HEP Software Foundation (HSF)
White Paper Analysis and Proposed Startup Plan

The HSF Startup Team
Version 1.1, January 7 2015

For more information see hepsoftwarefoundation.org

Table of Contents

Revision history

V1.1 Jan 7 2015: Revisions in light of feedback from V1.0

- Addressed a contradiction about funding for projects and personnel, in the executive summary and then later in the document. Clarified that while the foundation will not in the beginning provide funds for development projects, it may well require real resources (effort, physical facilities) to run some of its activities and services. (Currently the support comes from our organizations allowing people to participate in HSF on a best-effort basis.) Also, it cannot be excluded that in the future the HSF might become a vehicle for supporting development projects.

V1.0 Nov 11 2014: First draft release

Executive Summary

The HEP Software Foundation (HSF) is an international, HEP-wide, bottom-up initiative to create a community body that will facilitate coordination and common efforts in high energy physics software and computing. This document lays out initial ideas and proposals for building the Foundation. Ultimately the HSF will be what people bring to it; all are encouraged to get involved in creating a useful organization.

Current HEP software is the result of many decades of development and must now evolve to meet the challenges posed by new experimental programmes by fully exploiting the new technologies available in the computing landscape. Within the HEP community, it has become progressively clear that these issues can only be addressed by pursuing commonality between HEP projects and by engaging with non-HEP partners, to share all available expertise and make the most of limited resources.

The objectives of the HSF as a community-wide organization are in sharing expertise; raising awareness of existing software and solutions; facilitating the use; giving visibility to new ideas and catalyzing new common projects; promoting commonality and collaboration in new developments to make the most of limited resources; aiding developers and users in creating, discovering, using and sustaining common software; disseminating knowledge about HEP software; supporting career development for software and computing specialists; and sponsoring training and outreach.

A recognized community organization can also provide a forum to attract effort and support, and provide a structure for the community to set priorities and goals for work. It can also facilitate wider connections; while the HSF is a HEP community effort, it should be open enough to form the basis for collaboration with other sciences.

The HSF will not, at least in the beginning, be a source for funding projects, or for directly supporting personnel. Rather, the HSF is attempting to marshal existing resources so they are used more effectively in the community. HSF effort currently comes from our organizations allowing people to participate in the HSF on a best-effort basis. The HSF may well however require real resources (dedicated effort, computing facilities) to run some of its activities and services, depending on what those turn out to be. Also, it cannot be excluded that in the future the HSF might become a vehicle for supporting development projects. The HSF will not dictate conditions under which projects may participate, such as mandatory coding standards or the use of particular software infrastructure.

The HSF had its beginnings in a proposal to launch a HEP Software Collaboration [1] that was circulated in early 2014 and was followed by a workshop [2] in April 2014 at CERN to discuss it. The workshop led to a call for White Papers from across the community to contribute ideas on what shape and form a collaboration should take to be useful to the community. Ten White Papers [3-12] were submitted by different teams and individual authors, covering a wide spectrum of diverse geographical areas and software domains that are active within the

community. All interested people were then invited to be part of the HSF Interim Foundation Board [13], and a Startup Team [14] was set up to develop a startup plan for the HSF on the basis of the White Papers, community consultations and other inputs. A principal first task for the startup team was to develop this document.

This document has two interwoven threads, reflecting its twofold purpose. In the following sections the many proposals and ideas expressed in the White Papers are analysed and summarised, pointing out the main areas of agreement between them, as well as the areas where discussions may still be needed to reconcile some differences between the suggested approaches. This constitutes the first purpose of the document: a summary of the White Papers themselves.

Our second purpose is to take the next step beyond a bare White Paper summary: to synthesise them, together with other inputs and the deliberations of the startup team, into a proposal as to how to proceed. Accordingly, sections include, as well as a factual summary, *the startup team's assessment of how the HSF should approach the area, italicized*.

Finally, the document concludes with a summary of the startup team's recommended (and partially underway) course of action in the form of a proposed startup plan for the HSF's first months. The plan will evolve as practical experience is gained and as a result of further discussions, in particular those at the upcoming HSF workshop [15] in January 2015 at SLAC.

The White Papers

Answering the call made during the April workshop, ten White Papers were submitted by different teams and individual authors between April and June 2014. These papers cover a wide spectrum of geographical areas and software domains that are active within the community:

1. Six papers [3, 5, 7-10] provide the input from **members of the HEP communities in Germany, Netherlands, France, US, UK and Italy**. Most of these represent personal views of the authors, rather than official views of the HEP institutions in those countries.
2. One paper [4] provides the personal views (not an official Collaboration view) of the present computing coordinator of one of the four **LHC experiments**.
3. Two papers [6, 12] provide personal input from teams of **Grid software developers** and from the **CERN Openlab** CTO Office.
4. Finally, one paper [11] describes the **experiences from the Geant4 Collaboration** and the software design and organizational principles that allowed its success.

