

## PreCompiler Session 01 - Tuesday 8:00

### Machine Learning: Taking your ML Models to Android and iOS

*Wes Eklund*

Once you've developed a kickass Machine Learning model, you need a way to get that model to your computing devices (phones) to start doing your predictions! Most Machine Learning projects in production will 'train' the model on cloud servers, then 'deploy' the model to an API server or mobile device. This session will introduce the attendee on using TensorFlow Serving and Apple CoreML to deploy Machine Learning models to a mobile app.

**Prerequisites:** [Download Here](#)

---

### Build a Natural Language Slack Bot for your Dev Team

*Michael Perry*

Many project teams use Slack as a means of communication with one another. Why not also use it to communicate with your infrastructure? Invite a helper into your conversation that can perform routine tasks tirelessly, conversationally, and interactively.

In this 4 hour workshop, you will build a Slack bot that understands natural language and integrates with your DevOps pipeline. You will use the Slack Events API to receive messages, and the Slack Web API to send them. You will use LUIS to interpret language and extract intent. You will execute actions against the Visual Studio Team Services Web API in response to user requests, and subscribe to Webhooks to notify your team of important events. In the end, you will have a new member of your team who can help you with your build and release pipeline.

Workshop outline:

*Slack API Authorization - OAuth and API key verification Events API - respond to posts Web API - post messages Identify users and channels LUIS Specify intents and entities Train your app Evaluate utterances Call LUIS from your Slack bot VSTS Web API - interrogate source repository and build status Webhooks - respond to checkins and build completions Call services asynchronously to keep the conversation going Respond to the requester when events happen*

**Prerequisites:** [Download Here](#)

---

### Getting Real without Getting Fired -- Saying Things in a Way People Can Hear

*Allison Pollard, Marcus King*

Are important words often left unsaid at your place of work? Do you feel like you're navigating a complex maze in conversations? Does your message tend to miss the mark with co-workers, who increasingly seem to be impediments to reaching your goals? Are these unspeakable truths in your workplace that you wish someone would resolve for you?

Trust and communication issues within the workplace can hollow out an organization. Invisible lines get drawn. Alternate forms of communication open up to subvert perceived rivals. Allies are recruited, reinforcing an us vs. them behavior cycle. Organizations are suffering from a lack of trust, and it's costing them speed, productivity, and collaboration. What can YOU do about it?

Regardless of your title, you can be a leader in your organization, and a leader's first job is to inspire trust. In this session, Allison and Marcus will share models to evaluate your own behaviors and facilitate activities to help you find your voice for speaking the truth in a way that builds trust. Softening the truth can feel comfortable in low trust environments--it's simpler, nicer, and can make you look like a team player. It can also lead to miscommunication, undelivered news, and blame shifting. On the other hand, saying the truth in all of its ugliness is risky and potentially career-limiting. Finding the sweet spot of communication to become a trusted leader takes self-awareness and practice. Attend this workshop and learn to recognize how your behavior is building trust--or not--and practice speaking hard truths so that others can hear it.

---

## **Devour the Cloud with Locust Swarms - Hands on Load Testing**

*Steve Jackson, Nick Barendt*

Your website works great with a handful of users. What happens when there are 10k or 100k concurrent users? Where are the performance bottlenecks? How do you scale it? Before you can do that, you have to be able to consistently apply a simulated user "load" to the site, and measure the performance. In this session you will start with a functional Python Django web app running in AWS along with some vaguely defined load targets. From there we'll walk through establishing a user interaction model and implementing it in locust.io, a Python-based distributed load testing tool. That user model will be used to load the web app under test, establish baselines and pick some metrics and interactions to target. We'll then experiment with a few common scaling techniques to demonstrate quantitative improvements. You'll gain practical, hands-on experience with load testing concepts and methods.

Languages: Python

**Prerequisites:** [Download Here](#)

---

## **Build Cross-Platform Desktop Apps with Electron**

*David Neal*

Many developers are turning to Electron.js to build amazing desktop applications that run on Windows, Mac, and Linux using familiar Web technologies. In this hands-on workshop we will learn the fundamentals of building desktop applications with Electron.js. Our goal is to walk away with a solid framework for building and deploying your own applications!

Requirements:

- Laptop running macOS 10.8+, Windows 7+, or some modern version of Linux
- [Node.js](#) v6 or higher
- [Visual Studio Code](#)

**Prerequisites:** [Download Here](#)

---

## **Build Your Second Mobile App**

*Anne Cahalan*

You've done a "build your first mobile app" precompiler, or maybe some online tutorials. You've connected to an API and made something appear on a screen, even! It's time to move beyond that and dig a little deeper. In this precompiler, we'll explore two bigger topics in iOS development, one

foundational and one flashy. Protocol-oriented programming is a fundamental paradigm in Swift development and powerful way to extend functionality without the complications of inheritance. ARKit, announced at WWDC 2017, provides a platform for stunning new augmented reality experiences. We'll combine the two into a fun "an Elf On The Shelf game" game that will level up your Swift skills and provide a solid foundation for exploring iOS development even further.

**Prerequisites:** [Download Here](#)

---

## **Introduction to Concurrent Programming with Elixir**

*Rob Keefer*

Whether you have experienced the awesomeness of developing with Elixir and Phoenix, or simply want to learn how to take advantage of modern multi-core CPUs, the concurrency that Elixir provides is a great start to amping up the performance of your applications. In this hands-on workshop you'll be introduced to concurrent programming concepts and the methods for implementing these concepts in Elixir. At the end of the workshop you will have built a fault-tolerant, multi-process communication application.

Topics covered in this workshop include working with multiple processes, plus Elixir specifics such as GenServers, Supervisors and Workers.

**Prerequisites:** [Download Here](#)

---

## **Effective Gherkin: Quality Requirements for the Enterprise**

*Thomas Haver, Tony Makeka*

Developing software is costly. It is made more costly by miscommunication. Developers and Testers make mistakes because they misunderstand the business requirements. Business analysts make mistakes because they misunderstand the stakeholders. The result are defects that typically go unnoticed until the software has been implemented, which leads to expensive and time-consuming rework. Quality requirements are the nexus of understanding for stakeholders, designers, developers, testers, and customers. In this workshop, you will learn how to develop a set of quality requirement standards to reduce the cost of quality and meet the customer's needs. The training will leverage the industry practice of Behavior Driven Development, which promote requirements by example & collaboration, to develop a shared language across not only a team but the whole enterprise. The shared language is written in Gherkin format (Given, When, and Then specifications) to define requirements independent of application type and focused on delivering value that everyone in the organization understands.

**Prerequisites:** [Download Here](#)

---

## **First Person View Drone Racing (Day 1, Part 1)**

*John Chapman, David Resseguie*

Have you ever dreamed of flying? Have you seen videos of people piloting mini pod racers or watched drone racing on ESPN? Do you want in on the fun? Join us as we demystify these magical machines. In the first part of this session, we will learn all about the technology that makes quadcopters fly: flight controllers, motors, ESCs, receivers, transmitters, etc. Participants will then have the opportunity to build a micro sized racing drone and put their skills to the test in the ultimate

first person view Codemash Drone Prix. Will YOU be crowned the ultimate drone racing champion of Codemash?

**Prerequisites:** [Download Here](#)

---

## **Going all in with functional C#**

*Ed Charbeneau*

In this workshop attendees will learn about functional C# by building a poker scoring game using C# 7+ features. We'll learn what new features were added in C# 7+ to support functional programming. The pros/cons of functional vs. imperative programming will be explored to understand the strengths of each style.

In this workshop we'll learn:

- Immutable Types
- Basic LINQ concepts
- Advanced LINQ concepts (Yeild)
- Func Delegates
- Expression Bodied Members
- Extension methods and pipe-lining
- Thread Safe Collections
- Tuples

**Prerequisites:** [Download Here](#)

---

## **PreCompiler Session 02 - Tuesday 1:00**

### **CodeMash Capture the Flag Tournament**

*Mike Woolard*

There is more to being a hacker than hooded sweatshirts and pasty skin. Hackers pride themselves on being problem solvers, thinking logically through not only what a web applications should do, but everything possible for it to do. We have all sat through the boring Top10 lectures and read the secure coding standards. Now let's learn what we are protecting our self from. Capture the Flag (CTF) tournaments are a good way of honing your security skills, or discovering skills you never knew you had. The CodeMash CTF kicks off in mid-December and runs through the final day of CodeMash. This precompiler session will be ran in an open house format, to answer any questions you may have, and provide an area for you to sit down with others participating and network your skills to help solve the challenges. Attend for 4 minutes or 4 hours. Stop by before or after attending a different session, or during a break at a different class. We will give you the rundown and you can stay and play for as long as you like! More information can be found at <https://codemashctf.com/>

---

### **From zero to a production quality webservice with Go**

*Zev Goldstein*

Go is hands down the best general purpose language out there today, but nothing you've read or heard is as good at experiencing it for yourself. This hands-on training assumes absolutely no prior knowledge of Go. It will walk you through setting up your local development environment and then help you evolve your first Go program from "hello world" to "real world." Along the way you'll learn

the core features of the language (including its concurrency model), become familiar with key components of the standard library, and learn the idiomatic patterns of the language. Attendees will leave with the tools and know-how to write their next project in Go.

**Prerequisites:** [Download Here](#)

---

## **Build Your First Voice-Enabled Experience with Alexa**

*Jeff Blankenburg*

The Amazon Alexa team will be on-site at Codemash to help you bring your very first voice skill to life. We will show you how to think about voice design, and what a conversational user interface looks like.

You'll leave this pre-compiler with a working Alexa skill that is ready to submit for publication, but more importantly, the knowledge you need to build the next one. This session will be very hands-on and technical, but anyone should be able to participate.

Before you arrive, you will need to set up a couple of things:

1) An account at <http://developer.amazon.com> 2) An account at <http://aws.amazon.com> 3) A laptop with a browser.

**Prerequisites:** [Download Here](#)

---

## **Custom NGINX Module Development**

*Aaron Bedra*

As systems grow they often benefit from layers of processing capability that help keep the overall system latency under control. There is a staggering amount of work that can and should be done at the edge that is often done inside of applications. Things like blocking known bad actors, first level authentication/session lookups, data enrichment, throttling, preliminary user validation, and a host of other problems can be safely pushed to the edge and provide a very useful boost to overall system performance and stability.

Join Aaron as he demonstrates how to create custom NGINX modules to solve these issues. Aaron will begin by demonstrating the NGINX lua programming model and then dig into traditional module development in C. He will leave you with working custom module code that you can use and adapt, as well as useful skeletons, local test environment tricks, and reusable test harnesses that ensure you can get moving on your work quickly and efficiently post-compiler. Attendees will walk away with the following knowledge.

- How to build and deploy a custom NGINX module in Lua and C
- How to test custom NGINX modules
- A deep understanding of the NGINX request lifecycle
- How to create self contained local development environments
- How to tap into advanced NGINX functionality

**Prerequisites:** [Download Here](#)

---

## **Leading in a Complex World**

*Doc Norton, Diane Zajac*

As a leader, be it as a manager, team lead, or individual contributor, how do you know which leadership style is best? And how are decisions best made on the team?

When, if ever, is command and control appropriate or even a “best practice”? What about collaborative consensus building? Should a single expert make the call or should the team agree on everything?

The answer is probably somewhere in the middle.

In this half-day workshop, you’ll explore the complexity of work and examine how it maps to leadership and decision making. You’ll learn when command and control is actually best versus when collaboration is most effective. Finally, Doc and Diane will teach you tools to improve your collaboration efforts.

Return to the office a more aware leader, prepared to help your team have the highest impact through your effective leadership and decision making.

---

## **Introduction to Kotlin**

*Andrew Fitzgerald, John Keyes*

What is Kotlin? Kotlin is a statically typed programming language that is built with a focus on being safe, approachable, and pragmatic. It was started in 2010 by JetBrains, the company behind IntelliJ Idea and Android Studio, and has grown exponentially over the last few years. It’s recently gained first class support for Spring and Gradle, and is used in the products of big companies like Pinterest, Evernote, Atlassian, and Netflix. And, if you had any doubts of whether Kotlin was here to stay, Google announced Kotlin as an official language for Android at Google I/O 2017!

Join us for an in depth, interactive introduction to the language. We'll cover *Basic syntax* Nullability *Object-oriented programming with Kotlin* Functional programming with Kotlin *Java interoperability* Building DSLs

**Prerequisites:** [Download Here](#)

---

## **Where have all the Cukes gone?**

*Doug Morgan*

So you can sling some code and automate a web browser, and even work on a team that has a fair bit of UI automation around your "favorite" web application. No doubt you have found yourself spending too much time fixing an inconsistent suite of tests. You might even be asking yourself "are these tests even worth it?". (the answer is yes!)

Lets explore where things typically break down in the acceptance testing process. Starting with breaking user stories into acceptance criteria, and how the words the team chooses can affect the software that is built. Then taking a deep dive into best patterns and practices in automating the scenarios with cucumber and other ruby modules. Along the way answering questions such as how should I manage my test data? Should I ever delete tests? And, how do I keep my test pyramid from becoming a test cupcake?

Attendees will walk away from this session armed with information to help them write better acceptance criteria, and tips to keep their test suites clean and less brittle.

**Prerequisites:** [Download Here](#)

---

## Modern CSS

*Philip Zastrow, Catherine Meade*

### **Take your CSS skills to new heights with today's techniques.**

It's no joke that things in the web move fast. A few years ago, staying on the forefront of CSS was knowing the vendor prefixes for the latest CSS3 implementations and poking around at Sass or LESS. Now, in addition to the preprocessors, we have postprocessors, build systems, and CSS components—not to mention the latest and future CSS features! In this workshop, we'll help get you up to speed on these various approaches to writing CSS, their advantages, and how you can start using them. Whether you're a regular CSS practitioner ready to level up or a backend dev expanding your skillset, we'll cover all you need to organize, manage, and write modern CSS.

### **We'll cover many of the new tools, practices, and techniques prevalent in CSS today.**

- Naming conventions (BEM, SMACSS, etc.)
- CSS processors (Sass, PostCSS, Autoprefixer, etc.)
- Build tools
- Mixins and Functions and Plugins
- New CSS features and how to use them (plus previews of CSS4 features)
- Tips and tricks to common CSS problems

### **What to bring:**

- A practical knowledge of CSS—we'll be diving into some advanced areas in this workshop.
- A CodePen.io account (they're free!)—we'll be putting several concepts in practice, utilizing CodePen in some cases.
- Optionally, we have examples of more advanced tools—in which case, it would be helpful to have Node.JS installed beforehand.
- And a computer, of course!

**Prerequisites:** [Download Here](#)

---

## First Person View Drone Racing (Day 1, Part 2)

*David Resseguie, John Chapman*

Have you ever dreamed of flying? Have you seen videos of people piloting mini pod racers or watched drone racing on ESPN? Do you want in on the fun? Join us as we demystify these magical machines. In the first part of this session, we will learn all about the technology that makes quadcopters fly: flight controllers, motors, ESCs, receivers, transmitters, etc. Participants will then have the opportunity to build a micro sized racing drone and put their skills to the test in the ultimate first person view Codemash Drone Prix. Will YOU be crowned the ultimate drone racing champion of Codemash?

**Prerequisites:** [Download Here](#)

---

## A Whirlwind Tour of Recurrent Neural Networks

*Sarah Sexton, Jennifer Marsman*

Neural Networks. Deep Learning. TensorFlow. What are these buzzwords? What is the latest Artificial Intelligence craze? This advanced session contains cutting-edge information *not* easily found online, and does *not* require a PhD in Machine Learning to understand. Recurrent Neural Networks (RNNs) can be used to generate text that will look like its original training data. There are many [articles](#) out there that show hilarious end-results of such adventures, but start-from-scratch walkthroughs that show the raw code, like this one, are hard to come by. This presentation will demonstrate what you need to join in, grab your own data set, process it, train it, and sample it. Training the data on a CPU can take hours, but in this session you can learn how training on a GPU with hardware acceleration takes only seconds. Come away from this session with your own datasets to “randomly” generate new bodies of text!

Hands-On attendees will find and pull their own dataset, pre-process it, sample, and train it. If time permits, we will explore GPUs on Azure.

**Prerequisites:** [Download Here](#)

---

## PreCompiler Session 03 - Wednesday 8:00

### Webapp Pentesting for Developers and QA Persons

*Brian King*

Penetration testing is just QA testing, with a different focus. Where QA is about making sure your app actually does what it should do, reliably and smoothly, pentesting is about making sure your app doesn't also let people do other things - things they shouldn't be allowed to do, like steal your database - and that when it has to fail, it fails cleanly and safely. The tools are similar, and the objective is similar, but the mindset is a little different.

In this precompiler, you will spin up a VM you can keep, with some webapps you can test, and tools you can use to test them. You'll learn how to use these free tools to find some vulnerabilities, and - more importantly - understand why they happen. What you learn will make you a more versatile and confident developer or tester, and just might chop a couple of pages off your next pentest report.

**Prerequisites:** [Download Here](#)

---

### Functional Programming from None to Infinity

*Nathan Dotz*

In this workshop, we'll start by exploring two functional programming languages that these days are relegated largely to academic study, highlighting the influences they've had on modern functional programming paradigms and reinforcing these ideas by recreating the abstractions that set the groundwork for functional programming as we know it today. First, we'll cover ML, a language whose ideology remains largely unchanged for over 40 years and considered the common ancestor of modern functional languages like Haskell, OCaml and F#. Next, we'll cover Racket, a LISP derived from Scheme which shares similarities and probably inspiration with a number of currently popular languages, perhaps most obviously, Clojure and perhaps less obviously, Javascript. Lastly, we're back to where you work: I'll present exercises that help to reinforce the ideas learned from ML

and Racket to be solved either in the language of your choosing, or Javascript (the de facto language of examples and solutions du jour).

Join me for an enlightening foray into some languages largely forgotten or ignored by the industry that are certain to help you bring new ideas about functional programming to your daily practice, whatever it may be.

