MASHED CODE MAGAZINE

2012 Issue

THIS YEAR'S FEATURES

A letter from Dianne Marsh
Interview with the Koan Masters
Getting into Mobile: App or Website?
Oh, the Methods You'll Compose

Welcome to CodeMash v2.0.1.2

Welcome to the second issue of *Mashed Code Magazine*. The large number of downloads of the first issue (5000 and counting) gave us the steam to start earlier and work harder for all of you. We hope you find the result even more valuable than you did last year.

This magazine's intent is to expand on the conference—think of it as the conference guide you've always wanted but never had, with bonus technical content. To support that, we have expanded coverage of the conference, yet maintained the technical content. Dianne Marsh, conference organizer extraordinaire, begins by setting the context for us this year and grapples with the staggering 20 minute sellout of 1200 tickets and over 700 session submissions. Then Daniel Hinojosa continues the trail that Corey Haines blazed last year by investigating another set of internationally known projects with CodeMash origins: Ruby and Scala Koans. Following that, fearless KidzMash leader Melissa Insko shows us why CodeMash is the most family-friendly conference around.

There is a boat-load of technical content too. Once again, we have several stellar contributions from the CodeMash community on a variety of technical topics. If you're still contemplating whether to go with a mobile app or mobile website, Amelia Marshall is back this year with an article that will help you decide. And if you didn't catch Tim Berglund's sensational video "Oh, the Methods You'll Compose" (written in iambic pentamater, no less), we've got the prose version of it here.

Thank you—and thanks again and again—to the world-class conference organizers for trusting this project and allowing the magazine to become a part of the conference. Please stop them in the hall (if they're not racing off somewhere to help someone out) and personally thank them. The reason they sacrifice an inordinate amount of personal time for CodeMash is to make sure you enjoy it. We also want to thank the authors and advertisers for giving us something to publish. Finally, thanks to the second-time *volunteer staff*: Jason Gurik, Matt Darby, Matt Casto and Sara Smith.

Whether you are new to the conference or you are a CodeMash veteran, you will find insightful information about the conference and other topics in here. Enjoy the conference, chill out in the water park and be thankful for the chance to be at the hottest developer conference in the country.

For questions, comments or suggestions, contact us at mashedcodemag@gmail.com or on Twitter @mashedcodemag.

Creator & Curator
Nick Watts

Table of Contents

2012 Volunteer Staff

Nick Watts
Creator & Curator

Jason Gurik
Curator

Sara Smith
Graphic Designer

Matt Darby
Web Developer

Matt Casto
Sponsor Coordinator

CodeMash: A Small Giant

by Dianne Marsh

Small Giants are companies that choose to be great, rather than big. That doesn't mean that they can't be big, but rather that they will prioritize culture and quality over size. This is a natural way to describe CodeMash. It's a conference that chooses to be great. Judging from the demand from the developer community—selling out 1200+ tickets in 20 minutes this year—we could be a whole lot bigger. But, much like with the small companies that many of us own and operate, we're carefully growing to ensure that our growing pains don't ruin our culture or the quality that our attendees have come to expect. Quality speakers, friendly atmosphere, and relevant topics are hard to preserve with uncontrolled growth. In fact, we have grown every year, but we've been very deliberate about that growth.

We can look to Small Giants companies like Zingermans to provide some clarity in our thinking about growth. Zingermans is a Midwest success story, and one of the companies depicted in Bo Burlingham's book, Small Giants: Companies that Choose to be Great instead of Big. In spite of pressures to move the business and to franchise, Zingermans has chosen to remain in Ann Arbor and develop a community of businesses rather than franchising. The company itself has become quite big, but they're focused on the "why" of their existence. They want to provide great food and phenomenal service. CodeMash, too, is focused on its "why". We want to continue to provide a great conference experience in the Midwest, as we educate developers on current practices, methodologies, and technology trends in a variety of platforms and development languages.

This year, CodeMash will accommodate about 1200 attendees, in addition to staff and speakers. 1200. Recall that the first year of CodeMash, in 2006, we had 200. One of the things that people said that they liked about the conference was being able to talk to one another at mealtimes. Rather than spreading all of you out at meals into multiple rooms, we limited attendance in years 2-5 to accommodate all of us being able to eat in one room. We just kept pushing the tables closer together, and asking more of you to sit at a single table. Face it, developers aren't typically small-framed folks! But you squeezed in and it all worked out. Yes, our CodeMash 2011 mealtime was a little loud and that was hard for the people who wanted to hear the keynote, but what had we accomplished? We had DEVELOPERS .... NETWORKING. And not by wifi! We couldn't stop you guys from talking! And that's a good thing. But some of you were annoyed when we tried to make it quiet for those who did want to hear the speaker. This year, we've made some changes to accommodate talking at mealtimes. We're confident you'll let us know how it works out.

We've had some other growing pains too. Did you see us running around with the Kalahari folks at lunchtime last year, trying to get more food lines situated in different hallways? The long food lines were certainly a good thing to resolve at 900 attendees, rather than at 1200.

These are little things that we hope you DON'T notice. We don't want our growing paints to be obvious, or painful. You, our attendees, ARE our customers. And we don't take that lightly. We want to preserve the magic that you've come to expect. We want you to want to come back.

Admittedly, we have a problem to solve in how we manage registration for the 2013 event. That's a pretty cool problem to have, but it's a problem nonetheless. But the CodeMash board members (Jim Holmes, Jason Gilmore, and Brian Prince) are some of the most creative folks you will ever meet. They will come up with a solution. And that solution will be as equitable of a solution as possible.

CodeMash hasn't just grown in size; it's also evolved. The precompiler was added a few years ago, as an optional "day 0" so that folks could get in-depth training in hot technologies. But not all of the CodeMash changes have been driven by the Board. The CodeMash jam session evolved from a bunch of people doing what they loved to do. The Game Room and the Water Park Party were suggestions from attendees. The band was a sponsor-suggestion. This year, we introduce pecha kucha (lightning talks in a fast-paced style similar to Ignite).

You're an integral part of that mission. We hope that you will continue to attend CodeMash and that you will be generous with your feedback so that we can continue to provide an amazing experience. We will continue to grow CodeMash in interesting ways, and we're grateful for your suggestions and continued enthusiasm!

Crash KidzMash

By Melissa Insko

It seems almost counterintuitive that something could grow "organically" from a technology-centered conference, but that's exactly what has happened with the mini-conference KidzMash. If you haven't heard of KidzMash, it's probably because it's one of the best kept secrets of CodeMash. Geared toward those up-and-coming tech-nerdy types, KidzMash offers myriad activities to the families of convention-goers.

Foremost, KidzMash is a place where parents can bring their kids to learn a bit more about technology and have some fun while they're doing it. During the mini-conference, we block out several session times on both Thursday and Friday to present kid-friendly technical classes. In the past, these have ranged from online safety for our younger attendees, to Lego® Mindstorm, to parent remote control, to Kids' Programming Languages. It's tough to decide which gets the kids more actively engaged, hitting their elders in the ankles with robots, or sending them through crazy obstacle courses—controlled by nothing but the child's imagination!

KidzMash also offers CodeMash families a "home base" of sorts. Located in the Acacia room, we are here to help families connect during the conference. More than one person has commented that their family has made a life-long friend—not to mention pen-pals for the kiddos—through this program. The Acacia room is strictly dedicated to KidzMash activities, and remains open from sunup until bedtime. The room comes complete with board games, a karaoke machine, crayons and coloring books and the occasional voodoo doll. *Please note that KidzMash is not a daycare service. Children must be accompanied by one parent at all times, or they may end up in the stew pot.*

Beyond the core classes, KidzMash also offers several supervised fun activities to allow attendees to dry out after a long day at the water park. While at the Kalahari, KidzMash will host a dance party, Kinect® bowling, a movie night, a scavenger hunt, group pottery making, the opportunity to share meals with new friends, and even more. This year, KidzMash will also sponsor a poolside cabana from Thursday through Sunday. Although we take no responsibility for your belongings, this will provide attendees with a place to stash their stuff and to meet up with other families for some waterpark fun!

We welcome the families of all CodeMash attendees to stop by and check out KidzMash. Check out our Facebook Page too—it's where we make most announcements regarding schedule changes and the chiefs' proclamations.

History and Evolution of KidzMash

I blame the creation of KidzMash on the tropical environment at the Kalahari Resort! The mix of 3-foot snow drifts and palm trees seems to cause all sorts of strange things to happen.

The seed for KidzMash (formerly CodeMash Families) was initially planted at the very first CodeMash in 2006. My family was invited to attend—by way of my husband's career as a programmer—and I was blown away at the idea of a water park in January! Curiosity drove us to check it out. I figured it could either be the time of our lives or that they'd find our frozen bodies come Spring.

