

# Don't let the Pendulum Swing too Far

*“A methodology is a recommended collection of phases, procedures, rules, techniques, tools, documentation, management, and training used to develop a system.”*

*– Association for Computing Machinery*

*Author:*

Timothy F. Bixler

*Sponsor:*

MS Course ITEC-700, Information Systems and IT Staffing,  
Jill Klein Professor

*Date Published:*

December 12<sup>th</sup>, 2005

It has been stated earlier in this work, and rightly so, that unnecessary rules and regulations can stifle the creativity of an individual and prevent thoughtful work. On the organizational level, methodologies are criticized for being one-dimensional; in other words, they adopt only one approach to the development of projects that may well not address a particular organization's underlying issues or problems. There is an appropriate use for methodologies, however, and it could be argued that without such framework, much of the creativity we want from our programmers would remain untapped and job satisfaction would be lower.

Due to their misuse, the use of methodologies has been the topic of much discussion among information systems authorities in recent years. Two such authorities, David E. Avison of the ESSEC Business School of Cergy-Pontoise, France and Guy Fitzgerald of Brunel University, Uxbridge, Middlesex, U.K. said "Strict adherence to the methodology rule book has been described as slavish adherence to the methodology and the fetish of technique, that is, the methodology is allowed to inhibit creative thinking."

History has taught us, however, that methodology applied properly can actually enhance productivity, but more importantly to this work, enable creativity. The idea is that if all of the little mundane tasks that we need to do daily can be simplified then we could get on with our real work. In essence, you don't want people to struggle with the simple repeatable tasks, but to get on with the development of new strategies.

To this end, standards should be applied to those things that need to be done in a systematic way. Furthermore, the definition of a task and its parameters, leads to a greater success rate and greater job satisfaction. The simple act of creating a grocery list and checking off items that have been acquired is an example of how a general methodology can help to clarify the parameters of your task. This system clearly identifies the items you need to purchase and enables you to track your progress.

Historically, the pendulum has swung both directions – from lack of structure to too much structure. The software development of the 1960's and 1970's was largely done without formal methodologies and the results were less than satisfactory. The emphasis was on programming and solving technical problems and not on meeting end user and business requirements. Developers at that time were trained in computer technology but rarely understood the business or organizational contexts in which the systems were implemented. The approach programmers took to development was typically individualistic, often resulting in poor control and management of projects.

The recognition of these limitations led to a new appreciation of standards and a more disciplined approach to information systems in organizations. The result was the development of methodologies such as the Systems Development Life Cycle, or, more commonly, the waterfall model. The SDLC was developed for the mainframe computing environment and was essential to manage complex and time-consuming software development efforts.

Due to their earlier successes, these traditional methodologies were applied in the client-server environments of the 1980's and 1990's. They quickly proved to be inflexible, not allowing changes to requirements during development. This led to the creation of such techniques as Rapid Application Development (RAD), Accelerated Systems Development (ASD) and Computer Aided Systems Engineering (CASE) to address more aggressive market-driven release schedules. Up to this point, few techniques of organization recognized or addressed the critically important social, political, and organizational dimensions of development.

Since the late 1990's there has been more emphasis on targeting the feature set and creating an integrated development environment. This has led to the creation of Internet-oriented methodologies that enables developers and stakeholders to work together to prioritize requirements. Once such methodological best practice is called Joint Application Development (JAD), where all stakeholders in a project follow an intensive, structured process for defining and prioritizing the requirements for a project. Another is the Rational Unified Process (RUP) that contains six best practices of software development: manage requirements, control changes to software, develop software iteratively, use component-based architectures, visually model software and verify software quality. Today, there are new experimental methodologies such as eXtreme Programming (XP) that focuses on streamlining the overall analysis and development process

while still allowing for high quality solutions.

In order for these methodologies to be readily and universally available, common libraries of programming methods have been created. The idea is that once a method is created it is published into the library for other developers to use. Efficiency and that dreaded word, productivity, are key results here, for developers can leverage work that has already been done and reduce the time it takes for them to produce applications that are core to the business.

