An Agile ISD Process to Develop a Medical Simulation

Dr. Susan Dass
ICF
Fairfax, Virginia
Susan.Dass@icfi.com

Victor Cid
National Library of Medicine
Bethesda, MD
vcid@nlm.nih.gov

ABSTRACT

The Bethesda Hospitals' Emergency Preparedness Partnership identified a need to design training systems for hospital emergency management scenarios. As part of this partnership, the National Library of Medicine was challenged to develop an engaging, learner-centered simulation to specifically address hospital procedures for highly infectious diseases (HIDs) for multiple healthcare roles. A cognitive task analysis was used to collect the rich psychomotor, procedural, and cognitive data necessary for the design of a serious game for handling HIDs. Based on that data, a proof of concept prototype was developed in Second Life. That success has led to continued development. In this paper, we discuss the use of a novel Agile ISD process used to develop a new prototype using Unity 3d as the platform. An agile process was explicitly followed for this project to accommodate the limited budget that had abundant project visions. It was anticipated at the outset that the project would likely require continuous project re-scoping and refinement of product requirements that would need to be addressed smartly and efficiently. Furthermore, questions about fidelity in terms of visual representation as well as instructional integrity would still need to be addressed simultaneously. The project was conducted within a flexible contract environment that accommodated the anticipated re-scoping and refining during development. An overarching vision statement helped keep the product development moving forward with evolving requirements. The agile sprint process was invaluable for iterative reviews throughout the development process which afforded changes and redirection. However, usability testing revealed a significant change that was not noted as part of the iterative review process. This paper will briefly describe the Agile ISD process, show how it was applied to develop medical training on doffing personal protective equipment, share lessons learned, and briefly discuss flexible contract environments.

ABOUT THE AUTHORS

Susan Dass, Ph.D., is a Senior Instructional Designer, Technical Specialist at ICF. Dr. Dass received her Ph.D. in Education and her M.Ed. in Instructional Design. She has over 17 years of experience in instructional design supporting both private and government entities. Dr. Dass has many publications including co-authoring three book chapters that explore the design of help-seeking and self-regulated learning skills in 3D learning environments as well as faculty adoption of virtual worlds as a learning environment. Current projects include designing 3D serious games for the medical industry and exploring data analytics for learner performance and course improvements.

Victor Cid is a Senior Computer Scientist at the Disaster Information Management Research Center, Specialized Information Systems Division of the National Library of Medicine. His R&D work involves hardware and software solutions to improve the resiliency of health facilities and systems in the U.S. and abroad. Over the last 11 years, he has been researching and applying computer simulation and gaming techniques to training the health workforce for crisis events.

An Agile ISD Process to Develop a Medical Simulation

Dr. Susan Dass
ICF
Fairfax, Virginia
Susan.Dass@icfi.com

Victor Cid
National Library of Medicine
Bethesda, MD
vcid@nlm.nih.gov

INTRODUCTION

The Bethesda Hospitals' Emergency Preparedness Partnership identified a need to design training systems for hospital emergency management scenarios. As part of this partnership, the National Library of Medicine (NLM) was challenged to develop an engaging, learner-centered simulation to specifically address hospital procedures for highly infectious diseases (HIDs) for multiple roles. Within NLM, the Disaster Information Management Research Center (DIMRC) provides the resources for disaster remediation and conducts research on information solutions to support disaster preparedness. DIMRC took the lead within NLM to explore disaster prepared solutions for HIDs. The solution would provide training that addressed hospital staff preparedness for suspected HIDs patients from point of arrival at the hospital to patient care outfitted in personal protective equipment (PPE). DIMRC decided to pursue virtual worlds (VWs) as the information technology with the intention to push to a virtual reality (VR) delivery mechanism if the preliminary desktop prototype proved plausible and advantageous. The initial prototype which focused on donning PPE was developed in Second Life and received positive reviews from NLM management and the BHEPP member staff. While Second Life afforded DIMRC to explore the use of virtual worlds with in-house resources, Second Life ultimately presented technological and implementation challenges. For example, customizing the user interface had limitations, some behaviors could not be readily programmed, and there were privacy concerns. Therefore, DIMRC decided to use Unity 3d as the platform in the next phase of the project which would focus on doffing PPE.