I cannot accurately describe what solidified my family's allegiance to this group of Hawaiian shirt clad tech-nerds. As newcomers, I was excited by how we were greeted as




friends, and ultimately impressed that my husband walked away re-inspired to get creative in his coding. He had a fantastic time. Unfortunately, there's only so much time one can spend at a water park, so by day two my daughter and I were wandering around alone and bored. At that point, I had no way of knowing if anyone around me was also a CodeMash widow. I found myself asking random strangers "Are you here for the conference? Is your [spouse] with CodeMash?" I looked (and felt) like a stalker.

Iteration 1—CodeMash Families

Because my husband was adamant that we'd attend the next CodeMash, I knew I had to do something. I was convinced there must have been other families wandering around the Kalahari alone, so I approached Jim Holmes and the rest is history. During the 2007 conference, we hosted a suite for "CodeMash Families" with the intention of introducing those of us outside of the conference to each other. It was initially intended as a social group, so we hosted play times, a movie and story time every evening. Around fifteen families attended based on nothing but a sign at the main registration desk.

Iteration 2—CodeMash Families Gives Life to KidzMash

During the 2008 conference, we christened the program with its second name—KidzMash—which would usher in computer-based classes for the offspring of those attending the main conference. During this time, we offered classes on Lego® robots (Duane Collicot) and building video games in Scratch (Arnulfo Wing). 2008 saw us grow from fifteen to 40 families, while the classes both grew and broadened in scope.

Iteration 3—KidzMash Expansion

In 2009, KidzMash added a third class to the mix as Andrea O.K. Wright hosted a class on Ruby string manipulation. Each of our participants used a laptop and jumped right in to programming under Andrea's direction. This year found us with 75 families registered and participating in the mini-conference.

2010 remained basically the same for KidzMash, although we did try to "grow" some of the classes at a rate to match our youngsters. We had over 100 families register for KidzMash this year.

2011 was, by far, the most extravagant and exciting conference that KidzMash has known so far. We added a second day to the classes, including Aydin Akcasu with a Parent Remote Control class, and Melissa Insko and Julie Fuller presenting internet safety classes for tweens and teens. We saw a 20% increase in attendance over the course of the mini-conference this year, hosting 120 families.

Iteration 4—KidzMash Now

KidzMash 2012 promises to be "bigger and badder" than ever before. Through November, we are entertaining proposals to expand our class schedule, and are also planning a multitude of events to keep families happy and occupied. Stay tuned to our Facebook Page for details on the first annual KidzMash Skee Ball championship, talent competition, and more!

Interview with the Koan Masters

By Daniel Hinojosa

CodeMash V2.0.1.2. will be featuring the *Scala koans*: *An Interactive Approach to Learning Scala* for the first half of the day on January 11, 2012 and will be presented by Dianne Marsh and myself. Scala koans offers small Zen-like lessons for learning a brand new language. It fosters questions and excitement and you can go at your own pace. Truth be told, CodeMash attendees go at a pretty fast pace. This year we will have more koans with intermediate and advanced topics than we did at CodeMash 2.0.1.1.

The purpose of this interview is to find out the History of the koan phenomena from their co-creators Jim Weircich and Joe O'Brien. I also interview Dianne Marsh and ask her more about Scala koans.

Daniel Hinojosa: Jim, the word *koan* isn't quite in the standard English vernacular. What is it? And how did you learn about this concept?

Jim Weirich: I first learned of koans from Douglas Hofstadter. He uses stories, dialogs and Zen koans as teaching aids in his book *Gödel*, *Escher* and *Bach*, often getting the student to think about computing, thought and reality in ways that were definitely outside the box. Koans are short, enigmatic stories that usually involve some kind of epiphany. That seemed to me to be the perfect choice for what we were trying to achieve.

"Koans are short, enigmatic stories that usually involve some kind of epiphany. That seemed to me to be the perfect choice for what we were trying to achieve".

-Jim Weirich

And of course, hacker culture is littered with koans and stories, particularly about the early days of the MIT AI lab. My favorite on is about Tom Knight (the inventor of the Lisp Machine):

A novice was trying to fix a broken Lisp machine by turning the power off and on. Knight, seeing what the student was doing, spoke sternly: "You cannot fix a machine by just power-cycling it with no understanding of what is going wrong."

Knight turned the machine off and on.

The machine worked.

Furthermore, the test framework we decided to use was heavily based on Ara Howard's Metakoans Ruby Quiz entry (http://rubyquiz.com/quiz67.html), so there was that word again. It seemed a straightforward decision.

Daniel Hinojosa: What about yourself, Joe?

Joe O' Brien: The Ruby Koans themselves has a bit of a back story. Jim and I were pairing one day when I made the observation that I wish I could have simply recorded our session that day and played it back as a conference talk. See Jim is one of the best teachers I have ever met. I mean teacher [in] the traditional sense of ensuring you learn and understand a subject, not someone who can simply get in front of a room. He's always had this way about him where he will identify teachable moments and let you discover information yourself. A small example of this was my first question to him about ruby through email. I could not understand why a unit test was failing. I sent him the test and asked him if he could help me. Instead of telling me the answer, Jim sent back three unit tests.
This process of self discovery is what koans are all about and is standard part of the Zen philosophy. The Socratic method of teaching talks a lot about this. If you discover an answer yourself, you will retain it. It's about learning information, not having it pushed to you.

When CodeMash asked us to do a Ruby introduction class three years ago, Jim and I sat in a room and talked a great deal about it. I've been wanting to do test-driven teaching of a language for a lot of years now. Mike Clark wrote a blog entry in March of 2005 that talked about how he had learned Ruby through unit tests simply because the barrier of entry to writing a test in Ruby was trivial and it played into how his mind works. Mike Clark and his wife Nicole now head the Pragmatic Studio, a training company dedicated to providing intense hands-on learning of many subjects from Ruby to iOS to Clojure.

This blog entry sparked a desire in me to create a class that walked students through writing tests that taught them the specifics of Ruby. This would have the added side effect of notes the students could always refer back to if they needed to reference them. It also shows the culture of Ruby along with teaching the technology. The ruby community treats many practices not as 'ways to be better at your work' but as ways of doing things. Unit testing, for instance, is not an optional practice in the Ruby community.

Combining the koans idea of self-discovery with this desire for a test-driven approach to teaching helped drive this idea forward. The last major component was writing them.

The first idea came up when it occurred to me that you could define a method for __ (a character combination not allowed in many method definitions. It reminded me of those worksheets we all filled in back in school:

"The _____ president of the United States was the only president to ..."

Then the testing framework was wrapped to give our own output and only show information for the current test you were on. Staring at a daunting list of failures isn't a great psychological move in terms of motivation. All of that was the 'easy' part.

The real trick to the koans was a painstaking process (which was all Jim) of deciding what to introduce in what order. We wanted to let people write code as quickly as possible. We all know though that you need to know _ *some*_ things in order to write your first method. So the goal was to get to something like the triangle exercise quickly in order to give everyone a feeling of accomplishment. At the same time, we have to be careful not to overwhelm new users with really hard topics and demotivate them to learn. This is something I really wish other koans implementations would take more seriously.

Dianne Marsh: I agree with Joe's assessment that the key is to keep people motivated to learn. Joe pointed out that the order of the exercises is critical (and difficult!). Another challenge we've felt with the Scala Koans is that it's "easy" to come up with the very simple fill-in-the-blank exercises, while filtering in more interesting examples is tricky. I think that the Ruby Koans have managed a very nice balance. I see that there are now Javascript Koans, Coffeescript Koans, Clojure Koans, F# Koans. I suspect there are more.

Daniel Hinojosa: Who were the first people who tried Ruby Koans? What was the initial response? Was there anything that you fixed after the first session?

Joe O'Brien: We tried them at CodeMash three (or was it four?) years ago to a very excited crowd. We were quite amazed at just how well it worked. I remember there being small issues here and there, but not anything I can think of. Jim did an amazing job on the details of it and we've never had to do any major adjustments (other than getting up to [Ruby] 1.9 later).

We were also asked to give them a year later at a Ruby Conference. We thought it was kind of strange. So we flew out to LA to a packed room (about 40 or 50 people). We still didn't believe they had understood what we had intended so we asked them "who here has done Ruby before" and only a few hands went up. Kind of crazy. We asked and it turns out they loved the idea of having a tutorial on it the first day that would allow them to get more out of the rest of the conference.

It wasn't until about a year later that we launched rubykoans.com and felt they could stand on their own. We have been utterly thrilled with the response and the traffic. Over the last year, Marc Peabody has been working very hard on really putting them online. Recently he launched the online koans allowing you to run through them right there in your browser (and no, we are not faking it, it really does run on the server and come back).

Amazingly enough, we have not changed much since they were first launch[ed].

Dianne Marsh: Joe, I think the first Ruby koans were in 2009 because I think I missed the first year. I went back and looked at the CodeMash schedule and I don't even think you called it "koans" in 2009 although I think you used them. The precompiler was called "Ruby 101".

