OOPSLA 2001

Mark Windholtz, ObjectWind Software

In this report …

Process	Agile Is In! XP to RUP and back again. Everything is Agile
Architecture	.NET and J2EE
Technology	Java, C#, XSLT, AOP, O-R Mapping, Meta-Matters

Conference Quotes:

- Objects are dead. Just let go. –Don Box
- Success leads to failure – Henry Petroski
- Turbulence dominates economics – James Highsmith
- I've never seen a methologically successful SEI or ISO9000 project –Alistair Cockburn
- I got into this industry for the babes – Ron Jefferies

This report will step through my daily sessions and then provide some overall impressions at the end of the report. OOPSLA 2001 had 1321 attendees from 36 Countries half were developers, the rest were users, customers, management and academics. 1/3 were at OOPSLA for the first time. It was my 6th consecutive visit to OOPSLA.

Sunday 14-Oct-2001

Don Box: XML and C# .NET

In a series of two tutorials Don Box first discussed XML technology and standards and then discussed Microsoft's response to XML in the .NET architecture and C# language.

XML is making a big step forward with the completion of the **XML Schema** specification. With XML Schema: DTDs are replaced and much of SOAP is no longer needed. Most people currently consider XML to be merely a good way to

connect various legacy systems. The thought is that data can be represented in XML and manipulated and transformed in various programming languages. Box takes the idea further asserting that with the full and rich type system provided by XML Schema coupled with transformation tools like **XSLT**, the whole need for representing programs in Objects become old-fashioned. So Box asserts: "Objects are dead", so we should all "Just let go" and move to the next paradigm.

The second tutorial, after half-convincing me that Object languages were in peril, was a tutorial on the next Object language C# and the .Net architecture to support distributed objects. The irony just now occurs to me.
Anyway, .NET strengths include: versioning, type extension, reflection, threading, and message redirection. .NET supports many languages (not just OO Languages) on a **Common Intermediate Language** (CIL). C# is the path of least resistance for current .NET work. I've hear various people claim that C# was "Just Microsoft's answer to Java". C# seems to me to be much more. C# appears to have more power and consistency that Java but C# remains more complex than Java. Not much was said about VB 7 although it also runs on .NET. VB7 may actually be Microsoft's real answer to Java. We'll have to wait and see.

References:

- XSLT Quickly by Bob Ducharme
- Microsoft .Net
- Introducing .NET by David Platt

Books by Don Box

- Essential .NET: The Common Language Runtime (Volume 1)
- Essential COM
- Essential XML : Beyond MarkUp

Monday 15-Oct-2001

Kyle Brown: Patterns and Architectures for J2EE Systems

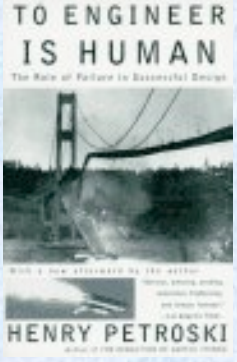
Kyle Brown separates architectural layers into: Presentation, Control, Business, and Infrastructure. And provides patterns for implementing each of these in turn. The patterns he presented were very useful and clearly presented. Rather than typing them all in here, for details I'd refer the reader to his book

References:

- Enterprise Java Programming with IBM Websphere by Kyle Brown
- Martin Fowler's Information System Architecture
- Core J2EE Patterns: Best Practices and Design Strategies by D.Alur & D.Malks

Tuesday 16-Oct-2001

Keynote: Henry Petroski: "Success and Failure in Design"



Engineering is pro-active failure analysis.

The very first engineering book around 100 B.C. talked in detail about failures.
It included concerns about how to move stones and ideas about moving stones that failed. He also covered a short history of suspension bridge building and the successes and failures in that field.
Typically, when a successful design is pushed, some phenomenon emerges that was not important in the small previous implementations. In a way, success leads to failure. I suspect this is the more general case of Fred Brooks' well-known **Second System Effect**. Petroski explains that bridge building failures seem to occur in 30-year cycles. He cautions that engineers must understand the range of applications for a design and not use the design outside of that range. I'd point to the constraints on a pattern as a way to document the understood part of that safe range.
It is only by looking at things in the context of failure that successful designs evolve.

This of course plays right into the hands of XP advocates who insist that no production code should be written without first writing a **failing** test.

Books by Henry Petroski

- To Engineer Is Human : The Role of Failure in Successful Design
- The Evolution of Useful Things
- The Book on the Bookshelf

Panel: Choosing the Appropriate Process

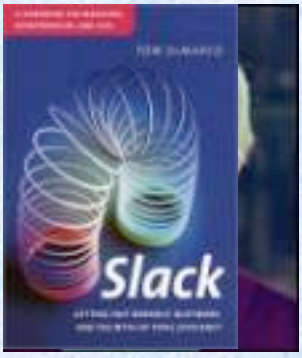
Comments from the Panel: (**Warning**: I tried my best to be accurate, but I could have written some of these comments down wrong. Always consult a trusted financial advisor before investing money based on these notes.)