DIMRC explicitly requested an agile development process, i.e., a process that could accommodate change and support a collaborative team approach between DIMRC and ICF. An agile process would also perceivably accommodate a limited budget that had abundant project visions. Re-scoping goals and expectations were anticipated as design and development challenges were encountered and addressed during the project. Fortunately, the contract environment was flexible and could readily accommodate changes from rescoping. A flexible contract environment allows for refinement or re-direction of the project requirements as need be without need for a contract modification.

In this paper, we will first describe the Agile Instructional Systems Design (ISD) process applied in this project. An Agile ISD process offers a relatively new way to apply the more common sequentially-based ISD model of analysis, design, develop, implement, and evaluate (ADDIE) process. While ADDIE professes to be flexible, the Agile ISD process inherently supports flexibility. The Agile ISD process still relies on the same steps and produces some of the same outputs as with the ADDIE process, such as a design document, but development happens with more flexibility to jump across steps and accommodate change. Next in the paper, we will present how the Agile ISD process was applied to develop a simulation-based training on doffing PPE and share some lessons learned and insights gained when compared to the ADDIE process. Lastly, we will comment on the contract environment for an agile process.

AGILE ISD PROCESS

In this section, we will first review the most common instructional design and development model: the ADDIE Model. Then we will look at the Agile Software Development which is the basis for the Agile ISD process to understand its values, principles, and philosophy. In closing, we will describe the Agile ISD Model and processes used in this project with a comparison to the ADDIE Model.

ADDIE Model

The ADDIE model, shown in Figure 1, was designed as a waterfall process calling for a complete deliverable at the end of each stage: analysis (A), design (D), develop (D), implement (I), and evaluate (E) (Piskurich, 2006). Deliverables are provided to the client for review and their subsequent comments addressed before moving to the next stage. It is not unheard of for some parts of a deliverable to be explored or developed early, out of the normal stage cycle, but this is adhoc and the model does not account for this type of aberration. Some may note that relatively early in the ADDIE process a prototype could be created to reduce risk but commonly not all content or functionality may be presented. Therefore, ripple effects of a change late in the process can mean re-work in other areas of the product,

making for a potentially inefficient process. To mitigate potential inefficiencies and associated additional costs, contracts often include stringent requirements for a deliverable to be 'accepted' by the client before moving on in order to preclude ripple changes. Other costly re-work could be introduced if strict adherence to this model is followed. For example, the ADDIE model does not require user testing until the implementation stage. Technical difficulties or other user-derived problems could become apparent late in the process, and costly re-work would be required. These inefficiencies and costly updates could be reduced by using an agile process. Let's look at the Agile Software Development philosophy to understand how that type of process could alleviate these inefficient, costly updates.

Figure 1. ADDIE Model

Agile Software Development

The Agile Software Development process has been around for many years but began formalizing in the 1990s. Then in 2001, a group of software developers "sympathetic to the need for an alternative to documentation driven, heavyweight software development process" created and published the Manifesto for Agile Software Development (Highsmith, 2001). The process calls for an iterative development and review of functioning pieces of the overall software package as opposed to a single product delivery for review at the close of the project. In this iterative process, pieces of the product are prioritized and selected for development followed by a review with stakeholders and potential end users. This piecemeal build with review is repeated throughout the duration of the project. The process has four core values as provided in Table 1. These core values call for individuals to self-organize and work together; produce functioning software that can be tested while foregoing comprehensive documentation; review the iterative products with stakeholders such as the customer and end users throughout the development process; and lastly embrace change. Project planning is dynamic, high-level with more clarity and specificity added as the iterative development process continues. Product direction is guided by the results of the iterative reviews.

Table 1. Agile Software Development Values

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

With those values in mind, this same group of software developers also identified 12 guiding principles. While many are software specific, some principles appear to have crossover application to instructional design and development. The example crossover principles in Table 2 explicitly support the Agile Software Development values with perhaps the exception of the last example principle which deals with team and process improvement. At the end of each iterative product review; the team discusses what went well and what went wrong with the intention of identifying improvement actions. With these values and guiding principles in mind, and recognizing the ADDIE production model for instructional design, we developed an Agile ISD Model as explained in the next section.

