

Spring 5-19-2017

# Applying modern software development methodologies to eLearning

Martin Tyszka

La Salle University, [tyszkam1@student.lasalle.edu](mailto:tyszkam1@student.lasalle.edu)

Follow this and additional works at: <http://digitalcommons.lasalle.edu/mathcompcapstones>



Part of the [Adult and Continuing Education Commons](#), [Online and Distance Education Commons](#), and the [Software Engineering Commons](#)

---

## Recommended Citation

Tyszka, Martin, "Applying modern software development methodologies to eLearning" (2017). *Mathematics and Computer Science Capstones*. 33.

<http://digitalcommons.lasalle.edu/mathcompcapstones/33>

This Thesis is brought to you for free and open access by the Mathematics and Computer Science, Department of at La Salle University Digital Commons. It has been accepted for inclusion in Mathematics and Computer Science Capstones by an authorized administrator of La Salle University Digital Commons. For more information, please contact [careyc@lasalle.edu](mailto:careyc@lasalle.edu).

Applying modern software development methodologies to eLearning

Martin A. Tyszka

### Abstract

Modern rapid application and agile development methodologies have been key to getting software to market quickly and ensuring that it meets customer needs. The improvements brought about by these methodologies are not limited to the product, but also improve the development process through the feedback approach integral to the methodology. eLearning, the content and technology that enables learning via the Internet, has been adopted by businesses based on its advantages in lower costs, increased productivity, and flexibility when compared to traditional training approaches. This has resulted in a growth in this area. eLearning has been developed following traditional Instructional Design methods. This project sets out to examine eLearning content development and undertake the development of an eLearning lesson using a modern development methodology to comment and reflect on the practical use of this approach.

This project identifies that expected benefits of early start, earlier first product release, responsiveness to change and issues along with the advantages of process improvement and reflective examination were seen. This project also calls to attention the importance of the design element of Instructional Design when using the Scrum approach. This project concludes that the Scrum approach is a valid and valuable method to produce an eLearning lesson.

## Applying modern software development methodologies to eLearning

*Guidance to the reader:* To assist the reader the explanation of certain terms is included in the glossary found at the end of this paper. Terms shown in *italics* in the body of the text are detailed in the glossary. The first occurrence of each term is shown in this fashion.

Modern rapid application and agile development methodologies have been key to getting software to market quickly and ensuring that it meets customer needs. These methodologies have helped to enable and drive the pace of technological change. With the cost of IT decreasing and the capabilities increasing, these methodologies have supported the evolution of software and systems. The improvements brought about by these methodologies are not limited to the software produced, but also improve the process to develop software through the feedback approach integral to the methodology.

Performance and organizational effectiveness are key to ensuring that organizations meet their strategic objectives. In addition to the improvements brought about directly through technology resources, organizations also focus on another of their resources for process and performance improvements - their human resources. Development of these human resources occurs through training. The cost, flexibility, and scalability advantages offered by eLearning solutions has resulted in continued growth and adoption of this training solution.

eLearning brings technological innovation with the development of content in many forms. The products of both software development and eLearning development are unique technology based products, where each application or training course delivers a specific solution. New releases and products in software and eLearning domains alike may be based on the same framework as their predecessors, but deliverables are unique based on creative design. For

effective creation of these products, the development process is not one of a production line, but a continual refinement based on prior experience. Software development and eLearning development both feature continuous evolution. In eLearning, this continual content creation is used to drive learner engagement and implements a continuous learning approach. Parallels can be seen in the way that eLearning is developed and the methods used for software development.

This project sets out to undertake the development of an eLearning lesson using a modern development methodology to comment and reflect on the practical use of this approach. This project will determine whether the benefits expected from the use of such an approach are seen when applying it to eLearning. This project will also identify any shortcomings encountered.

### **Setting the stage**

#### **What is eLearning?**

There is no cut and dried definition of eLearning. Hartley (2001) defines eLearning as learning enabled by the Internet, Intranets and other electronic networks as well as the development delivery and evaluation of content provided to learners through these networks. eLearning is also interpreted as learning utilizing electronic technologies to access educational curriculum outside of a traditional classroom. (elearningnc.gov).

Computer enabled learning originated with Computer Based Training (CBT) in the mid-1970s, when training was delivered using early mini-computers and workstations. The advent of Personal Computing eventually gave rise to the distribution of CBT via CD-ROM. From here CBT evolved into eLearning with the adoption of delivery over networks, with the term recognized as being introduced at a CBT Systems seminar in 1999 (Hubbard, 2013). eLearning maintains a link to distance learning (Lewis, 2012) and correspondence courses. From the

origins of distance learning for remote university studies (Anglia Ruskin University), the term eLearning has come to broadly cover training and education delivered over the Internet.

While the term eLearning may lack a precise formal definition or well defined boundaries, a flavor from eLearning can be taken from the earlier term “multi-media” that was previously used to describe online learning (Lee, 2004). The breadth of the term has not only been used to cover both synchronous and asynchronous delivery (Hrastinski, 2008), but also to encompass a variety of aspects of the development and delivery of online training. These can be as varied as the development of standalone learning objects or the development of software to support programmed instruction, where simulation-like software is used to promote both assessment and learning (Bettio, 2013).

The adoption of technology globally along with the introduction of new technologies has expanded the reach of eLearning. Businesses have seen the potential in eLearning, with a 2014 Training Industry report showing that 28.5% of all corporate training hours were delivered via online or computer based technologies (Pappas, 2015). The same report identifies that this trend is growing. Businesses are selecting eLearning based on lower costs, increased productivity, and flexibility (Kuhlmann, 2010) when compared to traditional training approaches.

### **How is eLearning developed?**

To achieve quality instruction, the creation of eLearning like any other form of instruction requires learning and instructional theory to be considered (The University of Michigan, 1996). Sarasa-Cabezuelo and Sierra-Rodriguez of the Complutense University of Madrid (Universidad Complutense de Madrid), identify that eLearning projects are characterized by a great complexity in functionality and include the participation of many different actors.

These factors along with the requirements for strong interoperability and reusability serve to highlight the complexity of producing eLearning.

