

■ KENNETH E. KENDALL, Feature Editor, School of Business-Camden, Rutgers University

Extreme Programming in Practice: A Human-Valued Approach to the DSI Conference Management System

by Kenneth E. Kendall, Feature Editor

I am a strong believer that 'there are no bad people, just bad systems.' Rarely am I proven wrong in that belief. [Scott Sampson, Brigham Young University, 2004]

In the last few years, ecommerce entered the DSI world. Everything involving the DSI annual meeting is now done electronically. Paper submission, reviewing, tracking, notification of results, scheduling, submission to the Proceedings, you name it. It's all done without passing a paper from person to person. This month's column is about extreme programming (XP). Extreme programming is a development process that is more similar to prototyping than it is to the systems development life cycle (SDLC) or to object-oriented programming. As a methodology well-suited for building dynamic, Web-based ecommerce systems, XP is differentiated from other systems analysis and design methodologies because of its activities, practices, and values. The examples used here are drawn from my experiences interacting with the DSI conference management system which was conceived, developed, and implemented by Scott Sampson. I share this overview with you.

Scott Sampson may not have realized it, but his conference management system was developed using the principles and values of extreme programming.

As I am writing this article, I am getting ready to send the DSI annual meeting schedule to the printers, and I have an opportunity to reflect on using

the conference management system: what it looked like at this time last year, how well it worked, and how it can be improved. I won't elaborate on any of the problems that arose because the conference management system wasn't developed using traditional SDLC or waterfall approaches. Instead, we'll look at some of the successes.

Extreme Programming (often abbreviated as XP, perhaps to confuse all of us who use Microsoft Windows XP) is the practice of developing systems using good principles taken to the extreme. This can mean not only asking the customer to state requirements, but getting the customer to help develop the system.

Four Basic Activities in Extreme Programming

Extreme programming involves four basic activities: coding, testing, listening, and designing. Obviously, coding and testing need to be done no matter how you develop a system, but listening is very important in XP. I have worked with many developers and network administrators over the years. Although talented in other regards, some were just not good listeners.

My experiences with Scott demonstrated beyond a doubt that he was a wonderful listener. He assumed that that if a user was taking the time to ask a question about a potential bug in the program, there was a significant probability that that person was correct. Not every developer has the ability or in-



Kenneth E. Kendall

is a professor of ecommerce and information technology in the School of Business-Camden, Rutgers University. He is one of the founders of the International Conference on Information Systems (ICIS). He

is an associate editor for Decision Sciences and International Journal of Intelligent Information Technologies, and he is on the senior advisory board of JITTA. Dr. Kendall was named as one of the top 60 most productive MIS researchers in the world, and awarded the Silver Core from IFIP. Ken is a Fellow of the Decision Sciences Institute. Books include Systems Analysis and Design (6th ed., Prentice Hall) and Project Planning and Requirements Analysis for IT Systems Development (2nd ed.). Dr. Kendall has published in MIS Quarterly, Management Science, Operations Research, and Decision Sciences.

Kenneth E. Kendall
School of Business-Camden
Rutgers University
Camden, NJ 08102
(856) 225-6586
fax: (856) 424-6157
ken@thekendalls.org
<http://www.thekendalls.org>

sight to listen, which means that XP may work for some but not others.

Another basic activity of XP is designing. Those of you who submitted papers to the 2002 DSI Annual Meeting will realize how the process evolved. A front end (paper submission process) was written at Wake Forest University by some of their capable systems developers, but it became apparent that separate parts of the system could not pass data between the front end and the preference-based scheduling system due to the manner in which the front end allowed authors to identify themselves. There was no way to uniquely identify an author at that time.

So an integrated system was developed in 2003 by Scott Sampson, who worked with 2003 Program Chair Mark Davis of Bentley College to add many features to the system including a manuscript review tool, Web-based proceedings submission, and a number of data management tools to allow adding authors, changing titles and abstracts, and reporting corrections.

In 2004, I worked with Scott as he improved the system further by correcting minor problems and adding new features. Several of these features are not evident to you because they involve the report and conference management tools required to balance the paper presentation sessions and to keep the entire project on schedule.

Four Core Practices of XP

Four core practices differentiate XP from other approaches. They are short releases, the 40-hour work week, on-site customer; and pair programming. Let's use these practices to evaluate the conference management systems from an XP perspective.

Short releases means that the development team compresses the time between releases of their product. Whenever I asked Scott to add a feature, it was taken care of quickly. There was continual improvement. If a feature was added and I wanted it modified, Scott would turn it around in no time.

Forty-hour work week means that XP development teams purposely endorse a cultural core practice where the team works intensely together during a typical 40-hour work week. While I certainly don't want to compute the time it actually took to run the meeting, I must admit that we did take breaks, especially on Sunday. A day of rest got everyone reconnected with their families and gave a fresh perspective on any system challenges we were facing.

On-site customer means that a user who is an expert in the business aspect of the systems development work is on-site during the development process. While we didn't meet in the same room, we did meet virtually a lot. We used email, phone, and fax (sometimes all at the same time). How did they ever do this without the benefit of cell phones? I should add that working with Scott, in any medium, was always a pleasure.