Table 2. Agile Software Development Example ISD Crossover Principles

- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Business people and developers must work together daily throughout the project.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile ISD Model

While other agile models and implementations may exist (e.g., Ackles, 2018; Allen, 2012), the Agile ISD Model used in this project draws from both the Agile Software Development philosophy and the ADDIE Model. As shown in Figure 2, the model consists of three overlapping stages. These stages have similar tasks as found in the ADDIE Model. Stage 1 is Analysis & Ideation which kick starts the project. Stage 1 encompasses tasks normally associated with the analysis and design stages in the ADDIE Model. Stage 2 is associated production and refinement based on reviews. Stage 3 is usability and requirements testing for subsequent release. The stages are purposely overlapping indicating that while one stage is in progress accomplishing certain tasks, another stage could simultaneously be in progress accomplishing different tasks; the iterative deliverables could be from any stage at any point in time. For example, when a particular type of Stage 1 analysis is complete such as completing an analysis on technical requirements, follow-on Stage 2 activities such as creating the digital framework to house the online training product could begin development. While this Task 2 activity may begin, other Stage 1 analyses could still be underway such as determining learning objectives. Hence the overlap in the stages. At the center of each stage is the sprint process. Sprints are the relatively short development time cycles for the iterative process. Sprints have their own process: planning what is to be done, then executing that plan, followed by the review of that piece of the product, and then closes with a team reflection to improve and evolve their process. Sprint results are what drive the direction of the product.

Figure 2. Agile ISD Model

The other key element to the Agile ISD process that drives the product direction is the use of Stakeholder Stories. The needs of a learning product are defined through these stories as perceived from the different stakeholders. Stakeholders can include supervisors, managers, learners, and development team members (instructional designers, programmers, user experience (UX) designers, and graphic/media/animation experts). These stakeholders provide their needs in the format: *As a <role>, I want to be able to <goal> so that <benefit/reason>*. These Stakeholder Stories identify who the stakeholder is, what they want, and to what end. Early in the project, the stories are generally broad but then become more specific as the project progresses. For example, an initial stakeholder story such as: *As a learner, I want to practice doffing as the Caregiver so I can develop automaticity in the procedures*. This broad statement helps to identify the problem space but later this story may transition to: *As a learner, I need to be able to select each piece of doffing equipment so I can practice the procedure step-by-step.* This transition gives more specificity to the simulation need and refined direction to the project.

The Agile ISD Process

After validating that the problem solution is indeed best addressed through a learning intervention, the Agile ISD Process then begins by developing an Agile ISD Management Plan. The purpose of the Agile ISD Management Plan is to focus the Manager and team members on systematically identifying, tracking, and documenting the broad scope of the learning product needs from the start of the process. Doing so reduces the risk of unknown challenges. The Agile ISD Management Plan is a living document that is updated throughout the development process as new information is acquired. The Plan has several key sections:

- The Vision Statement describes what will be accomplished, why it's valuable, and defines what success looks like.
- The Stakeholder and Team Members section lists all the stakeholders and members along with their role and responsibilities.
- The Implementation Plan describes the activities necessary to deliver and implement the learning product.
- The Evaluation Plan describes how the learning product will be assessed continuously against performance requirements after implementation.
- The Constraints section lists the factors that will limit or control the scope or delivery of the learning product.
- Milestones and Deliverables outlines a preliminary schedule with any known deliverables and hard deadlines.
- Costs and Budget begins the estimation of costs for development, implementation, and maintenance (if applicable) to compare against funding.
- The Backlog is a list of prioritized Stakeholder Stories.

