

Each group should appoint a leader, a scribe, and a reporter (who might be the leader or the scribe or a third person). The leader's job is to keep the conversation focused and making progress, and to make sure everyone has an opportunity to contribute. The scribe's job is to talk notes, using one of the following google docs. And the reporter's job is to get ready to report back on the discussion in the next session, in 2-3 minutes.

Each group might want to start by having everyone briefly introduce themselves. They should then talk about software sustainability in the context of HEP.

The output of each group's discussion should be 3 specific actions that the HEP community could take to make software more sustainable. At least 2 should be things that could be accomplished within 2 years, or at least that would make significant progress that would lead to a measurable difference within 2 years.

A suggestion is to talk generally for 35 minutes, and then to focus on the specific outputs in the last 10 minutes, based on the previous discussion, but this is up to each group to decide.

Participants

Rob Kutschke (FNAL) Mu2e, formerly ARGUS, CLEO, FOCUS, BTeV, SID

Ben Krikler (Bristol), CMS and LZ (previously on COMET), fellow with UK's SSI

Nils Krumnack () ATLAS

Claire David (York University / FNAL) ATLAS / DUNE

Exchange (rough transcript for now)

Rob: continue integration on multiple platforms / push for the 'multiple'. Make CI a real tool

Ben: what are the obstacles?

Rob: establishing the infrastructure. We have resources (e.g. grid job submission at FNAL) CI running on GPU (CMS). Buying a '6 month software engineer' too big investment for small experiment.

Nils: GPU dev at ATLAS ... worth it?

Claire: what about sharing skilled scientists between experiments and reduce costs this way?

Rob: debugging/improving experiment-specific algorithms to take advantage of attributes of their hardware.

Nils: my experiment with a share → better technically / with permanent people some beneficial experience with the historical aspect / sharing people too thinly = risk.

Ben: improving quality of documentation and CI reduces requirements on continuity?

Nils: even if sharing you need the [?] continuity? impossible to fully document [?]

Claire: would an award for document be beneficial or backfire?

Rob: backfire, seen as a poor decision to invest their time. Reflected in the future career path

Nils: we could accept code only if it comes with documentation

Rob: but what about giving credits?

Ben: award = piece of bigger scheme: how do we incentivise? in large experiment it is useful.

Citing software in paper?

Rob: in astronomy, ongoing model of being cited. Not enough room on citation.

Ben: sign repository / not necessarily a paper. Just better referencing

Rob: but could we do that at scale in our community? Need other mechanism.

Nils: can work with individual-domain expert, each having responsibility

Rob: job = identify documentation and get it on the domain-expert worklist

Nils: a postdoc would write the software [...]

Ben: position is physics-trained postdoc? would an award for documentation backfire?

Summary

Expand CI and other testing

- **multiplatform**

Reward documentation

- Develop incentive for spending time on documentation
- Warning on award: could backfire → coding too time-consuming
- Citing software in paper not ideal. Need a mechanism to promote developers

Develop Career paths for those who work in computing and software