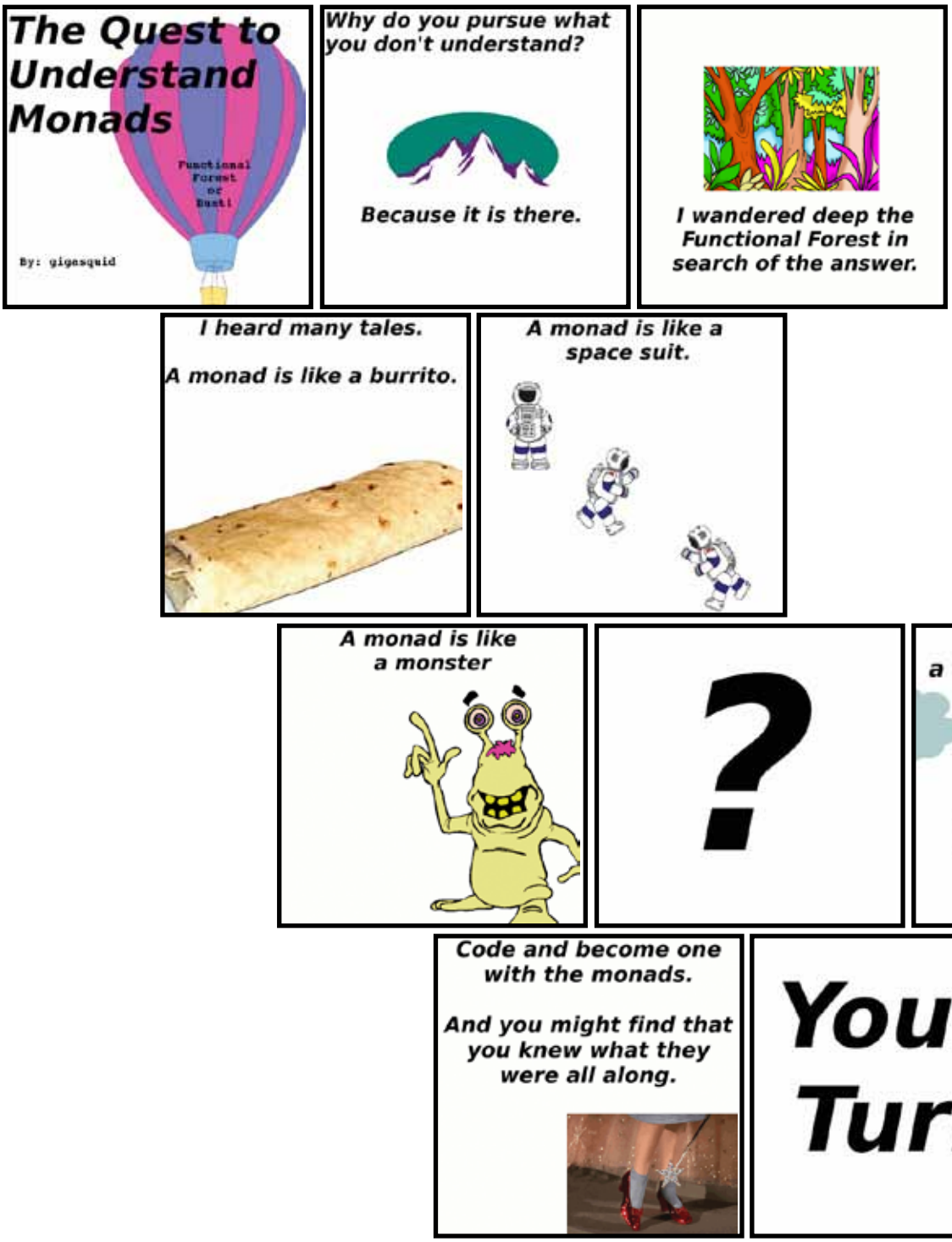
Dan Hinojosa: Dianne, how did Scala Koans get started?

Dianne Marsh: At CodeMash 2010, Dick Wall, Chris Marinos, and Amanda Laucher took up residence in a room at CodeMash to write koans. Dick focused on Scala, while Chris and Amanda worked on F#. The "functionalkoans" project was born, and for a short while embodied all of these koans. They have since been split off since it didn't make sense to manage them as one repository tree.

Dan Hinojosa: And, what's new with Scala Koans?

Dianne Marsh: I'm in the process of bringing up a scalakoans.org website because it's so much more friendly than GitHub or BitBucket for information. I draw all of my inspiration from Jim and Joe, who introduced me to the koans as a Ruby Koans student at CodeMash in 2010.

For more information on Ruby koans, visit http://rubykoans.com/. For information on Scala koans http://scalakoans.org/. For information on Scala koans Precompiler visit http://www.CodeMash.org. Thank you to Jim Weirich, Joe O'Brien, and Dianne Marsh for participating in this interview, and I look forward to seeing all of you again at the most awesome CodeMash.



Oh, the Methods You'll Compose

By Tim Berglund

http://www.youtube.com/watch?v=3QbheQitF74

When a coder sits down to start banging out code
The first thing to start crowding his cognitive load
Is whether his program will do what it should
Correctness, he says, is what makes my code good

It's the function that captures the coder's attention
Behaviors and inputs and outputs are mentioned
As if the one good that a coder can bring
Is to spin the right wheels on some Turing machine

And compiling and linking and running are great
(We need to do these to put food on our plate!)
But the shocker that might leave you scratching your head
Is that actual code is less written than read

We spend more of our time in maintaining our stuff
Than we ever spend writing the simplest of cruft
Which means that unless you've got something the matter
You'll try to learn just a few code style patterns

So coders and countrymen, lend me your ears
As I teach you some lessons won hard through the years
From that Beckian book about implementation
And patterns that derail code suckification

A classical problem is how to name things
(Oh, the anger and fights and dissension this brings!)
Like off-by-one-errors and cache expiration
A permanent answer's beyond expectation

But a class should be named to describe its intent
Not its implementation, though that's how we're bent
A superclass name should be pithy and short
And the subclass's name a more detailed retort

When you look at the name of a class you should find
The idea that hatched in the first coder's mind
And just what is the thing this class wanted to do?
And what should you be thinking when first you call "new?"

When you can't find a name for a class, it's a sign
That the metaphor's actually escaping your mind
A good metaphor helps more than comments or training
To inform other coders just what you are saying

The next thing we'll consider together is state
(Which wouldn't be bad if it wouldn't mutate)
The functional people may think that they profit
But objects we code will change state; we can't stop it

It's not just concurrency where it can bite us
Although many suffer from thread-lock-wait-itis
The way that we organize pieces of state
Can make all the difference between good and great

Group similar state close together and see
Just what happens in time to your code quality
If you think of the reason your data is altered
Your sense of the meaning will be less assaulted

The things that are changing together should be
Very close to each other, viewed all on one screen
The data whose purpose is common? Same thing.
If they all work together, keep'em all in one scene

Remember the scopes an imperative language
Gives to you to gather together your baggage
The method, the instance, the class scope are able
To keep you from having too much on the table

The changing of state is a serious problem
And I think that we're starting to locate the bottom
But as long as assignment is part of our ken
We've got try hard to keep data reigned in

See, the coder who's reading this pile of junk
Is bounded in what he can think of at once
Don't make him scroll all up and down every file
To find and recall every identifier

Now the methods we write can get out of control
When we make the one reading them scroll, scroll, scroll, scroll
A method that's long and meandering is bad
But a method that's short and composed makes us glad

To compose a good method, just think of the scope
Of the things that it does, and with that you can hope
To keep all of its actions around the same level
Or else its abstractions will leave you disheveled

A method composed by a coder who tries
Will read like a story with just one plot line
Each part of the story it tells is the same
As the rest of the method, with all the same aim

To understand just what composing's about
Imagine a story of when you went out
And started the night by first changing your clothes
Then switching to tell me the distance you drove

Then up and describing the way you shift gears
And a long, pointless tale about your rear-view mirror
Then changing your story to cover the dinner
And how it was cooked and how long the sauce simmered

Then skipping ahead to the movie you saw
But not telling me even one detail at all
Now what would you think of this crazy approach?
My mental disturbance would be hard to broach!

