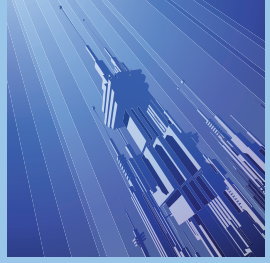
Speakers



Jim Coplien

Jim "Cope" Coplien is the father of Organizational Patterns, one of the founders of the Software Pattern discipline, a pioneer in practical object-oriented design in the early 1990s and is a widely consulted authority, author, and trainer in the areas of software design and organizational improvements. He is currently a partner in Gertrud & Cope in Denmark, and in the Scrum Training Institute. His work ranges from programming and performance evaluation, to organizational improvement assessments, organizational development, and process improvement, and to consulting on system architecture. He regularly speaks at conferences and seminars around the world on Lean Architecture, Scrum fine-tuning, Patterns, and Agile software development and project management practice.

www.gertrudandcope.com

Neal Ford

Neal is an application architect at ThoughtWorks, a global IT consultancy with an exclusive focus on end-to-end software development and delivery. He is also a designer and developer of applications, instructional materials, magazine articles, courseware, video/DVD presentations, and author of various books including The Art of Java Web Development and the 2006 No Fluff, Just Stuff Anthology. His primary consulting focus is the building of large-scale enterprise applications. He is a regular speaker at various conferences worldwide.

www.nealford.com

Dave Wheeler

Dave is a software consultant specialising in Microsoft UI technologies, fondly known by his Rock Solid Knowledge colleagues as the "Colouring in Guy". He's particularly fond of WPF and Silverlight, but to remain grounded in reality he also works extensively with ASP.NET and is a moderator on Microsoft's ASP.NET forums. When not writing software, Dave writes and teaches various .NET courses for DevelopMentor, writes articles and speaks at various conferences.

rocksolidknowledge.com

Allen Holub

Allen is a software design consultant, programmer, trainer and author, specialising in OO analysis and design, patterns, UML, Java, and Agile process. He is a contributing editor for JavaWorld, and has written dozens of technical articles for various leading publications. He regularly speaks at conferences in the US, and he has written nine books on programming topics, including the highly-rated 'Holub on Patterns: Learning Design Patterns by Looking at Code'.

www.holub.com

Kevlin Henney

Kevlin is an independent consultant and trainer based on the UK. He specialises in programming languages, OO design, patterns, development process and software architecture. He has been a columnist for various magazines and web sites, including The Register, Application Development Advisor, Java Report and C/C++ Users Journal. Kevlin is co-author of two recent volumes in the Pattern-Oriented Software Architecture series. He is also a regular speaker at various conferences, including DevWeek.

www.curbralan.com

Simon Brown

Simon is a senior consultant at C5 Alliance in Jersey. He's a hands-on software architect with a BSc in Computer Science, and over the past 12 years he's been involved in projects ranging from rich desktop clients and web applications through to highly scalable distributed systems and service-oriented architectures; predominantly within the finance industry. He's also undertaken consulting and training roles with a broader focus on people, process and technology. Simon is additionally active in wider aspects of the IT industry. He founded the 'Coding the Architecture' web site, and he has also written and co-written a number of books about Java EE web technologies, spoken at a various conferences, developed a software architecture training course, and actively maintains a public blog.

www.simonbrown.je

Christian Weyer

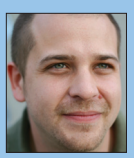
Christian is co-founder of thinktecture, a European company aiding and supporting software architects and developers in designing and implementing distributed solutions. He has modelled and implemented distributed applications with Java, COM, DCOM, COM+, Web Services and other technologies over the years, and recently he has been focusing on the ideas and concepts of service-orientation and their practical translation in customer projects, primarily using Windows Communication Foundation (WCF) and Windows Workflow Foundation (WF).

www.thinktecture.com

Jesus Rodriguez

Jesus is the Chief Architect of Tellago, Inc. He is also a Microsoft BizTalk Server MVP, an Oracle ACE and one of a small group of architects worldwide to be a member of the Microsoft Connected Systems Advisor team. He is an active contributor to the .NET and J2EE communities, and a regular speaker at various conferences, including Microsoft TechEd, SOAWorld, Oracle Open World, Web Services Security Conference and the Microsoft MVP Summit. He has written articles for various publications, including MSDN Magazine, Microsoft Architecture Journal, SOAWorld and Web Services Journal, and he is a prolific blogger on all subjects related to integration.

weblogs.asp.net/gsusx



Dominick Baier

Dominick is an internationally recognised expert on the security of .NET and Windows applications. He supports companies worldwide with design and implementation of security features in their software. He also has several years of experience in code auditing and reviewing, penetration testing and security policy definition and establishment, and is a certified BS 7799 (ISO17799) Lead Auditor. As one of the few "Developer Security" Microsoft MVPs, he works directly with various security teams in Redmond. One of the offsprings of this cooperation is the book 'Developing More Secure Microsoft ASP.NET 2.0 Applications', which was published in 2006 by Microsoft Press and quickly became a key reference for ASP.NET security. Dominick also leads the security curriculum at DevelopMentor, authoring and teaching courses about .NET, ASP.NET, WCF and Windows security.

www.leastprivilege.com

Eric Nelson

Eric joined Microsoft in 1996 as a Technical Evangelist, and has spent most of his time working with ISVs to help them architect solutions which make use of the latest Microsoft technologies – from the beta of ASP 1.0 through to ASP.NET, from MTS to WCF/WF, and from the beta of SQL Server 6.5 through to SQL Server 2008. In July 2008 he switched role from an Application Architect to a Developer Evangelist in the Developer and Platform Group, and his current interests include digging into the ADO.NET Entity Framework, ADO.NET Data Services and the Windows Azure Platform.

