Proceedings of Informing Science & IT Education Conference (InSITE) 2010

Technology in Education: An Agile Systems Approach

Bill Davey
RMIT University,
Melbourne, Australia

Bill.Davey@RMIT.edu.au

Kevin R Parker
Idaho State University,
Pocatello, Idaho, USA

Parkerkr@isu.edu

Abstract

Innovative use of technology in education often comes about through a champion exploring possibilities. Because this approach is often informal, the need to assess the effectiveness of the technology is often overlooked. An agile systems approach to the introduction of new technology in education can provide effective ways of measuring outcomes. The agile approach also allows us to build on the experience gained from using technology in other areas. Evidence from two case studies is used to demonstrate the value that can be gained from taking an agile systems approach.

Keywords: characters, agile methods, education, educational technology, animated pedagogical agents, volatile

Introduction

This paper arises from difficulties encountered in attempting to determine the effects of using animated pedagogical agents in programs across two universities. Animated pedagogical agents are animated characters designed to interact with users in computer-based environments. At RMIT in Australia and Idaho State University separate teams of researchers have been attempting to improve delivery of disparate IT-based courses through the use of on-screen characters. In both cases the projects were initiated with a view to exploring whether animated pedagogical agents could enhance the delivery of course content. It was not until some education material had been produced that the issue of outcome measurement was considered. It seems counterintuitive that people with extensive experience and detailed knowledge of how to implement and evaluate information systems neglect to put this knowledge to work in educational technologies.

The majority of educational technology projects can be categorized by some commonly occurring scenarios. New technologies arise so often and so quickly that an educationalist is often confronted with the question, "Can I do anything with this technology?" The educationalist might then attempt to construct something that works. After some facility with the new technology is achieved applications in the classroom can be attempted. Because requirements are ambiguous, involving exploring whether anything useful can be constructed or if students will find the new technology useful, there seems little point in specifying detailed requirements for the exemplar system.

Although the use of online technologies is becoming ubiquitous (Wong & Tatnall, 2009), it is far from certain that we have an accurate useful model of the way in which students use online facilities. Recent studies have shown that current assumptions about the way in which we expect the current generation of students to use technology are naïve and in error (Anagnostopoulou & Priego-Hernandez, 2009; Donnelly & O'Rourke, 2007; Franklin & Van Harmelen, 2007; McCrindle, 2003; Spennemann, 2007).

New technologies often present previously unanticipated possibilities. As such technologies become available, early adopters may conceive potential applications in their particular domain, and undertake the development of those applications. As deeper knowledge of the technology is gained, it often leads to significant changes in the requirements of the project. Although these new requirements inevitably emerge, traditional methods do not handle requirements volatility well. This common problem has contributed to the development of the approach called agile methods.

Agile Methods

Pressman (2010) defines agile methods as "A philosophy and set of development guidelines. The philosophy encourages customer satisfaction and early incremental delivery of software … the development guidelines stress delivery over analysis and design." The agile approach views requirements engineering as a continuous, iterative process. System components are constructed and delivered early, and used as the basis for communication and client trust. These components are similar to the earlier concept of functional prototypes. Supporters of Agile Methods claim that there is no other way of dealing with systems in which the client learns and therefore changes what is required, in which requirements are unknown for some reason, or in which the original environment for the project changes during the project. (Hanssen & Fægri, 2006; Paetsch, Eberlein, & Maurer, 2003; Tomayko, 2002; Turk, France, & Rumpe, 2005)

The agile approach has many advantages in educational settings that are rich in experimentation. Pressman (2010) identifies three issues that agile methods are seeking to address. First, software requirements are very difficult to predict in advance and often change or evolve as the project proceeds. Second, design and construction are interleaved in many software projects because difficulties encountered while constructing software systems often force changes in the design. Thirdly analysis, design, construction, and testing are not predictable. The agile alliance (Agile Alliance, 2009) has identified twelve principles of agile methods.