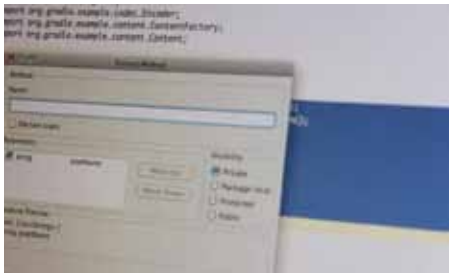
It's the very same thing when we factor a method
Each one should stay small with its purpose embedded
In a series of readable sub-method calls
And inside those methods go all their details

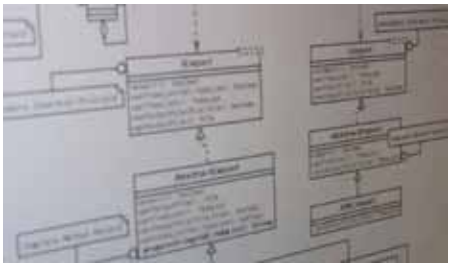
When people are learning, they sometimes prefer
First to know all the details and from them infer
All the concepts producing the detailed design
Either concept or detail can govern the mind

When composing your methods, keep this fact in mind
And please think of the coder who's struggling to find
The whole shape of the picture zoomed all the way out
Or the flipping of bits when that's what it's about

I think if you've listened a little to me
You might start to catch what I want you to see
It's good when we write code that passes its tests
But mere functionality isn't our best

It's the human who sits down to work with our code
Our ideas we want to this one to be showed
So remember this saying, to the best that you're able
The reason you write code is to love your neighbor





A Brief Forage into Functional Thinking

By Seth Juarez

Introduction

I have long considered software engineering to be a craft that must be honed over years of careful study and precise implementation. What we learn in the labs of school and contrived examples are indeed important but only when carefully used in our day to day work. As such my challenge to you, dear reader, is to try to find at least one application where thinking functionally might assist you in your daily endeavors to produce excellent software.

Years ago I took a graduate class in functional programming. The professor allowed each student to implement a series of exercises in a functional language of their choice. At first my imperative background clashed a great deal with this new (at least to me) approach to writing software. Finally the thought occurred to me: "think of functions as data types." Now while this revelation is indeed a simple one, it completely altered my approach to solving problems in software. Here we will look at the evolution of functions as data types in C# and look at a series of practical applications that produce more readable and concise code.

Delegates

There have been many approaches to describing delegates. I think the best and simplest is as follows:

A delegate declaration simply defines a new data type that is inherently a deferred unit of work.

Consider the following line of code:

```
int x = 3;
```

There are three basic parts to this statement: the data type, the variable name, and the assignment. While the following code is a bit more complex, the same basic elements also apply.

```
delegate int doWork(int x, int y);

static int add(int x, int y)
{
   return x + y;
}

static void Main(string[] args)
{
   doWork work = add;
}
```

The penultimate line is the one that is most similar to them simple integer declaration. We have:

1. the Data Type is *doWork*,
2. the variable name is *work*, and
3. the assigned value is called *add*.

Notice that while the first and third elements are a bit different, the basic concepts are still very much the same.

Functions, at their core, have a basic signature that includes the return type, the parameters, and a name. Delegates are just a definition of a new function type. The challenge I had early on was thinking of functions as nouns rather than verbs. Using intellisense one can readily see that the variable *work* from above has a series of properties and methods (one of which includes executing the function). Just like objects, delegates can even be passed around as parameters to other functions. Above we see that we have declared a new delegate that returns and integer, takes two integers as parameters, and is called *doWork*.

The function (or method) *add* from above matches the delegate *doWork* in both the return type as well as in the required parameters. As such, creating a new variable called work and assigning it the value *add* is perfectly legal. Change any two aspects in the function *add* and the compiler is sure to complain. Now that we have this object called *work* we can treat it just like any other object.

Lambda Expressions

Lambda expressions are simply syntactic sugar over the same ideas from above. Below we see the exact functionality from above using a lambda expression:

```
static void Main(string[] args)
{
   Func<int, int, int> work = (x, y) => x + y;
}
```

Instead of calling the delegate *doWork*, we use the built in generic delegate called *Func*. The basic premise behind this built in delegate is that parameters are the first set of types used in the generic declaration and the last is the

return type. A similar delegate called *Action* is used when the function has no return type. The assignment portion has the actual lambda expression. The first portion provides names to the parameters used in the function. The arrow delineates the beginning of the actual function and the final section specifies the type of work performed. Notice that in this case there is no need to use the keyword *return* as this is implicit in the delegate itself. In the case of multi-line lambda expressions one need only use curly braces to begin and end the function declaration and specify the return value at the end.

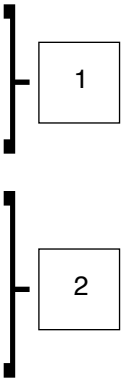
An Example

I am currently working on an open source machine learning library. In it there is a considerable amount of linear algebra used to implement many of these algorithms. Oftentimes this requires the ability to slice a matrix based upon certain properties of the row or column vectors. Standard implementation would simply dictate one create a number of for loops to evaluate the appropriate criteria and then reassemble the correct matrix. Consider a more functional approach:

```
public Matrix this[Func<Vector, bool> f, VectorType t]
{
  get
  {
    int count = 0;
    if (t == VectorType.Row)
    {
      for (int i = 0; i < Rows; i++)
        if (f(this[i, t]))
          count++;

      Matrix m = new Matrix(count, Cols);
      int j = -1;
      for (int i = 0; i < Rows; i++)
        if (f(this[i, t]))
          m[++j, t] = this[i, t];

      return m;
    }
    else ...
  }
}
```



The usefulness of this approach is not readily apparent until it is actually used:

```
Matrix m = new[,]
  {{ 1, 2, 3},
   { 4, 5, 6},
   { 7, 8, 9},
   { 4, 9, 1}};
```

```
Matrix col1 = new[,]
        {{ 1, 3},
         { 4, 6},
         { 7, 9},
         { 4, 1}};

Assert.AreEqual(col1, m[v => v[0] != 2, VectorType.Column]);
```

Let's first talk about the usage and then move on to the implementation. The example shows a matrix m and a matrix called col1. The last line simply instructs the matrix to construct a new matrix whose columns (second parameter) do not have their first member equal to 2 (first parameter). The beauty of doing it this way is that one can construct any number of new matrices by defining a deferred unit of execution that can be subsequently passed in to the matrix. The imperative alternative is to create a set of nested for loops every time a slice operation is required.

The implementation is quite simple and can be divided into two parts:

1. figure out the new size of the matrix, and
2. build the new matrix.

Those two parts are clearly labeled in the implementation above. The delegate we are passing in takes a vector and returns a boolean. In essence, its basic job is to tell us which elements to keep. The second parameter tells us whether we are working in the column or row space of the matrix. This minor change has allowed users of the machine learning library to more concisely work with matrices and vectors.

Conclusion

Now while the things we discussed are not earth shattering, keeping this main concept in mind can definitely change the way you have implemented any number of algorithms to a more concise and compact representation that perhaps you have heretofore never considered. I would love to see how! Drop me a line if you have found a piece of existing code that benefits from this slight shift in thinking.

Say What You Mean!

By Matt Snyder
snyder2112@me.com
@msnyder
http://www.cranialpulse.com

5 years ago I wrote my first line of Ruby. It was at a coding dojo while pairing with Corey Haines. At the time I was deep into the world of .NET and C# and had not yet seen a line of Ruby code. So when it came time for me to take the keyboard and make something happen I didn't have a clue where to start. That's when Corey said, "Just write it how you think it should be written". So I did. And what I had written, while not exact Ruby syntax, was pretty close!

"But Corey, what about testing in Ruby?" Corey then introduced me to the most important Gem in all of Ruby, RSpec. I was already an avid TDDer in C# using NUnit, so testing code was very familiar to me. However, NUnit always seemed lacking. No matter how expressive I tried to make the tests they never seemed to convey thoughts as clearly as I'd like. RSpec was not like that at all! Within minutes of pairing with Corey writing specifications in RSpec, I realized I had found my soul mate in RSpec! It was awesome!! I was able to Say What I Meant in code without any ceremony.

Yukihiro Matsumoto said "Code is an expression of the thoughts, attitudes, and ideas of the programmer." When it comes to expression of thought, Ruby is definitely an enabler!

Regardless of what language you are using to build your next application for world domination, pay attention to how you convey your thoughts in your code. Because without attention to how you organize your code, it can be very easy to get lost in the code and take a wrong turn. And the next thing you know you are pointing your Mega Laser at the wrong place during runtime!! (Hopefully your tests covered pointing the laser and saved your arse)

Avdi Grimm takes expression to the next level, emphasizing that developers should confidently express themselves in their code. The first step in building confidence and clarity in your code is to establish a consistent narrative structure. Avdi's proposed narrative structure is this:

1. Gather Input
2. Perform Work
3. Deliver Results
4. Handle Failure

Now that you are armed with this narrative approach to writing your code, don't forget to just simply Say What you Mean! Your co-workers will thank you in the end.

Why You Should be Using Aspect-Oriented Programming

By Matthew D. Groves
webmaster@mgroves.com
@mgroves
http://mgroves.com

The single responsibility principle states that every object should have a single responsibility, and that responsibility should be entirely encapsulated by the class. It should do one thing, it should do it well, and it should do it *only*. However, in any real production application, secondary concerns are often scattered throughout otherwise single responsibility code: logging, threading, caching, exception handling, just to name a few.

