# acf.objective() collaborative notes 2011

Doc Created By: mark.drew@gmail.com
Other Authors: troy.murray@gmail.com
jean.ducrot@interfolio.com
kevin@morrisreport.com
Editors: brandonmoser@gmail.com

## Table of Contents

# Day 1

Stephen Hauer, Mark “Cayshing” Mandel, Bob Silverberg, Jason P Dean, Ray “Beardy” Camden, Jared Ripka Hauer, Scott Stroz

## Keynote

- Speaker evaluations for each session are on the site, so that attendees to fill them out for each session.
- “God is a PHP Developer, it took him 6 days to create everything, if he was a CF developer it would only take 4 days, and then a 3 day weekend”
- Scott Stroz doing his “shtik”
- Scott is really funny :)

- The thing that sets us apart is the community
- Ray Camden talks about ColdFusion Builder
- “Programming is like sex, one screw up, and you have to support it for life” --Jared RH
- Ray showing CFBuilder shortcuts.
- JThose who use it love it
- Don't mention CF in your Job Postings: say “web developer”, “experienced web developer with complex applications” or “experienced OO web developeason Dean: ColdFusion is not dying, but, we do have a dying breed of CF developers
    - r” so that you get the widest range of developers
    - Talk about Jason Dean is up (don’t confuse him for Mark Drew)
    - Less coldfusion developers out there.
    - Talking about how to Make ColdFusion Developers, why search for a “ColdFusion” developer? Why not get good web developers, and train them.
    - Don’t post your jobs with “ColdFusion” say , Web Developer or more descriptive.
    - hire students, ambition is better than experience
    - Who should create them? Adobe? No. Schools? No. Managers? No.
        - It’s the developer’s responsibility.
    - Teach at community colleges, train the right people
- Bob Silverberg (He’s Canadian doncha know?)
    - Good comeback from Bob Silverberg to Scott Stroz (quote?) “Scott would get extra “Virginian” points for saying to everyone to but a blindfold and come round his house” … you had to be there.
    - Contributing to Open Source
    - You should give back to the community from where he got a lot of help too.
    - You learn a lot, in your own time, such as TDD, Version control
    - “Bob: How can I involved in Open Source Projects?” - “Lots of people at #cfobjective that are actual contributors already!”
- Mark Mandel: ColdFusion News
    - JRun is being removed in favor of Tomcat in the next version of ColdFusion
    - Verity will be removed in the next version of ColdFusion and completely replaced with Apache Solr
    - Axis 2 is going to be in CF
    - Improved Scheduled services
        - chaining: run task after another task
        - conditions: if CPU below a certain thresh-hold, run now
        - application specific: tasks doesn't have to be setup on instance for everyone
        - jobs: asynchronous message queue

    - `cfjob` functionality (which Railo also already has!)
    - JavaLoader will be integrated in the next version (btw, Railo has had this for ages!)
    - HTML5 and JQuery - cool things are coming.
    - Demos?!
    - Closures are coming to ColdFusion (although personally the syntax is dumb) (it’s not too late to change the syntax. get on the alpha)
    - `ArrayEach(departmet.getEmps(), closure(employee){ … do something});`
    - OWASP is going to be integrated into CF
        - `encodeForHTML()`
        - `encodeForHTMLAttribute`

        - `encodeForJavaScript()`
        - `encodeForCSS()`
        - `encodeForJS()`
    - ColdFusion X is coming sooner than expected!
- Stephen Hauer on closing notes
- Get out there and do it!

## Progressive Enhancement: What is it and how do I do it?

- Examples:
    - Escalator is graceful degradation, it's awesome but scales down as a stair case
    - Elevator is not a progressive enhancement, more like an RIA, if it's working it's great, if it's not working then it can't be used at all.
- What is it:
    - Start a very low level and work up from a working application.
    - CSS is an example of progressive enhancement
    - Important with Javascript to make sure there is a graceful degradation for older browsers (IE6 I'm looking at you) or Javascript is turned off (e.g. corporate policy)
- Graceful Degradation
    - Target high-end browsers
    - Degrade for other User Agents
        - Such as rounded corners or client-side validation
- Semantic Markup
    - Describe functionality and structure of the application
        - HTML4 is reasonable
        - HTML5 standardizes semantics
            - CSS example
                - `.red` is not a correct semantic structure
                - `.error` would be correct semantic structure
            - While HTML5 isn't a standard yet, it does degrade fairly well with its `input` tags for example
- Element Targeting
    - CSS
        - Present semantics such as targeting error messages or navigation.
        - Targeting specific areas of the application
    - JS - enrich semantics
- Functional Targeting
    - Don't assume a feature is available, such as geolocation
        - If the person has this feature, we can make their experience better; but if not, then it's still functional
    - `jQuery.support.`*xxx* = does checks to see if the function is available on the client
    - Modernizr can also help target specific features for your browser
- Progressive Enhancement
    - *Crux of matter is it works at a basic level, but if the client has the ability it will enhance where possible.*
    - In doing this, start simple and add additional things
- Client-side Manipulation

    - DOM manipulation
    - Partial-page injection
        - Example would be the page asking the server for a chunk of content, like a table, and then inserting it into the page
    - Watch out for “state synchronization”
        - A pager on your web-page, you click to go to the next page, make sure all of the other objects on the page that might be tied to that are also updated or refreshed.
- Example
    - A form which detects if the browser supports geolocation.
        - If it does, then it attempts to lookup the lat & long
        - If it doesn’t have it, then it provides the user an input for their zip code
    - User form
        - Form to add a new username
            - If the client supports AJAX it will return to the user immediately a message if the username if available or not
            - If the client doesn’t support AJAX then on submit it should return the message to the user.
    - Delete links will submit a GET request, which according to spec could be cached by the browser.
        - You could have the link go to a page asking if the user is sure, then submits the confirmation as a form post.
        - This could be enhanced using a modal window using something like jQuery
        - Or if the client supports AJAX you could capture the click and run the method to delete the record.
        - Regardless you still need to have the pure HTML static version for the user that doesn’t have the ability for jQuery or AJAX working first
- *This is the take-away (aha moment!)*
    - Progressive Enhancement gist
        - Develop for an old browser (e.g. IE 6), then add enhancements for modern browsers (e.g. Chrome)
        - This will make sure that the baseline functionality is there
        - This is an easier way to develop
        - Build working, make it awesome!
    - Graceful Degradation gist
        - Develop for a modern browser (e.g. Chrome), then work backwards for older (e.g. IE6) browsers.
        - This is a more painful way to develop
        - Build awesome, pray it works
    - Semantic structure is needed with markup alone and working
- Soapbox
    - If you’re not using version control, start.  Before you write another line of code.
    - If you don’t use a Front Controller framework, start.
    - Learn about your tools.  They’re all far more powerful than you believe.
        - Eclipse / CFBuilder has so many things to help you but you’ll never benefit until you start looking.

## Undocumented and Off Script: ColdFusion & Ehcache

- Caching Architectures
    - In-Process (L1 cache)
        - very fast
    - Out of Process (L2 cache)
        - separate JVM
        - much more scalable
- Caching Strategies
    - Non-deterministic (cache aside)
    - Deterministic (cache as SOR (system of record))
        - always go to cache
- Cache Eviction Algorithms
    - Time Based
        - Time period
        - expiration date
    - Cost based
        - First In, First Out (FIFO)
        - Least recently used (LRU)
        - Less frequently used (LFU)
            - unique to Ehcache
- http://java.dzone.com/articles/building-high-performance
- @styggiti

## ORM Zen

Marc Esher , http://bit.ly/cformzen (presentation and samples)

- ORM starts off simple, but then it starts getting complicated.
- Why won't you save relationships? Does inserts and updates? WTF?
- ORM looks like a cockpit to start off with, now it looks like a couple of knobs. (yes, I wrote that, not Marc)
- Objects will be saved to the DB at the end of the request even if you want them to, to change that you can do `automanageSession="false"`
- Use transactions to flush and thus save objects inside a transaction
- `fieldtype` is one of the knobs.
- Once you add relationships, make sure you are setting an object as the related object, rather than the simple value, as the error is kind of odd.
- Watch out for the one-to-many relationships' "inverse" attribute
- (Marc is using Mylyn for Eclipse) Check out the talk this week at cf.Objective called "Task Management"
- "one-to-many" is very gluttonous, if you see one to many properties, and you are NOT using them for saving, you really need to check if you needed to do that. Since it's pretty hefty to return the "many" properties.
- ORM and relationships are causes of suffering.
- Inverse:
    - Attendee & Events, Attendee Has Attendences
    - Use `cascade` & `inverse`, they also give you a lot of pain
    - When you add a new related object to a collection, it wont save - there has been no error, but

there is no ID.
    - No insert in the console.
    - Attendee has `inverse="true"` has removed the insert then update, but it's not saving the children.
    - Then we turn on the `cascade="all"`, now it saves the children too.
    - TWO missing attributes caused the the problem.
    - Need `cascade="all"` on the one-to-many property.
    - If you try and delete, by removing the attendance and setting it to `null`
    - Use `cascade="all-delete-orphan"`, finally does the saves, does the proper inserts, as well as deleting the orphans
- Concurrent Modification Exceptions
    - If you are looping over an array in a `for` loop, and you change the array (by deleting using `ArrayDeleteAt`), you will get a "Concurrent Modification Exception"
    - Use a backwards array loop, with the `ArrayLen()` INSIDE the loop
    - Do NOT use persistent objects in CF Sessions.
- Drink 32 bottles of Scotch
- @marcesher

