



The Knight Rider Methodology to Software Development

Eli White

<http://eliw.com/>

Knight Coder

“A shadowy flight into the dangerous world of a man who uses tools & technologies to make better coders.”

Guidelines for this talk

These are suggestions

I won't mention all options

Use what makes YOU better ... experiment

Editor

Most important piece of technology!

Good Features:

Code completion & Tool tips

Regular expression search & replace

Context sensitive highlighting

Extensibility

Editor



Profiler

Discover where you need to optimize

Profiler



valgrind / callgrind



xdebug



kcachegrind

APD

Load Tester

Stress your code

See at what point it breaks

Document benchmarks to test improvement

Load Tester

http_load

siege

Apache Bench

Performance Tools

Use technologies that make performance easy

Performance Tools

APC – Instant Performance & Caching

Memcached – Distributed Caching

Gearman – Parallel & Batch Processing

Libraries

Don't reinvent the wheel!

Use other coders' time as your own

Know when to walk away though

Libraries

PHP:



JS:



YUI



Debugger

Code is never perfect the first time

Use technology to help you find the problems

Debugger



Zend Studio



APD

WARNING: Maybe not so easy!

Debugger

Don't Forget:

```
echo '<pre>'; var_dump($var); echo '</pre>';
```

Debugger



Web Developer Toolbar

Test Suites

Tests are great

Tests keep your code running

Make writing tests easy for yourself

Test Suites



PHPT



Selenium



Multi-Tier Servers

Have multiple servers with different purposes

Control the flow of development & testing

Not always separate machines

Multi-Tier Servers



Development

Initial independent work by programmers



Integration

Making sure all previous work plays well together



Testing

Running unit & regression tests



Staging

As true to production as possible for final tests



Production

It's live and hopefully working

PHP Error Reporting

E_ALL & E_STRICT

It is your friend. 'Nuff said.

Version Control

Just do it!

So many benefits:

Recovering from mistakes

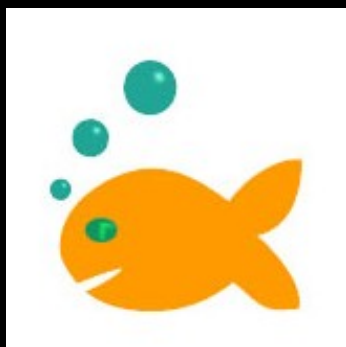
Integrating code from multiple sources

Concurrent work streams

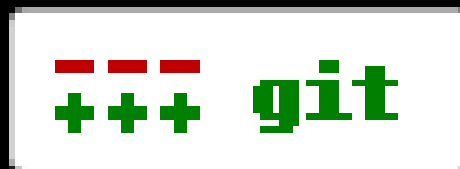
Finding old code

Assigning blame

Version Control



CVS



Lint

Automatically find broken or suspicious code

Hook into version control for total protection

Lint

```
php -l  
Language syntax only
```

PHPLint

Strict, Simple code required, Type validation

CodeSniffer

Coding standards are important for teams > 1

CodeSniffer will ensure compliance

Hook into version control

GUIs

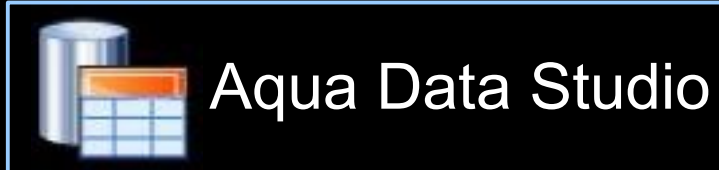
Yes I said GUIs

Use them to make complicated tasks easy

DB maintenance, Version Control, etc

GUIs

DB:



Version Control:



SmartSVN



SmartCVS



Little Things

Don't underestimate the power of:

rsync

grep

find

sed

cut

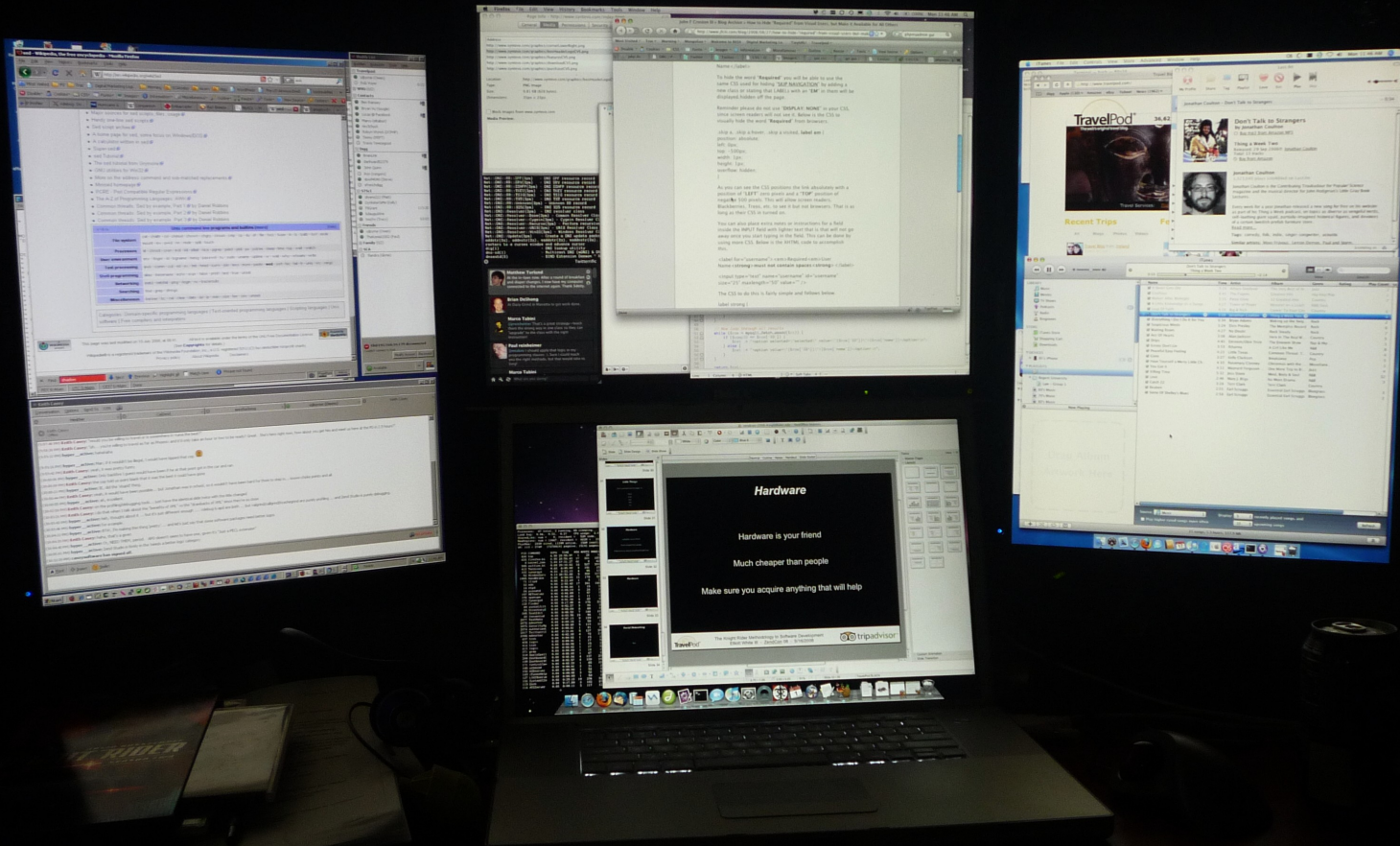
Hardware

Hardware is your friend

Much cheaper than people

Make sure you acquire anything that will help

Hardware



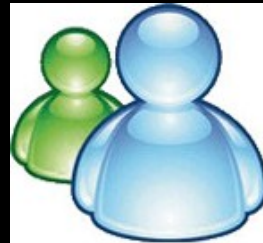
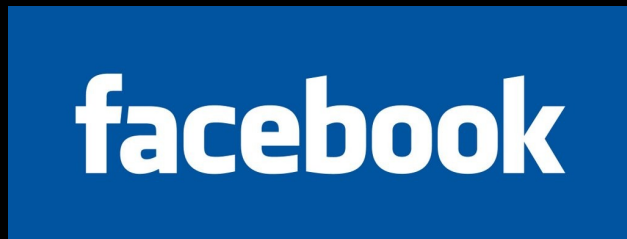
Social Networking

Make contacts

You are at the best place for that right now

Stay in touch

Social Networking



Any Questions?

For this presentation & more:

<http://eliw.com/>

