IMPLEMENTING DESIGN INTO INSTRUCTION: INTERSECTIONS BETWEEN THE WATERFALL MODEL AND ADDIE MODEL

Ryan Eller
Masters of Instructional Science and Technology - Program Coordinator
California State University, Monterey Bay
Email: reller@csumb.edu

Abstract: *Instructional design and software design, while two different fields, have many commonalities and finding intersections between the two fields is of great importance to instructional designers, subject matter experts, and ultimately, the educators who use their deliverables. Many instructional designers work on large projects, which in certain circumstances; will include the creation of new learning software. This paper suggests that combining the Waterfall software-design model, with the ADDIE instructional-Design model, could create a more collaborative process between software designers and instructional designers. In this newly proposed design model*, a combination of features of the two previous models allow for a seamless step-by-step design process.

Keywords: *Instructional Design, Software Design, Waterfall Model, ADDIE Model, E-Learning*

1. INTRODUCTION

Higher education (Campbell, Schwier & Kenny, 2009), secondary and primary schools (Cannistraci, 2011) , amongst other private learning centers (Gedik, Hanci-Karademirci, Kursun and Cagiltay, 2011) all make use of instructional designers and a multitude of instructional design processes. Use of designers, their impact, and ultimately, their success has been a source of debate in the greater academic community. In a massive study conducted by David Cook, et. al., instructional design was shown to have many useful outcomes for simulation-based education (Cook, et. al, 2013). However, Cook asked a markedly important question of the greater academic community, "we simultaneously highlight the need for further research elucidating what works, for whom, under what circumstances." (Cook, et. al., 2013, p. 876) Learning habits, the learning environment, instructional outcomes, and individual study habits were all found to have a marked impact on a student's individual success (Cook, et. al., 2013). In short, instruction designers, and ultimately the educators that use their final products, cannot always be sure of how effective their instructional materials may be.

To help in the design of effective materials, many instructional designers make use of a brevity of instructional design models. The ADDIE Model, Dick and Carey Model, Kemp's Instructional Design Model, among others, all can be used by educators to help create instruction that is guided by systematic approach. As time passes however, previous instructional design models may become less effective due to factors not limited to new technology, outdated learning methodologies, and new approaches for distance learning in higher education. (Khodabandelou & Samah, 2012)

Similarly, software design models are used in the greater world of software engineering and design to create programs, applications, and software that takes less renditions for a strong finished product. Models such as the Waterfall Model, V-Model, Dual Vee model, amongst others, help to give users a guided approach to engineering design and creation of their given product. Even though software engineering design models have proven effective (Gorschek, Tempero & Angelis, 2014), many senior designers do not use a systematic approach to design, unless they have a significant academic background (Groschek, Tempero & Angelis, 2014).

With the creation of open-source educational technology, and its prevalent use in higher education, the author recommends a combination of a both the ADDIE instructional design model and the Waterfall software design model. The author believes that a combination of the two models will allow subject matter experts, instructional designers, and software engineers (who are informed by the prior two groups) to work in tandem with one another. The author has selected these two models because of their history of usage, potential for successful combination, and their ease of use. (Molenda, 2003; Lott, 1997)

2. THE ADDIE MODEL

The ADDIE Model has a varied past rooted in the creation of instructional systems design (ISD) near the end of World War II. (Allen, 2006) In fact, there is no one "correct" form of the ADDIE model, instead it is believed to have been developed over time by a slew of researchers and instructional designers. (Molenda, 2003; Allen, 2006; Tennyson & Michaels, 1991) The model itself stems from a systems engineering approach (Allen, 2006) and is focused on creating a student-focused, yet instructional-designer driven creation of instructional materials, and ultimately, instruction itself.

The ADDIE Model is a five step model, where each letter in the acronym stands for a structured design process. Each step builds upon the last to allow the user to take a coherent and focused approach to instructional design.

