Introduction to Maven

Tim's Notes on Class:

- I'd like for this class to be focused on activity rather than theory. There's nothing worse than standing up in front of a class and having to talk for an hour about the theory of dependency management.
- There's no Eclipse here. I did that on purpose. I think we can teach the content with any editor or IDE, and I think we should keep the initial content somewhat flexible. Many don't use Eclipse, I'm noticing a trend toward IntelliJ. Also there's a large population using Rational tools, which, while they use the same Maven plugins present frustrating differences. If we're keeping this confined to five hours, I don't think we should lose 30 minutes to IDE integration - let's focus on POM.xml
- There's going to be a lot that isn't covered in this class, but this is alright. From my perspective, this class is for users. The point of this class is to get people to a point where they can look at a POM and begin to understand it.

Prerequisites

- Understand how to set an environment variable on an OS.
- Know what the command line is and are comfortable with it
- Have some experience with Java

Target Platforms

Operation Systems:

- Windows 7 +
- OSX Mountain Lion +

Java Version:

- Java 7

IDEs:

- Eclipse Kepler

Goals and Objectives

At the end of this class, students should be familiar with the following:

- How to download and install Maven
- How to run a simple build
- How to configure and locate dependencies for a project
- How to configure the Surefire plugin to run unit tests

Modules

Welcome and Objectives (10 minutes)

- Welcome Message
- Instructor introduction
- Brief overview of course objectives
- Planned class structure (we will learn this, then this, then this…)
- How to provide feedback during class

Length:

```
Min: 5 minutes
Max: 10 minutes
```

Tim Notes:

- Don't dwell. This section needs to start and end in a manner of minutes.
- You might see this as a filler section, but it's important to provide a "roadmap" for the course.
- We should make sure to define what the instructor introduction is (and what it is not)
- Giving people a way to provide feedback during class is critical for virtual.

Install and Run Maven (15 minutes)

- Downloading Maven
- Installing Maven
- Running Maven
- Questions

Length:

```
Min: 10 minutes
Max: 15 minutes
```

Tim Notes:

- Instead of starting with a VERY BORING "What is Maven?" section. We start with a demo.
- Pay attention to prerequisites - if someone doesn't get how to set an environment, they shouldn't be in this class.

Build a Simple Project (35 minutes)

- Checkout a Simple Project from Version Control
- Examine the pom.xml
- Introduce coordinates
- Introduce dependencies
- Talk about dependency scope
- Explain directory structure

- Modify code
- Modify a dependency
- Examine the Output

Length:

```
Min: 25 minutes
Max: 35 minutes
```

Tim Notes:

- Don't go into all the detail. This class is not a reference book.
- Don't talk about where resource filters go, they are not there yet. src/main/java, src/test/java - that's it.
- You'll note that we introduce dependency scope out of order here. This is fine, we're hinting at the next section.

BREAK (10 minutes)

Working with Dependencies (45 minutes)

- Coordinate Details (5 minutes)
- Where do dependencies come from?
- Checkout a Project from Version Control
- Run a Build and Identify a New Requirement
- Locate the Appropriate Dependency in Central
- Add a new Dependency
- Modify Code to Use New Dependency
- Demonstrate Maven Downloading a Dependency
- Where dependencies go?
- Repository Format
- Release Versions
- What if your dependency isn't on Central?

Length:

```
Min: 35 min
Max: 45 min
```

Tim Note:

- This is going to seem strange, but there's no mention of a snapshot here. Here's my thought: most users don't need to know what a snapshot is until they think of in-house software releases. Almost no one is depending on SNAPSHOT versions from Apache. Introducing SNAPSHOTs right now is confusing.
- Also note that this module isn't called 'Dependency Management'. That's a fancy name that means nothing to a junior developer that just needs to use a new dependency.

Writing and Executing Unit Tests (45 minutes)

- Checkout a Project from Version Control
- Demonstrate Surefire Plugin
- Introduce Surefire Plugin
- Show TestNG Tests
- Execute TestNG Tests
- Discuss Test Resources
- Demonstrate Test Resources
- Including and Excluding Tests
- Running a Single Test
- Parallel Test Execution
- Surefire Plugin Configuration Examples

Length:

```
Min: 35 min
```

```
Max: 45 min
```

Tim Notes:

- Unit tests are such a fundamental requirement, we should spend an entire module on it. ##

BREAK (15 minutes)

Building Your Software (30 minutes)

- When you run a Maven build...
- Maven just runs a series of Goals (2 slides - QUICK!)
- These goals are contained in Plugins (2 slides - QUICK!)
- Goals are executed within a Lifecycle (2 slides - QUICK)
- Let's review: Goals are units of work, Plugins group similar goals, and Maven has a set lifecycle to which goals are bound.
- So when you run your build, this is what's happening (show default lifecycle)
- Demonstrate the lifecycle
- Hit them with a bit of theory… All maven projects follow this lifecycle, it is configurable, but this is why we standardized.
- Talk (briefly) about convention over configuration
- Talk (briefly) about what life is like without Maven

Tim Notes:

- Notice how this section is a bit more theory than the others. It is also smaller for a reason.
- There's a bit of propaganda here. We're talking about benefits, but we're putting it in the middle of the class after the students have seen Maven's strong points.
- You can talk for an hour on goals, plugins, and the lifecycle, but you really shouldn't. This section is the stuff that everyone touch Maven should know, but never does.

Packaging Your Software (45 minutes)

This module does two things. It focuses on the WAR plugin and it uses a simple Assembly:

- Remember the lifecycle form the previous section, well the packaging changes it...
- Checkout a project from source control, a WAR project
- Demonstrate the WAR build
- Inspect the WAR output
- Discuss the WAR Plugin
- Hit upon some customization points for the WAR plugin
- Run the WAR in Jetty or Tomcat using Maven

- Switch gears to a Standalone Java program

- Demonstrate (but don't fully delve into) the Assembly plugin
- Build a program that creates a jar with dependencies
- For completeness talk about the JAR plugin and just mention the EAR plugin

- Introduce SNAPSHOT versions

Tim Notes:

- Don't fully explain the Assembly plugin. Leave that for either the advanced class or for consulting.
- Get out of the mindset that you have to read the students everything about these plugins. Really this is just a starting point, 5 hours isn't enough to tell them everything about the WAR plugin.

BREAK (15 minutes)

Creating a Multi-module Project (45 minutes)

- Checkout a multi-module project from source control
- Show the file system structure with a top-level directory
- Show the pom.xml files with relationships
- Why would you do this? Why group projects together?
- Define Inheritance and discuss what is inherited
- Define Aggregation and discuss what is aggregated
- Talk about how version numbers are often aligned in a multi-module project
- Demonstrate a Multi-module Build
- Discuss the Reactor and how it orders builds
- Checkout a big multi-module project from source control
- Demonstrate multiple levels of hierarchy
- Build a big project that has many pieces and deploy it to an application server...

Tim Notes:

- You can't leave an intro Maven class without having some exposure to multi-module projects.
- That last bullet item. It is audacious, but I feel like the class should build up to something big, right? Like at the end of the Maven intro class, a pretty basic class. We end up standing up an application server and deploying and application to it.

Summary and Goodbye (10 minutes)

- Hey, thanks for coming. I hope you had a great time.
- We have another class that opens up the advanced parts of Maven, here's a link to sign-up with a 15% off discount code.
- If you have any questions, we're always available to help. Here, if you want free help… here's the number for our sales team.