geekswithblogs.net/iupdateable

Jim Webber

Dr Jim Webber is director of professional services for ThoughtWorks, where he works on dependable distributed systems architecture for clients worldwide. Though his current interests are in using the Web for building distributed systems, Jim was formerly a senior researcher with the UK E-Science programme where he developed techniques for dependable, global-scale high performance computing. As an architect with Hewlett-Packard, and later Arjuna Technologies, Jim was the lead developer on the industry's first Web Services Transaction solution. Jim is a regular speaker at conferences across the globe, and in addition to authoring 'Developing Enterprise Web Services – An Architect's Guide' he is working on a new book on Web-based integration called 'REST in Practice'. Jim holds a B.Sc. in Computing Science and Ph.D. in Parallel Computing, both from the University of Newcastle upon Tyne.

jim.webber.name

Richard Blewett

Richard has been working in the software industry for over 20 years, starting with mainframes through the early years of client/server to today's service-oriented world. He has spent his professional life working on large distributed systems, including being the middle tier architect on the UK national police systems. He now focuses on technologies that enable developers to build large-scale systems on the Microsoft platform, such as WCF, BizTalk, Workflow and Azure. He is a regular speaker at various conferences including TechEd, DevWeek, Software Architect and Oredev, and he can often be found helping people on various newsgroups, mailing lists and web forums, including being a moderator of the MSDN WCF Forum – activity that has resulted in Microsoft awarding him MVP status for Connected Systems. Richard is also a DevelopMentor instructor, for whom he writes and teaches material on a wide range of technologies.

rocksolidknowledge.com

Dino Esposito

Dino is a trainer and software consultant based in Rome. A member of the IDesign team, he specializes in Microsoft .NET technologies, and spends most of his time teaching and consulting across Europe and the USA. He has hands-on experience in architecting and building distributed systems for banking and insurance companies. A prolific author, Dino writes columns for various magazines, and has written a number of books on various cutting-edge topics. An ASP.NET MVP, he regularly speaks at industry conferences around the world, including TechEd, DevConnections, Software Architect and DevWeek.

weblogs.asp.net/despos

Ian Robinson

Ian is a principal consultant with ThoughtWorks, where he helps clients create sustainable service-oriented development capabilities that align business and IT from inception through to operation. He's written guidance for Microsoft's patterns and practices on implementing service-oriented systems with Microsoft technologies, and has published articles on business-oriented development methodologies and distributed systems design – most recently in the ThoughtWorks Anthology (Pragmatic Programmers, 2008). He's currently co-authoring a book on Web-friendly enterprise integration with Jim Webber and Savas Parastatidis.

iansrobinson.com

Tobias Komischke

Tobias has been working in the area of user experience for over ten years, leading projects and teams to create user interfaces in various industries. He is Director of User Experience at Infragistics, a software company which specialises in user interface development tools and services. While he is deeply familiar with all aspects of user experience, his specialty is Human Factors Engineering, which is rooted in his academic background in cognitive psychology. Tobias is a frequent speaker at various conferences and author for technical journals, magazines and blogs.

www.infragistics.com

Hadi Hariri

Hadi is a developer, speaker, podcaster and Technical Evangelist at JetBrains. His passions include software architecture and web development. A book author and frequent contributor to developer publications, Hadi has been speaking at various industry events for over a decade. He is based in Spain where he lives with his wife and two sons, and runs the .NET Malaga User Group. He is also a C# MVP.

hadihariri.com

► *More speaker details on page 6*

Pre-conference workshops

TUESDAY 19TH OCTOBER 2010

The following workshops run for a full day (from 09.30 to 17.30), with a short break in the morning and afternoon, and a lunch break at 13.00.

Unless otherwise noted in the description, they are presentation-based in style rather than 'hands-on' labs.

Lean Software Architecture

JIM COPLIEN

WORKSHOP REF: SA01

The misguided, CASE-heavy practices of the 1980s fueled the proto-Agile rhetoric of the 1990s and survived full-force into the advent of Agile practices such as Scrum and XP in the past decade. Part of that rhetoric has been to avoid the sins of the 1980s as much as possible by discarding up-front requirements (instead we have a promise for a future conversation between a developer and a customer) and architecture (instead, we have had a succession of short-lived ideas including "metaphor" and TDD). Experience and recent research have both borne out the value of architecture in software development in general, as well as its value in sustaining high velocity and change resiliency in Agile projects.

In this talk, Agile expert Jim Coplien will provide tips for putting architecture back into your Agile project without dragging it back into the dark ages – and all within the framework of the Agile Manifesto. This talk will show a specific approach to architecture based on domain-driven design and Trygve Reenskaug's new DCI architecture that fulfil the original vision of object-oriented programming, allowing us to write software as if people mattered.

Building Web Applications the .NET 4.0 Way

RICHARD BLEWETT & ANDY CLYMER

WORKSHOP REF: SA02

.NET 4.0 introduces a number of technologies into the .NET world. This workshop looks not only at these technologies but, more importantly, how they can be combined into compelling applications. We will cover ASP.NET MVC2, WCF 4.0, Workflow 4.0 and Entity Framework 4.0 along the way, and show you not only the benefits of these technologies but also where they can cause problems.

During the workshop we will build a functional n-tier application that demonstrates the technologies "in-situ" and shows patterns that you can use to keep your architecture clean.

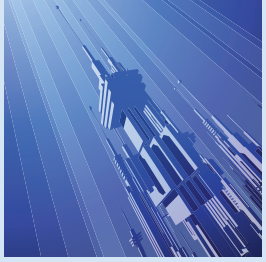
Software Architecture in a Day

SIMON BROWN

WORKSHOP REF: SA03

This one-day workshop is an interactive introduction to software architecture and what it means to be a software architect. It's aimed at software developers who are looking towards their first software architect role, developers who want to become more architecturally aware and software architects that are new to the role. We'll be asking and answering the following questions:

1. What is software architecture?
2. What is the role of a software architect?
3. How do you define software architecture?
4. How do you share software architecture?
5. How do you deliver software architecture?



Software Architect 2010
LONDON
19–22 OCT '10

Silverlight 4 for Architects

DAVID WHEELER

WORKSHOP REF: SA04

Silverlight 4 is a line-of-business-ready, Rich Internet Application (RIA) platform from Microsoft.

This one-day workshop will get you ready and primed for using Silverlight for your next-generation business applications. We'll focus on what it can, and can't do; we'll look at its many deployment options and security features; we'll compare it to WPF and ASP.NET; and we'll examine its performance, architectural solidity and maturity.

