Upgrading Evergreen: A Case Study

Jeff Davis
Sitka Implementation Team
BC Libraries Cooperative
jdavis@sitka.bclibraries.ca
How to upgrade Evergreen

Preparing the code

Testing

Upgrading production
Preparing the code

• The big stuff:
  – server code (application layer)
  – database upgrade scripts

• The other stuff:
  – OPAC skins
  – config files
  – building the staff client
  – etc.
Server code – local vs. upstream

- our own local git repo
- includes:
  - Sitka-specific customizations (B, E)
  - fixes/features shared with upstream (C)
  - backports from upstream (D)
Server code – local vs. upstream
Server code – rebasing

- git interactive rebase
  - create new sitka_2_2 branch from sitka_2_0
  - git rebase -i rel_2_2
- resolving merge conflicts
  - git mergetool
- PROTIP: keep a log!
Server code – interactive rebase

```
1 pick f17f6a2 (sitka) disable the holds library selector for privacy purposes
2 pick 686a0b1 (sitka) allow universal barcodes, do not strip whitespace
3 pick a25c93e (sitka) user editor canadization
4 pick 30f83cf (sitka) custom OPAC js functions
5 pick 0d123d5 (sitka) using a frog for transit popup to improve UX
6 pick 3c87ec4 (sitka) opt-in patch - makes opting in more intuitive in multibranch organizations
7 pick 54a194c (sitka) dummy page for disabling things like acq
8 pick 064b4d9 (sitka) customizing en-ca locale with SITKA urls
9 pick a930689 (sitka) adding production modified circ scripts
10 pick 061ed10 (sitka) fixing google books link as per RT#2621
11 +---210 lines: pick 096c579 (sitka) patch for holds queue -- limit to within requested selection area of
221 pick 364808c Revert to a more EG community style OPAC LCN display
222 pick f5ab879 decreasing default MARC Federated search hit count to 5 from 10 - rt#12274
223 pick 97783b6 SITKA - library short name fixes in predue and overdue emails
224 pick a41e24d (sitka) increase subobj limit for bt_parms, improves lcn on items with many copies
225 pick 794f3c3 (sitka) -- new link addresses for find a library, help, privacy, and about
226
227 # Rebase d1679f5..794f3c3 onto d1679f5
228 #
229 # Commands:
230 # p, pick = use commit
231 # r, reword = use commit, but edit the commit message
232 # e, edit = use commit, but stop for amending
233 # s, squash = use commit, but meld into previous commit
234 # f, fixup = like "squash", but discard this commit's log message
235 #
236 # If you remove a line here THAT COMMIT WILL BE LOST.
237 # However, if you remove everything, the rebase will be aborted.
238 #
```
Preparing the code

- **server code**
- database upgrade scripts
  - `config.upgrade_log` is your friend
  - `pg_dump` and `pg_restore` for snapshots
  - time-consuming!
  - assigning permissions
- other stuff (OPAC skins, config files, building the staff client...)
Testing

• who are your testers?
• what is being tested?
  – multiple test servers (VMs)
  – evolving codebase – iterative process
• testing procedures (checklists)
• issue tracking and fixing bugs
  – prioritization
  – weekly summary
Milestones

- May 2: test server 1
- May 15: upstream 2.2rc1 released
- May 29 (Jun 4): test server 2
- Jun 3-11: bib record cleanup
- Jun 13: upstream 2.2.0 released
- Jun 13 (Jun 15): test server 3
- Jun 27: Sitka 2.2 code freeze
- Jun 30-Jul 2: upgrade production
Upgrading production

- pre-upgrade upgrades (PG 9.1)
- scheduling downtime
- spare brick
- maintenance page
- upgrading servers and database
  - automated upgrade scripts
- final testing before go-live
Production environment
What worked

- division of labour when preparing code
- internal wiki (milestones, tasks, procedures)
- support team testing
- ticket prioritization + weekly summary
- multiple test servers
- 2-day launch window
Areas for improvement

- MARC cleanup too close to upgrade
- final week of testing was too rushed
- formalized (and/or automated) testing
- managing multiple test servers
- automated deployment
- communication with third parties