

Agile Design Processes and Project Management

Theresa A. Cullen

Due to the changes in and flexibility of computing today, software engineering and instructional design have made major changes in their approach to development. This evolution to a knowledge economy required processes to change from approaches where planning and communication happen up front to more agile processes where projects are completed in smaller chunks with greater communication between team members and clients. Adopting these agile processes may enable instructional designers to create more flexible designs that better meet the needs of clients and allow for greater collaboration with others involved in the development process (e.g., UX designers, programmers, media production).

What is Agile Development?

Agile development has its roots in a document written by 17 people at a retreat in 2001, when a group of software developers met together to decide how projects should be approached. They were frustrated by static lists of tasks that were developed early in projects that could not easily be changed, creating a process that lacked flexibility and feedback. This kind of static list was known within the industry as “Waterfall,” referring to the slow trickle of development that happens from a prescribed list of designs (Nyce, 2017). The group had championed different approaches during their extensive careers, but it was not until they came together in 2001 that they laid the groundwork that would change how many products were designed. They agreed that good programming and design had 12 key principles. As agile processes have been adopted by other fields such as business, education, health care, finance, and marketing (Oprins et al., 2019), the foundation of the approach has been based on these 12 principles, which make up the Agile Manifesto (Beck et al., 2001):

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.
7. Working software is the primary measure of progress. Agile processes promote sustainable development.
8. 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.

Summary of the 4 Values of the Agile Manifesto

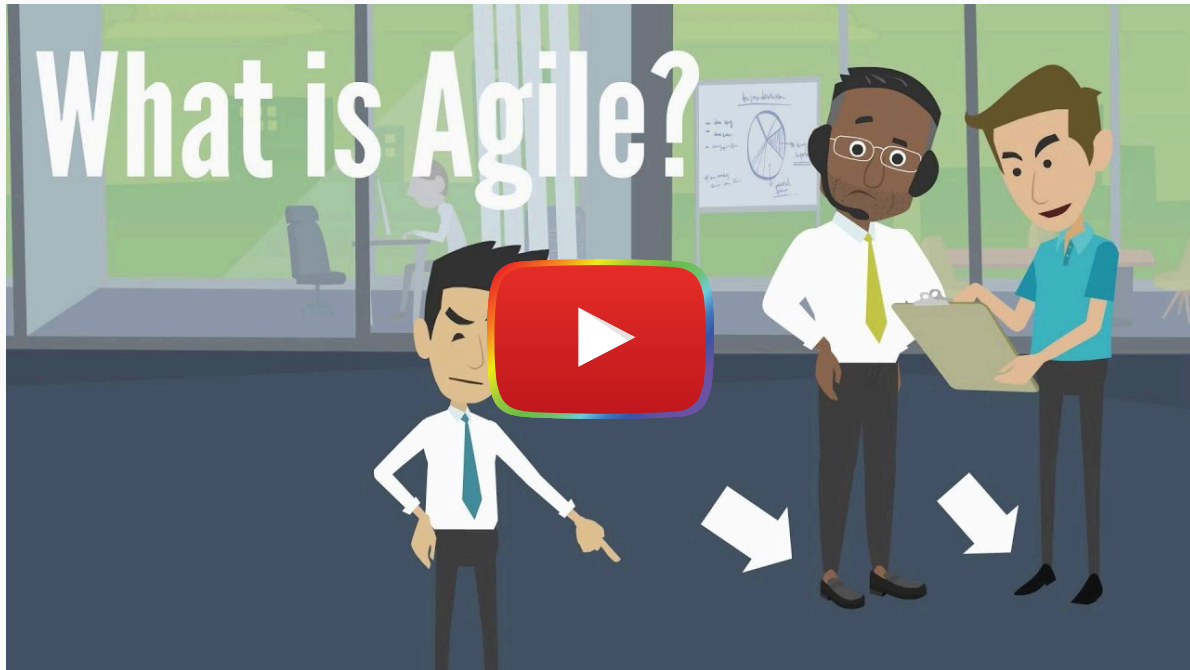


[Watch on YouTube](#)

Agile has become a generic term for processes that adhere to the agile principles laid out in this Agile Manifesto, much like ADDIE is a basic process for instructional design or design thinking is a generic process for approaching design projects. For example, there are different instructional design approaches (e.g., Dick and Carey; Morrison, Ross, Kemp, and Kalman; and Smith and Regan), but they all include basic principles such as needs analysis and evaluation. The same is true of agile processes, as there are different approaches to realize the key components of the agile manifesto in practice.

