



# Development Strategies for Web Applications



# Background

- Started with PHP/MySQL in 2000
- BS and MS in computer science from MSU
  - What I missed in the process
- Six years of web development at NSCL
- Why did I decide to give this talk?



# Initial Configuration

- Production web server
- Production database server
- Development web server with debugging enabled
- Development database server
- Network accessible production code
- Network accessible development code
- Copy script to move files from development to production



# What problems are we trying to solve?

- Passwords open to public or developers
- Code changes need to move to production
- Developers overwriting each others code
- Development code affecting production
- No way to confirm defects or behavior
- No way to go back to a working version
- Updates to database schema made by hand



# Three Tier Model

- Each tier will have its own configuration file for passwords, database server, and paths
- Production
  - Only updated with source control checkout
- Test / QA
  - Configured the same as production
  - Limited access
- Development
  - Every developers desktop


# What problems are we trying to solve?

- ✓ Passwords open to public or developers
- ✓ Code changes need to move to production
  - Developers overwriting each others code
  - Development code affecting production
  - No way to confirm defects or behavior
  - No way to go back to a working version
  - Updates to database schema made by hand

# Apache Friends on the Desktop

- Free cross platform PHP enabled web, MySQL, and mail server
- Allows the developer to schema changes
- Mail is only delivered locally
- Database changes will not impact production
- Limited path into production

# Development


- Each developer modifies her own configuration file
- Check out the most recent version for version control 
- Deploy a backup of production database or use a dummy schema if security dictates
- Developer commits code based on your own standards



# What problems are we trying to solve?

- ✓ Passwords open to public or developers
- ✓ Code changes need to move to production
- ✓ Developers overwriting each others code
- ✓ Development code affecting production
- No way to confirm defects or behavior
- No way to go back to a working version
- Updates to database schema made by hand

# Test / QA

- Important to be kept up to date
  - All code needs to go through QA
- Those in charge of testing have access
- After testing concludes, code is tagged as a release version 
- Code can be replaced with production at any time

# What problems are we trying to solve?

- ✓ Passwords open to public or developers
- ✓ Code changes need to move to production
- ✓ Developers overwriting each others code
- ✓ Development code affecting production
- ✓ No way to confirm defects or behavior
- No way to go back to a working version
- Updates to database schema made by hand



# Communication Between Developers



- Concurrent development is possible since each user has their own version
- Identify independent portions of a project to break it up
- Code merging must be regularly practiced



# Version Control

- Much more than just source code
- All database schemas (create statements)
- All database changes (edit statements)
- Stored Procedures, Triggers, and Views
- Ability to reproduce database corresponding to code version
- Could also include documentation

# Rollout Procedure

- After testing has been completed and version created
- Mark all features implemented and bugs fixed with this version 
- Do a checkout of code to production
- Run updates on production schema 

# What problems are we trying to solve?

- ✓ Passwords open to public or developers
- ✓ Code changes need to move to production
- ✓ Developers overwriting each others code
- ✓ Development code affecting production
- ✓ No way to confirm defects or behavior
- ✓ No way to go back to a working version
- ✓ Updates to database schema made by hand



# Considerations

- Time to deploy updates should remain short
- Convincing others of the benefits associated with the additional cost of development
- Rollout is a two step process





# Other Ideas

- Bug tracking
- Consistency of input
- Database class (mysqli, PDO, custom)
- Security <http://shiflett.org/php-security.pdf>

# Stored Procedures and Views

- SP's are a contentious issue
  - Pros (security, limiting traffic, abstraction)
  - Cons (new api, hide business process, no objects, testing, error reporting)
- Views for temporal tables



# Open Questions

- What does it mean to roll back a schema?
- How to improve the testing phase?
- How to handle file uploads?



# Demo

- Mail
- Tortoise
- WinMerge



# Software

- [www.apachefriends.org](http://www.apachefriends.org)
- [www.tortoisecvs.org](http://www.tortoisecvs.org)
- [winmerge.org](http://winmerge.org)
- [dev.mysql.com/downloads/gui-tools/5.0.html](http://dev.mysql.com/downloads/gui-tools/5.0.html)
- [PuTTY \(Pageant\)](#)
- Firefox Plugins
  - Web Developer
  - Firebug