Largely speaking, broad agreement can be found amongst all White Papers on the general motivations for a HEP Software Foundation initiative and on several specific aspects of it. Important differences also exist between the proposed approaches in various domains. It should also be noted that some aspects of the HSF are explicitly mentioned in (and are sometimes the main focus of) some White Papers, while they are not covered in detail and/or seem to be implicitly taken for granted by others. While it would be difficult and unnecessary to precisely list the suggestions made in each White Paper, we attempt to convey an impression of all these similarities and differences. In some cases, some suggestions put forward in the initial proposal for a HEP Software Collaboration [1] will also be pointed out in the coming sections.

General motivations, goals and scope

There is broad agreement in all White Papers about the general motivations and goals for a HEP Software Foundation. Some differences can be seen in the focus and priority attached to a specific goal, but *the following goals are clearly shared by most, if not all commenters, and can be regarded as an agreed starting point.*

1. **Evolve HEP software to optimize its performance** – to meet the challenges of fully exploiting the performance of (current and future) new technologies and architectures, and those posed by the increased requirements of new experimental programmes.
2. **Promote common software developments within HEP and with non-HEP partners** and the reuse of existing standard solutions – to avoid the duplication of efforts and facilitate synergies, sustainability, new funding opportunities and economies of scale.
3. **Promote and facilitate compatibility, interoperability and integration testing** of diverse software components – to enable software sharing between HEP experiments and with other communities, by removing the obstacles coming from different software interfaces, data formats or external library dependencies.
4. **Improve communication and expertise sharing about software** within HEP and its partners in HSF – to facilitate the flow of information between developers, users and other stakeholders, to set up a forum to discuss and promote common activities and to provide a recognizable mechanism for interacting with other communities.
5. **Promote software development skills, training and jobs**, increase visibility to the rest of the HEP community and to funding agencies, and provide training opportunities and career counselling. This aims to address the difficulties in attracting, hiring, paying and retaining skilled developers and to thus improve software sustainability.
6. **Incubate innovation and systematically foster new developments** and exploratory pilot projects – to provide a supportive environment for new projects combined with a low barrier for entry and without “management hostility”, while anticipating and avoiding disruptions to the existing data processing workflows and services.

Regarding the duration of a HEP Software Foundation, there seems to be general agreement that this initiative should not be time-limited and would exist for as long as it is useful.

The following sections analyse more in detail the specific suggestions put forward to better define the scope of the HSF and the means by which the HSF could reach the above goals.

Software focus areas

The diversity in the provenance of the White Papers is to some extent reflected in the assumed software scope for the proposed HEP Software Foundation:

1. **Monte Carlo simulation, reconstruction frameworks and data analysis** are clearly mentioned or implicitly assumed by all White Papers as relevant to the HSF. **Grid middleware** is mentioned by some.
2. **General purpose toolkits like Geant4 and ROOT** are clearly mentioned or implicitly assumed by all White Papers as relevant to the HSF. **Experiment-specific software frameworks and libraries** are also mentioned by some White Papers as relevant to the HSF, in the sense that they have a potential for being reused by other experiments or, alternatively, for being replaced by different, common, software components.

From the white papers and other discussions there is general agreement that the analysis chain from TDAQ and generators through to analysis tools should be the initial focus area. Particularly, optimizing software for parallel software & computing and future technology challenges are priorities. and its optimization should be the primary initial focus. Grid software (or more generally distributed software) and experiment-specific software projects are welcome to participate in HSF where there are willing proactive participants, even though initial core efforts will be targeted primarily at the above. Priorities and scope should be re-discussed and refined as HSF progresses.

Technology challenges

There is general agreement among the White Papers about the technology challenges faced by HEP software, i.e. the reasons why it is essential to evolve it to optimize performance for the present and future experimental programmes:

1. The main challenge that HEP software is already facing today is widely recognized as the paradigm-shift resulting from the evolution of CPU architectures towards the use of **parallelism (multi-core CPUs and vector units).** Our current code is sub-optimal in this respect and has to be reengineered to fully exploit the available performance.
2. Some White Papers such as that from Openlab [12] additionally point out the challenges that are coming from the emergence of **GPU accelerators, low power computing cores (e.g. ARM) and heterogeneous architectures**, which may eventually lead to another paradigm shift away from x86 computing.
3. One White Paper [7] explicitly mentions the challenge of **efficient access to large volumes of distributed data**.
4. The need to adapt our software and computing models to new resource provisioning technologies such as **Cloud computing (including commercial providers), HPC facilities and volunteer computing resources** has also been mentioned [1,10].
5. More generally, all White papers explicitly or implicitly recognize that the HSF would be an asset in facing **any new technological challenges that may arise in the future**.

