



3 LESSONS MICROSOFT TAUGHT US ABOUT BRANDING

WHEN DEEP POCKETS DON'T MEAN A THING

BY DAVID BRIER — DBRIER@DBDINTL.COM

For years now, Apple with its *I'm a Mac. I'm a PC.* campaign has essentially established Microsoft's marketing position in the minds of consumers. In actual fact, Apple has positioned the entire PC world, but Microsoft, being synonymous with PCs, has become the greatest victim in that campaign's wake.

Most everyone seems to enjoy Apple's ads. The casting is brilliant, the ads are entertaining and the messages hit any sore points about Windows from Vista to tech support, and indeed, these ads have become culturally iconic.

THE WRONG THING TO DO

So what has Microsoft done over the years? From a branding standpoint, almost nothing.

They recently hired the super-hot agency Crispin Porter for a reputed \$300 million+ ad campaign. The first ad used Jerry Seinfeld with Bill Gates in what appeared to be an attempt at humanizing Mr. Gates and Microsoft. Ad critics grimaced. This ad was launched with the tag line *Life Without Walls* which became a punch line for Mac enthusiasts and beyond. Mac-loyalist blogs commented, "In a life without walls, who needs Windows?" Ouch.

THE WRONGER THING TO DO

Then, Microsoft delivered a series of ads where the position they were trying to

dislodge made up about 90% of its commercial copy lines. The *I'm a PC.* campaign was created with very loose, amateur-styled video techniques, again to humanize. The obvious goal was "How do we become cool and relevant?" Only problem is that it directly played into Apple's campaign. It's impossible to see one of those ads and not think of Apple. I could understand their thinking, but they were bringing nothing new to the table. It was all defense, with no strategic offense.

Even now, the Microsoft stores are being compared to the Apple stores.

WHAT HAVE WE LEARNED?

So, if the deep-pocketed Microsoft machine can make these missteps, is there anything we can learn from this so we can spend (waste) fewer marketing dollars in the marketplace to promote our brands and our own businesses?

Yes. In three simple steps.

THE THREE BRANDING LESSONS MICROSOFT TAUGHT US

1. Don't try to be something you're not. Pick your sweet spot and embrace it. Don't try to simply follow the lead of others because (even if you're Microsoft) if you're following, you're not leading. Just look at Zune (and its lackluster market share) as a case study.

What to do: Don't fake it. Elaine on Seinfeld once told Jerry that she'd "faked it." Totally shocked, Jerry asked, how many times? Her response was, "every time." Jerry compared Elaine to Meryl Streep for her incredible acting skills. When it comes to your brand, be real. Don't try to fake it. Find something you can get passionate about and something your brand can do remarkably well.

2. To do nothing is branding death. Saying and doing nothing or too little leaves your customers to look elsewhere to get the facts (or any ideas if facts don't exist). They'll take whatever information they can find unless better, more thought-provoking information comes along to supplant it.

If you don't like your fate being dictated at random, you had better speak up. Then improve what you say. Then increase how many people hear it. As the business guru Peter Drucker said, "You can't shrink your way to greatness."

CONTINUED ON PAGE 4

2010

BY JOHN W. STOUT - JOHN@STOUTSYSTEMS.COM

It's been said that all good things come to those who wait, but we can all attest that 2009 proved that was not true. Good things happen to those who make them happen. The smartest advice we incorporated at Stout was not to wait for things to get better, but to always try to make them better. Have a Happy & Prosperous 2010!

NOTICES

STOUT SYSTEMS WELCOMES NEW EMPLOYEES

BILL HEITZEG, GURPAL BASSALI,

TOM HARKAWAY AND LISA POLLARD

To subscribe: email news@stoutsystems.com and enclose your email or mailing address.

Job openings can be found at <http://stoutsystems.com/openings/>.

Copyright © 2010 Stout Systems Development, Inc. All Rights Reserved. STOUT and FUELED BY THE MOST POWERFUL TECHNOLOGY AVAILABLE: HUMAN INTELLIGENCE are registered trademarks of Stout Systems Development, Inc. All corporate, product, application, and publication named in this newsletter are trademarks or registered trademarks of their respective companies or owners. The user assumes the entire risk as to the accuracy and the use of this document. URLs to sites other than www.stoutsystems.com are provided for your convenience. Stout Systems Development does not provide, and is not responsible for, the content users may find on such sites.



Fueled by the most powerful technology available:

Human Intelligence®

IMPROVING JAVA SOFTWARE BUILDS WITH MAVEN – PART 2

By ROBERT SHANAHAN – [RSHANAHAN@GMAIL.COM](mailto:rshanahan@gmail.com)

This is a two-part article. Part 1 is in the Fall 2009 issue available online at <http://stoutsystems.com/newsletters/2009-fall/>.