---

## **Building Serverless Applications in AWS Workshop**

*Jarred Olson, Christopher Judd*

Tired of trying to manage and maintain servers? Never have a large enough operations team? Don't have a budget for running lots of server? Don't want to pay for servers sitting idle? Afraid you might become so popular that you won't be able to scale fast enough? Don't worry, it is possible to alleviate these issues by moving to a serverless architecture that utilizes microservices hosted in the cloud. This type of architecture can support all different types of clients including web, mobile and IoT.

During this hands on workshop, you will build a serverless application utilizing AWS services such as Lambda, API Gateway, S3 and a datastore.

**Prerequisites:** [Download Here](#)

---

## **Build Your Own Arduino Workshop**

*Nathan Schlehlein*

Do you live with your head in a cloud? Do you fear small things? Is your mastery of virtualization and IAAS to the point where you haven't seen a real computer in months? Do you (most importantly) feel an unexplainable compulsion to open anti-static baggies handed to you by (almost) strangers and tinker with the contents therein?

If so, have we got a workshop, nay, a way of life for you!

In the Build Your Own Arduino Workshop, you and a few of your newest, closest, coolest friends will construct an Arduino-compatible microcontroller board out of a bunch of inexpensive parts.

While constructing the board, we will discuss and tinker with each component (clock source, power regulation, etc.) to see how they work. Then we'll boot 'em up and (provided the blue smoke hasn't escaped by this point) start writing some code and talking to some devices.

**Prerequisites:** [Download Here](#)

---

## **Infrastructure as Code: Automate all the Things!**

*Franklin D, Kimber Dowsett*

This hands-on workshop guides participants through the steps necessary to stand up a Linux environment in AWS using terraform. The often confusing and sometimes complicated steps necessary to build out hosts in AWS are reduced to repeatable recipes which are easily stored and in Github. Further provisioning steps will be accomplished using puppet.

Participants are expected to bring their own laptops & have AWS accounts created before the workshop begins.

**Prerequisites:** [Download Here](#)

---

## **Let's learn Android! (Kotlin Edition)**

*Michael Yotive*

Have you ever wanted to learn how to develop mobile applications for Android? Have you heard of Kotlin, the newest JVM language from JetBrains who Google has officially partnered with for Android development? In this workshop, we'll cover everything you need to get started on your journey to becoming an Android developer with Kotlin. Even if you're already an Android developer, this workshop will give you a great foundation in a hot new language! Bring your laptop with Android Studio installed and be ready to learn a lot!

**Prerequisites:** [Download Here](#)

---

## **Improv for Geeks and Those Who Want to Work with Insanely Great Ensembles**

*Russell McMahon*

This workshop will teach participants some of the basics of improvisation and ideas on how it can be used in at work. There are studies that suggest improv does help us all to become better team members, learners, innovators, and communicators. Companies are using the improv methods as a way of creating more innovative and collaborative teams, for brain storming, and creating a work environment that says "Yes" before "No". Improv training can help students become better learners and make learning more enjoyable. Come and learn about improv and why companies such as IDEO, Google, Marriott, and Twitter have embraced this to build a culture that promotes better communication, collaboration, and team building. This workshop is an interactive workshop. Please attend and have fun learning how to be more positive, vulnerable, attentive, and playful in your daily grind.

Upon completion of this course, attendees will be able to 1. Identify areas where improv can be used in their work. 2. Perform some basis improv components. 3. Use improv games as tool to innovate and ideate. 4. Understand how failure fits into life-long learning. 5. Exemplify the "Yes And" mantra in their work.

---

## **The Hidden Requirements: Exploring Emotions with Placebos**

*damian synadinos*

The way we feel is important!

All that we think, do, or say is influenced, to some degree, by emotions. Many successful businesses and people recognize the importance of emotional considerations.

The way we feel is important and should be considered!

All software is intended to help solve some problem, and both problems and solutions evoke emotions. Software requirements are simply wants or needs, which often stem from core emotions. Research shows that emotions can affect the acceptance or rejection of software.

The way we feel about software is important and should be considered!

A placebo is designed and used primarily to evoke emotions. Things like sugar pills, false elevator door close buttons, and fake office thermostats aim “to please”, rather than have any other physical effects. Placebo requirements focus on emotions. And so, considering software through the lens of a placebo can help emphasize emotional considerations, and provide a valuable perspective on bugs, ethical design, and much more.

In this workshop, I support the claims above, suggest some methods to elicit and test emotional requirements, and finally, use placebos as a lens to view software design and testing. Using presentations, demonstrations, and interactive discussions and exercises, we collaboratively explore why “The way we feel about software is important and should be considered!”

Workshop Takeaways: *Practical methods to get and measure emotional requirements* A different and deeper understanding of emotions and requirements \* An introduction to the fascinating and spooky world of placebos

---

## First Person View Drone Racing (Day 2, Part 1)

*John Chapman, David Resseguie*

Have you ever dreamed of flying? Have you seen videos of people piloting mini pod racers or watched drone racing on ESPN? Do you want in on the fun? Join us as we demystify these magical machines. In the first part of this session, we will learn all about the technology that makes quadcopters fly: flight controllers, motors, ESCs, receivers, transmitters, etc. Participants will then have the opportunity to build a micro sized racing drone and put their skills to the test in the ultimate first person view Codemash Drone Prix. Will YOU be crowned the ultimate drone racing champion of Codemash?

**Prerequisites:** [Download Here](#)

---

## Real-world React from the ground up

*Nathan Loding*

Learning React can get awfully complicated. Flux, Redux, server side rendering, React Native, higher order components, functions as child components, CSS in JS, and more. Let's simplify the equation: build a real world application using React and Redux, learning patterns and practices to make reusable, composable components as we go. Whether you are brand new to React or have some experience with it, you should walk away with a solid understanding of how to compose React applications.

**Prerequisites:** [Download Here](#)

---

# PreCompiler Session 04 - Wednesday 1:00

## Unleashing Your Creativity

*Christina Aldan*

For innovation to flourish, organizations must create an environment that fosters creativity; bringing together multi-talented groups of people who work in close collaboration together— exchanging knowledge, ideas and shaping the direction of the future. Creativity is a skill you can develop with

practice and a process you can manage. In this creativity workshop, you will:

- Understand the dynamics of creative work, the most common pitfalls, and how they affect teams
- Learn to identify and eliminate the things that stop creativity
- Discover specific, everyday practices that lead to long-term effectiveness
- Understand the pillars of effective collaboration and creative leadership
- Develop the mindset, skillsets, and toolsets of highly creative people in business, arts and science.
- Discover your natural capacity for creativity through a range of hands-on activities that will appeal to both sides of your brain
- Explore some of the thinking strategies of innovators in business, art and science, and how you can apply these strategies to your everyday work
- Experiment with ideation tools that supersede traditional brainstorming in a fun, inspiring action-filled environment
- Find the business case to bring creativity into your organization

---

## **Always Be Delivering**

*James Balmert*

What do Facebook, Google, Netflix, LinkedIn, WordPress and Etsy all have in common? They are all well known IT organizations that practice continuous delivery and deployment (CDD). This helps them reduce the risk associated with releasing code, improve system resiliency, and reduce overall software development costs. At any point and often multiple times per day, they deploy the latest changes to their production environment with full confidence because their builds have been tested automatically.

In this hands on session, you will learn how to continuously deliver with confidence. You will set up a development pipeline for a sample application. Step by step, you will assemble individual tasks and link them together to form a pipeline that takes a code change from source control all the way to a live production environment. But a pipeline is just the start. Along the way you will learn how you can use feature toggles, fast failures, small changes, and optimized tests to encourage a culture of CDD right now.

**Prerequisites:** [Download Here](#)

---

## **Machine Learning with R**

*Matthew Renze*

R is a very popular open-source programming language for machine learning. Its interactive programming environment and powerful data analysis capabilities make R an ideal tool for machine learning.

This workshop will provide an introduction to the R programming language using RStudio. In addition, we will demonstrate how we can use R to train a series of machine learning models. We'll cover classification, regression, clustering, anomaly detection. Finally, we'll learn how to deploy these models to production to make predictions given new data.

**Prerequisites:** [Download Here](#)

---

## **Demystifying ethereum to build your own decentralized (D-app) app using blockchain**

*Abhiram Ravikumar*

### **Context**

**Bitcoin** is the new buzz around town and blockchain is the technology behind this hugely popular crypto currency form. This hands-on workshop helps participants understand the mechanics of blockchain and help them build their first decentralized application using ethereum, an open source platform that leverages the power of smart contracts to build scalable blockchain-enabled applications.

The session will start with an overview of **blockchain** technology, move on to what a distributed ledger is, then introduce a real life example of bitcoin - its evolution, application and future.

Once participants realize the extent and use of blockchains, the concept of decentralized apps would be introduced along with the various trust models that make it work. After the context is set, the facilitator would introduce **ethereum** and dive into the hands-on [workshop](#).

## Workshop flow

- Setup development environment - Install testrpc, web3js and start test blockchain
- Smart contracts: Use solidity (an objected oriented programming language) to write out a simple contract that increments votes and returns the count of votes for each band
- Setup a simple website to invoke the contract methods
- Play with the application and see blockchain in live

The facilitator will consciously not be using any D-app frameworks so that participants appreciate the level of heavy lifting it does for them later on when they use the framework. I will start off with a few questions on the famous aspects of bitcoin like its current value, how many people use it, etc.

## Who is the talk for?

The talk has both a technology aspect and a socio-economic aspect to it and the facilitator folk from both these realms. *Full stack web developers (amateurs + professionals)* Bitcoin enthusiasts and investors \* The curious folk who want to know what all the hype about "bitcoin" is..

## Key takeaways

At the end of this workshop participants will, *Have a deeper understanding of blockchain technology and its encryption techniques and workflow* Have their own blockchain-enabled application *Learn to create simple contracts using solidity* See and understand the fully functional demo of [Korkscrew](#) \* Be able to educate others about blockchain and its use-cases

The workshop is an ignitor for participants to explore this field, the expectation is that they use the learning to build their own fully functional [decentralized applications](#) in the future.

**Prerequisites:** [Download Here](#)

---

## 3D Modeling for Makers and Game Developers!

*Robert Palmer*

Just bought a 3D printer and want to design custom parts for your robot army? Are you a developer looking to expand from curly braces and semicolons to Icospheres and UV Maps? Are you a maker ready to move up from browsing Thingiverse and on to creating your own content? This hands on session will provide you with the tools and techniques to make this happen using Blender, a free, open source software package for 3D art and animation. As this is an intro session, the hands on

examples will focus on basic 3d modeling techniques, with demos on texture mapping and simple keyframe animation. Attendees will leave with the knowledge needed to bring their creations to life!

**Prerequisites:** [Download Here](#)

---

## **Practical Functional Java**

*Jeff Butler*

You've read all about Elixir, Haskell, Clojure, etc. and are really excited by functional programming. But in your day job, you are a Java programmer. Can functional concepts be applied in a language like Java that is not a purely functional language? The answer is a resounding "yes"! If you will embrace functional concepts in your every day work, then your code will be cleaner, smaller, and easier to maintain.

In this pre-compiler, we will embark on a successive refinement exercise to refactor a horrible deeply nested looping monstrosity into something more elegant and functional. Along the way, we'll learn about the features in Java that are moving Java into the modern functional world including:

- Streams
- Map, Filter, Reduce
- Optionals

The code exercises for this session will be in Java, but the concepts are equally applicable for JavaScript and C#.

**Prerequisites:** [Download Here](#)

---

## **Workshop on Autonomous Vehicle Embedded System Development, Integration and Validation Techniques**

*DJ Daugherty*

Ever see a cobweb and wonder how the spider can create something so complex? Similar to that cobweb, embedded systems and the associated software are growing in complexity. Advancements in hardware are pushing the boundaries of software development techniques.

The development of autonomous vehicles is challenged with system integration and validation techniques. Traditional product development starts from a physical hardware point of view and then adds software. As hardware complexity increases, companies must drive product development with software. Companies with software at their core are able to accomplish speed to market faster than the spider can spin its web. This is done by driving development with modern software practices such as test driven development and continuous integration.

Experience an interactive workshop on how to develop an embedded system while using modern software practices.

Learn how to: - Test drive an embedded component. - Mock an unavailable component. - Apply Test Driven Development (TDD), Continuous Integration (CI) and Mocking to achieve a scalable software process on an embedded project.

**Prerequisites:** [Download Here](#)

---

## First Person View Drone Racing (Day 2, Part 2)

*David Resseguie, John Chapman*

Have you ever dreamed of flying? Have you seen videos of people piloting mini pod racers or watched drone racing on ESPN? Do you want in on the fun? Join us as we demystify these magical machines. In the first part of this session, we will learn all about the technology that makes quadcopters fly: flight controllers, motors, ESCs, receivers, transmitters, etc. Participants will then have the opportunity to build a micro sized racing drone and put their skills to the test in the ultimate first person view Codemash Drone Prix. Will YOU be crowned the ultimate drone racing champion of Codemash?

**Prerequisites:** [Download Here](#)

---

## See Your Data on the Network with Wireshark

*Wayne Goode*

Does your program send data over the network? When debugging, it is useful to know what data your program is sending and receiving. This presentation will explain basic networking, how to use Wireshark to see the data and how to use Wireshark Generic Descriptor to define custom protocols.

If your program sends data over the network, then you may someday need to know what it looks like on the network. When debugging, it is often useful to know exactly what data your program is sending and receiving. This presentation will give you a quick refresher (or fast introduction) to basic networking protocols, explain how to use Wireshark to see your data and how to use Wireshark Generic Descriptor to define you custom protocols.

**Networking Refresher** A review of basic networking, standard protocols (IP, TCP, UDP, etc.) and how the networking model uses this layers to build messages.

**Wireshark Overview and Demo** After an overview of how to use Wireshark, we will look at some of example data captures to practice using Wireshark, as a group and also in individual excercises. We will also have examples and excercises to illustrate the lessons learned in the Networking Refresher.

**Wireshark Generic Descriptor (WSGD)** If you have a custom protocol, you will need to define it so that Wireshark can display the data. We will look at the three options for creating dissectors. We will look at how to define a protocol using WSGD and how to configure Wireshark to use it. We will look at examples of Wireshark displaying data in custom protocols and have excerses in creating dissectors to display packets from an example custom protocol.

**Prerequisites:** [Download Here](#)

---

## General Session 01 - Thursday 8:00

### AWS Security Essentials

*Aaron Bedra*

Are you using or moving to AWS? Have you considered how you organize and secure your AWS environments? The growing push to cloud providers has allowed us to move faster and tackle problems more efficiently. The same freedoms that have allowed us to move faster have also created scenarios where security issues are exposed by accident and/or without proper

management and review. As companies move toward more and more cloud usage, teams are pushed harder to ensure the same compliance and security requirements that exist in slower moving private environments. This has the potential to put us right back where we came from.

Join Aaron as he talks through the most critical security decisions you can make for you AWS environments. He will identify issues and solutions in an automation friendly fashion that aim to fit seamlessly into the development and deployment lifecycle. This session will cover the following topics:

Account provisioning and IAM Credential management VPC setup and network design AWS services that boost your security posture Auditing AWS configurations to find security holes Creating a robust CI pipeline that ensures no obvious security holes are present within your environments In addition to these topics a heavy emphasis on both platform and server automation will be included. Please note that this session is heavily tuned to people using Amazon Web Services. If you are using another Cloud provider the ideas will still be relevant, but not all solutions will be available for your provider.

---

## **Take the Toxic Out of Your Team and Up the Efficiency With the Power of Google**

*Jason Blackhurst*

We all want our teams to be ultra efficient, but even a touch of the cowboy coder, the hero programmer, or the freeloader can be toxic to your team. Google's Project Aristotle (<https://rework.withgoogle.com/guides/understanding-team-effectiveness>) found that the efficiency of a team is driven by 5 factors: *Psychological Safety Dependability Structure and Clarity Meaning of work* \* Impact of work

Come and learn how to use these factors to identify and eliminate the toxicity killing the effectiveness of your team. Using stereotypically terrible teams, having members like the ones mentioned above, you'll learn why it's so important to provide transparency, set goals, and have the necessary conversations you've been putting off.

---

## **Accessibilty in Android**

*David Truxall*

Accessibility is often an afterthought when developing apps of any kind, yet if your app is not embracing accessibility, you are losing a large percentage of your potential market. In addition to lost market share, there is a legal aspect of accessibility, especially if you build applications for government organizations. Section 508 of the Rehabilitation Act mandates specific types of accessibility features in your app, and the rules for apps are different than the rules for web sites, so we'll learn what rules apply to mobile apps. Every app can benefit from applying these rules even if the government is not involved. This session demonstrates hardware and software tools and coding techniques which enable inclusivity in your Android app and embrace disabled users.

---

## **SQL Server on Linux? Wait ... What?**

*Don Schenck*

SQL Server on Linux.

Just let that sink in. What universe are we living in? What is SQL Server on Linux? Is it the same? How does it differ? What is the learning curve? Is this a sign of the Apocalypse?

Well, this session has the answers. Follow along as we explore our favorite RDMBS running on Linux, including a demonstration of SQL Server running in a Linux container. You'll see a website running in a container that uses SQL Server running on Windows, and then -- using an Availability Group -- we'll switch over to SQL Server running on Linux without missing a beat.

---

## **Fast Neural Networks... a no brainer!**

*Riccardo Terrell*

Artificial Neural Networks are a fascinating computational approach modeled to react similar to a biological brain to solve problems. Neural networks are a very powerful tool that enable machines to teach and evolve themselves. You can utilize this power in almost any application. Don't be intimidated by the equations and Greek symbols of this cutting-edge technology, I will guide you on how to slay this technology and make it bend to your will. In this session, we are going to demystify the apparent Neural Networks complexity without the need of mathematical background providing an approachable overview. There will be very little math and lots of coding. The goal is for everyone to become acquainted with Neural Network from a pragmatic standpoint. Just when you think it couldn't get any better, I will show you how to employ the functional paradigm to leverage multicore machines and GPUs to make your neural networks predictions infinitely faster through parallelism. By the end of this talk, you will learn the basic concepts of Neural Network and how to apply functional concurrency to estimate future stock prices at blazing fast speeds...and perhaps get rich while practicing.