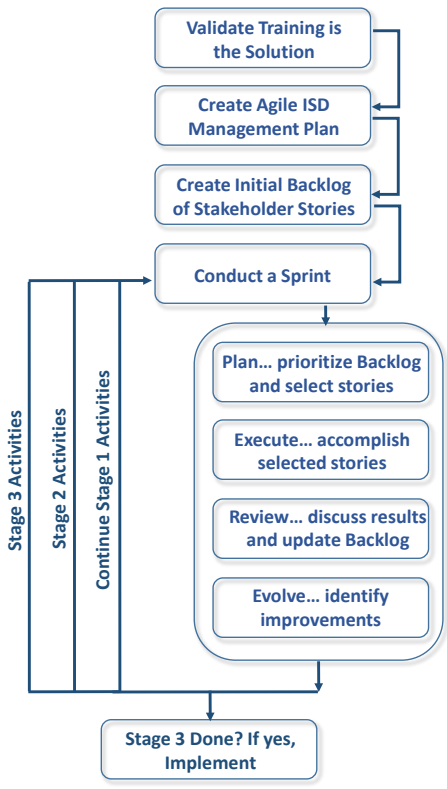

Figure 3. Agile ISD Process

The project manager will create the Management Plan except for the Backlog. The Backlog is a team activity. The team and stakeholders initially brainstorm to identify all Stakeholder Stories that describe the needs to develop the learning product as best known to date. The stories can range from describing how a learner might need to access the learning product (from a computer, on paper, in the field, on-the-job, etc.) to describing the learning goals (terminal and enabling learning objectives, performance measures, etc.). Stakeholder Stories can also spawn other stories, such as the need for a job aid or a customized animation. The stories are maintained in a single location called the backlog. The Backlog represents the learning product development needs. The Stakeholder Stories are prioritized based on criticality, value, prerequisites, available resources, and other factors depending on the learning product. High priority Stakeholder Stories are then selected to be accomplished within a sprint. Multiple sprints are used to develop the learning product.

A sprint is conducted over a set period of time, typically two to four weeks. Sprint durations can vary depending on the overall learning product timeline. For short learning product development timelines, the team will need to shorten the sprint duration. The Agile ISD process is useful for these shorter development timelines for the same reasons that it is useful for longer development timelines: it offers flexibility to adjust according to sprint results. Sprint duration is also controlled in part by the current tasks at hand. If a single task is longer in duration, then the sprint duration can be extended to accommodate it or the task broken into two parts.

During the sprint, Stakeholder Stories are accomplished that progress the development of the learning product. Several Stakeholder Stories may be executed in a single sprint depending on the availability of resources and team members. For example, while one team member is locating materials, another could be developing the lesson activity. The outputs created in each sprint are used as inputs and evolutions for future sprints and follow-on stages. Findings and review comments from each sprint are summarized and maintained in the Backlog in case decisions need to be re-reviewed for whatever reason. The later the sprint is in the project development, the more polished and refined the learning product becomes through repeated evaluation and continuous improvement as the product pieces evolve through the different production stages. This includes conducting Stage 3 activities such as usability and requirements testing on just pieces of the product. When all pieces of the product have successfully completed Stage 3 activities, the product is then ready for implementation.

Agile ISD vs ADDIE

While the Agile ISD Model may mitigate some shortcomings of the ADDIE Model, it is viewed as just another instructional design production model option in the toolbox. ADDIE is a very methodical and predictive process. Most clients are familiar with the process and trust the results from applying the process. Clients commonly have limited involvement in the design and development process which perceivably saves them time as long as problems don't occur, especially at the point of product delivery. ADDIE is best when the requirements are well-defined, the content is stable (not evolving), any technologies to be used are well-known to both the developer and the client, and the product is relatively small, of manageable size, or not complicated. Conversely, Agile ISD is best when the requirements are ill-defined such as when the desired training outcome is still very broad and the target audience undefined or their receptiveness to certain technologies is unknown. Agile ISD is also good when the content is still under development or evolving. Agile ISD affords the testing of new, high-end technologies so challenges and obstacles can be identified before an entire product is based on a technology assumption that is not compatible with the end-user requirements. Agile ISD is also useful when speed and collaboration is a necessity such as developing a multi-pronged training solution for a safety concern when developing the training, creating a 'marketing' program to promote the required behavioral change, and developing on-the-job aids to support that required behavioral change are being accomplished simultaneously. Agile ISD is a flexible model that requires continuous stakeholder interaction

throughout the project. Each end-of-sprint review demands appropriate stakeholders attend, review, and discuss the implications of the progress made in the current sprint. Not all stakeholders need attend but those relevant to the current iterative product need to attend. Unfortunately, stakeholders (clients, users, subject matter experts, etc.) may not want to invest in the time requirement or may not be interested in changing from the familiar ADDIE Model of production. Furthermore, use of the Agile ISD Model does not ensure prevention of re-work at the end of a project even though pieces of the product have been reviewed throughout the project as we will see in the case study later in the paper.

AGILE ISD AS APPLIED TO PROJECT

As stated previously, the project was to develop simulation-based training on doffing PPE. This is a two-person task-based activity. The Caregiver is the person who finished care with the patient and the Observer/Buddy directs the Caregiver on what doffing step to do in accordance with a checklist. The learning objectives were developed as part of the initial prototype effort. The objectives included identifying doffing equipment, locating the doffing area, following the doffing checklist, performing the doffing steps, recognizing when contamination has occurred, and recalling the ramifications of improper doffing. With this information, the second prototype effort began by developing the Agile ISD Management Plan.