A complex problem of this nature benefits from a systematic approach being taken. One approach, Instructional Systems Design, is rooted in system theory (Hodell,2016), wherein a complex system is broken down into component parts, while recognizing the system as a whole is greater than the sum of those parts. Instructional Systems Design applies to the development of any training, not specifically to that delivered electronically. For eLearning, the need for a structured approach for training is reinforced due to the additional complexity introduced by the delivery mechanism (Hodell,2016),

Viewing the development of eLearning as a system results in a practice that can be followed routinely, a mechanism to aid in understanding the complexity and a way for all involved to understand the progress of activities. It is this process which provides the framework to develop training in a consistent and reliable fashion. (Clark, 2015)

Just as the supporting technology and eLearning itself have advanced, so have the models and processes that drive their development. (Reiser, 2012). There are a number of Instructional Design models. İşman et al. (2005) identified thirteen instructional design models and noted that hybrid approaches are possible as elements from multiple models can be combined. These Instructional systems design models aim to cover the elements that take training programs from design through to completed development. One of the widely accepted Instructional Systems Design model is the ADDIE (Analyze, Design, Develop, Implement, Evaluate) model, developed for the US Army by the Centre for Educational Technology at Florida State University in 1975 (Forest, 2014).

The ADDIE model is comprised of five phases. Analysis, Design, Develop, Implement and Evaluate. In summarizing each of these five phases, research of the situation the training is to take place is performed in the Analysis. This phase seeks to answer the questions of who the audience is and what is the desired outcome of training. This phase also serves to identify constraints. The Design phase covers aspects that include the design of the overall approach through instructional strategy, defines the user experience, and the visual design. The Development phase focuses on content creation and assembly. Following this the Implementation phase addresses the process for delivery to the learner, combines all material and prepares for deployment. This phase is not limited to the perspective of the training developer, but also covers aspects from the student perspective such as student registration. The purpose of the Evaluation phase is to ensure that all stated goals of the learning process will meet the specified needs. This phase covers evaluation in two forms, the formative evaluation for each stage of the ADDIE process and the summative evaluation to gain feedback from the users. The phases of the ADDIE model are not specifically tailored to eLearning, but applicable to any form of education. In following the phases of the ADDIE model sequentially it can be regarded as a linear model linear one, where decisions need to be made in the early phases, and the outcome are relied on by later phases. This form of model is a predictive one, in that all the activities to be done need to be predicted in advance, at the start of the project and then executed.

The thirteen models identified by İşman (2005) can be seen as variations on the ADDIE model. The majority of these maintain similar linear approaches. Some attempt to simplify ADDIE into fewer phases; the Hannafin-Peck Model. Some introduce parallel linear streams (Smith, 2003); The Knirk and Gustafson Design model. Some are linear but introduce a final stage of analysis of feedback to provide refinement into the specification stages (Forest, 2016);

the Gerlach and Ely model. Certain models do move further away from the linear approach. The Kemp model presents a more holistic view and removes some linear dependencies, however leaves evaluation and improvement to the final stages (de-research.com). Tripp and Bichelmeyer introduce some overlapping activities in their Rapid Prototyping (Dawson) approach. While establishing variations in approach, each of these models can be regarded as predictive models, and subject to the challenges determining the design upfront.

The ADDIE model, while not strictly linear with its inclusion of evaluation applied to all stages, is generally treated in phases. Other models including the ASSURE (Analyze learners, State objectives, Select methods, media, and materials, Utilize media & materials, Require learner participation, Evaluate and revise) model (Ibrahim, 2015) and ILDF, the Instructional Learning Design Framework (Bruce, 2011) have elements common to ADDIE, with refinement in process to provide additional focus on certain facets. Models like the Dick and Carey model are based on the Systems Approach Model and further refine the Instruction System Design approach through the introduction of iterative aspects (Instructional Design Center, 2017). The Dick and Carey model (Forest, 2015) contains the same components as the ADDIE model, dividing Analysis, Design and Development into further steps. Under this model the step of revising instruction impacts all steps of the otherwise linear model. The benefit of the iterative aspect of this model therefore seems limited.

Russel (2015) identifies that changes, that occur naturally during development projects, do not fit in well with the top down methodologies that are formed around a prescribed set of repeatable steps. The models identified in the previous two paragraphs all follow a top-down, predictive methodology. Rosenberg (2004) raises criticism of the ADDIE model's linear nature,

and that in the real world the practice must be more iterative and heuristic, diverging from strict adherence to the model. The proposed solution is therefore to consider an Agile approach.

### **Software development methodologies and Instructional Design models**

The process of developing software can be seen to have many attributes in common with the process of developing eLearning training courses. The software development process executed in the real world follows a structure put forward by a software development methodology, where eLearning development follows an instructional systems design model such as the ADDIE or Dick and Carey models. Any software development project needs activities related to requirements, designing, building, testing and delivery to be addressed. A software development methodology describes the framework of how this is achieved.

### **The evolution of software development methodologies**

The Waterfall methodology is the traditional software design methodology. It takes its name from the linear approach that it adopts, where one element cascades down to the next. In following this methodology, a software development project undertakes each phase of the needed planning, design, building, testing, and delivery steps in turn (Clarke, 2016). This methodology relies on thorough up front requirements gathering, followed by analysis and then design prior to the writing of any code. Waterfall is linear in nature, and does not include the opportunity for feedback or iteration within its approach.

Under a Waterfall approach all requirements are specified in the earliest stage of the project. The creation of requirements documentation at that time defines the scope of the project. To cover all requirements at this early stage, both the breadth in requirements and the detail of

the requirements must be considered and formalized. This upfront development of requirements takes considerable effort, particularly since it must be relied upon to form the stable foundation of the project.

Waterfall is classified as a predictive model in that everything that needs to be done needs to be predicted in advance, at the start of the project and then executed. This prediction relies on experience (Stephens, 2015). In setting the direction from the outset of the project, with big design up front, changes after the initial phase may effectively mean starting the whole process over again. The use of a predictive model in an environment with incomplete, unclear or changing requirements will cause these issues to surface.

