



Chicago, May 16-18, 2007

Moving to PHP5 with Style

Aaron Wormus

HedgeCo Networks
www.hedgecowebsites.com
aaron@hedgeco.net

May 17, 2007

Hyatt Regency Chicago, Illinois

- History of PHP Upgrades
- Benefits of PHP5
- Why you want to Upgrade
- Common Issues
- 5 Steps to Stylish Upgrades



Who Runs PHP4 in Production?



Who Runs PHP5 in Production?

- Faster – Less Memory
- New PHP Delimiter Tags
- Numeric string conversion when using the + operator
- Change in Conditional statement Syntax
- Standardization in function return values
- There was a script to ease the transition

- Look Mom, New Engine
- Parser Changes
- Error Reporting
- Type Juggling

- Streams
- ZE2 Referencing Issues

- New Object Model
 - Object referencing
- Define Classes before instantiating
- No reassigning \$this
- array_merge() and friends
- Replace CGI with CLI
- No Mysql Bundled
- var replaced with PPP (then unreplaced)

- Reference handling issues
 - http://php.net/README_UPGRADE_51.php
- Abstract private methods removed
- Redefinition of Class constants removed
- SQLite 3
- File Class
- Date

- New error messages
- Extension changes
 - Bzip2
 - Datetime
 - SPL
 - Zip
 - and more



Upgrade Help is here!

<http://us.php.net/manual/en/migration5.php>

A young child is shown in profile, blowing a dandelion seed head. The child's face is partially visible on the left side of the frame. The background is a bright, slightly blurred field of dandelions. The overall image has a soft, ethereal quality with a light color palette.

Ryan Gordon was asked what the first step
in converting windows games to linux

Ryan Gordon was asked what the first step
in converting windows games to linux

“Put it on a linux box, run gcc and see what happens”



But does running PHP5 mean you
have a Modern Web Application?

XML

- SimpleXML
- XMLReader/XMLWriter

Data Access

- PDO
- SDO
- MySQLi
- SQLite

Security

- ext/Filter

OO Goodness

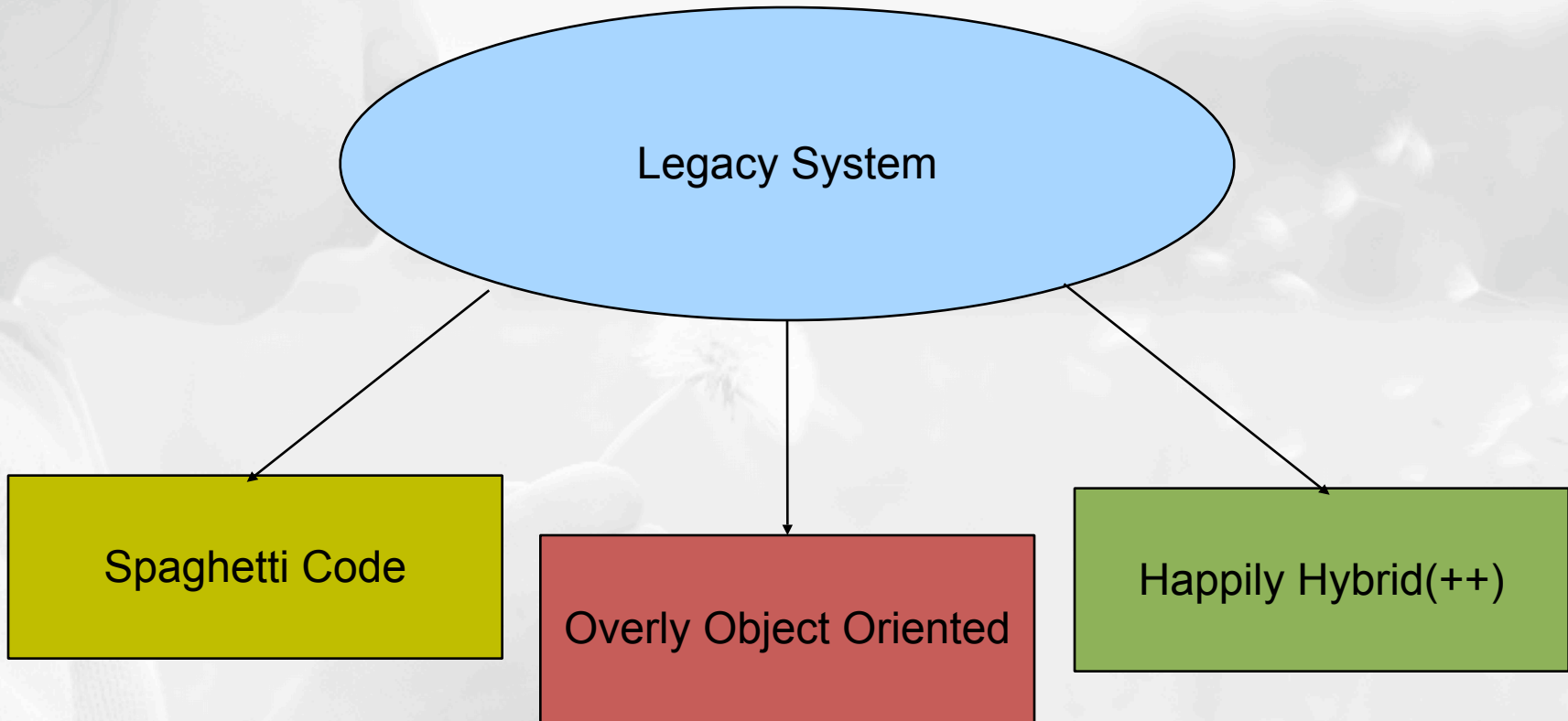
- SPL / Iterators
- New Object model
- PPP / Class Constants / Type Hinting / Magic methods
- Exceptions