The formality of creating the Management Plan was beneficial for this project so everyone had a common understanding of the project. Anticipating that we would be re-scoping the project based on encountered challenges and priorities, the Goal Statement helped keep the team on track. Projects developed with an agile process do not require detailed, formal documentation since the iterative process allows the details to organically establish with findings as shared during sprint reviews. The goal for this project was to *provide a viable prototype that the client can showcase to upper management and the healthcare community on the use of 3d simulations for training*. The significant element in this particular Goal Statement was the notion of a *viable prototype* which was understood by the team to mean the training simulation would be functional. This meant the focus would stay on functionality which would require quickly overcoming obstacles to ensure functionality would be achieved and not fixating on visual fidelity. This further implied that the training would focus on practice and would not address providing underlying basic knowledge unless time, budget, and priorities allowed. The Goal Statement was further clarified by the Vision Statement which set expectations:

> *The final product will support two players to interactively practice doffing PPE; one player as the Caregiver and one player as the Buddy, each player using their own computer but collocated. Players will be oriented to the gameplay and gameplay goals to include what constitutes correct actions. The Buddy will use a doffing checklist similar to that used in the real world to guide gameplay, i.e., direct the Caregiver. Random injects representing distractors/incorrect actions will be threaded throughout the gameplay through the use of knowledge checks to provide opportunity for players to experience and assess incorrect actions. Players receive feedback throughout gameplay. Refer to Proposal for other technical assumptions.*

The Management Plan also identified stakeholder roles and responsibilities. Identification of roles and responsibilities seems to be more important in the agile process than the ADDIE process. Agile is a very collaborative process with many opinions and contributing thoughts shared throughout the project, and especially as should be expected, during the sprint review meetings. Understanding roles helped identify who the decision makers are for any one discussion; the ISD would have final decision on instructional integrity, the user experience/user interface (UX/UI) would have final decision on styles, interactions, and presentations; and the programmer would have final decision on how to implement functionality. However, all three actively participate in discussions because their decisions are not truly independent of each other. These discussions can occur on a daily basis since the Agile ISD process calls for daily 15-minute standup meetings where team members share progress, concerns, and obstacles impeding progress. In comparison, the ADDIE does not require this level of collaboration. The ISD determines the direction during the design and storyboard stages although it is not uncommon for the ISD to contact the UX/UI person or the programmer for ideas or validation of ideas suggested by the ISD. However, the ISD generally makes the final decision based on information shared. So having these roles defined in the Agile ISD process helped manage discussions since all team members knew their role and responsibilities.

At the initial backlog development meeting, the team identified over 60 Stakeholder Stories. Some stories followed the typical format of role, need, and benefit although this particular story did not provide the benefit (while we profess a process, we are not perfect in its application): *As a learner, I want to see a Welcome/Introduction section upon*

starting the simulation. Other stories were considered Technical Stories (TS): *Create components for PPE*. Some stories were already known to be low priority but included anyways to maintain a list desired additions for future reference: *As a user, I want to be able to select from multiple avatars*. The current prototype would only offer to select a male or female avatar in order to stay focused on functionality. These stories were numbered and placed into the backlog using the software JIRA by Atlassian. Other software that tracks and manages stories is available on the market or you could use an excel spreadsheet although this would be more cumbersome.

The stories were prioritized for execution based on importance or if pre-requisite to another story. The stories at the top are in a preferred sequence while less time is spent sequencing lower priority stories as project priorities could change or the project takes a new direction making some stories moot. At the start of a sprint, top priority Stakeholder Stories are selected by each team member in accordance with their role to be accomplished during the sprint. Table 3 provides the Stakeholder Stories selected for this project's first sprint with an explanation of the required subtasks per story and the person responsible for that story.

