

The Bionic App



We Can Rebuild It,
We Have the Technology



D. Keith Casey, Jr
CaseySoftware, LLC



How do I build v2.0?

Rebuilding an existing application from the ground up is painful at minimum, expensive at best, and generally a great way to waste time.

So don't do it.

Who is this guy?

- D. Keith Casey, Jr. - CaseySoftware, LLC
 - Java developer who saw the light...
 - Core contributor to dotProject
 - Specialize in refactoring existing codebases
 - Have utilized these tools and concepts to support sites for a cable news network, dP installs for 1200+ users, and *“involvement in the SMS system for a live musical performance show”*

The Motivation..

- Types of Applications:
 - Legacy
 - Useful applications, outdated or done poorly
 - Zombie
 - The 5-year old “prototype”, still actively used
 - The Second Try
 - “I learned from all my mistakes last time...”

Reasons to rewrite...

- PHP has come a long way
 - PHP5 and the frameworks
- User interfaces have come a long way
 - Ajax, XUL, Table-less designs, CSS
- Development concepts & concerns
 - Design Patterns, Security, Security, Security

The Joys of Rewriting...

- None of your old bugs
- No code from “that one guy”
- Do it Right this time, 100% Coverage!

- Few of your lessons learned
- New application (aka training)

The Tools

- Smarty Templates
- Zend Framework
- Prototype / YahooUI

Smarty Templates

- Started early-2002
- Built to separate UI and business logic, designer-safe
- Can support simple (or advanced) display logic
- Well-established, effectively a standard, lots of projects
- Built-in caching, functions, expandable with plugins

Smarty Templates Howto

- Create a template structure
 - usually `/styles/smarty/modulename/`
- Initialize Smarty
 - configure template and compile directories
- Load the Smarty object
 - assign php variables to the template values
- Display the template

Smarty Templates Howto -cont

```
<tr>
  <td style="background-color:#<?php echo $obj->project_color_identifier;?>"
  colspan="2">
    <?php
      echo '<font color="" . bestColor( $obj->project_color_identifier ) . ""><strong>'
        . $obj->project_name . '<strong></font>';
    ?>
  </td>
</tr>
```

-VS-

```
<tr>
  <td style="background-color:#{ $obj->project_color_identifier}" colspan="2">
    <span style="color: {dPbestColor bg=$obj->project_color_identifier};">
      <strong>{ $obj->project_name }</strong>
    </span>
  </td>
</tr>
```

Smarty Templates Howto - cont

```
<?php
if (file_exists('/path/to/templates/templateName') {
    $smarty = new Smarty();
    $smarty->template_dir = '/path/to/templates/';
    $smarty->compile_dir = '/cachePath/';
    $smarty->assign('pagename', $pagename);
    $smarty->display('templateName.tpl');
} else {
    usePreviousPageGenerationMethod();
}
?>
```

Zend Framework

- Starting in 2005 at ZendCon
- Feels like a toolbox...
- Zend_Db, Log, Pdf, Feed, Cache, MVC, and lots of other shiny toys
- Led by Zend, some involvement by IBM, wide community support

Zend Framework Howto

- Include the pieces you need
- Initialize the pieces
- Use the pieces

ZF Howto - cont

```
require_once '/Zend/Db.php';  
require_once '/Zend/Db/Table.php';  
require_once '/Zend/Log.php';  
require_once '/Zend/Log/Writer/Db.php';
```

```
$SystemLogMap = array('level' => 'priority', 'message'  
    => 'message', 'dateCreated' => 'timestamp');  
$writer = new Zend_Log_Writer_Db($db, 'systemLogs',  
    $SystemLogMap);  
$systemLogger = new Zend_Log($writer);
```

```
$AccessLogMap = $SystemLogMap;  
$writer = new Zend_Log_Writer_Db($db, 'accessLogs',  
    $AccessLogMap);  
$accessLogger = new Zend_Log($writer);
```

ZF Howto – cont 2

```
$accessLogger->log($message, $level);  
try {  
    doMyVitalyImportantTask();  
} catch (Exception $e) {  
    $sysLogger->log($errorMessage, $level);  
}
```

ZF – More Shiny Objects

- Logging to multiple destinations
 - As example, but also files, display buffers, etc
- Zend_Db / Zend_DB_Table
 - Active Record, (almost) no SQL for me!
- Zend_Feed
 - Quick RSS/Atom parsing
- Zend_Pdf
 - Create, edit, modify documents

Ajax - Prototype / YUI

- Prototype

- Javascript framework
- Works by extending the DOM
- Complete control, greater complexity

- YUI

- Javascript/css toolbox
- Consists of numerous components
- Less control, less complexity

Prototype Howto

- Include the library
- Write your Ajax methods
- Write your data serving method
- Do something with the data
- Tedious...
 - /style/_smarty/tasks/list.projects.html
 - /modules/tasks/tree.js

Yahoo! UI Library

- Include the pieces you need
 - Be aware of dependencies
- Configure/customize the pieces
 - Trust me, just look at the examples
- Use the pieces

In Summary...

Which are good for me?

None!

All!

Who knows?

You.

Questions / Comments ?

The Bionic App

We Can Rebuild It, We Have the
Technology

Slides available:

[http://CaseySoftware.com/events/
keith@caseysoftware.com](http://CaseySoftware.com/events/keith@caseysoftware.com)

*Most code samples are from Open Source projects
and subject to their licensing, etc.*



D. Keith Casey, Jr
CaseySoftware, LLC