---

## **Jewelbots: How to Get More Girls Coding!**

*Jennifer Wadella*

Girls just wanna have fun ... coding their own programmable jewelry! Jewelbots are the latest in wearable tech with a great feature - they allow you to write your own code. This live demo of programming with Jewelbots will introduce you to the Jewelbots API, highlight fun projects, and talk through tips on teaching and encouraging young women in tech. Loops are easy to understand when accompanied by color coordinated led lights!

---

## **Real time traffic visualization in a microservices world**

*Roberto Perez Alcolea*

It's been a few years since people started to move into microservices architectures. By now, we've learned that we require several things in order to be production-ready for them.

You probably already have a lot of tooling such as externalized configuration, service discovery, distributed tracing, logging and others but... How do you know that everything is fine after you roll out a new deployment? Or... How do you start diagnosing an incident in production in the middle of the night?

Wouldn't be nice to have a tool that allows you to see how real traffic flows within your microservices and external dependencies?

This talk will have a live demo that goes through all the steps to use Hystrix in a JVM-based app and expose those metrics via Turbine for Vizceral.

Vizceral is a tool from Netflix to visualize traffic between components. This data needs to be fed from somewhere however. Hystrix provides fault tolerance, and a nice stream of all metrics. Turbine aggregates the hystrix streams per cluster.

---

## **Next Evolution of Unit Testing: JUnit 5**

*John Pendexter*

With recent updates to the Java language specification (most notably lambdas) the JUnit framework has been updated to take advantage of these changes. Have you ever wished you could better partition your tests? Or add metadata to your tests beyond categories, test names, or comments? This talk will explore JUnit 5 with a focus on making unit testing easier, more efficient, and more structured, using test factories, dynamic tests, default interface methods, custom annotations, and more. Finally, we will look at migrating from JUnit 4 and ways that JUnit 5 features can make taming unruly legacy test suites easier.

---

## **Exploring a NoSQL Option for Storing Data in your Mobile Apps**

*Priya Rajagopal*

If your mobile app stores data locally, you've probably at least looked at SQLite or Core Data . These come with their share of pain points. Database schema changes are inevitable with changing requirements and require tedious data migration. Working with SQLite often requires a clumsy mix of SQL with Swift/ObjC/Java/Kotlin and it's very easy to go wrong. And then, there is the potential impedance mismatch issue to consider while trying to model data as relational tables.

This talk will examine NoSQL as an alternative to locally persisting data on mobile platforms with Couchbase Mobile as a concrete example. We will provide an overview of the various flavors of NoSQL, focusing on the Document style. While NoSQL database technologies is gaining traction in large scale distributed applications, we will discuss it's benefits as an embedded database on mobile platforms for enabling both offline-first and offline-only use cases.

I will introduce Couchbase Lite, an open source, cross-platform NoSQL embedded data store for your mobile apps. We will discuss how you can integrate this in your mobile apps and walk through examples describing the CRUD and Query interface.

At the end of the talk, you will walk away with a good idea of the benefits of NoSQL as a embedded store. Anyone building a mobile app with local storage needs will benefit from this talk.

---

## **Restful Services with the Play Framework, MySQL, and a Security Level with JWT**

*Mercedes Wyss*

The Play Framework, based on a lightweight, stateless, web-friendly architecture, is a great framework for easier, faster, complex web development, including RESTful services, and is reactive. Based on Scala, you can choose between using Java or Scala for development. This session explores Play framework characteristics, how to create a RESTful service, connecting Play to a MySQL database, and adding a security level with JWT (JSON Web Token).

---

## **Crimson and Clover, Over and Over (and Over). Applying a Framework to Continuous Security**

*Josh Wallace*

Tommy James wrote “Crimson and Clover” while attempting to change the band’s sound and shift their focus from releasing singles to releasing entire albums. Continuous Security (or SecDevOps, AppSecDevOps, DevSecOps, Rugged Devops or whatever else you want to call it) is in need of a “Crimson and Clover.”

Most organizations are still overwhelmingly focused on the security of single applications. Those moving to continuous delivery models either ignore security completely or discover that security becomes a bottleneck in their quest to deliver an album...I mean code...rapidly.

In this session, we will share how you can create our own “Crimson and Clover” in the form of a Security in DevOps Framework. You can use a framework to allow organizations to establish a baseline of their environment and use it to make informed decisions when trying to fill the security gaps in their continuous delivery pipeline. The a security in DevOps framework offers a benchmark for continuous security improvements while also demonstrating not only that security is important, but that it can be achieved while hitting your DevOps targets. That’s a win-win for everyone involved. In this talk, we’ll discuss:

- A structured approach to solving the complicated problem of integrating continuous delivery and application security.
- How you can apply a framework to your continuous security model and inform your strategic decisions.
- How to collect evidence and metrics to provide application security guidance at speed and scale.

---

## **Unbreakable. Perfect: The Craft of Code**

*Joe Morgan*

The best welders create seams so perfect that flaws cannot be detected even by an X-ray. Most welders will never get there. Code is no different. Do you write code that ‘works’ or code that is crafted so perfectly it will never show a flaw? We will explore timeless principles that make great code: recognizing code smells, writing for testability, keeping things simple and clear. Many of these ideas were written down decades ago, but are still ignored. The most important first step is viewing code as a craft, something that is done for its own sake rather than something hacked out to solve a problem. In this session, you’ll learn how to create a plan for steady improvements. Syntax, languages, frameworks all fade away. A mindset that is focused on craft will never fail.

---

## **General Session 02 - Thursday 9:15**

### **Crypto for Developers: Making Sane Choices**

*Adam Caudill*

How do you secure the data in your database? How do you store passwords? From AES to Salsa20, RSA to Curve25519 - this session provides an overview of what you need to know to secure your data. Focusing on practical uses cases over math and complexity, this will help you make the right choices and implement a system that is truly secure. Data security has never been more important, and for many cryptography is a black magic, a world of complex math and obscure terminology; this session aims to demystify that world and bring cryptography into the light. You’ll be able to understand what you need to do, to keep the hackers out.

---

## **Building Holographic & VR Experiences with the Mixed Reality Toolkit for Unity**

*Nick Landry*

The Mixed Reality Toolkit is an open source project led by Microsoft and the HoloLens community aimed at simplifying the development of Mixed Reality experiences. Mixed reality blends 3D holographic content into your physical world, giving your holograms real-world context and scale, allowing you to interact with both digital content and the world around you. The Mixed Reality Toolkit consists of a collection of scripts and components intended to accelerate the development of holographic applications targeting Windows Mixed Reality. This session is a technical dive into the Mixed Reality Toolkit for Unity where we'll explore advanced holographic features such as gaze & gesture input, voice commands, spatial mapping, and more. All demos will be shown on the Microsoft HoloLens and the new immersive Mixed Reality headsets. Come learn how this open source toolkit – combined with Unity – can get you started quickly on your development journey into a world of holograms.

---

## **Most Useful Design Patterns**

*Steve Smith*

Design patterns provide common templates for solving similar problems. They also provide a higher-level language for software developers to use to describe approaches they might choose when designing part of an application. This session introduces and applies several patterns useful to web application developers.

Examples will primarily use C#/NET.

---

## **Maturing the Enterprise Quality Practice**

*Sarala Pandey*

The two primary contributors to poor quality in an organization are lack of involvement by management and lack of knowledge about quality. Without the right processes and people, quality will be either a cost center or forgotten component by development. To achieve organizational success, enterprise quality must take action to build quality from the top down. Management must accept responsibility for the quality practice within the organization and promote it across the organization. Everyone is responsible for quality, not just QA. The journey is fraught with obstacles - maturing the quality practice of an organization builds long-term success with robust processes and well-trained employees.

---

## **Getting Started with Deep Learning**

*Seth Juarez*

Machine learning is fast becoming the software spice of choice to spruce up your projects. Deep Learning is in the hot sauce category. In this session attendees will learn the basics of deep learning in an approachable and friendly way. The session will start from problem inception to actually training and improving a deep learning model. Attendees should come away with a game plan on how to incorporate these techniques in their existing software projects. I make jokes too.

---

## **Build a Delightful API with Serverless Technology**

*Rob Allen*

What makes an API or microservice delightful? An intuitive interface, consistent behaviour, and a reliable, scalable platform. This talk will cover how serverless technologies can help you to deliver all of these things for your API, ensuring that developers will want to work with your API over the competition. I'll cover how to design, build and deploy for serverless platforms, and how to use an API gateway to turn incoming HTTP requests into events that trigger those serverless functions. I'll share some top tips for solid API design - both from a technical HTTP standpoint and with developer experience firmly in mind. After this session, you'll be building and iterating on first-class APIs and microservices, and developers will be clamouring to use them.

---

## **EventBus and MVP - The Chocolate and Peanut Butter of Decoupling Android**

*Jim Kirkbride*

Render the view, execute the business logic, manage the network, handle errors. These are all separate concerns which should be handled separately. Unfortunately Android makes it far too easy to make a big, tightly-coupled mud pie out of these activities (pun intended). This makes implementing new features, testing, and maintaining existing code a nightmare.

So how do you build a complex Android app without coupling everything together?

Enter EventBus and MVP. In this session we'll describe how to build and test a flexible, maintainable app using the Publisher/Subscriber and Model-View-Presenter patterns to cleanly separate our concerns. Developers who attend this session will leave with new tools and techniques to separate the view from the business logic, and to cleanly implement network calls and error handling.

---

## **How to get started with robotics and IoT at home**

*Carla Siler-Maddalena*

Have you ever wished that there was an automated way to alert you that you left the garage door open? Or create a cool digital badge to wear at Codemash? Or create a small robot with your kids? Anyone can create projects like these but getting started can sometimes seem overwhelming if you have no experience with hardware/sensor integration.

In this session, I will introduce a variety of low cost, introductory projects and platforms that you can do at home that cross over between robotics and IoT. Many of which are child friendly so you can do them with your family. Then we will deep dive into two projects I have created; a sensor to alert me to open doors and a digital badge. We will go over the platforms, tools and techniques to create each of these at home. By the end of the session you should feel that you have the resources and knowledge to begin to create some of these project at home.

---

## **Devops Zen: Injecting Automated Tests into Infrastructure**

*Stephen Shary*

Devops zen is to make infrastructure predicable. At Kroger, we use Nginx as a reverse proxy to route traffic with a configuration that is thousands of lines long with pull requests from dozens of teams. Even with this scale, we are still able to complete on-demand deployments to multiple environments that serve dozens of domains and route to over 50 applications clusters with zero

downtime. We show an open source framework, SnowGlobe, which allows us to simulate and test every possible traffic routing situation before we deploy. We use a full CI/CD pipeline with 7,000+ tests run for every commit. If boring is zen, then our meditation is automated infrastructure testing.

---

## **Dev Thumbs, QA Thumbs and the End of "Ready for Testing"**

*Clint Hoagland*

Handoffs create waste; heavy handoffs create heavy waste. The way developers and testers collaborate can make the difference between a team that works as a cohesive unit, and a team that works against itself. As a developer, do you find yourself trying to understand bug reports about code you committed weeks ago? As a tester, do you worry that submitting a bug creates unplanned work, threatening the project's timeline? Over time, heavy handoffs can create a gradual accumulation of project risk, an ever-growing backlog of bugs, and, in the worst cases, an adversarial relationship between developers and testers.

In this experience report, software tester Clint Hoagland will discuss how TechSmith's Camtasia team moved off a heavy process where dev tasks languished in a "Ready for Testing" column for days or even weeks before being tested, and moved instead to a process using exploratory testing artifacts inside GitHub Pull Requests. This process revitalization enabled developers and testers to work comfortably together: reducing risk, minimizing rework, and keeping quality high as a sustainable practice. Clint will discuss where the team started, how it's working now, as well as specific techniques and recommendations the team found along the way.

---

## **Avoiding the Lock-In Fallacy**

*Casey West*

Lock-in is a misunderstood concept. Most of the time it's associated willy-nilly with vendors. Open Source is considered a panacea resolving the lock-in issue once and for all. In extreme cases some believe if they write everything from scratch the problem never surfaces. Unfortunately the opposite is true.

Lock-in is when you choose a bespoke solution, including one you create, and the cost of change exceeds your budget. It's not about vendors. It's not about Open Source. It's about making careful decisions about the cost of change.

Software is best designed by making explicit decisions when options are presented. The CAP theorem is a great example: choose consistency of data or availability of data in the face of a network partition. If you don't make an explicit choice when components can't speak to each other a choice will be made for you.

Likewise, when selecting components for your architecture the possibility of lock-in always there. Sometimes the organization gains more from making future change more costly by accepting lock-in. In that way lock-in may not be so bad! Often, though, lock-in represents too great a risk. It's important to evaluate these tradeoffs carefully.

In this talk I'll describe the thought exercises I've used in making architectural decisions around lock-in. You'll learn how to think strategically about the implications of building, buying, or using a component in your architecture. These techniques make software design and planning run more smoothly and help you eliminate biases of tribalism and fashion from the process.

---

## 10 Insane Tips You Can Use Today to JavaScript Effectively; #8 Will Shock You!

*Brian Genisio*

When the specification formerly known as ES6 (now ES2015) was ratified, we were all extremely excited. It had been 15 years since the last significant update to JavaScript, and the front-end world (as we knew it) changed. It has been over a year now and the excitement has ebbed, but our productivity is in flow. We've gotten past the novelty of the new features and learned how to use them effectively in everyday code.

These 10 insane tips will help you understand when and where to use these new features (including ES2016 and ES2017 features) of JavaScript. We'll touch on effective use cases of modules, destructuring, template strings, arrow functions, generators, and much more. Using real-world examples, you'll see how to leverage the new language capabilities for maximum readability and expression. If you haven't jumped into ES201x, we'll discuss how to embrace the future while preserving browser compatibility. If you have used it, see how others are using it. We'll also discuss some future features you can use today, if you feel like living dangerously!

---

## General Session 03 - Thursday 10:30

### Automating Security Testing with the OWTF

*Jerod Brennen*

When it comes to app security, scanning is good, but pen testing is better. That said, we're lucky if we can schedule (and budget for) a web app pen test once a year. Wouldn't it be swell if we could automate the security testing process so it turned up the same weaknesses in QA an attacker would likely try to exploit in Prod?

Well, then. You're in luck.

OWASP's Offensive Web Testing Framework (OWTF) was designed to help automate the web app pen testing process. By baking the OWTF into your own QA processes, you can benefit from the same knowledge and tools that the bad guys use to attack web apps. Better yet, you can run these tests as frequently as you like for FREE.

This presentation will show you how to use the OWTF, helping you improve both the efficiency and effectiveness of your app security testing process.

---

### Super Mario UX Designer

*Brad Colbow*

You jump into that next level! You presented with a game mechanic you have never seen before. And without a big popup window or a bunch of arrows pointing at the screen you are taught how to use that mechanic within the flow of the game. All while cosplaying as a racoon. In the end you feel rewarded and ready for the next challenge. It's like every level is a little onboarding experience. In this talk Brad breaks down some of his favorite levels in Nintendo's Mario games, talks about what makes them work and how we can apply these lessons to our designs in the real world.

---

### Decoupling the Frontend through Modular CSS

*Julie Cameron*

CSS is hard. It's a simple language, but getting it right and avoiding specificity hell can be a challenge if you don't have the right framework to back you up. Especially in large scaling projects, you might start adding ID selectors here and !important properties there and the next thing you know you've backed yourself into a corner where even the smallest of UI changes will take hours to work out. Ew.

Or how about this? Ever jump into a project and find that even the slightest markup change results in broken JavaScript AND sometimes even broken backend feature tests?! WTF. Ew.

This talk will look at how taking a modular, object-oriented approach to CSS can turn frontend woes into frontend wins. We'll examine modern CSS approaches like OOCSS, SMACSS, and BEM and demonstrate how they will help to not only decouple your CSS styles and reduce specificity conflicts, but how they will also help to decouple your CSS and HTML from your JavaScript and feature specs.

---

## **Build your own Voice Assistant**

*Jarred Olson*

Speaking is such a natural interface that we use it to interact with people all the time without even thinking about it. Why can't we do that with the systems we build every day? Maybe it is because historically you needed a PhD to understand natural linguistics. But now a days Amazon, Azure, Google and IBM are making their voice recognition capabilities available as a service. Come to this session to learn how you can use these services to build your own voice assistant.

---

## **Intro to Blockchain: What Is This For and Why It Matters**

*Shannon Wells*

In the last few years, cryptocurrency and blockchain have attracted the attention - and investment - of major financial institutions as well as governments. Many who have studied blockchain consider it a "foundational technology," meaning that which enables progress and applications in a variety of problem domains. There are uses for blockchain in particular that could change the way some things are done for the better - the way we purchase things, create contracts, get paid, prove copyright, etc. The history of blockchain, the basic mathematical principles underlying it, and some code examples will be presented. Last, the way blockchain is being used now, and the way it could be used in the future will be discussed.

---

## **Clean Architecture: Patterns, Practices, and Principles**

*Matthew Renze*

As software grows more complex, we need to manage this complexity by using various architectural patterns, practices, and principles. In this session, we will learn how software experts keep their architecture clean using a new approach to software architecture. We'll learn about domain-centric architectures, application layers, CQRS (Command-Query Responsibility Separation), event sourcing, microservices, and more. You can expect to hear practical advice and see real-world examples from over 17 years of architectural experience.

---

## **Java EE 8! Now what?**

*Ivar Grimstad*

Java EE 8 has been finalized a few months ago. In addition, Oracle has announced last summer its intention to open Java EE by moving the complete platform to an open source foundation. A bold move that is welcomed by the community and supported by companies such as Red Hat, IBM and Tomitribe. So it's clearly an exciting time for Java Enterprise Development!

This session will first cover in details the new enhancements of Java EE 8 like the new reactive JAX-RS Client API, Servlet 4's HTTP/2 support, Bean Validation 2, CDI 2 Asynchronous capabilities, new APIs such as the Security API and the JSON-B API, etc.

We will then conclude the session by looking at how the platform will move forward in the context of an open source foundation.

