

Automotive – SPIN Italia
Third Workshop on Automotive Software

Pisa (Italy), May, 15 2008



SAFETY & RESILIENCE ISSUES
IN AUTOMOTIVE
SOFTWARE DEVELOPMENT

PANEL

Automotive – SPIN Italia
Third Workshop on Automotive Software
Pisa (Italy), May, 15 2008



Safety Panel

- ✓ when 26262 will be issued, enforced ? What about 61508 ?
- ✓ by whom, which authorities ?
- ✓ who and how will verify conformance ? (statement of conformance)
- ✓ how costs and schedule are impacted (ASIL A vs. ASIL D) (es. 1X10)
- ✓ how to handle pre-developed software, validated tools
- ✓ software partitioning of mixed critical software
- ✓ ISO 9001, ISO/TS, ISO 26262, AutoSPICE, ISO/TR 15497:2000
- ✓ And security ?
- ✓ Relationship with AUTOSAR (Safety WP)

Automotive – SPIN Italia
Third Workshop on Automotive Software

Pisa (Italy), May, 15 2008



Open Discussion

Experience and drawbacks in
standards application

“Agility” for Small Projects

Background

- 1968 : the software crisis is declared in Garmisch: we cannot continue to develop software this way !
- Software Engineering is born...
- Waterfall, V model, iterative, evolutive, spiral, etc.
- DOD 2167A, DO 178B, PSS-05, CENELEC, etc.
- ISO 90003, TickIT, ISO 12207, SPICE, CMM, CMMI

- Late 1990s: a revolt against “heavy processes” with the Agile Challenge: Extreme Programming, Scrum, Crystal, Lean, ...

Manifesto for Agile Software Development - 2001

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck

James Grenning

Robert C. Martin

Mike Beedle

Jim Highsmith

Steve Mellor

Arie van Bennekum

Andrew Hunt

Ken Schwaber

Alistair Cockburn

Ron Jeffries

Jeff Sutherland

Ward Cunningham

Jon Kern

Dave Thomas

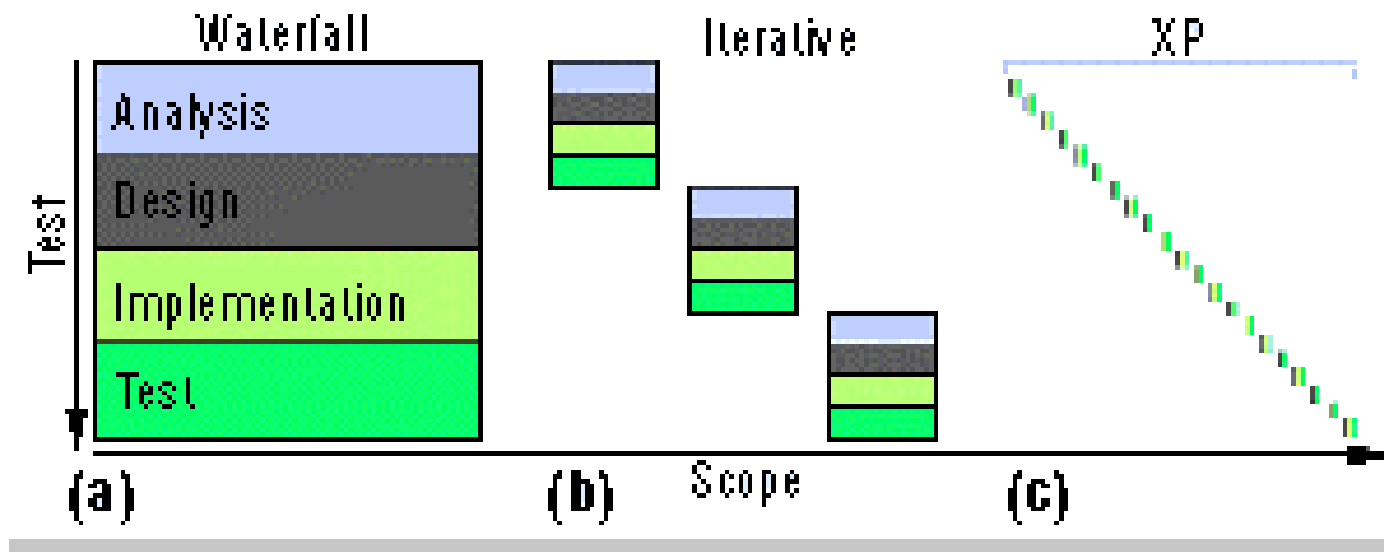
Martin Fowler

Brian Marick

Some principles of XP

- **Pair programming.** All production code is written by two people at one screen/keyboard/mouse.
- **Continuous integration.** New code is integrated with the current system after no more than a few hours. When integrating, the system is built from scratch and all tests must pass or the changes are discarded.
- **On-site customer.** A customer sits with the team full-time.
- **Collective ownership.** Nobody “owns his own code.” Anybody can change anybody else’s code.
- **Planning game.** After each iteration, the requirements and scope of the project may be changed.

From waterfall to daily/weekly delivery



The new buzzwords

- Planning game
- Product burndown
- Sprints
- Scrum master
- Team velocity
- Standup meeting
- Project wall-board
- Architecture slices
- Celebration
- Toolsmith
- Story points
- Wireframe
- Backlog
- Product roadmap
- ...

Lessons Learned From Agile

- Efficiency involves minimizing:
 - Minimize *unnecessary* scope / requirements
 - Minimize *unnecessary* documentation
 - Minimize *unnecessary* coding
- Iterative development appears to be central
- Although these are well-known best practices in *theory*, the agile movement is helping discover what is necessary to make them operational in *practice*

SERVE UN PO DI
AGILITY
NEI NOSTRI PROCESSI ?

Agile versus CMMI-like processes

- The big question: are they incompatible?
- Many are working hard to make them compatible (www.agilecmmi.com)
- The SEI and others (e.g. Boehm) have a strong interest in reconciling them
- Can a “CMMI team” be agile? The SEI Team Software Process tries to be

Agile Manifesto

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

How the SEI Team Software Process (TSP) Relates

TSP holds that the individual is key to product quality and effective member interactions are necessary to the team's success.

Project launches strive to create gelled teams. Weekly meetings and communication are essential to sustain them. Teams define their own processes in the launch.

TSP teams can choose evolutionary or iterative lifecycle models to deliver early functionality—the focus is on high quality from the start. TSP does not require heavy documentation.

Documentation should merely be sufficient to facilitate effective reviews and information sharing.

Learning what the customer wants is a key focus of the launch. Sustaining customer contact is one reason for having a customer interface manager on the team.

Focus on negotiation of a contract is more a factor of the organization than of whether TSP is used.

TSP teams expect and plan for change by:

Adjusting the team's process through process improvement proposals and weekly meetings. Periodically relaunching and replanning whenever the plan is no longer a useful guide. Adding new tasks as they are discovered; removing tasks that are no longer needed. Dynamically rebalancing the team workload as required to finish faster. Actively identifying and managing risks.

Agile Versus Certification and Standards

- Where could the issues lie?
- Emergent architecture – architecture “emerges” from the continuous design process
- Continuous changing (refactoring) of code
- Scope of projects (such as requirements) can expand and contract over the life of an agile project
- Where to standards fit into this paradigm?