And along the way you'll gain a solid understanding of just how great Silverlight 4 is at addressing common business requirements; how productive development teams can be when using Silverlight; and how you (and your teams) might need to re-evaluate their thinking in the face of technologies such as WCF RIA Services, Expression Suite, and Windows 7 Phone.

If you thought that Silverlight was about media, or that it wasn't ready for prime-time then it's time to think again. The mixture of conceptual and hard-core discussions in this workshop will prepare you for those tough discussions around your next UI platform.

Pattern-Oriented Software Architecture

KEVLIN HENNEY

WORKSHOP REF: SA05

Patterns offer a successful way of exploring, reasoning about, describing and proposing design ideas. There are many valuable aspects of pattern-based thinking that are overlooked in the common perception of design patterns. The original vision of patterns embodies a notion of incremental, feedback-based design – something that may come as a revelation to anyone who had mentally pigeonholed patterns together with heavier-weight design approaches. They are also somewhat broader in application than just OO design, although that remains one of their most popular domains.

This session will start off with basic pattern concepts and practices, with examples, and work through a number of more sophisticated ideas, such as the relationship between pattern-oriented thinking and incremental development, patterns and software architecture, and how you can mine patterns in your own systems.

The OO Design Process, from Requirements to UML

ALLEN HOLUB

WORKSHOP REF: SA06

Many people who think that they're doing OO actually aren't, and many designs suffer from being put together in the wrong way. For example, a dynamic model (that shows the objects at run time and how they interact with one another) should drive the design process; the static model (the class diagram) is really just a reflection of things you discover while doing the dynamic model, and is assembled in the course of modeling the dynamics of the program. Most programmers, however, attempt to create a class diagram out of context, with no dynamic modeling at all, and the resulting system tends to be unwieldy as a consequence. The process that you use influences the quality of the product.

This full-day class goes through the entire OO-Design process, with an emphasis on how to arrive at an optimal design. We'll look at every major stage of the process. In particular, we'll look at requirements gathering and problem-statement definition, use-case analysis, and the simultaneous construction of the dynamic and static models using UML.

Over the course of the day, we'll construct a model for a small program, so that you can see how each step plays out in a practical context. No prior knowledge of either OO design or UML is assumed.

Conference sessions

WEDNESDAY 20TH OCTOBER 2010

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5	TRACK 6
08.30	REGISTRATION & COFFEE					
09.30	**Opening Keynote** **Restoring Form and Function to Patterns** JIM COPLIEN More than fifteen years ago, we laid the foundations for the software pattern discipline – a discipline that has become an enduring staple of software architecture. The initial community distinguished patterns from other approaches with high ideals related to systems thinking and beauty. Many of these ideals reflect age-old thinking about systems and form, from fields as diverse as urban planning and biology. However, the community's body of literature soon degenerated to the point of showing little distinction from its contemporaries except in its volume. This talk lays a vision for systems thinking in IT architecture, drawing on the principles that originally launched the pattern discipline. The talk argues that by focusing on system form, including the form of function, we can build better systems with lower lifetime cost than is currently possible with patterns. We'll examine both popular and emerging architectural notions to ground this vision.					
11.00	COFFEE BREAK					
11.30	**Where do you start?** SIMON BROWN One of the hardest things about software development is being asked to come up with a design when all you're given is a set of requirements and a blank sheet of paper. Many software teams will dive straight into the code and while this can initially be very productive, the slippery slope of constant refactoring is awaiting those teams that haven't quite found a design that works. Often, a little forethought is all that's needed to get the development process heading in the right direction. So where do you start?	**10 things every architect needs to know about Windows Azure** ERIC NELSON In this session we will look at the 10 most architecturally significant features of the Windows Azure platform which directly impact how you architect solutions if you plan to deploy in the Cloud.	**An introduction to Windows Workflow Foundation 4.0** RICHARD BLEWETT .NET 4.0 introduces a new version of Windows Workflow Foundation. This new version is a total rewrite of the version introduced with .NET 3.0. In this talk we look at what Microsoft are trying to achieve with WF 4.0, why they felt it necessary to rewrite rather than modify the codebase, and what new features are available in the new library. Along the way we will be looking at the new designer, declarative workflows, asynchronous processing, sequential and flowchart workflows and how workflow's automated persistence works.	**Real REST: using hypermedia in anger** TIM EWALD REST has made a lot of headway as the preferred model for SOA in the real world. On the cutting edge, developers are using REST to implement HATEOAS, or "hypermedia as the engine of application state". This talk examines this model as it is put into practice in a large scale distributed system using JRuby and Java. Topics include mapping hyperlinking and forms, common schemas, auto-generated proxies, versioning and more. It is a must attend talk for anyone curious about how far REST services can take you in the real world.	**Testing the entire stack** NEAL FORD Most talks you see about testing cover one particular tool, and rarely delve into the strategies around when you should use a particular tool for a particular kind of testing. This talk differs because it covers testing the entire stack: unit, integration, functional, behaviour-driven, databases, user acceptance, mocking & stubbing, and other topics and strategies. I discuss the merits of "known good state" vs. "nuke & pave" for databases, the differences between ClassicTDDers vs. Mockists and how they approach testing. Throughout, I provide strategies and heuristics to help guide you when making decisions about how, when, and why you are testing some part of your infrastructure.	**Architecture with style(s)** KEVLIN HENNEY Style is a term often used in the context of code, but normally to describe either formatting guidelines or as an indication of quality. Style has other connotations, most of which are concerned with describing the general philosophy and typical features of the code. In software architecture this use of the word style converges with its use elsewhere, in building architecture. Named styles have a great deal in common with patterns, describing the essence of a solution structure in response to some kind of problem. This session explores some of the typical styles available to software architects, with the aim of expanding both their design vocabulary and the set of solutions they consider.
13.00	LUNCH BREAK					