One of the most prevalent agile approaches is called Scrum, which is used by businesses both in software engineering and other areas. The value of Scrum is that it has clear roles for different individuals and a variety of agile processes used in design. Even as agile processes are repackaged in a variety of products (Scrum, Adaptive Project Development [ADP], Kanban, etc.) they all adhere to these 12 principles that define agile development (Portman, 2019). Key components present in all products include constant communication with the client, support for the development team, a focus on deliverables that are fast enough to produce forward motion, and a focus on developing a reliable and robust product.

Review of Agile Principles



[Watch on YouTube](#)

To examine designing through an agile framework, let's look at some key components of Scrum. Scrum is defined as "a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value." (Schwaber & Sutherland, 2017). The Scrum processes and roles defined in Table 1 support agile processes in practice.

Table 1

Key Terms Related to Scrum Processes

Backlog	A list of tasks that need to be completed as part of the project. This list is prioritized by team members at the beginning of each sprint. The backlog allows the team to communicate priorities with a client and accurately predict the timeline of a project.
Sprint	A short interval of time (often two weeks) where the team decides on a set of backlog tasks to achieve as a team. An example sprint dashboard, representing the backlog and completed items on a project, is shown in Figure 1. An example sprint team is shown in Figure 2.

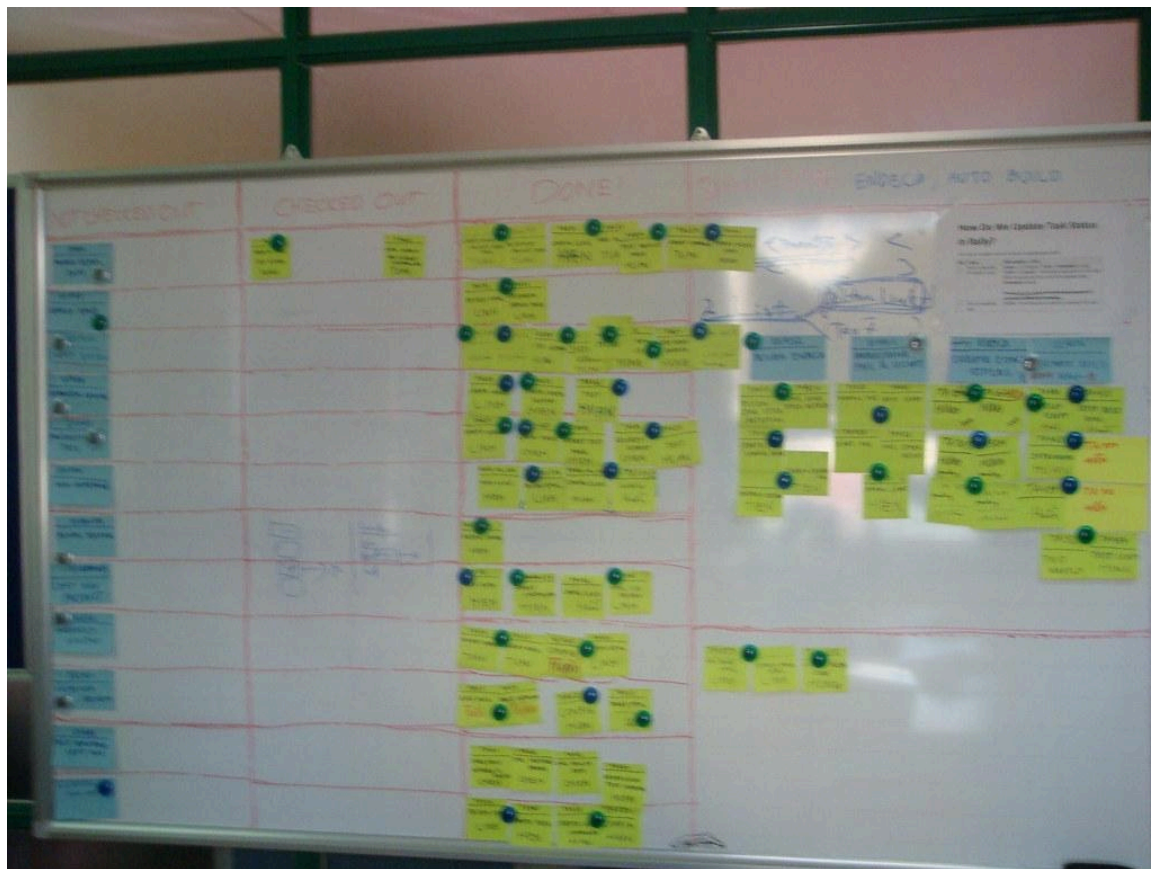
Sprint Retrospective	As in all agile processes, reflection is an important part of Scrum. At the end of each sprint, the Scrum team takes time to review how the processes went and make plans to improve processes in the future. They ask questions like, "What did we do well and what should continue?" or "What could we improve?"
Stand Up	A daily meeting that is designed to last 15 minutes or less to update the team on accomplishments, problems, and status. It is called a stand up because it is meant to be kept short by having everyone stand during the meeting. During the meeting, team members ask questions like, "What did I do yesterday?", "Am I blocked by anything?", or "What do I plan to do today?"
Scrum Master	The person managing the Scrum team who makes sure that all team members are getting the resources they need and adhering to the team plan.
Definition of Done	This is an agreed-upon level of fidelity for product production in each sprint. The team must agree what is the expectation of each team member's work.
Product Owner	This is the person who is responsible for the backlog. They work to develop an accurate timeline and keep the project on track. The Product Owner cannot be the Scrum Master.
Scrum Team	All of the people involved with the design of the product. This could include developers, UX designers, QA, and instructional designers, given the project. Different sprints could have different team members.