Web Services

- Soap
- json_*

- Cost Effective Maintenance
- Predictable Development Cycles
- Creating a code base on which programmers enjoy working
- Create a product that will add value to the company
- Use technologies that will benefit the company's bottom line

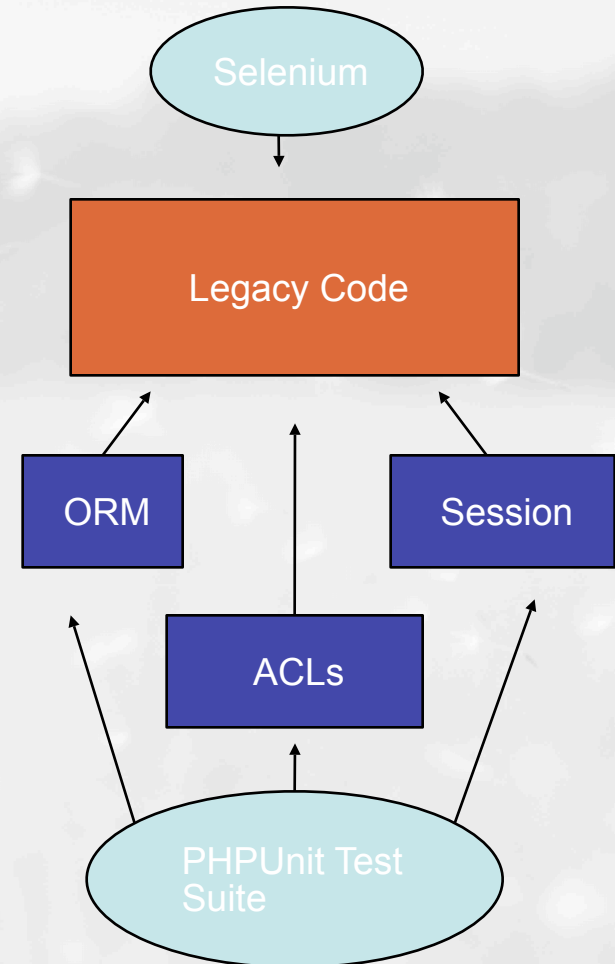
- **Unit Tested** – No need to delay a launch because a code addition broke another part of the system
- **Documented** – Enable new recruits to easily start working on a codebase.
- **Modular** – Enable multiple people to work on the same codebase at the same time
- **Running PHP5** – Allow you to deploy new technologies quickly



- Code that was migrated from PHP3 to PHP4
- Little if any separation from business and presentation logic.
- Lots of redundant code.
- Little use of modern technology (Web services, XML, etc).
- Lots of run-around code, manual xml handling code, ugly database libraries, security issues.
- Example: osCommerce, PHPNuke.