Royce (1998) identifies a number symptoms shown in troubled projects that follow the Waterfall approach. This methodology leads to the results of decisions made early in the process only being shown towards the end of development. With the focus of early activities being heavily on documentation, unforeseen risks are only discovered and resolved late in the project leading to budgets and time schedules being stretched. The resolution of issues late in the process without the opportunity for redesign addressed at the eleventh hour leads to a fragile end result, as issues are resolved under pressure of delivery.

A further symptom of using a process such as Waterfall that focuses on the documentation created is that it can result in an exaggerated value being placed on those documents rather than the deliverable product. The emphasis on the creation of documentation may outweigh the activities of later phases, after all Waterfall relies on the design being clear and correct from the outset. Insufficient focus on producing the tangible product may result.

Royce highlights a further issue with decomposition of software into functions driven by requirements. The structure developed is very different from a decomposition based on object-

oriented design and the use of existing components (Royce, 2000). Breaking down a problem in this way, while certainly satisfying requirements may limit component reuse as components are considered individually, and limits them being considered in context and as part of the bigger system. Addressing components individually is likely to increase timescale when compared to handling them with a more holistic approach.

This raises the question whether the symptoms Royce identifies are specific to the Waterfall approach, a linear process for software development or more generally seen in any linear process. While it may be seen as inflexible in an environment subject to change, where requirements are well defined and do not change or the project a small one, the Waterfall method achieves its goals. At the end of the day project success is achieving the goal on time and to budget.

New methods have been created to address the problems observed with Waterfall. Many of these methods aim to address the same type of problem, but undertake them in different ways. Some methods evolved from the foundations of Waterfall, among these the Sashimi waterfall method (Stephens, 2015) which overlaps the phases of Waterfall allowing a later phase to modify the preceding phase. Similarly, the Waterfall with Feedback model follows the phases of Waterfall and adds feedback to previous phases. Waterfall has also been used incrementally to achieve subsequent releases and thereby bringing in a somewhat adaptive approach. However these all demonstrate the characteristics of Waterfall and still rely on the complete design upfront at the beginning of each iteration and do little for flexibility. Other methods developed target individual issues; risk management is addressed in the Spiral and Universal Process approaches, while test-driven development, where test cases are developed before any production of code

(Dogša, 2011) and formal methods, an approach based on building a rigorous mathematical model (Collins, 1998), both target code quality.

Stephens (2015) cites Karl Scotland in introducing the definition of fidelity, the completeness of a feature. In doing so he identifies a classification of methods. Predictive methods result in all features being delivered at the same time with full fidelity. Iterative methods provide all features at lower levels of fidelity and later iterations increase the fidelity of all features. Incremental methods provide a minimal number of features but those features are complete at the highest fidelity. Later increments provide further features also at the highest level of fidelity. Meanwhile Agile methods start with the earliest delivery providing minimal features at low levels of feature completion, and then increase both fidelity of existing features and increments with additional features. This last approach offers the earliest initial release. This concept of earliest initial release goes beyond the goal of on time and on budget delivery of a project by making useful software available as soon as possible. For businesses, this can facilitate getting to market earliest, a competitive advantage and shortens the period to realize some return on investment.

This concept is addressed by several methodologies that fall under the rapid application development banner. The heart of rapid application development lies in taking the iterative approach and shortening the timeframe between iterations and in so doing increase their frequency. All elements of the project are treated to the same iterative approach from requirements gathering onwards.

The search for improvement in software development gave rise to the Agile principles that have further shaped software development methodologies. The principles seek to embrace the real-world issue of change during development, while balancing continual development,

collaboration, simplicity, and further adjustment of the process. A number of methodologies are based around these principles with the two most broadly used being eXtreme Programming, extreme in its use of continual code review through pair programming and the Scrum methodology (Mauer, 2006).

Scrum is a set of principles and practices that are used to develop innovative products and services. Scrum delivers products through continual development in short time boxed cycles of fixed length termed sprints. Each time boxed sprint sets the contents to be developed from a prioritized product backlog. This allows both iterative and incremental development of the product, both adding features and refining or improving existing in each cycle. This continual development brings about continual improvement to the product. This differs from the traditional approach that aims to deliver a single complete and finished product.

This enhancement is not limited to direct development of the product, but this process also sees the introduction of dynamic process improvement (Rubin, 2013). The Scrum Alliance (ScrumAlliance.org) identifies that Scrum originally was formalized for software development projects, but it works well for other activities where the scope of the work is to produce complex and innovative products.

### **The evolution of Instructional Design Models**

In examining the weaknesses of the ADDIE model Culatta (2013) includes the following; requirement for unrealistically comprehensive up-front analysis, detailed processes become so set that creativity becomes a nuisance, and no accommodation for dealing with faults or good ideas throughout the process. From this short list parallels are visible with the issues that could be identified with the Waterfall model that are addressed by Agile methodologies.

Russell (2015) identifies that the ADDIE methodology parallels the top down or Waterfall methodologies highlighting the issue of how change is handled when following the methodology. As details are uncovered or solidified during the implementation, some will impact the validity of the initial design, causing a change to this design. It is these changes that a linear model does not accommodate, effectively causing the practitioner to start over. The issues are mitigated by newer methodologies either by incorporating more stakeholder involvement up front, to reduce the risk of changes later, or by building on an iterative approach.

Russel cautions against the use of prototypes in an iterative model in that it can introduce death spirals, the practical struggle being a successive set of prototypes with seemingly no end. Russel identifies performance improvement in the production of eLearning as a driver for the use of new methodologies, identifying the SAM (Successive Approximation Model) and Agile methodologies. Allen (American Society for Training and D., 2014) as the creator of SAM, identifies the influencing factors in the movement to a new model to be the desire to develop training faster, and using teams with less accumulated knowledge of instructional design and human learning. A new model therefore needs to be practical, simpler, faster, more collaborative model than ADDIE while at the same time being one that encourages creativity. SAM holds many similarities to an Agile process with its iterative approach that does not require big up front analysis design. Where SAM relies on the development of prototypes in three cycles, Scrum aims to deliver a viable product in each sprint and the number of sprints defined by the needs of the project.