Scrum Basics and Roles



Figure 1

Example Sprint Dashboard



Note. "[Sprint dashboard](#)" by [Tiendg](#) is licensed under [CC BY-NC-ND 2.0](#)

Figure 2

Example Sprint Team



Note. *"The Agile PM Game (Aug '11)"* by [VFS Digital Design](#) is licensed under [CC BY 2.0](#)

Need for Agile Processes in Instructional Design

By nature of their work, instructional designers (IDs) collaborate with diverse groups such as UX designers, programmers, media creators, and a variety of subject matter experts. It is to be expected that instructional design processes may be influenced by those other fields and IDs may even be required to use processes from other disciplines such as programming. One problem that many teams find is the need for quick results and to maintain good communication with a client throughout the design process. Adnan and Ritzhaupt (2018) summarized the criticism that traditional instructional design approaches like ADDIE are not flexible and are less able to produce dynamic projects—especially those that require flexibility and updating. The flexibility of an agile approach allows for both speed of design, but also better repurposing and tailoring for different design problems. Being knowledgeable about agile processes in both instructional design and other fields will enable better team collaboration and client communication (Oprins et al., 2019).

Fernandez and Fernandez (2006) examined agile versus traditional approaches to project management. In a traditional approach, instructional designers may meet with a client at the beginning of a process, and then create designs, only to unveil them when the project is done. They found that these traditional or waterfall approaches did not meet the needs of the fast-changing markets and the need to have products available quickly to stay competitive. They found that business practices were changing towards shared responsibility and team collaboration. Leaders were no longer in charge of projects, but instead they were in charge of teams that have different skills but were all committed to making the client's project a reality. Agile is a mindset above all else that includes shared responsibility and design, regular client communication, and embracing change throughout the process.

While an agile approach is different from traditional instructional design approaches, our field has a history of flexible design approaches too. The most notable was rapid prototyping, proposed by Tripp and Bichelmeyer in 1990. Rapid

prototyping comes from software engineering's approach to design where they create prototypes, test them, and then quickly revise them based on the results. Tripp and Bichelmeyer (1990) argued that instructional problems cannot be defined fully at one time and therefore a new flexible approach would allow for more adaptability and response to deep learning issues that become apparent through the design process.