These are all important aspects of the necessary evolution of HEP software and computing and are appropriate to be addressed by the HSF, partly through training, but also by supporting relevant R&D.

Software process – Policies, guidelines

Essentially all White Papers advocate recommending some guidelines and policies, to improve the software process, promote common developments and reuse of existing solutions, and enable interoperability.

1. There is general agreement that **open-source license models** (and even some specific types of these, e.g. [7] copyleft models à la BSD or Apache2) should be recommended. Indeed, making code public is widely felt to stimulate open, bottom-up, collaborations on software projects (and to ultimately pave the way to "social coding" [7], especially if combined to the appropriate web-based source code management services).
2. There is also general agreement that **open standards and existing solutions** should be adopted as much as possible, to facilitate interoperability and to avoid the syndrome of "reinventing the wheel".
3. Specifically, some White Papers propose that guidelines for **release management and software dependency management** should be recommended, to achieve various goals including clear version numbering and component interoperability, but also service stability [10,11] through transition periods (e.g. managing and avoiding disruptions from API changes).
4. Specifically, some White Papers propose that guidelines for **component interfaces and data format/representation** should be recommended (using open standards wherever possible), to facilitate interoperability (even cross-language [7] if possible).
5. Several White Papers remind of the importance to recommend **QA guidelines for component tests** (and possibly for stress/scale tests [9] too).
6. Several White papers propose that **guidelines for documentation and tutorials** should be recommended.
7. Some White papers also suggest that **guidelines for performance benchmarking** should be recommended.
8. Most White papers suggest that **peer reviews** should be recommended.

Other than these, White Papers tend to differ in the level of development and coding guidelines they recommend. Concepts such as **Agile, user-driven and modularity** are often mentioned. Several White Papers (most notably that [5] by Nikhef) explicitly point out that all groups should be free to follow their own development model and that any policies must have clear benefits, otherwise they would be a waste of everybody's time. One White paper [4] suggests that **security guidelines** should be recommended.

In many discussions as well as in several White Papers, it has been stressed that guidelines should be considered as ***recommended, rather than enforced***. An interesting proposal in this respect is made in the White Paper by IN2P3 members [4], where it is suggested that different levels of compliance to these guidelines may be required for different levels of membership (e.g. for ***"hosted" vs. "endorsed" projects***); this will be discussed again in section 2.10 below. More generally, there is widespread support in the White Papers to the idea that the HSF should "support" projects rather than "manage" them [3].

Recommendations regarding policies and guidelines:

- *The guideline and policy areas itemized above all should be considered in HSF planning (with appropriate priorities).*
- *Guidelines should generally be considered as* ***recommended, rather than enforced;*** *adherence to guidelines is not a prerequisite for HSF participation.*
- *Compliance requirements (realistic ones) could however be considered for certain levels of HSF membership (see members section below) or endorsement.*
- *The HSF should "support" projects and should not foresee a role in "managing" projects in the near future.*

Software process - Common infrastructure and support teams

Some, but not all, White Papers propose that a common computing infrastructure and some common support teams should be set up by the HSF, to promote common development, interoperability and integration testing.

1. Setting up and maintaining a **common build and test infrastructure**, for nightly builds and/or continuous integration builds, is advocated by several White Papers. This would allow the regular build and validation of both individual software components and integrated software stacks.
2. Likewise, setting up **common certification and integration teams** is also advocated by a few White papers. These teams could eventually enable a HEP software distribution mechanism (similar to that existing today for the LCG Application Area software [8]).
3. Access to **common computing resources on many platforms including emerging hardware (CPU, GPU, ARM...)** is advocated by a few White Papers.
4. Access to **common software tools (compilers, profilers, optimizers...)** and availability of guidelines for their use are recommended by a few White Papers.
5. Setting up **common collaborative tools (web site, mailing lists, twiki)** is generally recommended by most White Papers.
6. Setting up **common issue trackers and source code repositories** (in some cases also binary repositories) is advocated by some White Papers, while others explicitly mention the option of using existing ones like GitHub. In any case, the economies of scale [10] that are possible by avoiding the proliferation of several independently maintained infrastructures are explicitly or implicitly recognized in most White Papers.
7. It is generally felt appropriate that the HSF should provide some level of **IP support** for the choice and implementation of the appropriate license model (which should be open-source as described in section 2.5 above).
8. It is also generally agreed that the HSF should provide some **support for organizing peer reviews** (recommended as described in section 2.5 above).
9. Several White Papers also propose that the HSF should be able to **set up and staff dedicated task forces or "SWAT teams"** [6] to address some specific issues reported by some of its stakeholders.