### **An Agile approach for eLearning**

Bettio et al. (2012) undertook a practical development of digital media learning objects for face-to-face undergraduate courses using Scrum. The use of this agile method addressed the ability to deliver learning objects in a short period of time and subject to the modifications. Scrum also provided a mechanism to enable joint and articulated work involving subject matter experts (professors) and technical workers. Each Scrum sprint resulted in a number of learning object projects being completed. The use of Scrum in this instance focused on the management of the development process, and refinement of the production process rather than the iterative and incremental development of a single product. The use of Scrum was identified as successful for the purpose, with further refinement for further sprints suggested as part of the *sprint retrospective*.

Training is seen as a way to solve problems, a remedy to issues and a way to affect behavior. Training is even highlighted in high profile incidents including airline crashes (ABC, 2015). Problems evolve over time and new information about issues comes to light. Training therefore needs to be responsive to change in the situation or environment that training is conducted in. Training development, including eLearning could benefit from the advantages that the Scrum approach offers. To gain insight and a practical understanding of Scrum as applied to eLearning development, the project associated to this paper will follow the Scrum approach to produce a functional example of an eLearning product.

Among the areas in which these benefits could be realized are the drivers for continuous learning approach, the ongoing development of skills, abilities and knowledge (Deloitte, 2017). The continuous learning approach requiring the continuous production of learning materials.

The continual improvement in the product that results from Scrum seems to be a cohesive fit for this challenge. eLearning as a product in a commercial setting can benefit from Scrum, in focusing on iterative product development it can deliver a product earlier, a competitive advantage in getting the product to market sooner. Scrum also provides further benefit through its adoption of change, in areas such as technology where the subject matter is changing, the training remains in a state of change.

This project sets out to verify whether Scrum is a workable approach for eLearning. It will also highlight the advantages uncovered and issues found when following this approach. Finally, it will suggest how the use of the Scrum approach can be further adopted for use in the development of eLearning.

### **Approach to trialing Scrum for eLearning**

To gather the experiences in the practical use of Scrum in this area the scope of eLearning development must be defined. Where the term eLearning covers a wide variety of ways in which training is delivered, this project focuses on the production of content for asynchronous self-paced classes.

The strategy adopted by this project was to undertake the development of an eLearning lesson. The chosen topic for the training product being developed was itself Scrum for eLearning and envisaged for a target audience of eLearning content developers adopting the Scrum approach. The approach taken was to cover the development of this lesson in two sprints. This would serve to exercise the Scrum approach and allow for review of the developed product and retrospective inspection of the Scrum approach as well as gain exposure to the impact of change in the subsequent sprint.

This project was constrained by being performed solo. A number of areas where the Scrum approach is focused on the development being performed by a team are identified below, where the process was limited for this project by this constraint.

### **eLearning authoring**

eLearning content can take many forms with a multitude of different delivery mechanisms to deploy content in a wide variety of formats from static web-pages to videos to highly interactive content. The development of this content is supported by a great range of tools and choices in delivery mechanism and format, which will determine how the content is developed and may shape how the implementation is undertaken.

To remove the effects of specific implementation needs on the project, a simulation of the authoring process would be undertaken. In order to achieve this a familiar and generally used tool that could be used to create content in a simple fashion was needed. For this Microsoft PowerPoint was chosen as the tool to author and deliver the eLearning lessons. This allowed focus to be retained on the approach, and avoid the specifics of particular choices of technologies and toolsets.

### **The development process**

Initiation of any project to develop a product or service is driven by a need. For eLearning, this need could be expressed as the student's desire to learn or the training providers drive to inform, instruct, transfer knowledge or affect the student's behavior. No matter how the eLearning development process is undertaken, be it by following a traditional linear approach, or by following an agile approach like Scrum, some form of requirements must be detailed.

While the requirements definition under the traditional approach can be seen as both wide and deep, the Scrum approach does not necessitate the expansion of requirement to these levels of detail, regarding them at a high level and focused (Thomas, 2008). This is achieved using a product backlog, a list of the desired features for the product. The product backlog includes product backlog items of different types, the size or level of detail of the items is not specified. In following the Scrum approach the product backlog will be treated to prioritize and order the list.

Prior to the first sprint an initial product backlog was assembled. In a software development context, the items in a Scrum product backlog may be identified by type specific to that domain, e.g. Feature, Change or Bug fix. Product backlog items often take the form of user stories, a structured phrase that describes the desired feature benefit from the user perspective along with identification of who that user is. User stories are further refined from initial broadly scoped epics to more detailed user stories during the Scrum process. The Product backlog may also include items which support the development of the product but not necessarily part of the product, such as knowledge acquisition in the form of developing a prototype to evaluate alternative technical solutions (Rubin, 2014).

For this project the starting product backlog was based on the expansion of a single epic user story that covered the overall goal of the lesson to be developed; “As a content developer for eLearning I want to understand what the Scrum process is so that I can start working in a team using this method”. This epic describes the desired outcome of the lesson and benefit from the user perspective, and so shaped the development for this project.

The product backlog was built through brainstorming and reference to the cited material that described the adoption and activities of Scrum, particularly Rubin (2014). Consideration

was given at this point to the delivery mechanism of eLearning that offers a more responsive, interactive approach than the more linear style of a text book. The items defined in the product backlog were targeted specifically for the intended audience of the lesson, which differed from the audience of the reference material.

The next task was to establish an estimate for the resultant user stories. The T-shirt sizing approach was selected, where backlog items were sized as Small, Medium, Large and eXtra Large. This method of estimate gives a relative indicator of effort, and was chosen for the context of a solo project, where no historical information was available to support estimation.

In Scrum the act of refining and prioritizing the product backlog is called grooming. In grooming the product backlog a focus on the needs of the audience is key. Maintaining the user perspective is aided by the product backlog items themselves being audience centric user stories. The audience for this lesson would be content developers without any experience of Scrum, therefore it was necessary to prioritize the fundamental aspects of Scrum as they would need to be delivered first. For this reason, the overview and benefits of Scrum were given high priority. While all activities of the Scrum sprint need to be taught, there is a natural order in which they are undertaken and therefore this order was followed in developing the priority order. Product backlog items detailing tools, tips, and techniques were assigned lower priority as a foundation in Scrum would be needed first to direct the student in the purpose of these items.