There are many similarities between rapid prototyping discussed in 1990 and agile processes now, specifically, the focus on the product and being open to change in design through regular communication with clients. That is not to say that most instructional designers do not communicate with clients regularly, but rather that choosing an agile approach places the client at the forefront while still not conflicting with key components of the instructional design process like establishing need, breaking down learning processes, and designing effective evaluation.

Now that you have some of the terminology and history, let's compare traditional instructional design approaches to agile approaches in Table 2. Using the ADDIE acronym to compare how each method approaches important tasks in designing effective instruction allows us to see that both approaches deal with the same information and issues and both can produce effective instruction.

Table 2

Comparison of Traditional Instructional Design to Agile Processes

Task	Traditional Instructional Design	Agile Processes
Client Involvement	Utilizes a single or a few major delivery points and feedback points with the client.	Relies on delivery points to the client in short time intervals (often 2 weeks). Focuses on constant iterations.
Analysis	Perform needs and task analysis at the beginning of the design process. Emphasizes depth.	Generates user stories throughout the process to illustrate needs which are revisited at the beginning of each sprint. Emphasizes speed.
Design	Communicates overall design by producing design documentation at the beginning of the process that is used throughout the entire process.	Communicates overall design by creating a backlog of tasks that the development team chooses from to set goals for each sprint. Design is revisited at the end of each sprint.
Development	Produces large parts of a project at once based on learning objectives or content topics. Emphasizes producing a complete learning unit.	Produces small components of content throughout the process focused on delivery to address items in the backlog. Emphasizes forward movement on content development.
Implementation	Implements a complete project or complete module with all parts of instruction and assessment complete.	Releases completed components at the end of each sprint. In a web or app-based design, the team can "push out" parts of the project regularly. The release may not produce a complete product at every update, but instead focuses on continual improvement of released content.
Evaluation	Evaluated as a complete unit with feedback from users and clients.	Engaged in constant evaluation due to the retrospective process at the end of each sprint. Project is constantly

going through feedback loops and adjusting based on client and updated user stories at each sprint.

An Example of Scrum in Practice

At a university where I worked, our Information Technology department used Scrum processes to manage large projects. The department set out to redesign the student and faculty portal. They started by having focus groups of faculty and students about how they used the existing tools and what they thought was missing. This would be very similar to learner and needs analysis processes in instructional design. The team used these focus groups to create user stories. Each user story highlighted a different stakeholder and what they needed from the product they were designing. Then, the Product Owner took that feedback and created a backlog of tasks with different priorities that had to be completed (see Figure 3 for an example backlog). They created these with input from all members of the team with a goal of forward movement and the ability to release improved functionality at regular intervals.

For example, in this project, the first sprints focused on interface design. Members of the sprint team included people from marketing and web design to make sure that the overall look matched the brand and other components used by faculty and students. After several sprints to design the look, the product owner moved people down the list of priorities to begin to design the functionality. Not all tools were redesigned at once. In fact, the Scrum team decided to focus on student tools first like enrollment and financial services. About halfway through the year-long project, members of the Scrum team visited faculty and student meetings to ask for input on what they had designed so far. They announced that it would be several months until faculty functions would be the priority in the backlog and continued to refine student functions based on feedback.

Throughout the process, the Scrum team published new tools and functions in the portal and had students and faculty start using them. They gained feedback, reflected on what they had already designed and changed their priorities and the product moving forward. Redesigning an entire university records and communication portal is a major undertaking, but by using Scrum processes the team was able to show results and continue to tailor their product to their stakeholders. They were also able to push out different usable products throughout the year without waiting for the entire project to be finished.