Table 3. Sprint 1 Stakeholder Stories with Required Subtasks

#	Stakeholder Story	Required Subtasks (with Role)
1	TS: Create components for PPE	Create boot covers, outer gloves, inner gloves, gown, face shield, hood, N95 mask (UX/UI)
6	As a learner I want to see a Welcome / Intro section upon starting the simulation	Create content for Welcome screen, purpose screen, and gameplay intro (ISD)
29	TS: Create user Avatars	Create male and female avatar (UX/UI)
39	TS: Create Supporting Documents	Create assets list, overarching Design Document, and interaction table (ISD)
47	As a Learner I want to experience a PPE Doffing simulation in a multiplayer environment so that I can practice the procedures	Create menu, establish hosting/joining network, load players into the environment (Programmer)

For those familiar with the ADDIE process, it may be surprising to see that a Design Document has not yet been created but the programmer and UX/UI team members are starting to develop the framework and the assets in the first sprint. In this project, it was already established it would be a two-player environment and therefore the programmer could set up that framework. It was also known that the players would be wearing PPE and the Caregiver would be removing the PPE component-by-component. Therefore, the ISD provided imagery in a PowerPoint presentation to illustrate each PPE component so asset development could begin. This relatively early start on development activities may not be applicable for all projects. In this project, the initial prototype phase allowed the ISD to collect background information, create learning objectives, and confirm the use of Unity 3d as the platform, i.e., analysis type activities. But having said that, it's also clear that creating PPE components and a two-player framework would be required effort at minimal risk.

By the end of Sprint 3, the team had developed the training environment using a floor plan layout provided by the ISD based on an existing hospital layout. A video of the simulation environment was made from an avatar's first person view and provided to BEHPP staff for review of technical accuracy as part of the sprint review process. Images from that video are provided in Figure 4 for context. Two lengthy discussions ensued based on this preliminary design. In an agile process, discussions are expected as the design is not perfect but intentionally intended to spawn informative discussions. In the first discussion, the patient room was considered austere, void of common patient room equipment such as a sharps-disposal container, call buttons, and blood pressure equipment. Additionally, flowers and cloth-based furniture would not be in a room intended for a HIDs patient. Given our Goal Statement to produce a viable prototype, the team would add patient room assets if already owned or readily available without modification; other assets would not be pursued at this time so focus could remain on developing a functioning viable prototype. However, these items are noted and are maintained with the project documentation on our collaborative wiki. The chairs were switched to

plastic and flowers removed because incorrect items would be a distraction to learners who would recognize them as inappropriate. Incorrect items also reduce the creditability of the training product by the learner.

Patient Room View from Patient Room to Doffing Area Doffing Area

Figure 4. Preliminary Simulation Area for Doffing PPE

In the second discussion with BEHPP staff, the walking distance from the patient room to the doffing area was considered too far. Staff indicated that one would exit directly into the doffing area to minimize surface area contamination. The simulation environment would need to be corrected. Again, in an agile process of iterative, collaborative development this is not unexpected. The preliminary provided the context to have a deeper discussion. Discussions addressed the number and location of doors in the doffing area, the contents of the room, the number of chairs, and the addition of wall posters that illustrate the doffing steps. A floor plan was created with proposed corrections in a PowerPoint presentation which went back and forth between the developer team and BEHPP staff until all agreed to the floor plan. The accepted floor plan with actual environment is shown in Figure 5.

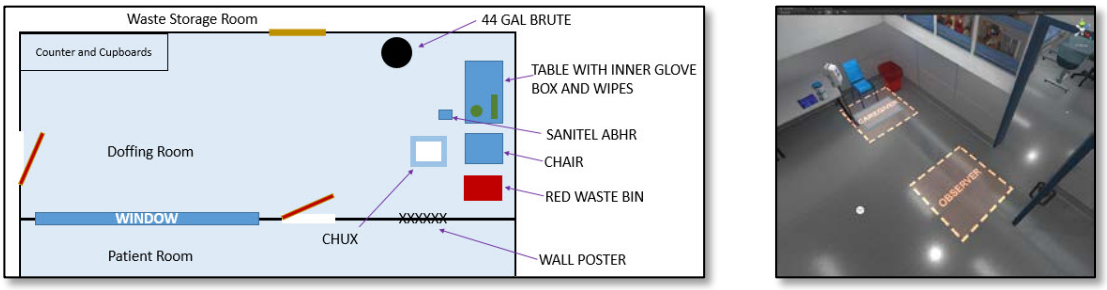

Figure 5. Accepted Floor Plan and Subsequent Implementation

However, during usability testing, staff noted that the new layout would still not work because the walking distance was still too great for potential contamination and the Observer location was co-mingled with the Caregiver's footsteps. Even though the Observer is fully outfitted in PPE, risk of contamination was to be avoided. The floor plan would have to be changed again. The existing asset layout would be rotated 90 degrees clockwise such that the chair would be on the immediate right of the door exiting the patient room. Even though all were in agreement of the layout in plan-view through our iterative process, it wasn't until staff was actually in the environment did it become apparent that the layout could be improved from a safety perspective. The use of a 3d environment seemed to introduce a complexity in the review process. Two-dimensional imagery did not convey the actual feel of how the user would feel or move in the environment. It wasn't until the reviewer was in the environment did the sensation of distance become apparent. Our iterative process, in retrospect and in the future, should include the user actually interacting with the environment sooner. The leap from flat 2d images to 3d is too far.