---

## **How Pro Wrestling Helped Make Me a World Champion Tester**

*Jenna Charlton*

Have you ever considered that software testing is like a pro wrestling match? Through my pro wrestling fandom, I've changed the way I think about delivering high quality software. Wrestlers look to find the perceived vulnerability in their opponent, and exploit it just like a tester would! During this session, you will learn about applying similar techniques to your test strategies, and how to apply them within your team. And just like wrestling there are multiple match types to consider, ways to take down your opponent, and a variety of skills to leverage to your advantage. We'll discuss...

- Don't hurt your opponent- viewing the developer as a partner in executing quality
- The lock up- sizing up the task
- The chair shot- picking the right tool for the job
- The reversal- How to pivot when testing turns you on your head
- The tag match- partnering with automation
- And many more concepts to help make you a 5 tester!

---

## **Imposter Syndrome: Overcoming Self-Doubt in Success**

*Heather Downing*

High achievers can run headlong into self-doubt, that can cripple forward progress. This talk approaches the issue scientifically, and presents ways to deal with this phenomenon, eventually moving beyond it. Geared for developers specifically, but most can identify with a crisis of confidence. In this session we will bring this sensitive subject to light by exploring:

- Why having this mindset is detrimental to your career
- Identifying the differences between healthy concern and illogical doubt
- What it actually means to 'fake it till you make it'
- How believing something negative about yourself can make it really happen
- Ways to cope once you experience a sincere failure
- Learning to graciously express confidence

---

## **Caring for New Users: Adoption, Onboarding, Permissions, Empty States, Feedback Herding, Friction, Hooks and More**

*Scott Showalter*

What does it take to craft a great experience for new users of our app? Once we're ready to move past MVP-stage, having learned what we needed to learn, there's still more to think about than merely what capability to give our earliest adopters and how that translates into functionality for the team to build and scale.

Enter the concept of "New UX". This session explores the idea of transforming new users into power users. It'd be nice if we could all hone our intuition skills and create absolutely 100% intuitive apps 100% of the time, but that's more fantasy than reality. So we must leverage techniques that help us make our new users feel awesome!

You'll learn: *How to overcome the struggle to drive adoption* Considerations for short attention spans, limited scratch memory or temporary disabilities, and avoiding avoiding committing funnel homicide from cognitive burnout. *The many types of onboarding, and which method is optimal for our users' in learning how to use our app.* Permission priming, permission pouncing and other concerns for user privacy when our apps need access beyond various device limitations. *What empty states are, and how you need to think about them in the context of your product and your users' goals in order to make them useful as well as delightful.* Sources of friction in the experience and in growing our user base. \* How we can keep new users coming back, through consideration of habit formation tactics

---

## **Writing Code Like a Poet**

*Brandy Foster*

Writing quality poetry means crafting words that evoke emotions in your readers. It's a way of telling a story that creates a bond. Writing quality code means crafting logic that evokes understanding in other developers. It's a way of creating a story of how your app should function. Both serve as a creative and therapeutic outlet. In this session you will learn how to utilize steps for writing poetry to improve your code quality, including the value of brainstorming, how to write tests that edit your code for you, and how to name your methods and variables like Amir Baraka.

---

## **Automation Trepidation: Does DevOps freak you out?**

*Ryan Foote*

It's not as scary as it sounds. Let's start with simple automation. By automating your build and release pipeline, you can begin to implement a DevOps culture in your organization. We'll explore the tools that will enable you to achieve this both quickly and easily.

In this session, we will start from scratch by building a brand new application and finish with a fully functional Continuous Integration / Continuous Delivery pipeline utilizing Visual Studio, Visual Studio Team Services and Azure App Services. We will explore the DevOps tooling provided by Visual Studio Team Services and developer workflow using branch policies with pull requests and gated check-ins.

---

## **Designers and Developers Working Together - Mass Hysteria! (And Accessible Websites!)**

*Caitlin Geier*

It's hard to get accessibility right when only one person is working on it. It takes a combination of thoughtful design and good, robust code to make any website or app truly accessible. Leaving accessibility to the designer doesn't guarantee good code, and leaving it to the developer often

means working harder to fix inaccessible designs on the fly. But working together, even when it's towards a common goal, is often hard to manage. This talk will go over tips and tricks to help designers and developers break down responsibilities, communicate decisions, and make good, accessible products.

---

## **Avoiding Microservice Megadisasters**

*Jimmy Bogard*

You've spent months re-architecting your monolith into the new microservices vision. Everyone gathers around to flip the switch. You navigate to the first page...and nothing happens. Refresh... still nothing. The site is so slow, it won't respond for minutes. What happened?

In this session, I'll walk through a post-mortem of a real-life microservice disaster. I'll show the modeling, development and production problems we found, and how we slowly morphed the new distributed monolith into our final picture of sanity.

While we can't prevent project failures, we can at least identify problems early on in our design so that our final product results in a clean, robust distributed system.

---

## **General Session 04 - Thursday 11:45**

### **Securing Your API Endpoints**

*Seth Petry-Johnson*

It's never been easier to expose services over HTTP. It's also never been easier to inadvertently expose security holes via those same services. This session is designed for the average developer/architect that wants a brief overview of API security without deep dives into cryptography or complex authentication frameworks. You'll learn about OAuth, API Keys, HMAC authentication and more. Don't worry if those things sound foreign; they'll be explained in a clear, practical way.

---

### **Refactoring a Legacy Application with ReactJS**

*Ali Orlando*

Testability, UI consistency, code reuse, maintainability. These are the properties of a healthy codebase... and what most legacy applications are lacking. If you're struggling with an application on life support, ReactJS can breathe new life into your front-end. Attendees will learn a proven approach for implementing ReactJS in a jQuery application and the problems you are likely to face along the way. This session covers why you would consider a refactor, setting goals for your desired outcome, an approach to adding ReactJS incrementally, and managing application state.

---

### **Getting OAuth Into Your Dev Toolbox**

*Brett Berliner*

It seems like there are two types of developers - those who have encountered a hacked together custom OAuth implementation, and those who have never used OAuth. The simplicity of the protocol encourages redevelopment where none is needed, which creates avoidable design and security headaches. This session is designed around providing a strong base in building an OAuth application, exploring what questions to ask and which strategies to use. Attendees will also see a

few real world examples, both good and bad, of implementing OAuth with both custom and out of the box solutions.

---

## **One Million Little Gophers: Building a Concurrent Web Service Using Go**

*Josiah Sleppy*

Concurrent programming is in! Delegating work to many running processes at once can increase most programs' performance and scalability. Many languages have libraries or frameworks that provide tools for writing concurrent code, but far fewer bake concurrency right into the language. The Go programming language not only does, but also provides synchronization across processes when needed through messaging. Go's channels and goroutines provide powerful, scalable, and convenient concurrency without having to download countless third-party packages. These features of the language make concurrent Go programs enjoyable to write and doubly efficient--both in developer time and actual program execution. In this session, attendees will discover the advantages of using Go through the creation of a simple, concurrent web service, and will walk away ready to harness the power of goroutines for fun, productive coding and performant programs.

---

## **The Science of Great UI, part 1 (Efficiency in Thought & Motion)**

*Mark Miller*

Explore the how and why of great UI. If you believe you're not an artist, that UI is subjective, or that Great UI takes too much effort, then this session is for you. Learn how to enhance clarity, reduce visual noise, lower barriers to entry, and make your interfaces discoverable, responsive, and a pleasure to use. It's all about making customers satisfied, and this entertaining and information-packed workshop will show you how. Part 1 of 3.

---

## **Press Start: Game Development for the Uninitiated**

*Shawn Rakowski*

What inspired you to start programming? I wanted to build games. What was your first program? A game. Are you building games now? No. Why not?

It has never been easier to begin building games. The paths to entry are numerous, the resources are abundant, and the benefits are many. A hobby in game development provides an outlet for artistic expression, an opportunity to geek out about math or physics, the chance to make a little money on the side, and, if nothing else, a concrete way to improve your skills as a professional software developer.

This session seeks to inspire you to adopt a hobby in game development, and will provide you with a variety of methods and resources for doing so. We will discuss why, as a professional software developer, it may be worth taking a look at game development, and will cover such topics as game jams, fantasy consoles, engines and frameworks, tools for creating graphics and audio, resources for learning, channels for distributing your game, and more.

---

## **Making Accessibility Testing Suck Less: An Intro to Pa11y.**

*Jennifer Wadella*

Often the hardest part of any problem is simply how to get started. On the ever-evolving web accessibility is a matter of ongoing importance: the brilliance of your code or sleekness of your UI is inconsequential if your app or website is unusable to some of your users. With a million other issues already on your plate how do you find a way to started on accessibility testing? Pa11y to the rescue! Pa11y is a lightweight command-line accessibility testing tool with enough flexibility to integrate results into your current testing process. This talk will explain what pa11y does and does not cover, review examples of both command line and scripted usage, dive into the pa11y webservice and show how to modify output to work in your current testing setup. Bonus content: how to convince the rest of your team and business why accessibility is worth prioritizing and how getting started with low-hanging fruit can vastly improve your product.

---

## **Services Assemble! Apply MVP Principles to Create Cutting-Edge Microservices Without Chaos with Spring Cloud/Netflix OSS**

*Mark Heckler*

In this session, learn how to use proven patterns & open source software to rapidly build a robust portfolio of microservices that provide a solid foundation for your dynamic and growing microservice architecture. This session addresses vital points such as:

- Configuration services
- Microservice registration and discovery
- Circuit breakers for graceful degradation
- Load balancing and intelligent routing
- Asynchronous messaging
- Reactive services, events, & backpressure

The presenter will demonstrate how to develop & effectively manage microservices using OSS tools employed by Netflix to keep movies streaming globally 24x7.

---

## **A Research Study into DevOps bottlenecks**

*Baruch Sadogursky*

We asked a Fortune 500 software delivery leaders what holds them back. This talk is the analysis of their insights on what bottlenecks they encountered in their DevOps journey. We share discoveries on what helped them to overcome the bottlenecks and how they plan to deliver even faster. Using this we define some unifying themes on what areas provide the greatest return in investments of time and resources.

---

## **Coding Naked – Unit testing those hard to reach places**

*Caleb Jenkins*

Code coverage with quality unit tests are your first line of defense to reducing technical debt, increasing code quality and accelerating your ability to change and adapt code (without breaking it) while continuing to add new features. Most TDD sessions focus on the easy to test areas of your code base that are almost never what you experience getting back to your desk. Come learn why TDD is not a fancy practice for the coding elite, but an understandable, obtainable and practical approach to delivering value for every developer, and how, when done properly, will increase communication and design between the business stake holders and developers. We will focus on practical steps to moving towards & embracing TDD. We'll overview the normal roadblocks that

people typically run in to, and practical coding strategies to overcome those road blocks on your way to embracing a Test Driven Development lifestyle - make coding without tests as uncomfortable as coding naked! - From the author of Automated Unit Tests chapter in the Wrox Book "Real World .NET, C# and Silverlight - Indispensable Experience from 15 MVPs, we will learn: *Distinguish between the 4 major elements of automated unit tests. Code, Tests, Testing Framework and Test Runners and how they interact with each other to round out your engineering practices.* Discover how Mocking Frameworks and DI make your tests easier to read and write in everyday life. *Dig in to better ways to write and organize your tests so that they communicate intent, document your code for you and bridge the gap between development and business needs.* We'll take a more specific look at those "hard to reach" places like the edges of your code, extension methods and other interesting scenarios

- Note everyone will leave their cloths on - it's not that kind of talk!

---

## **Essential Tools for Xamarin Developers!**

*Sam Basu*

Cross-platform mobile development has been democratized for .NET developers – thanks to Xamarin & Microsoft. However, professional Xamarin development isn't a piece of cake. Sure, tooling has come a long way, but successful mobile apps need a lot more.

Do you understand the whole Xamarin technology stack? Do you know what platforms to target & maximum code reuse? Let's do a run-down of the essential tools, frameworks and utilities that should be in every Xamarin developer's arsenal. The ubiquitous IDEs, necessary plugins, cloud backends, deployment tools, testing solutions and polished UI – all ready for your app development pleasure. How about adding some cloud-powered AI or Alexa voice assistance to your app? What can you do to delight your users, maintain developer sanity and deliver high quality apps? Let's stop reinventing wheel and ship apps faster – the right tools help!

---

## **Project Vienna: Machine Learning Process For Engineers**

*Jennifer Marsman*

Project Vienna is a project that is still NDA, but it will be made public at Microsoft's Ignite conference (last week of Sept 2017). It involves machine learning. It is also cross-platform and can be used with multiple languages, like R and Python, so it seems like a good fit for CodeMash.

---

## **Machine Learning at Scale, How to Keep Learning as Your Data Keeps Increasing**

*Matt Winkler*

Every year at Codemash, there are more sessions, that each keep getting more well attended, with people learning about Machine Learning. In this talk, we're going to address a challenge that's all too easy to encounter, the data you're using keeps growing and gets bigger than what you can handle on a single machine. Learn first about some of the key challenges with scaling machine learning to big data, and then what tools and techniques you can use to solve it. We'll cover software such as Spark for data intensive processing, and deep learning frameworks like Tensorflow, Chainer, and Cognitive Toolkit, and then we'll talk about the latest hardware advances in GPUs for speeding up processing. This talk will use Azure for processing, but the approach is applicable on-premises and other clouds.

---

## **Becoming a Servant Leader, Leading from the Trenches**

*Sarah Dutkiewicz*

When you're doing the work alongside those you manage, you may be seen as a servant leader. But what does that mean? What is servant leadership? Sam Walton of Walmart, Jack Kahl of Manco (Duck Tape), and many others have blazed this trail. For those of us who enjoy both our tech and management roles, servant leadership is something to understand how we can best serve those that we lead while in the trenches alongside them. In this session, we will explore what makes a great servant leader and how we can stay techies while being managers as well.

---

## **General Session 05 - Thursday 1:00**

### **Secure Applications, By Design**

*Craig Stuntz*

There is a lot of good security advice in the world, but checklists like the OWASP Top 10 do not tell you how to design security into your application. Where should a developer even begin? You'll leave this session with a process for building security in depth into your application architecture, using a human-centered user experience design, threat modeling, partitioning, defense in depth, and static analysis in continuous integration. Not yet another checklist, you'll learn how to make security the foundation on which the rest of your application is built.

---

### **A Brief Introduction to Concurrent Programming**

*Rob Keefer*

Unless systems you build are developed with concurrency in mind, they probably do not take full advantage of modern multi-core CPUs. Concurrent programming facilitates the use of multiple processors, but is difficult to implement using traditional approaches. Fortunately, new approaches taken by functional languages have begun to address these difficulties.

In this highly interactive session, volunteers from the audience will help present the basics of concurrent programming concepts, how these concepts have been addressed historically, and how functional languages such as Erlang, Clojure, and Haskell address these concepts today. With this knowledge you will begin to think about software architecture differently, and may even decide to introduce concurrency into your next project.

---

### **Putting the Visual into the Visual Studio Debugger**

*John Wright*

You've hit your Visual Studio breakpoint, opened the watch window and you're looking at a wall of nested collections and object graphs. You know the data you want is in there, but digging it out of the built-in data table UI feels like the "doing it the hard way" part of an infomercial.

Now, imagine being able to shape your data the way that makes it easiest to understand and debug, limited only by your imagination (and time, of course).

In this session, we'll take a look at the Visual Studio Debugger Visualizer feature, available in all version of Visual Studio (even the free one!) that lets you create your own graphical display of your .NET data structures while debugging. Shape the data the way you want to make it easier to digest

-- hide unimportant info, quickly expose deeply nested elements, compare large object graphs, or even instantly search Stack Overflow for help with an Exception. Display it in whatever way is best for your data and your understanding of it, act on the data and even modify it from your custom view. We'll talk through the options available to you, some of the limitations, and security risks/considerations involved.

This session doesn't require much deep knowledge of .NET, but the examples will primarily be in C#, WinForms and WPF. You don't need to know any of those beyond what you can Google or find on Stack Overflow, though the results will be useful to novices and well-weathered developers alike. (Note: While there is support for C++ visualizers, that won't be covered in this session.)

---

## **Teaching kids how to code with Sphero robotics**

*BJ Allmon*

"You teach what you know but you replicate who you are."

One of the most radical and rewarding things in life is experiencing kids learning something new. As a homeschool parent of five kids, I've had this opportunity over and over and it never gets old. It's about spending time with them and helping them solve problems.

Sphero provides a really great robotic software development platform with a robust community geared toward educating kids on robotic software development with an array of sphero devices and accessories.

Personally, I've used Sphero in homeschool co-op and youth code camp settings and every time it's a blast for the kids and me.

Learn how to take advantage of this platform to help kids quickly grow from learning programming basics to building complex games with Sphero.

---

## **Writing Scalable JavaScript at Cox**

*Cory House*

Scalable JavaScript involves much more than script tags. Today, package managers, modules, transpiling, bundling, dynamic loading, reusable component libraries, automated testing, and much more are attainable. In this code-oriented session, I'll share our unique approach for writing scalable JavaScript at Cox.

---

## **Hacking healthcare: real-world applications and challenges of health informatics**

*Marina Fedner*

Our healthcare system is at a tipping point, and fixing it is one of the most important challenges we face today.

In this session, you'll learn what we technologists bring to the table to improve patient outcomes and make healthcare safer and more cost effective. You'll also learn what goes on with your medical and insurance data behind the scenes at your doctor's office.

No specific programming language knowledge is required for this birds eye view of the taxonomies, workflows, standards, and APIs that form the infrastructure of our nation's health care system.

---

## **Command and Conquer the Front End with Elm**

*Tyler Jennings*

Tired of undefined exceptions in JavaScript? Heard that JavaScript can be a functional language, but have trouble breaking those imperative habits? Wanted to learn functional programming, but don't understand the fancy jargon? Learn to write reliable front end code on the first try with Elm by using the Elm Architecture to control the undefined, unknown, and bad patterns. Intended for those learning functional programming. By the end of the session you will have a good understanding of the Elm Architecture and how to integrate Elm into your existing codebase.

---

## **Walking the High Wire: Patching Erlang Live**

*John Daily*

Two years ago at CodeMash we discussed how Erlang was a paradigm shift masquerading as a programming language. Last year we illustrated how fun it is to write. This year we show off one of its most powerful aspects: maintaining a production system.