Pair programming is an important core practice. It means that you work with another programmer of your own choosing. Well, I guess the lack of following this practice was an exception that proved the rule. Scott did all the Perl coding himself. While I didn't write any code, there were several times when I lapsed into computer lingo. I was often eager to just jump in and create my own reports, but I did not. Scott will tell you whether my natural bent toward computer-geekiness helped or hindered the process.

Four Values of XP

Most importantly, extreme programming contains the four values of communication, simplicity, feedback, and courage. Taken together, these values work to create an environment where both developers and businesses can be adequately served.

Let's begin with communication. Systems projects that require constant updating and technical design are especially prone to communication errors, breakdowns, or misunderstandings. We worked to communicate as much as possible as quickly as possible so that problems would be resolved rapidly,

accurately, and to everyone's satisfaction. At times, the Associate Program Chair, Sameer Verma, or the Proceedings Coordinator would jump in and offer an opinion in their areas of expertise. Christine Roundy at BYU was a critical part of the system because she could interpret human requests and take the necessary actions.

Simplicity is the second value of XP. Whenever a report was needed we tried to keep it simple. Whenever a human request was made, we attempted to adhere to a set of policies so that any one of us would be able to offer the same advice. Consistency and simplicity worked hand-in-hand here.

Feedback is the third basic value. Good, specific feedback that is useful to the programmer, analyst, and customer (in our case, the authors) can occur within seconds, minutes, or days. I hope we accomplished this, but I often noticed that people who requested changes (for example, asking that an author be added to a paper) expected the change to occur instantaneously as if the system were an OLAP implementation. When authors and track chairs took the time to imagine the possibilities, they soon realized that all requests took time to be verified for accuracy before we could post them to the schedule.

Courage is the fourth value enunciated in extreme programming. The value of courage has to do with a level of trust and comfort that must exist in the development team. I can safely say that I had the very best team any program chair could have in running this meeting, and, yes, it does take courage.

As you can see, extreme programming is a human-valued approach that stands in sharp contrast to an object-oriented approach to development. It has its flaws, but XP has its advantages as well. I hope this discussion about the development of the conference management system helps deepen your understanding of extreme programming and how it can be particularly useful in the development of Web-based systems.

ECOMMERCE, see page 30

NAMES IN THE NEWS

CAROL LATTA, Executive Director, Decision Sciences Institute



Robert J. Mockler, St. John's University, is co-author (with Marc E. Gartenfeld and Leor Sillman) of *Cases in Multinational Strategic Management: Latin American Companies*, which was published by Strategic Management Research Group in September 2004. The book is used to supplement cases in Mockler's strategic management texts. The cases were developed at St. John's University's Center for Case Study Development at its New York and Rome MBA campuses. Many of its cases have received international

awards and been published in other authors' textbooks. Also, in May 2004, Mockler presented "How the Treatment of Business Subjects in Drama Provides Lessons Useful in Business Management Training," at the 28th Annual Comparative Drama Conference in Columbus, Ohio.

mockler@stjohns.edu



Lawrence Schkade, Regents Garrett Professor Emeritus at the University of Texas at Arlington, recently received the 2004 Distinguished Re-

search Publication Award from the College of Business Administration. His winning article, "Estimating the Number of Potential Organ Donors in the United States," appeared in the *New England Journal of Medicine* (August 14, 2003). In the article, Schkade and his co-authors studied the size and composition of the national pool of brain-dead organ donors during a three-year period and, on the basis of these data, considered ways to increase the rate of donation.

schkade01@comcast.net

George Summers DSI Past President

George Summers (University of Arizona, retired), who served the Decision Sciences Institute as its second president, passed away on June 6, 2004, due to complications from leukemia.

"When I think of him, I recall his easy smile and the sparkle in his eyes," says Lawrence Schkade, who served as president from 1976 to 1977.

"George was a founding father of DSI, who help to craft the philosophy of DSI, and he exemplified the ideals of the Institute in his professional life," writes Schkade. "In part,

it was his genial manner, academic stature, and commitment to quality teaching that inspired me early on to become more involved in the Institute. Later, it was my great pleasure to participate in awarding the certificate that named him a Fellow of DSI. George was among the first group of distinguished members to named DSI Fellows. I know that many thousands students, colleagues and readers of his works have benefited from his many substantial academic contributions and leadership throughout his career. I think of the expression, 'Well

done, good and faithful servant,' as a fitting epitaph for his life."

In addition to serving DSI as president (1971-1972), George also served as program chair (1970), council member (1969-1970, 1974-1975), president of Alpha Iota Delta (1975-76), and was named a Fellow in 1977.

Condolences may be sent to Susan Summers (George's daughter) at 9162 Lost Trail Drive, Tucson, AZ, 85715. ■

Carol Latta, Feature Editor
Executive Director
Decision Sciences Institute
clatta@gsu.edu

ECOMMERCE, from page 15

In 2005, the Decision Sciences Institute will be using a new in-house conference management system developed and maintained by Andrea Williams, so this will be the last year DSI will use Scott Sampson's program. It was a great system to use, and Scott was

wonderful to work with. The lessons learned are invaluable. We all appreciate you, Scott, and the contribution you have made to the Institute. ■

References

- Kendall, K. E., & Kendall, J. E. (2005). *Systems analysis and design* (6th ed.). Upper Saddle River, NJ: Prentice Hall.
- email communication, Tue, 3 Aug 2004 23:44:27 UT.