During the last sprints, the After Action Review (AAR) for the training product was created when its Stakeholder Story became a priority. The AAR would provide performance feedback after training completion. We had decided near the start of the project what we would track for performance in accordance with the learning objectives so the programmers could flag and track those interactions. But the AAR wasn't designed until one month prior to beta delivery of the prototype because it was considered a capability that could be dropped if time and budget did not allow. So during the last sprints, the feedback had to be created, the look had to be designed and implemented, and any user interactions had to be programmed. Had this project been developed following the ADDIE process, the AAR look/feel and feedback would have been designed before the simulation was even created. If the AAR was not going to be included due to unforeseen problems that occurred and the client did not have additional funds, the effort spent

designing the AAR with feedback and proposed look/feel would have been lost effort. In contrast, the agile process did not require this effort to design and develop the AAR until it was time to create it. One could argue that in the agile process, the programming for performance tracking was lost effort if not fully realized. Interestingly, we did have that conversation at the project start because we knew we may have to prioritize what gets accomplished. It was decided that programming and not using was cost effective over going back and adding the tracking later.

Fortunately, our contract environment afforded prioritization of the prototype features which is not always the case. In many cases, a request for proposal is let with a Statement of Work (SOW) that outlines the training need with explicit requirements, especially when the training need is well-defined. The contractor responds with their solution and the cost for that solution. The contractor will generally list their assumptions, so the potential client understands exactly what is included in the solution for the quoted cost. The solution with the assumptions define what will be accomplished; deviations will require a contract modification. For example, a client may agree to the use of a synthesized voice but upon hearing the synthesized voice, decide that a live voice actor is preferred. This has cost and schedule implications which would require a contract modification to accommodate that change. Depending on the agency, contract modifications can take weeks to enact. During this time while the contract modification is being executed, the contractor cannot progress on those elements being modified. Progress halts.

In other situations, the SOW may have flexible or general requirements when the needed training support services don't initially have well-defined product requirements, or the estimated size and labor requirements of the project is unknown or if working in a research environment where requirements evolve or emerge based on findings. In some flexible contract environments, entire training products can be removed or completely changed such as from instructor-led to online computer-based training. No contract modification would be required. One can understand that an agile process that readily accommodates change would work well in a flexible contract environment that also readily accommodates change.

In our project, the advantage of a flexible contract was exemplified in the use of embedded video clips. At the end of each doffing step, a video clip of live actors performing the step would be shown in the form of a knowledge check. The user would be presented with a randomly selected correct or incorrect video clip. The user would assess if the step was performed accurately. This was included as an instructional strategy to help the learner recognize real-life accuracy since the 3d environment was focused on the procedural steps and did not offer high visual fidelity. At the start of the project, we knew there were existing video clips of the steps online and we would use those clips. We also knew that incorrect video clips would be needed and those weren't available online but there were some backup thoughts. We could switch out the knowledge check as simply an example of how to accurately perform the step in real-life. We also had some thoughts for creating incorrect clips. We could do the simple steps such as glove removal with little effort and only have correct video clips for the other steps. It was after all a prototype. But how we would handle this challenge would only be addressed when that Stakeholder Story rose in priority. This was an agile process.

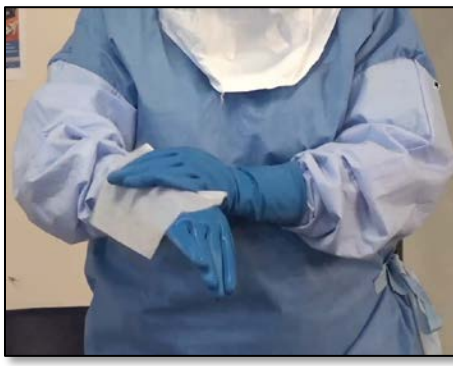

Figure 6. Knowledge Check Video Inserts

When the Stakeholder Story for video clip knowledge checks did become a priority, we discovered that some clips no longer aligned perfectly with current guidelines. We would not have correct video clips for all steps and would not have some or all incorrect clips. The options were to remove them altogether or develop our own. The client offered to record the video clips with their in-house professional staff while the contractor would plan, provide the actor, and edit the clips for inclusion. The use of the video clips was deemed important instructional strategy for the learner to assess performance in real-life not just with avatars which lacked the nuance fidelity that only live video could capture, i.e., understand what real-life errors in the doffing procedures could look like. This addition of planning, recording, and editing of video clips would have certainly caused a contract modification if requirements had been explicitly defined in the SOW.

