Agile Accessibility: Ensuring accessibility throughout the Agile development process

Andrew Nielson, CSM, PMP
New Editions Consulting, Inc.
anielson@neweditions.net

Tyler Matney
New Editions Consulting, Inc.
tmatney@neweditions.net

Cindy Ryan
New Editions Consulting, Inc.
cryan@neweditions.net

Abstract

U.S. federal government agencies are increasingly migrating toward Agile methodologies for Information Communication Technology (ICT) development, creating a need for government IT professionals to revise their system development lifecycle approaches and related governance policies and practices. Like other forms of IT governance, Section 508 programs often struggle to force conformance validation designed for a traditional waterfall methodology into an Agile methodology. Seasoned Section 508 Program Managers accustomed to coordinating Section 508-related activities in the context of waterfall must still educate stakeholders about accessibility requirements while reconciling Section 508 conformance with sometimes unfamiliar Agile terms and concepts. After providing a brief overview of Agile concepts, this paper presents considerations and implications for incorporating accessibility throughout an Agile development process, including conformance governance and documentation.

Accessibility in Agile Methodologies

What is Agile?

Agile refers to a broad set of concepts that share the same four common values (from http://agilemanifesto.org/):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
The twelve principles of Agile ([http://agilemanifesto.org/principles.html](http://agilemanifesto.org/principles.html)) have led to various interpretations with application in a number of models, including those focused more on practices (e.g., Extreme Programming), those focused more on work flow (e.g., Scrum and Kanban), as well as a number of hybrids and combinations. The approach described below relies mostly on the Scrum approach (one of the most common models), to illustrate how accessibility practices should fit within Agile. Many of the same principles can be applied to other models of Agile.

### The Agile Approach to Accessibility

#### Accessibility as an integral part of Agile development

As depicted in Figure 1, a product is built in constrained increments of time known as sprints in Agile. A sprint is typically two to four weeks. Prior to initiating a product build, all requirements are added to a Product Backlog. When planning for a sprint, a subset of these requirements is moved to what is known as a Sprint Backlog. These requirements are then used to build a product increment within the sprint. These products may be “done” (ready to provide to an end-user) – or not (requiring additional work in another sprint). Daily scrum meetings are conducted to assist the development team in collectively building the product increment within the sprint.

An effective Agile approach to developing accessible software and web content starts with the fundamental understanding that an application or website must be functional and usable for all users, regardless of ability. Accessibility is a habitual expectation that all development team members accept as part of the normal course of work. Naturally, adequate training is necessary for developers to know how to write code that is accessible and conforms to accessibility standards. Likewise, testers must be able to follow a standard test process to reliably validate conformance and overall accessibility.

![Figure 1 - Accessibility should run throughout the Agile Scrum process.](image-url)
Key accessibility integration points within the Scrum process include:

1. The development team has knowledge of accessibility standards and test processes and access to accessibility expertise for consultation and validation at key milestones.
2. Iterative development and testing includes developing functionalities that work for all users, including those with disabilities, and testing to validate conformance to accessibility standards.
3. The “definition of done” for all product increments must include conformance to accessibility standards.

The sections below further describe other key elements of accessibility in Agile development:

Include conformance to accessibility standards in the “definition of done”

Agile development teams must form a shared understanding of what it means for work to be complete. The “definition of done” can vary significantly from one development team to another, depending on client or product owner requirements, development team preferences, and the environment. All product increments must conform to Section 508 standards in order to be considered “done.”

Incorporate accessibility standards in key artifacts

Regardless of an organization’s specific software development methodology, ICT accessibility must be included, at a minimum, in all key artifacts.

- **Requirements:** Whether described as elements of a Product Backlog, System-Wide Specifications, or other designation of requirements, the development team must establish accessibility standards as technical requirements for its ICT development.
  - Each Agile team determines the best method to document these requirements for the team. Some teams include accessibility requirements in acceptance criteria, as separate user stories, or in other ways.
  - Whatever the documentation method, the team must understand the necessity for incorporating accessibility in requirements.
  - As with other system requirements, teams should identify any deficiencies in the team’s knowledge, skills and abilities to implement accessibility requirements early in the process in order to better identify solutions to meet the accessibility requirements (e.g., training, hiring or subcontracting to an IT accessibility resource, working with an agency’s Section 508 office, etc.)