AC: Alistair Cockburn.
IJ: Ivar Jacorbson
JH: Jim Highsmith
RJ: Ron Jefferies
CL: Craig Larman
BM: Bob Marcus

- I've never seen a team of 100 people ever do *anything* useful – IJ
- You can't have one process for all sizes of projects not even RUP – AC
- Any process needs agility to create and respond to change – JH
- A process without tool support is just an academic idea – IJ.
- Advanced programming tools are needed, process tools should remain simple –RJ
- You should have less process than is really necessary – JH.
- US and Canada both tried to replace their Air Traffic Control Systems. Canada succeeded by using an iterative approach. US tried to do it all at once and failed. – CL
- Most common mistake in using RUP is to try to use everything – IJ
- Adopting RUP must be done in a iterative fashion. – IJ
- Climbing the CMM ladder often leads to a loss of integrity between the documented process and the real work practices. - BM
- Alistair was surprised when reviewing an XP team he found that the documented practices were actually being done by the programmers. – RJ
- RUP is a bunch of best practices. Every process is an instance of RUP including no practice at all.
- I've never seen a methologically successful SEI or ISO9000 project - AC
- I got into this industry for the babes –RJ
- A CMM auditor had told me that he was sure an XP team COULD validate at Level 3. However, no team has ever tried -RJ

There was consensus that iterations, Agility, and improvement were necessary in process implementations.

Tom DeMarco "Software: The New Realities"



Slack is a degree of freedom in time, manpower, budget, etc that makes

change possible. Business now needs to be less efficient and more agile this requires more slack. *Kent Beck will be more important in this decade, than Watts Humphrey was in the last decade.*

Everyone is feeling busy. Busy feels productive but is not always so. Increases in productivity require that we reflect, experiment, and do non-optimal exercises in order to find ever better ways to work. These increases in productivity are hard to find in companies and cultures that fill all available time with defined tasks. What is needed is slack. Blindly cramming more tasks into less time is not a way out. The way out can only be found by honest prioritization of tasks. Currently, schedules are missed and projects lengthened because of a missing disciple of prioritization. To prioritize companies must "Grow-up". They must learn to say "no" to less desirable projects and tasks, so that they can adequately manage the highest priority ones. (This exactly is what the XP Planning Game accomplishes). In other words, the key to going fast is to choose what not to do. Death March projects (well described in Ed Yordon's Book) are usually of stunning *insignificance*. If anyone really had wanted them to be successful, the projects would have gotten adequate funding. Death March projects are not about time, they are about money. They have low budgets because nobody really thinks they are important. If adequate prioritization had occurred they would not have been done at all.

Tom DeMarco invented the early Data-Flow-Diagram techniques. But was soon appalled that people were using them to write huge specification documents. Process building is terminally additive. Processes just get bigger and bigger. The agile movement and XP is a backlash to this unhealthy trend. XP is subtractive.
Inspections are killing the software industry, and Pair Programming works because it does review *without boring everyone to death*.

On War by Carl Von Clausewitz considered the changing balance between Armour (defense) and Mobility throughout history. Medieval knights had the advantage of Armour defense until the fast archers of Genghis Kahn beat them with mobility. Today heavy Processes are defenses. XP is mobility. In the past it was said: only the fit survive. **Today, Only the Fast will survive.**

Prescription for a new era:

1. Become less efficient
2. Reduce Processes, Increase Skills
3. Learn to prioritize
4. Choose what not to build
5. Invest in Human capital.

Books by Tom DeMarco

- The Deadline : A Novel About Project Management
- Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency
- Peopleware : Productive Projects and Teams

Panel: How do Requirements Relate to Objects?

Here I watch the few remaining Big Design Up Front (BDUF) folks completely misunderstand the Agile folks. It is interesting to see that some very intelligent leaders in the OO movement still do not at all grasp the very simple ideas behind Agile design and process. Do these smart people not understand? Or do they really see situation where Design Up Front continues to work well? If the later, why are they having such a hard time communicating about the places where BDUF works?

Birds of a Feather: Eclipse Project

I only attended the end of the BOF on the Eclipse project. This is unfortunate because it seems like such a big and important move for the Java world. IBM has moved VisualAge for Java onto a CVS source code system, added refactoring tools, and is about make the whole thing open source! As Microsoft is about to provide a common execution environment across languages, IBM is about to provide a common development environment across plugable tools.

Wednesday 17-Oct-2001

Speaker: Don Box: "Pervasive XML: Infoset-Based Software Integration"
Pretty much the same as the tutorial reviewed above but shorter and less convincing.
"Objects are Dead. Just let go" yadda yadda yadda… Very entertaining. He has an amazing amount of guts to write live code examples in front of 2000 people.

