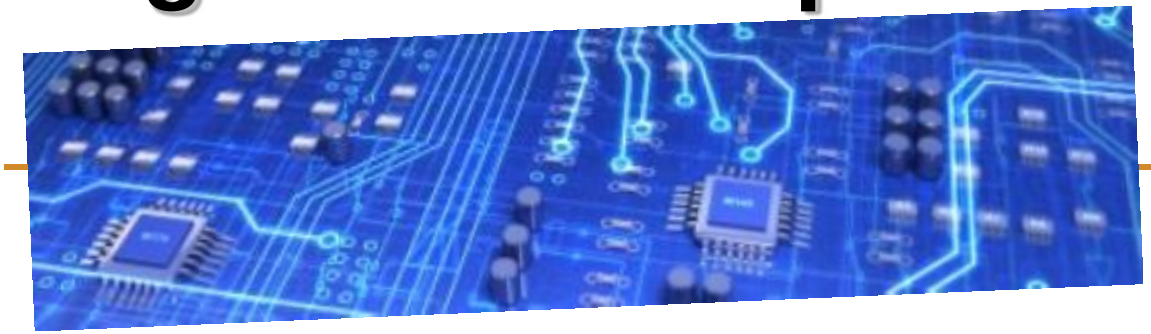
Agile for Aerospace



Andrew Buter, Curt Stienstra, Steven H. VanderLeest

GLSEC 2008

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?

II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace

III. Q&A

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?
II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace
III. Q&A

DO-178 Waterfall Approach

- Safety-critical software:
  - Independence
  - Traceability
  - Verification/validation
- Waterfall organization of the DO-178 standard
- Problems with Waterfall

Waterfall Defined

1. Define up front, in detail, the requirements
2. Define the “design” (text and diagrammatic descriptions of the software and hardware elements)
3. Implement the system (programming, etc.)
4. Integrate and test the components

From Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

Problems with Waterfall

- Works for projects with little change, little novelty, and low complexity
- Pushes high-risk and difficult elements to end of the project
- Aggravates complexity overload
- Inability to deal with changing requirements
- Late integration
- Unreliable up-front schedules and estimates

From Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

Problems with Waterfall

- MacCormack 2001 study of projects
  - Evolutionary appproach yielded higher quality results faster
  - Waterfall approaches uniformly performed poorly
- MacCormack 2003 study
  - Weak relationship between detailed upfront design specifications and low defect rate
  - Earlier release with less functionality (compared to waiting for more functionality) resulted in a lower defect rate and higher productivity
  - Daily builds with integration and regression testing produced lower defect rate and higher productivity
- Harrison 1996 study at Bell Labs identified iterative development as one of the consistent patterns in highly successful projects

From Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

Problems with Waterfall

- Thomas 2001 study of over 1,000 IT projects in UK found single largest factor in failed projects was scope management related to attempting waterfall practices
- Brooks 1987 committee concluded about DOD-STD-2167: "In the decade since the waterfall model was developed, our discipline as come to recognize that setting the requirements is the most difficult and crucial part of the software building process, and one that requires iteration between the designers and users. In best modern practice, the early specification is embodied in a prototype, which the intended users can themselves drive in order to see the consequences of their imaginings. Then, as the design effort begins to yield data on the cost and schedule consequences of particular specifications, the designers and the users revise the specifications."
- DOD replaced DOD-STD-2167 (waterfall) with MIL-STD-498 (iterative)
- Jones 1995 study of 6,700 projects found 4 out of 5 key factors contributing to failures were associated with waterfall models

From Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

Benefits of Agile

- Development costs reduced by up to 70%
- Quality more than 3 times better than industry average
- Customer satisfaction 4.9 on a 5 point scale
- 70% developer satisfaction with process

From ITEA, *Project AGILE*, http://www.agile-itea.org, 2006.

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?
II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace
III. Q&A

Who is doing Aerospace Agile?

- Master's thesis by Ron Chisholm at Royal Military College of Canada: "AGILE SOFTWARE DEVELOPMENT METHODS AND DO-178 CERTIFICATION“
- Pilot program at Barco (Belgian avionics equipment supplier), reported in Wils, Van Baelen, Holvoet, and DeVlaminck, "Agility in the Avionics Software World“
- Mayford Technologies reports that Space shuttle primary avionics software developed using Agile-like processes [http://www.mayford.ca/download/TransitioningToAgile.pdf]
- Foliage: http://www.foliage.com/what-we-do/software-development.php uses Agile for development of software under FAA DO-178 and also FDA requirements

Who is doing Aerospace Agile?