Ingo Rammer

Ingo is a co-founder of thinktecture, a company which specializes in helping developers and architects create distributed applications which perform and scale as necessary. He is a distributed application expert (technologies around WCF, WF, ASMX, MSMQ, Remoting), with a knack for optimizing scalability and performance of medium to large applications. He is a Microsoft MVP for Solutions Architecture, and a Microsoft Regional Director for Austria. Ingo was the co-author of Advanced .NET Remoting, and has written numerous technical articles.

www.thinktecture.com

Mike Taulty

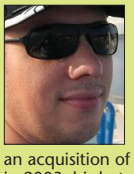
Mike works in the Developer and Platform Group at Microsoft in the UK, where he has spent the past few years helping developers understand and get the best out of the Microsoft platform. Prior to this, he spent three years with Microsoft Consulting Services as a consultant on developer technologies. Before joining Microsoft, Mike spent the previous nine years working as a software developer for a number of enterprises, consultancies and software vendors, using a variety of operating system, client, communication and server technologies. He is a regular speaker at various technical conferences, including TechEd Europe, DevDays Holland, MIX UK and DevWeek.

mtaulty.com

Tim Ewald

Tim is a Senior Consulting Software Engineer at SeaChange International, the world's leading vendor of video-on-demand and ad-insertion systems for television operators. He is responsible for the architecture of the company's next-generation video delivery platform. Tim has 20 years experience building complex distributed systems on a range of technology stacks. He is an internationally recognized speaker who always tries to have something interesting to say.

www.schange.com



Andy Clymer

Andy has been working as a software engineer since 1993, with a variety of languages, technologies & operating systems. A large part of his software engineering experience was at Cisco Systems, arriving there in 1997 as part of an acquisition of a small UK start-up. Andy finally left Cisco in 2003, his last role being Lead Architect for Cisco's identity framework, coordinating an international team of developers. Since leaving Cisco he has focused on .NET technology, and splits his time between consultancy with Rock Solid Knowledge and teaching for DevelopMentor, specialising in patterns and the chaotic world of multi-threading/parallel programming. He is a regular speaker at conferences like DevWeek and Software Architect.

rocksolidknowledge.com

Michael Stal

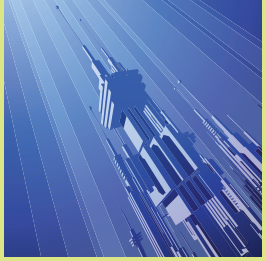
Dr Michael Stal is a Certified Senior Software Architect and Principal Engineer at Siemens Corporate Technology. He is the co-author of several books, including Pattern-Oriented Software Architecture Volume 1 (A System of Patterns) and Volume 2 (Patterns for Networked and Concurrent Systems). He has been involved in several large-scale software development projects. In addition, Michael acts as a editor-in-chief of JavaSPEKTRUM and co-host of the podcast Software ArchitekTOUR.

stal.blogspot.com

Eoin Woods

Eoin is head of application architecture at BlackRock (formerly Barclays Global Investors) and on a day-to-day basis is also responsible for the design of a new equities portfolio management system being built for the company's well-known active management group. Prior to BlackRock, Eoin has worked in the software engineering field for over 15 years, for a number of companies including Groupe Bull, Sybase, InterTrust and UBS Investment Bank. His main technical interests are software architecture, distributed systems, computer security, and data management; he is co-author of the book 'Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives', published by Addison Wesley.

www.eoinwoods.info



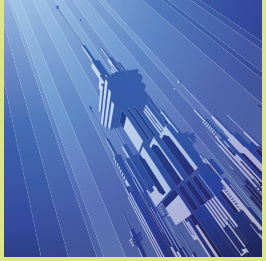
Software Architect 2010
LONDON
19–22 OCT '10