Consider a requirement that every function call is required to log the function name, date/time called, return value, arguments passed in, etc. This may sound absurd, but there may be a number of good reasons–business or technical–for this requirement. Consider a naive approach to this feature:

```
public class MyClass
{
    public string MyFunction(int param1, string param2)
    {
        Log.WriteLine("MyFunction: " + DateTime.Now);
        Log.WriteLine("param1 (int) argument: " + param1);
        Log.WriteLine("param2 (string) argument: " + param2);

        // perform primary function
        var returnValue = "some return value";
```

```
        Log.WriteLine("Return value: " + returnValue);
        return returnValue;
    }
}
```

This approach solves the problem, but the flaws should be obvious: production applications contain hundreds or thousands of methods, and adding all this logging code would take a considerable effort. Consider another slightly more sane approach that uses reflection:

```
    public string MyFunction(int param1, string param2)
{
    Log.WriteMethodInformation(MethodBase.GetCurrentMethod());

    // perform primary function
    var returnValue = "some return value";

    return returnValue;
}
```

Now we can use the MethodBase object to dynamically log the method name and the parameter names/types without having to hand code that information. But this approach isn't much better than the naive one. We still have to copy and paste the call to Log.WriteMethodInformation into every function. If that method's signature ever changes, we have to again update every single function. Also note that this approach doesn't completely fulfill the requirement, as there's no way to get the arguments or return values using reflection, so we'll still have to go back and hand code that logging.

Now let's try using AOP. Here's an example with PostSharp, a .NET AOP tool.

```
[Serializable]
public class LoggingAspect : MethodInterceptionAspect
{
    public override void OnInvoke(MethodInterceptionArgs args)
    {
        var methodName = args.Method.Name;
        var parameters = args.Method.GetParameters();
        Log.WriteLine(string.Format("{0} timestamp: {1}", methodName,
DateTime.Now));
        for (var i = 0; i < args.Arguments.Count;i++)
        {
            Log.WriteLine(string.Format("{0} argument #{1}, {2} ({3}): {4}",
methodName, i, parameters[i].Name, parameters[i].ParameterType, args.
Arguments[i]));
        }
        args.Proceed();
        Log.WriteLine(string.Format("{0} return value: {1}", methodName, args.
ReturnValue));
    }
}
```

To apply this aspect, PostSharp uses .NET Attributes (for instance, decorating a method with [LoggingAspect] would do the trick) to signal to a post-compiler where to weave the aspect. Attributes can be multicast out to every method to avoid repetition.

```
[assembly:LoggingAspect(AttributeTargetTypes = "SomeNamespace.*")]
```

Other AOP frameworks like Spring.NET would use a very similar Method Interception implementation, which can be added via a runtime proxy through dependency injection. Now notice that the code for performing the logging lives in one completely encapsulated class, and the code being logged can be changed at will without affecting the completeness of the information being logged. When the main code changes, the logging code doesn't need to change, and when the logging code changes, the main code doesn't need to change. Each class has one and only one reason to change. Each class does one thing, does one thing well, and does one thing only.

What to Build and When?

By Peter Bell

Probably the biggest waste in software development is building features that nobody uses. Here are some questions you can ask to increase the likelihood that the next feature you code will actually end up being used and adding value.

What's the business intent? What is in it for your organization? If the software was to be built, delivered, and used, why would that be a good thing? Would it be worth the effort expended? Is it worth the opportunity cost of the software you could have otherwise built in that amount of time?

What is the definition of success? If the software is successful, how will you know? What metric (qualitative or quantitative) will improve? Are you set up to measure that metric? Will you be able to isolate the impact of the feature by split testing the functionality provided to a subset of users and tracking them using cohort analytics?

Why would anyone use it? The "As a <role> I want <feature> so that <benefit>" can become too repetitive to write for all of your stories, but you should always be able to state any story in that format. If you can't even envision in your most optimistic moment why a user would want to use the feature (what the benefit is), they probably won't use it.

Could it be simpler? Could you provide the same benefit by implementing a simpler feature? For example, do you really need to implement semantic analysis of text to provide recommended articles or could you just recommend them based on the known behavior of other similar users?

As hard as building the right thing is prioritizing which things to build first. One of the better approaches I've found for doing this is "metric driven development". At any given time there is usually one metric that can be prioritized for an application. Example metrics could be increasing the number of registrations, improving conversions from free to paid plans, improving cross-sell effectiveness on a commerce site, increasing engagement or improving the viral coefficient (the number of referrals per user in a specified period of time). One interesting approach is to pick a metric for each sprint and for that sprint to prioritize tickets based on how likely you believe they are to improve the metric. This often makes it much easier to prioritize the backlog when there are lots of competing demands.

Let's Get Chocolatey - Kind of like apt-get for Windows

By Rob Reynolds
ferventcoder@gmail.com
@ferventcoder
http://ferventcoder.com

Let's Get Chocolatey - Kind of like apt-get for Windows "If only there was some way to quickly and silently install applications and tools on my windows machine." Chocolatey (http://chocolatey.org) is kind of like an apt-get, but for Windows. It is a machine level package manager that is built on top of NuGet command line and the NuGet infrastructure. Jason Jarrett recently described it as the free/OSS windows app store. What that means for you is that you can install and update software (applications and tools) on your machine with a few keystrokes and Chocolatey does the rest!

Just how easy is it to install an application? From the command line, PowerShell, or Package Manager Console in visual studio you can type something like: cinst windirstat and watch it download and silently install WinDirStat on your machine. A picture is worth a thousand words in this case:

If you want to try a tool (something that doesn't actually install on your machine), try baretail, nodejs, or ravendb. Creating Packages is also very simple: https://github.com/chocolatey/chocolatey/wiki/CreatePackages Below is all that is required to install WinDirStat on your machine. Install-ChocolateyPackage 'windirstat' 'exe' '/S' 'http://

windirstat.info/wds_current_setup.exe'. Include Chocolatey in your development environment setup! This means a one line command to get a collaborator's environment all set up for working on your source code (tools to apps installed if they are missing): https://github.com/chocolatey/chocolatey/wiki/DevelopmentEnvironmentSetup. Check out a living example - https://github.com/davidalpert/nuserve#readme

FAQ

What can I do with Chocolatey? - Since it uses PowerShell, you can do nearly anything you can do with .NET. Install applications, download tools and put them on the path, set up contributors machines for hacking on your code, install Powershell commands, etc. Your imagination is the limit!

What's your best example of the power of Chocolatey? - One line Ruby DevKit install. Seriously. http://groups.google.com/group/rubyinstaller/browse_thread/thread/8245c53f990d1ea6 I'm convinced!

How do I install Chocolatey? - We try to make that simple as well. Open Powershell, make sure execution policy is unrestricted (Set-ExecutionPolicy Unrestricted), and paste iex ((new-object net.webclient).DownloadString("http://bit.ly/psChocInstall")).

I have included tools (executables) in my nuget.org packages, like Statlight and Fubu. Can I use Chocolatey to "install" them? - Yes, just call install like normal, it will check chocolatey.org first and then nuget.org. If it finds an executable in the package, it will automatically put it on the path.

How is Chocolatey different from other Windows machine package managers? - It has PowerShell instructions for how to download native installers from the distribution source and install applications on your machine. It uses PowerShell so you can give it any instruction you want for install and configuration. It automatically makes batch command file links for executables you have included in your package or have downloaded to the package directory with the PowerShell script.

I'm not convinced, where do I find more information? - See below References.

- http://chocolatey.org/, https://github.com/chocolatey/chocolatey/wiki
- http://groups.google.com/group/chocolatey, http://twitter.com/chocolateynuget
- Chocolatey In Action (11 apps/tools in less than 7 minutes!): http://www.youtube.com/watch?v=N-hWOUL8roU
- Create a Chocolatey Package: http://www.youtube.com/watch?v=Wt_unjS_SUo

Is Your Phone Alive?

By Samidip Basu
samidip@live.com
@samidip
http://samidipbasu.com

Ahoy CodeMashers! First, I am a big supporter of the Windows Phone platform and its development ecosystem. So, this article will come across as fanboyish .. you have been warned :) However, competing mobile platforms are equally good and we shall all coexist for a very long time. So, let's make peace, shall we?

Now, why do you as a developer begin developing for a new ecosystem, above and beyond your day job? Yes, we like staying on top of what's new in technology, but more importantly, we really need to like the product we are developing for to have fun, right? Thankfully, that's where Windows Phone steps in for me. The Windows Phone platform launched November 2010 and it is really nice to see the product become on par feature-wise with other dominant mobile platforms in less than a year.