DEPENDENCY MANAGEMENT

SCOPE

Dependency scope is used to limit the transitivity of a dependency, and also to affect the classpath used for various build tasks. There are 5 scopes available:

Compile - this is the default scope, used if none is specified. Compile dependencies are available in all classpaths.

Provided - this is much like compile, but indicates you expect the JDK or a container to provide it. It is only available on the compilation classpath, and is not transitive.

Runtime - this scope indicates that the dependency is not required for compilation, but is for execution. It is in the runtime and test classpaths, but not the compile classpath.

Test - this scope indicates that the dependency is not required for normal use of the application, and is only available for the test compilation and execution phases.

System - this scope is similar to provided except that you have to provide the JAR which contains it explicitly. The artifact is always available and is not looked up in a repository.

EAR WITH WAR(S)

With respect to JEE ear packaging, Maven dependency management is sorely lacking. There is no clear cut way to tell Maven that a webapp dependency is provided by its containing ear. Using “provided” scope results in no manifest entry in the webapp's META-INF/manifest.mf file. Configuring a dependency as `<optional>true</optional>` will prevent a dependency from being included in the webapp's WEB-INF/lib directory and it will include it in the manifest. So, there you have it, right? Wrong! Unfortunately the optional configuration doesn't prevent inclusion of all of the transitive dependencies of an optional dependency, which will end up in the WEB-INF/lib directory of the webapp's war file. Using the optional config option doesn't feel right anyway. The correct

solution is to define a new scope called “ear” that would honor the JEE ear packaging specification. We can hope Maven 2.1 will plug this blatant hole. In the mean time, I present an ugly workaround:

(See the code at <http://stoutsystems.com/newsletters/2010-winter/>)

The ugliest aspect of this workaround is that if you specify a classpathPrefix (common to put 3rd party jars in lib directory of ear) for the manifest configuration, the prefix is also added to jars that will be packaged with the war in WEB-INF/lib. Fortunately the classloader correctly resolves all jars stored in WEB-INF/lib regardless of war manifest specification.

MULTI-MODULE PROJECTS

Maven supports multi-module and hierarchical projects. The most common example of a multi-module project is a JEE project with an ear file containing zero or more ejbs, one or more wars and zero or more utility jars.

A typical ear structure:

```
ear
+lib
  utility1.jar
  utility2.jar
  3rdparty1.jar
  3rdparty2.jar
+war1
+WEB-INF
+lib
  web1_1.jar
  web1_2.jar
+war2
+WEB-INF
+lib
  web2_1.jar
  web2_2.jar
ejb1.jar
ejb2.jar
```

A corresponding multi-module Maven project structure:

```
JEE Project
ear
web1
web2
ejb1
ejb2
utility
```

Such a project has a top level pom.xml, which serves as the parent to each of the sub-module poms. The parent pom must explicitly identify each sub-module.

WARNING: *The following article crosses the threshold of common English into geek speak. This article was written by, and for, software engineers.*

```
<modules>
  <module>utility</module>
  <module>ejb1</module>
  <module>ejb2</module>
  <module>web1</module>
  <module>web2</module>
  <module>ear</module>
</modules>
```

“DEPENDENCYMANAGEMENT” TAG

With few exceptions, each sub-module will depend on the same version of a shared dependency. To this end, Maven supports centralized management of dependency versions with the dependencyManagement tag, which simply specifies the version of each common/shared dependency. A top level pom:

```
<dependencyManagement>
<dependencies>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring</artifactId>
  <version>2.5.5</version>
</dependency>
<dependencies>
</dependencyManagement>
```

For dependencies specified in the parent pom, a sub-module needn't specify version, which obviously simplifies the task of changing dependency versions.

```
<dependencies>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring</artifactId>
</dependency>
<dependencies>
```

DEPLOYMENT

With the basic dependency management in place, the next challenge is getting jars to deploy to the appropriate directory within the ear. Ejb modules, according to JEE convention (and our defined ear layout), go in the root directory of the ear and require an entry in the

ear/META-INF/application.xml file. Wars likewise reside in the ear root directory and have entries in application.xml.

ear module pom.xml excerpt:

(See the code at <http://stoutsystems.com/newsletters/2010-winter/>)

We discussed previously a hack-around to configure war dependencies provided by an ear (Ear with War(s)). It turns out there is a much simpler technique; however the technique requires that the target app server honors a war/META-INF/manifest.mf classpath specification that consists of only a directory name.