Figure 3

Example Backlog for the Portal Project

Backlog for Portal Project

In Progress	Soon	Future	Completed
Faculty: Advising	Student: Billing	Parent: Billing	Overall Interface Design
Faculty: Course Listing	Faculty: Book requests	Adminisitrator: Reports	Log in and authentication
	Faculty and Student: LMS launch	Student: Jobs	Student Feature: enroll
		Student: Organizations	Student Feature: advising
January 2 - 15	January 20- February 4th	February 7th - 21st	Oct 1 - present
Retrospective: January 16	Retrospective: February 5th	Retrospective: February 22nd	
Product Owner communicates with client and reviews backlog for next sprint.	Product Owner communicates with client and reviews backlog for next sprint.	Product Owner communicates with client and reviews backlog for next sprint.	

Conclusion

Following more agile processes can be a choice by an instructional designer, or it can be part of a company's culture. Agile processes are not at conflict with good practices in instructional design. In fact, steps like creating a backlog that prioritize features, gaining customer feedback on designs during the process, and being reflective is good practice. Taking an agile approach to instructional design can benefit the team dynamic and instructional product. The team dynamic is improved through improved client communication, flexibility, and creating components that could be better reused in other projects with similar user stories. Tripp and colleagues (2016) found that a workplace that embraces agile processes could increase job satisfaction among its employees. Fernandez and Fernandez (2006) found that agile made projects more adaptable and able to produce products faster. Oprins and colleagues (2019) point out having an agile approach emphasizes the importance of people in an organization, builds empathy, and guards against automation. Agile processes, when followed, can improve team communication and keep team members from pursuing dead ends or wasting important time because all of the team were not "on the same page."

There are also downsides to following agile processes. Regular communication with team members and clients takes time and can slow down some aspects of design. Since agile processes are designed to always be flexible, it can be frustrating to live in constant change, even if it produces a better product. For many, following agile processes requires a change in approach and communication style which can be difficult. Finally, agile is a buzzword: There are many companies that say that they use agile processes but do not have trained individuals, necessary resources for team members, and do not embrace the agile mindset. This kind of workplace can be incredibly frustrating because it can produce unpredictable results. Agile processes take commitment from all stakeholders and the leaders of an organization or company.

Next Steps

Instructional designers have many opportunities to become more knowledgeable of agile processes.

First, there are many resources available about agile processes and thought processes available online. In addition to the Agile Manifesto itself, many Scrum professionals start with the Scrum guide (<https://edtechbooks.org/-rZPW>) to learn about agile processes in practice.

Second, talk to people working in the industry. Reach out to alumni from your instructional design program and ask them about the project management processes that they use.

Third, for those interested in pursuing an agile philosophy further, consider pursuing a professional certification as a Scrum Master (<https://edtechbooks.org/-jNf>). The certification can be earned after taking a short workshop about agile processes and then passing an exam. The workshops can range from \$1000 to \$5000, but the training produces a certification that can be included on a resume or LinkedIn profile.

Takeaways

As an instructional designer, you will work with a variety of teams within a company (instructional designers, content experts, trainers, HR, etc.). Understanding different ways that projects are managed within a company not only makes you more valuable within the organization, a better team member, but also helps you to be more flexible to your approach to instructional problems. Many companies that have adopted this approach would value instructional designers who are both aware of and have practiced agile approaches to be able meet the changing needs of the organization and its clients. If this is the way that you enjoy working, then become more knowledgeable on agile processes and look for a company that clearly integrates it into their culture.

Links and Resources

- Scrum Guide (A good place to start) <https://edtechbooks.org/-rZPW>
- Scrum Alliance (<https://edtechbooks.org/-jNf>).
- Agile Instructional Design Course on LinkedIn Learning <https://edtechbooks.org/-XDT>

Agile Activities

Following are six activities designed to help you think critically about agile processes:

1. (A collaborative slides version of this activity is available view only at <https://bit.ly/agileactivity>. To be able to edit, choose *make a copy* from the file menu, then save it to your own Google Drive.)

An instructional design project on training about workplace bullying was handed to a team that had been designed in a traditional way. The new team uses agile processes. How could they break down this project into smaller chunks (aka create a backlog) to allow for prioritizing parts of the task and providing logical places to stop and receive feedback from the client throughout the process?