As you know, the major 2011 update is codenamed "Mango" and it should have made its way to all new and existing Windows Phones by the time you are reading this. While tech products are subject to one's choice, Windows Phone Mango has received overwhelming positive reviews and feels like a fresh well-thought-out Mobile OS. There is whole lot of love for developers to help our Apps shine with about 1500 new APIs. However, I shall not talk about developing for Windows Phone Mango, as you would have gotten a heavy dose of that from the CodeMash sessions. This is about not getting lost in API details as a developer, about stretching your legs and just enjoying a tech product. Yes, there have been a thousand reviews of Mango from a consumer standpoint, but here are thoughts from a fellow geek on five Windows Phone ecosystem features that should provide something for the techiest of users:

1. Text to Speech: Surely you don't text and drive, right? Well, Windows Phone Mango tries to help! With just a couple of settings, the phone will announce that you have received the all important text from the wife and then proceeds to read it aloud if allowed. Next, it goes on to ask if you want to respond to the SMS and if so, tries to convert your speech to text, doing a fairly good job unless you speak jibberish. All this completely hands-free! Though not revolutionary, pretty amazing to see this working out over a long SMS conversation over a car's Bluetooth. Now, if you car comes equipped with smartphone MMI integration through a micro-USB port/BT, you're in for even better integration. Now you can simply throw your phone into the glovebox and go completely hands-free with SMS and also have your whole Zune collection accessible! There is, of course, potential to make things better. Somewhere I heard rumors of Sync® framework in Ford to go all Metro; but I possibly speak too much. Surely now you don't want your emails read aloud, or do you?

2. Social Integration and Grouping: Like others, Windows Phone Mango now has much of your Social integration built-in, maybe in the best way. Here's what works: you can have your hooks into many social networks, but you see all aggregated updates in your ME Live tile and also get to post to any social network you wish. It lacks some of the finesse of dedicated apps, but the grouping and Live Tile features are invaluable to quickly check if you need to respond to something. One could easily create a group of people from disparate social networks and see all their updates in one place. Maybe in the future we could simply pin a Twitter list instead of having to create new groups.

3. Flipping Live Tiles: The enhanced Live Tiles combined with Push Notifications are among the best Windows Phone Mango features for me. Sure, we had Live Tiles before; but now they can be flipped with animation to expose a Tile backside, thus conveying double the information. As a developer, secondary Live Tiles supported by Push Notifications are fairly easy to implement and give your app an immediate edge. The home screen experience on the phone simply comes alive with all these flipping Live Tiles. You choose as a user, each conveying key information in a fun way! Combine this with the new Fast Application Switching and you can totally dance around between the Live Tiles and their supporting apps and games.

4. Bing and App Connect: The Bing Search experience gets even better. Things like Bing Scout (which has local suggestions), Bing Music (read Shazzam) and Bing Vision (for QR codes/tags etc.) are built-in, alongside voice-controlled searches, so you are always one click away from the stuff you usually want to look up. Another little enhancement is Search Extensibility or App Connect. If you search for things like Products/Movies/Locations, Bing gives you little Quick Cards on top of search results. These help in bringing context-sensitive information from multiple sources right on the Quick Card. One swipe of the pivot and you see apps! Yes, these are Marektplace apps that you either have installed or Bing is suggesting that you install, because each of these apps have signed up with Bing to declare that they can do something with your Search results. So, you search for a product in Bing, and Search Extensibility puts the product (with associated information) in a Shopping List app you have installed. This leads to rich end-user Search experience. App Connect is a unique way for developers to showcase and market their Apps to users, simply by tapping into Bing results. A must-do in my opinion.

5. Web Marketplace: Ok, this is clearly not a Windows Phone Mango feature, but it was released with Mango and is simply a delight to use. So, head over to http://windowsphone.com and you'll see a new Marketplace tab. Yes, it is the full Windows Phone App Marketplace that you see through your phone or Zune. You can browse for Apps and Games just as you expect. However, if you sign in with your Live ID, voila! Now the Marketplace knows you and your phone. It keeps track of apps/games that you had installed in the past, in case you ever need to get them back. And the best part is, you can browse for apps and simply hit install! The apps show up auto-magically on the phone that you may have lying around somewhere else. Simple yet cool, right?

To end things off, let me say what has been repeated many times in Windows Phone Mango reviews—the phone just feels fluid and natural to use. It just feels alive with action! Of course, every platform has shortcomings, this being no exception. But I didn't have any incentive to write up all this, except for purely loving the product. So, trust a fellow geek and head to your carrier's store to try a Windows Phone. They will try their level best to not sell you the phone—it's an issue we are quite aware of. But give the phone a spin anyways, you might just like what Windows Phone Mango has in store for you. Adios and Happy Geeking.

Silly Rabbit, Parsing HTML is for Kids!

By Tian Valdemar Davis
tiandavis@gmail.com
@tiandavis
http://techoctave.com/c7

If I had a nickel for every time a developer said, “Don’t worry, I’ll just whip up a RegEx and parse that HTML in no time.” I’d retire this very second. I’ve grown less and less shocked over the years. To be honest, I’ve been there, done that. That’s why I know it’s generally a bad idea. Given, there are times when you have to suck it up and use RegEx. But experience has shown me that this is rarely one of those times.

Talking a developer out of using RegEx to parse HTML is like talking a good friend off the ledge. Curiously, no one ever says, “Hey, let’s use RegEx to parse XML!” So what’s the deal with HTML? Is it the familiarity? Hell, even Jon Skeet can’t parse HTML using RegEx and the last guy to try went bat shit crazy. See, Jeff Atwood’s Parsing Html The Cthulhu Way. Time is too valuable. Spend what little time you have implementing your core logic. This is not a fight you need to pursue. Just say no!

Where do I begin? What can I say that hasn’t already been said (see StackOverflow Question 1732348)? Honestly, I don’t know where to begin. The truth is there are hundreds of scenarios that you did not, could not and will never think about. RegEx will set your project in a never ending cycle of break-fix because chances are there will always be HTML to break your little parser. Technical Debt is not something to take on lightly. It’s not something to take on at all. But for God’s sake, don’t take it on to parse HTML.

You don’t need to roll-your-own parser. The pragmatic solution is simple—use an HTML parser library. Ideally, one with DOM search support via XPath or CSS3 selectors. I understand this is easier said than done. But the benefits will far outweigh the thrill of slinging on your mouse and riding into the sunset with your trusty Aeron deluxe. I’ll do you one better. Instead of just saying, “Grab an HTML Parser”, I’ll point out your best options:

Ruby: Nokogiri
JavaScript: jQuery
PHP: PHP5 DOMDocument
.Net(C#): HTML Agility Pack
VB6: MSHTML (IE’s DOM Parser)
Python: lxml
Perl: HTML::Parser
Java: HTML Cleaner

One of these full-fledged HTML parsers will do the trick. HTML parsers aren’t some new bag of tricks. Don’t be surprised to see jQuery-like syntax for some of the more syntactically pleasing libraries.

You don’t have to be a hero. I’m pretty sure you have a task to accomplish. I’m equally sure that task isn’t to write an HTML parser. So don’t! You’ll thank yourself in a couple weeks when the Red Bull wears off. It’s really not worth it! Granted we all have to start somewhere. Think of this as your new beginning.

CodeMash Game Room: Fun Nights and Networking Too!

By Jeff Blankenburg,
Steve Smith and Michelle Smith

The CodeMash Game Room is designed to fill that time between when the sessions end and midnight (you should sleep a couple of hours each night, right?). In 2011, we had nearly 100 gamers each night in the Game Room, playing games ranging from Settlers of Catan to a 24-person Texas Hold’em Tournament. Some games draw a larger crowd, but if it’s a game, there’s likely someone there willing to play it. One of the best parts of the CodeMash Game Room is the networking opportunity it presents. You end up sitting at a table with a bunch of like-minded software developers (you’re all at CodeMash, aren’t you?) and get to spend some time playing games and getting to know one another. This room has definitely become a popular destination in the evenings of CodeMash, so much so that it fills two entire rooms now and has a dedicated bartender too! So bring a game of your own or wander in to join any game that looks fun.

A big thanks to Nimble Pros for sponsoring the room with snacks and pop each year!

Popular games from past CodeMash game rooms

Ticket To Ride
Killer Bunnies
Fluxx
Munchkin
Settlers of Catan
Are you a Werewolf?
Dominion

Kalahari Expansion == More CodeMash

By Nick Watts

If you've attended CodeMash in the past, you might have noticed that the burgeoning conference has started to fill the Kalahari Resort—Sandusky convention center. That won't be true again for many years to come thanks to a $22 Million, 120,000 square-foot addition. The Kalahari Resort—Sandusky convention center, now an impressive total of 215,000 square-feet, has been expanded to include a "38,000-square-foot grand ballroom / expo center, a 12,000-square-foot junior ballroom, 14 meeting rooms, a 4,000-square-foot themed outdoor plaza connected to an indoor ballroom and 30,000 square feet of pre-function space" according to a January, 2011 press release.