The five steps in the instructional design model are as follows (Allen, 2006):

1. **Analysis**: The first step of the design model is focused on finding out the basic needs of the learner to structure instruction upon. A learner analysis or survey is a common application of this stage in the design process.
2. **Design**: During this phase, instructional designers either develops plans for new instructional materials or revise existing materials to better fit the current needs of the learning group. Formative evaluations of instructional goals and strategies are used to continue to formulate learning materials and the course structure.
3. **Development:** After the final design plan is in place, the development of the final learning products can begin to take place. As each phase, material, or module is created, reviewers (internal review, students, teachers, subject matter experts) can provide insight for continued formative evaluation.

4. **Implementation:** After the entirety of the learning materials are created, implementation of instruction takes place.
5. **Evaluation:** Formative evaluation, which takes place during each step of the design process, and summative evaluation, which takes upon completion of the implementation phase, allows the instructional designers to constantly improve the class, learning materials, or instructional software, both during its use and for future renditions.

The hallmark of the ADDIE Model is that it is a model that allows the user to analyze the need of instruction all the way to the completion and its outcome. As such, the model can be used in any educational or training-focused setting and each step is malleable to the needs of the instructional designer, subject matter experts, and the intended users.

3. THE WATERFALL MODEL

The Waterfall Model stems back to research and design approaches created in 1970, by Winston Royce (Royce, 1970), as a formulaic approach to design that relied on each step being completed before moving on to the next. Similar to the ADDIE Model, the Waterfall Model is more of a collection of ideas from various scholars, focused on this approach to software design. The model allows the user to define the scope of their work and follows up the actual software design (Lott, 1997).

The seven primary steps in the model can be described as (Royce, 1970):

1. **System Requirements:** A detailed analysis of the operating system requirements, and limitations, is first performed.
2. **Software Requirements:** After delineating the system requirements, an outline of software functionalities and uses is created.
3. **Analysis:** After the limitations and scope is created within the two previous confines, an analysis is performed to determine if the software is filling user needs and specifications.
4. **Program Design:** The complete overview of the software is now created and the operating procedures can be described.
5. **Coding:** The creation of the software and its functions begins.
6. **Testing:** Software engineers and programmers test the functionality of the software to see if its intended function is being addressed or if further coding will take place (or in some cases, a radical redesign of the program itself).
7. **Operations:** Once testing is complete, installation and usage by the target audience can be performed.

Royce actually believed his own design to be too rigid and risky based on following these steps systematically (1970). The model is noted as being good for large projects, with limited interactivity due to the heavy amount of documentation, lengthy design timeframes, and vast amount of coding (Boehm, 1988). However, by limiting the design process to remain within each step until adequate completion, the Waterfall Model aims to reduce costly redesigns and reworks of software. (Boehm, 1988)

4. APPLICATIONS OF BOTH MODELS TO INSTRUCTIONAL DESIGN

Instructional design is a growing field. With the advent of new learning technologies, a creation of a hybrid software-instructional design model will benefit the greater ISD field, software designers who create educational software and the end-users who receive not only an instructionally sound course, but a completely functional program as well. As such, the author recommends a hybrid model based on the following ideas:

i. Instructional Designers, who are also sound programmers, will be able to articulate their design process from a dual-perspective. By following the ADDIE Model, yet also stemming the creation of their materials through the Waterfall Model (regardless of if they are creating software), they will be able to slow down their design process, limiting errors, and saving time on redesigns of course structure, learning materials, and course goals and objectives.

ii. The Waterfall Model can be applied to non-software design. If a designer is creating a course, he or she could very likely change the "system" requirement step of the model into the "institution" requirement. Thus, each step can be redefined as needed to fit the designer's needs.

