Challenges of Applying Conventional Software System Safety to Agile Software Development Programs

Melissa A. Emery; A-P-T Research, Inc.;
Huntsville, Alabama, USA

David B. West, CSP, P.E., CHMM; Science Applications International Corporation; Huntsville, Alabama, USA

A-P-T Research, Inc.

SAIC

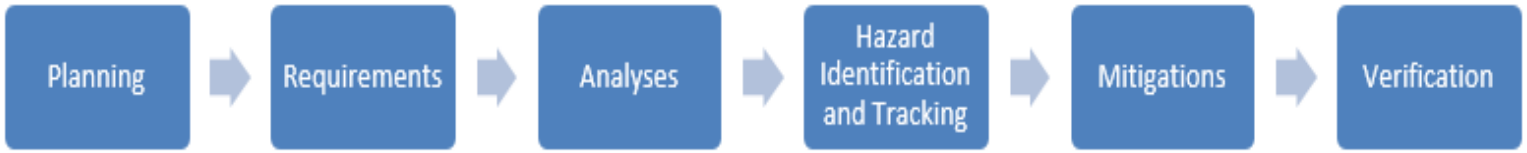
Order of Presentation

- Introduction
- Conventional Processes for SW Development and SW Safety
- Overview of Agile SW Development Process
- Issues with Agile and Conventional Processes
- Suggested Tailoring of System/SW Safety Process for Agile Development

Conventional Processes

Waterfall Flow

Typical Safety Process



Overview of Agile SW Development Process

Agile Development – A Brief History

- Rooted in movements to reduce waste in manufacturing and production
  - Toyota Production System (1948-1975)
  - Lean Manufacturing
- Manifesto for Agile Software Development (2001)
- Four Key Values from the Agile Manifesto:
  1. Individuals and interactions over process and tools.
  2. Working software over comprehensive documentation.
  3. Customer collaboration over contract negotiation.
  4. Responding to change over following a plan.

Overview of Agile Development (Cont.)

© Scott Adams, Inc./Dist. by UFS, Inc.

Overview of Agile Development (Cont.)

Guiding Principles of Agile Development

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

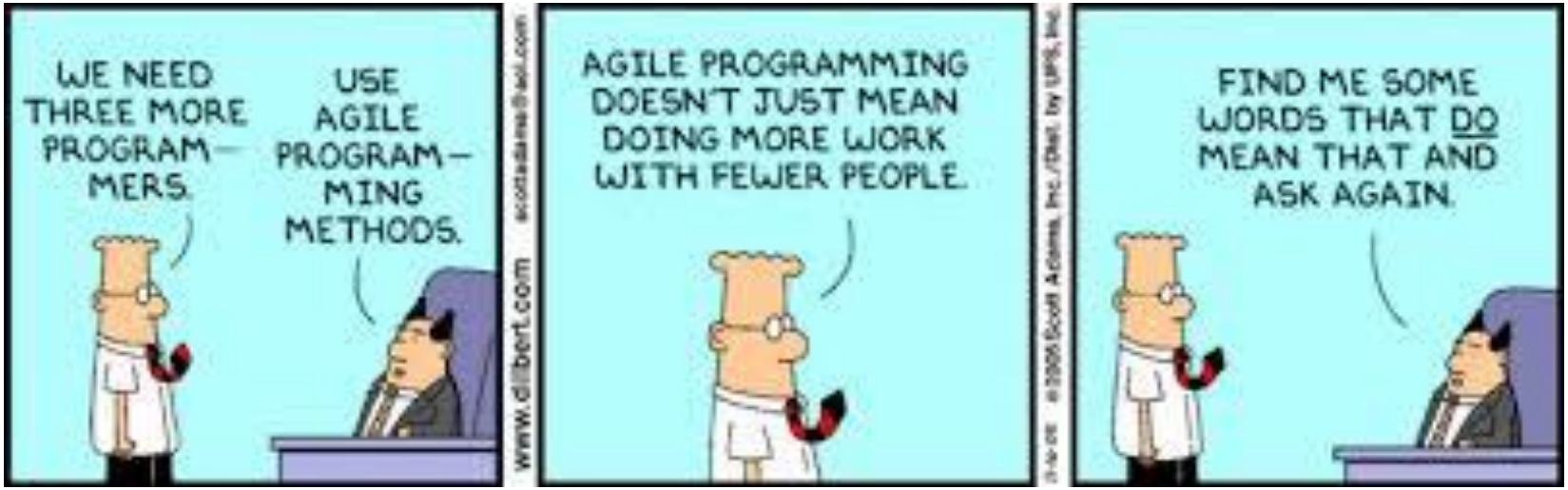
Overview of Agile Development (Cont.)

