Contributing to GHC

Joachim Breitner
Haskell Implementors Workshop 2014
6 September 2014, Gothenburg
The take away: git

Synopsis

- `git clone --recursive git://git.haskell.org/ghc.git`
- `git pull && git submodule update --init`

More on

- [https://ghc.haskell.org/trac/ghc/wiki/Building/GettingTheSources](https://ghc.haskell.org/trac/ghc/wiki/Building/GettingTheSources)
- [https://ghc.haskell.org/trac/ghc/wiki/WorkingConventions/Git](https://ghc.haskell.org/trac/ghc/wiki/WorkingConventions/Git) *(and subpages)*
The take away: Phabricator

Synopsis

- arc diff *(to create a revision)*
- arc diff *(to update it)*
- arc land *(to push it)*

Remember

- Your commits are gone and squashed at the end
- The phabricator revision description becomes the commit message

More on

- https://ghc.haskell.org/trac/ghc/wiki/Phabricator
Recap: A patch workflow

1. Open ticket in trac, or find an existing ticket
2. Assign it to yourself
3. Create phabricator revision
4. Link it with the ticket (Phab:Dnnn)
5. Get code reviews and CI results
6. Improve code and documentation
7. Validate
8. Push the code using arc land
9. Close the ticket
Other nice tools

Communicate!

- ghc-dev mailing list
- #ghc IRC channel
- Comments on Trac Tickets, Phabricator diffs
- ICFP

Review and validate!

- https://phabricator.haskell.org/herald/:
  Watches for certain commits and trigger actions, e.g. an audit.
- https://phabricator.haskell.org/harbormaster/:
  Builds all suggested revisions and commit commits
- https://travis-ci.org/ghc/ghc: Quick validation per commit
- http://ghcspeed-nomeata.rhcloud.com/: History of benchmark results
Contribute to GHC!

And don’t forget to comment your code.
Thank you for your attention.

Cover image © CC-BY-SA http://commons.wikimedia.org/wiki/File:Apadana_Persepolis_Iran.JPG