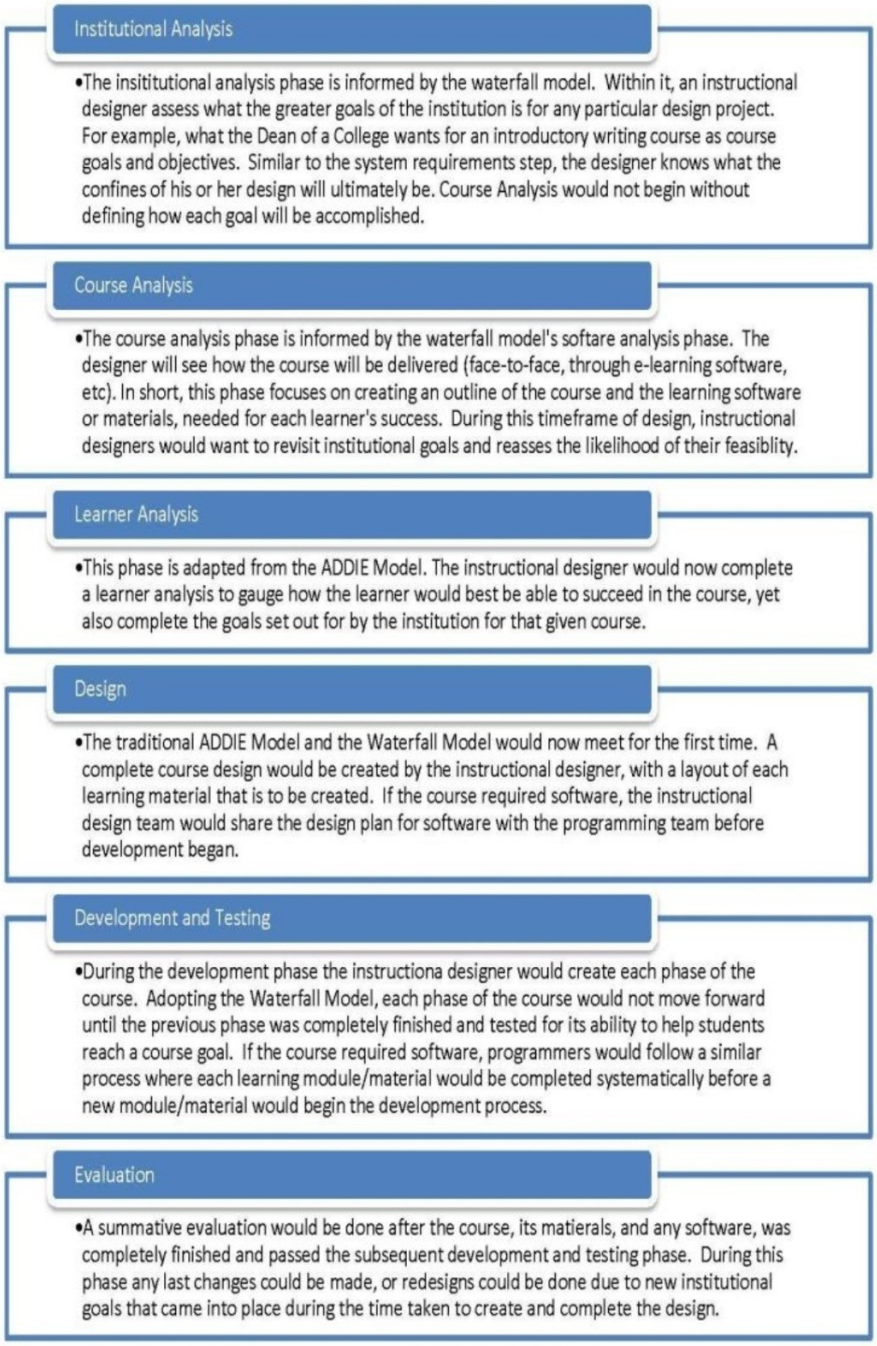
iii. By creating a co-model, large instructional design projects can be systematically created with the designers and programmers in mind. By adhering to a more rigid step-by-step process, institutional-level design projects can be completed with less need for complete overhauls. In short, checklists for both groups can be created to keep each stakeholder in line with the overall design objectives.

5. PROPOSING A CO-MODEL

By combining the two models, the author proposes that steps, and the processes within each step, of each model can coalesce for a cohesive model that allows a team of instructional designer (and a potential team of software designers, if the project requires) a more simplistic, step-by-step design process. In Figure 1, the new co-model is proposed with a description of each new step. In the new model, an emphasis on the initial analysis is present. In many higher education institutions, knowledge of what students need to learn to be successful, post-graduation, is lacking (Nunley, Bers, Manning, 2011). As such, the author recommends that instructional designers, and course instructors, regularly reassess learning needs and how they meet institutional goals and objectives. This creates a three-way feedback loop between knowing what learners currently know, what the course instructor thinks his or her student must know, and how the necessary institutional outcomes (such as graduation requirements and degree completion timeframes) can be met in an effective manner. After a more thorough analysis took place, design could begin with design in regards to course outcomes, learning materials, and any software needed to obtain these goals (this becomes magnified if the course is reliant on technology). Design of learning materials would not begin without the course outcomes being completely solidified. Each phase of design would thus follow the Waterfall Model's recommendation to not begin the next phase of design, without the previous being completed. Adopting from the ADDIE Model, formative evaluations could be done at the course, learning

