Agile in Systems Engineering

Matt Ingram

May 23, 2019

This report accompanies the presentation of *Agile in Systems Engineering*, an event sponsored by the Midwest Gateway Chapter of INCOSE, described here: **http://incose-mgc.org/agile**.

INCOSE and agile systems

The INCOSE created a working group for Agile Systems and Systems Engineering. Its goal is to create a body of knowledge for systems to operate "….in environments of caprice, uncertainty, risk, variability, and evolution."

The concept of operating in capricious and evolving environments is not new. In an interview to *Business Week*, Apple CEO Steve Jobs said the following:

> Some people say, "Give the customers what they want." But that's not my approach. Our job is to figure out what they're going to want before they do. I think Henry Ford once said, "If I'd asked customers what they wanted, they would have told me, 'A faster horse!' " People don't know what they want until you show it to them. That's why I never rely on market research. Our task is to read things that are not yet on the page.

The idea that "people don't know what they want until you show it to them" can also be expressed as "our requirements are ambiguous" or "our requirements are volatile."

The dilemma of *caprice*

Systems engineering and Agile development attempt to resolve this dilemma in opposite ways. The systems engineering approach is to engineer customer and system requirements until no ambiguity exists. The customer and the development team iron out an agreed set of requirements, trace all subsequent work to those requirements, and verify that those requirements are met.

The Agile development approach is to acknowledge that the customer will not provide final validation of the product until they see it. So the development team defines, builds, tests, and demonstrates a two-week sub-product to the customer. If the customer accepts the sub-product, then the team continues down that path, every two weeks. If the customer rejects the sub-product, then the team gathers feedback and tries again, and it is only out two weeks of work (and has a better idea of what the customer really wanted in the first place.)

Agile and scrum

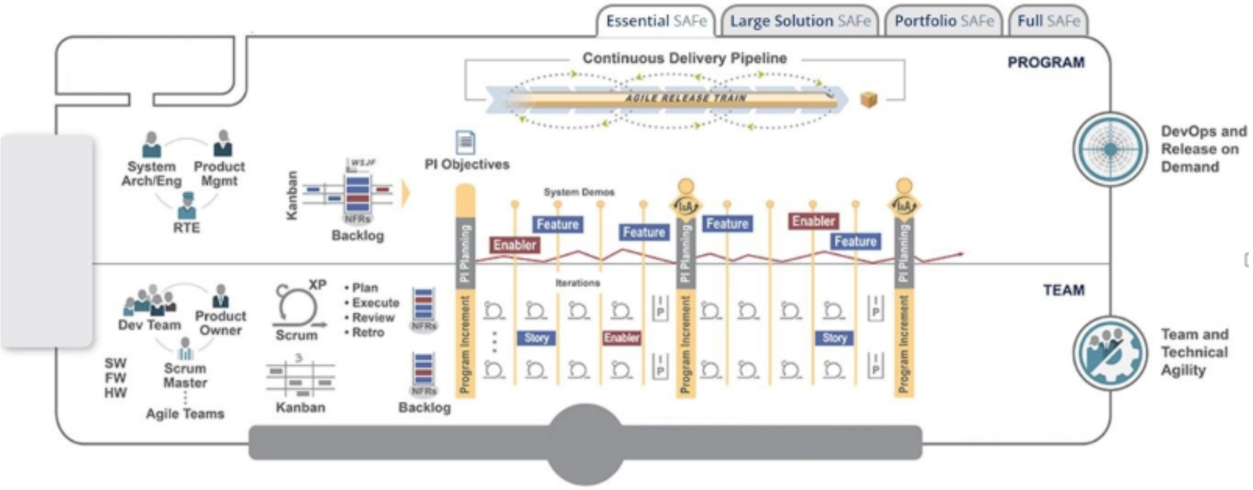
Agile Manifesto co-author Jeff Sutherland provides an entertaining introduction to the origins and philosophy of agile development here: https://www.youtube.com/watch?v=s4thQcgLCqk

Two additional training videos provide an introduction to the Agile practice of Scrum:

- https://www.youtube.com/watch?v=9TycLR0TqFA
- https://www.youtube.com/watch?v=XU0llRltyFM

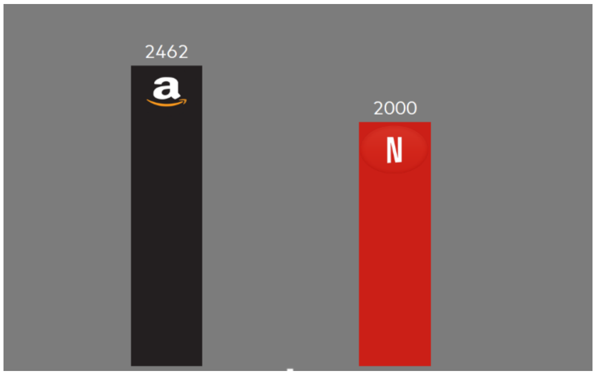
The scaled agile framework

The Scaled Agile Framework (SAFe) is a set of practices and methods to allow an organization to scale the development of its various agile teams into a large and complex system development activity. Many good training and educational resources are at the SAFe website at https://www.scaledagileframework.com/



The article at https://www.playbookhq.co/blog/wsjf-weighted-shortest-job-first explains “weighted shortest job first” and contains an example calculation.

Continuous integration and deployment has been successfully implemented at several companies. See the article at https://techbeacon.com/devops/10-companies-killing-it-devops



The graphic shows the purported number of daily software releases to the production environment for both Amazon and Netflix. The data is based on the *TechBeacon* article statements below:

- “Within a year of Amazon's move to AWS, engineers were deploying code every 11.7 seconds, on average.”
- “…today engineers deploy code thousands of times per day.”