Guiding Principles of Agile Development (Cont.)

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

[Source: The Manifesto for Agile Software Development (2001)]

Overview of Agile Development (Cont.)



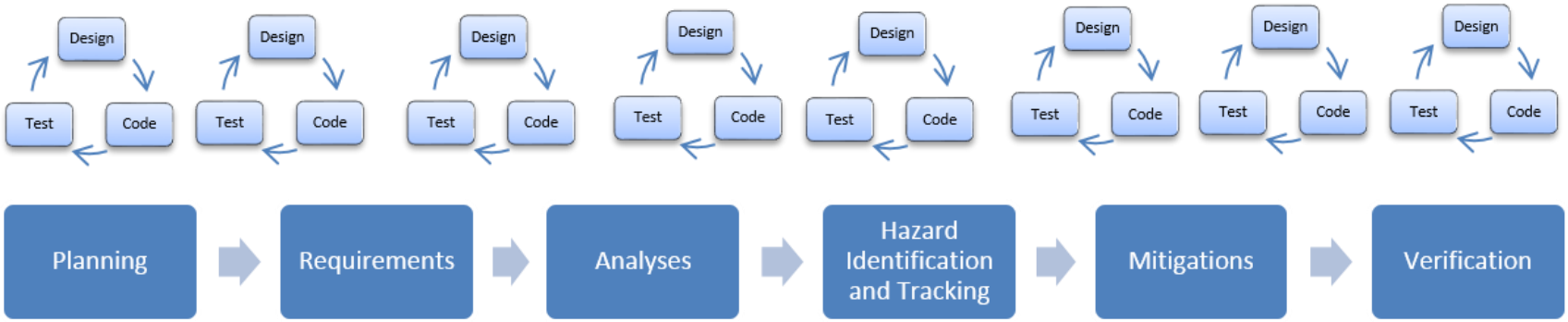
Overview of Agile Development (Cont.)

- Requirements expressed as "User Stories"
- Development done in discrete "sprints" of constant length
- Planning and conducting each sprint consists of:
  1. Selection of the customer's highest priority user stories
  2. Estimation of user story points for selected user stories
  3. Identification of tasks for each user story, and estimation of effort by developers
  4. Elaboration of user stories, design, code, and test
  5. Demonstration of a working system
  6. A retrospective and improvement of the process
- Progress tracked by no. of user stories completed
- Slip user stories, if necessary, but not release dates