- **Design and Architecture:** Conformance to accessibility standards influence design and architecture decisions, whether it relates to choices about specific types of technologies or user interface (UI) design considerations. The development team must include accessibility standards conformance in the earliest stages of design and architecture to promote the ability to conform to the standards in the subsequent development of the application.
  - User experience (UX) designers must address all requirements (including those for accessibility) when developing wireframes and visuals.
Strong UX designers typically have experience or education in designing to include elements that are universally usable – meaning that they are designed for all user types, including those with disabilities.

- **Test Plans**: A standardized accessibility test process for ICT content informs test planning, serving as elements of test scripts and describing the specific processes necessary to validate that “working software” really will work for all users.

**Minimize subjectivity by following a standard accessibility test process**

A development team should adopt and implement a standard test process for each application and for all features or functions of ICT to avoid confusion and minimize subjectivity in validating conformance to accessibility standards. Documenting the standard test process and following the applicable test procedures throughout iterative development establishes a common understanding of accessibility requirements and helps improve confidence that the ICT conforms to the standards before it moves to production.

**Using Test-Driven Development and automated testing in Agile accessibility**

The concept of Test-Driven Development (TDD) is one of the common hallmarks of Agile methodologies. In a nutshell, TDD requires writing tests to validate that functional and technical requirements are met before actually writing any code.

Writing the tests before writing code encourages developers to focus more acutely on requirements before shifting focus to develop features. The process also tends to produce simpler designs as it discourages developers from developing code outside of the scope of requirements (only write tests that validate requirements, and only write code sufficient to pass the tests).

Assuming that accessibility is incorporated within requirements, TDD likewise requires understanding the accessibility requirements in order to write tests to validate that specific interface elements and content conforms to the requirements as they are developed.

Certain forms of TDD often make use of automated testing tools to facilitate code validation and increase the volume of tests that can be performed. A number of automated testing tools exist with rulesets devised to evaluate conformance with many of the WCAG 2.0 Success Criteria. Such tools can drastically decrease the time necessary to write and perform tests while improving the ability to identify flaws. However, in many cases, whether related to accessibility requirements or other functional requirements, some evaluations still require human cognition to determine whether certain features meet the requirements.

Given the automated testing tools currently available and the nature of some of the WCAG 2.0 Success Criteria, no amount of automated testing can completely eliminate the requirement for manual inspection of certain ICT content. Therefore, even high-functioning teams effectively using automated testing will require strong familiarity with accessibility in order to incorporate manual tests to evaluate conformance with all of the accessibility requirements.

**Provide Section 508 subject matter expertise**

Developers should be able to build accessible ICT and testers should be experienced in validating that it conforms to accessibility standards. Nevertheless, it may still be necessary to consult subject matter experts to confirm conformance or nonconformance to standards, help
troubleshoot and remediate issues and evaluate the risks associated with non-conformant work products.

Experienced members of the development team may serve as accessibility subject matter experts, or the development team may need to rely on subject matter experts external to the team. Each development team should identify, based on the overall skills and experience of the team, how to distribute the time and attention of the accessibility subject matter experts throughout the development process to assist when necessary and promote full compliance with accessibility standards.

**Major Differences Between Agile and Waterfall Development**

This paper does not seek to promote Agile over traditional waterfall development. Like waterfall, Agile also has distinct benefits and drawbacks. IT accessibility practitioners, Section 508 Coordinators, and Program Managers can take better advantage of benefits and mitigate drawbacks to promote accessibility by understanding them.

**Benefits of traditional waterfall development**

<table>
<thead>
<tr>
<th>Benefits of Waterfall</th>
<th>Accessibility Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Well-defined milestones and deadlines</strong></td>
<td>• Makes testing timelines more predictable</td>
</tr>
<tr>
<td></td>
<td>• Helps Section 508 Program Managers coordinate subject matter experts and testing resources across a portfolio of projects and activities</td>
</tr>
<tr>
<td></td>
<td>• Lends to the ability to coordinate IT accessibility resources under a more centralized management structure</td>
</tr>
<tr>
<td></td>
<td>• Helps facilitate governance and review of conformance at specific development milestones</td>
</tr>
<tr>
<td><strong>Standardized documentation</strong></td>
<td>• Facilitates location of accessibility-related information in standard process and system documentation (e.g., requirements, test plans, milestone reviews)</td>
</tr>
<tr>
<td></td>
<td>• Helps facilitate governance and review of conformance at milestones via documentation of conformance assessments</td>
</tr>
</tbody>
</table>
Drawbacks of traditional waterfall development