- The tasks developed by the traditional team included:
- Explain terminology: bullying, bystander, and victim.
- Outline the roles that each individual takes in a bullying incident.
- Outline what employees should do if they witness bullying.
- Outline what employees should do if they experience bullying.
- Create a design for the look of the materials to create consistency between a face to face and online learning module.
- Create a list of resources available for additional information and training.
- Outline the company policies on bullying.
- Outline the processes for reporting bullying.
- Create example stories or cases with different perspectives (bully, bystander, and victim).
- Develop face-to-face workshop that will last 90 minutes.
- Develop an online tutorial that can be used to document compliance.
- Develop discussion questions for in person training.
- Develop quiz questions for an online module which can be recorded for compliance.
- Create a video with a bullying scenario from the workplace for in person training.
- Create a video with a bullying scenario for the online training.
- Develop a script and support materials for a face-to-face facilitator.

After breaking up the task into smaller groups, then plan the backlog. Many companies use a table design to show the progression of a project. Assign priorities to the groups you created above and explain why you arranged them that way.

In Progress	Soon	Future	Completed

Did the original team forget any task they might need? What were the tasks? How does this agile process help to refine the project and identify gaps?

2. You are designing a remote learning activity to be used by a teacher for a middle school classroom. Create a user story for the stakeholders involved. Think about parent, student, teacher, and curriculum coach. Explain what their needs may be and think about how your design may need to incorporate those needs.

3. Agile teams have been shown to be more effective than traditional teams. Why do you think this is the case?

4. Explain how agile processes value the relationship with the client.
5. At the end of a sprint, an agile team takes time to do a retrospective before starting the next group of tasks. How does scheduling time to reflect on a project in process increase efficiency when designing?
6. Read over the agile manifesto. Give examples of how it honors collaboration and the value of stakeholders.

References

- Adnan, N.H. & Ritzhaupt, A.D. (2018). Software engineering design principles applied to instructional design: What can we learn from our sister discipline? *TechTrends*, 62, 77–94. DOI: 10.1007/s11528-017-0238-5
- Agile Manifesto (2019) Agile Manifesto retrieved from <https://agilemanifesto.org/iso/en/principles.html>
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). *Manifesto for agile software development*.
- Budoya, C.M., Kissake, M.M., & Mtebe, J.S. (2019). Instructional design enabled agile method using ADDIE model and feature driven development method. *International Journal of Education and Development using Information and Communication Technology*, 15(1), 35–54.
- Fernandez D.J & Fernandez J.D. (2008) Agile project management—agilism versus traditional approaches, *Journal of Computer Information Systems*, 49:2, 10–17, DOI: 10.1080/08874417.2009.11646044
- Monteiro, C. V., da Silva, F. Q., dos Santos, I. R., Farias, F., Cardozo, E. S., do A. Leitão, A. R., Neto, D.N. & Pernambuco Filho, M. J. (2011, May). *A qualitative study of the determinants of self-managing team effectiveness in a scrum team*. In Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering (pp. 16–23).
- Nyce, C. M. (2017, December 8). *Agile software development: A history*. The Atlantic. Retrieved from: <https://www.theatlantic.com/technology/archive/2017/12/agile-manifesto-a-history/547715/>
- Oprins, R. J., Frijns, H. A., & Stettina, C. J. (2019, May). *Evolution of scrum transcending business domains and the future of agile project management*. In International Conference on Agile Software Development (pp. 244–259). Springer, Cham.
- Portman, H. (2019). *A project manager's guide to agile methodologies*. Retrieved from: <https://thedigitalprojectmanager.com/agile-methodologies/>
- Tripp, S. D., & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design strategy. *Educational Technology Research and Development*, 38(1), 31–44.
- Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job satisfaction in agile development teams: Agile development as work redesign. *Journal of the Association for Information Systems*, 17(4), 1.
- Sutherland, J., & Schwaber, K. (2013). The scrum guide. *The definitive guide to scrum: The rules of the game*. Scrum.org, 268. Retrieved from: <https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide>
- Willeke, M. H. (2011, August). Agile in academics: applying agile to instructional design. In 2011 Agile Conference (pp. 246–251). IEEE. doi: 10.1109/AGILE.2011.17.



This content is provided to you freely by EdTech Books.

Access it online or download it at https://edtechbooks.org/id/agile_design_process.