The expansive addition to the convention center is due to the growing needs of events such as CodeMash. Maris Brenner, the Kalahari Resorts—Sandusky Director of Sales, states that "Although groups want to bring larger statewide or national conventions to us, in the past we […] couldn't accommodate their functional space needs". That's been rectified now and Brenner notes their readiness to "draw significantly more attendees" to the updated convention center. The direct result for CodeMash is the increase in total attendees from 800 in 2011 to 1,400 this year.

Although there's much more room to grow, CodeMash president Jim Holmes cautions that further conference growth will be conservative. Jim had the following to say on the topic:

> *While we now have a space that will support over 5,000 CodeMash geeks and geekettes, we've always been careful about growing the conference at a sustainable pace. Every decision we make about the conference, be it trying to record sessions, add new features, or increase our size, falls back to one fundamental question: how will this decision help us keep delivering kickass sessions to our attendees?*
>
> *Growing to a size of thousands sounds cool, but we won't go there except by small steps—and we may not go there at all. There's nothing wrong with wanting to remain a smaller(ish) conference that focuses on ensuring attendees have a great time, learn fascinating things, and come away re-energized to go do amazing things.*
>
> *Besides, at the end of the day we're still figuring it out as we go!*

The Kalahari's Grand Ballroom

To keep up with news about CodeMash 2013 and beyond, join the CodeMash Google Group at http://groups.google.com/group/codemash or follow @Codemash on Twitter.

Getting into Mobile: App or Website?

By Amelia Marschall

Choosing between developing a mobile website or mobile application depends on a variety of factors: budget, purpose, and target market to name a few. Much money and resources can be wasted creating a mobile app that isn't well-planned. If your organization is itching to jump on the mobile bandwagon, I often advocate that creating a mobile website is a great place to start. This is especially true if:

- You don't know what your "new app" should do
- Budget is limited
- You already have a well-coded website that is a major proponent of your business.

A mobile website can actually be a fairly quick and inexpensive process to add on to your current website. I'm a fan of using CSS media queries to target mobile screen sizes and easily modify your current site for a mobile-friendly layout. If your website is created with semantic HTML and CSS stylesheets, you can quickly reduce column widths, hide less important content, and modify images for a great mobile experience. For websites using master pages of CMS templates, mobilizing websites with hundreds of pages does not have to be a daunting task.

If you already have a mobile site, or have a new idea that you think could make a great app, there are still a lot of aspects to consider before pulling the trigger. First, make sure you have a clear idea of WHY you want to develop a mobile app – then think about what you want it to do. Does your app do something new and unique? Does it do something a website can't? Is it actually realistic to turn a profit from sales? With the saturated app markets these days, if you're looking to make a commercially profitable app, a rule of thumb is to either be the first, the best, or have the most money. Make sure you do your research and know that you're app will have a clear target market and no competing apps that are too similar. Even a great idea can get lost in the sea of apps without enough marketing or brand recognition to propel sales.

An example of a mobile website created by dynamically adjusting the regular website with CSS media queries

So a stellar app with mass market-appeal and a huge ROI might be a pipe dream. But I also strongly believe there is a place for internal business apps to help increase productivity and profit. An app for a sales team to enter data out in the field can provide benefits that are easy to quantify. Save employees' time, provide customers with accurate quotes on the go, or send leads to your CRM system for instant business benefits.

HTML5 also presents a lot of opportunities for web-based apps with a lot of mobile capabilities. But a device app is a necessity if:

- You need access to the device camera, microphone, accelerometer, address book, media library…etc
- You will be using the app store/market for payment
- You need to send push notifications
- You need to run as a background service

If you decide that a device mobile app is the best route, another aspect of consideration is whether to use a multi-platform framework (i.e. PhoneGap, Titanium) or go native. At Gravity Works, we've built more than a dozen apps for iPhone, iPad, Android, Blackberry, and Windows Phone 7. At first, we were very much in favor of build once, deploy on multiple platforms. This stemmed from the hopes of providing more bang for the buck for our customers. But, with any new technology, sometimes those frameworks posed some mighty time-consuming problems.

We suggest that if your app is going to need to use many native phone functions – such as video – stick with native development. If your app provides functionality similar to what exists on websites – such as a directory – a program like PhoneGap can be a great resource. The melding of HTML, Javascript, and native programming languages can speed up development and allow designers to more finely control the app's UI.

Budget is another consideration within the framework or native development debate. Nothing beats a native UI if project budget can accommodate standalone development of each. But for a tighter budget where an app is still highly desired, Titanium or PhoneGap can speed development and provide apps for several platforms. Regardless of what development technique you use, the more you want an accurate UI for each device, the more time you will need to spend.

Before diving into a huge mobile project, make sure that you have considered all the options, and have made the best decision for your organizations desires, needs, and capabilities.

An app requiring video should be developed for devices natively

An app developed in PhoneGap for iPhone and Android, using web services with an existing website directory.

Local User Group Listings

We know that you are an active member of your local software development community and we want to help you to stay active. Over the last year we have compiled a list of interesting user groups in the region where most CodeMash attendees live: Michigan, Ohio and Pennsylvania. If you're looking for a new user group to attend, you'll probably find it here. Check the user group's website for details on meeting times, dates and exact locations.

If you have information about a user group that is not listed here, or a correction to one that is, please email the pertinent information to mashedcodemag@gmail.com.

Michigan User Groups

Topic	User Group	Meeting Location	Website	Twitter
Agile	Agile Groupies		http://agilegroupies.groupsite.com/main/summary	
Agile	Grand Rapids Testers	Grand Rapids	http://groups.google.com/group/gr-testers	
Agile	Michigan Agile Enthusiasts Group		http://tech.groups.yahoo.com/group/Michigan_Agile_Enthusiasts_Membership	
Agile	MidMichigan Agile Group	East Lansing	http://www.meetup.com/Mid-Michigan-Agile-Group/	@mmagile
Apple	Grand Rapids Area Macintosh Users Group	Grand Rapids	http://www.gramug.org/index.php	
Cloud Computing	Greater Detroit Cloud Computing Users Group	Detroit	http://www.detroitcloudgroup.org/	@detroitcloud
Computers	Computer Club of Western Michigan University	Kalamazoo	http://yakko.cs.wmich.edu/	
Drupal	Grand Rapids Drupal User Group	Grand Rapids	http://groups.drupal.org/grand-rapids-mi	
FoxPro	Grand Rapids Area FoxPro User Group	Grand Rapids	http://www.grafug.com/	@grafug
iOS/Mac	CocoaHeads Ann Arbor	Ann Arbor	http://cocoaheads.org/us/AnnArborMichigan/index.html	
iOS/Mac	CocoaHeads Detroit	Birmingham	http://cocoaheads.org/us/DetroitMichigan/index.html	@CocoaHeadsDet
Java	Ann Arbor Java User Group	Ann Arbor	http://www.aajug.org	
Java	Detroit Java User Group	Troy	http://sites.google.com/site/detroitjug	@detroitJUG
Java	Grand Rapids Java User Group	Grand Rapids	http://www.gr-jug.org	
Java	Greater Lansing Java User Group	East Lansing	http://groups.google.com/group/greaterlansingjug	
Linux	Grand Rapids Linux Users Group	Grandville	http://wiki.grlug.org	
Linux	Greater Lansing Linux User Group	Lansing	http://www.gllug.org	@gllug
Linux	West Michigan Linux Users Group	Grand Rapids	http://www.wmlug.org/	
Microsoft .NET	Ann Arbor .NET User Group	Ann Arbor	http://www.aadnd.org	@aadnd
Microsoft .NET	Great Lakes Area .NET User Group	Southfield	http://www.migang.org	
Microsoft .NET	Greater Lansing User Group .NET	Flint	http://www.glugnet.org	
Microsoft .NET	Greater Lansing User Group .NET	East Lansing	http://www.glugnet.org	
Microsoft .NET	West Michigan .NET Developer Group	Grand Rapids	http://www.wmdotnet.org	
Microsoft Administration	West Michigan NT Users Group	Grand Rapids	http://www.wmntug.org/	
Microsoft Development	Microsoft Developers of Southwest Michigan	Kalamazoo	http://www.devmi.org	@mdsm
Mobile	Mobile Monday Detroit	Detroit	http://mobilemondaydetroit.org/	@MobileMondayDet
Python	Grand Rapids Python User Group	Grand Rapids	http://www.grpug.org	@grpug
Python	Michigan Python User Group	Ann Arbor	http://groups.google.com/group/michipug	