A sprint length needed to be set for all sprints. The Scrum approach is generally applied within a business environment where the resources available for a sprint are constrained, particularly by team members working hours. Sprint length was set to ten days to enable the cycle to repeat within the confines of the project timeline.

Sprint one was initiated with the activities that fall under sprint planning. The sprint goal; to establish a lesson that presented an overview of the Scrum approach to eLearning for content developers who had no previous exposure to Scrum.

The use of Scrum allowed for a very quick start of productive development activities. This quick start was possible because of the nature of creation of a product backlog where only those backlog items that are of high priority and will fill out a sprint need to be defined in detail. This contrasts to the need to complete the big upfront design and the production of the detailed requirement documents along with the associated research and analysis under a traditional Waterfall model, or the Analysis and Design phases of the ADDIE model before development occurs. In this way, it was possible to start the productive development activities under the Scrum much sooner than would have been possible under those predictive linear approaches, where the specification of requirements would still have been in progress.

A sprint backlog was created by selecting the items from the top of the product backlog list. Items were selected in order, until sufficient items were selected to fit the time available in the sprint, based on the previously defined estimates. The tasks needed to implement each of the user stories that formed the sprint backlog were identified. With these backlog items all being similar, in that they each covered the introduction of a new topic, the same tasks were identified for each of the items.

Sprint execution occupied the majority of the time and effort within the sprint. In executing the tasks defined in sprint planning each of the elements of the lesson was completed. There was a natural order to the tasks undertaken, and most needed to be completed in a linear fashion for each element or backlog item. Certain tasks benefitted from being completed in succession across each element or backlog item. That is to say that for the explanation on the

benefits of Scrum user story, the research task was undertaken ahead of the task to record voice overs and then in turn the tasks for assembly and test followed in sequence, if not immediately one after the other. Some efficiencies were realized by performing the tasks that were similar across user stories. In this way, the execution of the design layout tasks for all user stories were grouped together. This not only allowed for a focus on a particular type of task, but also addressed interdependencies, such as screen navigation, in the lesson.

*Learning objectives* set out to identify to the student the aims of the lesson, identify what they will learn and form a mechanism on to determine whether both the student and lesson have been successful upon completion. The learning objectives are a further thread through the entirety of the lesson and their development was included in sprint execution. Each objective focused on a topic included in the lesson and was derived from the user stories selected for the sprint backlog.

The solo nature of this project certainly had some bearing on tasks were conducted during sprint execution. As an individual undertaking the tasks the tendency to focus on individual tasks and avoiding less productive multi-tasking results in a fairly linear execution of tasks. A Scrum product development team with multiple members would certainly have undertaken a number of tasks in parallel and offers the opportunity or complexity of collaboration. While the use of Scrum as an individual may not fully exercise all elements of the Scrum approach, it does not hinder the evaluation of Scrum on those elements that were used. The principles of Scrum have been seen to offer a practical approach for the individual by Cohen (2015) and Wax (N.D.).

The *daily standup meeting* activity was one area that was not fully realized in the solo execution of this project. The solo approach did not allow for the separation of roles of *Scrum Master* and development team within the daily standup. In the solo approach, the daily standup

served as a status check on progress, an outlook on what was to be done next and formed a point at which to reflect on issues or potential issues.

The *inspect and adapt* activities of the *sprint review* and the sprint retrospective perhaps offer the most value to this development process. A traditional predictive linear approach does not cover these aspects as it assumes the correct design and requirements specification from the outset. Additionally, in such a project the endpoint is one of a completed product whether any review of the product takes place, there is no opportunity given to address those findings within the project. Where there may be successive developments of products, in say a series of Waterfall projects to produce incremental versions of a product, a lessons learnt process may be undertaken but is not an in-built function of the approach nor applicable in-line. The ADDIE model does go some way to address this based upon the interpretation of the practitioner of the Evaluation component. This component can be regarded as the evaluation of the learner or extended to include the evaluation of the training created and the process used to do so. Scrum includes these activities to both improve the product and form the basis of process improvement.

Scrum's review and retrospective activities allow for the detection of issues. The frequent occurrence of these during the sprints to build a more complete product, allow the participants to not only learn from their mistakes but also address them, and through the next sprint evaluate the correction.

During the review of the first sprint several technical issues with the implementation were raised. This included a navigation issue that would allow unintended navigation to sections. These issues would be added to the product backlog and addressed in the second sprint. This review also raised whether the user stories in the product backlog, learning objectives in the lesson and the content to deliver that learning objectives were aligned to each other. The way in

which these links could be document was raised as an issue with the process and would be considered during the sprint retrospective. Ensuring that the stated learning objectives and how the topic was demonstrated were aligned reinforced the need for Instructional Design.

The review benefitted from the involvement of the capstone adviser to provide a different perspective. This was not only beneficial due to the predominantly solo nature of the development but also highlights the benefit of the Scrum approach of involving stakeholders, including those external to the daily development activities, during each cycle with the opportunity to include feedback and aim for a product that better fits the customer's needs.

The retrospective for the first sprint identified an issue with estimation and time taken during sprint execution. In addressing the issue of estimation, research identified other commonly used methods to achieve estimates for user stories in Scrum. The first based on a points value which may offer more granularity in estimation than the T-shirt size approach used. A further alternative is the use of the *planning poker* process to refine estimates. This process achieves an estimate by aiming for a consensus amongst a group, each individual refining their own estimate based on discussion with others until they agree. It was therefore seen as applying to team based development only. Based on the first sprint, consideration was given to the sprint backlog for the second sprint to ensure a lower total for the estimates of items planned into the sprint. This is an imprecise approach when using T-shirt sizing approach to estimation. The decision on revising the estimation approach was left for a later retrospective, once this first reduction was implemented and after time tracking for tasks could be implemented. Here Scrum offers the opportunity for a course correction, rather than the late delivery of product that would occur under a traditional approach.

In a Scrum approach the activity of product backlog grooming is a continual one. For the solo approach undertaken here, the backlog grooming was conducted in linear fashion between the first and second sprint, once items related to technical issues from the sprint one review activity were added to the backlog.

