Adapting the Agile Process to Digital Reconstructions of the Temple of Apollo at Delphi

Taylor D. Baldwin
Coastal Carolina University, USA. *tdbaldwi@g.coastal.edu*

Arne R. Flaten
Coastal Carolina University, USA. *arflaten@coastal.edu*

Abstract:
Ashes2Art is a multi-disciplinary, upper level undergraduate course and project that primarily focuses on accurate digital reconstructions of culturally significant architecture and archaeological sites. Digitally reconstructing the 4th century BC Temple of Apollo at Delphi, Greece, required the distinct analysis of both the knowledge domain and functional solutions. The primary, and often only, source for reliable data regarding the temple's reconstruction are the Fouilles de Delphes, archaeological reports published over the past century by the French Archaeological School in Athens with limited availability (although they are largely online now, some key issues are missing from online resources and the quality of the scans ranges from excellent to unusable for reconstructions). Coupling the research on the temple with the need to familiarize oneself with a robust, full-featured software suite presented a continual process of experimentation and revision. Relying upon personal experience with software development, the Agile Process was most aptly suited to this task.

The Agile Process attempts to mitigate an evolving scope by focusing on iterative, cyclical production coupled with frequent meetings and exchanges. The adaptation of the Agile Process to the creation of the temple model allowed for continual revision while still retaining a production ready model. By combining individual and cooperative work, the project benefited from quick, energetic progress tempered with careful cross-examination. In the final stages of the project, significant progress and accuracy were achieved by using an adaptation of the paired programming model. Rather than two members work jointly on a single workstation, as typical of the paired programming model, one member worked while overseen by an advising professor. The entire project presented an intriguing adaptation of the Agile Process to fit a cross-disciplinary project which showed remarkable progress while continually growing in scope.

Key Words: *Agile Process, Digital Reconstruction, Delphi, Ashes2Art*

Motivation

The process of digitally reconstructing an accurate model of the Temple of Apollo at Delphi presented significant challenges. These challenges broadly fell into four categories.

Accuracy and uncertainty

Any reconstruction undertaken in the Ashes2Art (www.coastal.edu/ashes2art/about.html) project has the explicit requirement of adhering to accuracy foremost. Accuracy, however, can be difficult to quantify and evaluate (Flaten 2009). Since all students

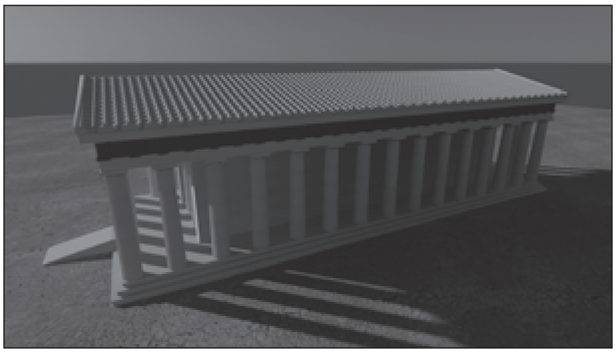

Figure 1. The Temple of Apollo of Delphi, Greece from 4th century BC.

Figure 2. The reconstructed paving of the temple entrance.

involved are both undergraduates and untrained in architectural principles any interpretation of plans and elevations should have been treated as dubious. In one instance, an early model of the Temple featured Doric columns was erroneously topped with circular abaci, based on one modeler's interpretation of a column elevation. A render of the model featuring the circular abacus accompanied a submission for journal publication, and the mistake was only realized after the journal had gone to press. The problem was corrected (Fig. 1), but the damage had been done. In order to prevent architectural inaccuracies resulting from uncertainty or otherwise, we desired to incorporate a continual, yet unobtrusive process of review.