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5	TRACK 6
14.00	**Design patterns in the real world** ALLEN HOLUB Most books on design patterns present each pattern in splendid isolation, as if your program contained only a single pattern. In the real world patterns overlap one another and interact in complex ways. This class takes a unique approach to teaching patterns by analyzing a real computer program in terms of the patterns used to implement it. We'll look at how the design patterns are actually used in practice, and see how the strengths and weaknesses of the patterns play off one another. You'll also get a chance to see how real-world realizations of the patterns can differ from the Gang-of-Four examples and how a given pattern can be implemented in various ways. The examples are in Java, but C++ and C# programmers should have no problem following along. Some familiarity with the Gang-of-Four patterns is assumed – you should, at minimum, be able to identify them by name.	**Code correctness and software tools for .NET developers** DINO ESPOSITO Software contracts are not a new concept, but only now is the idea finding widespread implementation in a fundamental framework like .NET. In .NET 4, Code Contracts is an API that helps and encourages developers to formalize and assert under which condition each method is expected to run, which conditions it ensures will be verified at the end and which conditions always hold during the life of the object. This information can be consumed in many ways. It is helpful for statically checking the correctness of the code; it represents a form of exception handling; it can simply be part of the software specifications. Finally, it can be input to some smart test generator tools. This talk offers an overview of the Code Contracts API in .NET 4 and keeps an eye on emerging semantic tools like Pex to generate ad hoc tests.	**Building WCF services with Workflow 4.0** RICHARD BLEWETT Workflow is a powerful tool for building services – from clarity of data flow to automated persistence for long running functionality. In this talk I will take you though the building and consuming of WCF services with Windows Workflow 4.0 and introduce the powerful new message correlation infrastructure – a necessity for long running execution.	**Making integration easy: introducing REST-based integration** JESUS RODRIGUEZ In the enterprise architecture vocabulary, integration is often seen as being synonymous to complexity. This complexity stems from the fact that the keystone of every integration application is the implementing of interoperable data-message exchanges across heterogeneous systems and platforms. In recent years, the emergence of new architecture styles such as the Representational State Transfer (REST) have enabled a foundation for simpler and more open integration models. This session presents a series of patterns and best practices for implementing integration solutions based on the concepts of REST. The session illustrates how the use of REST or Web-based mechanisms such as HTTP stateless interactions, syndication formats, or resource-based modeling drastically improve traditional integration scenarios. To keep things practical, we have included a series of demonstrations that showcase real world integration scenarios powered by the use of the RESTful model on the Microsoft platform.	**Concurrency: we're doing it wrong** TIM EWALD Most developers know the conventional wisdom about concurrency in object-oriented programming: objects own data, data is protected by locks, objects own locks to protect their data, and all this is hidden from a thread, which accesses objects through their interfaces. Sounds plausible, maybe even not too hard, but in fact with multiple objects, multiple threads, and multiple cores in your multiple CPUs, it's almost impossible to get it both correct and efficient. Software Transactional Memory (STM) offers an alternative, and an implementation is shipping today in the Clojure programming language. This talk introduces the constructs Clojure uses to simplify programming in a highly concurrent world.	**Onion design** MICHAEL STAL There are many possible ways to create a software architecture. Unfortunately, books on software architecture rarely provide hints how to systematically address the task of software architecture design. Here the Onion Model comes to your rescue. It presents a model of systematically mapping project-relevant forces such as requirements or business constraints to architecture decisions, taking the type and priorities of these forces into account.
15.30	COFFEE BREAK					
16.00	**The uncertainty principle** KEVLIN HENNEY In software development a lack of certainty about something can be part of the solution rather than part of the problem. This point of view can, however, seem a little counterintuitive and more than a little disturbing. There is a strong tendency for humans to feel unsure about uncertainty, in two minds over ambiguity and a little wobbly with instability. Whether over technology choice, implementation options, requirements or schedule, uncertainty is normally seen as something you must either suppress or avoid. Of this many people appear, well, certain. That you should embrace it and use it to influence schedule, identify risk and inform design is not immediately obvious. This session looks at ways in which a lack of certainty offers the opportunity to highlight risk and reframe questions, making uncertainty part of the solution rather than necessarily a problem.	**Entity Framework and the Repository Pattern** ANDREW CLYMER .NET 4 comes with a version of Entity Framework that is fit for purpose. Entity Framework is Microsoft's ORM, their version of NHibernate. Entity Framework version 1 forced developers to be tightly coupled to Entity Framework, this goes against traditional ORM technology and creates problems for unit testing. Entity Framework 4 fixes this problem via POCO support. However it is still up to developers to work out how to utilise the technology to truly loosely couple application logic from business logic. This talk will take you through these new features and show how the Repository pattern can be deployed with Entity Framework to provide a testable and loosely coupled solution.	**Making WCF simple: best practices for testing, deploying and managing WCF Solutions in the big enterprise** JESUS RODRIGUEZ During the past few years, Windows Communication Foundation (WCF) has established itself as the fundamental technology to enable service orientation in the Microsoft platform. However, adopting WCF in the big enterprise requires complementing WCF's sophisticated capabilities with efficient practices in areas such as configuration management, unit/load testing, dependency management, and versioning among many others. Enabling these capabilities facilitates the adoption of WCF in the enterprise in an agile and simple manner. This session presents a series of solutions and best practices for addressing important aspects such as configuration management, endpoint discovery, automated unit/load testing, service dependencies, etc. We illustrate these solutions using a series of practical demonstrations that will help developers to implement large WCF solutions in a more efficient and agile manner. Additionally, the session presents three case studies that highlight how customers are implementing some of these solutions to adopt WCF at a big scale in the enterprise.	**Inside ASP.NET MVC** DAVID WHEELER There's a new ASP.NET in town, and it's lean, mean and testable. This talk will guide you through ASP.NET MVC, ripping open its innards and presenting them so that you can understand exactly how it works and how it can be extended. By focusing on testability and extensibility, this session will deliver the core insights into how you can make the most of this different approach to building large web-applications based around the ASP.NET pipeline architecture.	**Security 101: an introduction to software security at the architectural and process level** ALLEN HOLUB As more and more of our applications move onto the web, security becomes even more critical. Good security, however, has to be built in, not tacked on as an afterthought. Misconceptions about security – that an application can be made secure solely by using encryption (https) and firewalls, for example – abound. Moreover, security is more an architectural (at both the system and program level) and process problem than a coding problem. This session gives you an overview of what it means to make an application truly secure. We'll talk about how to design and build your applications so that they will be truly secure. Covered topics include security architectures, code and design review, penetration testing, risk analysis and risk-based testing, security-related requirements, static analysis, abuse cases, security operations, and crypto. Time permitting, we'll finish up by looking at a common "exploit" so that you can understand the sorts of things that make an application vulnerable.	**Implementing emergent design** NEAL FORD This session describes the current thinking about emergent design, discovering design in code. The hazard of Big Design Up Front in software is that you don't yet know what you don't know, and design decisions made too early are just speculations without facts. Emergent design techniques allow you to wait until the last responsible moment to make design decisions. This talk covers three areas: emergent design enablers, battling things that make emergent design hard, finding idiomatic patterns, and how to leverage the patterns you find. It includes both proactive (test-driven development) and reactive (refactoring, metrics, visualizations, tests) approaches to discovering design, and discusses the use of custom attributes, DSLs, and other techniques for utilizing them. The goal of this talk is to provide nomenclature, strategies, and techniques for allowing design to emerge from projects as they proceed, keeping your code in sync with the problem domain. This talk shows lots of examples of how to make this concept work in your environment.
17.30	DRINKS RECEPTION					

