Minutes of the EuroLLVM 2016 "Surviving Downstream" BoF
Quick recap of where Sony is now

Merge pain really depends on local changes.
Can be counted in number of patches, lines, number of full time merge engineers
Previously at Sony took 3 months to merge 6 months of LLVM work.
Manual merges ~1 per month
Experimenting with continuous merge bot. Conflicts ~2-3 per week. Seems reasonable for us.

Questions from the floor:

- How often should I merge from upstream?
- Nightly or every six months, depending on your resources and requirements
- Comment: it does depend on what you are doing, and how different it is from other work which is upstream.
- Associated comment - important therefore to have wide range of architectures upstream.
- Sony has a triple to cover all of our changes
- Q: What about performance. With every merge you may have a large number of performance regressions.
- We look at performance before release
- Q. Doing 6-monthly integration. Beyond just textually changes, refactoring can cause problems. Also semantic changes.
- Sheer volume is part of the challenge
- Q. Changes in undocumented assumptions in peoples heads.
- Less of a problem for Sony but common problem for others.
- Q. Some tests are based on code from customers which can't be shared with the community. How much of your time is spent on this?
- There was one monthly merge where our internal tests didn't fail... Typically 2 days to track things down and clean up.
- Typically we can share reduced test cases with upstream when small
- Q. Important to test upstream work early to distinguish issues between upstream and local.
- Test as much testing on upstream to make sure that is clean before mergeing
- Hal Finkel suggested best time of the day to do merges
- Q. Problems caused as much by why something has changed as what. Helps if the why is clear from the commit message.
- If this is a recurring theme then remind people.
- Q. Do you have IR level tests, or just C, full compilation?
- LLVM/Clang tests are smoke tests, we have no internal IR tests, we add full compilation tests.
- Q. Do you have any private/internal passes?

Tips and tricks for easier merging

- "Tag" comments to mark local changes
- git imerge
- git rerere
-

- Try using branches corresponding to the Google's stable releases (https://llvm.org/svn/llvm-project/llvm/branches/google/stable/, https://llvm.org/svn/llvm-project/cfe/branches/google/stable/, ...) instead of integrating from trunk. These undergo a serious testing and contain a higher quality snapshots of LLVM/Clang/... code compared to an average state of trunk (no patches, just snapshots).