Design Principles to Improve Website Accessibility

John Stamey
Instructor, Department of Computer Science
Carolina Coastal University
P. O. Box 261954
Conway, SC 29528-6054
843-349-2552
jwstamey@coastal.edu
Thomas L. Honeycutt
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
(919) 515-7001
tlh@csc.ncsu.edu

Introduction

We live in a maze of silver bullets and werewolves (Brooks, 1995), mythical man months and fears of humiliation if caught violating Brooks' Law (Brooks, 1975), mental models of Tom Sawyer and partitionable / nonpartitionable projects (Johnson-Laird, 1983), heavyweight PMBOK/ANSI (PMI, 2000) standards and the like. Successfully navigating this maze of project management laws, paradigms and standards is a formidable task for large projects. Exactly where is Van Helsing and his kit of monster-destroying tools when those smaller "demon projects" suddenly appear out of no where, replete with a cloud of gray smoke and the smell of brimstone?

We have been aware for years that only a fraction of all projects are completed on time and within budget. This study was a measure of larger information technology. We must also be mindful there is an entire class of small to very small projects that also need to be completed successfully. We will examine a project management methodology for smaller projects that comes by way of quality improvement, providing a necessary element of structure without the document and process overhead of traditional larger project management methodologies. Agile methods, so successful in software, now have a counterpart in project management.

Characteristics of smaller projects to which this methodology has been applied include:

- They are projects in the sense that there is a defined beginning and end, and not a repetitive endeavor.
- They have basically a zero budget (employees are expected to get them done as part of their daily work), hence Earned Value Analysis is not appropriate.
- They are restricted by time (there is a specific deadline in which to complete the project), hence computation of a Critical Path is not an issue; perhaps there are only two nodes, START and FINISH.
- They must be completed correctly, hence quality is an issue.
- Accommodation of smaller projects with a different, lighter project management strategy fits well as part of Agile Methods

Examples of smaller projects that fit these characteristics include:

- Development of a new course to be implemented the next semester while teaching a full load
- Mounting a new Internet web server and transferring 25 domains for hosting (in a one-week timeframe) with no release from full-time duties

Learning from Quality Improvement

Two terms that have been borrowed from the Japanese are HOSHIN (a breakthrough innovation or dramatic change in level of performance), and KAIZEN (an ongoing refinement of process). The Hoshin process is, first of all, a systematic planning methodology for defining long-range key entity objectives, while not losing sight of the day-to-day "business fundamental" measures required to run the business successfully. Kaizen (Kaizen, n.d.) is a Japanese word meaning gradual and orderly, continuous improvement. Kaizen has (and must contain) two distinct parts: improvement/change for the better, and ongoing/continuity. As an example, the phrase "business as usual" contains the element of continuity without improvement. The expression of "break-through" contains the element of change or improvement, but is missing the element of continu-ity.

Eddlestone and Roberts (1992) have illustrated two cycles of the Plan, Do, Check, Act (PDCA) cycle sometimes called the Deming or Shewhart cycle for improvement (ISO, 2002), with one cycle representing Hoshin, and another cycle representing Kaizen. The Agile PDCA Project Management Model (Figure 1) is a variation of this approach.

Application of the PDCA Hoshin/Kaizen Approach

Implications for the use of this as a project management methodology are based on the following eight steps:

1. PLAN - Develop process innovation (project) plans
2. DO - Implement plans
3. CHECK - Check capability (test and measure the success of the implementation)
4. ACT - Act on results (decision point)
5. PLAN - Develop process improvement (project) plans
6. DO - Implement plans
7. CHECK - Check variation
8. ACT - Act on results (decision point)