## Code Deployment Shouldn't Only Be FTP - Using ANT to Automate Your Build Process

- Another Neat Tool (ANT)
- ANT has a huge number of add-ons to do so many things
- Java-based and built into Eclipse and CFBuilder
- ANT was created to help build the Apache Tomcat server
- Can use to build a JAR, deploy an application or even generate documentation!
- Uses XML file to describe the build process and dependencies
- Benefits of using ANT
    - Reduce the amount of errors in tedious tasks
    - Can package and deploy our CFML application
    - Setup / manage database schema
    - Application testing
        - DBUnit
        - MXUnit
        - Selenium
- Common uses of ANT in CFML
    - Automatic CFC documentation generation
    - Automatic "var" scope checking
    - Email results regarding success/fail of targets
    - Build your own
        - MacroDef is used for this
        - Example: Remove the __MAC_OS_X file/directory that's automatically added to .ZIP files generated on that system
- The build.xml file is usually located in the root of the project
    - Can be run from command-line, Eclipse or Continuous integration product
    - Example can be found at http://trac.mach-ii.com/machii/browser/framework/1-9-0/trunk/build.xml
- Anatomy of build.xml file

    - “name” - Name of the project (not required)
    - “default” - Default target to execute if no target is defined (not required)
    - “basedir” - Usually left at “.” (not required)
    - Ordering of tags is loose, best practice is to group like tags together
    - If you use paths in the build.xml file it restricts it’s portability
        - This is a good reason to put build.xml at the root of your project
- Anatomy of targets
    - Uses the attributes of “name”, “depends” and “description”
    - Use a descriptive name
    - Can have none or multiple depends
        - All depends run BEFORE this target, NOT AFTER.
- Anatomy of properties
    - These don’t persist past the currently running build
    - Can hold only simple data types of name/value pairs
    - You can pass in a value for these from the command-line arguments when calling the ANT build process
    - These are immutable, the first one set wins FOREVER!
    - You can also reference property files
        - Example: each developer could have their own property file and the ANT build.xml would use these values
- Anatomy of taskdef
    - Piece of code that can be executed, this could be a third-party function such as working with Subversion (SVN) or FTP
- Create a build target
    - Copy files into this location
    - Delete, move and rename files as needed
    - Always work with copies so you don’t lose anything

```
<tstamp>
      <format property="buildDatetime" pattern="yyyy-MM-dd HH:mm:ss" />
</tstamp>
```

- Use `filterset` tokens to replace values in your code files to some value generated during the build
- Tips:
    - Make the default target a “help” target that lists all targets and functionality
    - Use `<cfexecute>` to call ANT
    - Include the JAR’s for any needed functionality for the build inside the project so everyone is using the same version
    - Put all of this, including the JAR’s, into version control with your project
- WikiBooks - http://en.wikibooks.org/wiki/Apache_Ant
- Best book is “ANT in Action”, ISBN: 193239480

## Maximum Security CFML

- Most common threat is negligence, 41% of issues are the result of ourselves
- Your client/manager is expecting us to know what we’re doing with regard to web app security
- Ways to help you
    - Logging - make sure you’re logging things that maybe useful evidence if compromised
        - Exceptions

        - File operations
        - You could use CFlog, database, log4j or syslog
    - Auditing
        - Check successful or failed logins
        - Add, edit or delete data or users
            - In some cases read events (think HIPAA)
        - Other business specific requirements
    - Intrusion Detection
        - Detect attempted compromises in the requests
    - Authentication & Authorization
        - Passwords should never be stored in plain text, use a hash algorithm AND salt
        - Hashing is one way encoding of string
            - Unlike encryption which can be reversed
            - Different hash algorithms: MD5 (best not to use), SHA (developed by NSA)
            - SHA-512 best to use if you don't have to worry about compatibility with legacy application.
        - Salt passwords
            - The hash will always produce the same value, best to "salt" the hash by adding an additional unique string
            - Can use anything that's unique for the salt, best not to have a field named "salt" in the database table though
        - Account lockout
            - Record in the audit log
            - Sleep the failed login (has to wait to unlock), but be careful not to DOS yourself
            - Password strength requirements
                - No need to limit max length of password
                - Since you will hash the password it should be a consistent field length
            - Hash iterations
                - Put your hash i a loop to increase execution time to deter brute force
                - But beware of timing attacks
    - Session Hijacking
        - If your CFID & CFTOKEN or JSESSIONID can be determined, I can impersonate you
            - Never put these in the URL
            - Always set `addtoken="false"` when using `cflocation`.
        - Require SSL and secure cookies (will only send cookie if using SSL)
        - Set HTTPOnly FLag on cookies (see my blog for example code)
            - CF 9.0.1 Added JVM argument
                - `-Dcoldfusion.sessioncookie.httponly=true`
    - Insecure File Uploading
        - Very common, very dangerous
            - Risk = an attacker uploads and executes the file on your server
        - Vulnerable code
            - `<cffile action="upload" filefield="photo" accept="image/gif, image/jpeg, image/png" destination="#expandPath('./photos/')#">`
        - The accept attribute does NOT limit the types of files that can be uploaded
            - It doesn't check the content of the file
        - Tips:
            - If you upload files do it to a NON-web accessible directory (this is important!)

- Always validate file extension
- Never upload under web root
  - Only copy web root once you've verified the file
- Try/Catch & Delete on anything uploaded
- Prefer whitelist over blacklist
  - allow JPG, PNG, GIF, PDF
  - vs block CFM, CFC, etc
- Validate File Content if possible
  - `isImageFile()`
  - `IsPDFFile()`
  - `IsSpreadsheetFile()`
  - jHOVE - Java API for additional types
- Make sure you're running the latest JVM
- Deny execution for upload destination directory
  - ON Web Server
  - In ColdFusion (with Sandbox Security)
- Serve files from static content server
  - Build your own
  - Amazon S3
- Linux set MODE attributes

- SQL Injection
  - Use `<cfqueryparam>` on any variables that come from users
    - Works with INSERT, UPDATE and SELECT
    - Can be used in IN statement using attribute `list="true"`
  - Is possible with ORM
    - Use parameters to prevent this
- XSS
  - If URL values are used on the page and unchecked a hacker could insert Javascript
  - It's possible to create an entire form using this hack
  - To prevent this is more difficult
    - Not always realistic to strip all harmful characters < > ' " ( ) ; #
    - Encode variables to escape special characters
      - Best way to do this depends on where the variable is output, in a tag attribute, inside Javascript
  - `XMLFormat()` or `HTMLEditFormat()`
    - `XMLFormat` Escapes < > ' "
    - `HTMLEditFormat` Escapes < > "
  - Better to use the OWASP Enterprise SEcurity API
    - Java API that has encoder methods for each context
    - http://code.google.com/p/owasp-esapi-java/
  - AntiSammy for Java
    - Create policy for what's allow in HTML
    - ESAPI has integrated AntiSamy in it's validator implementation
      - `ESAPI.validator().isValidSafeHTML()`
    - As me who's "Samy" later.
  - ColdFusion 7 added ScriptProtect feature to "protect" form, url, cgi and cookie variables from XSS
    - Very limited and easy to bypass

    - Path Traversals
        - Problem
            - Allows the attacker to read any file CF has permission to read
                - `<cfinclude template="files/#url.page#">`
            - Instead of page.cfm?path=about.cfm
            - Hacker adds path=../../../<whatever file here>
            - It's limited to the same drive that the vulnerable file is stored on
            - Null bit injection
        - Solution:
            - Applies to `cinflude`, `cfmodule`, `cffile` and file functions or file paths
            - Validate supplied variable
            - Could use a `<cfswitch>` for expected values.
            - Don't trust inputs
- Security Guidelines
    - Do NOT trust what's coming in
    - Validate EVERYTHING, all inputs
        - More validation you can add to your code, the more secure it is
    - Fine grained validation is best
    - Remember that the entire HTTP request is an input
    - Careful what you output
        - Scrub and sanitize all outputs
            - Ensure that all variables are encoded and escaped properly
    - Bring security into your Unit Tests
        - Ensure that your app does not accept malicious input
    - Catch Exceptions with try/catch, `onError`, `cferror`
    - Don't disclose system details to end user, log the details
    - Could use virtual file system in ColdFusion 9 for file uploads
- Security Tools
    - Fuseguard
    - hackmycf.com
- @pfreitag

