



Software Design  
& Development

## Full Timetable & Session Profiles

19–23 May 2014  
Barbican Centre, London

For further details and booking,  
please visit [sddconf.com](http://sddconf.com)

Sponsored by



Mon 19 May 2014

### All-Day Pre-Conference Workshops

9:30–17:30

#### PRE-01 .NET programmers and architects: here are your options

Oliver Sturm

You're a programmer or an architect with experience in the Microsoft .NET environment? Your world view has changed considerably in recent times! Client UI is being repositioned, technologies like XAML and HTML unified in an architectural sense, and the HTML/JavaScript platform provides new opportunities, but also great challenges. Data access is no longer the same and what about service orientation? At least we still need a powerful web server. Or do we?

This full-day workshop sets out to describe the choices and opportunities available to programmers and architects alike, from a technical viewpoint with a .NET background. There will be room for questions, discussions and your own ideas!

#### PRE-02 Writing great Windows Store apps in XAML and C#

Jeff Prosize

For Windows developers, Windows Store apps are the future. This compelling new model can be used to write apps that run on desktops, tablets, and devices with other form factors.

In this pre-conference seminar, learn the essentials of writing Windows Store apps in XAML and C#, including how to craft modern UIs, how to build great touch interfaces, how to make your app come alive with live tiles and background tasks, how to integrate with the charms bar by implementing contracts, and a whole lot more.

#### PRE-03 A day of writing asynchronous code with C# 5

Richard Blewett & Andrew Clymer

Asynchronous programming is moving further into the mainstream with the introduction of the new `async` and `await` keywords. It is therefore becoming more and more important that we

write `async` libraries for our own application stacks.

During this intensive day we will do a deep dive into how these new features work, and into the mechanics of how best to deliver asynchronous functionality in your applications, focusing not only on asynchronous compute but also on asynchronous IO, and how you can adapt existing old-style APIs to work with the new coolness.

#### PRE-04 Modern JavaScript development

Kevin Jones

JavaScript as a language has been around for a number of years, however it's only recently that the power and expressiveness of the language has been brought into the mainstream.

This day-long session will examine the power of the language, and how to use JavaScript in modern web development. We will look at JavaScript functions and how they are the powerhouse of JavaScript. Building on that we will look at closures, explaining what they are and how they let us do things that would otherwise not be easy. Closures lead to the idea of modules, and we will look at a couple of module patterns. Using modules we can load code in a more structured way, and this leads us onto looking at code loaders and RequireJS in particular. Finally, you need to be able to test this code so we will cover testing JavaScript.

#### PRE-05 The architecture clinic

Juval Löwy

Wonder about architecture best practices, guidelines and pitfalls? Or you sense there is more to decomposing a system based merely on requirements? Wonder how to design world-class systems? You understand the concepts but not how to apply them?

In the first half of this fast-paced tutorial, Juval will explain his original approach to large system analysis design and his method. Then, he will present a comprehensive case study, outlining the architecture that addresses the requirements, discussing logical tiers, security, interoperability, scalability, transactions, and other aspects of a modern application. You will see how to approach rarely discussed topics such as allocation of services to assemblies, allocation of services to processes, transaction boundaries, identity management, authorization and authentication boundaries and more.

#### PRE-06 UX design for developers

Dave Wheeler

Increasingly, developers have to work with technologies that are created with designers in mind. WPF, Silverlight, Windows Phone and Windows 8 all benefit from having a designer on the team.

Which is a shame, as the majority of the teams don't have a designer.

This highly interactive workshop will kick-start your inner designer. Covering everything from the Savannah Principle to A/B Testing, and a host of stuff in between, you will leave this workshop with some practical skills in the areas of UI and UX design.

You'll need to bring a laptop or tablet, something portable that you can take pictures with, and pen and paper to get the most out of the exercises. Plus you'll need to bring plenty of imagination and a willingness to step out from being a mere developer.

#### PRE-07 Programmer's dozen: thirteen recommendations for reviewing, refactoring and regaining control of your code

Kevin Henney

When you look for it there appears to be no shortage of technical wisdom on what makes for clear and robust code. The quantity and consistency of such wisdom can, however, often leave programmers swamped. This may, in part, help explain why so much of the advice is ignored in so much code. Ideally there should be common themes that can be found across the many different specific practices identified as great and good, principles and overarching guidelines that help to make sense of the details and contradictions.

This session offers a thirteen-point list of recommendations (zero through twelve) which can be applied out-of-the-box to reduce code size and complexity, acting as guidelines for new code, indicators for refactoring and opportunities for review and discussion. The short list has no ambition to be all that you needed to know about design but were afraid to ask, but it does offer an easily memorable and rewarding set of practices for anyone who wants to ask better questions of their code. This workshop will also allow plenty of opportunities for discussion.

#### PRE-08 Design patterns in depth

Allen Holub

Without good OO structure, Agile development, which mandates constant refactoring, fails. Design Patterns help by providing classes of solutions to common programming problems.

Patterns, however, are usually presented in a catalogue format which gives you no feeling for how the patterns are actually applied in the real world, where the patterns interact in complex ways. This class discusses both good object-oriented structure and the most-commonly used design patterns, using an in-depth analysis of real code that demonstrates how the patterns work in context. We'll also cover interface-based design and the make-up of a well-structured object and class hierarchy.

The extensive code examples are in Java, but they should present no problem to C++ or C# programmers.

#### PRE-09 Agile software architecture sketches and NoUML

Simon Brown

Agility is about moving fast and this requires good communication. A consistent, shared vision is essential in order for teams to push in the same direction, but it's surprising that many agile teams struggle to effectively communicate the architecture of the software they are building. As an industry we do have the Unified Modeling Language (UML), yet many people favour informal "boxes and lines" sketches instead. The problem is that such diagrams rarely make any sense, usually need a narrative to accompany them and ultimately slow the team down. Although we can argue whether UML offers an effective way to communicate software architecture, that's often irrelevant because many teams have already thrown out UML or simply don't know it. Abandoning UML is one thing but, in the race for agility, many software development teams have lost the ability to communicate visually.

This hands-on workshop is aimed at everybody involved in the software development process and is about improving architectural communication. You'll see some patterns and anti-patterns related to NoUML diagrams, and you'll learn some simple techniques for communicating software architecture using informal sketches.

Tue 20 May 2014

9:30–11:00

**KEYNOTE PRESENTATION**

**Software architecture vs. code**

**Simon Brown**

Software architecture and coding are often seen as mutually exclusive disciplines, despite us referring to higher level abstractions when we talk about our software. You've probably heard others on your team talking about components, services and layers rather than objects when they're having discussions. Take a look at the code-base though. Can you clearly see these abstractions or does the code reflect some other structure? If so, why is there no clear mapping between the architecture and the code? Why do those architecture diagrams that you have on the wall say one thing whereas your code says another? In fact, why is it so hard to automatically generate a decent architecture diagram from an existing codebase? Join us to explore this topic further.

11:30–13:00

**TRACK 1 C# 5 async/await**

**Andrew Clymer**

.NET 4 introduced the new task abstraction, and C#5 will take advantage of integrating this into the language via the `async` and `await` keywords. In addition, the task abstraction promotes a new way of architecting asynchronous behaviour; not only for the client side but also for the server side, whether it be WCF or MVC. We will complete the session by building various `async/await`-friendly data structures.

**TRACK 2 WinRT is love**

**Dave Wheeler**

Don Box famously coined the phrase "COM is love". Fifteen years on, we're still feeling the love for COM as we embrace its latest incarnation with WinRT.