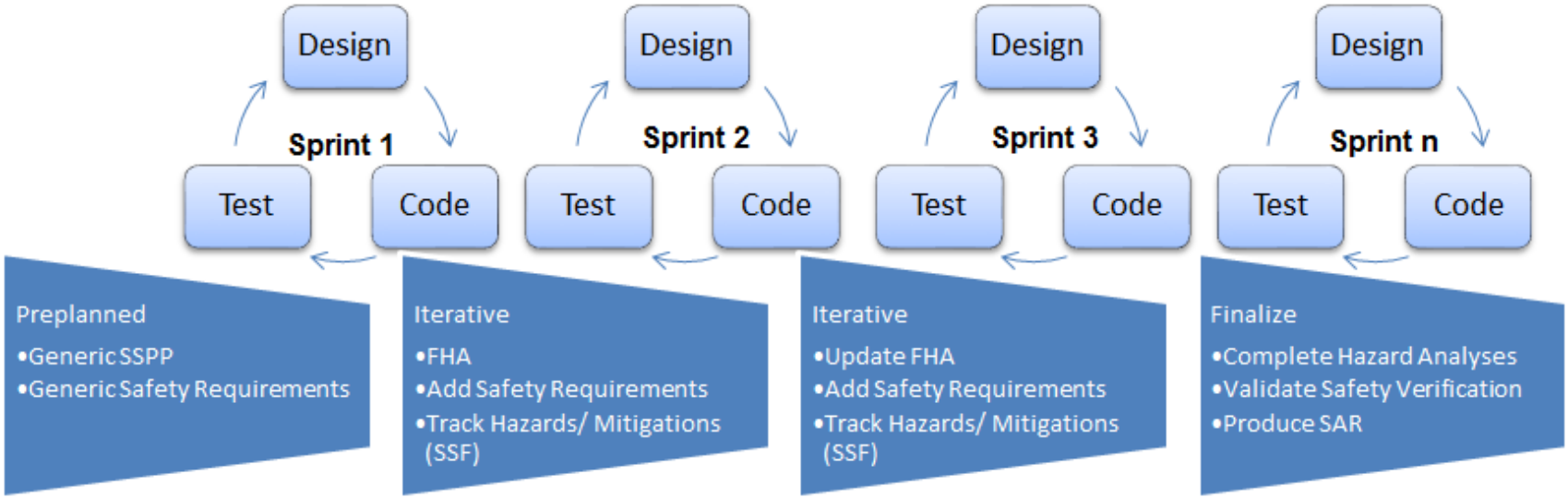
Issues with Agile and Conventional Processes

- Different Mindsets:
  - Agile SW Development – acknowledges need for flexibility; expects mid-course corrections
  - Conventional SW Safety – expects each successive safety document to be essentially finalized before moving on to the next one
- Different Time-Phasing:
  - Several weeks to more than a month for initial draft, review-and-comment by customer, finalizing, and obtaining signatures on SSPP
  - Development team could complete multiple sprints during SSPP preparation
- Safety analyses require multi-discipline team; developers usually not available (committed full-time to sprint planning and execution)

Issues with Agile and Conventional Processes (Cont.)



Software/System Safety Process Tailoring



Tailored Process

- Use a Standard System Safety Program Plan (or none—just refer to MIL STD 882). Don't spend unnecessary time creating this document. Review MIL-STD-882E and add any additional specifications or regulations to the Statement of Work (SOW).
- Impose Generic Safety Requirements at the Onset. This sets the tone for safety's involvement. It helps developers keep safety in the decision process. Allocate generic safety requirements for Design, Code, and Test activities (see JSSSEH for suggestions). For example, for Design, impose safe state, memory, integrity, and fault detection type requirements. For Code activities requirements about dead/unused code, storage, and other coding standards should be imposed. For Test activities, document the safety testing and level of rigor (LOR) testing required so these can be accomplished in the unit level testing and planned for in the system testing.
- Initiate a Functional Hazard Analysis (FHA) at the Onset. This provides management and guidance of safety questions with respect to degraded operation, loss of operation, etc. This allows the analysis to grow with the project instead of waiting for other typical analyses (e.g. PHA) and system documentation to be available. If time permits, conduct a PHA later in the cycle. The FHA will aid in generation of safety significant functions and requirements which should be modified, if required, throughout the process.
- Conduct Several Requirements Analyses: For Agile Development, it is more productive to perform several requirements analyses in contrast to using the final program documentation. Augmenting with change impact analysis ensures the requirements are constantly evaluated and documented in the program and safety artifacts.
- Attend Sprint Meetings to Understand Content, Changes, Future Expectations, etc. After each Sprint meeting, reevaluate the FHA and generic safety requirements. Implement the necessary changes and review with System Safety Working Group (SSWG) members on a regular basis.