## Coldspring 2.0

- Code name Narwal
- Ready for release just waiting on documentation
- Complete rewrite CF8+ compatible
    - Leverage some of the advancements in CF since version 8
    - NOT backward compatible but migration guide is coming up
- Coldspring.xml file now available so code completion will be available
- New scope attribute for beans
    - Mutually exclusive with singleton property
- Bean process interceptor
    - Intercept when a bean gets created
    - Useful for debugging
- XML schema debugger

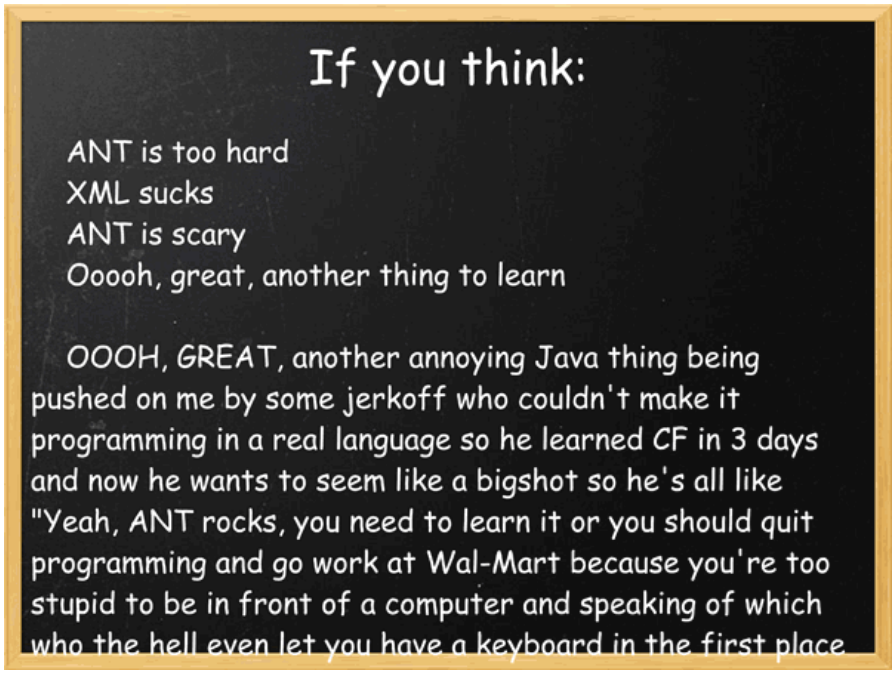
    - Add debug attribute to the XML schema definition, set it to debug
- AOP configurable with auto generated proxy
- Comes with its own gateway
- http://coldspring.sourceforge.net/ber/docs/xsd/beans/

## Forms That Don't Suck (Quick, Easy, & Clean Forms and Data)

- Solutions
    - cfUnifForm
    - ValidateThis
- Forms suck
    - Because of browser battle
    - Client-side data validation is aggravation
    - Repetitive writing of server-side data validation
- cfUniForm - custom tag library for CFML
    - Wrapper around the Uni-Form markup spec created by Dragan Babic
    - Semantic XHTML complaint markup
    - Consistent display across mot modern day browsers
    - Themed CSS
    - Write CFML, and don't be concerned with (x)HTML/CSS/JavaScript
    - Can be used inline or block for the layout
        - `<uform:fieldset class="blocklabels">`
    - Three themes packaged with the tag library
    - Has jQuery plugin integration for "special fields"
        - Date of birth (calendar)
        - Appointment time (time)
        - Ratings
        - Phone mask
- ValidateThis
    - 80% of more of the bases covered out-of-the-box
    - Extremely extensible to cover almost anything needed
    - Server and client-side validations from one simple definition file
        - XML
        - JSON
        - Annotations
    - Implement
        - Configure the application
        - Configure the form
        - Configure the controller
        - Enjoy the benefits of clean forms and data

## Continuous Integration with Jenkins, ANT, and MXUnit

Rating: http://www.cfobjective.com/index.cfm/sessions/continuous-integration-with-hudson-ant-and-mxunit/
Presentation Download: wiki.mxunit.org under Presentations

- Everything we do changes (db, tests, code)
- Continuous integration seeks to solve the deployment of this
- CI is actually easier than you think.
- What is life like with Jenkins? (Used to be called Hudson)
- CI: Designed to solve the distributed building of compiled applications
- CI does a lot, but it won't tell you what is wrong.
- Golden Age of CF Testing (like CFSelenium)
- No project ever started with 500 unit tests, they all started with zero. It's always a good time to start.
- You can't avoid Ant any longer
- 

- Ant cheat sheet
    - CFPARAM <property />
    - CFFUNCTION <target />
    - CFINCLUDE <import/> & <taskdef />
    - CFFILE <copy /> <delete />, etc
    - CFHTTP <http.../>
    - CFDIRECTORY <mkdir />
    - CFOUTPUT <echo />
    - CFDUMP <echoproperties />
    - CFZIP <zip />
    - #variable# ${property}

## CF Powered mobile apps

- Start with a mock up > Create a click through > Connect to CF > Create Flex app
- Mockups focus on information placement and navigation. No design should be discussed
    - Ask yourself:
        - What are the key points of the app?
        - Am i displaying too much?
        - What are the most popular features?

        - Where do I want users to go?
        - Is there too much clicking going on?
- jQuery mobile can help you create a click through. It doesn't mean you have to use it later.
- To show apps to clients you can use emulators or Opera mobile
- Purpose of click through:
    - Focus on info display
        - is it correct?
        - is it in the right location?
        - can the flow be broken?
    - Focus on navigation
        - is flow natural?
        - is mock up represented correctly?
    - NOT A DESIGN STEP, make sure client understands that.
- Adding ColdFusion back end
    - JSON or XML
    - Always 2 page loads: layout then content
    - Just mark your CFC as remote and you can call them as web services. That's it!
- Functional application
    - Pretty much
    - You can deploy it as a mobile web site if you want to
    - You can slap a FLEX app on top of it
- In CF9.0.1 there is an actionscript library which handles online/offline storage ffor connection with desktop apps
    - With CF you just need to implement a couple of functions: fetch and sync

# Day 2

## Setting up a Solid Local Development Environment

- What is a Local Development Environment?
    - A mini production server on your workstation
    - A place where you can experiment without worrying about affecting your team members or clients
- Why do we need it?
    - Standard development setup across all developers
    - Testing locally allows you to experiment
    - Overwriting files on a shared server can be disastrous
    - The closer your configuration is to prod allows you to identify changes that will be necessary in production
- Parts of a Development Environment
    - Web server with virtual hosts
        - not all versions of IIS support
    - One or more CF instances
    - A local DB or connection to development DB
    - A local checkout from source code control

    - Development tools to make you productive
- Web Server
    - IIS vs Apache
        - Apache is considered easier to manage
        - Apache is free and open source
        - Apache is the most popular web server
        - Consider using Apache locally even if you deploy to IIS so you have take advantage of virtual hosts
- Apache with Virtual Hosts
    - Configure a new virtual host in Apache by editing Apache's `conf/httpd.conf` file
    - Update your hosts files
        - located in `/etc/hosts` on OS X / Linus
        - located in `/Windows/System32/drivers/etc/hosts` on Windows
    - Restart Apache
- CF with multiple instances
    - Each instance  has its own settings, datasources, and mappings
    - One instance per client
    - One instance per environment
- Development Databases
    - MySQL vs MS SQL
        - you can run sql server on a mac using VMWare or Parallels Desktop
    - keeping db in sync across developers
        - DDLUtils
- Source Code Control
    - SVN vs CVS vs GIT
    - CVSDude offers inexpensive hosting
        - Consider installing Trac, Skweegee, or ViewCVS
    - Source code control directory setup
        - Root of your project as the webroot
        - One option: Use branches to split out your development projects from the truck which should match production

## Introduction to Browser Automation and Testing with Selenium

- Unit testing is testing our application from the inside out
    - This is what you use MXUnit for
- Functional testing is testing our application from the outside in
    - This is what you use Selenium for
- Firefox only
    - There isn't a Firefox 4 plug-in yet, you'll need to use Firefox 3 for now
- Locator's
    - Indentifier -> default
    - Link Text -> link=
    - CSS Selectors -> css=
    - XPath -> xpath=
    - DOM -> dom=
    - If you can use a ID, then you should!