- Satisfy the customer through early and continuous delivery of software.
- Welcome changing requirements even late in development.
- Deliver working software frequently.
- Customers and developers must work together daily throughout the project.
- Build projects around motivated individuals.
- The most efficient and effective method of conveying information is face-to-face conversation.
- Working software is primary measure of progress.
- Agile processes promote sustainable development.
- Continuous attention to technical excellence enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- Teams must reflect at regular intervals on how to become more effective and then tune and adjust behavior.

These principles seem to be particularly useful in an educational setting.

There have been some notable uses of agile methods in education. (Boyle et al., 2006) have used agile methods in a major project producing reusable learning objects. The design for learning program (http://www.wle.org.uk/d4l/) has used agile methods to construct a tool for capturing learning designs. In these two cases importance was placed on rapid iteration. Functioning prototypes were used as the basis for the development of the next prototype. We assert that these functioning prototypes must be used in the classroom, with students taking on the client role.

The Cases

As noted earlier, independent development projects – one in Australia and one in the United States – investigated uses of animated pedagogical characters in learning tools. Animated pedagogical agents have attracted considerable interest in recent years, mainly for their potential impact on the intuitiveness and effectiveness of user interfaces (Prendinger, Ma, Yingzi, Nakasone, & Ishizuka, 2005). Use of characters has been found to create a social environment capable of simulating many of the characteristics of the real world (Gulz, 2004). The highly adaptable and interactive technique has the following features:

- Male and female characters of various races are available, along with a wide range of mechanized voices. Actual recorded voices can be substituted if desired.
- Explanations can be demonstrated using multiple "whiteboards" like an actual classroom.
- Dialog can be reinforced with graphs, interactive graphs, and animated graphs.
- The interaction can incorporate multimedia to provide visual reinforcement of content. Such supplements are easily updated as topics evolve, insuring that they remain current.
- Links to online resources like definitions of key terms can be embedded in presentations.
- Agent technology allows user interaction through voice recognition or via text, permitting the student to converse with the agent in a Q&A type approach.

Animated pedagogical agents directly benefit students by integrating technology into the curriculum to improve teaching and student achievement. Animated pedagogical agents enable interaction with students through mixed-initiative, tutorial dialogue (Carbonell, 1970) in the role of a coach (Burton & Brown, 1982; Goldstein 1976) or learning companion (Chan, 1996). Agents can demonstrate how to perform actions (Rickel & Johnson, 1997) or can use locomotion, gaze, and gestures to focus the student's attention (Lester, Voerman, Towns, & Callaway, 1999; Noma & Badler, 1997; Rickel & Johnson, 1997). Such nonverbal devices are a natural component of human dialogues. The use of animated pedagogical agents enhances the rate and quality of student learning. The mere presence of an anthropomorphic agent stimulates students and increases their motivation to better perform tasks (Lester et al., 1997; Walker, Sproull, & Subramani, 1994). Users interact with animated pedagogical agents in an essentially natural way, following their verbal and non-verbal navigational directives. Unlike a textual interface, students can utilize multiple senses and therefore attend to animated agents in a balanced way, with shifts to and from the content being presented (Prendinger et al., 2005). Animated pedagogical agents increase the bandwidth of communication between students and computers, and studies show that students interacting with animated agents show statistically significant increases from pre-tests to post-tests. Students faced with more complex problems show more pronounced positive effects of animated pedagogical agents on problem solving (Johnson, Rickel, & Lester, 2000).

The use of animated pedagogical agents enhances the rate and quality of student learning. Johnson, Rickel, and Lester (2000) indicate that a well-designed agent in a well-designed learning environment does not distract students but rather creates successful learning experiences. Further, animated pedagogical agents that provide multiple levels of advice combining multiple modalities yield greater improvements in problem solving, indicating that there may be important learning benefits from introducing animated agents that employ both visual (animated) and auditory (verbal) modalities to give both "practical" and "theoretical" advice.

There are a number of ways of producing on screen characters that are capable of movement gesture and speech. The cases described below each utilized a different approach.