The resultant product backlog contained a mix of user stories and technical enhancements. There is some variability in the approach to Scrum where some imply that the product and sprint backlogs are only comprised of user stories. In that approach, non-functional features or items are included in the form of the user story, from perhaps a *product manager's* perspective, to give assurance that both action and benefit are identified. Alternatively, the suggestion is that the product backlog can include backlog items of other types (Rubin, 2014).

In setting the sprint goal for the second sprint to be the updates for technical issues along with the expansion and addition of subject material, Scrum's advantage of being both incremental and iterative could be examined. Some of the backlog items for the second sprint were the addition of subject areas to the training, similar to that of the first sprint. Under the first sprint each of this type of user story had been broken down into the same set of tasks. In sprint planning for the second sprint that approach was reconsidered, with the tasks re-evaluated. The tasks identified for those user stories in the second sprint, while similar to each other, were not identical. For this sprint, more attention was paid to develop them with the context of each user story, rather than being duplicated because of type.

Sprint execution for the second sprint followed the same approach as that of the first sprint. This sprint included the modification of areas of the product that were produced in the earlier sprint. In performing the re-work some of the implementation previously completed was

re-evaluated and the design choices made prior re-considered. Here Scrum showed its benefit in adapting to change.

A concern was raised following the execution of the second sprint. With Scrum elements of the product are completed piecemeal based on the backlog items, and this may lead to a tendency to bypass design and architecture of a solution. This area deserves some attention during all sprint activities.

A further issue was raised during the second sprint in how learning objectives are addressed when following the Scrum approach. The question raised particularly focused on whether they are considered as items in the backlog, whether user stories are used in their place or whether one derives the other. With user stories based on particular topics within the eLearning objectives, the learning objectives have the potential to correspond to a single user story, or may cover multiple, being better aligned with a theme or epic.

## **Results and Recommendations**

The Scrum approach was successfully used to create practical, workable eLearning content. Two releases of the eLearning lesson were possible in following this approach, that would not have been possible under a traditional linear approach due to its reliance on deep analysis and design at the outset of the project. The inspect and adapt activities of review and retrospective helped issue course corrections, discovering issues in product and process early. This allowed for them to be addressed early. In the case of review, techniques were found to handle transitions between slides and could be incorporated into new work as well as corrected quickly for the few incidents that existed. Finding these later would have involved significant re-work and additional testing.

This project achieved its aim to produce an eLearning lesson using a modern development methodology and enabled an evaluation of the approach in practice. The resulting product in the form of the lesson after the second sprint is in a form that can be delivered to its target audience or further refined. The evaluation of the training material produced and how effective it is at educating the students on the topics covered is beyond the scope of the project. The determination of how to perform and execute such an evaluation with a sample of the target audience is left to further research.

Based on the successful production and iteration of an eLearning lesson the first recommendation from this project is that when there is a need to develop eLearning material the Scrum method is considered. In using this approach for eLearning, its benefits are that it allows a rapid start enabling early delivery of an initial lesson, the process itself addresses change head-on allowing uncertainties and mid-process discoveries to be handled early and it leads to improved efficiency in developing eLearning content through process improvement brought about in the retrospective activity.

This project found particular benefit to the retrospective activity being integrated into the approach. The retrospective activity allowed for the reflection on issues that may have gone undiscovered and ultimately caused issues in the development process and at least ensured that further development work was conducted efficiently. It is the reflection introduced in the retrospective that highlighted the importance of design and overall lesson flow.

A further recommendation is that when Scrum is used to produce eLearning that attention is paid to the overarching design of the lesson. Care must be taken to ensure the Instructional Design elements related to learning approach are covered. This would include ensuring that learning objectives, the topic contents and the method in which the learning objectives are

evaluated are cohesive. This element was seen during the sprint review and retrospective, and addressed in part in the planning artifacts through the inclusion of a cross reference between user stories and the slides containing the topic contents. Scrum encourages the refinement of the approach itself to meet the needs of the situation under which it is being executed. This can be use beneficially to include the design aspects, by taking best practices from traditional Instructional Design approaches. The development of material on individual topics seems to be a natural fit for the Scrum approach where the parallels to software development, the origin of Scrum, are clearer.

Learning is built on either breaking down concepts or building them up from basics. The danger in handling lesson topics in piecemeal fashion is that this overarching design is lost or overlooked. Focus on individual backlog items and the subsequent division into tasks may promote this piecemeal approach. Additionally, this division or building of concepts requires that a certain topic or user story has pre-requisites. Good user stories are evaluated for being independent (Rubin, 2014). They should be loosely coupled or the dependency between user stories minimized. With pre-requisites for learning topics, attention to this principle is needed when developing user stories.

It is recommended therefore that in grooming the product backlog and during sprint planning that the interaction and dependencies of the backlog items are considered. Keeping this in focus when ranking items in the product backlog, along with the design elements previously mentioned, will support the production eLearning lessons that have a logical progression and are effective for the learner.

In comparing the commonly used ADDIE model to the application of the Scrum approach for eLearning, the equivalent of the ADDIE Analysis component occurs in the

development of the backlog items, be it initially or when refined through to sprint planning. The Design component is split between backlog handling activities and sprint execution. Develop and Implementation components fall squarely within Sprint execution stage. Evaluation in ADDIE has a wider exposure, covering the evaluation of the student, the lesson and to some the process for creating the class. The former needs to be addressed within the Scrum planning and execution steps, while the latter two correspond to the Review and Retrospective elements.

When applying Scrum to eLearning, an awareness of the ADDIE model is recommended for consideration of the activities that it encompasses rather than its linear approach. It is specifically the attention to Design that is needed due to Scrum in principle being more general than specifically an Instructional Design approach.

### **Conclusion**

The lesson produced can be launched by copying and pasting the following URL into your web browser: <https://goo.gl/sB4ctE>

In producing an example eLearning based lesson using Scrum, the approach was shown to be usable and practical for the task. The expected benefits of early start, earlier first product release, responsiveness to change and issues along with the advantages of process improvements and reflective examination were seen during this project. Please click on the hyperlink provided above to explore the resultant lesson.