Erlang (and Elixir, its younger sibling) allow you to not only trace the behavior of a production system, but also to query the data in memory, replace your code on the fly once you've found the problem, and *fix* the data that the old code mishandled, so the system keeps plugging away. Look, ma, no reboots.

Also discussed will be other criminally under-appreciated languages with similar features, because really, can you ever have enough magic tricks in your repertoire?

---

## **Custom Middleware & Microservices with ASP.NET Core 2.0**

*Ondrej Balas*

At the core of ASP.NET Core is Microsoft's implementation of the OWIN standard, giving developers full control over the ASP.NET pipeline. In this session I will explain what OWIN is and how it can empower you to quickly and easily reconfigure an entire web application. Now in full control, you can do everything from injecting custom code (middleware) into any stage of the pipeline to running lightweight applications and microservices without MVC.

---

## **Start a UX Revolution at Your Workplace**

*Rhonda Grimm*

Are you ready to start a UX revolution at your workplace? Usability meetings are great way to start the movement. Usability meetings promote team building, cross-functional collaboration, networking, learning, sharing experiences, and having fun! There are a multitude of usability topics you can discuss with your co-workers, like paper prototyping, low-fidelity and high-fidelity mockups, usability testing, accessibility, interviewing users, field observations, form design, lean UX and agile, and so many more!

In this session, you'll learn how to get started (it's easy!), tips for running usability meetings, where to find content, and ways to get people involved. The best part is you don't have to be a UX expert,

and you don't have to have "manager" or "lead" in your title either. Everyone has a role in user experience. Start the revolution!

---

## **Brain-inspired computing: the truth, the fiction, and where we are today**

*Amber McKenzie*

Recent initiatives like the Human Brain Project and IBM's release of the TrueNorth chip have brought artificial intelligence and neuromorphic (brain-inspired) computing to the forefront of the public's eye. This presentation will delve into the challenges inherent in both mapping out and creating our own brain for both research and computing purposes. We'll look at the newest innovations and challenges for the future of the field of neuromorphic computing in terms of software, hardware and applications, and what it all might mean for the future of computing in general.

---

## **Bluetooth Low Energy on Android: Top Tips for the Tricky Bits**

*Stuart Kent*

Now that 90% of Android consumer devices and 100% of Android Things devices run software that supports Bluetooth Low Energy (BLE), it's the perfect time for Android developers to dive into the Internet of Things and start building companion apps or custom smart devices. Unfortunately, Android's Bluetooth stack has a well-deserved reputation for being difficult to work with. Join me for a journey through battle-tested strategies and code that will provide you with a roadmap for navigating the nasty parts. No prior experience with BLE is required; a gentle introduction is included.

---

## **Fearing the Robot Overlords**

*Christina Aldan*

We fear robot overlords as we do vampires, werewolves, and other baddies that go bump in the night. We worry that their intelligence will take our jobs, our livelihood, our freedom, and enslave us to their bidding; or at least that's how it happens in the movies. Popular culture capitalizes on this emotion to sell Terminator tickets and Asimov novels, but this is not our future. We can celebrate Artificial Intelligence as our greatest accomplishment, and harness it for the tool that it really is. This talk addresses our relationship with computers as tools and how we should allow computers to do the work, so people can do the innovating. Our horizon awaits.

---

## **Zero to DevOps in Under an Hour with Kubernetes**

*Dale Alleshouse*

The benefits of containerization cannot be overstated. Tools like Docker have made working with containers easy, efficient, and even enjoyable. However, management at scale is still a considerable task. That's why there's Kubernetes. Come see how easy it is to create a manageable container environment. Live on stage (demo gods willing) you'll witness a full Kubernetes configuration. In less than an hour, we'll build an environment capable of: Automatic Binpacking, Instant Scalability, Self-healing, Rolling Deployments, and Service Discovery/Load Balancing.

---



## General Session 06 - Thursday 3:30

### **NoSQL Shouldn't Mean NoSecurity**

*Matthew Groves*

As NoSQL databases increase in popularity, they also increase in popularity with hackers. NoSQL databases are vulnerable to traditional attacks like SQL injection (yes, really). Further, the rush to productivity leaves some of these databases insecure-by-design. As a result, ransom notes have plagued databases like MongoDB, ElasticSearch, Hadoop, and CouchDB. This session demonstrates security mistakes and prevention. We'll also look at what NoSQL vendors are doing to mitigate future attacks. Both devs and devops should come to this session, because the last thing either of you want to see is "SEND 0.2 BTC TO THIS ADDRESS 1zaGVjj9NcyvDLyYpCh33Msq TO RECOVER YOUR DATABASE!"

---

### **App architecture without RDBS vs NoSQL drama**

*Jeff Putz*

Between the tried and true, enterprise relational database systems that have been around for decades, and the relative new hotness of open source document databases, an entire subculture of zealotry has insisted that you choose a side. However, real life requirements demonstrate a time and place for both flavors of data persistence, and often a mix of the two. Regardless of your chosen development platform, the careful composition of more than one type of database can dramatically simplify your application and better handle larger workloads.

Objectives:

- Know when to use relational data, and when not to. Fight the urge to normalize everything.
  - The secret sauce to scale is often precalculation. Aggregate querying is a huge pinch point for many LOB applications.
  - Documents can hold a ton of serialized data that can be formatted and served to users. You might be surprised to find that relational databases are pretty good for this as well.
  - Cloud resources open up a world of possibility, where storage is cheap.
- 

### **You Will F\*\*k up & that's Ok (How to be a Male Ally in Tech)**

*Ronda Bergman*

Men, you have it easier than women when it comes to working in tech.

If this statement makes you angry good! Hopefully it makes you angry because you want to be a part of an industry that works for everyone. Do you read all the news reports about the diversity problems in tech and think someone should do something about that? Maybe that someone should be you!

You might be thinking how can I help fix this problem as a guy? Will the women around me think I'm sincere? Yes, you can be a big part of the solution. And, yes they might question your sincerity at first, but with time and effort you can show everyone that diversity is important to you too!

There will be mistakes along the way and that's ok. We make mistakes too. But by learning from those mistakes and continuing to try, we can make tech a better place for everyone.

Some of the topics we'll cover are: *Why diversity should be important to everyone* The difference between being an ally and being a knight in shining armor *What to do if you see inappropriate behavior* How to make your workplace more inclusive \* Actions matter even when it's just the guys

---

## **Risk Adaptive Application Security Using Machine Learning**

*Michael Meadows*

Policy-based authorization allows developers to implement rules-based authorization that extends far beyond role validation. This pattern is an important part of an approach called **Risk Adaptive Access Control (RADAC)**. One RADAC strategy that is enabled by policy-based authorization is leveraging machine learning to authorize actions based on behavior patterns as well as environmental factors. This allows developers to implement non-functional security features such as recognizing when critical data is being scraped by a script or a user manipulating data in a suspicious manner. In this session, we will review the concepts behind RADAC, and show real-world scenarios where it can be applied using Azure Machine Learning and policy-based authorization in ASP.Net Core MVC.

---

## **The Science of Great UI, part 2 (Design Like a Pro)**

*Mark Miller*

Get a big boost on your UI skills. Regardless of whether you're building interfaces for watches, phones, tablets, desktops, elevators, automobiles, or interplanetary spaceships; you'll learn how to expertly apply tools from the designer's palette to your designs. We'll discuss contrast, color, shadow, size, opacity, layout, proximity, borders, information dimension, symbols, and graphic languages. Part 2 of 3.

---

## **The Business Case for Blockchain: It Ain't Just Cryptocurrency**

*Heather Wilde*

"This changes everything."

Bitcoin's blockchain technology has triggered a new wave of innovation. Payments, websites, file storage, software, and even social networks will all change.

Sir Tim Berners Lee himself has expressed interest in blockchains recently.

Attend this session and prepare for how these technologies will affect your business and how developing with them can prepare you for the future.

---

## **Software is Not Hardware – Why the test rack is not the place to find a software defect.**

*DJ Daugherty*

Don't get me wrong, testing on the rack is important, but shouldn't be your only strategy... or your starting point. In this talk we will discuss effective ways of testing embedded software and hardware utilizing modern software practices, minimizing errors, testing time and budget.

Defects are expensive and undetected defects can be devastating. The earlier a defect can be identified and eliminated, the cheaper the cost. If a developer can find issues while writing unit-tests,

the developer can easily fix the issue because they have all of the information available. Integration testing provides the next level of protection... detecting specification and architectural problems.

Testing your hardware shouldn't be any different than testing your software.

---

## **Beyond JavaScript Frameworks: Writing Reliable Web Apps With Elm**

*Erik Wendel*

In times where a jungle of JavaScript frameworks wants to solve every conceivable problem in web app development, creating headaches and javascript fatigue in the process, Elm offers a different approach.

Elm is a functional programming language that you can use instead of JavaScript — or alongside it. It builds on the concepts virtual dom and unidirectional data flow popularized by React and Redux, but with a sound typesystem, built-in immutability, and an amazing compiler that catches errors before you even run your code. Simply put, Elm is a great language that will make your backend team jealous (unless they're already on the very similar F#).

In this talk you'll see how Elm works and learn how to use it to build a web app. I will emphasize the advantages and disadvantages it brings compared to the traditional JavaScript tech stack used at my most recent project.

This talk is not aimed at experienced functional programmers coming from languages such as Haskell, but rather JavaScript developers seeking a more functional approach in their daily work. It serves both as a introduction to Elm and typed, ML-style functional programming.

---

## **Do More With Less: A Kotlin Primer**

*John Keyes*

Have you been hearing a lot about Kotlin recently? Well, there's good reason. Created in 2010 by JetBrains, Kotlin is a statically typed programming language that was built with a focus on being safe, approachable, and pragmatic. Over the last few years, it has seen a dramatic rise in popularity due to first class support in Spring, Gradle and, most recently, Android. Used by companies like Pinterest, Evernote, Atlassian, and Netflix -- Kotlin looks like it's here to stay.

In this talk, we'll take a quick tour of the Kotlin syntax, what makes it so powerful, and set you on your way toward coding bliss.

---

## **Easy CQRS with ASP.NET Core & MediatR**

*Ryan Foote*

Are your controller actions gigantic? Is your service layer bloated? Are testing and maintenance a nightmare? Need a scalable solution?

The CQRS (Command Query Responsibility Segregation) development pattern can help address these issues. But implementing a distributed CQRS architecture can be intimidating. If not well designed, it can make your application exponentially more complex and difficult to maintain.

So why not start simple?

Using the popular .NET library, MediatR, it's super easy to write simple, testable, scalable CQRS applications.

In this session, we will explore CQRS and its core concepts while building a mind-blowingly simple ASP.NET Core application with MediatR. We will emphasize separation of concerns, dependency injection, code reuse, thin controllers, and unit testability. You will leave with knowledge and confidence to start using MediatR in your own projects.

---

## **Fitting Big Data Into Web-Based Dashboards: Pitfalls and Helpful Hints**

*Christina Holland*

Visualizing big data already comes with the innate challenge of packaging overwhelming mountains of data into a human-digestible format. The browser adds an extra level of challenge in limits on how many DOM elements can effectively render, limitations of HTML canvas, and the quirks of JavaScript. Understanding these limits is a great start to making the most of the browser.

Attendees will learn about: *The pros and cons of canvas, SVG, and non-SVG elements when it comes to interactivity and volume of data. When to use a library, and which library (low-level like D3 or high-level like Highcharts) When to roll your own visualization (no library) Pitfalls and tips of integrating into a framework (like React) Designing new non-standard elements to meet new use cases (with a real life case study in visualizing Spark apps) Workarounds for when there's too much information to fit on the page*

Expect to leave armed with information and tips on making useful, approachable big data visualizations in those little browser windows.

---

## **Configure, Control, and Manage IoT with Mobile**

*Jared Rhodes*

The internet of things allows for communication with devices through various means (without touch, mouse, keyboard, or a screen). Mobile devices give users a dynamic interactive experience with these devices by communicating over several different wireless protocols or through the cloud. In this presentation, we will see how to use Xamarin to create a cross platform mobile application to control devices of all shapes and sizes. After this presentation, attendees should be able to create a basic mobile application and have that application communicate with peripherals over Bluetooth and the cloud.

---

## **Prod Deployments: What should be the easiest part of your day.**

*Stephen Shary*

Deployments to production can be a point of fear and tension. We cover strategies that used at Kroger to move new features and fixes to the customer quickly and safely. Industry techniques increased production releases from once a quarter in the middle of the night to dozens of daytime production releases every day. We cover our use trunk based development, dark environments, blue-green deployments, hot toggles, and QA in production to reduce risk. We show how advanced metrics and multi-variate testing strategies can advance you to understand the true value of production changes to customers.

---

## Patterns for Reusable JavaScript Components

*Cory House*

Today's popular JavaScript frameworks offer powerful models for creating components. But are you designing your components for reuse? Whether you're working in Angular, React, Vue, or the long list of alternative libraries, there are universal principles and practices to consider for creating truly reusable components. This session explores hard lessons learned from publishing a suite of reusable JavaScript-based components at Cox Automotive. It's time to stop reinventing the wheel!

---

## Application Insights: The Magic School Bus for your Web Applications

*Kevin Griffin*

Ms. Frizzle and the kids are going on an adventure. Not into the human body. Not into outer space.

The Magic School Bus is going into your web applications to answer the questions that have been plaguing developers for centuries. How fast are my requests? How many requests am I getting per second? What resources are people asking for? What's throwing errors?

Application Insights is an application performance service for web developers. It works across multiple platforms, including .NET, Node.JS, and more. It is designed to watch your applications at the granular level, and report back everything it sees. All this while not impacting your performance.

In this presentation, we will explore adding Application Insights to an existing web application and looking through its cloud-based portal at overview health and performance. Examples from existing production applications will be used to show off many of the features and cool things you can do!

---

## General Session 07 - Thursday 4:45

### Threat Modeling for Secure Software Design

*Robert Hurlbut*

Threat modeling is a way of thinking about what could go wrong and how to prevent it. Instinctively, we all think this way in regards to our own personal security and safety. When it comes to building software, some software shops either skip the important step of threat modeling in secure software design or they have tried threat modeling before but haven't quite figured out how to connect the threat models to real world software development and its priorities.

In this session, you will learn practical strategies in using threat modeling in secure software architecture design and how to apply risk management in dealing with the threats.

---

### Unit Testing Your React App

*Steven Hicks*

Congratulations! You're building a sweet new web application with React. But what are you doing for unit testing?

Setting your environment up for JavaScript unit testing can be intimidating. React tooling is phenomenal, though, and makes setup a breeze. In this session, we'll look at how to get started writing tests for your React app. We'll dig into the features of Jest as a test-runner and test framework. Then we'll take a look at Enzyme, a framework for testing your React components

directly. Finally, we'll look at guidance and best practices. What should you test? When should you test it? How can you keep your React code friendly to unit tests?

You will leave this session ready to cover your React app with tests.

---

## **Everyday Elixir**

*Joel Byler*

Recently I've spent some time becoming more familiar with the Elixir by building a number of small applications. Some applications run on small hardware like a raspberry pi, others allow for real time communication like a chat room on a website, and others simply automate a few every day tasks. In this session I will take the audience through my journey of learning the elixir language, a few of the applications that I built along the way, and hopefully inspire others to pursue the task of learning a new language!

---

## **Build Your Own Teenage Daughter with Alexa and Azure Functions.**

*Ryan Miller*

Anyone with a 14 year old at home knows that life is awful (because we are lucky enough to be told on a regular basis.) However, sometimes you may find yourself sans-teenager either because your teenager isn't around or you have no teenager of your own (*sigh* poor you). In those teen-less times, who can you count on to remind you that life is so utterly unbearable? Who will tell you how much everything sucks? Fear not: with the proper skill, Alexa can help.

In this session, we'll see what it takes to build an Alexa Skill called Teenage Daughter. The code is C# and the service host is Azure Functions. We'll cover Custom Slots, Intent Schema, Utterances, and Testing; we'll delve into Azure Functions and how to create multiple environments to keep our production instance separate from development instance; and we'll cover any other tips and tricks that I've learned while creating my Teenage Daughter, which can be found at <https://github.com/Intranoggin/AlexaFunctions>.

---

## **Choice is Overrated - Designing Products That Know What You Want Before You Do**

*Heather Wilde*

According to CEO Aaron Shapiro, the next big breakthrough in design and technology will be the creation of products, services, and experiences that eliminate the needless choices from our lives and make ones on our behalf, freeing us up for the ones we really care about: "Anticipatory design". Rather than traditional UI/UX, where the tendency is to provide options in a participatory manner to determine a result, here you remove all choices from the user, and use predictive modeling to give an outcome to liberate them from so-called "decision fatigue,"

"Flow not friction," "convenience not choice," and "efficiency not freedom" are the mantras of anticipatory design.

In this talk, we'll explore some implementations of anticipatory design, discuss areas where it is done well, and issues with the overall movement.

---

## **Autonomy & Asynchrony: The key to designing reliable systems**

*Indu Alagarsamy*

The real world, with all its complexity, can at the same time be simple, elegant and beautiful. It thrives on autonomy and asynchrony, the two most important things that bring order to chaos. The real world does not pause for something to complete before moving on. And yet, when we write software, “Command and Control” is the norm. We find clever ways of doing this while keeping up with all the new technologies and the languages in fashion, all the while ignoring the realities of life.

In this session, explore an alternate universe in which Event Driven Architecture can power even the most complex mission-critical systems. Learn how communicating asynchronously via events leads to building systems that are autonomous and much more reliable. Embrace asynchrony and autonomy. Make the complex simple.

---

## **Scala for the Java Developer: It's Easier Than You Think**

*Justin Pihony*

Scala is arguably the de facto language of choice for the big data world but with a dauntingly steep learning curve. This session will immediately dispel that concern by exposing the minimal syntactic differences between Java and Scala. Participants will then learn how to use Scala for writing cleaner, more readable code. Once you've seen how easy the switch can be, you'll get hooked by its expressive features. Expression-based coding can return a value directly from an if statement and pattern matching can create clear, type checked code paths. This and more allows you to write less boilerplate and more maintainable, business code. You'll leave this session with enough of the basics to be able to make the switch. But be warned, once you go Scala it's hard to go back.