- When in the IDE you can double-click on each step to walk through the tests steps you've defined
    - If you add a new step as you're doing this, it will add it to the bottom of the test list, but you can drag that test list up to the correct location.
- You can create a test suite of multiple tests
- You can use pattern matching
- Assert vs Verify
- Limitations
    - Firefox only
    - No Ifs, Loops, etc
    - Minimally programmable with JavaScript
- Selenium-RC (Remote Control)
    - Any browser
    - Fully programmable
    - Can use multiple languages, including **CFML!**
    - Runs as a Java server
- What is CFSelenium
    - Single CFC that you instantiate
    - Source code: https://github.com/bobsilverberg/CFSelenium
- Selenium 1 vs Selenium 2
    - Many feel that "Selenium 1" is dead, you should be using Selenium 2, but it's not quite finished yet.
    - CFSelenium uses Selenium 1
-
- Presenter: Bob Silverberg
    - Blog post about CFSelenium: http://www.silverwareconsulting.com/index.cfm/2011/2/22/Introducing-CFSelenium--A-Native-ColdFusion-Client-Library-for-SeleniumRC
    - @elegant_chaos

## Requirements and Estimating

- Almost 60% of features requested in software are almost never used
- Will cover the following:
    - Requirements
        - Intent Driven Design
        - User stories 201
    - Estimation (estimate is "I think it will take this long" vs commitment "will be done by this day")
        - How much?
        - Estimating scope
        - Estimating duration
    - Managing risk
        - Four types of features
        - Dealing with dark matter
    - Managing commitments
        - Fixed duration
        - Fixed price
        - Breaking the iron triangle

- Requirements
    - What should we build?
    - Intent Driven Design
        - Business intent
            - Why should we build this?  What's the value?  Why?
            - Good projects has 1 to 5 good business reasons
            - If it needs more then this, it should be broken into different projects
        - Audiences
        - Objectives
        - User stories (tasks)
            - What are the tasks that other people need to do on the website
            - "I want to check my order status"
            - This allows us to define acceptance criteria for this task
            - Write each one of these on a 3x5 index card
                - This isn't a legal document
                - It's a basis for a conversation with the business team
                - Should have a clear definition of "done".
            - Invest
                - Independent
                - Negotiable
                - Valuable
                - Estimable
                - Small
                - Testable
                - "I'm going to log into this system, click on this button, and see this result when I do"
                    - This can now become your Selenium test
            - Use tracer bullets, not tasks → "As <u>who</u>, I want <u>what</u> so that <u>why</u>."
                - I need to create the controller
                - I need these database tables
                - Etc
            - Splitting stories by...
                - Data
                - Validation
                - Paths
                - Edge cases
            - Isolate cross cutting concerns
            - Make it work, then pretty/fast
- Estimation
    - Why estimate?
        - Good reasons to estimate
            - Go/no go based on cost
            - Market window
            - ROI comparison
        - Bad reasons to estimate
            - Just because.....
        - Ask whether:
            - Time, price or both?

                - Will estimate matter?
                - Accuracy required?
                - The project shouldn't produce a 10% increase, but a 10x increase
        - Estimating scope
            - Ideal days (load factor)
                - How many times to move this pile of sand to that area?  How many people? How many steps?
                - Start with estimate how of much uninterrupted ideal days
                - Over time this is problematic, because rarely do we have ideal days
                - Load factor = how many ideal days per week.  Maybe only 2 - 3 ideal days per week
            - Story points (relative for task, we decide the value)
                - 1,2,3,5,8
                - 0? = never use this, if task is small, roll up multiple into single
                - 10.20,30,51?
            - T-shirt sizes
                - Small  = build registration page
                - Medium
                - Large
                - X-Large = build own eBay auction system
            - Story count
        - Planning poker
            - Keys:
                - Delphi technique = works well until you have too many stories
                - Independent estimates
                - 1-3 hours, 1-3 sessions
                - 2-6 people
        - Estimating velocity
            - Historic rate
            - Run iterations
            - Make forecasts
        - Burn down charts
- Managing risk
    - Four types of features
        - Rocket science
            - When someone asks you to do something that you're not sure has ever been solved before
            - This challenges functionality requirements
        - Lab experience
            - This doesn't challenge functionality requirements
                - Needs to support x users using browser x under this load on this hardware
        - New to you
            - However long you think it will take, double it to be safe
        - With a twist
            - We have a list of products, now we need to have a list of categories
            - Will usually take 20% longer then you expect
    - Dealing with dark matter

            - What you don't know does hurt you...
                - But obviously...
                - The dreaded API and technical risk
                - Well, that's almost right...
                - Now that I see it..
            - Mitigate:
                - Set expectations
                - Technical spikes
                - Track and refine
- Managing Commitments
    - Fixed duration
        - Optional features
        - Team size
        - Real options
        - Schedule buffer

## Everything you wanted to know about REST and more

- History of REST
    - Term coined by Roy Fielding
    - Describes architectural style of networked systems
    - Co-founded the Apache HTTP Server Project
- What is it and what it does
    - Representation State Transfer (REST)
    - Trying to make something stateless, client-server, cacheable communication protocol
    - Architecture style for designing networked applications
    - Lightweight alternative to RPC, SOAP, WSDL, etc
    - REST is NOT a ***standard!***
        - It's more of a style then anything
- Key principles
    - Humanly readable URL's
        - www.mydomain.com/products/bouncy-ball/comments
    - Link things together
    - Use standard methods
        - GET, POST, DELETE, HEAD
    - Resources with multiple representations
    - Communicate statelessly
        - If normally uses authentication, how do I handle this since no previous actions are required
        - Common variance with this is to use a token
- Rest architecture components
    - A "thing" is a resource
        - Product
        - Category
    - Everything that can be identified should have an ID
    - Resource names and ID's improve the readability of URL
    - ID's do not have to be numeric

        - Could be text as it's more human readable
    - Hypermedia as the engine of application state (HATEOAS)
    - Provide full inks to referenced resources
    - Provide full links to additional information
    - Do not force the user to create their own link
    - Linking allows the application to easily move between states
    - Providing the link lets you alter location sand structures later
    - Example:

```
<posts link="http://www.simonfree.com/blog/posts"/>
    <post link="http://www.simonfree.com/blog/posts/123"/>
```

    - Standard methods
        - Get - returns data = should never change the state of the data, only return information!
        - Post - saves data = creates the item (blog post, comment, etc), avoid using for updates
        - Put - updates data
        - Delete - removes data
    - Resources with multiple representations
        - Different formats of a resource should be available
        - Important that an application knows what format it will be receiving
        - Use the accept feature of a request to specify the requested format
        - Try not to use file extensions if possible
        - Utilize http error codes
    - Communicate stateless
        - To be stateless, a server should not have to keep a communication state stored for any client (session, cookie etc)
        - This is often broken when security is involved.  Hashed keys of u/p is an alternative.
    - REST as a lightweight web service
        - Should be platform-independent
        - Should be language-independent
        - Standards-based
        - Can be used in the presence of firewalls
        - No inbuilt security or encryption features
        - SOAP = verbose response, like a letter
        - REST - simple response, like a postcard
    - Server responses
        - Server responses are often XML, but you can be whatever format you want
        - Unlike SOAP, REST is not bound to XML in any way
        - One response option not acceptable is HTML
    - Architecture components
        - Some components of REST architecture:
            - Resources - represented by logical URL's
            - A web of resources - linking resources together
            - Cacheable - uses expiration date/time
    - Design Guidelines
        - Do not use "physical" URL's (not point at a .cfm file)
        - Queries should not return an overload of data
        - Make sure responses are well documented
        - Always return URLs rather then making the client construct them
        - GET access requests shoudl never cause a state change

- Common mistakes
    - Overuse of POST
    - Putting actions in URI's
    - Using sessions and cookies
    - Not using status codes
    - Targeting specific languages
    - Not limiting results set (pagination)
    - Not escaping unsafe characters
    - Do not return the error detail to the user other then the error code
- Doubts about REST
- Some examples

```
<cfparam name="url.format" default="xml" />
<cfsetting showdebugoutput="false" />
<cfinclude template="functions.cfm" >
<cfscript>
      //set up the variables used for the page
      availableMethods = 'get,post,put,delete';
      dataMethods = 'post,put';
      stResponse = StructNew();
      stResponse['statusCode'] = ;
      stResponse['statusMessage'] = ;
      stResponse['statusDetails'] = ;
      stResponse['statusData'] = ;

      requestData = getHTTPRequestData().content;
      requestData = toString(requestData);
      stRequest = StructNew();

      if(len(trim(requestData))){
            requestData = xmlParse(trim(requestData));
      stRequest = xmltostruct(requestData.request);
      }

      if(!listfindnoase(availableMethods,cgi.request_method)){
            stResponse['statusCode'] = '405';
            stResponse['statusMessage'] = '405';
            stResponse['statusDetail'] = '405';
      }
      <cfset GetPageContext().Getout().ClearBuffer()
      />#formatResponse(stResponse['resp
```