- "Use of Agile Techniques in the Development of a Safety-Critical Rail Application", Dave Nicoll, Project & Programme Manager, Silver Software, *Agile Business Conference 2008*, September 23-24, London.
- *Lean Software Strategies: Proven Techniques for Managers and Developers* by Peter Middleton and Jim Sutton [small mention of Agile techniques for Avionics]
- "Incorporation of Spiral and Agile Development into an Existing CMM Level 3 Process," Peter Hoffman and Carl Scott, *Proceedings of the 49th International Instrumentation Symposium*, May 4-8 2003, Orlando, FL, p 105-117.

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?
II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace
III. Q&A

DO-178

- Design Assurance vs. Process Verification
- Applying legacy processes to an agile environment
  - Planning and Certification processes remain unmodified
  - Planning documents see the most change

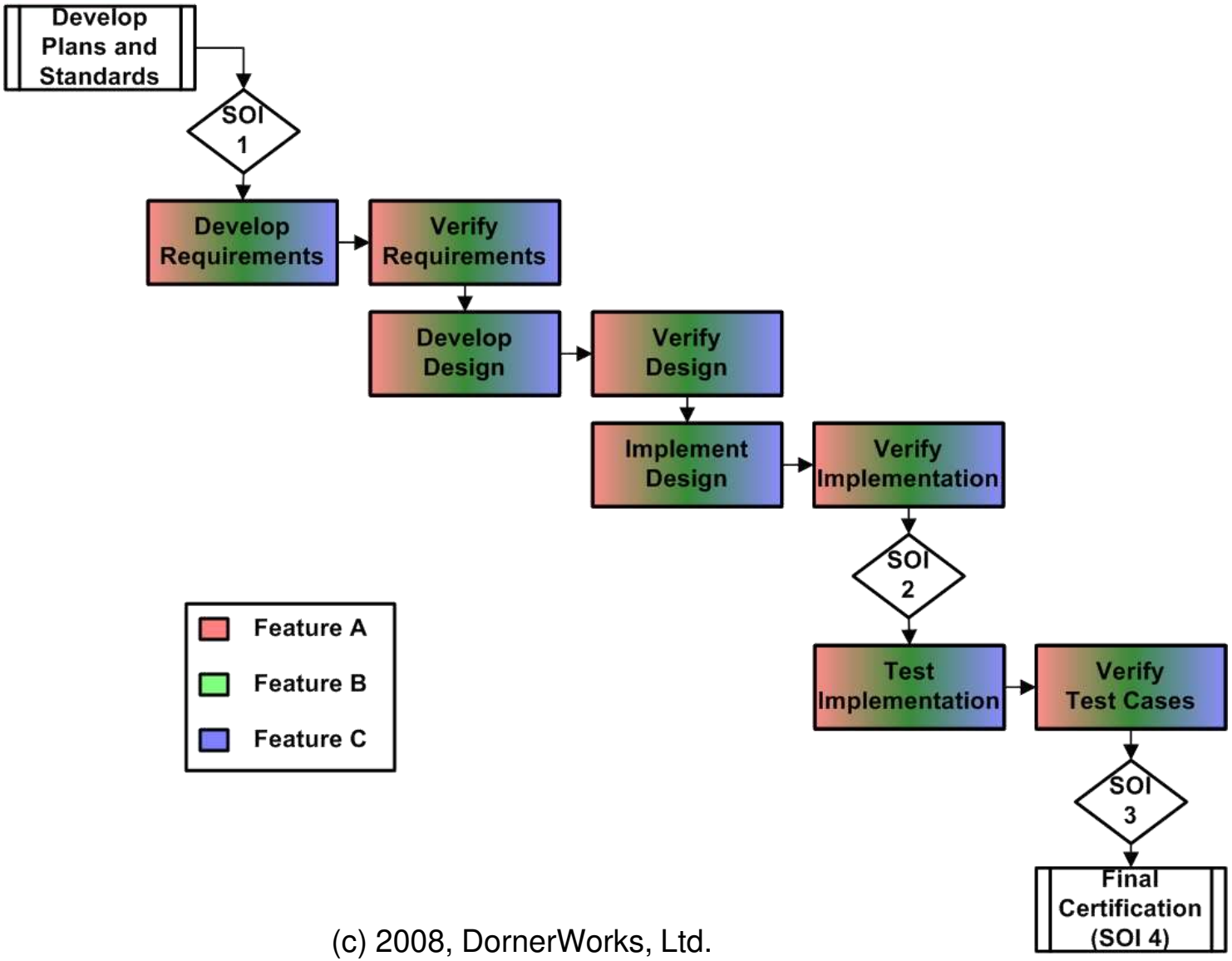
Process Verification

- Requires fully exercise of software through input stimulation
- Exhaustive testing ensures that all possible input/output combinations are identified during testing
- Since all I/O combinations are known, all errors can be identified and corrected prior to certification

