Drupalcamp Vienna 2009

Development workflow and deployment at

ABSOLVENTEN.AT

DAS JOB- & KARRIERENETZWERK

Klaus Purer
2009-11-28
http://klausi.fsinf.at

Who am I?

- Student at the Vienna University of Technology
  - Software Engineering & Internet Computing
- Software developer & server admin at
- Google Summer of Code Student 2009
  - Work for the Drupal Rules module
- Passionate free & open source user/developer

Who are you?

- You are Drupal developers
- You want to know how others do their daily Drupal work
- You want to organize development and deployment for your use case
- You have basic knowledge about version control, databases, server administration

Who is Absolventen.at?

- Job exchange platform in Austria
- For school and university graduates
- Drupal based
- Highly distributed development team
- ca. 7 developers
- Incrementally evolving the code base

Absolventen.at software stack

- + custom Drupal themes
- + official Drupal modules
- + custom Drupal modules
- + custom server scripts

Goals in development

- Preserve all code changes ever made
- Rollback to any previous state possible
- Seperate new features from bug fixes
- Work in parallel on different parts
- Avoid conflicts by editing the same code
- Show features and tests to others

Version control with Subversion

- Solves the code history problem
- Allows working in parallel
- Does not help separating bugs from features
- Solution: devel branch and stable branch
  - Requires Merging between them (not funny in Subversion < 1.5)
- Problem: a commit must not break the code
- Solution: big new features in feature branches
- Does not avoid conflicts for you

Project management

- Solves the edit conflict
- We run an internal project management site (Drupal-based)
- Includes an issue/bug tracker with developer assignments
- Each module/functionality is assigned to one developer (maintains updates, ...)
- Organizes the overall development process in milestones

Documentation

- Implementation needs documentation references to spread the knowledge
- Often a view pointers suffice to explain functionality
- Custom patches need documentation
- Development & Test guidelines
- Design guidelines
- OpenAtrium offers a good documentation and collaboration basis

Communication

- Project web with issues, comments, blogs, docu pages etc.
- Notifications via Email and RSS Feeds
- Jabber/XMPP interaction (+ group chat)
- Identi.ca private status updates (twitter-like)
- Commit-Log with useful commit messages
- Face to face meetings

Development work flow

- Similar to Scrum
- Time is sliced into one week sprints
- Features and minor fixes are committed into the devel branch
- More critical bug fixes are committed into the stable branch
  - Production site is updated immediately
- Every Monday development is merged and populated to the production site

Bloody Monday

- Production database is dumped and provided to the developers
- Subversion branches get merged
  - Conflicts are reported to the developers

- Simpletests get executed and verified
- Stable branch is freshly cloned from the devel branch
- Production site is updated
- Caches are cleared

Bloody Monday (2)

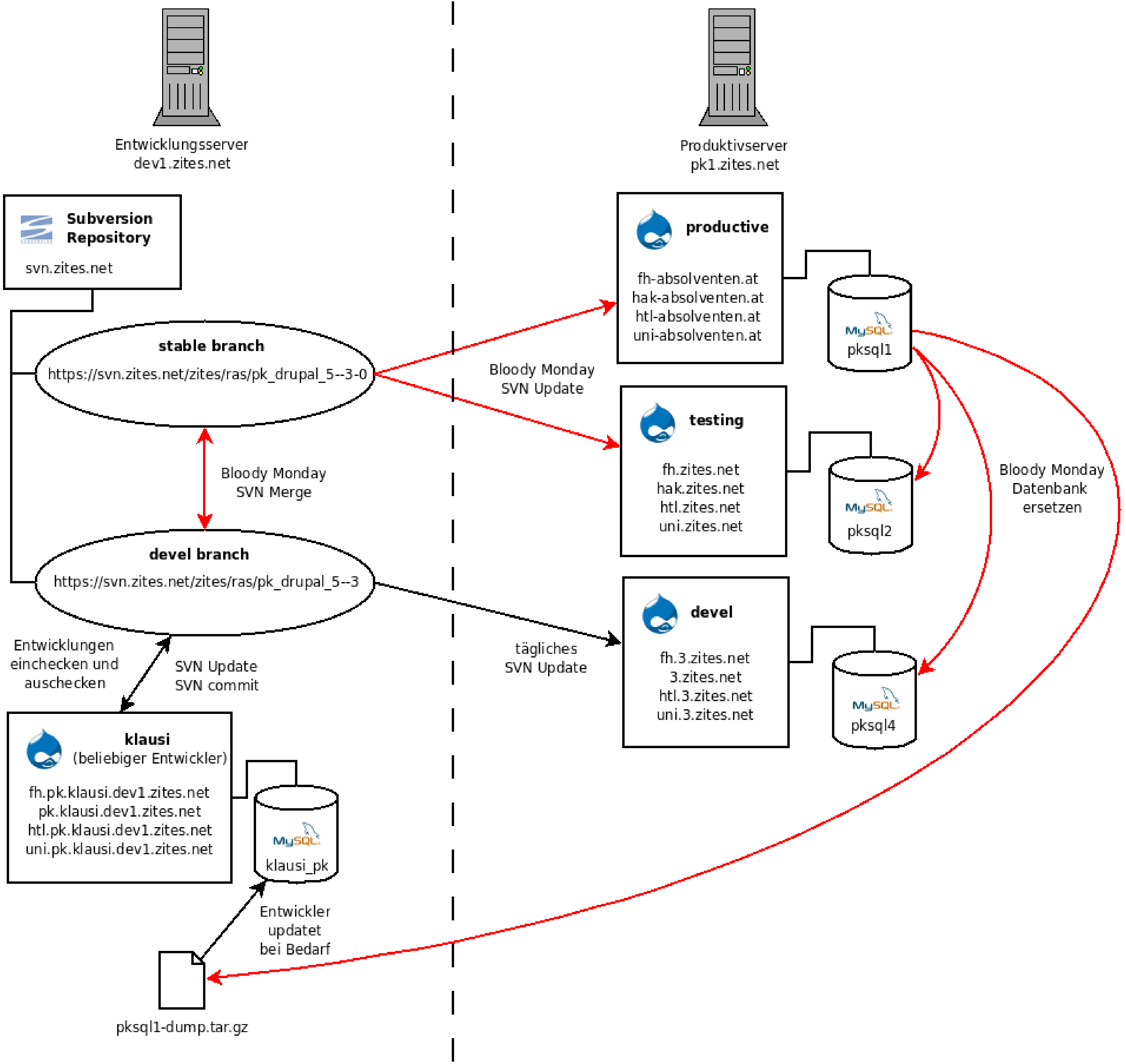
- Watchdog logs are examined
  - File not found errors – are there any dead links on your site?
  - PHP errors – did you check that array for null values?
- Additional DB changes are deployed
- Merge-log that summarizes all abnormalities as a report