While some aspects of Scrum were not fully exercised due to the nature of this being performed as a solo project, they were not seen as an impediment to following the Scrum approach, rather some further benefits of Scrum may be seen in team based development. The one area that needs focus when using Scrum for eLearning development is in the design aspect.

Where other domain specific models, dedicated Instructional Design models specifically highlight this area, the first D in ADDIE is design, Scrum being more general does not call this out but relies on the capabilities of the practitioner to achieve this.

### **Further Research possibilities**

Following the completion of the two sprints in developing the eLearning lesson for this project there is the opportunity to progress this lesson further or develop additional lessons to form a series. In terms of both research and the content covered in the class, Learning Objectives and their integration into user stories and the development process offers an avenue for further input. The execution and refinement of the Scrum approach for eLearning along with exploration of Instructional Design model could be progressed further. Likewise the evaluation of the effectiveness of the developed lesson could be undertaken by presenting it to the target audience and contrasted against a lesson developed following an Instructional Design model.

During the research the following concepts were encountered, but fell out of scope for this project that concentrated on the core elements of Scrum. These elements are therefore mentioned by name, but exploration is left to the reader in the form of cited materials. This project made light use of the *Definition of Done* and could expand to make use of *Acceptance Criteria* or *Conditions of Satisfaction* for user stories. This could be investigated in conjunction with the Degree aspect of the structured phrasing of the ABCD (Audience, Behavior, Condition, Degree) Learning objective model.

This project was completed as a solo activity. The extension of considering this approach for a team based eLearning development project offers further possibility for research in this area. In so doing the dynamic of Scrum roles can be observed. In response to the element of

design that needs attention in Scrum, the introduction of an *Instructional Designer* as a product manager would allow the introduction of more learning design approaches that could shape content.

## Glossary

- **Acceptance Criteria** – A list of tests associated to a user story to verify has been implemented correctly.
- **Conditions of Satisfaction** –Criteria that are used to identify that a user story has been implemented completely and as intended
- **Daily standup meeting** – A brief meeting that is part of the scrum activities that is held daily for all team members. Each team member provides an update on what was achieved in the previous day, what is next and raise any issue.
- **Definition of Done** – Establishes what activities need to have been completed to indicate the completion of a user story. This may include that all material must be tested with no issues, that material is reviewed or spelling checked and that particular documentation is complete.
- **Inspect and adapt activities** – Term used to cover the Sprint Review and Sprint Retrospective. Both activities inspect what was achieved or performed during the sprint and suggest improvements.
- **Instructional Designer** – Individual responsible for the educational aspects and approach that are implemented.
- **Learning objectives** – Structured phrases used to identify what the student will learn, and also used to evaluate the student on completion of learning.
- **Planning poker** – Team based estimation approach, in which each team member proposes an estimate of effort for a particular task. When each of the estimates do not agree, a discussion is held and each team member refines their estimates. This is repeated until consensus is achieved.

- **Product manager** – A role identified in Scrum for the individual who drives the direction of the product and is responsible for grooming the product backlog.
- **Scrum Master** – A role identified in Scrum for the individual who facilitates the Scrum approach and Daily Standup meetings. This individual aims to remove any impediments to the ongoing development process.
- **Sprint retrospective** – A Scrum activity that supports continuous improvement of the implementation of the Scrum approach. It allows members of the Scrum team to reflect on what went well and what went badly during the sprint, and generate process improvement ideas to implement in future.
- **Sprint review** – A Scrum activity to allow participants and stakeholders to provide feedback on the features that were implemented in a sprint. The comments raised in this pertain to the product produced and target product improvement.

## References

- ABC (2015, December 2). *AirAsia QZ8501: Poor pilot emergency training, negligence behind plane crash, analysts say* Retrieved from <http://www.abc.net.au/news/2015-12-02/poor-pilot-emergency-training-behind-airasia-crash-analysts/6995644>
- American Society for Training and D., & Biech, E. (2014). *ASTD Handbook : The Definitive Reference for Training & Development, 2nd Edition*. Alexandria, VA: American Society for Training & Development.
- Anglia Ruskin University (N.D.). *Distance Learning: A History of Flexibility* Retrieved from <https://elearningindustry.com/the-history-of-distance-learning-infographic>
- Bettio, W. D., Alves Pereira, D., Ximenes Martins, R., & Heimfarth, T. (2013). *The experience of using the scrum process in the production of learning objects for blended learning*. *Informatics in Education*, 12(1), 29-n/a. Retrieved from <http://dbproxy.lasalle.edu:2048/login?url=http://dbproxy.lasalle.edu:2053/docview/1352758723?accountid=11999>
- Bruce, P.A. & Heller, L.M. (2011, December 17). *An Integrative Learning Design Framework* Retrieved from <http://bestpracticesonline.blogspot.com/2011/12/normal-0-false-false-false.html>
- Clark, D. (2015, September 29). *Instructional Design* Retrieved from <http://www.nwlink.com/~donclark/hrd/learning/development.html>
- Cohn, M. (2015, January 6). *How I Work, and Use Scrum Personally* Retrieved from <https://www.mountaingoatsoftware.com/blog/how-i-work-and-use-scrum-personally>
- Collins, M (1998). *Formal Methods* Retrieved from [https://users.ece.cmu.edu/~koopman/des\\_s99/formal\\_methods/](https://users.ece.cmu.edu/~koopman/des_s99/formal_methods/)

Clarke, G. (2016, January 22). *The Evolution of Software Development Practices* Retrieved from

<http://garyclarke.us/technology/the-evolution-of-software-development-practices/>

Culatta, R. (2013). *Weaknesses of the ADDIE Model* Retrieved from

[http://www.instructionaldesign.org/models/addie\\_weaknesses.html](http://www.instructionaldesign.org/models/addie_weaknesses.html)

Dawson, D. (N.D.). *Rapid Prototyping as an Instructional Design Process* Retrieved from

[http://uwf.edu/ddawson/d3net/research/rapid\\_prototype.htm](http://uwf.edu/ddawson/d3net/research/rapid_prototype.htm)

de-research.com (N.D.). *Analysis of Three Instructional Design Models* Retrieved from

[http://www.de-research.com/PhDFinalPapers/CT\\_3IDModels.pdf](http://www.de-research.com/PhDFinalPapers/CT_3IDModels.pdf)