---

## **Cloudy with a (high) chance of uptime**

*John Pendexter*

Whether you are building a cloud or on-premise enterprise application, being able to recognize when you need to use a particular integration pattern is crucial to architecting a resilient and fault tolerant system. This talk will enable you to identify and describe the patterns of configuration management, graceful failover and service discovery. You will learn how and when to apply these patterns to ensure resiliency for your applications. Finally we will take an in-depth look at how Spring Cloud project's open source Spring Cloud Config, Hystrix and Spring Cloud Consul enable you to develop and support modern complex applications while applying these enterprise integration patterns.

---

## **Data Driven Devops**

*Baruch Sadogursky*

Devops is usually viewed from a traditional perspective of a collaboration of Dev, Ops and QA, driven by the change in Culture, People and Process. But how do you know where you stand and where to move? As in almost any field, data and metrics give you the gauges and instruments. In this talk we'll talk about the key measurements for the DevOps transformation process and provide you with 3 metrics you can start measuring tomorrow.

---

## **Lessons in Ethical Development I Learned From Star Wars**

*Jameson Hampton*

Star Wars fans may have been excited about last year's release of Rogue One because of the courageous new rebel characters it introduced, but it also provided a lot more insight into the Empire's Corp of Engineers and how their career paths ended with them working on building the Death Star. Engineers and developers, both in Star Wars and in the real world, have a responsibility to think about the ethics of our work and how it might be repurposed. Talking about good and evil can feel a little abstract, but Star Wars provides a great framework for thinking about morality in the context of our lives and work. This session will discuss ways of deciding for yourself what kinds of projects you are or aren't comfortable working and tips for self-accountability in your career, but all through the lens of characters we know and love from Star Wars - because talking ethics doesn't have to be scary or boring!

---

## **Batch and stream processing finally together with Apache Beam**

*Mete Atamel*

In this session, we will first take a look at the history of massive-scale data processing at Google with MapReduce, FlumeJava, MillWheel and how they contributed to the creation of Dataflow, a new paradigm integrating batch and stream processing. Next, we will learn about the details of Dataflow and some of its differentiating elements. In 2016, Dataflow SDK further transformed into an open source project called Apache Beam. We will take a look at Apache Beam and how it enables pipelines to run against multiple backends such as Dataflow, Flink, Spark and more.

---

## **Fire Up Your Mobile App!**

*Suganthi Giridharan*

Have you ever had an idea spark that would make a great mobile app? But when confronted with managing the data, authentication, and hosting, it's all too easy to give up on building the next million dollar product.

**Firestore to the rescue!** This session will take a look at a Codemash conference app built using Xamarin Forms for the front end and Firestore for the back end infrastructure. You'll see how the app uses Firestore to store and sync app data, authenticate users using Google and Twitter OAuth providers, track analytics, and monitor app performance all in one central place.

You will leave this session with an understanding of how to integrate Firestore with your mobile front end.

Let's go build apps!

---

## **Plug and Play JavaScript Components with Riot.js**

*Joe Erickson*

There are many web applications where a full, one page UI doesn't make sense, but we would still like to use self contained JavaScript components to modularize and modernize its functionality. Enter Riot.js, a component-based framework that lets you use regular old HTML, JavaScript and CSS to create reusable UI components and fits into a standard multi-page app. We'll look at how Riot.js fits into a typical web application and we'll build a couple of dynamic UI components and get them to communicate with each other.

---

## **A SOLID approach to Test Automation**

*Branden bellanca*

SOLID. The essential principles for developing within the OOP paradigm. When combined, these design principles make it easy for a programmer to develop software that is both maintainable and extendable. While our software is often analyzed and designed with this acronym in mind, our test code is often not. How can we take what we know about Object Oriented design and apply it to the way we design our testing assets? In this session, we will take these celebrated OO design principles and explore the ways we can apply them to our tests.

---

## **General Session 08 - Friday 8:30**

### **How to embed security into your process**

*Timothy DeBlock*

So you're interested in security. Maybe you're looking to get into the field. Maybe you understand the importance of security and want to improve in that area. Most developers want to get security right, but where do you start? This talk will provide actionable advice on embedding security into your current process. I will show you the resources, tools, and activities at your disposal. This will make your application more secure and also more efficient. Security vulnerabilities are ultimately just bugs in an application. Catching issues early saves time and money.

We'll talk about what I see developers are doing right. I'll share the difference between dynamic and static analyzers and where to use them in your process. We'll talk about tools such as the Zed Attack Proxy, Fiddler, and useful plugins. We'll also get into activities that will improve security and the process. Activities like Code review, threat modeling, and patching. I'll highlight the available resources to developers. The Open Web Application Security Project (OWASP) is a big resource for developers. We'll also dive into some of the leaders in the application security field. The last few years I've had the opportunity to work with development teams and help them embed security into their process. This talk is based on those experiences. Allow me to show you what the wonderful world of security and how to embed it in your process.

---

### **Mocking .NET Without Hurting Its Feelings**

*John Wright*

Unit testing has become an accepted part of our lives as .NET programmers. To help focus our tests to only the code we want to validate, Mocking Frameworks are a powerful tool in our toolbox. Like many tools, if you have an understanding of how the tool works under the hood, you can bend it to your will (and also know where it'll break if you bend too much).

In this session, you'll learn about the two main types of mocking frameworks: constrained frameworks (like RhinoMocks and Moq) and unconstrained frameworks (such as Typemock Isolator and Telerik JustMock). I'll dig into how the two actually do their magic and we'll discuss the pros, cons, and limits of both. We'll look at examples of how to use the frameworks in your tests, mocking out dependencies from your own code and even third-party logic.

You'll get the most out of this session if you're comfortable reading C# code and have a general understanding of class inheritance in .NET, along with some experience writing and running unit tests. Prior experience using mocking frameworks is not necessary.

---

## **I have people skills! The Importance of Emotional Intelligence and Your Career!**

*Michael Eaton*

“Developers don’t have people skills!” It may be a stereotype, but for some, it’s a badge of honor to be the the anti-social, soda-drinking hermit that people are afraid to talk to. Movies, TV shows, and comics make light of it, and so do many of us, but it really is no laughing matter. We strive to be the absolute best we can be at the technical side of things—we buy books, attend conferences and stay up late hacking on our side projects—but do very little to improve our core skills. Unlike IQ, EQ can actively be developed and improved. In this session, you’ll learn about the components of EQ, why they matter, and some strategies you can use to improve your own EQ!

---

## **Kubernetes Development on Azure made easy with Helm and Draft**

*Raghavan Srinivas*

Microsoft Azure can be used for container deployment in a variety of ways including support for Orchestrators like Kubernetes, Docker Swarm and Mesos. However, the abstraction for app development that support application self-healing, scaling and so on may not be at the right level. Helm and Draft makes this a lot easier.

In this primarily demo-driven session, we will cover Kubernetes deployments on Azure with some simple examples. We will look at Helm and Draft and how they can simplify app development significantly, like app. Scaling, rollback, etc. Helm is a tool that streamlines installing and managing Kubernetes applications, like the apt/yum/homebrew for Kubernetes. Draft works with pre-provided charts to deploy the apps. via Helm.

After attending this session, attendees will be able to walk away with a thorough understanding of Draft and Helm and how they can use these tools for deploying apps into Kubernetes clusters.

---

## **GraphQL -- Flexible APIs for Adaptive Clients**

*Brian Genisio*

REST is dead! Long live GraphQL! (Just kidding... this isn’t a zero-sum game). As the proliferation of clients emerge (Android, IOS, Web, etc), we are learning that REST does have some limitations, and you’ve probably seen them first-hand. GraphQL, by FaceBook, is an attempt to fix the limitations of REST by allowing the client to specify the data it needs via a single endpoint. Servers can expose any data from any source via a typed interface, and clients can gain the freedom to ask for exactly the data they need without over-fetching. In this session we will discuss the limitations of REST, the power of GraphQL, and some implementations. We’ll discuss some best practices, and the tooling that has emerged in this space to make your developer experience top-notch.

---

## **How does my computer talk to me? Peeling back the hood of your favorite voice assistant**

*Amber McKenzie*

Virtual voice assistants have become a major offering by all the big digital-oriented companies with Siri, Cortana, Ok Google and Alexa becoming common household names. But have you ever wondered how they actually work? We’ll walk through what goes into creating a voice assistant and how they function in the real world. The talk will also delve into future technologies and innovations

that might be coming. I'll also show a demo of my DIY voice assistant using Google Assistant API for Raspberry Pi 3.

---

## **Of Abstraction and Precision**

*John Daily*

The purpose of abstraction is not to be vague, but to create a new semantic level in which one can be absolutely precise. – Edsger W. Dijkstra

The entire history of software engineering is that of the rise in levels of abstraction. – Grady Booch

We live in a world dominated by a virtual machine (the JVM) and a virtual platform (AWS). These are concrete abstractions, despite the apparent contradiction in terms.

Virtualization, however, is just one form of abstraction. We'll talk about the evolution of programming languages, networking, hardware, systems orchestration... and while abstraction will undoubtedly be extolled as a good thing™ we'll discuss the sea monsters that dwell hidden behind the curtain, to mix some abstract metaphors. What is a leaky abstraction and what can you do about it?

This is, indeed, an abstract discussion; practical applications of this talk are left as an exercise for the attendee.

---

## **Brighten your day with Elm**

*Ingar Almklov*

What we ultimately want from the tools we use for development is for them to make our lives easier, and in this session we will look at how Elm can do exactly that. We will explore how we can use Elm's type system to make the compiler do much of the thinking for us, and how by putting some thought into how we model our data we can end up with code that is easier to read, understand and maintain. I will also share my experiences from being a frontend tech lead and talk about why it is important to focus on developer happiness.

Also, did you know that the compiler can output a JSON file with documentation for your code? This is pretty cool, and I will show you how we can use this to create living documentation with code examples and even show previews of your view functions!

If you are new to Elm you will leave this session with a deeper understanding of why developers using Elm really like it, and if you have used Elm before you will hopefully learn some new tricks to apply to your own projects.

---

## **Psychology Behind Compelling Apps**

*David Truxall*

An unfortunate truth about software is that most users don't stick with new apps very long. Sadly, there is no one secret, but addressing human psychology can make your code more compelling for users. This session examines the human psychology behind what makes some software stick and become part of our daily routine. As a developer, you need to employ these ideas when designing features you plan to implement or products you plan to launch. This session covers three separate

techniques rooted in psychology you can implement to make your user's experience awesome and keep them coming back.

---

## **Building Java Apps with Intelligence**

*Brian Sherwin*

With the explosion of the cloud comes the explosion of compute power to solve complex problems. We have been playing with artificial intelligence for many years. Cloud services are now powerful enough that you can begin to build AI into your applications with just few lines of code. In this talk, we will work together to build an Android app that will use computer vision to intelligently respond to live video. We'll explore creative new ways to make your apps come alive with incredible new intelligence capabilities.

---

## **Digging into DevOps with Terraform**

*Dusty Burwell*

Terraform, by Hashicorp, is one of a few tools core to the DevOps practice at Stripe. Terraform provides a powerful declarative programming language and runtime model tailored specifically toward the deployment and management of cloud resources. With Terraform, developers are able to quickly add or modify servers in the network, add new load balancers and DNS records, and, because it's all in code, they could even rebuild the whole network from scratch on a moments notice.

This talk introduces several patterns for working with Terraform. You'll leave this talk prepared to provision servers with configuration management systems like Ansible or Chef, to modularize your infrastructure deploys, to maximize uptime through rolling deploys, and to effectively use Terraform in a team setting. This talk assumes some basic knowledge of cloud infrastructure.

---

## **Introduction to React + Redux + Redux Observables**

*Jason Farrell*

React is a popular JavaScript framework written by Facebook to deliver highly responsive views. While smaller in scope than a framework like Angular, it integrates well with tooling like Redux to enable highly deterministic state management. In this talk, we will explore the ReactJS ecosystem and show how to implement an app that uses Redux to manage state and explain how use of Netflix's ReduxObservables can bring even more flexibility to the existing Redux state management model.

---

## **Ride the rails: Handling errors the functional way**

*Sam Hanes*

Invalid email addresses. Malformed JSON. Database timeouts. Though we frequently focus on the happy path of a given use case, sooner or later we have to account for everything that can (and will) go wrong. Learn a powerful, functional approach to error and exception handling that will allow you to compose your workflow cleanly and simply. This session will focus on Railway Oriented Programming (ROP), otherwise known as the Either monad. We'll dive into the nuts and bolts of the pattern, and then use it to simplify some real-world F# code examples. No functional programming

experience required – just an interest in learning a functional solution to a ubiquitous coding problem.

---

## **299,792,458 Meters per Second**

*Anthony Eden*

Network latency is real. If you build systems that communicate on the same machine or the same local area network, then you probably don't feel the pain, but the moment you start distributing your systems across multiple data centers that pesky speed of light limit starts to rear its head.

In this talk I'll teach you how to laugh in the face of physics, building systems that give you the performance you need, spanning across the globe. Using real-world lessons learned from building and operating DNSimple, I'll show you how to decide what tradeoffs you have to make to meet the needs of your customers, developers, and operators. By the time we're done, you'll have a better understanding of design techniques that you can use to make your systems perform well, even if they are spread across the planet.

---

## **General Session 09 - Friday 9:45**

### **JSON Web Tokens Suck**

*Randall Degges*

JSON Web Tokens (JWTs) are all the rage in the security world. They're becoming more and more ubiquitous in web authentication libraries, and are commonly used to store a user's identity information.

In this talk Randall Degges, Lead Developer Advocate at Okta, will take you on an extensive tour of the web authentication landscape. You'll learn how JWTs and Sessions work, and why JWTs are the worst possible solution for solving web authentication problems.

You'll also learn the real reason behind JWTs rise to fame, and better ways to secure your websites that don't involve misplaced hype.

---

### **Introducing Domain-Driven Design**

*Steve Smith*

Domain-Driven Design (DDD) provides a set of patterns and practices for tackling complex business problems with software models. Learn the basics of DDD in this session, including several principles and patterns you can start using immediately even if your project hasn't otherwise embraced DDD.

Examples will use primarily C#/ .NET.

---

### **Yoga & Mindfulness for People Who Stare at Computers All Day**

*Julie Cameron*

In this increasingly busy world we live in, office and tech workers often find themselves hunched over computers, pushing to meet deadlines, to clear out todo lists, and to keep up with our ever-changing, high-speed, information economy. If left unchecked, this drive to keep up - and the work

habits we develop to do so, can start to build tension, stress, and even illness. All of this can impact overall happiness, health, and ability to perform at work.

Join me for a look at how crafting a personal yoga and mindfulness routine can help to reduce the negative effects of our digital-based lives. We'll walk through some quick and simple mindfulness techniques to help better your day. We'll discuss some of the reasons you might want to introduce yoga and mindfulness practices in your workplace. And finally, we'll end our session with a short, chair-based yoga sequence that you can practice just about anywhere.

---

## **Hacking Radiowaves for fun (but not profit)**

*Craig Hills*

As a child, did you ever play with walkie talkies? Do you remember how fun it was to talk with friends and family from the other end of the house, or across the neighborhood?

Capture that feeling again, except this time learn some science behind it and how to use the same technology to contact someone on the other side of the country with just a radio and some wire. This session is an introduction to amateur radio aimed toward IT professionals. Accordingly, the focus will be on how to get licensed, and introducing the more technology focused aspects such as digital communication modes, antenna design, and commonly used software.

While amateur radio cannot be used for commerce, many lessons learned in the hobby are almost directly applicable to everyday work and you will walk away with a better understanding of wireless technology.

---

## **Event-Driven UX in the Real World with Angular and Socket.io**

*Michael Meadows*

Event-driven UI programming using WebSockets is great! The problem is that every demo app is a chat app or a game. This session focuses on real-world examples using socket.io and Angular (with a little bit of reactive programming sprinkled in) to demonstrate how event-driven user interfaces can dramatically improve a user experience. Attendees will leave able to design practical event-driven applications that provide a richer level of interaction with a lower level of programming effort.

---

## **Public Speaking without Barfing on Your Shoes**

*David Neal*

Sooner or later, we are called on to speak. Out loud. In front of an audience. We may have to present something to our team, our leadership, or even a larger group of (--gulp--) mostly strangers. Public speaking strikes fear into the hearts of nearly everyone... especially those of us who are very shy. Imagining people in their underwear does not help.

I have been speaking at conferences, user groups, and meetups for many years. However, as a shy introvert, this is not something that comes natural to me! If I can do it, so can you! In this fun and light-hearted talk, I want to share with you what I have learned from my hard-won experience. Introverts and extroverts alike will walk away with practical tips on preparing and giving effective presentations.

---

## **JSR 375: New Security APIs for Java EE**

*Ivar Grimstad*

The Java EE Security API 1.0, JSR 375, is part of the Java EE 8 platform. New functionality includes authentication mechanisms, identity store access, and a new security context. This session describes the new APIs, takes a detailed look at each API's syntax and semantics, and provides examples of how they can be used by applications.

---

## **Accessibility Cookbook: 10 easy recipes**

*Martine Dowden*

If you ever use the elevator, subtitles to watch that cat video during a boring meeting, or tabbed through a form, you have used an accessibility feature.

As a community we understand that accessibility is necessary but when it comes to implementation, we often fall short. In this talk you will learn

- Why accessibility is important
  - 10 common accessibility fails
  - How to test for them
  - Solutions anyone can implement
- 

## **Electron in the Real World: Architecture, Capabilities, and Performance**

*Matt Behrens*

Electron gets a bad rap for being a slow and resource-hungry wrapper around Chrome, but in reality, it's a very capable desktop application framework that's well-suited for building complex software. Leveraging the capabilities of Node.js, you can read and write files, work with databases, speak bespoke IP protocols to control connected devices, and then wrap it all up behind a responsive, quick, and friendly user interface.