## Advanced Web Application Security

- Storing passwords securely
    - Hashing
        - 1-way password encryption

        - Do NOT store
        - Algorithms
            - MD5
                - Do NOT use MD5
            - SHA-256
            - SHA-512
        - Collisions - two or more strings that return the same hash value
            - avoid by using larger/stronger algorithm
        - Iterate over hashing algorithm 1000x or more
    - Salting
        - adding a string to the beginning and/or end of the plaintext password to make hashed password more unique
        - generate different salt for each user
        - makes brute force attack much more difficult
    - JBCrypt
        - std hashing alg are fast, really fast
            - not a good thing, it makes brute force attacks easier
        - slows down hashing and salting process by introducing a work factor
        - future-adaptable to changing processor speeds
- Strong Cryptography in ColdFusion
    - Encryption Algorithms in CF
        - CFMX_COMPAT
            - barely encryption
        - [*lost internet connection and this slide of notes didn't get saved to GDocs, does anyone else have the list of algorithms in CF and Jason Dean's recommendations?*]
    - Modes of Operation
        - Electronic Code Book (ECB)
        - Cipher Block Chaining (CBC)
    - Even stronger cryptography
        - by default, JEE limits the strength of crypto
        - due to export restrictions
        - Unlimited strength policy files installation
            - Download files from Oracle
            - Place in <coldfusion>/runtime/lib/security directory
            - Restart CF
        - PKI Crypto in CF
            - Possible
            - Allows Public/Private Key Asymmetric crypto
    - In-Transit Crypto for more than just the web server
        - TLS/SSL is fine for the web server
            - Can be used for communication between app server & DB server
            - FTP
            - Email
            - LDAP
            - IM
        - SSH
            - to connect two machines securely
        - Aren't security certificates expensive?

            - Yes, unless you are your own Certificate Authority
            - You need:
                - OpenSSL
                - Place to securely store your keys
- ColdSpring for Security
    - Remote Proxy
    - Aspect-Oriented Programming
        - Take security out of objects and create a separate security object that acts as an interceptor and handles permissions-based access to each object
- Insecure Direct Object References
    - In this context, objects are
        - DB records
        - files
        - folders
        - other assets
    - Mitigating DOR Vulnerabilities
        - Encrypting URL and Form values
        - Access Control and Validation
        - Object Reference Map
        - Access Reference Map
- Web Application Firewalls
    - "An appliance, server plugin, or filter that applies a set of rules to an HTTP conversation" --OWASP
    - Inspects HTTP traffic looking for patters that match rules
    - Performs actions based on those rules
    - Inspects both inbound and outbound traffic
    - actions
        - log
        - allow
        - block
        - redirect
        - return status (i.e. 404)
        - proxy
        - execute
    - Uses
        - Protection from common attacks
            - OWASP's Top 10 (SQLi, XSS, XSRF (cross site request form))
        - Access control
        - Virtual software patches (0-day exploits)
        - Traffic logging (including http body logging)
        - Positive security modeling (whitelisting)
    - Types
        - Hardware appliance
            - exp, fast
        - Embedded
        - Reverse-proxy
        - Application level
        - Servlet level

## Holistic Program Quality and Technical Debt

- When you defer to do work in the future, you're borrowing time from the future that you may not have.
- The simplier that you make your application, the more secure it is
- When do you have a problem
    - Growing dislike for the system admitted by devs
    - Small bugs never fixed
    - Bad UI
    - Logs of TO and FIXME style comments
    - Code that can't be refactored
    - Code is too sloppy to make sense of
    - Poor variable naming
- Declaring bankruptcy
    - Rewrite.  Throw it all away and start over
    - Consider losing an hours work
    - You can write better software in less time, now that you know what the system does and how it works
- How avoid technical debt
    - Micro, easiest win: Make your code readable
        - Indentation
        - Commenting
        - Naming
    - Macro
        - Architecture
        - Planning
        - Loose Coupling
- Separate layers, keep them that way
    - Javascript in HTML = use jQuery
    - HTML in Javascript
    - CSS in Javascript
    - CSS in HTML
    - Javacript in CSS
    - Model-View-Controller
- Features Have Cost
    - Development time
    - Deployment
        - Always longer that you think
    - Maintenance
        - more features means more to maintain
        - more bloat
    - load cost
        - Downloading time, bandwidth, CPU & memory usage
- Use source control software
    - Delete commented out code, you don't need it and if you do it's in your source control software (right???)

- Aggressive code management
    - Watch what's going into your repo
- Document by automation
    - Functional tests
- Make debit visible
    - if you can write it on a whiteboard.....
- Conclusion
    - Problems of software quality can be overcome, it is long hard road
    - It's your job to write good software
    - Secure programming is good programming
    - Take calculated technical debt risks when you can, track the debt
    - It can take years to find the right approach
    - Watch for the warning signs
    - Make your code readable

## Building HTML5 Applications

- HTML5 Dive
    - Now it's just HTML
    - Combination of markup and JavaScript APIs
    - Put this at the top of your documents

```
<!DOCTYPE html>
```

    - Platform Support
        - WebKit
        - Safari 4x
        - Mobile Safari
        - Chrome 3x
        - Firefox 3.5x
        - Opera 9.5
        - IE 9
    - Tag is not always what it seems
        - Markup (e.g. header, footer)
        - API tags (canvas, video, audio)
    - Some new tags

```
<article>
<aside>
<command>
<details>
<summary>
<figure>
<figcaption>
<footer>
<header>
<hgroup>
<nav>
<progress>
```

```
<section>
<time>
<wbr>
```

- Can use “document.createElement(“<my tag name here>”) to “create” a tag, such as “<ie_sucks>”
- JavaScript Libraries to Help
  - Modernizer - www.modernizer.com
  - John Resig HTML5 Shiv
  - Remy Sharp HTML5 Shiv
- New Form Features
  - Attributes
    - Required
    - Autocomplete
    - Autofocus
    - Placeholder
    - Height & width
    - Min, max and step
    - Pattern
  - Types
    - tel = will display correct keyboard on mobile browsers
    - email = will display correct keyboard on mobile browsers
    - search
    - datetime / datetime-local / date / month / week / time
    - range
    - color
    - speech
- HTML5 API
  - Communications API
    - Can use console.log in JavaScript to write to the browsers console for trouble-shooting and development
    - Web Sockets (server port 81, secure port 815)
      - Say goodbye to pooling
      - Events
        - onopen
        - onmessage
        - onclose
    - Cross Document Messaging
      - Working across URL’s (origins)
      - Work between frames, tabs, and windows
      - Defines the postMessage API
        - Standard for sending messages
    - XMLHttpRequest Level 2
      - Eliminate server proxy for your mashups
        - The server you want to call has to have a server configuration as well
      - Browser support is half the battle
      - New Progress Events
        - loadstart

            - progress
            - abort
            - error
            - load
            - loadend
    - Geo Location
        - Finding out where you are
        - Users have to agree
        - Not GPS
        - Check
            ```
            if(navigator.geolocation(){// do something}
            ```
            - This is communicating with the BROWSER, the browser then communicates with the device GPS
        - getCurrentPosition() = one time
        - watchPosition() = continues to poll
    - Graphics
        - Canvas
            - 2D and 3D context
            - Draw lines, shapes, curves
            - Realize that you can script it and interact with it
            - Text can be rendered
            - Gradients
        - SVG
            - Scalable Vector Graphics
            - You love XML
    - Media
        - Can use this now
        - Can't secure your content
        - Fallback is Flash
        - Audio
            - Codecs
                - AAC, MPEG-3, Ogg Vorbis
            - Controls
                - load, play, pause, canPlayType
            - Read-only attir
                - Duration, paused
        - Video
            - Very similar to the audio
            - Codecs again?
            - Cool canvas effect with video
    - Offline
    - Selector API
    - Web Storage
    - WebWorkers
- Second Part - CSS3 Dive
    -
- Third Part - Build an application