This is a deep dive talk into all things WinRT, from the way that Windows 8 Store Applications get launched through to the language projections and the type system.

So no matter whether you work on Windows 8, in C++, C# or JavaScript, you really need to know what's under the covers. And that's what this talk delivers.

**TRACK 3 Introduction to ASP.NET WebAPI**

**Brock Allen**

This session provides an introduction to ASP.NET WebAPI, which is the de facto framework in .NET for building HTTP-based (and RESTful) services.

We will cover the motivation for HTTP-oriented services as an alternative to SOAP-based services. We'll examine the major tenets of HTTP-based services including the importance of URIs, HTTP methods and status codes, content negotiation and hypermedia. We'll conclude with the new client-side programming model for HTTP services with the `HttpClient` API.

**TRACK 4 JavaScript for non-JavaScript developers**

**Oliver Sturm**

For years now we've been fighting it... but life must go on. JavaScript is everywhere these days, knowing the basics is hardly optional anymore and sometimes it's hard to avoid having to use it, too.

In this talk, Oliver shows that it can be fun (and funny!) to learn a little something about the language. Look at it like Chuck Norris might: JavaScript will get to know us!

**TRACK 5 Improving web performance**

**Robert Boedigheimer**

This session will start with a traditional ASP.NET web site and show step by step how to improve it for both client experience and scalability. Review the basics of caching and learn how to avoid costly server round trips by using expirations to maximize use of the client's browser and also reduce server side execution time with data caching on the server. Use HTTP compression, minification of JavaScript and CSS, and image optimization to reduce client downloads by 50-75%. Take advantage of free CDN networks to host jQuery and Ajax files. See how tools such as Fiddler and Google Page speed can be used to help diagnose and verify improvements. Use jQuery to lazy load images only as they are about to be displayed. Learn the techniques that can have the largest impact on web performance with the least amount of work.

**TRACK 6 Designing for mobile**

**Nathaniel T. Schutta**

The word just came down from the VP – you need a mobile app and you need

it yesterday. Wait, you've never built a mobile app... it's pretty much the same thing as you've built before just smaller right? Wrong. The mobile experience is different and far less forgiving. How do you design an application for touch? How does that differ from a mouse? Should you build a mobile app or a mobile web site? This talk will get you started on designing for a new, and exciting, platform. Whether that means iPhone, Android, Windows Phone or something else, you need a plan. This talk will help.

**TRACK 7 The meaning of Agile**

**Kevlin Henney**

Agile is a word that accurately describes a philosophy and practice of development that has gained credibility and voice in recent years. The practice and motivation of what many companies, teams, managers and developers are calling Agile, however, often seems a long way from the everyday use of the word 'agile' or what was meant originally by those who championed the term and the movement. Scrum, for example, is often portrayed as a simple adjunct to classic project management practices, its empirical basis and self-directed team philosophy often lost in a blur of micromanagement. The technical practices of Extreme Programming are often ignored as inconvenient or not core to a management-centric view of Agile. The same oversimplification and cherry picking can be seen in the wave of Lean thinking adopted into Agile as approaches such as Kanban enter the mainstream and suffer their own dilution.

This talk rewinds the clock and explores the motivation and implications behind the original Manifesto for Agile Software Development, the role of technical practices, the importance of quality in code and architecture, the evolutionary and adaptive nature of Agile, and the psychological and social underpinnings for many of the practices and concepts to be found in the successful application of Agile thinking.

**TRACK 8 Introduction to 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 responsi-

ble moment to make design decisions. This talk covers four 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.

14:00–15:30

**TRACK 1 Applied NoSQL in .NET**

**Michael Kennedy**

Perhaps you've heard about the next generation of databases, roughly classified as NoSQL databases? These databases are generally much better than RDBMS at scaling, performance, and ease-of-development (e.g. in NoSQL the object-relational impedance mismatch usually disappears). Unfortunately, many talks on NoSQL are very academic and general. Not this one.

We'll explore the NoSQL landscape and look at the various options out there. Then we'll learn how to leverage MongoDB (a popular NoSQL DB) to build .NET applications using LINQ as the data access language. From there we will build out a .NET application using LINQ and MongoDB in a series of interactive demos using Visual Studio 2012 and C#.

**TRACK 2 Proximity networking in WinRT**

**Jeff Proisie**

Tap two devices together and transfer data between them? You bet! Learn about the Near-Field Communication (NFC) API in Windows and Windows Phone and see how to put them to work to connect devices more easily than ever before. The future of networking is here, and it is spelled NFC!

**TRACK 3 OWIN and Katana: the new HTTP host for .NET web applications**

**Brock Allen & Dominick Baier**

OWIN is a fairly new specification for modelling an HTTP server API for

hosting web applications. The intent is to decouple the host from the application with the goal of providing a simpler programming model with potentially better performance than traditional web servers. While OWIN is a specification, Katana is Microsoft's implementation of an OWIN host.

In this session we will further discuss OWIN, Katana and how to write applications and middleware that take advantage of this new and exciting hosting environment.

**TRACK 4 AngularJS for .NET developers**

**Christian Weyer**

Not sure which UI technology to build on? In recent years several frameworks for the development of large and maintainable HTML5/JavaScript applications have popped up in the market. One of the most successful and widely used is Google's AngularJS.

In this session you will learn how to apply it to develop business applications in a way that resembles the well-known MVVM model from WPF and Silverlight. Separate the view from the logic and "feel good" with JavaScript. Christian Weyer shows you what structured, maintainable and testable applications today may look like on the web platform, whether desktop or mobile.

**TRACK 5 Embracing HTTP in the era of Web APIs**

**Hadi Hariri**

In this talk we're going to get back to basics and understand the HTTP protocol and appreciate it for what it is: an application protocol. We will see how to effectively use HTTP to create a robust API for our applications which serve not only as end-points for integration but for the UI as well. We'll see the differences between REST and HTTP APIs, and the benefits which the latter adds.

This is a session of back-to-the-metal, learning the ins and outs of what makes the web today.

**TRACK 6 Boosting the user experience through attractiveness and rich interactions**

**Tobias Komischke**

User experience is far more than usability. This presentation covers two crucial topics that can boost (or wreck) the user experience of software products. Attractiveness is one goal that every stakeholder wants to have for a

product, but what is attractiveness? How can it be designed? What are the dos and don'ts? Rich interaction capabilities allow modern development platforms like WPF and Silverlight to provide much nuanced user-application interactions. Yet, are those rich features always really useful, usable and desirable? The presentation shows good (i.e. beneficial) and bad (i.e. disruptive) uses of rich interactions.

**TRACK 7 Agile in the large**

**Nathaniel T. Schutta**

Almost every example of an Agile project involves a single team and while many successful projects are delivered that way, most enterprise software requires the interaction of several teams. But how do we scale Agile beyond a single team? What practices translate and which ones don't? In this talk we'll discuss some of the issues you'll encounter as you move Agile beyond a single group and how you can keep multiple stakeholders happy. While it isn't as simple as having a "scrum of scrums" it isn't as hard as replacing every line of COBOL.

**TRACK 8 Patterns for the people**

**Kevlin Henney**

Apparently, everyone knows about patterns. Except for the ones that don't. Which is basically all the people who've never come across patterns... plus most of the people who have.

Singleton is often treated as a must-know pattern. Patterns are sometimes considered to be the basis of blueprint-driven architecture. Patterns are also seen as something you don't need to know any more because you've got frameworks, libraries and middleware by the download. Or that patterns are something you don't need to know because you're building on UML, legacy code or emergent design. There are all these misconceptions about patterns... and more.