Topic	User Group	Meeting Location	Website	Twitter
Ruby	Ann Arbor Ruby Brigade	Ann Arbor	http://groups.google.com/group/a2rb	@a2rb
Ruby	Lansing Ruby Users Group	Lansing	http://www.meetup.com/Lansing-Ruby-Users-Group/	@lansingruby
Ruby	Michigan Ruby Users Group	Grand Rapids	http://www.meetup.com/mi-ruby/	
Science/Technology	The Geek Group	Grand Rapids	http://www.thegeekgroup.org/	@thegeekgroup
Social Networking	DetroitNET		http://www.detroitnetworking.org	@detroitnet
Software Craftmanship	Software GR	Grand Rapids	http://www.softwaregr.org	@softwaregr
Software Development	Ann Arbor Computer Society	Ann Arbor	http://www.computersociety.org	
Software Development	Coffee House Coders, Ann Arbor	Ann Arbor	http://www.coffeehousecoders.org	@coffeehousecode
Software Development	Coffee House Coders, Detroit	Madison Heights	http://www.coffeehousecoders.org	@coffeehousecode
Software Development	Coffee House Coders, Downriver Area	Wyandotte	http://www.coffeehousecoders.org	@coffeehousecode
Technology	ConnecTech		http://www.connectech.org	@connectechD
Technology	West Michigan Small Business Technology User Group	Grand Rapids	http://wmsbtug.org/	
Technology	West Michigan Technology Association	Grand Rapids	http://www.wmta.biz/	
Unix & Linux	Michigan!/usr/group	Farmington Hills	http://www.mug.org	
Web Design	Refresh Detroit	Ann Arbor	http://www.refresh-detroit.org	@refreshdetroit
Web Development	Grand Rapids Web Developers Group	Grand Rapids	http://www.meetup.com/grwebdev/	@grwebdev

Ohio User Groups

Topic	User Group	Meeting Location	Website	Twitter
Adobe Flex	Cincinnati Flex User Group	Mason	http://cincyflex.groups.adobe.com/	@cincyflex
Agile	Cincinnati Agile Round Table	Cincinnatti	http://www.agileroundtable.org	@agileroundtable
Agile	Cleveland Agile	Cleveland	http://www.meetup.com/ClevelandAgile/	
Application Lifecycle Management	Central Ohio Application Lifecycle Management Group	Columbus	http://www.coalmg.org	
Business Process and Enterprise Integration	Mid-Ohio Connected Systems Developers Group	Columbus	http://www.mocsdug.org	
C# & VB.NET	Cleveland C#/VB.NET Special Interest Group	Independence	http://www.clevelanddotnet.info	@SamNasr
Clojure	Columbus Clojure User Group	Westerville	http://www.inclojure.com	@inclojure
Dynamic Programming Languages	Dayton Dynamic Languages Special Interest Group	Dayton OH	http://www.dma.org/sigs.shtml#Dynamic	
Information Technology	Computer Erie Bay Users Group	Sandusky	http://www.cebug.org	
iPhone Development	Columbus iPhone Developers User Group	Columbus	http://groups.google.com/group/cidug	@cidgu
Java	Central Ohio Java User Group	Dublin	http://www.cojug.org	@javajudd
Java	Cincinnati Java User Group	Mason	http://www.cinjug.org/	@cinjug
Java	Cleveland Java User Group	Independence	http://www.meetup.com/cleveland-java/	@javausers
JavaScript	Cincinnati JavaScript User Group	Cincinnati	http://blog.cincijs.com/	@cincijs
JavaScript	Columbus JavaScript User Group	Columbus	http://groups.google.com/group/cbusjs	@cbusjs
Microsoft .NET	.NET Special Interest Group	Independence	http://www.bennettadelson.com/ComingEvents.aspx	
Microsoft .NET	Central Ohio .Net Developers Group	Columbus	http://www.condg.org	@condg
Microsoft .NET	Cincinnati .NET User Group	Mason	http://www.cinnug.org	
Microsoft .NET	Dayton .NET Developer Group	Dayton OH	http://www.daytondevgroup.net	
Microsoft .NET	Findlay Area .NET User Group	Findlay	http://www.fanug.org	@fanug
Microsoft .NET	Northwest Ohio .NET User Group	Toledo	http://www.nwnug.com	@nwnug
Microsoft SharePoint	Central Ohio Sharepoint User Group	Columbus	http://www.cospug.org	@buckeyespug
Microsoft SharePoint	Cleveland Sharepoint User Group	Cleveland	http://www.sharepointcleveland.com	@sharepointcle
PHP	Columbus PHP Meetup	Columbus	http://www.meetup.com/phpphp/	@columbusphp

Topic	User Group	Meeting Location	Website	Twitter
PHP	Ohio, Indiana, Northern Kentucky PHP Users Group	Cincinnati	http://oink-pug.org/	@oinkpug
Polyglot Programming	Columbus Polyglot Programmers Meetup Group	Columbus	http://www.meetup.com/The-Columbus-Polyglot-Programmers-Meetup-Group	
Python	Cleveland Area Python Interest Group	Cleveland	http://www.clepy.org	@clepy
Ruby	Cincinnati Ruby Brigade	Cincinnati	http://www.cincinnatirb.org	@cincinnatirb
Ruby	Cleveland Ruby Brigade	Cleveland	http://www.meetup.com/ClevelandRuby/	@clerb
Ruby	Columbus Ruby Brigade	Columbus	http://www.columbusrb.com	@columbusrb
Software Architecture	Cincinnati Software Architecture Group	Cincinnati	http://www.cinnug.org	@cinnug
Software Architecture	Columbus Architecture Group	Columbus	http://www.colarc.org	@colarc
Software Craftsmanship	Columbus Software Craftsmanship	Columbus	http://groups.google.com/group/columbus-craftsmanship	
Software Craftsmanship	Hudson Software Craftsmanship Group	Hudson	http://hudsonsc.com/	@hudsonsc
Software Development	Bitslingers	Mason	http://www.cinnug.org	
Software Development	Cincinnati Programmers Guild	Mason	http://cincypg.org/	@cincypg
SQL	CincySQL	Mason	http://www.cincysql.org	
SQLServer	CBusPASS	Columbus	http://columbus.sqlpass.org/	@daveschutz
SQLServer	Ohio North SQL Server Users Group	Independence	http://ohionorth.sqlpass.org/Home.aspx	
Windows Presentation Foundation & Silverlight	Cleveland WPF User Group	Independence	http://www.clevelandwpf.info/	@SamNasr

Pennsylvania User Groups

Topic	User Group	Meeting Location	Website	Twitter
.NET	Pittsburgh .NET Users Group	Pittsburgh	http://codecamppgh.com/default.aspx	@pghdotnet
Agile	Agile Philly	Philadelphia	http://www.agilephilly.com/	
Clojure	Pittsburgh Clojure Users Group	Pittsburgh	http://www.meetup.com/Clojure-PGH/	@clojurepgh
Functional Programming	Philly Lambda	Philadelphia	http://groups.google.com/group/philly-lambda	@phillylambda
Java	Philadelphia Area Java Users' Group	Philadelphia	http://phillyjug.jsync.com/	
Java	Pittsburgh Java User Group	Pittsburgh	http://www.pghtech.org/networks/PittJug/default.aspx	
Java/Spring	Philadelphia Spring Users Group	Philadelphia	http://www.phillyspring.org/	
jQuery	jQuery Pittsburgh	Pittsburgh	http://jburgh.com/	@jquerypgh
Ruby	Philly.rb: Philadelphia Ruby User Group	Philadelphia	http://phillyrb.org/	@phillyrb
Ruby	Pittsburgh Ruby Brigade	Pittsburgh	http://pghrb.org/	@pghrb

COLUMBUS STATE
community college
Center for Workforce Development

YourWorkforcePartner

Project Management

PmCentersUSA

Project Management Training in partnership with PMCentersUSA

Course Offerings

- Project Management Masters Certificate Track
- Project Management Overview
- Project Leadership
- Project Scope & Quality Management
- Project Schedule & Cost Management
- Project Risk & Contract Management
- Project Human Resource & Communications Management
- Project Management Professional (PMP) Examination Preparation*
- IT Project Management*
- Managing Multiple Projects*
- Turning Around Problem Projects*

**Course is not required for Masters Certificate in Project Management.*

Columbus State and PMCentersUSA©'s project management training is designed to meet the standards established by the PMIT, as well as to transfer knowledge and develop skills in the project management discipline.

This courseware was created to be compliant with the PMI's Project Management Body of Knowledge and structured to prepare individuals with knowledge and course hours for the professional certification examination. The Master's Certificate Track consists of seven focused courses in project management, delivered in four evening classroom sessions. When the program is completed, students receive a Master's Certificate in Project Management Students also earn 84 PDUs for classroom learning or 72 PDUs for virtual learning.

Discounts for enrolling in complete series and group pricing options are available.

For more information:
Visit: www.cscc.edu/workforce/it-professionals/projectmanagement.shtml
Contact: Dave Watts | 614.287.5787 | dwatts@cscc.edu