## Database Performance Tuning

- Clustered Index
    - Leaf pages of index are data pages of table
        - Oracle calls this an "Index Organized Table"
        - MySQL can only cluster on primary key (InnoDB)
        - Can only have one clustered index per table
        - by default, MS SQL will create primary key index as clustered
- Query Optimizer (MS SQL)
    - Estimated Subtree cost should be in the hundredths (0.01, 0.02, etc)
    - Ctrl+L to open/refresh plan
        - Set IO
- Identifying a DB Performance Problem
    - Symptoms
        - "Current Requests" is at "Max Simultaneous Requests"
        - Slow/no response from CF
        - Low CPU usage on CF server
        - Usually, high CPU usage on DB server
        - All active requests waiting on queries in SeeFusion / FusionReactor
    - Finding
    - Resolving
        - WHERE clause
            - Find the most selective column or columns and apply index(es) there
            - Consider a Covering index
                - an index that contains all of the columns that you are interested in
            - Consider converting to Ad Hoc
                - or use SWL Server optimizer hint: `OPTION (RECOMPILE)`
            - Example, the "Last 5 minutes" query
        - Clustered index for ranges of data (dates!)
- Physical Performance
    - When query tuning just isn't enough
    - Indication: long disk queue, long IO waits
    - Separate (even multiple) physical device groups for:
        - table / clustered indes (RAID5/10)
        - transaction log (RAID1/10)
        - indexes
    - The more IO paths the better!
- Tips
    - Growing tables don't scale linearly
    - Global OR in WHERE clause
        - `firstName = 'Dave' OR lastName = 'Johnson'`
        - `UNION` two queries with one `WHERE` clause each
    - Joining too many tables in one query
        - Sometimes multiple smaller queries can provide better performance
        - Maybe put a `<cfloop>` within `<cfquery>`
    - Non-selective indexes just won't be used
        - Non-clustered index must *exclude* 90% of table rows to be useful

        - **Index columns *have* to be used in order left to right!**
            - Try reordering columns in index
    - Parameterized query / StoredProc not always better
    - Avoid Business Intelligence (OLAP) queries on Live (OLTP) databases
        - `<cftransaction isolation="read_uncommitted">`
- Conclusion
    - Indexes good!
    - Always tune before throwing hardware at the problem
    - Monitoring / proactive tuning
    - Everything is a trade-off
        - Each index slows down UPDATES
- Questions
    - no performance based on column order in SELECT
    - no performance based on ordering of WHERE clause (except in MySQL?)
    -

columns can be included (INCLUDE) in index w/o affecting index ordering
inner no diff
left outer makes diff
    condition in join is applied before join
    where is applied after

# Day 3

## CF911: Pinpointing and Resolving ColdFusion Performance Issues

- Has about 100 customers a year and the most of the problems are the same
- Many developers are focused on writing code and have little or no experience with CF server trouble-shooting
- Nearly all concepts will apply to any form of deployment (server, multiserver, WAR)
    - Also to any operating system (Windows, Linux) or CFML engine (Railo, OpenBD)
- Charlie Arehart
    - Over 300 blog entries, 80+ articles, 90+ presentations
    - cf411.com
    - cf911.com coming soon
- “Server down” may not be what you think
    - Has CF crashed been stopped, or is it merely unresponsive?
        - Big difference among these.
        - Solution is not to restart ColdFusion because the underlying root cause may not be resolved.
        - Only way to prevent this is to trouble-shoot while the problem is currently occurring
    - Are all request hung? Or are no request running at all? Are some?
        - Again, big difference. You need to use tools to see this.
        - Maybe only one application that’s having a problem on a server running multiple applications.
    - What’s the right solution? Change JVM settings? Admin settings? Tweak code?

        - Stop.  Take a breath
        - Find the right diagnostics, connect the dots, apply a solution.
            - Sometimes will find a blog saying “do this” or “do that” but don’t explain why or the implications of making that change.
            - It’s just like CSI, follow the evidence chain.
- Basic tools to consider
    - Task Manager / Processlist
        - Can be misleading
    - CF logs
        - Not generally as helpful
    - JRun/CF runtime/stdout/console logs/cfserver.log
        - There are often vital.  For some problems, first place to look
    - CFSTAT/perfmon
        - Useful option, not in multiple instances or J2EE deployment
        - This is a command-line tool
        - Looks at the running requests AND the requests in the queue
        - Note that there is a bug in CF8 that causes problems with perfmon but a hotfix corrects
    - JRun metrics
        - Can be setup in any version of CF
        - Tomcat approximate equivalent using jmxproxy, specifically:
    - PID (hotspot compiler) logs
        - Often missed (in same dir as jvm.config)
        - In ColdFusion/bin directory or JRun/bin directory
        - Would mention problems with compilation or memory errors
    - Web server logs (IIS, Apache)
        - Can show you what requests are running, how long, how often.
        - Also maybe IP address, user agent, which can be important
        - Can show cookies sent during session
        - On CF411 has some links to log tools, one by Microsoft is very helpful (sorry I missed the name)
    - CF monitoring tools
        - Free and commerical products
    - JVM monitoring tools
        - Not always easy to use, especially for CF runnign as service
- Common Problems
    - Out of memory?  What kind?
        - Heap? Permgen? out of swap space? Unable to create new native thread?
            - Each very different, different solutions
            - What they mean, how to determine, how to resolve
            - A "out of memory" isn't always a heap space memory problem, could be another area.
                - Modifying the heap without considering which one is the problem could only make the problem much worse
                - Look for message right before "out of memory" error
        - Out of swap space often means something else was using the memory, such as a database server running on the same server
            - Need to know what was going on when the crash occurred
            - SQL Server can be configured to use a certain amount of memory and by default

it's set at unlimited.
        - Best to move SQL Server off to another box
        - Sometimes the simple answer is to actually lower the stack space
    - Why “high memory” is not necessarily a problem
        - How JVM may just be lazy, and a forced garbage collection may recover no-longer-”used” memory
        - JVM 1.5 had an “ergonomics” change that allowed it to be lazy in garbage collection.
            - JVM 1.4 and 1.6 are better at this
            - Can manually force garbage collection using the CF Administrator
        - Don’t panic just because the memory is climbing
    - What can be holding memory, if GC does not resolve things?
        - RAM drive
        - Cached queries
            - If I use these, what is the timeout value?  During this period the memory cannot be freed or garbage collected
        - Sessions, might not think we have that many user sessions but hackers, spiders and bots can consume this memory
            - Also automated tools can be consuming sessions
        - Scopes 'application' and 'server' is another area
        - Any requests that are currently running
        - var scope
    - CPU spinning wildly?
        - What might be root cause?
        - Could be CF during garbage collection, but maybe not the root cause, could be running out of memory because more memory is locked, etc, etc
- Understanding things better
    - How to see count (without details about each)
        - CFStat, JRun metrics, JMXProxy on Tomcat, or CF monitors
    - Are *all* requests hung?
        - They’re likely all waiting for something (long running `cfquery`, `cfhttp`, `cflock`, etc.)
    - Are only *some* requests hung?
        - Some could be getting through while others are hung
        - Means problem is only with some applications/pages/sites
    - Are requests not timing out when expected?  Why not?
        - Are you trying to kill requests?  Why don’t they die?
        - Could be something outside of your CF server if the code is calling out to someone else
        - Charlie says that 80% of his traffic on his site are spiders and bots
            - This might be hitting a page that’s calling out and causing the problem.
- Tools to view running processes
    - CF Monitor
        - Running requests
        - Buttons at the top on starting monitoring, profiling and memory tracking
            - Each uses more resources, monitoring, profiling and memory (in that order)
                - Memory tracking can easily consume too much memory, be

careful!
        - If you turn these on and leave CF they are still running!
    - Fusion Reactor
        - Running requests
        - Logs every request, every query and every 5 seconds the stack
        - Sometimes you're being killed by thousands of requests not just one or two
            - "Death by a thousand cuts"
    - SeeFusion
        - Server -> Active Requests
    - Tools to view history of past requests
        - Monitors, alerts, logs
    - Tools to stack trace running requests
        - Live or via email
- Areas of concern that people often miss
    - Applyig hotfixes, cumulative hotfixes
    - Updating to free CF updaters
    - JVM update: why, how
        - JVM classloading bug affecting CF 8, 8.0.1
    - Impact of client variables, even if you "dont' use them"
- Consdiering settings for simulaneous requests
- MaxWorkerThread default setting of 25 in CF9 & IIS
    - This is a problem by default, Google this and find the blog entry for the fix
- File uploading memory bug and fix CF 7/70.1/7.0.2 (no fix for 6.x)
- Why CF may sometimes nto stop when you requst it
    - why sometimes when you restart it, a flood of requests coudl hurt
- Didn't talk about these because they aren't always the solution, should not be the first thing to consider.
    - VM tuning
    - Code review/tuning
    - SQL tuning
    - DB tuning
    - Caching
    - Hardware/Architecture, clustering, load balancing
    - Switching CFML engines
- Where to learn more?
    - There are many resources, mine and others
        - And I have many which poitn to those others
    - cf911.com - wiki of troubleshooting tools / resources
    - carehart.org

## Application Intrusion, Detection and Tracking

- Intrusion = access to an application with malicious intent
- Detection = using one or many tools to locate and detection application level access that has malicious intent (not stopping but seeing they're in there)
- Tracking = using tracking tools to uncover the traffic path of a malicious user (this is the most difficult part)
- Out of scope for talk