How do we clean this up?

- Determine reusable components within the existing code.
- Create classes which abstract the functionality out of the existing codebase.
- Find classes that already provide this functionality (ezComponents, PEAR)
- Create Unit tests for the classes & possibly browser compatibility tests for front end testing.
- This can be completed on an incremental basis



- Code that was re-written when PHP4 was released.
- Focus on Object Orientation.
- Overly Complex Code
- Slow
- Difficult to make changes to the lower level system without breaking the entire platform.
- Example: eZpublish, other content management systems which have come and gone

- There may not be a problem
- If there are problems, they may be showstoppers.
- Refactoring OO code is easy
- Providing you have a stable and unit tested API

Steps:

- Standardize API
- Create missing Unit Tests
- Fix any PHP5 issues in the code.

- Code which wants to do the right thing
- No coding Standards
- A lot of code, varying quality
- Some unit tests, some documentation
- Usually run well enough to discourage anyone from touching them
- Most Corporate systems are consist of this type of code
- Hybrid++ can also be systems like Wordpress or drupal which maintain a procedural codebase which

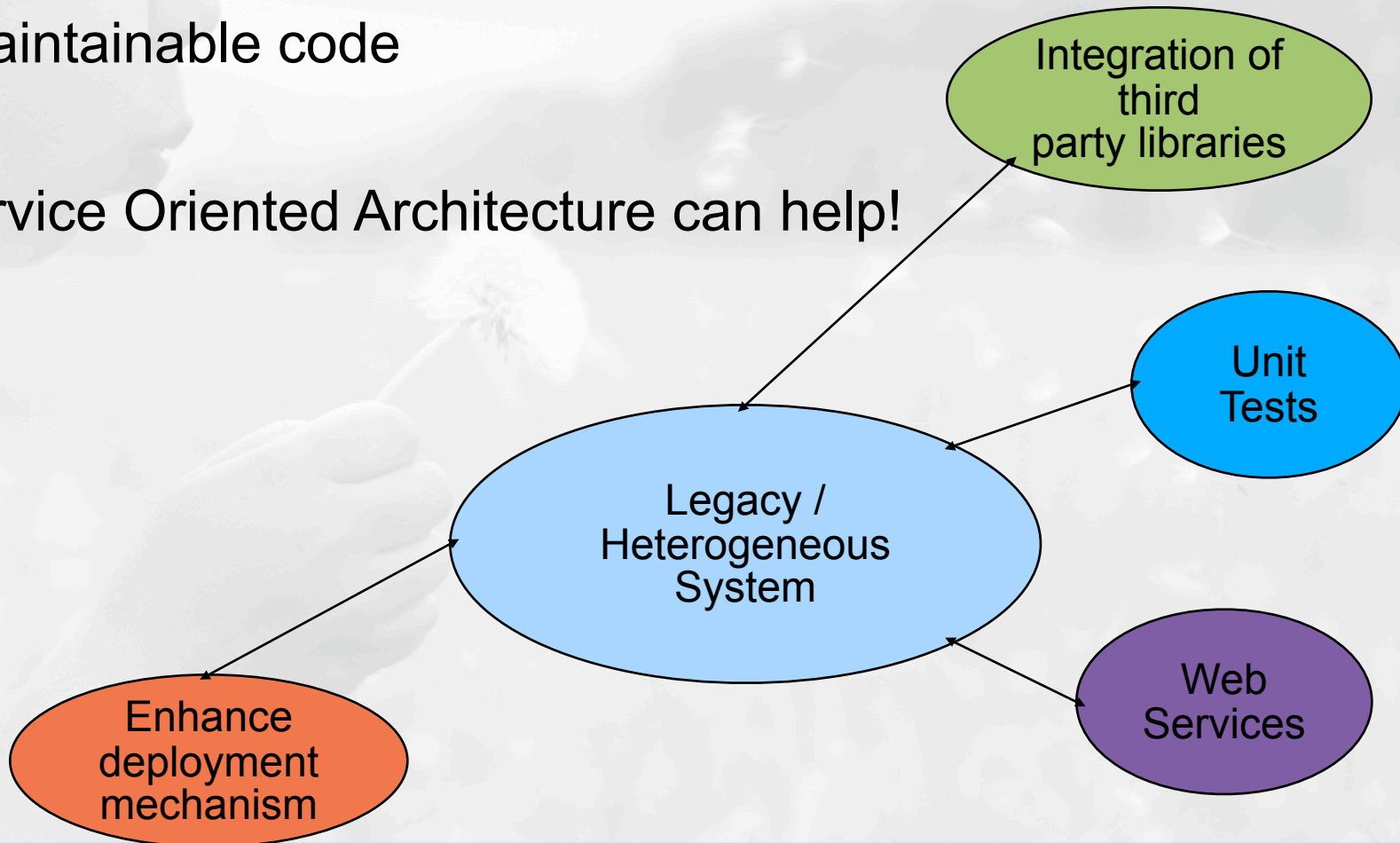
- Document the existing code and processes
- Ensure optimum code reuse, with a good class heirarchy.
- Unit test your classes