material, and software level as each part of the course was being finalized. Development and testing would be able to take place after a complete design of the course and its materials were created. Development would also follow a strict pattern of completion before a new development phase would begin. Testing would be interwoven into this process, with a team of testers (a sample student, instructor, subject matter expert) testing instructional content, while software would be tested by beta-testers.

Figure 1: Co-Model Steps of Design



Lastly, a summative evaluation would take place once the entire course was created. After the summative evaluation, if the course adequately meets the needs of the learners, has sound design, and has been tested for flaws, then it would be rolled out formally.

6. CONCLUSIONS

The new Co-Model shared here gives instructional designers an opportunity to adopt both an instructional design process, while also incorporating software design principles. Even if the instructional designer learns that software design does not need to be present, adopting aspects of the Waterfall Model will allow designers to take a more fully-focused approach to their designs. Due to the model's focus on completion before progression, it is arguable that fewer mistakes will be made during any facet of the design process, from learning objectives to potential software design.

By combining the two models, instructional designers will be familiar with the design practices in the ADDIE Model. In short, separate teams of software designers and instructional designers will have their needs met as content-creators, since this model aims to take best practices from both models.

However, it is arguable that by combining these models, a potential user may be overwhelmed by having to adopt a new style of design. As such, the author recommends that a potential user of the design principles shared here adapts the model to the needs of their institution, design team, and ultimately, the students being served by them.

REFERENCES

Allen, W. (2006). Overview and evolution of the ADDIE training system. Advances in Developing Human Resources, 8(4), 430-441.

Boehm, B. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61-72.

Campbell, K., Schwier, R., & Kenny, R. (2009). The critical, relational practice of instructional design in higher education: An emerging model of change agency. Educational Technology Research and Development, 645-663.

Cannistraci, L. (2011). The value of instructional technology in a K-12 district. Distance Learning, 8(1). Retrieved from http://go.galegroup.com/ps/i.do?id=GALE%7CA284016023&v=2.1&u=csumb_main&it=r&p=AONE&sw=w&asid=2d4375b0464a7344ddf536ad607a96d0

Cook, D., Hamstra, S., Brydges, R., Zendejas, B., Szostek, J., et al. (2013). Comparative effectiveness of instructional design features in simulation-based education: Systematic review and meta-analysis. Medical Teacher, 35(1), 867-898.

Gedik, N., Hanci-Karademirci, A., Kursun, E., & Cagiltay, K. (2011). Key instructional design issues in a cellular phone-based mobile learning project. Computers & Education, 58, 1149-1159. doi:doi:10.1016/j.compedu.2011.12.002

Gorschek, T., Tempero, E., & Angelis, L. (2014). On the use of software design models in software development practice: An empirical investigation. The Journal of Systems and Software, 95, 176.

Khodabandelou, R., & Samah, S. (2012). Instructional design models for online instruction: From the perspective of Iranian higher education. Procedia - Social and Behavioral Sciences, 67, 545-552. doi:10.1016/j.sbspro.2012.11.359
Lott, C. (1997). Breathing new life into the waterfall model. IEEE Software, 14(5), 103-105.
Nunley, C., Bers, T., & Manning, T. (2011). Learning outcomes assessment in community colleges. Illinois: National Institute for Learning Outcomes Assessment.
Molenda, M. (2003). In search of the elusive ADDIE model. Performance Improvement, 42(5), 34-36.
Royce, W. (1970). Managing the Development of Large Software Systems. Proceedings of IEEE WESCON 26 (August): 1–9
Tennyson, R. D., & Michaels, M. (1991). Foundations of educational technology: Past, present and future. Englewood Cliffs, NJ: Educational Technology Publications.