In this talk, let's take an alternative tour of patterns, one that is based on improving the habitability of code, communication, exploration, empiricism, reasoning, incremental development, sharing design and bridging rather than barricading different levels of expertise.

16:00-17:30

### TRACK 1 Patterns for parallel programming

Andrew Clymer

"The free lunch was 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 that parallelizing anything 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 meal you will have to deploy a range of tricks for your algorithm to take full advantage of those multiple cores.

### TRACK 2 Windows and Windows Phone – building for both with C#

Mike Taulty

If you're building modern apps for Windows and Windows Phone then you'll know that there are challenges in re-using code and XAML across the two platforms. In this session we'll take a hands-on approach using code-based demos to investigate what code can be shared across Windows and Phone, and show some techniques for getting the most from that sharing

### TRACK 3 Securing ASP.NET Web APIs

Dominick Baier

If you want your services to be usable by arbitrary clients like browsers and mobile/desktop native apps, you need to model them as "web APIs".

ASP.NET Web API is the framework from Microsoft to build such services and software systems with .NET. To effectively secure web APIs you need to understand the fundamental application types, HTTP security and a bunch of new technologies like OAuth2 and JSON Web Tokens.

We'll walk you through typical web API architectures and their security requirements, and explain how you can implement these applications using the .NET stack on the server and any technology on the client.

### TRACK 4 The HTML5 canvas API

Jeff Prosize

The canvas API brings rich 2D graphics to the Web platform and is universally supported by HTML5 browsers. Learn how the API works and how you can use it to build rich, interactive UIs that wouldn't have been possible just a few years ago.

### TRACK 5 Modern JavaScript

K. Scott Allen

JavaScript is a dynamic, functional, ubiquitous language that has many hidden secrets. In this session we will take a deep look at the core JavaScript features that many contemporary libraries leverage, including constructor functions, modules, closures, hash parameters, method chaining, and more.

Having a solid grasp of these features will not only help you write more maintainable JavaScript code, but also allow you to take greater advantage of today's JavaScript libraries.

### TRACK 6 Smart UX design for smart phones

Tobias Komischke

Designing for smart phones is not the same as designing for desktop or web applications. While many fundamental user experience design principles can be applied 1:1, there are unique factors that impact the UI design. This session provides concrete and applicable design considerations specifically for smart phones. The following topics will be addressed together with examples:

- Special considerations for small form factors (navigation & user flow, providing orientation, rich interactions, notifications vs. alarms, data entry, text size etc.)
- Special considerations for the context of use (one hand usage, usage while walking, contrast and glare effects, environmental noise impact)
- Ways to mitigate the drawbacks of touch screens; recommendations on gestures and touch target sizes, etc.
- Walking the line between ideating novel design and adhering to phone

OS UI guidelines such as Microsoft's Metro style guide

### TRACK 7 Lean thinking and what it means to the Agile mindset

Howard Deiner

Long before the Agile revolution for software development began, industry had learned that efficient production of goods required intense attention to quality, teamwork, and continuous improvement. These themes of Lean Manufacturing (which was further refined into the Toyota Production System) were never part of the original formulation of the Agile Manifesto, and are rarely mentioned as part of the traditional Agile/Scrum recipe for teams transforming to the new "Agile" mindset.

The reality is that the traditional Agile/Scrum recipe is actually a "dumbed down" version of the Toyota Production System, and makes it easier for organisations to grasp and start from. However, if organisations really want to achieve the goal of producing the software they need in a fashion that leads to High Performance Teams and Sustainable Engineering, they will need to understand the principles of Lean so they can incorporate them into their unique process. This session teaches the basics of Lean, and demonstrates how they apply to Agile development.

### TRACK 8 The architect

Juval Löwy

The software industry is in a deep crisis. Projects suffer from low quality; schedule slips and cost overruns are common. But this is hardly ordained, since there are plenty of examples of projects done on schedule, on budget, on quality in a repeatable manner. The key for solving the crisis is to practice software development as an engineering discipline, just like it is done in mechanical or electrical engineering. That is exactly the responsibility of the modern software architect, who must address software systems from an engineering standpoint, applying age-old engineering practices to software development. Armed with that crucial observation Juval will share his perspective on what is the key set of skills required of the architect, the role architects have to plan in order for the project to succeed, what it implies both to companies and to individual architects, and how should architects navigate and manage their own career path and personal growth.

Wed 21 May 2014

9:30-11:00

### TRACK 1 Death to the procedure: modern C# is non-linear

Oliver Sturm

Or at least it looks that way. Are you among those programmers who still code lots of loops? Especially the kind that don't even have a "yield" statement somewhere? Do you frequently use the Add method on collection types?

These are ideas of sequential programming, and that's outdated – seriously, it's true.

Come to this talk and learn more about the background to this statement!

### TRACK 2 Writing readable and maintainable code

Dino Esposito

When it comes to code readability, consistency is more important than actual conventions, but conventions determine for the largest part the level of readability. Conventions apply to naming of methods, classes, and variables, quantity and style of comments, length of methods, (mis)use of patterns. This session discusses the dos and don'ts.

### TRACK 3 18 ways your brand new ASP.NET MVC project can be better

Michael Kennedy

So you're ready to start that new and ambitious ASP.NET MVC project. Maybe you're kicking off a new start-up, or just finally moving that old-and-crusty Web Forms project into the modern development world.

Either way, this talk will give you some easy things you can do immediately after creating that new MVC project that you will thank yourself for as your project grows in complexity.

### TRACK 4 HTML5 Web Workers: multithreading for the web

Jeff Prosize

The Web Workers specification brings multithreading to JavaScript, allowing HTML5 apps to use background threads to make applications more responsive to user input and better able to leverage today's multi-core CPUs. And it's now supported in all major browsers, including Internet

Explorer. Learn how to use Web Workers to build multithreaded HTML5 apps and see examples of Web Workers in action.

### TRACK 5 The JavaScript developer's toolchain

Nathaniel T. Schutta

Back in the day, web developers had to rely on their wits and a plethora of alert statements – to say our toolkit was spartan would be an understatement. But with the increased importance of web front ends and the rise of JavaScript MVC frameworks, a modern web developer toolkit is finally emerging. We've evolved from text editors to full fledged IDE's with code completion and refactoring tools but our toolchain doesn't end there. With multiple testing libraries, mocking frameworks, test drivers and even code coverage tools, today's web developer gets to walk downhill on a sunny day. In this talk, we'll discuss the various tools that you can assemble into your own full fledged JavaScript development pipeline from code to deployment.

### TRACK 6 Functional thinking

Neal Ford

Learning the syntax of a new language is easy, but learning to think under a different paradigm is hard. This session helps you transition from a Java writing imperative programmer to a functional programmer, using Java, Clojure and Scala for examples. This session takes common topics from imperative languages and looks at alternative ways of solving those problems in functional languages. As a Java developer, you know how to achieve code-reuse via mechanisms like inheritance and polymorphism. Code reuse is possible in functional languages as well, using high-order functions, composition, and multi-methods. I take a variety of common practices in OOP languages and show the corresponding mechanisms in functional languages. Expect your mind to be bent, but you'll leave with a much better understanding of both the syntax and semantics of functional languages.

### TRACK 7 Agile architecture: Part 1

Allen Holub

Successful Agile projects have an architecture that reflects the needs of the process. You simply can't accommodate changing requirements or refactor effectively if your system isn't designed to accommodate those practices.