Australian Case

At RMIT, as part of a much larger project, one academic has been attempting to improve a course in project-management using on-screen characters. The opportunity identified is simulation of conversations with actors involved in the project. Students are exposed to a project plan in which changes that are common in all projects are introduced by on-screen characters. The use of the characters is to simulate personal interactions with people involved in the project in a somewhat realistic manner. The animated characters and verbally delivered information attempts to mimic an industrial setting in which project managers must be able to interpret the information contained within telephone calls or other conversations in terms of the impact on the project plan. Changes that impact the project plan come about at random times and are often delivered using informal communication methods. Using a character to deliver these relevant pieces of information verbally parallels important aspects of the way in which projects proceed in the real world.

The project management simulation system characters were developed using Media Semantics' Character Builder. Character Builder allows the project management simulation system to integrate virtual people or avatars that interact and converse with one another in order to deliver specific information about project parameters. Character Builder can utilize a wide range of provided characters and allows the use Text-to-Speech (TTS) voices as well as recorded audio. The Character Builder creates Flash and video files that run download-free practically anywhere.

The animated characters simulate real world situations and conversations, allowing students to "eavesdrop" on simulated real-world discussions with the goal of conveying project details in a more interesting form. The project management simulation system utilizes additional approaches to convey information as well. Not only do students eavesdrop on conversations, but they also listen to phones messages, view news reports, and participate in visits to a doctor.

United States Case

The project at Idaho State University involved developing a small set of tutorials using animated pedagogical agents to present core concepts in particular subject areas, specifically introductory programming and database design courses. The project was driven by the recognition that students often read an explanation in a text but do not fully understand the concept until they hear it explained in class. Further, if they do not understand the lecture or their minds wander during class they are unable to review it to fill the gaps in their knowledge. To alleviate this problem a series of animated tutorials were developed to provide students with further explanations of core concepts, as well as viewable demonstrations of those concepts.

The core concept tutorials make use of the Microsoft Agent technology, which allows the tutorial system to incorporate interactive personalities in the form of animated characters that can move freely across the computer display, gesture, speak aloud, display their conversation onscreen, and explain concept details to student. The animated characters contribute toward an engaging environment for students, with the goal of making learning an enjoyable experience. Microsoft Agent characters and their voices can be easily changed, and actual voiceovers can be substituted for the mechanized voice if desired.

The core concept tutorials utilize multiple techniques to explain concepts. Not only does an animated character provide a verbal explanation of the concept, accompanied by text so that students can utilize multiple senses, but there is also a component that visually demonstrates key points that make up the concept. For example, when explaining normalization for a database design course the visual portion consists of a series of animations that demonstrate how the normalization process is accomplished. Normalization is traditionally a difficult concept for students to grasp, and this allows students to review either all or portions of the lecture at their leisure. Examples from the programming course include explanations of control structures – sequence, selection, and iteration – as well as verbal explanations and visual demonstrations of how memory is allocated with pass-by-value and with pass-by-reference.

Additional types of media, like video clips, can be incorporated into the lectures so that students will be able to envision how topics are applied in the workforce. The advantage of this type of media is that content can be immediately updated as topics evolve, insuring that content remains current. Links to online resources, such as a brief syntax guide for database topics or definitions of key terms can also be embedded in the presentations.

Using Agile Methods

Putting agile methods into practice requires a process model. There are quite a number of process models available for agile methods. These include adaptive software development, dynamic systems development method, feature-driven development, lean software development, agile unified process, and agile modeling (Pressman, 2010) . Each of these process models starts with identifying requirements that might be constructed to satisfy some client need. Next, a design is produced for a system that is then coded, tested, and implemented rapidly. Evaluation of the deployed code is then used to identify the client needs to be satisfied. Each iteration produces deployed software with incremental improvements and extensions satisfying feedback from the previous deployment. (Douglas, 2006) identifies the advantage of agile methods in educational systems as being an adaptive and people-oriented approach whereas previous information systems approaches were predictive and process-oriented.

