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

## Improved Tool Chain - **Brandon**

### Continuous Delivery Pipeline - **Brandon**

#### Continuous Exploration

#### Continuous Integration

#### Continuous Deployment

**SOLUTIONS**

One of the most important principles of agile and specifically, the Scaled Agile’s SAFe framework, is Continuous Delivery. Continuous Delivery is divided into three 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 that includes 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 or ANT Builds. Immediate feedback will allow them 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]. 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.



***Continuous Deployment***

Again, Scaled Agile provides a framework for deployment. 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

**REFERENCES:**

Continuous Deployment – Scaled Agile Framework. (2018). Retrieved July 19, 2019, from https://www.scaledagileframework.com/continuous-deployment/

Continuous Exploration – Scaled Agile Framework. (2018). Retrieved July 19, 2019, from https://www.scaledagileframework.com/continuous-exploration/

Continuous Integration – Scaled Agile Framework. (2018). Retrieved July 19, 2019, from https://www.scaledagileframework.com/continuous-integration/

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

**BLUE = Content**

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

**RED = References**