Documentation

A consequence of an ongoing project, the initial collection of documentation on the temple was completed a year prior to when modeling began. The *Fouilles de Delphes* reports concerning the Temple were loaned to the program by various universities and relevant images were digitized for future reference. As there was much to scan and the students were uncertain what information would be useful, they scanned only those pages featuring obviously useful drawings and schematics and few pages exclusively containing text were scanned. As we modeled the temple, questions arose that our accumulated references were unable to answer. We needed to reacquire the reports, yet limited availability meant we would be idly waiting until a copy became available. We discovered online scans for some of the reports available at archive.org and cefael.efa.gr. However, due to poor scanning, many pages are illegible and in some cases entire volumes are missing.

Learning as we go

While one primary focus of the Ashes2Art course is digital reconstructions, no professors with extensive digital modeling expertise were involved or available (this is no longer the case). Unless a student had acquired prior experience in digital modeling, the student was expected to independently develop the ability to use the modeling software while creating the reconstructions. Rather than focus on one specific software package, students are encouraged to use whichever software they prefer. The Temple of Apollo was created using 3DS Max, Mudbox, Gimp, Blender, and several Adobe products. Without experience using the modeling software, the possibilities and practicalities of the reconstruction can be difficult to judge. In 2009, the temple was restarted on multiple occasions because uninformed decisions were easier to scrap and redo than fix. In order to maintain academic growth we needed to balance the natural process of learning with the desire to produce worthwhile contributions.

Structural

Without a clear process to follow, the temple was constructed *ad hoc*. The individual components of the Temple were created within the production model, rather than as separate components. When placing the acroteria we realized the dimensions would not work, forcing us to check our measurements. We noticed several components were of the wrong dimensions, eventually tracing the error back to the architrave. Each component above the architrave assumed the architrave was accurate. Regular verification would have caught this error in time. Instead, we lost time fixing each component that inherited the error. In order to isolate inaccuracies to individual components, we needed a clear process to abstract complex objects into separate components.

The overlap

Concurrently, a parallelism between software engineering and digital reconstruction began to emerge. Like the algorithms of an application, the temple features distinct components that follow specific rules for construction. Both applications and reconstructions follow a life cycle of design, creation, and then use. Given the overlap, translating a software development process to digital reconstructions should provide benefits that outweigh the costs of modification. Considering the structure of the Ashes2Art course, the process needed to accommodate small teams with diverse skills. Since each year brought new students, the direction, ambition, and experience changed regularly; we required a process suited for a chaotic environment (Flaten 2009). As some students participated over several years, the process also needed to allow projects to grow in scope and complexity. Of the software engineering principles, the Agile Process featured prominently as best mapped to our desires.

Implementing the Agile Process

The Agile Process features four common motifs that provide useful structure for digital modeling.

Working progress over complete documentation

Given the inevitability of change creating a complete design before commencing the work will likely result in wasted work as the design will inevitably need to be reworked (Nerur et al. 2005). This observation also holds true for digital modeling. When beginning the creation of a column for instance, one needs to know the order and the dimensions but not the paint color used on the adornment. Furthermore, how a model is created will likely determine how the model is then textured. By segmenting the full construction of an object, like a column, into less complex tasks, each task then receives individual focus. With each task receiving individual focus, the research and design of each task lessens and therefore each task becomes more manageable.

Frequent, collaborative meetings

Effective communication is paramount to the Agile Process and maintaining consistent open dialogue provides key benefits (Vliet 2008, 54-68). Digital reconstruction, most certainly in the case of the Temple of Apollo, is an amalgamation of archaeology, architecture, digital modeling, and computer science. As a team of individuals with different disciplines, an open dialog allows free exchange of knowledge necessary to fuel progress. While a newer modeler was attempting to recreate the interior Ionic columns, that modeler spent significant effort achieving a reconstruction with millimeter

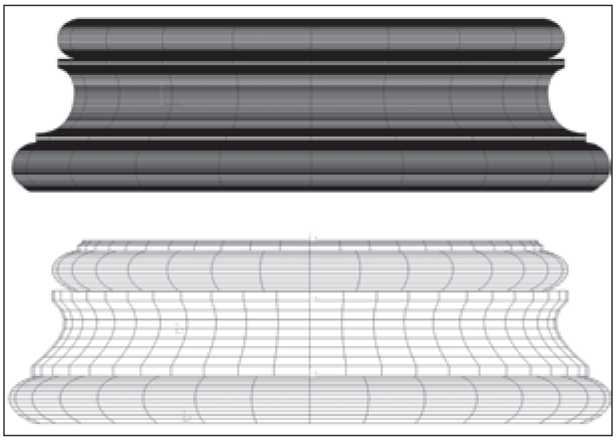

Figure 3. Two different designs offering similar effect.

