Panel: Cracking the Software Paradox

John Daniels, Syntropy Limited, co-moderator
Jim Amsden, IBM
Larry Constantine, Constantine & Lockwood
David E. DeLano, AG Communication Systems
Martin Griss, Hewlett-Packard
Ivar Jacobson, Rational
Else-Marie Östling, Icon MediaLab
Rebecca Wirfs-Brock, Wirfs-Brock Associates, co-moderator

Introduction

Can we develop software faster, cheaper, while remaining in control of quality? The advent of the web and the e-business phenomena mandates that scaleable, highly reliable, user-centered software be developed in a very short time. In this new world, consumers won't accept clunky, clueless, brittle software. We've got to get it right the first time and then be able to rapidly change it to meet new market demands. This panel explores how development processes and tools need to be tuned, tamed or reframed in order to crack this development paradox and deliver quality software in e-time.

Each panelist brings a unique perspective on the nature of the problem and proposes some solutions. The key to cracking the development paradox is multi-faceted. It involves changes to tools, technologies, platforms, and fundamental ways we develop software. This panel will be conducted as a roundtable. Each panelist will have an e-time moment to make an introductory statement, which will then be followed by fellow panelist and audience reactions. Panelists will explore ideas in rapid response to a changing set of audience and fellow panelists' demands!

John Daniels

John is a consultant at Syntropy Limited, working on system architectures and development processes. He was previously Application and Technical Architect for Bankers Trust in London, and before that Managing Director of consulting and training company Object Designers Limited. John is an object technology pioneer, with more than 15 years experience of object modeling and implementation in a range of industrial and commercial applications.

Ivar Jacobson

Ivar is vice president of business engineering for Rational Software Corporation. He is the founder of Objectory AB in Sweden, which merged with Rational Software in 1995. He was one of the three original designers of the UML, which was officially adopted as a standard by the Object Management Group in 1997.

We have heard it many times: software needs to be developed faster and at the same time with higher quality. The difference today is that in e-business, software is not just a tool for the business. It is the business! And since the business changes quickly and has to always work well, the same applies to our software. This time it is serious. We need to develop faster and better than ever before—we need to crack the software development paradox.

The solution to our development paradox is obvious:
1) Let every member in the development team know what to do, when to do it, and how to do it effectively. This implies a process accompanied by integrated tools that address the whole development lifecycle.
2) While doing it, don't reinvent the wheel. This means developing for reuse and development with reuse. Reuse of all kinds of assets: requirements, analysis, designs, implementations, and tests.

This time we are on the track to crack it!

Martin Griss

Martin is a Principal Laboratory Scientist at Hewlett-Packard Laboratories, Palo Alto, California. At HP since 1982, he researches software reuse, object-oriented development, component-based software engineering, software agents and e-service management systems. He is co-author of the book "Software Reuse: Architecture, Process and Organization for Business Success."

Pervasive web-based mobile appliances, agents and services will have a radical effect on the way we develop novel e-commerce solutions. We need rapid, systematic processes to construct high-quality, efficient, and scalable software systems. E-service middleware, components, autonomous agents and workflow promise flexible, robust systems. The key is much more systematic reuse of architectures, frameworks, patterns, components, agents, classes and aspects. Incremental, model-driven, processes will incorporate architecture, design and feature-oriented domain analysis to define reusable models, frameworks and components, and aspect-oriented technology to create and customize compatible components and agents using aspect-oriented languages, generators, templates, and code. UML and XML provide compatible representation languages, for design, construction and configuration, and meta-data driven execution.

Else-Marie Östling

Else-Marie is Chief Technical Officer at Icon MediaLab. Icon MediaLab provides internet, intranet, extranet, e-commerce and wireless strategies and solutions to clients for business-to-business, business-to-consumer and business-to-employee projects. She has over 15 years of experience in methods and process development, with an emphasis on object technology.

Everyone in the eBusiness solution space deals with the same paradox: more aggressive time-to-market needs, and yet more insistent requirements on scalability, and reliability. Currently we can address these challenges with "generic" components that are readily available. Nobody builds a custom commerce engine today; instead we make use of reliable, existing components, whether it is a B2B or B2C site. However, as eBusiness solutions become more customized to vertical industries, it will be more difficult to find appropriate pre-built "generic" components. To crack the paradox we need to re-address how we identify, specify, design and build components, such that they can more easily be made "generic" and reused across projects.

Jim Amsden

Jim is a Senior Software Engineer at IBM. Jim has been involved in developing and using software development methodologies and tools that support them since 1988. He is currently responsible for helping develop and deploy a tool integration strategy to enable ISVs to integrate their tools with IBM products in order to facilitate end-to-end application development.

Developers are faced with conflicting demands for high quality applications built in Web time using unstable technologies. Web application design patterns help manage this complexity and indicate the required tools. Tools need to be integrated and manage the relationships between their associated artifacts. A tool integration platform would provide the required "integration-ware" enabling the development of products consisting of integrated, configurable, roles appropriate, and end-to-end development tools based on best practice design patterns. To reduce acceptance barriers, this integration-ware must be based on open standards as much as possible and Open Source where standards don't exist. To be effective, tools need to be integrated into a common workbench. The workbench and integration-ware must support flexible tool integration appropriate to vendor business objectives by providing various conformance levels of integration. Conformance levels define what any vendor must do to integrate at a level, what technologies are required, and any integration expectations.

Rebecca Wirfs-Brock

Rebecca, co-inventor of Responsibility-Driven Design, is president of Wirfs-Brock Associates, a firm specializing in the transfer of object analysis and design expertise through consulting, training and mentoring. Rebecca co-authored the classic book "Designing Object-Oriented Software."

The only way to crack the development paradox is to approach development and design differently than we do today. I suggest we divide our work into core, mundane (the 80% of the work that consumes 120% of our time), and revealing. Core elements are vital to our success. They can include the "ilities"-reliability, scalability or portability. Solutions for these "ilities" won't emerge on their own-to crack them we need tunable, pre-specified designs and pre-built components (design solutions in-a-box). We need tools and methods that increase productivity for the bulk of our mundane work. We must spend more quality time developing solid cores while exploring solutions to revealing problems. Revealing problems are hard because they cover new territory. They require creativity, insight… time to think and experiment. To help solve revealing problems we need a development environment and process that encourages and supports highly creative, non-predictable side-excursions, while at the same time getting the rest of the work done.

Larry Constantine

Larry is a pioneer of modern software engineering methods and practices whose career spans nearly 40 years and whose contributions in the computer and social sciences are widely recognized.

David DeLano

David is currently working on Voice over IP products at AGCS, which has a long history in the telecommunications business.

In this new world, traditional development companies will be forced into acting like small, agile companies. However, there is a legacy of expectations to maintain—quality, robustness, feature content, and support. Development must occur in smaller pieces, by fewer people. Processes must become less formal. Quality expectations will be tempered in favor of time to market and cost. Communications between team members becomes critical. Tools used in the past no longer suffice. Individual expertise becomes a critical project resource. In the end we have to be careful not to become too informal. The trick is to walk the fine line that separates control from chaos.