This class shows how good (and bad) structure affects agility, and explores design principles that keep your design on track in an Agile environment.

We'll look, specifically, at the SOLID principles (Single-Responsibility, Open-Closed, Liskov-Substitution, Dependence-Inversion, and Interface-Segregation Principles), and at how design patterns improve agility, and at several common practices that can sabotage your project.

We'll also look at Fragile Base classes and inheritance-related structure issues.

### TRACK 8 Driven to tests

Kevlin Henney

These days testing is considered an increasingly sexy topic for programmers. Who'd have thought it? What is the motivation for unit testing? And what constitutes a good unit test? There's more to effective unit testing than just knowing the assertion syntax of a framework.

Testing represents a form of communication and, as such, it offers multiple levels and forms of feedback, not just basic defect detection. Effective unit testing requires an understanding of what forms of feedback and communication are offered by tests, and what styles encourage or discourage such qualities.

What style of test partitioning is most common, and yet scales poorly and is ineffective at properly expressing the behaviour of a class or component? What styles, tricks and tips can be used to make tests more specification-like and can scale as the codebase grows?

11:30-13:00

### TRACK 1 Inside the Garbage Collector

Richard Blewett

For many .NET developers the GC generally works in the background largely unnoticed. However, a clear understanding of how the GC works and is tuned is critical to writing efficient .NET code, and has a big impact on how you design your code.

This talk looks at the GC in terms of how it works and is tuned, and gives recommendations about practices that will either help or hinder the GC.

### TRACK 2 Lightweight web-based architectures with Web APIs and push services

Christian Weyer

Let's face it, current trends and developments especially in the area of mobile platforms & devices and cloud computing will cause a re-thinking in architectural aspects for many software projects. If you ignore this today you may be in big trouble tomorrow. How can I integrate my systems and application parts in a lightweight and interoperable manner? How am I able to push data in near-real-time fashion from my services to my clients?

Christian Weyer will show you in a pragmatic way how to face these new challenges. Come and see all of this coming to life by using technologies and frameworks like ASP.NET Web API, SignalR, .NET- and HTML5-based clients – mobile or not.

### TRACK 3 Identity management in ASP.NET

Brock Allen

Identity management is the unglamorous (yet extremely important) task of managing account information for users (including passwords). Identity management in ASP.NET has had an interesting and somewhat tumultuous history. In this session we'll discuss the important characteristics of proper identity management, and then we'll survey the pros and cons of the various offerings in ASP.NET including membership, simple membership, and the most recent iteration with ASP.NET Identity.

### TRACK 4 jQuery fundamentals

Robert Boedigheimer

jQuery provides a very productive environment for client side programming in JavaScript and is used by a majority of major web sites today. It takes advantage of existing knowledge of CSS selector syntax to offer a powerful and efficient alternative to accessing elements. The use of operation chaining and implicit iteration lead to a very compact and productive syntax. The library is very lean at a mere 32K, yet provides a strong base and a great extensibility model which has led to a large number of plugin extensions to simplify web development. This session will review how to use the library for very useful features while avoiding browser inconsistencies, and making AJAX calls to the server. Several plugins will be demonstrated which provide stunning client experi-

ences with as little as one line of code! Learn how jQuery greatly simplifies client side development.

### TRACK 5 A finger is not a fat mouse pointer!

Dave Wheeler

Touch is one of the most misunderstood aspects of modern user interface design. In this talk, you will learn how to build user interfaces that work with all forms of input: mouse, keyboard, stylus and touch.

You'll see how to adapt the UI to work across different form factors; consider aspects such as momentum and inertia; and see how with a small amount of work you can create a truly responsive interactive UI.

### TRACK 6 From the user to the unit – how to start using tests

Kevin Jones

One of the key questions in testing is "how do I start?". In this talk we will try and answer that question. Tests should start with features, so the talk will briefly cover requirements gathering in the form of user stories, and how to turn those into scenarios and features. From these scenarios we can generate the initial set of acceptance tests and then start to home in on the code. From this we will discover how the unit tests should drive the code forward until the feature tests pass. All the while we will keep the process in mind and see how you can make this cycle into your development process.

### TRACK 7 Agile architecture: Part 2

Allen Holub

This session builds on Agile Architecture Part 1. We'll look in depth at Implementation Hiding and Encapsulation issues, focusing on accessor/mutator (getter/setter) systems and their problems. We'll see how to solve those problems through user-focused design, looking in depth at how a properly defined Agile story becomes a high-quality OO design that both makes it easy to accommodate business-requirement changes and also eliminates the need to violate Encapsulation.

### TRACK 8 Consequences of Emergent Design

Neal Ford

At ThoughtWorks, we're big fans of evolutionary architecture and emergent design, which allows great technological and business flexibility. But like many accelerants, it isn't entirely free. This talk explores decisions made and consequences (both positive and negative) from a real world project that has used these techniques aggressively for four years. I discuss the distinction between architecture and design, the impacts of change on each, and how you can sometimes exchange them.

14:00–15:30

### TRACK 1 Power debugging

Andrew Clymer

For many developers, their use of debugging tools starts and ends with Visual Studio. However, there are a large number of problems for which Visual Studio provides very little support – particularly threading and memory management issues.

WinDBG and the plugin SOS.DLL bring a new set of tools to .NET developers which provide insights that help you solve bugs that you see during testing, but also allow you to diagnose issues occurring in production systems where the only data you can get is a crash dump file.

### TRACK 2 Windows Azure in the real world: from idea to production in a few months

Jesus Rodriguez

I've been developing, speaking and writing about the Windows Azure platform since its first release. I thought I was an expert on the subject until a few months ago, when I decided to start working on what has become one of the biggest platforms running on Windows Azure. Kidozen is an enterprise mobility platform currently powering hundreds of mobile applications and processing millions of messages every month. What did I learn? This session will explore the best practices, patterns and techniques that we've learned when building large scale, multi-data-centre Windows Azure applications in the real world. We will deep dive into some of the traditional patterns of Windows Azure applications such as data storage, messaging, caching, security, monitoring among many others. Additionally, we will cover some of the best practices and tools that you can use to

streamline your Windows Azure application development in areas such as testing, continuous deployment, provisioning, etc. To keep things practical, we will provide a series of demos that will clearly illustrate the patterns and techniques covered in this session.

### TRACK 3 What's new with ASP.NET MVC 5

K. Scott Allen

This session will look at new features in the new ASP.NET MVC 5 framework as well as the framework's supporting cast. From attributed routing to tips and tricks for the new Visual Studio editor, the goal is to provide experienced ASP.NET MVC 5 developers with all the information needed to move to the next version of ASP.NET and the MVC framework.

### TRACK 4 Web security: threats and mitigation

Brock Allen

This session on security describes many of the common attacks against websites, including SQL injection, cross-site scripting and cross-site request forgery attacks.

We will see what ASP.NET provides to thwart or mitigate these and other types of attacks. We'll also look at many of the common vulnerabilities in applications which are unknowingly introduced by application developers themselves, and how to remedy these problems.

### TRACK 5 Is JavaScript maintainable?

Hadi Hariri

JavaScript, the language that we all laughed at when our usage of it was limited to copy/paste of scripts to have some dynamic content in a web page, has all grown up. From jQuery on the client-side to Node.js on the server, everywhere you look there's JavaScript. Unfortunately there's also a ton of horrible, twisted and somewhat convoluted code that would make any developer that remotely cares about clean code suicidal.