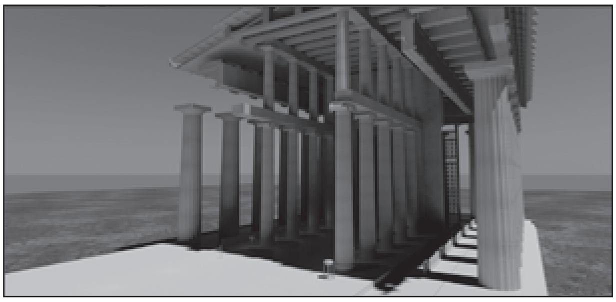

Figure 4. A cutaway featuring conjectured coffered ceiling.

precision. The model was smooth and accurate, yet so complex that its polygon count was half as much as the entire temple combined. Even with powerful workstations, the model became unwieldy. Closer, experienced supervision at early stages would have prevented this functional restriction from ballooning into an unusable component. With frequent meetings, the review process becomes an ongoing event hopefully preventing wasted effort.

Steps must also be taken to ensure openness with the dialogue. In an open environment, conflicting ideas may be reasonably discussed. Adding scrutiny to questionable decisions can refine those decisions or prompt the consideration of new possibilities (Nerur and Balijepally 2007). While the temple likely featured a cella, pronaos, and adyton, the archaeological and corollary evidence provide no clear consensus on arrangement or adornment (Courby 1927, Middleton 1888). We were forced to conjecture over which interpretation was most reasonable and would work best with our model. Creating rapid prototypes for each conjecture allowed the team to view each possibility in the context of the temple. Comparing different conjectures, such as the coffered ceiling and interior architrave, within the 3D space of the model provided a perspective that available reference materials were unable to deliver.

Spiral development model

The spiral begins in the center, representing only a basic plan and minimal details. As the model progresses, the accuracy (and therefore complexity) of the model iteratively expands outwards. Each rotation denotes a single cycle composed of four ordered phases: Analysis, Evaluation, Implementation, and Review (Vliet 2008, 54-68). In the analysis phase, the overall goal is formed and any relevant references are accumulated. In the evaluation phase, the overall goal is broken down to clear, discernible objectives. References are then analyzed based on relevancy, usability, and confidence levels. In the implementation phase, the model is constructed to match the stated objectives. The final phase, review, compares the outcome against the desired objectives.

The adapted spiral pattern was applied to the construction of the ionic columns, since that reconstruction had proven difficult in past experiences. In the first phase of analysis, we compiled relevant documents, including renderings of the column from the *Fouilles de Delphes*, photographs collected in Greece, and online images of ionic columns from a wide variety of sources. For the first evaluation phase, creating the base and shaft of the column were determined reasonable, leaving the capital for the subsequent phase. After being constructed in the implementation phase, the model was

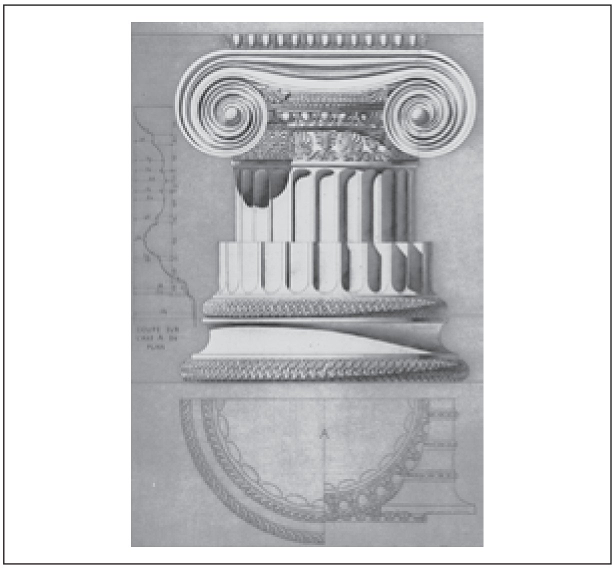

Figure 5. The reference document used for constructing the Ionic column.