Joe Yoder: Patterns for Making Business Objects Persistent

This tutorial covered concrete examples of how to build an O-R Mapping framework in Java. And gave insight into the many issues that surface when bridging this chasm. An important tool in making the mappings is the Meta Level facilities in Java and in Relational Database systems.

O-R Mapping References:
www.joeyoder.com/Research/objectmappings
www.joeyoder.com/Research/Frameworks
www.joeyoder.com/papers/patterns
www.joeyoder.com/papers/patterns/Reports/

http://st-www.cs.uiuc.edu/users/droberts/evolve.html

Crossing Chasms
www.ksccary.com/article5.htm
www.ambysoft.com/mappingObjects.pdf
www.objectarchitects.de/arcus/
www.joeyoder.com/Research/metadata

Distributed XP Tool Demo

University of Calgary milos project demo showed a tool for conducting distributed XP projects. The whole idea seems dangerous but it's interesting to see a group trying to push the boundaries. Perhaps eventually tools like this and others may provide a way to better coordinate distributed teams. For the present I'll maintain my skepticism.
http://sern.ucalgary.ca/~milos/milos/components.htm

Thursday 18-Oct-2001

Alistair Cockburn "Harnessing Convection Currents of Information"

Q: What does the act of developing software consist of?
A: Nothing but making ideas concrete, in an economic context.
We have a problem, a language, and a solution. All of which are changing as we work.
Keeping up is a matter of dealing with the flow of information around us.

Software development is a cooperative game of invention and communication.
Players in the game are people. People are highly non-linear, spontaneous, active devices. People are weak at consistency, disciple, and following instructions. People are strong at: looking around, taking initiative, coping and modifying, and communicating. People want to be good citizens and take initiative in solving problems.

In the software development game…

1. Primary goal is to deliver the software.

2. Secondary goal is to set-up for the next game.

The speed of a project is determined by the rate of the information flow between people. The office floor plan effects development cost calculated to be $50000 per minute for a typical 6 month project, in the time it takes to get an answer from the right person who might be next door or down the hall. These numbers are backed up by rigorous measurements done in Managing the Flow of Technology by Thomas Allen in the 1970's. And it turns out the richest, cheapest flow of information is accomplished by the practice of Pair Programming. Alistair proposes calling the concept of sitting together a more impressive name like: *Radical Co-location* in order to get people to pay attention to this important concept. (XP calls it Open-Workspace).

He uses the metaphor of convection in information flows. People get information by walking around. They are in the flow of information. There are Information radiators like useful and up-to-date charts. And information drafts, like noise from a nearby project interfering with communications.

Projects can lower the cost of invention by Goal Alignment, Increasing Amicability (good will), Community, Skills, and Talent.

People are more important than Process for productivity, but Office Politics is more important than either. Ultimately, the speed of a project is determined by the rate of the information flow between people.

Books by Alistair Cockburn

- Agile Software Development
- Surviving Object-Oriented Projects
- Writing Effective Use Cases

References:

Managing the Flow of Technology by Allen, Thomas J.

Summary Impressions

I was amazed by a number of things at this OOPSLA.

One was the number of Graduate students who claimed to be doing work in the area of Aspect Oriented Programming (AOP). I talked to three graduate students all starting work on AOP. At the same time I talked with a few reputable and long time OO practitioners who claimed that AOP is just a complex way of doing what good OO design could do for you in a simpler way. My worry is that AOP may become a distraction from people learning about general Meta-Level development and focusing exclusively on AOP with all it's whiz-bang tools. I like AOP and have followed AOP since 1997. AOP is a good friend of mine. But as an industry practitioner, I have deep concerns that it may remain too complex for more widespread and practical

usage. Standard meta-level facilities are already more widespread (if still underused). The .NET architecture will bring more of this power to standard development platforms. So, AOP may just become a pleasant distraction.

It is sooooo cool to be Agile. And there is no price not to be! So the effect is that of course every Methodology, process, development tool, technique, and Consulting company is calling itself Agile. I am amazed how quickly this Buzz-Word caught on, and nearly fell from my chair when Ivar Jacobson himself claimed that RUP is a good example of being Agile. *Agile*as an umbrella term is quite useful in setting the direction for what we all consider to be *goodness* in a process. I am very happy that there remains a distinction in Agile in general and XP as a defined set of Values and Practices. It's good that everyone is learning to want to be *Agile*. Perhaps, someday they may discover that to actually *be Agile* requires a few of the things XP has been talking about for so long.

Hope you found this as interesting as I did,
Send your comments to: Mark@Objectwind.com

-Mark Windholtz
www.objectwind.com Cincinnati, Ohio; Oct 2001