In this paper we concentrate on what requirements might form the basis of the initial deployments of educational systems. At RMIT initial discussions of the aim of the first deployment centred on questions of feasibility, such as "can I make a small unit of work using a generated character and distributed through the University learning hub?" Work at Idaho State was inspired by the similar question "can I use animated characters that combine voice and gestures to explain course content, such a piece of code on the screen?"

These requirements were sufficient for the two academics to commence the project and start work on code. They are typical of the requirements set by academics and researchers starting out with a new technology. The problem stems from the fact that the requirements are based on the programmer rather than the learner, and such requirements do not naturally lend themselves to evaluation or incremental improvement of use of the technology.

Applying Agile Methods

To apply agile methods we need most of all to be able to deliver working software that will satisfy the client, i.e., the student, in a very short time. This means that the initial set of requirements must be framed in such a way that we can realistically produce a system quickly. Table 1 shows a possible client-centered set of initial requirements for the RMIT project.

Table 1: Two Iterations at RMIT

Requirement type	First iteration	Requirements arising from feedback of first implementation
Project Drivers	Techniques must be developed to enable character-based units to be delivered on the web	Students need information about changes to their projects delivered in a verbal form
Project Constraints	Delivery through the blackboard system	Students must be informed of the way they get information needed for their assessments
Functional requirements	Students should be able to access the unit offsite using the learning hub	Character units must become clearly available at exactly the correct time and in sequence
Non-functional requirements	Programming of the character must be possible for a semi-skilled academic to accomplish	Characters should be identifiable as persons in the fictional project and be easily identified by students as being part of their project
Project Issues	Software licensed to the University must be used. Units must be constructed and tested before the start of semester.	Selection of appropriate material for delivery by characters

The requirements for the first iteration were very much a "proof of concept" that would normally be associated with the first testing phase of a project with much more ambitious requirements. These requirements are framed in a way that allows for the overall direction of the original project brief. The requirements neglect to mention any educational outcomes even though the case selected to verify that the software performed adequately in the delivery system was chosen from the educational content of the course. This makes assessment of the outcome of the first iteration much less complicated. In the second iteration, now having some knowledge of what the system looks like, we can establish some meaningful concrete educational goals. Again the requirements tend toward the direction of the project brief, but attempt only to provide immediate satisfaction for a particular requirement. Table 2 shows the same information for the project at Idaho State.

Table 2: Two iterations at Idaho State University

Requirement type	First iteration	Requirements arising from feedback of first implementation
Project Drivers	New functionality must be added to the online delivery system that allows character based material to be delivered	Students need interactive lecture supplements to reiterate core concepts from lecture and text.
Project Constraints	Requires installation of animated character, .Net framework, Microsoft Agent core components, text-to-speech engine, and Microsoft SAPI runtime binaries.	Must fit within the normal time frame of the learning units to be supported
Functional requirements	Students should be able to access lectures from any computer with proper software installed.	Characters must be clearly associated with the learning unit in a way that students find logical
Non-functional requirements	Programming of the character must be possible for a semiskilled academic to enter	Characters must be engaging and keep students' attention.
Project Issues	Embedded visual aids must be synchronized with verbal presentation.	Core concepts must be selected and lectures developed.

Discussion

The cases described implement all of the aspects of agile methods. In neither case were teams working together, and the assessment of achieving requirements and satisfying the customer are dependent on the individual academic delivering the course. The core idea we were working is that of setting goals, producing working systems that satisfy specified need, and undertaking the next iteration having gained knowledge from the previous. There are clear benefits to taking this approach to the implementation of new technologies in education.

Using the Knowledge Developed in Information Systems

Each problem domain has its own characteristics. When implementing an educational innovation a wide range of issues can be considered. Education theory, cognitive science, social interaction, and psychology are disciplines in which new information is often the trigger to implement a new educational technique. It is not uncommon for an innovation to be prompted by some breakthrough. In business this is often something that comes from management theory of marketing. When a project brief has been defined that involves implementation of new technology the next step in most domains is to use a systems methodology. This seems very uncommon in education. It also seems common in education for there to be subsequent difficulties in determining if the innovation has been successful. It would seem sensible that systems in education be developed in the same way as in other domains.