<table>
<thead>
<tr>
<th>Drawbacks of Waterfall</th>
<th>Accessibility Implications</th>
</tr>
</thead>
</table>
| Difficult, costly change | • Because accessibility testing often occurs late in the project lifecycle and only at specified milestones, accessibility-related defects often go unnoticed until after completion of accessibility testing  
• The later in the development lifecycle any defect is identified, the costlier it is to remediate; accessibility defects, in particular, can require significantly more time and effort to remediate than for those issues identified earlier in development; accessibility defects that make their way to production become exponentially costlier to remediate |
| Slow delivery and subsequent release schedules | • Because waterfall development projects tend to deliver entire collections of identified features and functionalities in a single production release, accessibility issues identified for remediation are typically grouped with similarly large collections of enhancements or modifications targeted for subsequent releases  
• Accessibility defects tend to have longer lifespans while awaiting the next release to include fixes to address the accessibility issues |
Benefits of Agile methods

<table>
<thead>
<tr>
<th>Benefits of Agile</th>
<th>Accessibility Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Adaptability</strong></td>
<td>• Shorter development cycles with smaller, incremental iterations of functionality provide the ability to reprioritize requirements to respond to evolving project objectives, including the ability to prioritize accessibility issues when identified</td>
</tr>
</tbody>
</table>
| **Immediate user feedback**                  | • Because Agile projects release functionality in small increments, users, including those with disabilities, begin to use the functions and features earlier in development and can provide feedback to developers  
  • Agile teams can identify and remediate accessibility issues in early development rather than perpetuating the defects throughout later stages of development |
| **Quicker delivery and shorter release timelines** | • Accessibility-related defects, once identified, don’t have to wait for a large collection of enhancements and modifications as part of a large subsequent release  
  • Shorter, more flexible release timelines can more easily accommodate updates to specifically address accessibility issues |
## Drawbacks of Agile methods

<table>
<thead>
<tr>
<th>Drawbacks of Agile</th>
<th>Accessibility Implications</th>
</tr>
</thead>
</table>
| **Ambiguous timelines** | • While one of the commonly held principles of Agile development instructs teams that each product increment should result in a “shippable” product, actual release schedules often fall to the whim of the development team, which can make it difficult to predict precisely when accessibility enhancements will move to production  
• Unclear release schedules can make it difficult to coordinate IT accessibility resources for those organizations that manage accessibility subject matter expertise under a more centralized structure  
• Organizations or project teams that attempt to apply a waterfall-oriented accessibility test process within Agile can find it difficult to conduct accessibility testing as a separate activity (i.e., attempting to test an application from top to bottom through a comprehensive accessibility test at each iteration); identifying Success Criteria applicable to only those elements developed and delivered within a particular product increment will enable the team to test only what is necessary to test as part of an integrated development and test process |
| **Dependence on team’s skills** | • Because agile teams tend to be smaller (4-9 developers/testers), some team members may need to perform more than one role, and not every team member may have specific knowledge of accessibility best practices  
• Accessibility subject matter expertise can become spread thin, especially across Agile teams; naturally, training for all team members to familiarize them with accessibility requirements can mitigate the need for consultation with accessibility subject matter experts |
| **Neglect of documentation** | • While organizations may still enforce documentation requirements, and Agile development does not preclude robust documentation, the ideology nevertheless focuses on functioning work products over documentation; inadequate documentation of accessibility requirements and test procedures can lead to inadequate incorporation and validation of accessibility |
Keys to Success in Implementing Agile Accessibility

To summarize, incorporating accessibility in Agile development methods will succeed when development teams view accessibility as an integral part of the process throughout the entire development lifecycle.

Key points of integration to successfully promote accessibility in Agile development:

- Include accessibility conformance in the “definition of done”
- Incorporate accessibility standards in key artifacts
- Minimize subjectivity by following a standard accessibility test process
- Follow TDD practices
- Provide Section 508 subject matter expertise to development teams, and ensure that all team members have adequate knowledge of accessibility requirements

Copyright notice

The 2017 ICT Accessibility Testing Symposium proceedings are distributed under the Creative Commons license: Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0, https://creativecommons.org/licenses/by-nc-nd/4.0/).

You are free to: Share — copy and redistribute the material in any medium or format.

Under the following terms: Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial — You may not use the material for commercial purposes. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.