A Model-Based Agile Process for DO-178C Certification

David J. Coe and Jeffrey H. Kulick
Department of Electrical and Computer Engineering
University of Alabama in Huntsville, Huntsville, Alabama, USA

Abstract - Increasing complexity has driven aerospace companies to consider the use of Agile processes for development of safety-critical systems. For other domains, Agile processes have been shown to improve cost, schedule, and quality metrics. Airworthiness certification under the Federal Aviation Administration (FAA) guidelines imposes unique challenges that require adaptation of Agile processes. The FAA’s mission is maintaining safety within the National Air Space, and the certification process that the FAA has adopted is a process-oriented standard RTCA DO-178C. Here we present a Model-Based Agile Process (MBA process) that will allow companies to benefit from some of the efficiencies inherent in Agile methods while maintaining compliance with airworthiness certification requirements. Model-based requirements capture using the Unified Modeling Language (UML) facilitates iterative and incremental capture, refinement, and verification of requirements using executable requirements models, maintaining the Agility of the requirements elicitation process.

Keywords: RTCA DO-178C, RTCA DO-331, Model-Based Agile Process, MBA Process, safety critical systems

1 Motivation

Modern aircraft have become increasingly dependent upon computers for control of critical functions including engines, brakes, flight controls, navigation, and communications. The F-35 Joint Strike Fighter has approximately 9.5 million lines of code on board and a total of 24 millions lines of code for this system [1]. Late software releases for the F-35 have resulted in delays in testing, training, and delivery, and they have contributed to cost overruns. While delays and overruns have been common in the development of complex military systems, increased hardware and software complexity is also appearing in civilian aircraft systems.

The Boeing 787 Dreamliner is an example of a civilian aircraft that is projected to have over 6 million lines of code with major subsystems such as engines, flaps, and landing gear all incorporating network connections that will allow engineers to log half a terabyte of data per flight [2-3]. The added capabilities and complexity have resulted in significant cost with software development and integration issues resulting in delivery delays and reports of over 200,000 hours expended during the Federal Aviation Administration (FAA) airworthiness certification effort not counting the recent battery fire issues [4-5].

To cope with cost, schedule, and quality issues, the aerospace industry has started to explore the use of Agile methodologies for the development of safety-critical aerospace software systems. Agile processes offer a number of advantages in comparison to traditional Waterfall-based methods including higher quality, lower cost, higher productivity, and improved schedule performance [6]. An open question has been the compatibility of Agile methods with the FAA airworthiness certification process. Below we discuss the airworthiness certification process, principles of Agile development and the potential conflicts with the certification process, and our modified Agile process that addresses the areas of concern.

2 Airworthiness Certification

The FAA currently utilizes the RTCA DO-178C standard for certification of airborne software [7]. Rather than mandating a particular process, DO-178C requires that any development process used for airborne software satisfy a list of process-oriented objectives, with the specific subset of required process objectives dictated by the criticality of the software to safe operation of the aircraft. Some examples of DO-178C process objectives include end-to-end traceability, change control and configuration management, and requirements-based testing. A safety analysis process is used to determine the criticality of the software’s function in the context of the overall system. It is important to note that safety is an emergent property of the system as whole, and that
the safety of a component cannot be established outside of the context of its use [8]. Thus, system-level requirements are an input to the safety analysis process. While no particular safety process is mandated, SAE ARP4761 is an example of a commonly used safety process that includes Functional Hazard Analysis (FHA), Fault Tree Analysis (FTA), Failure Modes and Effects Analysis (FMEA), and Common Cause Analysis (CCA) [9].

Two key results emerge from the safety analysis process. First, the safety analysis establishes the Design Assurance Level (DAL), which dictates the specific DO-178C process objectives that must be satisfied by the software development process and the level of rigor that must be demonstrated for each objective. The safety analysis also identifies safety requirements that the software product itself must demonstrably satisfy as part of the verification process. As summarized in Table 1, the number of required process objectives that must be demonstrated dramatically increases from DAL E up to DAL A as does the number of the objectives requiring independence. In Table 1, xx/yy indicates that xx objectives must be satisfied with yy of them satisfied with independence.