The war pom.xml excerpt:

(See the code at <http://stoutsystems.com/newsletters/2010-winter/>)

JBoss 4.2.x and WebSphere 6.1.x both support this technique. Geronimo 2.x does not.

MAVEN AND ANT


Maven supports Ant integration via the AntRun plugin. The intent of this plugin is to facilitate migration from Ant to Maven and Ant tasks should only be used where Maven functionality is yet lacking. One such example is the Ant sshexec task, which has no Maven equivalent. The following will deploy (scp) an ear a remote host.

(See the code at <http://stoutsystems.com/newsletters/2010-winter/>)

Alternatively, use existing build.xml:

(See the code at <http://stoutsystems.com/newsletters/2010-winter/>)

Robert Shanahan is an independent consultant currently in the role of Chief Architect for Alas Software. Robert brings his clients more than 20 years of software development experience and expertise. Among Robert's primary responsibilities at Alas are establishing technology, architecture and development standards, ensuring product quality and coherency across the Alas software product line. Prior to engaging with Alas in 2007, Robert served in a variety of technology leadership positions, including Chief Architect for Document Processing Systems and Technology Director for Cambridge Technology Partners. Robert is a graduate of the University of Michigan, a member of the World Wide Institute of Software Architects and an active supporter of open source software initiatives.



WITH EIGHT OUT OF TEN SOFTWARE DEVELOPMENT PROJECTS NEEDING OVERHAULS WITHIN THE FIRST YEAR, YOU WANT BETTER THAN “GOOD.” WHEN WE ROLL OUT A JAVA PROJECT, OUR STANDARD IS “WORLD-CLASS.” WE ARE THE ONLY COMPANY FUELED BY THE MOST POWERFUL TECHNOLOGY AVAILABLE: HUMAN INTELLIGENCE. CALL US TODAY AT 734-663-0877 OR VISIT US AT STOUTSYSTEMS.COM.

What to do: Something. Anything. Provide a regular stream of information that's informative, educational, interesting, engaging, and preferably, *new*.

3. If your branding is defensive, you're promoting the war, not your personal brand. Branding has often been compared to war on the battlefield. I like this analogy better: A brand is like a person. A person can engage someone or bore them. So can your brand. You can *be* interesting or you can *try* to be interesting (just like a brand). You can be passionate or monotonous. Inventive or ho-hum. In each case, your brand can also embody those qualities.

Here's a good acid test: If your brand were a person, *would you want to hang out with him on your time off?* If the answer is no, then the odds are others will have a similar response, leaving your brand as something one buys when it's needed versus being something that is passionately sought out.

What to do: Isolate the qualities of brands that you admire. Isolate the qualities of brands that gain not just acceptance in the market, but generate enthusiasm. Compare the two. Have an insight to come up with something new that is about you and not the branding war you may be in the middle of. Then be bold and stand for something.

David Brier, author of the book *Defying Gravity and Rising Above the Noise*, is known for annihilating cliché-based branding solutions. David is also the award-winning Chief Gravity Defyer at DBD International, a company that transforms their clients' brands into compelling brand leaders. Visit RisingAboveTheNoise.com to find out more.

SOFTWARE DEVELOPMENT STUDY GROUP

Ready for a quick and dirty intro to Ruby, Ruby on Rails, Silverlight, F#, Python or other software technologies? Then the Software Development Study Group is for you. The group focuses its short, highly interactive learning sessions on old and new software technologies in one concentrated hour every week.

Check the Software Development Study Group's Web site at <http://bit.ly/StudyGroup>

or <http://stoutsystems.com/calendar> for times and locations.

BILL HEITZEG JOINS STOUT SYSTEMS TO PROVIDE BUSINESS DEVELOPMENT AND TECHNICAL SERVICES



I've been aware of Stout Systems for over ten years. I've seen the tremendous success and growth Stout has experienced in that time and recently in particular.

I'm happy to be part of the next chapter of the Stout success story. In the coming years we'll be reaching a much larger and broader audience with our focus on the Human side of Technology development.

Feel free to contact me to talk about how we can help you get your project done. Email billheitzeg@stoutsystems.com or call me directly at 734-663-0877 x260.



Stout Systems
PO Box 2934
Ann Arbor, MI 48106

ADDRESS SERVICE REQUESTED

Presorted Standard
U.S. Postage
PAID
Permit No. 187
Ann Arbor, MI

Fueled by the most powerful
technology available:

Human Intelligence®

In our quest for answers to today's technical needs,
enjoy this latest issue of **most significant bits**
brought to you by Stout Systems

stoutsystems.com