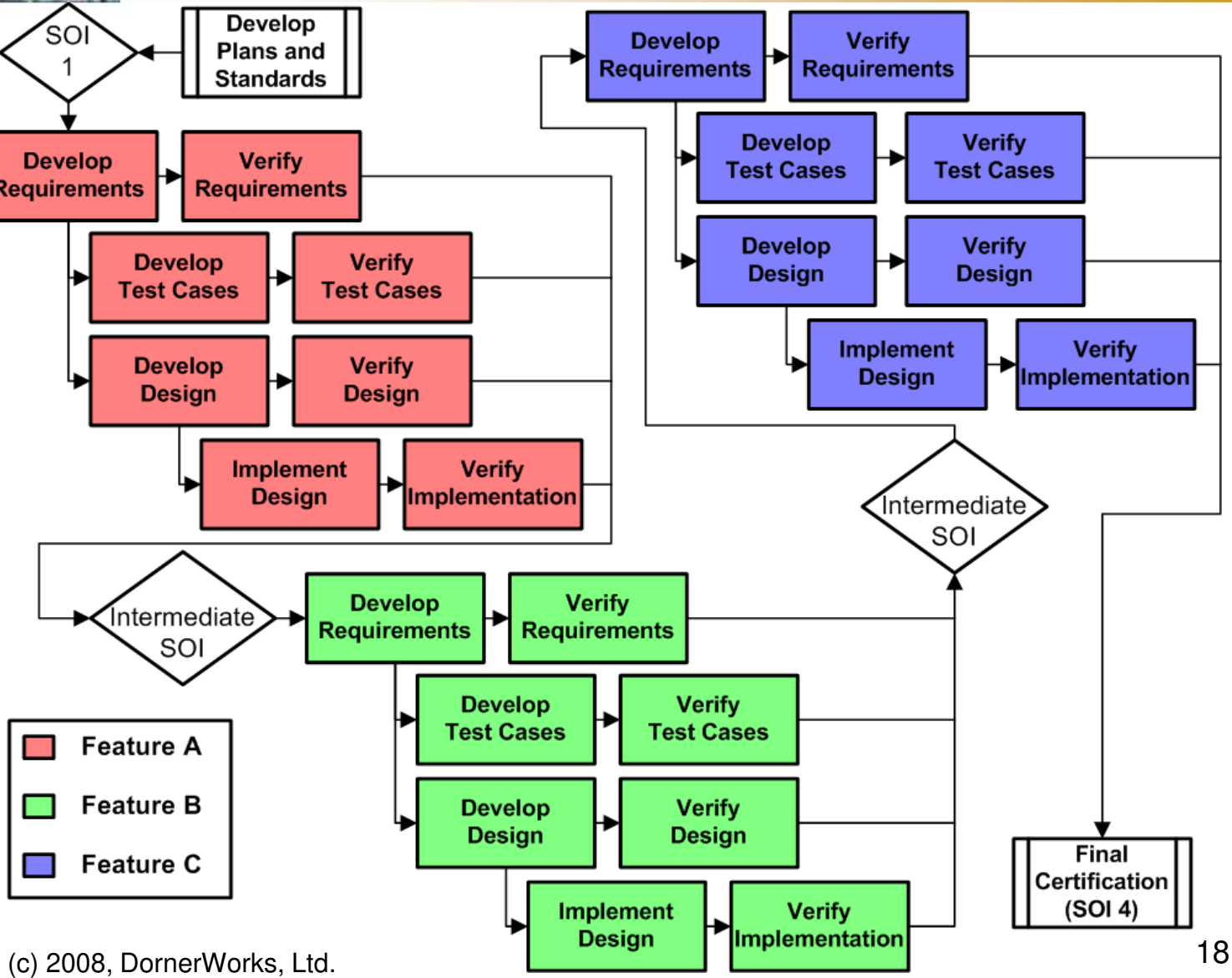
Design Assurance

- Acknowledges difficulty of fully exercising software through inputs
- Addresses risks involved with non-exhaustive testing by analyzing software for identified categories of errors
- Aims to simulate long time-in-field through rigorous testing and analysis
- Admits that not all errors will be addressed prior to certification

DO-178 Waterfall Process



DO-178 Agile Process



Agile Impacts

- Insuring independence
  - DO-178 requires independence where Agile encourages co-operation
- Managing traceability
  - The management of the traceability is not inherent to Agile
- Achieving verification/validation
  - Software testing is Agile, document verification is Waterfall