The code will now run in PHP5 and fulfil the requirements for being a Modern Web Application

At a future point you can optionally

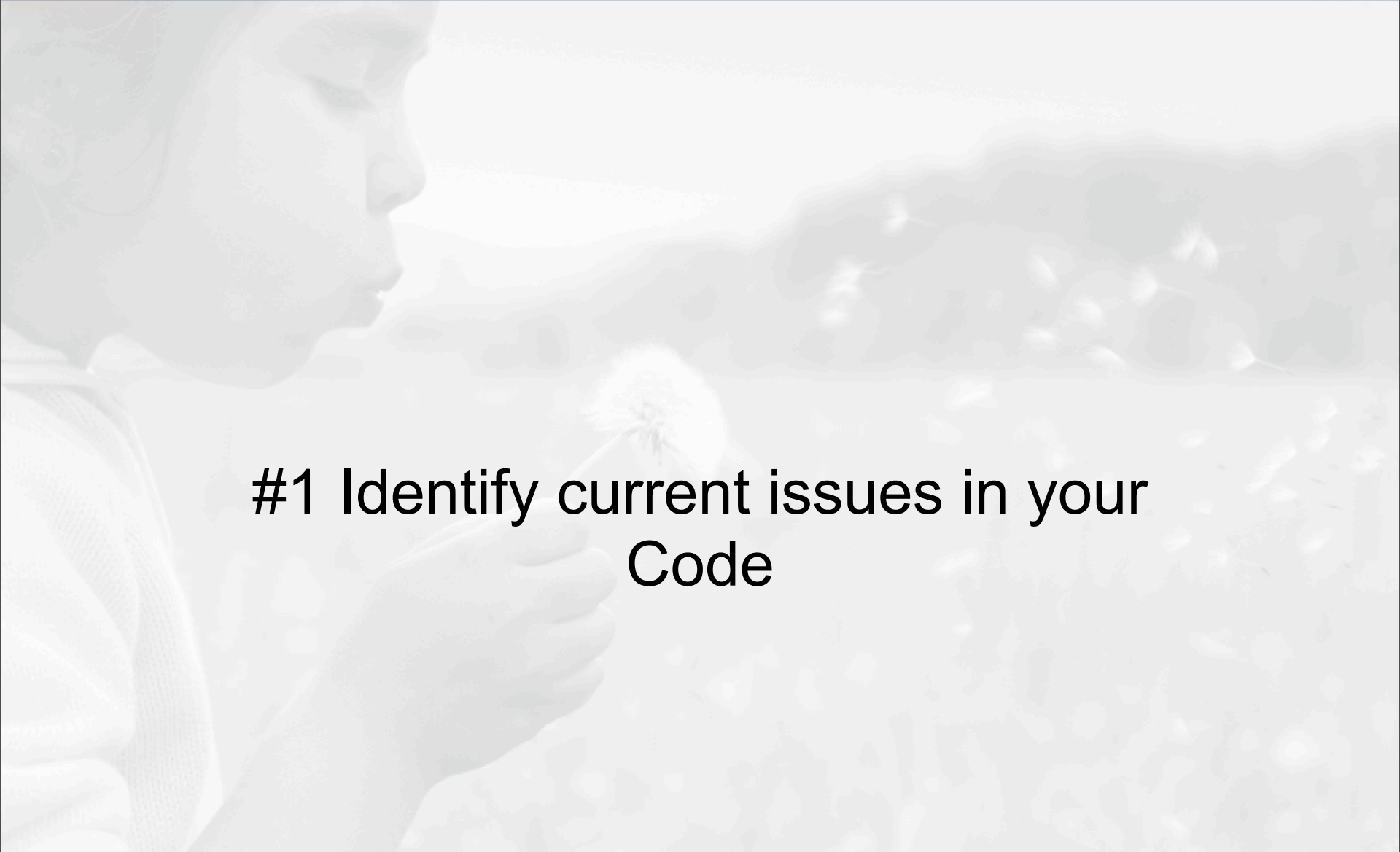
- Implement a front controller for ease of use and management
- Move to a fully OO MVCish framework