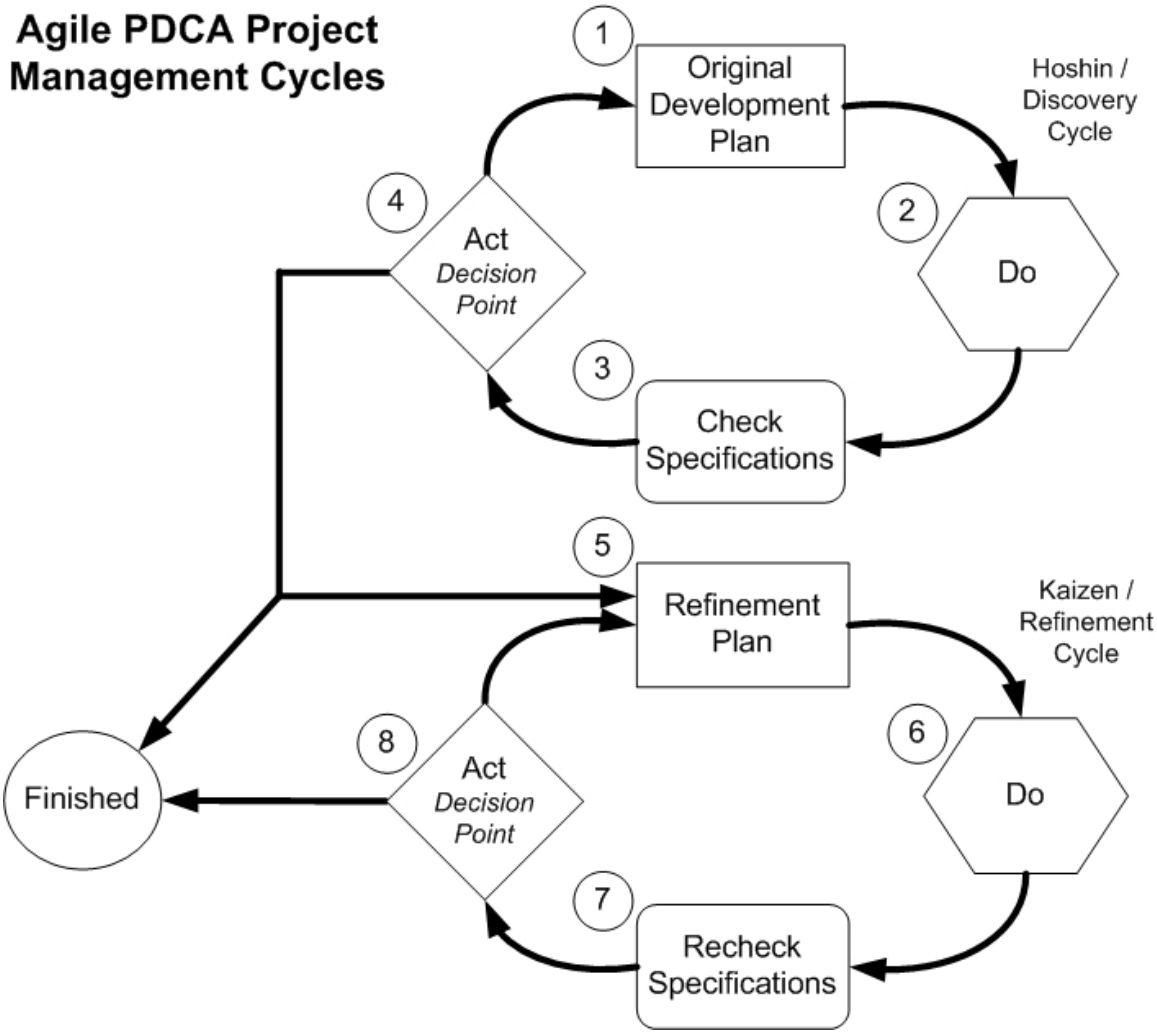

Figure 1. The Agile PDCA Project Management Model

It is to be noted that as we are dealing with a project that has an end, we have eliminated the path from Step 8 to Step 1, found in the original diagram. This feedback loop is used for the case when the process developed has become unstable and needs further breakthrough or development (Hoshin).

We will explain each of these eight steps as they are used in the practice of project management of small projects:

Step 1: Plans can be developed in a standard hierarchical outline form. It is also advisable that a second, parallel, plan be created in the form of a Pattern (as attributed to Christopher Alexander, 1975). The use of a pattern approach to describe a solution to small project implementation is basically a cross-check to validate completeness of the project steps.

Step 2: Implementation of a working prototype serves two key purposes: validation of the specifications given by the project owner, and opportunity for the developer to have a trial run at the

development of the final system, entirely consistent with Fred Brooks' "Build One to Throw Away" approach from the Mythical Man-Month Essays (Brooks, 1975).

Step 3: Test the prototype against the specifications. The project owner may now be involved to test usability and functionality. At this point, the prototype will be determined: finished; needing changes; or, needing redesign and redevelopment.

Step 4: If the project has been completed, then it is HIGHLY advisable to go to lower PDCA (Kaizen) cycle for further refinement. Usually these refinements come from input by the project owner, and do not constitute major changes or a redesign of the system. Otherwise, another upper cycle needs to be completed, with a Refinement Plan, new implementation, and new testing/validation.

Step 5-8: Patterned after Steps 1-4 above, with the Plan step now being a Refinement Plan, and the Check step actually a validation the project now meets the required specifications.

The Plan (Step 1) Written as a Pattern

A sample of a modern version of Alexander's pattern template, found in The Knowledge Enabling Organon (Honeycutt and Pichou, 2001) is found in Appendix A. Below is an adaptation for the development of a new service course at Coastal Carolina University (“CCU”), and was used with the PDCA Methodology for managing this project.

NAME. Computer Science Service Course

ALIAS. CSCI 105, CSCI 106, CSCI 120 (Other service courses taught at CCU)

CONTEXT. University departments must provide a variety of service courses that not only serve majors in other departments, but provide a relevant learning experience for their majors as well. They can successfully be constructed to be of high quality as a service course, while serving as a quality elective for the hosting department's majors.