On a corporate level, a development team may use a central electronic repository for reference manuals, documentation, coded examples and other productivity enhancing materials. This level of collaboration is essential to a software development team's ability to communicate and produce quality applications. You will find that these standards are created with little effort because good developers understand the need to leverage the team's knowledge to make the project a success.

There are algorithms that computers understand and algorithms that they do not. A human is best suited for the job when the algorithm has not yet been programmed for the computer. For instance, imagine a manager faced with the challenge of forming a team of the right talents, skills and drive to accomplish a project. A common mistake is to think that a computer can provide a better way of doing something that is not already defined. They are only capable of increasing the capacity and accuracy in a given task. Yet, the risk for organizations not using any methodology at all should be recognized and the lessons of history not completely ignored. We are now in a new era of adaptable methodologies. Hopefully we will continue to find the center point of that pendulum.

## References

- Armour, Phillip (May 2003). In the Zone: The Need for Flexible Roles. *Communications of the ACM*. Retrieved December 3, 2005, from: <http://portal.acm.org/citation.cfm?id=769800.769815>.
- Avison, David E. and Fitzgerald, Guy (January 2003). Where Now for Development Methodologies? *Communications of the ACM*. Retrieved November 20, 2005, from: <http://portal.acm.org/citation.cfm?id=602421.602423>.
- Bloomberg, Jason (October 11, 1999). Software methodologies on Internet time. <http://www.developer.com>. Retrieved November 19, 2005, from: <http://www.developer.com/tech/article.php/616711>.
- DeMarco, Tom and Lister, Timothy. (1999). *Peopleware: Productive Projects and Teams*. New York: Dorset House Publishing Co., Inc.
- Glass, Robert L. (August 2004). The Mystery of Formal Methods Disuse. *Communications of the ACM*. Retrieved December 3, 2005, from: <http://portal.acm.org/citation.cfm?id=1012037.1012052>.
- Glass, Robert L. (November 2005). The Plot to Deskill Software Engineering. *Communications of the ARC*. Retrieved November 20, 2005, from: <http://portal.acm.org/citation.cfm?id=1096000.1096020>.
- Hardgrave, Bill C., Davis, Fred D., Riemenschneider, Cynthia K. (2003). Investigating determinants of software developers' intentions to follow methodologies. *Journal of Management Information Systems*. Retrieved November 26, 2005, from: [http://jmis.bentley.edu/articles/v20\\_n1\\_p123/](http://jmis.bentley.edu/articles/v20_n1_p123/).
- Jones, Bradley L. (May 9, 2001). eXtreme Programming. <http://www.developer.com>. Retrieved November 19, 2005, from: <http://www.developer.com/tech/article.php/761841>.

I don't know how broadly this story represents TJ's management policies, or how many of the cheerful crew members I run into are only feigning happiness. It should be noted that the chain gets very high marks from some employees "high enough to get it onto the Glassdoor Best Places to Work Employee Choice awards, which are based purely on anonymous employee feedback. Just like I don't know how accurate are the persistent rumors that Disneyworld is a less than magical place to work. . But stories like these sure put a darker spin on the whole "customer experience" thing. They also add a cynical Pendulum Swing Interpretations. On the one hand, an interpreter may crassly ignore human depravity and the necessity of Christ's saving work, reading the moral demands in this psalm as an indication that people are capable of obedience if they'll just put their minds to it, no grace needed. On the other hand—and this is so insightful—a "strict Lutherizing" may produce. Ortlund first instructs us to "let the full hortatory weight of this psalm land on us." He fingers the gospel-centered movement for sometimes finding ways to "squirrel out from under the moral instruction" of psalms like Psalm 15. We should not, he says, "prematurely apply comfort to our own hearts or the hearts of others for the many ways we do not live out this summons." The Freelancer- Work/life Balance is ðŸ™© - Don't let the pendulum swing too far. Discover 3 alternatives like The Freelancer - How to build a great portfolio and network and The Freelancer - Qualifying Clients. As with most things, if your focus swings the pendulum too far, for too long, there'll be consequences. We can't work 24/7 (not humanly possible) or even work all of our waking hours without sacrificing health, family and relationships. Upvote.