Recognition and Incorporation of the Well-Known Volatility of Requirements When Implementing New Technologies

One thing that has been learned from information systems research is that system requirements are volatile. This is particularly true when new technologies are being implemented. There are two important factors. The first is that when new technologies are involved we have little idea when we start as to what is possible. As the technology is used and we learn what is possible it brings to mind new requirements. The second factor is that as new technology is implemented the environment does not remain static. This is as true in the education sector as any other. Even as innovations are being implemented curriculums undergo changes. This volatility of requirements has led the information systems discipline to develop approaches that incorporate requirements volatility.

Continual Delivery of Working Systems That Satisfy a Need

The use of agile methods has been particularly useful in innovative projects as stakeholders can see and use system components all through the project. It is possible to interact with aspects of the new technology before the completion of the project. Stakeholder involvement in the project leads to informed feedback for the development team. All of these characteristics are useful in an educational setting.

More Realistic Requirements That Can Be Assessed in a Realistic Way

The described projects had not yet been completed after more than a year. Time frames from conception to completion of the implementation of an innovation often cover more than two semesters. Implementation of new technology in education often starts with grant proposals. In a grant proposal we must convince the funding body that we have a clear idea of what happens when the technology is implemented. For a really new technology this is obviously a silly idea. This means the technological innovation in education is often assessed in terms of requirements written in a grant proposal that happen well before any idea of the impact of the technology can be possible.

Assessment done in this setting is often impossible because outcomes cannot be compared before and after a project when the cohort of students is completely different.

Conclusion

This paper reports on two similar projects being undertaken on either side of the Pacific. These projects were seeking to implement a new technology, and underscore the issue of requirements volatility in educational technology. In systems implemented outside of education it has been found that requirements volatility can be addressed effectively using agile methods. We have shown that the agile methods found to be so useful in other knowledge domains have the potential to be equally as useful in the educational domain.

References

Agile Alliance. (2009). Retrieved 17 Nov, 2009, from www.agilealliance.org/home

Anagnostopoulou, D. P., & Priego-Hernandez, J. (2009). An exploration of perceptions of learning and e-learning held by students who withdraw and those who persist with UK higher education [Electronic Version]. *Brookes eJournal of Learning and Teaching, 2*. Retrieved 20 November 2009 from http://bejlt.brookes.ac.uk/article/an_exploration_of_perceptions_of_learning_and_e_learning

Boyle, T., Cook, J., Windle, R., Wharrad, H., Leeder, D., & Alton, R. (2006). *An agile method for developing learning objects*. Paper presented at the ASCILITE Conference, Sydney, Australia.

Burton, R. R., & Brown, J. S. (1982), An investigation of computer coaching for informal Activities. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press.

Carbonell, J. R. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems, 11*(4), 190-202.

Chan, T. W. (1996). Learning companion systems, social learning systems, and the global social learning club. *Journal of Artificial Intelligence in Education, 7*(2), 125-159.

Donnelly, R., & O'Rourke, K. C. (2007). What now? Evaluating eLearning CPD practice in Irish third-level education. *Journal of Further and Higher Education, 31*(1), 31-40.

Douglas, I. (2006). Issues in software engineering of relevance to instructional design. *TechTrends, 50*(5), 28-37.

Franklin, T., & Van Harmelen, M. (2007). *Web 2.0 for content for learning and teaching in higher education*. Retrieved 20 Nov 2009 from http://staff.blog.ui.ac.id/harrybs/files/2008/10/web-2-for-content-for-learning-and-teaching-in-higher-education.pdf

Goldstein, I. P. (1976). The computer as coach: An athletic paradigm for intellectual education. *Artificial Intelligence Laboratory Memo 389*, Massachusetts Institute of Technology, Cambridge, MA.

Gulz, A. (2004). Benefits of virtual characters in computer based learning environments: Claims and evidence. *International Journal of Artificial Intelligence in Education, 14*(3), 313-334.