Generally speaking, most White Papers suggest that the use of these common infrastructure services should be optional and not mandatory. This infrastructure may be particularly useful for small projects that do not have the resources to implement it.

It should also be noted that, with respect to other aspects of the HSF, this is clearly one area that may require a larger ***commitment of computing and personnel resources*** and may depend more on the governance and resource contribution model of the Foundation. Quoting one White Paper [6], “the initial portfolio of services will reflect the needs of the stakeholders that participate in the Foundation and the resources that the Foundation can attract”.

Recommendations regarding common infrastructure and support teams:

- *Common infrastructure should be considered only on the basis of declared need and interest from participants.*
- *Where it exists, it should be optional.*
- *Tools and services from the open source world (e.g. GitHub or BitBucket) should be favored where possible over in-house developments. Such tools are mature and usually better than what has emerged from HEP specifically.*
- *Some scheme of organized peer reviews should be considered. And/or HSF participation in some way in externally organized reviews, as suggested by ATLAS computing management for the coming ATLAS software review.*
- *“SWAT teams” -- targeted teams assembled from the HSF community to address specific issues, contribute to reviews, develop certification standards, etc -- could be a good way of marshalling community expertise towards common ends.*
- *HSF will not have the physical resources to directly provide access to platforms, compilers, etc., but could work in concert with resource owners and funders to make such resources available and accessible to a broader community.*
- *The HSF should be open to the idea of a common software certification and integration team if this emerges as useful to participating projects.*
- *HSF should aid and advise in open source software licensing for participating projects.*

High-level coordination and support tasks

Many White Papers recommend that the HSF should be able to provide some coordination between projects and some high-level support tasks for all projects collectively:

1. Many White Papers suggest that the HSF should **actively promote and propose existing solutions and collaborations between projects**. It should discourage HEP specific solutions if standard solutions exist [5]. It should discourage fragmentation, while at the same time encouraging diversity [7], accepting a best of breed approach with several parallel developments [9] and allowing projects to compete for promotion [12].
2. To this end, some White Papers explicitly suggest that the HSF should **collect information about and maintain an up-to-date repository of existing and planned projects** (which may also be useful as a “marketplace” [7] to users and developers).

3. Some White Papers also suggest that the HSF should **propose a general roadmap** (and, in particular, centrally perform risk analysis [10] for HEP software projects).
4. Specifically, most White Papers suggest that the HSF should **incubate innovation**, by stimulating the launch of and providing a supportive environment for new projects. The Openlab White Paper [12] even more strongly suggests that the HSF should set up a “systematic process for innovation”.
5. Specifically, many White Papers also recommend that the HSF should **facilitate maintenance and lifecycle management** of software considered “of value to HEP and beyond” [6]. In other words, the HSF should be concerned with the sustainability [8] of such software components.
6. Many White Papers also suggest that the HSF should **provide support for and promote collaboration on the preparation of funding proposals**. This also includes promoting outreach to funding agencies [6], setting up a framework to attract funding [9], avoiding destructive competition for funding [7] and promoting synergies for community-wide calls like Horizon 2020 [10].

Recommendations regarding high level coordination and support:

- *The HSF should actively promote and propose existing solutions and collaborations between projects, and facilitate the incubation of new common projects.*
- *The HSF should facilitate collecting information about and maintaining an up-to-date repository of existing and planned projects, by providing an information gathering and exchange system that the community can use and contribute to. (The startup team has begun this in the context of the HSF website, hepsoftwarefoundation.org.)*
- *The HSF should seek ways to facilitate and ensure long term software sustainability and lifecycle management for key software.*
- *Other proposals -- general HEP software roadmap, HEP software risk analysis, an active role in developing funding proposals -- could be considered when the HSF matures.*

Communication and visibility within and outside HEP

Improving communication and expertise sharing about software within HEP and with non-HEP partners is considered an essential goal by most White Papers. To this end, there seems to be widespread (explicit or implicit) agreement on some suggestions:

