



The Menlo Institute LLC

The Value of Open Process Standards

Does your software development process have real value?

**Thomas Meloche © 2002
Fellow, The Menlo Institute**

**The Menlo Institute LLC
212 N. Fourth Ave
Ann Arbor, MI 48104
www.menloinstitute.com
(734) 665-1847**

Proprietary Development Processes

Software development organizations often realize, after a few spectacular failures, that they require a more formal approach to software development. Often they may assign one or two individuals the task of creating their own process and sometimes they go to large consulting companies to get help. Happy to oblige, many of these large consulting companies expend a great amount of time and effort developing their own internal proprietary development process which they offer as evidence of their programming prowess. Attempting to avoid failure is a laudable goal, but this approach suffers from multiple failure points. These include:

1. The uniqueness of the process.
2. A dearth of complimentary process documentation.
3. Limited training available on the process.
4. The narrow experience of the process developers leading to an incomplete process.
5. Lack of supporting tools.

This paper will review these concerns and propose an alternative to creating or using a proprietary development process.

Failure Point 1: The Uniqueness of the Process

A proprietary and internal development process is by definition non-standard and unique. It likely has its own vocabulary, rules and definitions that are not used anywhere else in the industry. Is it a good process? Who knows, as it has never been open to public scrutiny and evaluation.

A unique and private process is difficult to evaluate. Case studies are not public. Criticism is silenced rather than requested. I am aware of one large consulting company that had 35 people off by themselves doing product visioning and requirements for an e-Commerce exchange start-up. It was a waste. It was a bad process.

Proprietary methodologies result in constant special training and retooling as new team members are added. And because the knowledge is so specialized, those who know and understand the process often charge exorbitant rates. Words to heed: "Caveat Emptor" (or Buyer Beware). It is our observation that big consultancy processes seem more designed to spend your money than deliver your software.

Failure Point 2: Dearth of Complimentary Documentation

Proprietary processes have limited documentation. If a participant does not understand the process adequately from the training or the original set of manuals, there is nowhere outside the organization to go for help. Experience tells us that it often takes three or more exposures to new information before that information can be internalized effectively. Internal proprietary processes often only provide one or at best two exposures to that information.

Failure Point 3: Limited Training on Processes

A process proprietary to an organization also has limited training opportunities. It is a matter of record that internal training initiatives are always the first to suffer in budget allocations and corporate restructuring. Also, since training is an expense that doesn't generate direct revenue, it is typically one of the least respected internal initiatives. Proprietary internal processes suffer from a lack of formal training options. Typically there is only one class or curriculum to choose from to learn the process and if it is poorly developed and conceived there are few remedies.

Failure Point 4: Narrow Experience of Process Developers

The narrow experience of the process developers results in fatally flawed processes that are incomplete, inconsistent, and do not fit the needs of many projects the teams must deliver. Large organizations do hundreds to thousands of different development projects over the life of a development center. A process developed by a small group of internal process developers is not likely to be large enough in scope or vision to encompass a large organization's needs.

Failure Point 5: Lack of Supporting Tools

Internal proprietary processes often lack the supporting tools necessary to automate and implement the process. Therefore externally provided tools must typically be used. Since they were developed without the knowledge of the proprietary process, they are often a poor fit for the prescribed process workflows. Of course, if the tools are developed internally, they will likely suffer from the narrow experience and incomplete process on which they are based.

As a result of these limitations, proprietary processes often end up partially implemented, ignored or abandoned. It is reasonable therefore to conclude that these types of processes should not be the choice for your organization.

Open and Standard Processes

Fortunately, there are ready alternatives to proprietary development processes including, The Rational Unified Process® and Extreme Programming. These processes are open, published and supported standards. We feature a more detailed description of these processes in our papers: The Rational Unified Process® (RUP) and Extreme Programming.

If you are interested in learning more about these development processes consider attending one of the process overview classes provided by The Menlo Institute. We have extensive classes such as Managing Projects with RUP-Lite and Introduction to Extreme Programming that may be of interest. A list of all classes are outlined on the next page.

About the Menlo Institute

Founded in 2001 by Thomas Meloche, Richard Sheridan, James Goebel and Robert Simms, The Menlo Institute specializes in Software Development Process and Methodology with a special focus on an agile software development environment called The Software Factory.

The Software Factory

The Software Factory is a complete agile software development methodology created at Menlo. The Software Factory combines best practices from Alan Cooper's Interaction Design, Kent Beck's Extreme Programming, The Rational Unified Process and Six Sigma. The Menlo Institute has unique insight from real-world experience on how these practices can work together effectively.



**Thomas Meloche
President
The Menlo Institute**

Thomas Meloche is President of The Menlo Institute LLC and a founding member of Menlo Associates LLC. His passion is software development best practices and building successful software teams. Before founding The Menlo Institute, Thomas was the Director of Engineering at AppNet, Inc. For over seven years, he led a consulting staff that grew from

30 people to over 240 people which generated revenue of over 100 million dollars. As a result, he is intimately familiar with the real problems that befall real software development projects. He also knows how to rectify them. During his tenure his delivery organization had a remarkable 100% successful delivery record.

Thomas has a B.C.E. in Computer Engineering from the University of Michigan, and especially enjoys providing training back to his alma mater. He has taught classes and provided mentoring for both the University of Michigan Law and Medical Schools.

Menlo Institute Classes by Category		
Extreme Programming (XP)	Foundation Courses	Rational Unified Process (RUP)
Introduction to XP Coaching XP Teams Writing Story Cards XP BOOT Camp / Immersion	Secrets of Software Success Six Sigma Software Building a Software Factory Object Technology Overview	Managing with RUP-Lite Capturing Reqs With Use Cases Practical UML RUP BOOT Camp / Immersion
Requirements Gathering	Object Technology	Project Management
Capturing Reqs With Use Cases Writing Story Cards	Object Technology Overview Obj. Oriented Analysis & Design Advanced Obj. Oriented Design	Managing with RUP-Lite Software Project Mgmt Six Sigma Software