A dynamic language that has somewhat given us a licence to do anything we want has been abused to the point that much of the blame has been placed on the language itself as opposed to the developers. In this talk we'll discuss some of the patterns and anti-patterns of JavaScript, things you should and shouldn't do. If you care about writing clean and maintainable

code, you should care about it in any language!

### TRACK 6 The architecture of uncertainty

Kevin Henney

Ralph Johnson once defined architecture as "the decisions that you wish you could get right early in a project, but that you are not necessarily more likely to get right than any other". Given our inability to predict the future, how can we design effectively for it?

Much project management thinking is based on the elimination of uncertainty, and advice on software architecture and guidance for future-proofing code often revolves around adding complexity to embrace uncertainty. In most cases, this is exactly the opposite path to the one that should be taken, turning project plans into works of substandard fiction and weighing codebases down with speculative generality and technical debt. Rather than assuming that uncertainty is a problem to be fixed, we should instead view it as an important and equal input to an ongoing process of design and planning.

The talk looks at how uncertainty, lack of knowledge and options can be used to partition and structure the code in a system and the schedule of its development.

### TRACK 7 Agility at scale – platform versus product concerns

Howard Deiner

A common failure mode for organisations attempting to adopt an Agile style of software development occurs when an attempt is made to "Scale Agile". Suddenly, the organisation finds that there are scheduling problems between teams. Delivery team members suddenly find that they are required to serve on several teams at once. Dependencies surface, and teams find it difficult to come together in a common cadence to produce working software in a continuously delivered fashion. Many times, these issues become so grave that the organisation reverts back to the Waterfall model that they came to hate, but at least understood.

This session explores Agile scaling concerns, and places particular emphasis on an architecturally significant distinction in the software to be created, and the components produced to allow the software to be

created. That distinction revolves around cross cutting platform concerns versus product feature creation concerns. We will examine the distinctions and explore solutions that should help your organisation get past these issues when it comes to portfolio management, by paying attention to extrinsic versus intrinsic value metrics.

### TRACK 8 Software project design

Juval Löwy

Much as the need to design the system, you must also design the project: from scheduling resources behind the services, to tracking your progress across developers, services and phases of completion, to validating your plan, and presetting the project design options to management. The techniques of software project design are specific to software projects, but in the abstract there is nothing new. To use an analogy, if I ask you to design a system that is maintainable, reusable, extensible, secure (or safe), and of high quality, you cannot tell if I am talking about a mechanical system or a software system. Much the same way, if I ask you to design a project that will comply with a set budget and deadline, within acceptable risk, be traceable and manageable, you cannot tell if I am talking about a bridge or an ERP system. And not surprisingly, in the abstract, the techniques for designing both types of projects are identical. Software project design requires understanding the inner dependencies between services and activities, the critical path of integration, the available floats, the staff distribution and the risks involved. All of these challenges stem from your design and addressing them properly is a hard core engineering task – designing the project. Time permitting, you will see showing how to close the loop by tracking both progress and effort across developers and services and estimating the impact of changes throughout the project, allowing you to constantly stay on schedule and on budget.

16:00–17:30

### TRACK 1 Identity and access control for .NET 4.5 applications – and beyond

Dominick Baier

Every serious application needs authentication and authorization – and the .NET framework has provided built-

in facilities for that since its very first version. Starting with .NET 4.5, Microsoft completely revamped the built-in identity and access control plumbing and APIs, and integrated the notion of token-based authentication and claims into the base class library. These important base technologies are actually the foundation for the new application types and scenarios you going to implement in the years to come using the brand new Katana security runtime. How can we take advantage of these new features? For which types of application do they make sense? What new scenarios can be enabled using these technologies?

### TRACK 2 High performance NoSQL techniques

Michael Kennedy

You're one of the brave ones who has jumped into the NoSQL pool and found it a refreshing change. That's awesome. But there is so much more to being successful with NoSQL databases than simply getting started.

In this session we'll explore some of the issues, techniques, and best practices for being successful with NoSQL in general (and MongoDB in particular).

This includes exploring correct document/entity design, indexes, deployment, just to name a few of the topics. If you're getting started with NoSQL this session should help you take things to the next level.

### TRACK 3 Async ASP.NET

Brock Allen

Asynchronous programming is all the rage these days, was a major theme in the release of .NET 4.5, and is all but mandatory for any modern development project.

In this session we'll look at what async means for web applications for both client and server code. We'll see how ASP.NET has had async support for over 10 years, and how WebForms, MVC and WebAPI provide an easy-to-use async programming model.

We'll then look at async client programming using JavaScript and jQuery, HTML5 WebWorkers, and we'll conclude with the ever-popular SignalR library for supporting async communication.

### TRACK 4 HTML5 storage: web storage and Indexed DB

Jeff Proside

HTML5 provides two mechanisms for storing data on the client side: Web

storage for unstructured data, and Indexed DB for structured data. Learn about both of these storage APIs and how you can put them to work building cutting-edge Web apps.

### TRACK 5 Sci-Fi UI

Dave Wheeler

Why is it that UIs in films always look so cool, yet the stuff on the PC that you work with on a daily basis looks so bad?

In this talk, you will learn how to master styles, templates, triggers and VSM to the extent that you can create out-of-this-world application interfaces.

Be prepared to be blown away.

### TRACK 6 4 practical uses for Domain Specific Languages

Neal Ford

Domain Specific Languages seem like a cool idea, but where's the payoff?

This talk provides an overview of how to build both internal and external DSLs (including the state of the art tools), stopping along the way to show how this is practical to your day job. This talk defines DSLs, distinguishes the types of DSLs (internal and external), and shows examples of building DSLs of several kinds. It shows how to utilize DSLs for externalizing configuration (which you're already doing, whether you realize it or not), how to make your code readable to humans, how DSLs make developer tools better (and how to use DSL techniques to build your own tools), and how DSLs can provide your users unprecedented flexibility and power, by building DSLs customized to their job. This talk provides a good foundation for the subject if you've never seen anything about it, but keeps the focus on practical goals.

This talk provides an overview of how to build both internal and external DSLs (including the state of the art tools), stopping along the way to show how this is practical to your day job. This talk defines DSLs, distinguishes the types of DSLs (internal and external), and shows examples of building DSLs of several kinds. It shows how to utilize DSLs for externalizing configuration (which you're already doing, whether you realize it or not), how to make your code readable to humans, how DSLs make developer tools better (and how to use DSL techniques to build your own tools), and how DSLs can provide your users unprecedented flexibility and power, by building DSLs customized to their job. This talk provides a good foundation for the subject if you've never seen anything about it, but keeps the focus on practical goals.

### TRACK 7 Keep Agile development from becoming Fragile development – recognising and dealing with technical debt

Howard Deiner

For too many organisations, the Agile process is put into place and the Agile Manifesto principle of "continuous attention to technical excellence and good design enhances agility" is never addressed. This leads to a situation where, even when the delivery team does a good job at first, their ability to sustain delivery pace over time is

compromised because of what is commonly called "technical debt".

Organisations struggle with how to pragmatically address technical debt. How do we recognise its existence?

When is the right time to refactor? When areas of concern are all competing for the same resources, which areas do we address first?

This discussion will explore these dimensions of the problem and provide actionable advice to begin the conversations within your organisation to start to face the facts in this important area of software development. Because, without addressing the issue, you will, at some point, find that you are no longer doing Agile development, but rather Fragile development.