Designated Engineering Representatives (DERs) are the FAA’s embedded representatives within the development teams. DERs are in a unique position in that they represent the FAA while being paid by the company developing the product. The role of the DER is to use their engineering background and aviation safety certification experience to provide feedback to the team regarding the team’s compliance with DO-178C. The DER also interacts directly with the FAA to facilitate the certification process.

### 3 Agile Development Principles and Airworthiness Certification

As shown in Table 2, the term Agile development encompasses a family of processes that embrace a common set of core principles enumerated by the Agile Manifesto [11]. Examples of commonly used Agile processes include Scrum and Extreme Programming. Below we briefly examine potential sources of conflict between the principles of Agile development and DO-178C certification requirement.
Table 2 – Guiding principles of Agile development versus DO-178C principles [11].

<table>
<thead>
<tr>
<th>Agile principle</th>
<th>DO-178C principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individuals and Interactions</td>
<td>Processes and Tools</td>
</tr>
<tr>
<td>Working Software</td>
<td>Comprehensive Documentation</td>
</tr>
<tr>
<td>Evolving Requirements via Customer Collaboration</td>
<td>Rigorous Requirements Specification</td>
</tr>
<tr>
<td>Responding to Change</td>
<td>Following a Plan</td>
</tr>
</tbody>
</table>

3.1 Individuals and Interactions versus Processes and Tools

Agile processes emphasize face-to-face communication as the best way to convey information. No rigorous process is outlined for the method of communication nor are there any requirements to capture the result of the communication. Components may be added or deleted at any time without an impact analysis as to what the addition or deletion will have to already existing software.

DO-178C, however, requires clear commitments to processes and tools. The very first step in a DO-178C project is the PSAC – Plan for Software Aspects of Certification. This detailed plan outlines the roles of other plans and processes in the certification process. For each tool used the level of quality of the tool must be examined, particularly if the output will not be subsequently inspected such as by peer review [12]. For example, a tool that verifies the output is correct and agrees with regression runs, must itself be developed to the same level of rigor as the system itself. In other words, a DAL A product requires a DAL A tool if the outputs of the tool are trusted without verification.

3.2 Working Software versus Comprehensive Documentation

Proponents of Agile processes believe that working software is the best documentation of product requirements and design, and that it is also the best measure of progress. Alternatively, in Waterfall development, artifacts such as Software Requirements Specification and Software Design Documents must be maintained throughout the development lifecycle otherwise the information contained within such artifacts may begin to diverge significantly from the actual product, thereby reducing the end value of the documentation artifacts to future maintenance and enhancement efforts. For certification under DO-178C, however, the development team must demonstrate that the required subset of process objectives has been satisfied.

Failure to maintain adequate documentation of compliance with process objectives can lead to costly delays, even when the software is working error free. For example, the FAA requires documentation that shows end-to-end traceability for DAL A through DAL D software. End-to-end traceability means that a requirement must be forwards/backwards traceable through the design, the source code, the object code, and the associated requirements-derived tests. Delivery of the Airbus A400M military transport was delayed due to a subcontractor’s inability to demonstrate artifact traceability for the aircraft’s full authority digital engine controller to the European Aviation Safety Agency (EASA) [13]. While the EASA aircraft certification process is different than the FAA process, both agencies require traceability.

3.3 Evolving Requirements via Customer Collaboration versus Rigorous Requirements Specification

Agile methods emphasize customer collaboration as the best means of eliciting product requirements. Requirements documents do a poor job of capturing product requirements in no small part because the majority of customer requirements knowledge is internal, never documented information that may emerge in the form of new requirements when a customer gets to see and interact with an implementation of the product [14]. In addition, requirements specifications for complex systems may entail thousands of potentially conflicting requirements that are elicited over extended periods of
time, in some cases years. In an ideal Agile development scenario, the customer works closely with the development team in an iterative fashion to identify, prioritize, and refine product requirements and to develop a set of acceptance tests that will be used to verify successful implementation of those requirements by the end of each increment. By delivering working, tested software frequently, Agile processes have a tremendous advantage with their ability to elicit early feedback from the customer.