Conference sessions

THURSDAY 21ST OCTOBER 2010

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5	TRACK 6
09.00	COFFEE					
09.30	**What's the problem?** KEVLIN HENNEY The focus of software development and technology tends to be very solution-centric, often at the expense or in the absence of a proper understanding of what problem is to be solved. Without necessarily intending to, developers, architects and other technical roles often try to force fit the problem domain into code-based thinking. Business analyst says number, developer hears int, double or decimal. Customer says stock data, architect hears relational database. The problem domain and motivation are often abstracted away altogether or too early in the technical solution process. Although they need to act as a bridge between the world of the problem and the world of the solution, architects are often guilty of mistaking a design possibility for a requirement. They will also think in terms of the world of the solution outwards rather than the world of the problem inwards. This issue of overlooking the problem also occurs in the technical domain, so that patterns are often thought of as solutions rather than solutions matched to problems. This session explores these issues, their consequences and some techniques for addressing them.	**Cross-tier data exchange: old problem, new solutions** DINO ESPOSITO When designing a multi-tier system, one of the most critical decisions for an architect to make is how to move data around the tiers. Tiers imply a physical distance between servers or processes; and crossing a tier inevitably raises concerns about performance. In the .NET space, two approaches have emerged over the years in some way backed by the .NET Framework and related tools. The most popular, and still largely used, is the approach based on DataSets. Another common approach is based on data-transfer objects (DTO). For different reasons, none of them can be considered the definitive solution. In this session, we'll review the options offered by NHibernate and Entity Framework and discuss in detail self-tracking objects while keeping an eye on a completely different REST-based approach backed by .NET Data Services.	**10 things an architect should know about WPF (and Silverlight)** DAVID WHEELER So you're an architect now, but you remember what it was like to write Windows programs. Those heady days of message pumps, window procedures, handles, and producing your own UI in WM_PAINT handlers. But now your developers are speaking in terms of data bindings, ICommand, "lookless controls", dependency properties and styles. So come to this session for an insight into what a modern architect needs to know about the next generation of UI development, and how to get a feel for how the overall application architecture is impacted by the newer UI technologies.	**Building RESTful services on the Microsoft platform: when to use what?** JESUS RODRIGUEZ The increasing popularity of the Representational State Transfer (REST) architecture style has increased its adoption by various programming technologies. If you work with the Microsoft platform, you currently have a plethora of options for developing RESTful services. Technologies such as Windows Communication Foundation (WCF), ADO.NET Data Services, ASP.NET MVC and the WCF REST Starter Kit all provide the fundamental building blocks for building RESTful services. It is important for developers to determine which one of those technologies is the best fit when it comes to implementing real world REST scenarios. This session will deep dive into the different options that enable the implementation of RESTful services in the Microsoft platform. Specifically, we will explore the strengths and weaknesses of technologies such as WCF, ADO.NET Data Services and ASP.NET MVC within the context of real world RESTful application. We'll highlight a series of development techniques developers can use to implement some of the fundamental RESTful services patterns using different Microsoft technologies.	**Hands-on agile development** NEAL FORD BRING YOUR LAPTOP WITH YOU, BUT A LAPTOP ISN'T REQUIRED! Reading and hearing about agile practices is one thing, but actually doing it is completely different. This session puts you to work in an agile fashion, applying agile developer practices. During this session, we're going to take a problem and iteratively develop the solution, using test-driven development, pair programming, retrospectives, pair rotation, and other agile development techniques. We'll work through three 20-minute iterations during the 90 minutes, giving you a hands-on feel for real agile development. If you have a laptop, bring it, but only half the class needs one, so if you don't have a laptop, don't let it discourage you. Come and see what it's like to work on a real agile project, even if it's just for 90 minutes.	**Using design principles to unify software architecture and design** EOIN WOODS Enterprise architects, software architects and software development teams often have different priorities and this results in them choosing very different solutions: enterprise architects take the long view and worry about the organisation as a whole, software architects may also take a long term view but they're worried about their system in particular, while development teams often have to focus on the here-and-now, in terms of the next release that they need to deliver. These different priorities often cause a lot of spirited discussion between these different groups! However they're all trying to achieve the same goal, namely the delivery of the systems that their organisation needs. The question is how we avoid conflict and allow enterprise architects, software architects and development teams to work together cooperatively. In this talk I'll explain how the use of design principles – from the basic principles that underpin organisational choices, through to the detailed principles used to guide the design of a system – are a unifying concept that help these different groups to work together in something close to harmony.
11.00	COFFEE BREAK					
11.30	**Lean principles for software architecture** KEVLIN HENNEY Agile software development covers a broad range of practices and ideals, but is often characterized (and even caricatured) in terms of just being a synonym for Scrum or Extreme Programming or of having little concern for architecture. The truth is invariably more subtle, and it is worth examining agility from a different perspective to see where architecture and process variation play a role. Such a perspective is provided by Lean Software Development, which takes the lessons of Lean Product Development and Lean Production and applies them to software development. This session explores the implications of Lean thinking on software architecture and the design choices and styles that best support a more sustainable approach to development.	**Farewell flipchart? Visual Studio 2010 for real men – erm, architects** CHRISTIAN WEYER Architects draw on flipcharts or napkins, period. Already in 2005 Microsoft tried to make Visual Studio more accessible and useful for more than just coding, without big success. Now, with Visual Studio 2010 application architects get a new set of powerful tools. Whether you are interested in domain modeling and engineering with UML, want to get an understanding of existing codebases and dissect applications with the Architecture Explorer, or design new layered applications step by step: code is always part of the game. So, VS2010 seems to be the better napkin? Christian investigates with a critical eye, and also sheds some light on third party tools that might improve the experience.	**Presentation patterns** DAVID WHEELER Models, views, controllers, presenters, viewmodels, behaviors and commands. Whichever way you cut it, you're likely to find at least three, and maybe more, of these terms flying around when you're discussing your presentation tier. In this session, you'll get to see how to really design and build top quality presentation tiers, in WPF and Silverlight, as well as get a true understanding of what these terms actually mean in the context of the presentation tier. Not just a patterns talk; this is a patterns talk with consequences!	**REST architectures with ASP.NET MVC** HADI HARIRI In this session we're going to cover the basics of REST and how to effectively implement a REST architecture over HTTP with ASP.NET MVC. We'll see how to use the different HTTP verbs, how to leverage the power of Content Types and extend the ASP.NET MVC framework to implement custom objects to maximize our productivity by letting HTTP do all the work. This is a 100% code-oriented session. It's about writing applications in a nice, testable and maintainable way.	**Where did my architecture go? Preserving the software architecture in the implementation** EOIN WOODS A major problem with comprehending software systems is the difficultly in discerning the design and architecture from the source code. Today's programming languages do not allow concepts such as components, layers, patterns and mechanisms to be clearly identified and so we rely on naming conventions, comments, package structures and so on. These are a first step but they quickly become overloaded with multiple meanings and often aren't used consistently throughout the system. In this talk I'll provide practical ideas and guidance as to how the architecture of a system can be represented in such a way that it can be recovered and used in conjunction with software tools for the software architect to keep visibility of their architecture as the system evolves. The focus will be on enterprise Java, but the ideas can be applied to other technologies such as .NET and C++.	**SOLID software for a changing world** DINO ESPOSITO Gone are the days in which drag-and-drop of components were all you needed to build working applications. The era of RAD is probably gone forever, defeated by the growing complexity of modern applications, which has created the conditions for revisiting object-oriented design principles. An extension of classic OOD principles, SOLID principles are five design principles that when systematically and insightfully applied make your software loosely-coupled, testable, and especially easy to understand and maintain. This session will discuss the conceptual value of SOLID principles and illustrate them with a few examples.
13.00	LUNCH BREAK					