PROBLEM. While "all courses are equal," some courses "are more equal than others." Any new service course must have a legitimate and coherent content, be well-regarded by students and other faculty who place their student advisees in the course, and have the respect of the faculty who will teach the course. A service course can not be an empty placeholder course that is subject to ridicule as the material is considered "trivial" or not of value (out of date, etc.). Furthermore, it may be necessary for faculty to have to learn material with which they are not familiar to be able to deliver the course properly. This pattern is intended to detail the CS Department's experience with providing such a course.

FORCES. All disciplines are relying (or beginning to rely) on technology for origination, data collection, communication, and general advancement. More courses of this type will be needed in the future, so their creation should not be a mystery.

SOLUTION. Technology, as required today by all disciplines, can either be taught by the individual departments (on a one-by-one, individualized basis), or by the CS department (with enough variation so that it is not a one-size-fits-all approach).

When each department teaches such a service course, they each have to find someone technically competent to teach the course as well as provide the facilities. While optimal, this is not always possible. When one department offers such course material, a more general course is offered to serve several departments' needs (unless, of course, there are enough students to offer one specific course for one specific department of school).

The training should focus on skills necessary for the advancement of the discipline. These skills cross most all academic department boundaries and include:

- Communication
- Collecting data for research (quantitative or qualitative)
- Analysis of that data
- Presentation of the results
- Instructors in the CS Department, while literate in their field, may need to build some skill sets to be able to effectively offer a service course that targets a particular department.

METAPHORS.
Net-Centric Skills: Those without this general set of skills survive very well in our net-centric world. Data collection through the Internet and research skills (through search engines, etc.) are now a requirement and not an option.
Analysis Skills: Both business analysts and university researchers must have the skills outlined in CSCI 110 to be competitive (get the jobs), and persistent (keep the jobs)

EXAMPLES. Following is an example of using this pattern and the value in its use. The example course selected in CSCI 110, a new course that has been planned with the principles that were used to create this pattern:

- Developing a separate service course (such as for business):
- Enough students (estimated 750 per year)will be available for a large number of sections
- A common body of knowledge is desired by the School of Business that is unique to the School

From previous experience with service courses, the CSCI Department understands the following issues:

- How to handle training for large groups
- How to schedule resources for training large groups
- Access to the resources to provide the training (equipment and rooms)
- How to select and use one basic text for many sections of classes
- How to have the course sections set up and ready in time for advising so that we can place students into sections for Fall 2004

- Having a person that is well-trained and knowledgeable in the content area develop the curriculum in conjunction with the requesting party
- Coordinating the delivery of the curriculum with the Information Technology staff
- Having one person that is knowledgeable in the content area to coordinate faculty, curriculum, scheduling, and communication with other departments whose students are taking the course.
- The need for some standardization mechanisms (such as a standard exam) to make sure the professors all cover the material that is necessary for students to successfully complete the course
- The need for the students to learn a common body of knowledge so this course properly prepares the students for future tasks
- Proper hiring and availability of tutors for the students

RESULTING CONTEXT. The combination of course material, students from one or several disciplines, instructors (who may or may NOT have any experience with the material), and a general structure leading to a standard exit examination will provide a successful experience for the students.

Conclusions

In general, agile processes value production more than plan-driven processes, which consider planning artifacts as "first-class." (Cohn & Ford, 2003) The Agile PDCA Project Management Model is an example of such a process, concentrating on activities and their consequences while providing needed structure for small projects with little or no budget, and very little time to be completed.

References

Alexander, Christopher (1975) The Timeless Way of Building. New York: Oxford University Press.

Brooks, Fred (1995) "No Silver Bullet," Mythical Man-Month, 20th anniversary
Edition. Upper Saddle River, NJ: Addison-Wesley.

Brooks, Fred (1975) "The Mythical Man-Month," Mythical Man-Month, 20th Anniversary Edition. Upper Saddle River, NJ: Addison-Wesley.

Cohn, Mike & Ford, Doris (2003, June) “Introducing an Agile Process to an Organization,” IEEE Software.

Eddlestone, S.E. and H.V. Roberts (1992). TQM University Challenge. Milliken & Company, Spartanburg, SC, May 10-16, 1992.

Honeycutt, Thomas L., Pichou, Beatrix, (2001). The Knowledge Enabeling Organon. Raleigh, NC: KEI, LLP.

International Standards Organization (2002) "Environmental Management - The ISO 14000 Family of International Standards: The ISO 14000 Model," Retrieved May 1, 2004 from http://www.iso.ch/iso/en/prods-services/otherpubs/iso14000/model.pdf/.

Johnson-Laird, P. (1983). Mental Models. Cambridge, MA: Harvard University Press.

Kaizen Institute (n.d.) http://www.kaizen-institute.com/knowkaizen/knowkaizen.html/, Retrieved May 1, 2004.

Kenyon, David A. (May 1997). "Strategic Planning With the Hoshin Process," Quality Digest. Retrieved May 1, 2004 from http://www.qualitydigest.com/may97/html/hoshin.html./.

Project Management Institute (2002). The Project Management Body of Knowledge. Newtown Square, PA: PMI.