
SW Development, Deployment and Validation/Verification WG

— Friday, closeout session —

Progress this week

- We had 2 discussion sessions on Tuesday that covered significant missing pieces in the document and some of the smaller details
- We had a “writing” session yesterday
 - However, it turned out to be much more useful to discuss *what* we would now write before we actually tried to write it
 - We also had people who hadn’t managed to join on Tuesday and we wanted to take advantage of that
- In the next slides there’s a summary of what we discussed and our current ideas for the roadmap
 - There are detailed notes buried in the [live document](#)
 - Substantive content appearing in the [draft document](#) soon!
 - We believe the July deadline can be met


Recognition, attributions and citations

- Our community recognises published papers and citations
- So we think that software has to follow the same model
 - Currently many software developments are reported on at ACAT and CHEP
 - Usually there is a conference publication
 - Thus the additional effort to write a software paper for a refereed journal is not so high
 - We believe (hope!) that software journals are fine with this
- In parallel we also have to ensure that we cite properly papers for software that we use
- It will be impossible to cite every internal software package used in CMSSW or Athena
 - A general paper on the experiments' software would seem to fit here - analogous to the detector papers
- There are new methods of [citing software itself](#), rather than a software paper
 - These are really interesting, and we think worth supporting, but don't look like they can currently replace refereed academic citations

Case studies

- There was substantial agreement that we should learn the lessons of the past
 - Successes are great, but from failures you learn more
- Successful projects: CVMFS is a great example
- Failures are not hard to find: COOL/CORAL, Objectivity
- Generic lessons
 - Listen to users and respond
 - Solve one problem and do it very well
 - Developer team that communicates well
 - One project to rule them all generates a single point of failure (no pressure to succeed?)
- We will try to add a short section into the CWP, but propose a longer study of this in the next year

These are quite 'agile' points that we also want to distill



Roadmap

- The main lesson from our group is that HEP should be as generic as possible in its software choices
 - Thus we should let ourselves be driven by developments out of our control in the wider software world: 10 years we don't know what the landscape will look like to plan, even 5 years is too much
 - Synergy with industrial and external software experts were also mentioned in the sessions as interesting added value
 - Evaluation of key case studies within this context (e.g. GitHub)

Roadmap: 1 and 3 (or short and medium)

- Year 1 Plans

- Forum for inchoate ideas, new stuff and discussion of problems
 - We all agreed on the high weight of this forum creation as a way to discuss problems and track external developments
- Revisit packaging note, this time with more prototyping
- Revisit project template, with guidance on development methodology
 - Avenues to help projects get started: design, some development advice, etc. (incubatorish, but informal)
- Common training in core software skills [Training WG]
 - Benefit: raise HSF profile with younger collaborators

- Year 3 Plans

- Regression testing toolkit for experiment physics validation and DQ [Simulation WG]
- C++ refactoring
- Static analysis
- CI service for the community

Non HEP specific contributions are really encouraged!

- We want to embed an internal review into this roadmap - we really do need to keep checking that we're making progress