In this session, we'll explore a successful Electron application that does all of those things. You'll learn about Electron's general architecture and how it works to make all these things happen. You'll hear about how the team of React, Redux and TypeScript all worked together to make this complex application robust and understandable. We'll talk about lessons learned while building this application, including how some realizations and subsequent refactors simplified it and made it faster and easier to implement new features. We'll cover what was done when performance problems reared their ugly heads so the application could easily handle a hundred state updates per second, smoothly updating metrics and plotting devices' positions on live maps. We'll also talk about what the stack's limitations are, so you know what Electron is good at, helping you to leverage its strengths and avoid its weaknesses.

If you've been wondering whether Electron is right for your project, this tour through the development of a complex and successful application will help give you the information you need to make a well-informed decision and get you started using Electron to build it.

---

## **Securing without Slowing: DevOps**

*Wolfgang Goerlich*

We turn to DevOps for speed. We turn to Cloud for flexibility. We adopt faster, leaner, more collaborative processes to drive change. And then? We turn to information security for protection. But can we secure the technology without slowing the pace? This session presents an entirely fictional development organization adopting DevOps. We will discuss which traditional software security processes work, and which ones fail entirely. Awareness training, muscle memory, culture shifts, all will be brought together. The presentation will conclude with take-aways for applying security to your DevOps team without slowing down.

---

## **Let's Augment Reality in Unreal Engine**

*Ellen Mey*

This interactive step-by-step guide for getting started with augmented reality in Unreal Engine will give you the foundational knowledge for taking advantage of ARKit in your next UE Project. Using one of the built-in templates, we'll walk through the steps of implementing the ARKit plugin, adding an AR Camera, and allowing users to add objects to the screen with a tap.

---

## **Sondheim, Seurat and Software: finding art in code**

*Jon Skeet*

White. A blank page or canvas. The challenge: bring order to the whole. Through design. Composition. Tension. Balance. Light. And harmony.

The opening lines of Sondheim's "Sunday in the Park with George" were written with fine art and paintings in mind, but ring true for software engineer as well. As Seurat created grand visions from tiny dots, so software solves huge problems one line at a time.

Some parallels are obvious - the suggestion of "favor composition over inheritance springs to mind" - but others take more reflection. This talk offers musings on each element, and links each back to the cornerstone so often neglected in the midst of design docs, specifications and linters: our shared humanity.

What turns "good" code into "great" code? How subjective is this? How does it affect how libraries connect with their consumers, and how applications connect with their users?

At the end of this talk, you'll be left with more questions than answers - catalysts for your own thoughts on the nature of code and art, and for further discussion.

---

## **A Game of Theories: why languages do what they do**

*Rae Krantz*

How often do you search for "how to do [x] in [language]"? Maybe you're a Ruby dev who wants to know how Erlang ensures code is executed after an error, or maybe you're a Pythonista who thinks there *must* be a simple JavaScript equivalent to `range()`. This talk will go a step beyond the answer, and look at what we can learn about languages by comparing how they solve our common problems. What does a language's built-in-functions tell us about why they were created and how they are best used? Let's find out. (language list also includes Clojure and Go, but the questions asked are broadly applicable)

---



## General Session 10 - Friday 11:00

### Image Recognition with Convolutional Neural Networks using Keras and CoreML

*Tim LeMaster*

Machine learning is beginning to have a major impact on software development for all platforms and industries. However a life of developing software in a traditional explicit manner is not the best training for developing machine learning algorithms. Software developers faced with problems like "What is in this image", "Is image a face and is it smiling?", or "Is this a hot dog?" would face much difficulty without a machine learning algorithm approach.

This session explains what a neural network is in simple terms. That definition is expanded to include Convolutional Neural Networks and why they are great for image recognition. A sample neural network implemented in Keras and trained with the MNIST dataset will be demonstrated to recognize handwritten digits. This session will conclude with an sample iOS application using CoreML to run the neural network directly on a phone and recognize digits in real time.

You will leave this session with some ideas on how to begin to approach problems best solved with machine learning algorithms.

---

### How to Make a Dysfunctional Team Functional without Resorting to Death and Dismemberment

*Kim McGill*

Everyone has been there at least once; part of a team or group that has become so dysfunctional there seems to be no way to move forward. This type of behavior often leads to deep frustration and despair, as people in these kinds of environments seem to only respond to negative and often dramatic communication. However, this type of dysfunction is not usually the fault of any one person or group of people on the team but is the result of the team's inability to recognize and deal with a needed change in a positive and healthy way. In this interactive discussion, you will use the latest in behavioral change modeling to learn how to identify the stages of change. You will come to understand and leverage the decisional balance shifts at each stage. You will discover ways to help people move forward in modifying unwanted behaviors in a healthy way. And, most importantly, you will find out how to support people in maintaining those behavior changes into the future. This session will give you the tools you need to help move a team from dysfunctional to functional without having to remove a single head.

---

### git gone wild: how to recover from common git mistakes.

*Magnus Stahre*

"Oh no! I just typed something into git and now my work is gone! Ahh!"

Don't panic. Contrary to popular belief, git is one of the most non-destructive version control systems available. When used right, git never loses any data, and even when misused, chances are very high you can still recover.

This talk will present turn-by-turn directions on how to recover from a wide array of common git mistakes, and also provide some workflows to minimize them.

Examples include, but are not limited to: *Changing spelling errors in commit messages*. Split commits into multiple commits with logical separation. *Combining multiple commits into one*.

Resetting the base commit when you accidentally branch off of the wrong branch. \* Recovering from merge conflicts.

As well as more advanced topics, such as: *Extracting parts (e.g. sub-directory) of a repository as a new repository*. Merging multiple repositories into one while retaining version history. \* Removing confidential data accidentally committed.

---

## **Building an Artificial Pancreas**

*Timothy Mecklem*

Type 1 Diabetes sucks. People with this autoimmune disease have to constantly monitor & control their food, blood glucose, insulin, physical activity and other factors. Even with advancements like insulin pumps and continuous glucose monitoring, the process is still manual and error prone. What if someone with T1D could delegate the constant cycle of monitoring, predicting and controlling to a computer in their pocket?

In this session, you'll learn about an open source project called the Open Artificial Pancreas System, or OpenAPS, and how it closes the loop on a cycle that the rest of us take for granted. Then we'll explore how Elixir and the Nerves project are being used to build a next generation artificial pancreas system.

We'll touch briefly on the topics of hardware prototyping, reverse engineering and decoding binary data and how it all came together in one project. Before the session is over, you'll see an embedded Elixir application talk to a commercial insulin pump using off the shelf hardware.

---

## **The Science of Great UI, part 3 (Metrics and Business Case)**

*Mark Miller*

Think great UI/design is subjective? Is your organization reluctant to invest in improving user experience? See definitive methods, ranging from brain-dead simple to mind-blowingly high-tech to help you measure design quality. Learn how to find and use freely-available tools to analyze design quality. Mark will also show how to present a compelling business case to justify investment in great design. Part 3 of 3.

---

## **Designing Cross Platform Applications on .NET Core**

*Kevin Griffin*

If you have been developing .NET applications within the past 15 years, you have traditionally been supporting only Windows. Because, of course, the .NET Framework was designed to only run on Windows.

If you were adventurous, you could attempt to run your application on Linux or Mac using the Mono compiler. However, some inconsistencies in the implementation of common libraries meant you had to target one environment or another -- never both.

With the release of .NET Core, the prophecy of being able to build true cross-platform applications using .NET is coming true! However, it isn't as easy as File, New. Design decisions you make can impact your ability to easily work across platforms. In this presentation, attendees will learn how to create cross platform .NET Core applications and how to avoid the pitfalls that will land them back into Windows-only mode.

---

## **Intellectual Property Fundamentals for the Technologist**

*Jeff Strauss*

Today, more than ever, technologists need to be aware of the basics of intellectual property law and its broad effects on the decisions you make every day. There is a prevalent myth that if something is available on the Internet, it must be free to copy and use as we please. Nothing could be further from the truth. In this session, learn some of the core principles of IP, including copyright, trademark, patent, and trade secrets. Become a better informed technologist and gain the confidence to discuss these topics with your professional advisors and with your own team.

---

## **Fiber or Glimmer? Two Approaches to the Component-Centric Future**

*Tommy Graves*

The rendering wars have begun! Front-end templating frameworks like React, Angular, and Ember have all generally settled on a similar component-based paradigm, but they're still differentiated by their vastly different rendering engines. This talk will dive deep into React's rendering engine, codenamed Fiber, and Ember's rendering engine, Glimmer. Both engines have recently revealed brand-new and rewritten implementations aimed at optimizing for performance. Fiber attempts to do as much work as possible between each frame to make sure the browser is always free to paint the page. Glimmer takes a totally different approach; it pre-compiles templates into a bytecode that is then interpreted by a "virtual machine" in order to perform the most optimized updates possible. After we dig deep into the code behind the actual implementation of each engine, attendees to this talk will walk away with an in-depth understanding contemporary renderers along with some knowledge about what factors to consider when trying to choose a framework.

---

## **How To Write Really Bad Tests**

*Dan Wiebe*

Most of the time, when you attend a talk, or even a course, on test-driven development, you're taught how to write good--although frequently simplistic--tests. When you go back to the real world and are faced with real-world code where simplistic principles are hard to apply, there's seldom anyone next to you to show you what's a bad test: you have to learn to identify bad tests the hard way, by writing them and then discovering that they don't test what you think they test.

But in this talk, we'll discuss exactly how to write the world's worst tests, with a number of examples from real life and detailed evaluations of just why those tests are so bad. You'll learn fundamental principles for writing bad tests, so that this time when you go back to the world, you can write bad tests with perfect intention and purpose.

Or not: your choice.

---

## **Using Neural Networks to Generate "Game of Thrones" Scripts**

*Jennifer Marsman*

Machine learning enables us to make future predictions based on patterns learned from historical data. If we apply these principles to a large body of text, we can predict what the next word or character in a document will be and generate writing in a similar style. This will help us solve a

major #firstworldproblem: let's build our own George R.R. Martin to generate GoT scripts (yes, I know, he didn't write the TV scripts). In this session, I will teach the basics of neural networks, the different types, and where each excels. Then I will demonstrate how to generate text using machine learning by training a model on the scripts of the "Game of Thrones" television series, and having it output new text in the same style of writing. You will walk away with an enhanced understanding of machine learning and neural networks, and maybe even some predictions for season 8!

---

## **Springing into Kotlin: How to Make the Magic even more Magical**

*Mark Heckler*

Developing applications with Spring Boot is already a pretty magical experience, but could it be better? Can the magic get any more powerful than it already is?

In this talk, we'll look at how Kotlin can be used to reduce boilerplate, increase code quality, and elevate your wizardry. The presenter will demonstrate how to begin incorporating Kotlin into your existing Spring applications and, once you start your journey, where it can lead. Come to this session to level up on Spring+Kotlin! (Robes optional)

---

## **Flying High with Xamarin!**

*Sam Basu*

Cross-platform mobile development has been democratized for .NET developers – thanks to Xamarin & Microsoft. Let's build a cool aviation app together – targeting all platforms from the comforts of Visual Studio on Windows or Mac. Real world apps need cloud data connectivity, navigation, storage, lifecycle management, plugins and polished UI – let's do all that. Let's take a look at some must-have tooling for professional Xamarin developers. Let's dip into device capabilities, reuse libraries and elevate the user experience. Oh, and we will throw in some passion for aviation in the app, like private jets and supersonic possibilities. Loads of fun while learning cutting-edge mobile development – you in?

---

## **Test Driven Development for DevOps**

*Matt Williams*

Infrastructure as Code has been a DevOps mantra for several years, but there are still a lot of areas where operations can learn from Development. Test Driven Development aids creating stable applications; it can also be used for developing and testing infrastructures, systems, and architectures.

Tools such as Ansible, Chef, and Puppet are part of the solution, but there is more to do than to define an ideal state and let the tool figure it out. This session provides a pragmatic guide for using tests to drive Infrastructure and system development with real life examples of what to do and what *not* to do.

---

## **Chrome Developer Tools: Raiding the Armory**

*Greg Malcolm*

The Chrome Developer Tools are absolutely packed with features. Some are incredibly useful. Some are just plain surprising. For example: Turn the browser into an editor, fuzzy search source

filenames, display data in the console in tabular format and much, much more. In this session, we will fix up the storefront for Wacky Wanda's Wicked Weapons and learn some console wizardry in the process. In the course of our tribulations we will correct styling problems, enhance our debugging skills, and clean up server-side snafus.

---

## General Session 11 - Friday 12:15

### How Do I Secure My Micro-Services?

*John Melton*

Micro-services have become a very popular technique for application development. The paradigm brings significant value for certain types of applications, and the community certainly seems to be trending in this direction over the last few years. As with any architectural style, there are lots of trade-offs. When considering security, there are some unique benefits and challenges associated with micro-services. This talk will look at some of the interesting issues that arise when securing real world services and give some practical advice for those dealing with this task based on the presenters experiences. This talk will cover security-relevant architectural and design changes that arise when migrating to a micro-services architecture, and will provide recommendations on how to leverage the benefits and address the challenges from these changes.

In this talk you will learn: *Security-related architectural and design concerns related to micro-services* How these architectural changes benefit and challenge security in a system *Questions you need to address as an architect or developer to ensure appropriate security protections exist in your system* Provide hard-earned recommendations on how to leverage the benefits and address the challenges from these fundamental architectural changes to better secure your systems (read: I failed so hopefully you don't have to!) *Pointers to open-source tools that can be applied to benefit from these changes* How your team, architecture, developers, and operations staff are affected by the fundamental changes: and how to take advantage of the new paradigm to gain greater security

---

### Giving Clarity to LINQ Queries by Extending Expressions

*Ed Charbeneau*

In this session we'll learn about .Net Expression trees by discovering how they work and applying the knowledge to LINQ using the pipes and filters pattern.

In this session we'll learn about .Net Expression trees by discovering how they work and applying the knowledge to LINQ using the pipes and filters pattern. LINQ and Entity Framework are both commonly used in the .Net ecosystem, but even well-written applications can have LINQ queries that are difficult to understand. Because LINQ is so flexible, it can be written in ways that fail to communicate the developer's intent. Well-written LINQ should be so clear as to be self-documenting. To write clear LINQ, it helps to understand the details of a few LINQ components that improve LINQ's readability.

We'll be showing how to use a pipe and filter pattern to make LINQ queries easier to comprehend. We will take a deep dive into expression trees to understand how they work, and how to manipulate them for maximum re-usability.

---

### Breaking FrontEnd Paradigms with Vue.js

*Ignacio Anaya*

Vue.js is more consolidated than ever. According to [JS Frameworks Benchmark](#), it is one of the JavaScript frameworks with better performance. Vue.js is the future of the client-side MV\* world, and is one of the most promising alternatives, competing with industry giants frameworks, such as React and Angular. We will review the core concepts of the framework, and demonstrate how you can scale progressively, even when you have complex requirements.

---

## **Domain Driven Design: The Good Parts**

*Jimmy Bogard*

The greenfield project started out so promising. Instead of devolving into big ball of mud, the team decided to apply domain-driven design principles. Ubiquitous language, proper boundaries, encapsulation, it all made sense.

But along the way, something went completely and utterly wrong. It started with arguments on the proper way of implementing aggregates and entities. Arguments began over project and folder structure. Someone read a blog post that repositories are evil, and ORMs the devil incarnate. Another read that relational databases are last century, we need to store everything as a stream of events. Then came the actor model and frameworks that sounded like someone clearing their throat. Instead of a nice, clean architecture, the team chased the next new approach without ever actually shipping anything.

Beyond the endless technical arguments it causes, domain-driven design can actually produce great software. We have to look past the hype into the true value of DDD, what it can bring to our organizations and how it can enable us to build quality systems. With the advent of microservices, DDD is more important than ever - but only if we can get to the good parts.

---

## **How to Count Your Chickens After They've Hatched**

*Gary Short*

When you're one of the leading free range chicken farmers in the UK, and you buy your fertilized eggs by the ton, how do you know how many chickens you own after they hatch? How many are in each barn when you lock them in for the night? If someone stole a large quantity, would you even notice?

In this session I'll walk you through the IoT and Machine Learning solution I developed to solve this problem. Along the way I'll teach you several machine vision techniques as well as demonstrating to you how I adapted a well known clustering algorithm in order to help me short circuit the machine learning model training phase of the project.

---

## **Securing the last mile of IoT solutions**

*Anders Lybecker*

Security is a top concern for IoT implementers. IoT platforms have great security stories and track records around the cloud, but IoT security is much more than just the back-end. The server and desktop security knowledge does not apply directly to small less capable devices or \$5 sensors deployed in the field.

Make sure your webcam, electronic door lock or toaster does not take down Netflix, Twitter and Amazon like the IoT devices controlled by the Mirai botnet in 2016.

This session will discuss IoT security fundamentals and show the importance of applying security best practices, including how to implement connectivity and device security.

Objectives: *Understand the IoT security challenges* Learn the fundamentals of device design security *Discuss device security and provisioning from birth to grave* Provide a solid overview of the Microsoft Azure security patterns

---

## **The Wonderful World of Recursion**

*Jason Voegele*

Recursion is an indispensable tool in the functional programmer's toolbox, but when used improperly it can be the source of disastrous errors. The world's most popular programming Q&A site is even named after the dreaded stack overflow error that is the result of misuse of recursion! How can we take advantage of this fundamental tool without falling victim to its pitfalls?

Join me as we journey through the wonderful world of recursion where we will discover a treasure trove of recursive techniques, including tail recursion, mutual recursion, and trampolines. We will also learn how tail call optimization is achieved (or not) in various languages. Example code and discussion is presented in various functional languages including Clojure, Scala, Elixir, Haskell, and F#.

---

## **Tales from the Front Lines: Modernizing a Legacy System**

*Mark Ryan*

Monoliths! Working with legacy software applications can be a harrowing experience for any developer. In this talk, hear some ideas for how to improve and migrate away from a monolithic software project to a more modern setup.

Learn about moving libraries out of an existing application while preserving the file history in git and how to continually migrate changes as they are in flight during the transition, setting up packaging workflows using git and a continuous integration server like Jenkins or TeamCity. Walk through an automated deployment pipeline for an application.

Learn from experiences managing and unraveling codebases in both Java and C#, as well as techniques that would apply to any legacy system.