- Legacy libraries
- Custom extensions
- Unmaintainable code
- A Service Oriented Architecture can help!



A grayscale photograph of a young child in profile, blowing a dandelion seed head. The child's face is on the left, and the dandelion is in the center. The background is a soft-focus field of dandelions. The text "5 Steps to Upgrading" is overlaid in the center of the image.

5 Steps to Upgrading



#1 Identify current issues in your
Code

- Issues that you need to identify
 - Security
 - Speed
 - Maintainability
 - Future feature additions



#2 Create your PHP5 strategy

- Modularize your code – remove code redundancy
- Create an API and standardize it for future development
- Document your API
- Unit Test your API

- Organize yourself before you switch
- You can't make a move to PDO before your database access has been centralized

#3 Run a lint test

```
php -d error_reporting=4095 -l script_to_check.php
```

A young child is shown in profile, blowing a dandelion seed head. The child's face is in focus, and the dandelion is held close to their mouth. The background is a soft-focus field of many other dandelions, creating a sense of a vast field. The overall image has a light, airy feel with a slightly desaturated color palette.

**#4 Make as many changes as
you can to your current site**

- Refactor your current codebase before forking
- Add any features into the PHP4 version

A young child is shown in profile, blowing a dandelion seed head. The background is a soft-focus field of many more dandelions. The image is overlaid with a semi-transparent white box containing the text.

#5 Branch your Code and Make changes

A young child is shown in profile, blowing a dandelion seed head. The child's face is in focus, and the dandelion is held close to their mouth. The background is a soft-focus field of dandelions, creating a sense of depth and movement as the seeds are blown away. The overall tone is light and airy, with a focus on the natural world.

#6 Test and Deploy




When portions of your code need
rewriting...

- Time spent fixing bugs is increasing
- Developers can't foresee issues or make accurate estimations
- Security issues
- Developers won't touch your code
- Your code has a bus-number of 1 – too complex

- Code has reached the end of its life cycle

- Optimize your efficiency by using PHP5 packages
 - Zend_Framework – not just a framework
 - eZcomponents – more complex and powerful

- Don't rewrite for speed
 - Speed optimisations usually involve 1% of the code.
- Refactor, don't rewrite
- If your code is buggy, rewriting from scratch has the potential of introducing new bugs.

A young child is shown in profile, blowing a dandelion seed head. The seeds are captured in mid-air, creating a soft, ethereal glow against a bright, slightly blurred background. The child's face is partially visible, showing concentration and joy.

Your developers will want to rewrite
code.

Don't let them!



Questions?