Figure 6. A model view of the reconstructed Ionic capital.

reviewed for accuracy. While the base met the objectives; the shaft did not and was revised. Once the shaft had been sufficiently reworked, the Ionic capital was next and more references, mainly photos, were collected for the second analysis phase. The references (Fig. 5) portrayed an ornately decorated capital, featuring volutes that would be difficult to recreate. With the understanding that the adornment could be later added, the second evaluation phase established a geometric representation of the capital as an adequate step. The capital was segmented into five parts and in building those five parts separately four were quickly modeled. The fifth part, the spiraling volutes, required several approaches resulting in a lengthy implementation phase. Once assembled, the review of the model determined that the volutes needed further refinement before continuing. Several short cycles were devoted to achieving an accurate model. Later cycles added textures to the column, including using bump mapping to produce realistic adornment, and proper placement of the columns within the temple interior. Adhering to the spiral model by segmenting the reconstruction into separate, iterative steps reduced a previously overwhelming undertaking into manageable components.

Paired programming

With two programmers at one workstation, paired programming couples implementation and review. Having two perspectives increases accuracy and forces continual review possibly at the expense of rapid progress (Williams et al. 2002). In the Ashes2Art course, there are two specific instances where this trade-off proved worthwhile. By partnering a senior modeler with a new recruit, this paradigm creates a comfortable mentoring relationship. While the novice uses the controls, the mentor provides guidance at the novice's pace, offering suggestions and answering questions of proper methodology. While the mentor uses the controls, the novice is able to observe useful shortcuts and techniques that come from experience. Encouraging a dialogue of questions and explanations provides each team member with insights into the other's work. The novice is introduced to a tested workflow; the mentor gains a clear picture of the novice's ability.

Paired programming also allows a synergy by combining the domain knowledge of a professor with the functional knowledge of a modeler. The complexity of the temple occasionally prompted design questions requiring prolonged debate of architectural principles and functional ability (Flaten 2009). With the modeler in control, the professor is able to navigate a complex model without needing to know the software. From this perspective, the professor suggests potential projects and the modeler, in response, offers estimates of feasibility. As a joint effort, the pair can quickly weigh several potential prospects and ultimately make well informed decisions. Another advantage to this working pair is it affords the most intimate review of the modeler's work. Renders and movies can provide reasonable assurance of accuracy from the perspective of the modeler. In the unfiltered 3D space of the modeling software, shortcuts cleverly hidden (gaps in wall corners, etc.) from the render camera are obvious and unavoidable. While a very strenuous process, this review has revealed many inaccuracies and greatly enhanced the quality of the model. The overall success from using these initial adaptations of the Agile Process prompted the development of a clear set of goals to serve as a guide for future work.

Agile Manifesto

The beginning of the Agile Process is attributed to a convening of software developers in 2001. The meeting resulted in the publication of the Agile Manifesto, a document providing a cohesive structure of goals and practices for the Agile Process (Vliet 2008, 54-68 & Agile Software Development: http://en.wikipedia.org/wiki/Agile_software_development).

4 core values:

1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

12 Principles of Agile Development:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity – the art of maximizing the amount of work not done – is essential.

11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on

how to become more effective, then tunes and adjusts its behavior accordingly.