Our flexible contract environment supported these types of changes without the requirement for contract modifications. Two overarching cost contract types are commonly used to manage consulting projects: time and materials (T&M) and firm fixed price (FFP). T&M allows the contractor to bill by the hour and allows the client to more readily make changes along the way. FFP is generally associated with a well-defined deliverable that the client knows will be produced for the agreed upon cost. However, in the case of research, an FFP could be written more generally but then may require more client-contractor interaction (which agile does) to keep everyone abreast decisions related to product development. In our case, the specific definition of the prototype was anticipated to evolve with the agile process, but also by the end of the contract, a functioning viable prototype that met the overarching vision statement was required even if the specifics were unknown. Regardless if a T&M or FFP contract is issued for an agile development process, strong management with a strong customer-client relationship is required to ensure everyone agrees to the direction of the product during the agile process, i.e., during sprint reviews. Creative, goal-oriented thinking combined with advice from a contracting officer can help formulate a flexible contract environment that will best support the client's and contractor's needs.

CONCLUSIONS

The Agile ISD process is a relatively new instructional design and development process model in comparison to the traditional, waterfall-based ADDIE approach. The Agile ISD process can reduce risk, increase efficiencies, and inherently instill collaboration. Consistently following the sprint process (plan, execute, review, reflect) means training products are not designed and developed in a vacuum by discipline (e.g., ISD followed by programmer) but rather synergistically while collectively adhering to instructional principles, efficient programming practices, and UX/UI best practices.

The Agile ISD process model was a better development process model for this project in comparison to the ADDIE model. The agile process allowed prioritizing efforts so unforeseen issues could be readily addressed such as creating the new video clips but still producing a viable prototype. While the iterative review process through sprints is intended to reduce risk, it is not a perfect risk mitigation approach. In this project, pictures and video used during our iterative process were not enough for stakeholder reviewers to fully comprehend and assess the accuracy of the learning environment. They needed the firsthand immersive experience to accurately assess the learning environment. Getting our stakeholders into the 3d environment sooner, rather than later will become a priority in future projects. Additionally, the Agile ISD process provides flexibility in the final product and therefore aligns well with a flexible cost contract environment that can accommodate change.

The Agile ISD is not intended for all project but works well when requirements are ill-defined, the content is still evolving, the risks are high, the technology is new or high-end, and when speed and collaboration are necessary. Transitioning from the traditional ADDIE approach to an Agile ISD approach will take time and energy but we have found it inherently collaborative and cost effective, especially when developing 3d-based learning environments that tend to have more complex design considerations than two-dimensional computer-based training products. The simplest way to get started may be to implement the sprint process in your standard approach whereby accomplishments are reviewed with the team and the client or stakeholders on a regular basis such as bi-weekly. Additionally, while the instructional designer is still creating the content and design, consider identifying images, assets, interfaces, interactions, and code that are seemingly well-defined that can be created for sprint review. Consider exploring this process development model on your next complex, high risk, new technology-based training product to determine if it's appropriate for your project needs.

ACKNOWLEDGEMENTS

This work was supported by the National Institutes of Health, National Library of Medicine, as well as by the Bethesda Hospitals' Emergency Preparedness Partnership.

References

Ackles, B. O. (2018). Agile development instructional framework (ADIF): A new strategy for student-centered music education. *Choral Journal*, *59*(2), 22-35.

Allen, M. (2012). *Leaving ADDIE for SAM. An Agile Model for Developing the Best Learning Experiences*. American Society for Training and Development. Versa Press: Illinois.

Highsmith, J. (2001). History: The Agile Manifesto. Retrieved from http://agilemanifesto.org/history.html

Piskurich, G. M. (2006). *Rapid Instructional Design*. John Wiley and Sons: San Francisco.