/en/Target_date_planner_4_ISO_standards_development_tracks_2017.pdf>
17. <https://www.cax-if.org/>
18. <https://www.cio.com/article/3304276/speed-adaptation-and-the-pace-of-change.html>

# Appendix A: Supplemental Materials

Brief description of supplemental files – delete if not applicable.

1. To add materials on standards development time-line analysis
2. To add survey on standards development process
3. To add survey on standards development tool-chain

# Appendix B: Term Bank

1. Model-Based Standard – a standard based on and published as a model that can be reused or implemented directly within other software
2. STEP
3. Continuous Delivery
4. Continuous Exploration
5. Continuous Integration
6. Continuous Deployment
7. ISO

# Appendix C: Change Log

If updating document with errata, detail changes made to document – delete if not applicable.