### TRACK 8 Critical path essentials

Juval Löwy

Whether you acknowledge it or not, the entire project design revolves around its critical path: the chain of activities whose duration and dependencies represents the quickest possible way the project could ever be built. The project manager must identify the critical path and assign resources against it. But there is much more to the critical path. As a logical representation of project for planning and tracking it is a model admirably suited for complex software systems, providing accurate implications of change on cost, schedule and risk throughout the project. Determining and analyzing the critical path is a highly engineered design task requiring both the project manager and the architect to objectively work closely together to determine the best overall plan. This session presents the basics of the critical path method, universal concepts independent of your project, and the recurring techniques and terms required to design your project.

## Thu 22 May 2014

9:30-11:00

### TRACK 1 It's a kind of magic

Andrew Clymer

In the early days of C# there was virtually a 1-1 mapping between C# and IL. Much has changed since C# 1, the developer is encouraged to simply describe their intent and the compiler builds the appropriate code.

This talk will dive under the hood and show how various C# features like iterator methods, anonymous methods, extension methods and dynamic all result in code gen by the C# compiler.

### TRACK 2 Object orientation – when and why?

Oliver Sturm

The title of this talk suggests that sometimes you're better off without it – yes, without object oriented programming. Most programmers understand this instinctively, but occasionally it can be useful to sit down and ponder the different approaches and techniques at our disposal.

Object oriented programming is a tool, its use can pay off sometimes – but that's all! This talk is about the When and the Why of it.

### TRACK 3 Universal web apps: one site for multiple UXs and devices

Dino Esposito

It's a common opinion that you only have two options to let users enjoy a website from mobile devices: arranging a dedicated site or building a single responsive site which can adapt rendered content via CSS and media queries. This vision of the world, though, is quickly getting obsolete as it misses most of the role that devices have at present in the industry. A website agnostic of devices that just fits content into differently sized screens is too limiting. Following the universal vision of apps that Microsoft is applying to Windows, this talk presents an approach to mobile website development that is universal and covers a selected number of device profiles like smartphone, tablet, wearable and desktop as a special case of the device. In more detail, we'll see how to leverage device detection and ASP.NET display mode facilities to build a single website that switches views intelligently based on the capabilities of the detected device.

### TRACK 4 Web development with Node.js

Hadi Hariri

If you're coming from a .NET, Ruby or Java background and want to get up to speed in creating web applications in Node.js, this is the perfect session for you. We'll learn the basics of Node.js and how to get up and running creating web applications in no time with Node.js. We'll dig into code organisation, something very important in JavaScript, routing, view engines, models, authentication, session management as well as cover some of the more useful and important packages in the Node.js ecosystem.

### TRACK 5 Build your first Windows Phone app

Jeff Proside

Never built a phone app but interested in learning what it takes? Follow along as Jeff covers the essentials of Windows Phone app development, demonstrates how to build native wrappers around cross-platform HTML5 assets, and highlights some of the exciting new features in Windows Phone 8.1.

### TRACK 6 Design patterns in the real world

Allen Holub

Design patterns do not exist in splendid isolation. 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.

### TRACK 7 Introducing unit testing to legacy code

Richard Blewett

Legacy code presents a problem for introducing unit tests: you generally can't test without changing the code,

but if you change the code you have no tests to verify that it still works.

This talk lays out some techniques for introducing tests into your legacy codebase using coding patterns and tools like Microsoft Fakes shim support.

### TRACK 8 Zen of architecture

Juval Löwy

For the beginner architect, there are many options for doing pretty much anything. But for the master architect, there are only a few.

In this dense session, master architect Juval Löwy will explain his approach to large system analysis design, how to decompose a system into its comprising services, and reveal the most common mistake made in architecture.

Time permitting, Juval will discuss logical tiers, security, interoperability, scalability, transactions, and other aspects of a modern application.

11:30-13:00

### TRACK 1 An introduction to Python for the C# developer

Michael Kennedy

Python is used by all sorts of organisations: Google, NASA, Industrial Light and Magic and many more.

This talk looks at what Python brings to the table, and provides an introduction to programming with Python for people with a background in C# or Java.

### TRACK 2 Building universal apps for Windows, Windows Phone, and other platforms

Jeff Proside

One of the more exciting pronouncements at Microsoft's BUILD conference in April 2014 was the introduction of "universal apps", which run on PCs, tablets, phones, and, according to Microsoft, will one day run on Xbox as well. But what are universal apps, and how are they written? This session will answer these questions and more, and provide a tantalising look at the future of software development on the Microsoft stack.

### TRACK 3 The best of ASP.NET Web Forms

Robert Boedigheimer

ASP.NET Web Forms is a very mature platform with great features and years of established best practices. Review

some of the lesser known features such as adapters, response filters, external config files, substitution caching and more. Learn best practices such as base pages, global exception handling, validation controls, caching, etc. Discover useful techniques such as server side viewstate, page control tree processing and how to utilize "safe" functions. Understand new features for ASP.NET 4.5 Web Forms like model binding and unobtrusive validation. Web Forms aren't dead, learn how to take full advantage of them for your web site.

#### TRACK 4 Building interactive clients with knockout.js

Kevin Jones

Client side development in the browser is becoming more and more important. Clients expect highly interactive applications that respond quickly to changes. To manage this developers have to cope with the demands of JavaScript. While jQuery is a fantastic tool, sometimes it feels too low-level, and maybe needs to have something layered on top. One such layering is knockout.js. This provides an MVVM programming model for the browser.

This talk looks briefly at the basis for MVVM and then into the details of knockout.j. We will cover the data-binding that is at the heart of this toolkit, looking for example at how knockout does data updates and computed values. We will also see how knockout can work with your server side coding.

#### TRACK 5 Working with jQuery Mobile

K. Scott Allen

jQuery Mobile is a JavaScript framework for building touch-optimised sites targeting smartphone and tablet devices.

In this session we'll take an existing ASP.NET MVC application and improve the mobile experience using jQuery Mobile. Along the way we'll see how the widgets, events, API, and navigation features of jQuery Mobile work, and also see some pitfalls to avoid.

#### TRACK 6 It's all about the money, money...

Dave Wheeler

Monetization. It's how you make a living from the apps that you build.

In this talk you will learn how to work with the different models for making money out of applications; how to work with the different stores; and

most importantly how to monetize your app in an appropriate and user-friendly way.

#### TRACK 7 TDD and BDD (Test-Driven and Behaviour-Driven Development)

Allen Holub

TDD (and its specialization, BDD) are a dynamic design process particularly suited for Agile environments. That is, TDD/BDD is really a design, not a testing strategy, but it's a design strategy which produces a set of very useful tests. By building acceptance tests around tentative interfaces, then incrementally refining those interfaces using acceptance tests, you both design and code simultaneously in a way that yields an optimal design.

This class look at the basic principles (of both TDD and BDD). We'll walk through the process, showing how the design evolves as the process works. We'll also look at the role of mocking, and at how to use a mocking framework such as Mockito, PowerMock, or MOQ to quickly build tests around abstractions.

#### TRACK 8 Advanced techniques in project design

Juval Löwy

If project duration is of no consequence, you should design your project for lowest cost. But if cost is immaterial, you should design it for the least duration. The best solution for every project lies between these two extremes, yet finding it is a highly engineered design task. This session will provide you with the tools and techniques required to achieve a balanced and even optimal solution, enabling you to determine best overall plan for your system while eliminating gambling, death marches, wishful thinking, and expensive trial and errors. The sessions assumed knowledge of basic critical path concepts such as network design, resource allocations and floats, and will expand on these essentials to include finding your project time/cost curve, network compression techniques, activity crashing, risk analysis, staffing distribution options and project design. You will also see a comprehensive case study and walk through its various permutations in determining the best plan that will keep the project on time all the time at the best risk and cost available.