Software Architect 2010
LONDON
19–22 OCT '10

	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5	TRACK 6
14.00	**Patterns dartboard** ANDY CLYMER Imagine a (virtual) dartboard. That dartboard has a selection of patterns pinned to it. Members of the audience can throw darts at the dartboard. Where the darts hit will determine how the talk goes. The talk will be about patterns, although which ones it's impossible to know – it's up to you to choose what goes into the talk! Will it be MVC? Factory? Command? Your darts! Your choice!	**Introducing Windows Server AppFabric extensibility** JESUS RODRIGUEZ This session deep dives into the new application hosting, tracking, and persistence capabilities of .NET 4 and Windows Server AppFabric. We'll explore the benefits of different hosting options and how to choose the right option for your scenario. In order to keep things practical, there will be a series of demonstrations that focus on the techniques developers can use to build custom tracking providers in order to meet custom business data monitoring requirements or customer persistence providers in order to efficiently manage application state. Additionally, we deep dive into the different extensibility mechanisms of the AppFabric caching subsystem.	**Basics of the Windows Phone 7 architecture** INGO RAMMER In this session, Ingo will introduce the basic architecture of the Windows Phone 7 operating system and its applications. You will learn about the guiding design principles, and how your application can interact with the user and with the outside world using the phone's sensors. You will learn about the main use cases for phone applications and how the combination of Silverlight on the phone and WCF on your backend services will allow you to provide an integrated user experience.	**Implementing evolutionary enterprise architecture** NEAL FORD We're drowning in needless complexity in the enterprise architecture space: heavy, bloated tools, complex middleware, just-in-case architectural decisions, and vendor-itus. The side effect of all that complexity drives us further from our goals: architecture that is simple, free, supports business goals, loosely coupled, and evolvable. This session describes how to use web technologies (HTTP, REST, hypermedia, etc.) to implement robust, scalable enterprise architecture. This talk is based on original research and development done by ThoughtWorks, and represents the current state of the art in building truly scalable enterprise architectures. This topic combines the subjects of service oriented architecture with web technologies to create a hybrid providing you with the benefits of both approaches. You can build robust, scalable enterprise architecture that allows individual applications to evolve independently and rapidly. This talk describes how to make SOA suck less.	**Web 2.0 vs. Human 1.0** TOBIAS KOMISCHKE One reason why User Experience is gaining more and more momentum across industries is its rootedness in multiple disciplines that all contribute their own rich traditions and findings. Human Factors belong to the most intriguing and powerful of these disciplines. Understanding human characteristics and applying this knowledge to the user interface design can greatly boost the user experience. In this talk participants will get to know fascinating facts about capabilities and limitations of humans. Based on this, concrete dos and don'ts of front end design are explained. The goal is to equip participants with concrete knowledge that they can directly apply in the design or evaluation of web sites and web applications.	**BDD: adding value to the business** HADI HARIRI BDD is hot! It's the next best thing! You need to learn it! Actually, who cares? If you want to come to a talk that explains an acronym and points you to some tools, then this is the wrong session for you. BDD isn't new; in fact, once you understand the concept behind it, you'll realise that it's just common sense. If you want to see how to improve the software you write, eliminate the misunderstandings you and your team have with the business analysts, and ultimately add value to the customer, then come along. As a bonus, you might even get to see a couple of frameworks that help you accomplish things. But more importantly, you'll understand how communication is the key.
15.30	COFFEE BREAK					
16.00	**Agile engineering practices** NEAL FORD Most of the time when people talk about agile software development, they talk about project and planning practices but never mention actual development, as if development were an afterthought when writing software. This talk drills into the real details of how to do agile development. I discuss best practices like continuous integration, pair programming, how developers should interact with story cards, how to handle enterprise concerns like integration with other software packages, and a slew of other topics related to agile software development.	**Architecting claims-aware applications using Windows Identity Foundation and Active Directory Federation Services** DOMINICK BAIER Claims, Federation and Single-Sign-On are commonly used terms in today's distributed application security. And finally Microsoft added all these capabilities to the .NET Framework as well as the Windows Server platform. The developer framework is called Windows Identity Foundation (WIF) and tightly integrates claims and security token processing into WCF and ASP.NET. Active Directory Federation Services 2.0 is Microsoft's out of the box Security Token Service that can both act as an identity provider or federation gateway. When you combine these two pieces, you can build powerful applications where (rather complex) security can be nicely abstracted away – cross platform, cross toolkit and cross security boundary.	**Architecting for multi core** ANDY CLYMER Herb Sutter famously coined the phrase "the free lunch is over" back in 2005, when it became clear that processor clock speeds were no longer obeying Moore's law. If developers wanted applications to go faster they could no longer rely on greater clock speeds, they would need to think differently and restructure their code to take advantage of multiple cores in order to get better and better performance. It turns out parallelizing all but the most trivial piece of code is challenging. .NET 4 attempts to assist the developer by providing support in the framework to assist parallelizing algorithms through the use of parallel constructs like Parallel.For, Parallel LINQ and a variety of concurrent data structures. The framework vendors would like you to believe that the free lunch is now back, but whilst they can deliver a moderate free lunch, if you truly want a gut-busting free lunch you will have to deploy a range of tricks for your algorithm to take full advantage of those multiple cores.	**Understanding and implementing OData** MIKE TAULTY OData was announced with a splash at the MIX10 conference in 2010. The Open Data Protocol is an open specification consisting of a set of extensions to add abilities around discoverability, consistent querying and data formats to RESTful services. If you've looked at technologies such as "Astoria", "ADO.NET Data Services" or "WCF Data Services" in the past then you've already encountered OData. In this session we'll walk through an overview of OData, explain what it adds to standard RESTful services, and look at some of the server-side (WCF Data Services) and client side (Silverlight) libraries for building and accessing OData compliant services.	**The counterintuitive web** IAN ROBINSON The web doesn't care for your finely-honed application architecture principles – for your orthodox tell-don't-ask, information hiding dictums, separated concerns, and guaranteed and reliable delivery strategies. It's an irresponsible place, where exposing your data, polling for results and making your errors the client's problem are considered acceptable behaviour. If it wasn't so successful, it'd be dismissed as an architectural clown. But despite its disregard for polite architectural society, it consistently beats your enterprise application efforts – and all at massive scale. It's time to find out why. This talk is suitable for distributed systems developers and architects, and anyone curious about the web's place in the enterprise.	**How do you design for non-functional requirements?** SIMON BROWN You may have heard people talking about "non-functional requirements", mentioning things like performance, scalability, availability, security, flexibility and so on. What do you do if you're asked to design software that exhibits these qualities and furthermore, how do you do it in a pragmatic way? Come along to find out about some of the approaches that you can adopt and some of the trade-offs you'll need to make.
17.30	END OF CONFERENCE					

