

# HEP Software Foundation

---

Nikhef input on scope, purpose, intent, and organization

## Background

An initiative has been started to form a HEP Software Collaboration. This is timely given the increasing pressure on manpower, both in terms of delivered results (more efficient code) and capacity (harder to get and pay for good people).

Our own motivation for submitting this draft is mostly to address this last point. We believe that for good people, finding the relevant information about best practices on programming for collaboration and for high performance will not be a significant barrier; making a collection of good starting sources is not a task that requires a HEP Software Collaboration. The stated goals of enhancing recognition also do not seem particularly relevant to us; recognition that produces results comes in the form of some type of compensation (monetary or reduction in fees) or publications.

What a HEP software collaboration can offer is the potential to reduce obstacles to getting work done, and ensuring that software artifacts produced by the various teams work well together, removing another drain of valuable manpower. Our opinion is that such considerations should be the driving factor behind the HEP Software Collaboration, all activities of the Collaboration only being approved after being assessed for whether they are of value in this sense.

Following the spirit of Oxana's contribution, we list some bullet points here.

## Goals

- Achieve mutual compatibility of software packages
- Promote the use of **already existing** open standards, that is not inventing new standards
- Discourage the design, construction, and adoption of any HEP-specific packages when an equivalent non-HEP-specific package already exists. The aim here is to avoid the effort in developing it, debugging it, checking its compatibility, deploying it when an pre-existing package will already have these areas covered, and finally maintaining it – a definite win in the manpower department.
- Provide a very lightweight infrastructure for organization (from the bottom up) and meetings necessary for dedicated discussions on particular topics – e.g. a few mailing lists and indico pages.
- Provide a recognizable entry point for others wishing to collaborate with us, use our products, or learn from our work.
- Framework implementation:
  - Collaborative tools (mailing lists, Web site, Wiki etc)

- Optionally: code repository for newcomers
- Optionally: binary repositories for different OSes – for those softwares that have no other distribution channels

## Scope and Duration

The scope is those HEP software packages that have a demonstrated need for interoperability with other packages, and/or any HEP software groups that wish to participate. The duration is for as long as there is a demonstrated benefit from such collaboration.

## Development Model

All groups are in principle free to follow their own development model. We expect that a natural coordination will occur with regards to the release model, as this is the point at which interoperability issues surface. The particulars of the release model should evolve organically from actual agreement between the developing parties and relevant groups confronted with deploying the software; this agreement should again be directed at a global decrease in effort, proper weighting must be applied.

Previous software collaborations have been in some cases badly broken in this sense so we provide a couple of examples. If something costs a developer 100 hours that saves each of 1000 users 1 hour, the 1000 users win. On the other hand something that costs 10 development teams each 100 hours, which saves a five-person certification team a week, it is clearly better to spend the extra week of the certification team than to burden each of the 10 development teams.

## Policies

We have no particular opinion towards policies, other than that their existence (and especially their application by developers) needs to have a clear benefit in terms of global, properly weighted effort. Otherwise it is a waste of everyone's time to read the policies before ignoring them.

## Governance Model

This should be as lightweight as possible; it is probably unavoidable to have some governance, somebody has to be the one to send the emails and call the meetings. The governing parties should be chosen by development teams themselves, as the people in charge need to have an extremely clear view of what is really of benefit to the collaboration as a whole.

## Membership

Any HEP-related software development team should be accepted, provided they are willing to work together with the others towards the goal of minimizing total effort and maximizing compatibility and performance. Teams generally recognized within the community as not sharing these goals should also be deleted from the collaboration.

As far as equality of membership between the various groups, the only factor that is important is the extent to which the group shares the goals of the collaboration and does their share towards achieving them.