Goals in deployment

- Separate development environment from production environment
  - Development server vs. production server
- Provide a testing (staging) site that is an exact production clone
- Provide a development site for each developer
- Production site updates must be as fast as possible
- Production site updates must be consistent

Deployed sites

- Productive: the live site
- Testing: clone of productive, experiments allowed
- Devel: reflects current devel branch code
- Devel X: dev site of developer X
- Devel Y: dev site of developer Y
- Devel Z: dev site of developer Z
- ...


Entwicklungsserver
dev1.zites.net
Produktivserver
pk1.zites.net
Subversion Repository
svn.zites.net
stable branch
https://svn.zites.net/zites/ras/pk_drupal_5--3-0
Bloody Monday
SVN Merge
devel branch
https://svn.zites.net/zites/ras/pk_drupal_5--3
Entwicklungen einchecken und auschecken
SVN Update
SVN commit
klausi
(beliebiger Entwickler)
fh.pk.klausi.dev1.zites.net
pk.klausi.dev1.zites.net
htl.pk.klausi.dev1.zites.net
uni.pk.klausi.dev1.zites.net
klausi_pk
Entwickler updatet bei Bedarf
pksql1-dump.tar.gz
Bloody Monday
SVN Update
tägliches
SVN Update
productive
fh-absolventen.at
hak-absolventen.at
htl-absolventen.at
uni-absolventen.at
pksql1
testing
fh.zites.net
hak.zites.net
htl.zites.net
uni.zites.net
pksql2
devel
fh.3.zites.net
3.zites.net
htl.3.zites.net
uni.3.zites.net
pksql4
Bloody Monday
Datenbank
ersetzen

Deployment with Subversion

- Drupal core + modules + settings are checked into the repository
- **All** changes are committed to the repository and then checked out to production use
- Never ever edit files directly in your production site!
  - Inconsistencies can cause severe developer headache
- Testing/Development sites are able to checkout an exact copy

Deploying database changes

- Manually: do the changes on a testing site, do it again on the production site. Tedious :-(
- Export: do the changes once and import them on the production site. Must be module-supported (e.g. Views, Rules, …)
- Code: implement default hooks and commit them. Needs a developer.
- Features: Similar to export, but allows you to commit Features-generated modules.

Test driven development

- … we don't do that, actually
- Test supported development
- All mission critical features are covered by Simpletests
- Mostly black box tests that execute tasks on testing (!) sites and report any failures
- Developers are able to verify that their code does not break anything existing
- Not all bugs are found via Simpletest, but many are avoided from the beginning

Drush

- Drush = Drupal shell
- Executes Drupal tasks from the command line
- Automatically detects the Drupal site from the directory you execute it in
- We use it to
  - Run the Simpletests
  - Add/Upgrade/Remove/Enable modules
  - Run cron

Mysqldump speedup

- Slow:

```
mysqldump $DB_NAME > dump.sql
```

- Faster (30%):

```
mysqldump --tab $TMPDIR $DB_NAME
```

  - Import:

```
cat $TMPDIR/*.sql | mysql $DB_NAME
mysqlimport $DB_NAME $TMPDIR/*.txt
```

Subversion drawbacks

- Subversion is not perfect for merging
- Subversion is centralized, a feature must be working before you can commit it
- Subversion wants to do CVS right
- Linus Torvalds: “There is no way to do CVS right”
- “If you like using cvs, you should be in some kind of mental institution or somewhere else.”

Version control future

- Distributed version control systems
- Merging is an integral part of the design
- A commit is local
- Changes are pushed to and pulled from arbitrary locations
- Versioned development in an independent manner

Version control future (2)

- And remember kids:

  Die CVS, die!

- The good guys:

git

Bazaar

mercurial

darcs

Deployment Future

- Drush – Drupal command line scripting
- Aegir hosting system
  - Easy deployment of a new site with a view clicks
  - Management of many sites within a Drupal installation
- Drush Make – resolves dependencies
- Features module
  - Bundle your CCK, Views, Rules etc. settings and save them as “features”

Discussion

- Questions
- Answers
- Comments
- Feedback