14:00-15:30

#### TRACK 1 Creepy C#

Oliver Sturm

Writing cool and pretty code is easy – anyone can do that! Hmm, or maybe not. But we're all great at laughing about really scary code, although it can be useful to spend some time thinking about it as well...

In this talk we'll look at anti-patterns and similar constructs that can scare a C# developer.

There will be laughter, I promise!

#### TRACK 2 User-centered data visualization

Tobias Komischke

"Get rid of the numbers and get me some graphics!" Data Visualization is an ongoing trend in the industry because presenting data graphically allows for quicker understanding for those who interpret the data. Yet, from a usability angle some visualization forms are better suited than others for specific questions and uses. Also, applying standard software ergonomic rules like color treatments, label orientation etc, can significantly boost the effectiveness of a visualization. This presentation provides an overview of the topic and shows through many examples what you should look for when designing data visualizations.

#### TRACK 3 Design considerations for ASP.NET applications

Dino Esposito

The controller is the central element of any operation you perform in ASP.NET MVC. The controller is responsible for getting posted data, executing the related action and then preparing the view. More often than not, these apparently simple steps originate a lot of code. Worse yet, similar code ends up being used in similar methods and similar helper classes sprout up from nowhere. In this session, we'll explore an approach to ASP.NET MVC design that simplifies the implementation of the controller classes mechanizing the steps to take.

#### TRACK 4 Bootstrap for web developers

K. Scott Allen

Bootstrap is a front-end framework for web applications that you can use to build applications in less time. In this session we'll explore the core features of Bootstrap and see how to add the framework to an existing application.

We'll work with layout features and UI widgets, as well as see how to implement a responsive design to make a site look good on a variety of devices. As a bonus, we'll see how you can not only extend and customize Bootstrap, but also learn how to design a flexible CSS framework.

#### TRACK 5 JavaScript libraries you aren't using... yet

Nathaniel T. Schutta

You're all over jQuery – you write plugins in your sleep – and before that, you were a Prototype ninja. Your team treats JavaScript like a first class citizen, you've even written more tests than Kent Beck. Is that all there is in the land of the JavaScript developer? Believe it or not, the JavaScript party hasn't stopped. What other libraries are out there? What do they offer? This talk will survey the field of modern JavaScript libraries getting you up to speed on what's new. We'll dive in just deep enough to whet your appetite on a wide variety of libraries such as Backbone, Underscore, Zepto and more.

#### TRACK 6 User-interface architectures in OO systems: alternatives to MVC

Allen Holub

Model-View-Controller (MVC) is a hoary architecture, dating from the late 1960's. Frankly, we've learned a lot since then. MVC works fine for small things (implementing widgets, for example), but it doesn't scale well, is hard to maintain, and more to the point, doesn't really satisfy its main design goal: separation of concerns. This session will look at MVC, both strengths and failings, and provide several more robust UI architectures that are more appropriate for contemporary cloud-based systems.

#### TRACK 7 Design for testing

Kevin Jones

Many developers now recognise the importance of unit testing, and either use Test First or Test Driven Development. However, many developers and architects are still struggling to come up with designs that enable easier testing of their applications.

This talk will examine some common unit testing problems and use various unit testing patterns to solve those problems.

#### TRACK 8 Building enterprise-ready mobile apps in the real world: a developer deep dive

Jesus Rodriguez

Are you considering building mobile applications in the enterprise? Concerned about selecting the right tools or frameworks? Thinking how to effectively test, manage, distribute and enable enterprise-ready backend capabilities on your mobile apps?

This session explore patterns, best practices and techniques for building mobile applications in the enterprise. We'll take a holistic look at the mobile development ecosystem, exploring platforms like PhoneGap, Sencha, Xamarin and others from an enterprise perspective. Additionally, the session will highlight patterns and techniques for effectively enabling different mobile application lifecycle aspects such as testing, distribution, monitoring and other components of an enterprise mobile solution.

16:00-17:30

#### TRACK 1 Regular expressions in .NET

Robert Boedigheimer

One of the most overlooked features in many languages is the support for regular expressions. There are many times where they are the perfect solution, yet many if not most programmers fear them. Well, fear not, we will review the basics of regular expressions and where they can be utilized effectively. We will study named groups which allow you to not only match expressions, but capture the a portion of the matched expression to be used for other purposes. We will then review some of the more advanced features in .NET, and how things like compiled regular expressions can make programs perform extremely fast. If you ever knew that regular expressions were the right solution but covered away, come and face your fears...

#### TRACK 2 The computer says NO!

Dave Wheeler

There are great applications, and applications that suck. Often, the difference is simply down to the user experience.

In this talk you will see examples of excellent, and atrocious, UX design. With some laugh-out-loud demonstrations, and a lot of serious points, this interactive session (with prizes!) will make you look at software in a whole new light.

#### TRACK 3 Building rich input forms in ASP.NET MVC

Michael Kennedy

ASP.NET MVC has gained broad adoption over the last year. This is in part due to its clean and simple design. However, one aspect that newcomers typically get hung up on is building pages that accept user input in various manners.

In this talk we'll explore the powerful features of ASP.NET MVC that allow us to build rich forms that accept user input. We'll begin by discussing the built-in HTML Helpers and Model Binding. Next we'll add validation and show how we can do both client- and server-side validation using DataAnnotations. We'll see that sometimes using domain models as our form-bound objects doesn't make sense and so we will cover more advanced scenarios using View Models. Finally, if time permits, we'll see how client-side programming with JavaScript and jQuery can take this even further.

#### TRACK 4 Lesser known features in HTML5

Brock Allen

HTML5 has a huge number of features and many of them go unnoticed. This session is intended to open your eyes to some of these lesser-known features.

We will look at drag and drop support, the various streaming APIs which include file system, microphone and webcam access, networking features in XMLHttpRequest2 and WebRTC for real-time communication, the APIs related to the browser history and forms validation, SVG support, and if there's time some of the lesser-known CSS3 features and APIs.

#### TRACK 5 Backbone 101

Nathaniel T. Schutta

Model View Controller isn't just for the server anymore, in fact, many developers utilize those proven ideas inside the browser. Today's rich internet applications leverage client side frameworks and while there are more than you can shake a stick at, Backbone has emerged as one of the leaders of the pack. In this talk, we'll discuss the basics of client Model View Whatever walking you through the ins and outs of using Backbone to build richer, more responsive, interfaces.

This talk is aimed at experienced developers interested in learning more about how to develop modern web applications. Today's apps leverage JavaScript

to a level not seen before, but how do we do that and maintain our sanity? After this talk, you'll see how a lightweight framework can simplify the development of modern applications.

#### **TRACK 6 Web application architecture: the whole stack**

**Allen Holub**

This class provides an integrated look at the architecture of an entire "vertical slice" of the web-application stack, from the UI at the top to the database at the bottom, and all of the interior plumbing (messaging, caching, etc.) in the middle. We'll discuss various trade-offs and alternatives (several UI architectures are available, for example) as well as specific technologies that you can use to implement various components, and you'll see how all the pieces work together.

The focus is on good object-oriented structure throughout.