Post-conference workshops

FRIDAY 22ND OCTOBER 2010

The following workshops run for a full day (from 09.30 to 17.30), with a short break in the morning and afternoon, and a lunch break at 13.00.

Unless otherwise noted in the description, they are presentation-based in style rather than 'hands-on' labs.

Distributed Agile Development Workshop

NEAL FORD

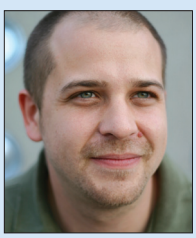
WORKSHOP REF: SA07

Reading and hearing about agile practices is one thing, but actually doing it is completely different. This workshop puts you to work in an agile fashion, applying agile development practices. During this workshop, we're going to take a problem and iteratively develop the solution, using test-driven development, pair programming, retrospectives, pair rotation, and other agile development techniques. To make it interesting, we're going to split the room into two continents and work through the issues faced by real distributed agile teams. We work through several 20-minute iterations during the workshop, giving you a hands-on feel for real agile development. If you have a laptop, bring it, but only half the class needs one, so if you don't have a laptop, don't let it discourage you. Come see what it's like to work on a real distributed agile project, even if it's only for a few hours.

End-to-end Application Architecture with .NET: Distributed, Secure, Pragmatic

DOMINICK BAIER & CHRISTIAN WEYER

WORKSHOP REF: SA08



All these technologies, approaches, patterns, do's and dont's… Frameworks and technologies like WCF, ASP.NET, WIF, different UI stacks and more are all available for you – but what to do with them, how to orchestrate them to build distributed applications?

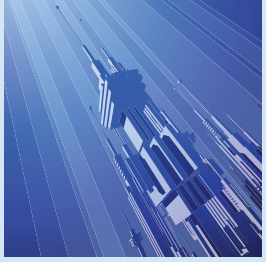
In this workshop Dominick and Christian show you in a practical manner how to solve questions around architecture, topology, communication, data handling and security. Come and see how to tackle a complex distributed system with a lace of pragmatism to make sure you don't get lost in the technology jungle out there.

Boxes & Lines

SIMON BROWN

WORKSHOP REF: SA09

Talking about software architecture will get you so far, but in reality it's all about experience and that's exactly what this workshop offers. Given a set of functional requirements, you'll be asked to identify the other architectural drivers before doing some design, choosing some technologies and drawing some boxes and lines. Join us if you want to know where to start with designing software and how to go about the software architecture definition process in a lightweight, structured and pragmatic way.



Software Architect 2010
LONDON
19–22 OCT '10

Real World Architecture with ASP.NET MVC

HADI HARIRI

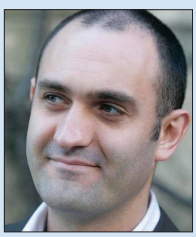
WORKSHOP REF: SA10

ASP.NET MVC is a great framework for those that embrace web technology. However, when it comes to creating robust, scalable and maintainable applications, it's very important to get the architecture correct. In this session we'll discover how to create applications from the ground up. We'll cover architectural concerns, infrastructure, testing, data access, view models, domain models, etc. – the fundamental concerns that arise when developing web applications using MVC.

REST in Practice

JIM WEBBER & IAN ROBINSON

WORKSHOP REF: SA11



The Web is fast becoming a serious competitor to traditional enterprise architecture approaches.

This tutorial will provide an introduction to RESTful Web Service techniques, both from a theoretical and practical perspectives.

The REST in Practice tutorial is broken down as follows:

- Introduction and Motivation
- The Web Architecture
- Simple Web Integration including POX and URI tunnelling
- CRUD Services using URI templates and HTTP
- Semantics using Microformats and RDF
- Hypermedia and the REST architectural style
- Scalability and how a text-based client-server polling protocol outperforms everything else!
- ATOM and ATOMPub for event-driven and pub/sub applications Security
- Conclusions and further thoughts

Participants should be comfortable with distributed computing concepts, but won't need any particular integration or middleware experience.

Moving to the Parallel Mindset with .NET 4.0

RICHARD BLEWETT & ANDY CLYMER

WORKSHOP REF: SA12

The move to concurrent programming is one of the big themes of .NET 4.0. The parallel framework extensions (PFx) update the .NET asynchronous APIs to ones that are tuned to working well with the multi-core environments in which we find ourselves.

This workshop introduces this new library and shows that although we have some powerful new tools in our toolbox, parallel programming requires a shift in mindset. We'll show you how the "trade show" demos you often see with PFx belie a much more complex set of interactions that you have to understand to write truly scalable parallel code.

We'll also introduce patterns and idioms that give you strategies for parallelizing algorithms.