My revisions

The use of the Agile Process within the Ashes2Art course has been a work of intuition, experimentation, and reflection. Without an explicit framework, however, this adaption provides no lasting benefit. With a desire to create an applicable process more relevant to an academic environment, we require goals mimicking the Agile Manifesto. Therefore, we offer this revision of the manifesto as a guiding structure:

4 Core Values:

1. Individuals and interactions over processes and tools.

2. Periodic progress over complete designs.

3. Collaborative goals over product mandate.

4. Responding to change over following a plan.

8 Principles:

1. Provide a production-ready model throughout development.

2. Create a sustainable pace through smaller changes over shorter timescales.

3. Build projects around motivated individuals. Give them the environment, support, and trust to get the work done.

4. Better designs emerge from self-organized teams.

5. Open dialog best facilitates effective and efficient communication within and between teams.

6. Simplicity is essential.

7. Frequent, cross-discipline meetings enhance decisions and promote accuracy.

8. At regular intervals, the team must analyze, then tune and adjust.

Even though reworded, the proposed four core values maintain the spirit of the Agile process. The first value maintains as much benefit in academics as in a good business model, a focus on the growth and development of current students rather than teaching a certain skillset. The second value, as reworded, preserves the desire to work in spite of uncertainty or limited data. Most difficult to translate, the third value provides motivation for students by giving them direct influence in the outcome of the project. The fourth value, essentially the motto of the Agile Process, remains unchanged.

The revised principles reflect the shift from a business model to an academic paradigm. The first principle combines the overlapping first and seventh principles. Rather than the model itself, the final product of a reconstruction is the rendered photos and videos. As such, maintaining a production ready model enhances revision and provides useable content throughout the lifecycle of the model. The second principle combines the original, overlapping the third and eighth. The pacing of a project is measured through consistently meeting deadlines. By using smaller, more frequent checkpoints the pacing can be vigilantly monitored and adjusted to maintain sustainability. Principle three remains unchanged as motivated individuals drive the project forward, and sustaining that motivation is vital to success. Principle four naturally follows by reinforcing that a team knows itself well and will govern itself to the benefit of the project. Expanding on the need for effective communication, principle five adds the explicit requirement for open dialogue. Encouraging informed debate through open dialogue provides an effective means for collaboratively refining decisions (Neruret al. 2005). Principle six has been simplified in recognition of itself. In addition to preventing unnecessary work, we strive to avoid unduly complex designs and to learn principles over tools. Principle seven combines the fourth and ninth by utilizing continuous exposure to perspectives from each discipline to ensure each decision is well reviewed and appropriate. The last principle broadens the scope of introspection beyond the individual to include the modeling team as a whole. Agility comes from growing and learning as a team benefiting from each individual's experiences.

Future Work

The ultimate goal is to provide a coherent, working method for the digital reconstruction process. Using the goals set forth by the modified Agile Manifesto a process can be implemented in the Ashes2Art course and in other projects similarly concern with accurate, collaborative reconstructions. By observing, testing, and refining the process, the sound methods of software engineering principles can be applied to digital reconstructions.

Bibliography

Courby, M.F. 1927. *Topographie et Architecture: La Terrasse du Temple*. Paris: École Français D'Athènes.

Flaten, A. R. 2009. "The Ashes2Art Project: Digital Models of Fourth-Century BCE Delphi, Greece." *Visual Resources* 25 (4):345-362.

Middleton, J. H. 1888. "The Temple of Apollo at Delphi." *The Journal of Hellenic Studies* 9: 282-322.

Nerur, S., and Balijepally, V. 2007. "Theoretical reflections on agile development methodologies." *Commun. ACM* 50 (3):79-83.

Nerur, S., Mahapatra, R. and Mangalaraj, G. 2005. "Challenges of migrating to agile methodologies." *Commun. ACM* 48 (5):72-78.

Van Vliet, H. 2008. *Software Engineering: Principles and Practice*. Sussex: John Wiley & Sons Ltd.

Williams, L., Kessler, R. R., Cunningham, W., and Jeffries, R. 2002. "Strengthening the Case for Pair-Programming." *IEEE Software* 17 (4):19-25.