Hanssen, G. K., & Fægri, T. E. (2006). *Agile customer engagement: A longitudinal qualitative case study*. Paper presented at the ISESE'06, September 21–22, Rio de Janeiro, Brazil.

Johnson, W. L., Rickel, J. W., & Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education 11*, 47-78

Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., & Bhogal, R. S. (1997). The persona effect: Affective impact of animated pedagogical agents. *Proceedings of CHI '97*, 359-366.

Lester, J. C., Voerman, J. L., Towns, S. G., & Callaway, C. B. (1999). Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents. *Applied Artificial Intelligence, 13*, 383-414.

McCrindle, M. (2003). *Understanding generation Y: Principal Matters, 2003*. Retrieved 20 Nov 2009 from http://www.learningtolearn.sa.edu.au/Colleagues/files/links/UnderstandingGenY.pdf

Noma, T., & Badler, N. I. (1997). A virtual human presenter. *Proceedings of the IJCAI Workshop on Animated Interface Agents: Making Them Intelligent*, 45-51.

Paetsch, F., Eberlein, D. A., & Maurer, D. F. (2003). Requirements engineering and agile software development. *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03).*

Prendinger, H., Ma, C., Yingzi, J., Nakasone, A. & Ishizuka, M. (2005). Understanding the effect of life-like interface agents through users' eye movements. *ICMI'05*, October 4–6, 2005, Trento, Italy.

Pressman, R. S. (2010). Software engineering: A practitioner's approach (7th ed.). New York: McGraw Hill.

Rickel, J., & Johnson, W. L. (1997). Integrating pedagogical capabilities in a virtual environment agent. *Proceedings of the First International Conference on Autonomous Agents*. ACM Press.

Spennemann, D. H. R. (2007). Learning and teaching 24/7: Daily internet usage patterns at nine Australian universities. *Campus-wide Information Systems, 24*(1), 27-44.

Tomayko, J. E. (2002). *Engineering of unstable requirements using agile methods.* Paper presented at the International Workshop on Time Constrained Requirements Engineering at IEEE Joint International Requirements Conference.

Turk, D., France, R., & Rumpe, B. (2005). Assumptions underlying agile software-development processes. *Journal of Database Management, 16*, 62.

Walker, J. H., Sproull, L., & Subramani, R. (1994). Using a human face in an interface. *Proceedings of CHI-94*, 85-91.

Wong, L., & Tatnall, A. (2009). The need to balance the blend: Online versus face-to-face teaching in an introductory accounting subject. *Journal of Issues in Informing Science and Information Technology, 6*, 309-322. Retrieved from http://iisit.org/Vol6/IISITv6p309-322Wong611.pdf

Biographies

Bill Davey is a Senior Lecturer in the School of Business Information Technology at RMIT University, Melbourne, Australia. His research interests include methodologies for systems analysis and systems development, information systems curriculum, and information technology in educational management. Bill and Kevin have worked together co-operatively on many occasions. They have co-operated on several joint research projects and coauthored a several of papers relating to management information systems, programming, computers in management, and IS curriculum.

Dr. Kevin R. Parker is a Professor of Computer Information Systems at Idaho State University. He has taught both computer science and information systems courses over the course of his nineteen years in academia. Dr. Parker's research interests include e-commerce marketing, competitive intelligence, knowledge management, the Semantic Web, and information assurance. He has published in such journals as *Informing Science: the International Journal of an Emerging Transdiscipline*, *Journal of Information Technology Education*, *Journal of Information Systems Education*, and *Communications of the AIS*. Dr. Parker's teaching interests include web development technologies, programming languages, data structures, and database management systems. Dr. Parker holds a B.A. in Computer Science from the University of Texas at Austin (1982), an M.S. in Computer Science from Texas Tech University (1991), and a Ph.D. in Management Information Systems from Texas Tech University (1995). Before entering academia Dr. Parker was a programmer/analyst with Conoco, Inc.