---

## **There Is No Spoon? Understanding 'Spoon Theory' and Preventing Burnout**

*Jameson Hampton*

Spoon theory is a metaphor about the finite energy we each have to do things in a day. While a healthy, advantaged person may not have to worry about running out of 'spoons,' people with chronic illnesses or disabilities and members of marginalized communities often have to consider how they must ration their energy in order to get through the day. Understanding how 'spoons' can affect the lives of your developers and teammates can help companies lessen the everyday burdens on their underrepresented employees, leaving them more spoons to do their best work, avoid burnout and lead fulfilling lives. This session will go over examples of how different people spend spoons in different ways and why empathy is so important! It will cover coping techniques for developers who are struggling to manage their spoons as well as suggestions for managers and leads to better accommodate their employees.

---

## **A More Secure World for Android Apps**

*Mercedes Wyss*

In the world of development, security is sometimes a subject that does not receive the importance it should have. This talk has as objective touch some points for make Android Apps more secure. Starting for something basic like use SSL/TLS certifies, use PUT and not GET for your http(s) requests, send and receive encrypted information using standards like JWT, adding reCAPTCHA, use Passwordless with technologies like Firebase Authentication or Auth0.

---

## **Creating Meaningful Motion**

*Brian Greene*

Animation, when done thoughtfully and with purpose, can provide meaningful feedback to users, reinforce your brand, and add visual interest to your website. There are many options for creating motion for your website, but more important than how to create animations, is defining why and when to add it.

This session will give you techniques to catalog existing animations, create and document a cohesive motion language, and we'll discuss accessibility implications.

---

## **Building Delivery Pipelines with Jenkins' Pipeline as Code**

*Kevin Fox*

*Jenkins*, originally known as *Hudson*, is the "leading open source automation server". For many organizations, it plays a key role in enabling continuous integration, continuous deployment and continuous testing. New in 2016, in support of common DevOps trends, *Jenkins* version 2 introduced a robust, extensible approach to managing complex automation pipelines as code. Offering a Domain Specific Language based on Groovy, a suite of plugins providing powerful pipeline abstractions and optimized support for common development platforms, *Jenkins'* Pipeline as Code provides a solid foundation for a team's continuous delivery practices. Join Kevin Fox, ICC's Enterprise Architecture Practice Lead, as he provides a practical overview of this important new capability for a popular open source platform.

This session will:

- Highlight the value of delivery pipelines for Agile development teams
- Explain the role of automation code in enabling delivery pipelines
- Introduce *Jenkins'* approach to Pipeline as Code
- Demonstrate practical aspects of pipeline code reuse
- Present effective options for managing pipelines for large development projects

---

## **Advanced Patterns for Automated UI Testing**

*Seth Petry-Johnson*

Automated browser tests can be a nightmare to write and maintain. Unlike unit tests they're tough to set up, slow to run, make permanent changes to data in the system, and they're incredibly brittle. Even a tiny CSS change can result in cascading failures!

You can't *totally* avoid these issues, but you CAN manage them by adopting a few key patterns and techniques. This session explores the lessons learned by one product team as they moved from manual regression testing to an automated process using Canopy and Selenium. You'll go beyond the basics of single-page testing and see real-world suggestions for managing test data, executing complex and multi-page tests, organizing and writing test code, and coping with "permutation explosion". You'll also learn techniques for involving the entire cross-functional team in the process.

This session is designed for developers and architects that know how to automate the browser, but are struggling to leverage that knowledge into a suite of maintainable UI tests that provide acceptable ROI. It focuses on patterns and principles, not a specific automation technology.

---

## General Session 12 - Friday 2:45

### **Enhancing Application Security: Understanding and Utilizing Browser Security Features**

*Kevin Cody*

Have you looked at HTTP headers lately? Not only are they unwieldy, but what do half of them mean? Furthermore, browsers are protecting us from more-and-more attacks, but what are all of these acronyms? SOP, CSP, XSS, HSTS, HPKP, CAA... at the end of the day, we just want useable AND secure applications. This talk will break down exactly what all of these acronyms and browser-enforced security policies mean. Attendees will learn implementation and long-term strategies in effort to increase security posture without potentially sinkholing your user's traffic. Whether you're a first time developer, multi-linguist application guru, or simply an app user who wants to know what all of this security fuss is about - this session will appeal to the entire security conscious gamut.

---

### **I'd Hire more Women if they Would Apply!**

*Ronda Bergman*

Maybe you're the one that's not really trying? Are you sending signals that say "Dudes Only"?

As a Chapter Leader of Girl Develop It (GDI) I have had numerous companies ask me why can't they get women to apply to any of their openings. They want to diversify but have a "pipeline" problem. My experience as a woman in tech and helping women enter the tech field through GDI tells me otherwise.

In this discussion we'll consider the some of the common reasons women mightnot apply to certain companies and or job postings, examine some of the "red flags" you might not realise you are waving, explore ways to make the hiring process more inclusive and discuss how to determine where the breakdown may be occurring.

---

### **Megahertz, Gigahertz, Registers and Instructions: How does a CPU actually work?**

*Kendall Miller*

For decades, we've been creating ever higher abstractions between ourselves and the computing hardware we're programming, but in the end whether you're writing JavaScript, Haskell, or Python it all comes down to 1's and 0's running through hardware patterns that were well understood twenty years ago.

We'll walk through the fundamentals of how CPUs "think" in an accessible way (no engineering degree required!) so you can appreciate the marvel that is the modern CPU, be it in a server data

center or your fridge at home. You'll learn how a CPU turns the code we feed it into actions, what's the big difference between an ARM and an Intel processor, how CPUs constantly optimize work for us, and where is it all going for the next few years.

---

## **Evolution of a RESTful Testing Framework**

*Kevin Shomper*

Subtitle: Increased Productivity on APIs for the Whole Team

Abstract: RESTful APIs are becoming ubiquitous in today's digital world. The Programmable Web now boasts over 18K APIs in its directory and the pace of publishing is accelerating. Every organization will need to provide or use REST APIs. Whether your team is a backend resource or API provider, client app development team, or you live in between these two as a proxy, you would benefit from a robust test framework. This session opens with a brief overview of the API landscape and unfolds the evolution of a test framework. Developed initially for API proxy developers, who needed simple, scripted, headless REST call and validation while maintaining client state; its execution model expanded for improved decoupling of API calls and asserts, API independence from client/provider, and dev/run perspectives. The concepts presented are applicable whether you choose to 'roll your own' solution or extend an existing REST testing client like Postman.

---

## **FP vs. OOP: Beyond the Bikeshed**

*Colin Jones*

In object-oriented languages like Ruby, people often say that "everything is an object"—but first-class *functions* have become standard for object-oriented languages too. C# has had them and other related features for years, and even Java is in the game these days! Some "functional" languages seem to have some very object-oriented-looking features, too. If that sounds wrong to you, let's talk about polymorphism and more in languages like Clojure, Elixir, and even Haskell! So where does object-oriented programming (OOP) end and functional programming (FP) begin?

In this session, you'll learn to look at these programming paradigms with a more critical eye, focusing on where FP and OOP folks fundamentally disagree, where these paradigms have more in common than the industry's rhetoric suggests, and how you can use those insights to make better decisions about the software you build.

---

## **Leveraging Spring Cloud Consul for Discovery**

*David Lucas*

Twelve Factor Microservices have needs no matter if you run them in the cloud or on premise. Externalized configuration, including service location, can be a barrier to entry. Join this session and "discover" how to leverage Consul and Zuul to achieve several of the 12 factors transparently. This session will include working code (Java and Kotlin) and exploring Spring Cloud Consul, Configuration, and Zuul.

---

## **Prototype, Collaborate, Innovate**

*Caleb Jenkins*

A UX Story from the worlds most used white labeled travel site. How we use Sketches, Prototyping and Innovation Games to collaborate with our customers and experience innovation breakthroughs. If a picture is worth a 1000 words, then a prototype is worth 1000 meetings! Come learn keys to better prototypes and to working across business boundaries.

---

## **R Performance (It's not R, it's You)**

*Tim Hoolihan*

Maybe another language has caught your eye because "it deals with big data better". Or you just think R is slow and showing its age. But perhaps these aren't irreconcilable differences. In fact, its ability to do operations on entire vectors or frames of data is relatively rare among languages. There are a number of techniques for dealing with common performance issues in R. Come explore some of the options available to speed up your code and deal with larger data sets.

---

## **Lessons Learned from Making Resilient Apps with Azure Mobile App Services**

*Matthew Soucoup*

Users of mobile apps are just that, mobile. And as such, any mobile app that deals with data needs to work reliably in all environments; from a WiFi connection with a high throughput Internet connection, to an intermittent cell connection that keeps dropping - it even needs to work for prolonged periods of offline use. Not only that, when the device gets back online, your app needs to gracefully handle data synchronization and any data conflicts as well.

In this session, I will share with you hard earned lessons on how to solve the problem of online/offline data resiliency and synchronization using Azure Mobile App Services based on using it in multiple Xamarin projects. You will learn how to solve the issues that arise with improper client-side data schemas, pushing data to the server without knowing when the connection will be interrupted, and reconciling conflicting data between the server and the app. At the end of this session you'll be well equipped to have your app ready for anything.

---

## **Rediscovering XP: Extreme Programming. Retro? or Resurgent?**

*Mark Windholtz*

In the late 1990's and early 20-aughts **XP: Extreme Programming** was disrupting software processes everywhere. The Values, Principles, and Practices have guided many software teams to success. Since then the term "Agile" was coined and has come to represent a variety of practices. In talking to team members, I discovered that many people no longer know what the original XP was about. Still today, successful teams are using some or all of what XP introduced. If your team is struggling, maybe you are missing an important part of the puzzle. This talk will be the story of XP and a review of the Values, Principles, and Practices. We will talk about how the Practices fit together and reinforce each other. And how some teams are missing either the Business Practices or the Engineering Practices of XP.

---

## **.NET IL: Into the Marianas Trench**

*Craig Stuntz*

Are you interested in writing compilers, targeting Web Assembly, finding security issues automatically, binary analysis, or understanding performance at a low level? While it's always good to know how your language works, the benefits of understanding the intermediate language extend to metaprogramming and analysis across multiple source languages. Learning how to work with intermediate languages allows you to write programs which would seem unattainable otherwise. You will learn not only how IL works but how it compares with LLVM IR, Java Bytecode, and other intermediate representations. No mere "deep dive," you'll leave this talk really understanding how C# turns into microcode and how to use that information to do "impossible" things.

---

## General Session 13 - Friday 4:00

### **ZAPping Security Vulnerabilities in Your Development Pipeline**

*Matthew Smith*

When your application has a security vulnerability, will you or an attacker find it first? Reviewing code and tracking down hidden flaws can be tedious and difficult. Let the free and open source OWASP Zed Attack Proxy (ZAP) help. This session walks through four ways to use ZAP: UI, command line, scripts, and automated development pipeline. You will learn how to attack a live application in all four ways, how to target specific areas of your application for heavier scrutiny, test for specific vulnerabilities, and incorporate ZAP with your development pipeline to automate the whole process including regression testing and report generation, so that vulnerabilities are discovered the moment they are introduced.

---

### **JSON parsing in Swift just got a whole lot easier!**

*Priya Rajagopal*

JSON is quickly becoming the de facto data exchange format on the web with more and more web services serving content in JSON format. In fact, with the growing popularity of NoSQL databases, content is even being stored as JSON. The ability to encode JSON to and from native language representations is quickly becoming integral to developing a web or mobile app. In this talk, we look at how we can accomplish this using Swift 4.

Swift 4 was introduced in summer of 2017 and with it came Codables. Swift Codables greatly simplify the process of encoding and decoding from native swift objects to/from other representations, including JSON. Codables does away with the tedious JSON serialization /deserialization logic you probably had in place in earlier versions. In this talk, we will introduce you to Codables and look at examples of how you can use it for JSON serialization.

Anyone building apps using Swift 4 will benefit from this talk.

---

### **Github Pull Requests for Everyone**

*Catherine Meade*

#### **Description**

Reviewing a pull request can feel like a chore. If done poorly, PR reviews can mean a few hours of attempting to understand both the problem and the solution, then checking that the result matches the design. Sure, many of us have the luxury of walking to our coworker's desk and getting a walk through. But what if the other dev isn't free? What if they live in another time zone? What if you

need a project manager or designer to look at your work, and they don't have a local setup or much dev experience?

**In this session, we'll go over some tips and technologies to make your pull request process a bit smoother. We'll discuss:**

- Writing clear issues/stories to build a good foundation
- Tools you can use for reviewing work with remote coworkers
- Keeping design reviews from turning into blockers
- Adding testing instructions to your PR description
- Leaving positive feedback so no one goes home grumpy

**This session is for you if your team uses GitHub, and....**

- You have a coworker or client in a different timezone
- You regularly need a designer to review your work
- You or your coworkers work remotely one or more days a week
- You regularly contribute to open source projects (or run your own!)
- You ever have code that needs a merge, like, now

---

## **Testing Layers in a Distributed Architecture**

*Karl Hughes*

Modern web applications often rely on internal APIs, multiple frontend applications, several third-party services, and dozens of vendor libraries. With complex distributed systems like this, how can we create a comprehensive test plan that covers everything without slowing development down to a crawl?

In this talk we'll discuss creating test plans for complex, service-oriented applications as well as some best practices for testing across a distributed architecture. We'll see how layers of unit, integration, acceptance, and end-to-end tests can greatly improve the reliability of our applications and make maintaining them much easier in the long-run.

---

## **Hey, You Got Your TDD in my SQL DB!**

*Jeff McKenzie*

When should we test our data? From an application development perspective, a unit test against a database isn't a unit test at all. Which makes sense -- connected systems introduce overhead and dependency, both of which reduce stability and decrease productivity during test-driven development (TDD). But if we wait for integration testing, critical functionality can get missed. In this session, we will discuss strategies for filling the data testing gap, directly within a Microsoft SQL Server environment. If you do a lot of work in T-SQL but aren't familiar with TDD, you'll learn the why and how of test-first development. If you're accomplished with unit tests, but never tried them in your database, you'll learn how to apply familiar concepts like setup, mocking, and assertion. We'll spend most of our time walking through a solution based on a real-world project, specifically using the open source tSQLt database unit testing framework.

---

## **Operating Petabyte-Scale Distributed Storage Systems with Human-Scale Humans**

*Dusty Burwell*

When operating a data system every byte is sacred. Trillions of bytes represent a trillion opportunities to ruin somebody's day. It's critical to manage data systems with the greatest of care. Your business depends on it.

In this talk you'll learn how we operate a large and growing storage infrastructure at Stripe. We'll talk about the tools and processes that lend us confidence to stay up to date with the latest software versions, migrate databases with billions of entries, and troubleshoot performance issues.

You should expect to walk away having learned important patterns that can apply to managing data systems at any scale. Every byte, no matter how small, is important to someone.

---

## **A Year of Rust**

*Stephen Zeidner*

Rust is a systems programming language with a focus on safety. I started learning Rust a year ago when I was looking for a different backend stack (I had been using PHP previously). Using Rust as a web language for a year taught me many things that I had not learned using other languages.

In this talk, I'll take you on a journey of what Rust taught me about programming and life. You will learn about subjects like the importance of composition, why object ownership is critical skill, and how Rust can give you more flexibility in your life.

---

## **Querying NoSQL with SQL: HAVING Your JSON Cake and SELECTing it too**

*Matthew Groves*

Until recently, agile business had to choose between the benefits of JSON-based NoSQL databases and the benefits of SQL-based querying. NoSQL provides schema flexibility, high performance, and elastic scaling, while SQL provides expressive, independent data access. Recent convergence allows developers and organizations to have the best of both worlds.

Developers need to deliver apps that readily evolve, perform, and scale, all to match changing business needs. Organizations need rapid access to their operational data, using standard analytical tools, for insight into their business. In this session, you will learn the ways that SQL can be applied to NoSQL databases, and what additional features are needed to deal with JSON documents. SQL for JSON, JSON data modeling, indexing, and tool integration will be covered. This session will focus mainly on CosmosDb and Couchbase, but it will also touch on emerging relational approaches to JSON data.

---

## **Understanding Your Application with Application Insights**

*Jeremy Hutchinson*

Stop flying blind and use Azure Application Insights to monitor availability, performance, feature usage, and exceptions. In this sessions, you will learn how to add Application Insights to any application, how to use it to make debugging production problems a breeze.kind of information it can provide and how to use metrics to make your application better for your users.

---

## Design and Dev Needs for VR and AR/MR Applications

*Amanda Lange*

Over the past few years many traditional developers have been looking for ways to dip their toes into the world of virtual, augmented, and mixed reality. However, there are a lot of important questions in creating this type of application. Developers want to know: how to get the right graphical fidelity for immersion, how to keep the user's attention, and, critically, how to stop the user from getting so dizzy they lose their lunch all over your nice headset? In this talk, Amanda Lange from Microsoft will use knowledge of the HoloLens and Oculus Rift to dive into some of these questions. This talk will focus on the Unity 3D Engine but also deliver practical design and dev tips that will help in these types of applications regardless of development platform of choice. We'll discuss the difference between VR, AR, and MR. No prior experience will be required in these fields so if you have ever wondered how to get started in moving your applications into VR this is the talk for you.

---

## How Not to Destroy Data

*Michael Perry*

CRUD apps destroy data.

Delete is an obviously destructive operation, but Update is no better. Updates overwrite the data that was once present. What would software look like if the only valid operations were Create and Read?

Historical Modeling is a technique for designing and building software systems that permit only inserts, no updates, no deletes. The only way to "modify" an object is to insert related records. You can then read the state of the object by querying for the presence or absence of those related records. We'll start with the assumption that all records are immutable and indelible. From there, we'll build a set of patterns that yield some surprising benefits.

With a historical model, synchronizing disparate nodes is trivial: just insert what you don't yet have. Building offline-first web applications is easy. Viewing a snapshot of the system at any point in the past is a simple query.

Let's build a historical model together, and see these benefits for ourselves. Let's put aside the assumptions of CRUD applications and discover a whole new set of patterns. Once you master these techniques, you will never destroy data again.

---