#### **TRACK 7 Modular enterprise development: minimising infrastructure efforts**

**Hadi Hariri**

We spend more development time doing busy-work, setting up infrastructure code and concerns than we do focusing on the problem we are trying to solve. We are not here to create frameworks but to help the business to be more productive. Many of us know this, but despite our efforts to avoid it, we continue ending up in the same situation time after time.

With a little bit of effort and following a series of principles in designing systems, we can however end up with a modular infrastructure setup whereby we can pull in different pieces as required and somewhat have things automatically wired up, avoiding the need to create heavy frameworks that never quite adapt to our different application requirements, and at the same time, not spending resources on this code of code. All it takes is some design principles and a few bits here and there.

#### **TRACK 8 Build your own technology radar**

**Neal Ford**

ThoughtWorks' Technical Advisory Board creates a "technology radar" 3 or 4 times a year, a working document that helps the company make decisions about what technologies are interesting and where we will spend our time. This is a useful exercise both for you and

your company. This session describes the radar visualization, how to create litmus tests for technologies, and the process of building a radar. You need two radars. As an individual, a technology radar helps guide your career decisions and focus your precious R&D time. For your company, creating a radar helps you document your technology decisions in a standard format, evaluate technology decisions in an actionable way, and create cross-silo discussions about suitable technology choices. Attendees will leave with tools that enhance your filtering mechanisms for new technology and help you (and your organization) develop a cogent strategy to make good choices.

**Fri 23 May 2014**

#### **All-Day Post-Conference Workshops**

**9:30-17:30**

##### **POST-01 From zero to app with NoSQL, MongoDB and .NET**

**Michael Kennedy**

Do you want leave this conference ready to build the next generation of maintainable, high performance applications in .NET? Then this is the post-con for you.

We assume you are a competent .NET developer but are otherwise basically new to NoSQL. We will build out an end-to-end data-driven web application in ASP.NET MVC and MongoDB (the most popular NoSQL database around these days).

Here are just some of the topics we'll cover:

- Why NoSQL, why document databases
- Installation quick start
- Your first app in MongoDB (.NET)
- Understanding the shell and native js query language
- Designing entities and models
- High performance techniques: indexes, profiling and plan detection
- Atomic operations, concurrency, and durability
- Management tools – may fold into 1 or 2
- Distributed file system: Grid.FS
- Performance / scaling mongodb (deployment techniques: sharding, replica sets, etc.)
- Map/Reduce

##### **POST-02 Designing and building applications with AngularJS**

**Kevin Jones**

Over the last couple of years there has been an upsurge of interest in frameworks for building applications in JavaScript for the browser. One of the most popular is AngularJS.

Angular comes out of Google. It can be loosely labelled as an MVC framework, but this label only constrains the definition. When using Angular you are almost compelled to use good coding practices, which as well as separation of concerns into Models, Views and Controllers, also means using Dependency Injection, View Models and being able to test.

In this workshop we will build an application from scratch. We will cover setting the initial framework code up;

creating controllers and models and binding them to the view; writing our own services; using the built-in services; understanding how to use build and use directives; and understanding how to test all our code.

##### **POST-03 Continuous delivery workshop**

**Neal Ford**

Getting software released to users is often a painful, risky, and time-consuming process. This workshop sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users.

Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers and operations, delivery teams can get changes released in a matter of hours – sometimes even minutes – no matter what the size of a project or the complexity of its code base.

The workshop materials are derived from the best-selling book Continuous Delivery, and were created in collaboration with the authors and other of my ThoughtWorks colleagues.

##### **POST-04 Building data-driven applications: a master class**

**Dave Wheeler**

Binding and M-V-VM. It's what we do when we want to build modern Windows applications. And you need to do it correctly.

Looking at the problems from both a developer's and a designer's perspective, you will see how to truly bridge the C#/XAML divide with effective use of converters, attachable properties, commands and behaviours. With a hefty dose of Visual State Manager and DataTemplates to match.

You will learn how to use bindings properly; how and when to use ViewModels; how to integrate applications with Prism (for both desktop and Windows 8 Store Applications); and how to use two of the most popular M-V-VM libraries.

This is a day packed with examples and practical guidance with code that highlights complex navigation and multi-view data-bound applications.

##### **POST-05 A modular approach to the development process**

**Juval Löwy**

Architects always strive for modularity to achieve extensibility, maintainability and reuse. The technology for the

modules, the glue, changes over time, from objects, to .NET to WCF. And yet, all of these introduce inherent complexity. When you design a highly modular system, it would be naive of you to expect that the only things you will do differently will be limited to design and technology. The development process itself needs to be modular, accommodating the complexity of the modules as a system.

In this intense session, Juval will share his original techniques, metrics and ideas, all practised in real life, allowing for a module-oriented development process. This talk presents a battle-hardened development process that you can immediately apply to achieve robust applications, manage requirements, and ensure faster time to market. You will also see how the various team members – the architect, the project manager and developers should work in concert, and their respective tasks and responsibilities towards the modules.

##### **POST-06 A day of identity & access control for modern applications**

**Dominick Baier & Brock Allen**

Modern applications typically consist of Web API based back ends that can support arbitrary client technologies. They are cross-platform and (enabled for) mobile by definition. Modern applications can't necessarily rely on enterprise technologies like Active Directory and traditional means of securing them don't apply anymore.

To effectively master the security challenges of modern applications you need to learn a new set of technologies like OAuth2, OpenID Connect and JSON Web Tokens and understand the trade-offs involved.

Learn how all those pieces of the puzzle fit together and how to secure modern applications using .NET on the server and arbitrary technologies on the client.

##### **POST-07 Agile/OO-design from start to finish**

**Allen Holub**

Many people who think they're doing OO aren't. For example, the dynamic model (which shows how run-time objects interact) should drive the design process; the class diagram is an artefact you build while doing dynamic modelling. Fixating on the class diagram renders your program at best unwieldy, at worst non-functional. Similarly, basic OO architectural goals (like eliminating getter/setter functions)

seem impossible to do unless you understand how the design process actually works.

It turns out that the process you use influences both the quality and the basic structure of the design.

This class covers an Agile version of the OO-Design process, with an emphasis on how to arrive at an optimal design. We'll provide a quick overview of the process, then spend much of the class working through one (or more if we have time) real-world examples that show you the entire process, from front to back:

- Requirements gathering and problem-statement definition
- Use-case analysis (story development), and the simultaneous construction of the dynamic and static models using UML

##### **POST-08 Node.js workshop**

**Hadi Hariri**

Spend this full day workshop learning about web development with Node.js, and in particular with Express.js, an extremely lightweight Sinatra-influenced framework for Node.js.

We'll learn the basics of HTTP, headers, bodies, status codes, requests and responses – everything that you should have known about web development but that has been abstracted from us by heavy-weight frameworks. We'll then dive into the architecture of Express.js, and cover routing, views, models, sessions, authentication and security among other things. We'll see how to drive web applications using TDD, never losing sight of using good practices to create maintainable applications.

##### **POST-09 A day of design patterns and testing**

**Andrew Clymer & Richard Blewett**

Re-use solutions, not just code. Design patterns help you to identify problems that occur repeatedly in your code, and solve those problems in a standardised way. The principles which design patterns introduce are also those that make your code testable. The combination of Test Driven Development and design patterns allows you to make sure that your code is functionally correct and has a robust design. Design patterns also give you useful tools to help test potentially complex code.

This day will in no way cover all OO design patterns, but will immerse you in the world of loose coupling and TDD and set you in good stead to continue your learning, resulting in you building fully testable Object-Oriented solutions.