Addressing the Impacts

- Insuring independence
  - Agile practices do not preclude a process in which a separate developer writes the tests for an implementation
- Managing traceability
  - Traceability is established early (end-to-end trace after each iteration, exhaustive by last iteration)
- Achieving verification/validation
  - Documents (i.e. Requirements and Design) can be generated and reviewed during each iteration
  - A final coherency review would then be performed prior to certification

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?
II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace
III. Q&A

Agile

- Agile Methodologies
  - Crystal
  - Unified Process
  - SCRUM
  - Extreme Programming (XP)
- Agile Techniques
  - Test Driven Development
  - Refactoring

Agile Key Practices

- Continuous delivery
- Welcome changing requirements
- Customer/developers work together daily
- Build around motivated individuals with necessary environment and support, trust to get job done
- Face-to-face conversation most effective communication method
- Progress measured by working software
- Sustainable pace
- Technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular team reflection/adjustments to become more effective
- Risk-driven iterative development

Summarized from http://agilemanifesto.org/principles.html and Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

SCRUM Key Practices

- Self-directed and self-organizing teams
- Fixed length iterations
  - Client-driven adaptive planning at start
  - No external addition of work to an iteration, once chosen
  - Daily stand-up meeting with special questions
    - What have you done since last meeting?
    - What will you do before next meeting?
    - What is blocking progress?
  - Demo to external stakeholders at end

From Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

XP Key Practices

- Planning game
- Small, frequent releases
- System metaphors
- Simple design
- Testing
- Frequent refactoring
- Pair programming
- Team code ownership
- Continuous integration
- Sustainable pace
- Whole team together
- Coding Standards

From Larman, *Agile & Iterative Development*, Boston: Addison-Wesley, 2004.

Test Driven Development

- System tests written first
- Unit tests written before code
- Automated
- Organic, symbiotic

Refactoring

- Improvement of existing effort (code, tests, documentation) that was likely considered done
  - Developer's prerogative: Small changes achieved within iteration/sprint
  - Larger changes prioritized in iteration/sprint planning

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?
II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace
III. Q&A

Ranking

- No unique challenges
- Minor challenges
- Difficult
- Impossible

Agile Key Practices

- Continuous delivery
- Welcome changing requirements
- Customer/developers work together daily
- Build around motivated individuals with necessary environment and support, trust to get job done
- Face-to-face conversation most effective communication method
- Progress measured by working software
- Sustainable pace
- Technical excellence and good design
- Simplicity
- Self-organizing teams
- Regular team reflection/adjustments to become more effective
- Risk-driven iterative development

SCRUM Key Practices

- Self-directed and self-organizing teams
- Fixed length iterations
  - Client-driven adaptive planning at start
  - No external addition of work to an iteration, once chosen
  - Daily stand-up meeting with special questions
    - What have you done since last meeting?
    - What will you do before next meeting?
    - What is blocking progress?
  - Demo to external stakeholders at end

XP Key Practices

- Planning game
- Small, frequent releases
- System metaphors
- Simple design
- Testing
- Frequent refactoring
- Pair programming
- Team code ownership
- Continuous integration
- Sustainable pace
- Whole team together
- Coding Standards

Test Driven Development

- System tests written first
- Unit tests written before code
- Automated
- Organic, symbiotic

Refactoring

- Improvement of existing effort (code, tests, documentation) that was likely considered done
  - Developer's prerogative: Small changes achieved within iteration/sprint
  - Larger changes prioritized in iteration/sprint planning

Conclusions

- DO-178 Waterfall leads to problems
- Agile methodologies address many of the failings of Waterfall
- Agile methodologies can be applied to Aerospace software development, three challenges are:
  - SOI Reviews
  - Formal Documentation
  - Contracts: Firm Fixed vs. Time & Materials

Outline

I. Present State of SW Dev in Aerospace
   A. DO-178 and the waterfall approach
   B. Who is doing Agile currently?

II. Introducing Agile to Aerospace
   A. DO-178 on Agile practices
   B. Customizing Agile for Aerospace
      1. What is Agile?
      2. Impacts on Software Development for Aerospace

III. Q&A

Acknowledgements

The authors gratefully acknowledge the help and advice of the following individuals:

- Dewi Daniels, Deputy Head of Consultancy, *Silver Software*
- Nick Papageorgis, Director of Advanced Technology, *GE Aviation Systems*
- Leanna K. Rierson, *Digital Safety Consulting*, FAA DER
- Timothy Skutt, Software Architect, *GE Aviation Systems*