For applications subject to DO-178C certification, the iterative nature of Agile requirements elicitation impacts the safety analysis process. The safety analysis process requires as an input a rigorous requirements specification. Early and accurate determination of the DAL is critical since it determines the process objectives that must be satisfied for airworthiness certification. For example, Modified Condition/Decision Coverage (MC/DC) testing using requirements-derived tests is required only for DAL A software. Late determination that the software must be developed to DAL A standards may mean that insufficient schedule and budget remain to satisfy the additional process objectives associated with DAL A.

3.4 Responding to Change versus Following a Plan

Planning activities are part of Agile processes with the most detailed plans constructed just for the next increment. This is necessary because of the iterative nature of Agile requirements elicitation. It allows for great adaptability since the periodic re-planning activity gives the customer the opportunity to add, delete, modify, or reprioritize requirements.

Extensive planning, however, is a key element required for DO-178C certification. Project planning begins with completion of the Plan for Software Aspects of Certification (PSAC). The PSAC document is a comprehensive plan that states in detail how the development teams plan to approach development of the product to achieve all required process objectives. In addition to the PSAC, more detailed supporting planning artifacts are required including a Software Development Plan, a Software Verification Plan, a Software Configuration Management Plan, and a Software Quality Assurance Plan.

4 Model-Based Engineering and DO-178C Airworthiness Certification

The FAA recently adopted several new supplements to DO-178C to address certification issues related to model-based engineering, object-oriented technologies, formal methods and tool qualification. Model-based methodologies are of great concern to the FAA because it is unclear what role any simulation results derived from the models should play in the determination of airworthiness. The believability of the simulation results is in part a function of the fidelity of the model to the actual system. Models, however, can be constructed at different levels of abstraction to capture high-level requirements or various aspects of the architectural design, for example.

The certification of products developed using model-based methodologies is discussed extensively in RTCA DO-331, the recently released model-based development supplement to RTCA DO-178C [15]. DO-331 does provide some flexibility in how model-based methodologies may be used. A specification model may be used to explore the consistency and correctness of the modeled requirements. A design model may be used to verify various architectural details. It is acceptable to include a specification model with no design model, a design model with no specification model, both a specification model and a design model, or neither type of model (i.e. use no model-based methods).

DO-331 does provides a substantial opportunity to introduce Agile methods by distinctly separating the requirements processes and artifacts from the design process and artifacts by mandating that any specification model be distinct from any design model that is used for certification. As a result of this separation, any model from which delivered code was synthesized is considered a design model, not a specification model. DO-331 also stipulates that all top-level requirements must be in textual form.

5 The Model-Based Agile Process (MBA)

We propose a new software development process that combines key advantages of both agile development processes and model-based engineering methodologies
to produce a Model-Based Agile (MBA) process capable of satisfying FAA-mandated process objectives for software of all Design Assurance Levels. A key element of the MBA Process is the use of an Agile-style iterative and incremental approach to requirements elicitation, capture, and verification. Provided that an appropriate modeling tool is used that admits executable specification models (again, DO-331 forbids synthesizing deliverable code from specification models), one can start by developing use cases and iteratively refine them into executable models. These executable models will bring to bear the advantages of agile requirements elicitation while facilitating the complete capture of a set of requirements for a system before the detailed design and testing is begun.