Deloitte (2017). *Continuous Learning* Retrieved from

<http://www.bersin.com/lexicon/Details.aspx?id=12807>

Dogša T., & Batič D. (2011, February 26). *The effectiveness of test-driven development: an industrial case study* Retrieved from

<http://dbproxy.lasalle.edu:2053/docview/885705558?pq-origsite=summon>

elearningnc.gov (N.D.). *What is eLearning?* Retrieved from

[http://www.elearningnc.gov/about\\_elearning/what\\_is\\_elearning/](http://www.elearningnc.gov/about_elearning/what_is_elearning/)

Forest, E. (2014, January 29). *ADDIE Model: Instructional Design* Retrieved from

<http://educationaltechnology.net/the-addie-model-instructional-design/>

Forest, E. (2015, November 23). *Dick and Carey Instructional Model* Retrieved from

<http://educationaltechnology.net/dick-and-carey-instructional-model/>

Forest, E. (2016, November 24). *The Gerlach and Ely Design Model* Retrieved from

<http://educationaltechnology.net/gerlach-ely-design-model/>

Hartley, D.E. (2001). *Selling E-learning* American Society for Training & Development

- Hodell, C. (2016). *ISD from the ground up: a no-nonsense approach to instructional design* (4th ed.) Alexandria, VA.: ATD Press.
- Hubbard, R. (2013, October). The Really Useful eLearning Instruction Manual: Your toolkit for putting eLearning into practice. Oxford, GB: Center for Creative Leadership. Retrieved from <http://dbproxy.lasalle.edu:3759/lib/lasalle/detail.action?docID=10784818>
- Hrastinski, S. (2008, November 17). *Asynchronous and Synchronous E-learning*. EDUCAUSE Quarterly vol 31 no.4 Retrieved from <http://er.educause.edu/articles/2008/11/asynchronous-and-synchronous-elearning>
- Ibrahim, A. A. (2015, December). *Comparative Analysis between System Approach, Kemp, and ASSURE Instructional Design Models* International Journal of Education and Research Vol.3 No. 12 Retrieved from <http://www.ijern.com/journal/2015/December-2015/20.pdf>
- İşman, A., Çağlar, M., Dabaj F., & Ersözlü H (2005, July). *A New Model For The World Of Instructional Design: A New Model* The Turkish Online Journal of Educational Technology. volume 4 Issue 3 Article 6 Retrieved from <http://files.eric.ed.gov/fulltext/ED495698.pdf>
- Instructional Design Center (2017) *Instructional Design Models* Retrieved from <https://www.instructionaldesigncentral.com/instructionaldesignmodels>
- Kuhlmann, T. (2010, Feb 2) *Why E-Learning is So Effective* Retrieved from <https://blogs.articulate.com/rapid-elearning/why-e-learning-is-so-effective/>
- Lee, W. W., & Owens, D. L. (2004). *Multimedia-based Instructional Design : Computer-based Training, Web-based Training, Distance Broadcast Training, Performance-based Solutions*. San Francisco: Pfeiffer. Retrieved from

[http://dbproxy.lasalle.edu:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=111557&site=ehost-live&scope=site&ebv=EB&ppid=pp\\_1](http://dbproxy.lasalle.edu:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=111557&site=ehost-live&scope=site&ebv=EB&ppid=pp_1)

Lewis, P. A., Mai, V. A. T., & Gray, G. (2012). Bilingual asynchronous online discussion groups: *Design and delivery of an eLearning distance study module for nurse academics in a developing country*. Nurse Education Today. Retrieved from <http://dbproxy.lasalle.edu:2101/10.1016/j.nedt.2011.02.011>

Mauer, F. & Melnik, G. (206, May). *Agile methods: moving towards the mainstream of the software industry* Retrieved from <http://dbproxy.lasalle.edu:2324/citation.cfm?id=1134503>

Pappas, C. (2015, Jan 25). *Top eLearning Statistics and Facts for 2015*. Retrieved from <https://elearningindustry.com/elearning-statistics-and-facts-for-2015>

Ranakrishnan, S. (2013, July 18). *Managing Technical Debt* Retrieved from <https://www.scrumalliance.org/community/articles/2013/july/managing-technical-debt>

Reiser, R. A., & Dempsey, J. V. (2012). *Trends and issues in instructional design and technology*. Boston: Pearson.

Royce, W. (2000). *Software project management: a unified framework*. Boston: Addison-Wesley.

Rubin, K. (2014). *Essential Scrum: a practical guide to the most popular Agile process*. Upper Saddle River, NJ: Addison-Wesley.

Russel, L. (2015, April 08). *Methodology Wars: ADDIE vs. SAM vs. AGILE* Retrieved from <https://www.td.org/Publications/Blogs/L-and-D-Blog/2015/04/Methodology-Wars>

Sarasa-Cabezuelo, A., & Sierra-Rodríguez, J. (2013). *Software engineering for eLearning*. Retrieved from

[http://dbproxy.lasalle.edu:2324/ft\\_gateway.cfm?id=2536550&ftid=1415683&dwn=1&CFID=727356996&CFTOKEN=34308211](http://dbproxy.lasalle.edu:2324/ft_gateway.cfm?id=2536550&ftid=1415683&dwn=1&CFID=727356996&CFTOKEN=34308211)

ScrumAlliance.org (N.D.). *Learn About Scrum* Retrieved from

<https://www.scrumalliance.org/why-scrum>

Smith, J. (2003). Instructional Design Retrieved from <http://www.latech.edu/globalcampus/id.pdf>

Stephens, R. (2015). *Beginning Software Engineering*. Somerset, US: Wrox. Retrieved from

<http://dbproxy.lasalle.edu:3770>

The University of Michigan (1996, October 18). *Definitions of Instructional Design* Retrieved

from <http://www.umich.edu/~ed626/define.html>

Thomas, S. (2008, May 31). Agile Project Scope Retrieved from

<http://itsadeliverything.com/agile-project-scope>

Wax, D. (N.D.). *Scrum for One* Retrieved from

<http://www.lifehack.org/articles/featured/scrum-for-one.html>