    - Network level intrusion
    - Desktop or device compromise (such as customer support desktop)
    - Worms / Viruses
- Anyone with enough tie and motivate will get in
    - Anyone with a site that has a login screen has probably been compromised at least once and don't know it
- Intrusions are usually for monetary gain or to make a statement (Look what i can do / did)
- Sony Playstation Network was massively hacked
- Want a 100% guaranteed way to prevent an instruction on your system?
    - Don't connect it to anything, including power.
    - As soon as the system powers on, someone will try to get into it
- When dealing with an intrusion...
    - It is not 'How can I stop them?'
    - It is "How do I know what they did?"
- Basic intrusion
    - Intruder gains rights to system by impersonating someone with legitimate reasons for access
    - Intruder finds unsecured pop-up screen (programmer made security assumption, bad)
    - Intruder writes program to make thousands of calls to the system
    - Intruder spreads intrusion attack over multiple weeks only attacking the system for 10 minutes at a time
    - They try to make the traffic look normal, trying to hide and look like a real person
- Intrusion Tracking
    - Very application specific on what it's protecting and what it's tracking (and how)
    - You can't add tracking after the compromise, this should be part of design from day one
    - Most systems don't track enough data, or don't track anything
    - Most basic tracking down is not reliable
        - Web server / network log
    - Just because they are logged in does not mean they have access
    - No one ring to rule them all
        - It takes multiple tools to track intruders
- What to look for .. the basics
    - Traffic that doesn't follow site map
        - If they have to go to page1, then open pop-up1 but pop-up1 is being accessed without page1, probably intruder
    - If page views that happen faster than an person could actually do them
    - Abnormally large traffic days
    - Unknown or wrong browser / OS combinations
    - Invalid entry points (on page that's three deep without going to those three pages)
- What will stop an attack?
    - HTTPS will not stop an attacker.  It will just encrypt their instruction attempts point
        - Prevent man in the middle
    - Well written code
        - Any data coming form the client should be treated as bad until validated
            - Use `cfqueryparam`
    - Programmers that can think like an attacker
    - Web application firewalls will not stop legitimate traffic
        - Some examples include:
            - SecureIIS

            - ModSecurity
            - FuseGuard
- Look at code
    - Application with excess JavaScript
        - Will look at the code on your site, perhaps your JavaScript library files
        - Find Media Player function
        - Look for functions that may include variables
        - Find a path and now can use URL hack and navigate the file system to get files
    - Login screen (insecure direct object reference)
        - Perhaps restricted downloads
        - The page requires a login, but the list has incremental ID's for the files to download
        - Use URL hack to select ID's that aren't displayed on the page
        - Once again, can now download assets that weren't displayed but by using the URL could access them.
        - Security was only on initial login screen, but no security on the actual downloader.cfm file to make sure the person can retreive the file
        - Can alter the header referer and user-agent when making the requests as the cfloop runs
        - Solutions
            - Add a session variable to store what they actually should be seeing
            - Downloader would check to see if the session variable is available
                - If it is available, is the file being requested inside that variable list?
    - Track http calls in AIR-based application
- Development Tips
    - Don't load excess JavaScript
        - Perhaps breakup Javascript into multiple files and only load the files you need for that area of the application OR the access rights that user has
    - Don't give clues to hidden functionality
    - Instead if hiding a DIV or element don't render them
        - Remove them server side
    - If the site requires a login don't expose functionality that only works post login
        - Only load on that page what's required to login to the application
    - Hidden form fields are not "hidden" to view source visibility or Firebug
    - Remove legacy code
        - Don't just comment it out!
        - You could use ColdFusion comments since it's not rendered to the browser, but better just to remove it
        - Javascript function calls that are just commented out could be displayed to the intruder
    - Debugging tools can be used with a malicious intent
        - Firebug
        - Selenium
        - Charles
            - Will not display traffic to localhost, FYI
- Use the right amount of security for what you are protecting, but don't go overboard. You have to find the balance between the effort and the value.
    - If you have an image on your application that could cost you $1,000,000 if compromised, you should be putting $1,000.000 into protecting and securing that resource.
    - Make sure the boss/manager/supervisor is the one to make the decision to limit any security

that you'd put in place, not you.
- Whitebox test is where you give a security person a copy of your code and they try to compromise it.
    - Penetration testing is also a good thing to do on a regular basis

## Running Multiple CFML Engines on Apache Tomcat

- What is a CFML engine?
    - Are all Java Web Applications under the hood, all CFM and CFC code is handled by a servlet.
    - Tomcat doesn't care if your using CFML or anything else
- Why run CFML on Tomcat?
    - Flexable development environment, can run more then one engine and version on a single system for testing
    - Extremely fast, small testing by Matt saw Tomcat handle 3x what JRun could
        - JRun hasn't been updated in four years
    - JRun will be replaced by Tomcat in ColdFusion X/10
- What is Tomcat?
    - Servlet container & HTTP server together
    - It can be your web server if you don't need Apache
    - Handles all of the plumbing for Java web applications
- Why Tomcat?
    - Other options, Jetty, JBoss, etc
    - Tomcat was the reference implementation of the spec, been around longer
    - Seems to be one with the most documentation
- The setup
    - Firefox -> http -> Apache -> http | ajp | mod_jk -> Java web apps -> Railo | OpenBD | ColdFusion
- Demo time
    - Tomcat installation
        - Downloaded Tomcat 7.0.12
        - Unzip
        - Copy to home directory
        - Start Tomcat /tomcat/bin/startup.sh
        - http://localhost:8080
    - Tomcat configuration
        - /tomcat/webapps
            - All web applications go in here
        - /ROOT
            - Doesn't have a context path
            - http://localhost:8080 -> webapps/ROOT
        - Optimize the memory settings as the default can be low for CFML
            - nano /tomcat/bin/setenv.sh
                - export JAVA_HOME=/opt/java/jdk1.6.0_24
                - export CATALINA_OPTS="-Xms1024m -Xmx2048m -XX:MaxPermSize=512m"
            - Restart Tomcat (ps -wef | grep tomcat", kill -9 xxxx)
    - WAR = bundled web application archive (just a ZIP file)
        - Railo and OpenBD provides these and you drop them in

        - During CF install you select WAR file and it generates the file that you can drop in
        - Copy these into the /webapp directory and it deploys this for you (expands them)
            - Filename is context (minus extension)
        - Go to http://localhost:8080/context
        - Go to http://localhost:8080/cf9/CFIDE/administrator/index.cfm
        - Note: If you remove the WAR file, Tomcat thinks you've undeployed the application and will remove the directory
    - Apache configuration
        - Edit hosts
            - 127.0.0.1 cf9.local
            - 127.0.0.1 railo.local
        - Enable name based virtual hosting

```
<VirtualHost *:80>
      servername cf9.local
      ProxyRequests off
      ProxyPreserveHost on
      <Proxy *>
            Order deny, allow
            Allow from 127.0.0.1
      </Proxy>
      ProxyPass / ajp://cf9.local:8009/
      ProxyPassReverse / agp://cf9.local:8009/
</VirtualHost>
```

    - Tomcat configuration
        - nano /tomcat/conf/server.xml
            - Add a new host block