As with a traditional agile process, the development team works closely with the customer to identify requirements and to develop acceptance tests that will be used to verify the correctness of the requirements as captured in the specification model. The acceptance tests are executed on the specification model, and the results used to verify correctness, completeness, and consistency of the specifications. It should be understood that the test cases used to exercise and test the requirements model would most likely not be directly applicable to any subsequent design model without substantial refinement due to their lack of detail. For example, messages might only contain message types for exercising the specification model while design model messages will require detailed values in the message body for exercising any design model.

For the MBA Process, we propose the use of a Unified Modeling Language (UML) tool such as IBM’s Rhapsody to capture these requirements as they emerge from face-to-face interactions with the customer. Requirements capture via UML has been shown to be an effective means of communicating requirements information among stakeholders. Moreover, the Rhapsody tool allows the construction of executable UML models using non-synthesizable components of UML such as sequence diagrams. Commercial UML tools such as Rhapsody also provide interfaces to industry standard textual requirements management tools such as IBM’s DOORS. This allows the developers to maintain traceability from the top-level textual requirements to the specification model as mandated by the FAA.

In our MBA process, the initial iterations are focused only on eliciting, capturing, and refining requirements for input into the safety analysis process. Once the customer and the team are satisfied that all requirements have been identified and verified, the safety analysis is performed to determine the DAL and identify any safety requirements. The safety requirements are integrated into the UML specification model and verified during the next iteration. A test coverage analysis on the model can be performed to ensure adequate testing of the specification model to the required DAL. Since only design models can be used for code generation, future iterations will focus primarily on implementation of the captured requirements.

As with our previous modifications to the requirements elicitation process to enable agile development of safety critical systems, the design process may be similarly modified to accommodate construction of the optional design model, if desired. The forced separation of specification models and design models permits the use a different modeling tool for the design model including one that is more amenable to code synthesis and formal verification, if desired. Once the textural requirements and specification model are completed, then the implementation of the requirements can proceed piecemeal with testing and verification of each unit, component, subsystem, and system in turn.

6 Impacts of the MBA Process on Certification

The first part of the safety process, which occurs primarily during the requirements phase, is the functional hazard analysis. This process is applied to the aircraft functions, not to the components design, which has not taken place at this time. The proposed agile requirements process can be extended to permit incremental functional hazard analysis as subsystem components are elicited. Since no design is taking place at this time, the safety processes that follow design processes, such as FEMA, will not be done prematurely.

Concomitantly, if the design proceeds using agile processes, it is possible, but not as clear, that incremental FMEA and FTA analysis will be possible. The problem is that analysis at this level is intended to generate additional safety requirements, if necessary,
and these probably should not be developed piecemeal. However, development within a given component or subsystem may be made small enough so that incremental FMEA and FTA analysis is possible.

One must also keep in mind that DO-331 also stipulates additional required process objectives specific to model-based methodologies that include verification of simulation cases (scenarios), simulation procedures, and simulation results with explanation of any discrepancies for both specification models and design models if present. The decision to use model-based methodologies impacts other process objectives as well. For example, if both a specification model and a design model are developed, one must establish forwards/backwards traceability starting with the top-level text requirement through the specification model, design model, source code, object code, and requirements derived tests. Configuration management and change control must be extended to include the modeling and simulation tools themselves and the set of tool configuration options selected.

7 Conclusions

The proposed Model-Based Agile process should facilitate the use of agile methodologies in the development of safety-critical systems. Moreover, the MBA process was developed specifically to be compatible with the FAA-accepted RTCA DO-178C airworthiness certification standard for airborne software and the RTCA DO-331 model-based engineering supplement. Given the relatively recent release of DO-178C and DO-331, it remains to be seen what consensus will emerge among practicing DERs as to what evidence will be considered an acceptable demonstration of satisfying the model-based engineering process objectives when it comes to certification of an actual aircraft.

To validate the MBA process, the authors are currently planning on developing a small safety critical system, such as an Unmanned Aerial System and ground controller, using this Model-Based Agile process to produce the artifacts necessary for FAA certification. This work will be conducted in collaboration with local DERs to ensure that acceptability of the process and process artifacts to a practicing DER.

8 References