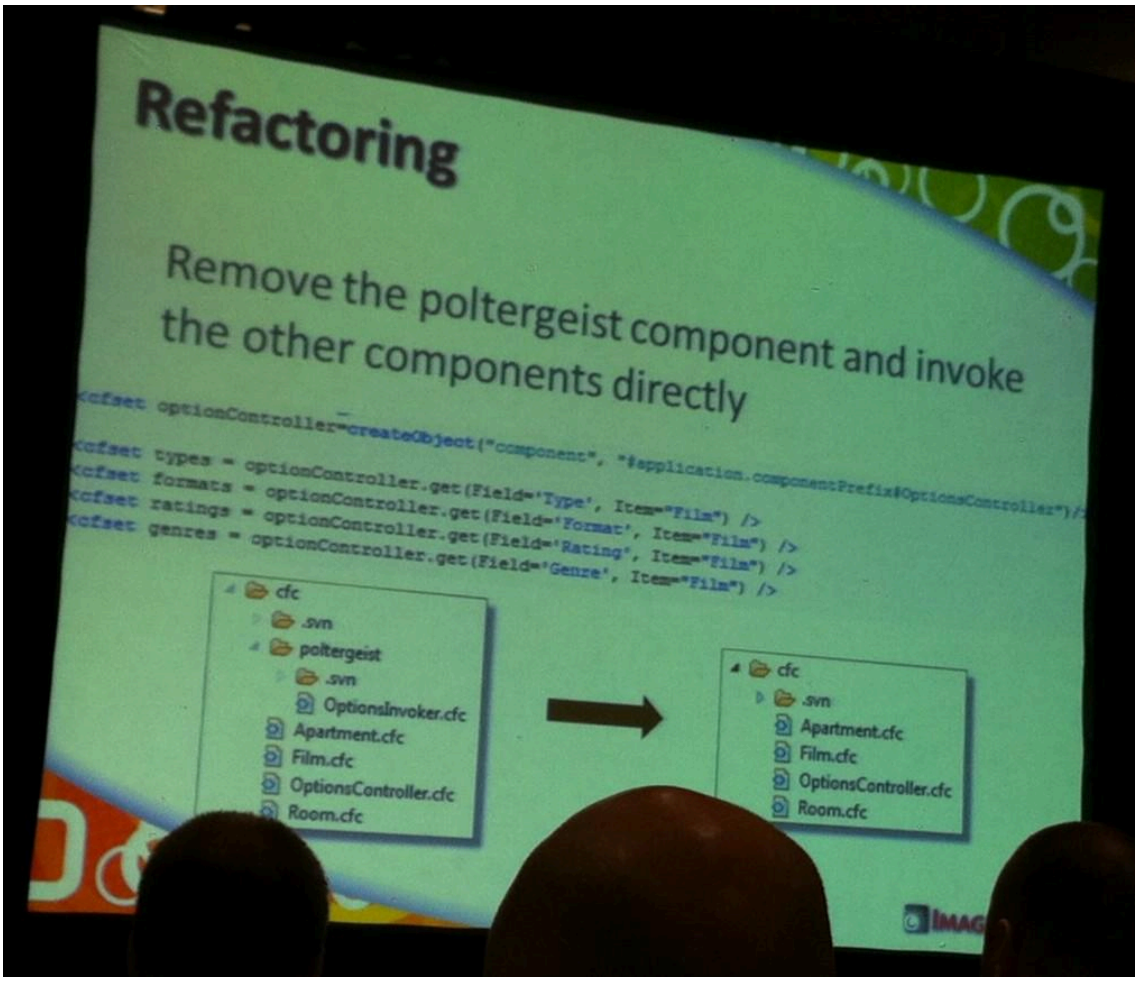
```
<Host name="cf9.local">
      <Context path=""
      docBase="/home/mwordward/tomcat2/webapps/cf9" />
</Host>
```

        - Restart Tomcat
- ajp protocol is binary and more efficient, performance is better but uses more memory and allows each instance to be isolated and handle the request
- Good to learn Tomcat now as it'll be part of the next version of Tomcat and allows for multiple instances

## Getting to Know AntiPatterns

- AntiPatterns = not just a crappy solution
  "The essence of an AntiPattern is two solutions... The first solution is problematic, the second solution is called the refactored solution."
- Benefits of AntiPatterns
    - Help developers recognize problems
    - Provide common vocabulary
    - Provide proven solution
    - Save existing projects
- Contrasting with Design Patterns
    - Design Patterns assume from-scratch programming

    - AntiPatterns start with a problematic solution and refactor
    - Design Patterns become AntiPatterns in the wrong context
- Refactoring
  The process of changing a software system in such a way that it does not alter the external behavior of the code, yet improves its general structure.
    - Remember:
        - Do it iteratively
        - Live scaffolding
        - Test as you go
- Functional Decomposition
    - One main routine calling many subroutines
    - Individual subroutines become separate components
    - Code base looks like an episode of Hoarders
    - Large Domain Model
- God Object
    - A component that knows or does *everything*
- What's changed?
    - Methods and variables grouped more intuitively
    - Each component only knows about *itself*
    - Maintenance does not affect the entire system
    - Components communicate directly
- Anemic Domain Model
    - All logic implemented in a service layer
    - Domain Objects are "Bags of Getters and Setters"
- Components created with `cfinvoke` are stateless and have to be recreated each time
    - Forces Poltergeist symptoms on all components
    - Refactor code:




- Many More AntiPatterns
    - Boat Anchor
    - Circular dependency
- Take Aways
    - AntiPatterns do not only describe bad code
    - AntiPatterns are powerful tools
    - Mostly stem from misunderstandings and lack of communication

## Time Management for Developers

- If you continue to work at a hectic pace, it will have a negative affect on your life.
- What causes developers' time famine?
    - Interruptions = heavy cost
    - Multitasking = causes us to perform less then optimally
    - Communication Demands = less time focused on purely programming
    - Disjointed Work flow = instead of working within 1 IDE, split over multiple applications or at times multiple systems
    - Changing Priorities = start day working on most important task, manager arrives later and reorganizes the tasks and now that task is the least important but we don't know that
    - Information Overload = seeing too much in IDE or having to remember too much
- Developers' Time

- Useful Time
  - 18% - Reading code
  - 16% - Editing code
  - 10% - testing code
  - 5% - reading docs
  - 2% - Reading tasks
- Wasted Time
  - 22% - Handling Interruptions
  - 13% - Navigating Code
  - 10% - Searching code
  - 4% - switching apps

- Friction Points
  - Problem: Many Friction Points
    Solution: Eliminate Friction Points
  - Integration
    - 1980s - Paper ToDo Lists
    - 1990s - Digital ToDo Lists
    - 2011 - Software ToDo Lists
      - Bugzilla
      - Email Inbox
      - JIRA (http://www.atlassian.com/software/jira/)
    - Scattered Task List
      - Problem: Having to monitor Bugzilla, JIRA and Outlook for different tasks in different places
      - Solution: One Task List
        - One place to look for the next thing you're going to do
        - Mylyn
          - Free, open source
          - Connects to JIRA, Bugzilla, Trac, Mantis, CollabNet
        - Tasktop
          - Commercial add-on focused on Enterprise tools
          - Connects to Microsoft TFS Accept, ClearQuest, Polarion, JIRA, Bugzilla, Trac, Mantis, CollabNet, ThoughtWorks, IBM RTC
        - Potentially use Mylyn & Tasktop together?
    - Internet Wait
      - Problem: Cloud-based task list introduces latency and Internet-connection dependency
        - Can add up to ~5 minutes per day
      - Solution: (Local?) Task Editor
    - Tracking Changes
      - Problem: How do you tell which task(s) have changed?
        - Subscription to "bug mail" just clutters inbox
      - Solutions:
        - Incomings
        - Task Editor Incomings
  - Productivity
    - 2011 - GUI
    - Multitasking

                - Solution:  One-Click Multitasking with Mylyn
            - Sharing Knowledge
                - Solution:  Sharing Context
            - TPS Reports
                - Solution:  Automatic Commit Comments with Tasktop

## SQL Performance for the Common Developer

- Common SQL Performance Issues
    - Slow Load
        - Reporting
        - ETL (Extract-Transford-Load)
    - Slashdotted
        - Server overload
    - Query Blocking
    - Host System hardware
- Remember about your SQL Server
    - It's an application server
    - requires proper config
        - defaults are poor choices
        - high availability settings not necessarily for small sites
        - if using mysql, learn the different engine options
        - if use MS Sql, size of DB will determine options
            - enterprise vs std, online indexing
        - has it's own development language
            - t-sql
            - ddl
            - etc
    - what can cuase perf issues with your sql server
        - bad/incorrect config
        - file io
        - server overload from other processes
        - not enough resources
        - poorly written queries
- how to run sql explanitaion
    - mysql
        - append EXPLAIN to any query
    - ms sql server
        - usage SET SHOWPLAN_[ALL | TEXT | XML] ON
- Terms
    - Table Scan
        - searches table row-by-row for data requested
            - slowest
            - pulls data from table row
    - Index Search
        - searches index row-by row for

    - Index Seek
        - fastest
- Thinking like a SQL Developer
    - JOINs vs sub-SELECTs
- Transaction Isolation Levels
    - SERIALIZABLE
    - REPEATABLE READS
        - Locks data during transaction, both Read & Write
    - READ COMMITTED
        - Restricted to Committed Data
    - READ UNCOMMITTED
        - "Dirty" Read
        - least restrictive, lowest isolation
        - transaction can "see" not-yet-committed changes
- Query Design Tips
    - Breaking down queries into smaller, more efficient queries
        - Bonus:  easier to debug
    - Summary Tables for Big Data reference
    - Using temporary tables to process large data vs complex joins
        - Takes data out of immediate production and into memory
    - Add Statistics to your Table rows, instead of querying for Details
        - In a Categories table, add NumItems, instead of getting "real-time" data
- Data Search Example
    - Store email addresses in reverse for quick lookups
        - One column EmailAddress for display
        - One column RevEmailAddress for quick lookups
            - How many times would these appear in the Reverse Email Address?
            - "moc.liamg@..."
            - “moc.liamtoh@...”
            - moc.loa@...”
        - Reverse helps the index sort
        - Searching LIKE '[email _address]%' uses the index
- CFQuery Tips
    - Reduce calls to database
        - Multiple queries vs qry of qry
        - Timeouts are not respected
        - Cache common data
    - COUNT on the primary key or another indexed column, rather than *