1. The HSF should set up **developer forums to allow expertise sharing** between developers from different software projects.
2. The HSF should also provide **forums for the communication of projects and developers with users**. Organizing topical “events” [6] may be useful to this end.
3. The HSF should promote the **communication between users about common needs**. Organizing topical workshops [6] may be useful to this end.
4. The HSF should promote **contacts and organize workshops with non-HEP partners**, specifically inviting members of different communities.
5. More generally, the HSF should promote the **visibility of HEP software outside HEP**. Several specific suggestions include the publication of an HSF newsletter [3], outreach

activities [4], maintaining a public face of the HSF web site [6] aimed at non-HEP sciences, and even more generally making sure that the HSF is a recognizable entry point [5] representing the public face of HEP software to other communities [9] that may wish to collaborate with us.

6. Also, the HSF should **promote the visibility to funding agencies of HEP software** (and/or of some specific HEP software projects).

Recommendations regarding communication and visibility:

- *All the activities itemized above should be facilitated and pursued by the HSF. The information and communication tools offered by the HSF will be a principal means of doing so.*

Software developer skills and careers

Supporting software developer skills and career paths is also considered an essential goal by most White Papers. To this end, there seems to be widespread (explicit or implicit) agreement on some suggestions:

1. The HSF should provide **training opportunities for developers**. This should include the organization of software schools (like CSC and Bertinoro) for young developers and of a permanent training infrastructure [10] for accomplished developers.
2. The HSF should provide **career support for developers**, for instance by listing job opportunities [6] and by helping to shape well defined career paths [9] that should provide advancement opportunities on par with those in, e.g., detector construction [10].
3. More generally, the HSF should **increase the visibility and recognize the value for HEP of software developers**, for instance by raising the profile [9] of this career and recognizing it of equal value to scientific research [10], as well as by acknowledging and promoting specific “champions” [8] in the field. Eventually advocate and support an open-access peer-reviewed journal [3].

Recommendations regarding software developer career support:

- *The HSF should pursue all of the above, targeting primarily training at first and then extending to career support as opportunities arise to have influence on shaping S&C career paths. The HSF should support an open-access peer-reviewed journal initiative.*

Stakeholders and membership

All White Papers explicitly or implicitly recognize that many different categories of stakeholders exist for the HSF initiative. These include at least developers/projects, users/experiments, management (and/or resource providers) and non-HEP partners. The most exhaustive attempt at classifying stakeholders and their concerns can be found in Richard Mount’s White Paper [6]. Most White Papers express the concept of HSF “membership” applying to software projects. In particular, the HSF may be seen as a “partnership of projects” [8]. This is surely not seen as excluding membership applying to individuals; the HSF will be successful if it is made up of interested, engaged individuals, not just “projects”. But regarding project membership, several White Papers propose conditions and benefits for software project members of the HSF. As mentioned before, an interesting suggestion is made in the White Paper by IN2P3 members [7], where it is proposed that different levels of compliance to HSF guidelines may be required for different levels of membership, which would lead to different levels of benefits. Specifically, a “partner” project designation, projects sharing the goals and vision of the HSF to some extent, would have both fewer conditions for admission and fewer benefits than “endorsed” projects.

About conditions for project membership, the following points were mentioned:

1. There is general agreement that **HSF membership should be voluntary**. This is the first essential ingredient to ensuring that the HSF is a “buy-in” [8] bottom-up initiative and that it does not risk being seen as a “top-down” initiative.
2. Most White Papers suggest that **projects become members if they are accepted by the HSF**. In the IN2P3 White Paper [7], a formal acceptance is only required for “endorsed” projects, while it is much easier to become a “hosted” project. Another White Paper [4] suggests that several different areas (like CHEP tracks) should exist in the HSF and that one such area must also accept the project (or a new area is created if none of the existing ones is appropriate).
3. Most White Papers suggest that, **in becoming HSF members, projects accept its policies** and in particular accept some statement of principle that they are “willing to work together to minimize effort and maximize compatibility and performance” [5]. The IN2P3 White Paper [7] suggests that this only applies for “endorsed” projects, and from these projects it actually asks for even more stringent conditions, such as a pledge for future support/maintenance and also an acceptable measured performance.

About benefits of project membership, the following points were mentioned:

1. Many White Papers agree that **the main benefits of HSF membership are visibility and recognition** [8]. Admission to the HSF is seen as a synonym of visibility [3] or a quality acceptance certificate [4] and may recognize wide adoption by user groups [8]. Differentiating between hosted and endorsed projects [7] makes it possible to make a subtle distinction between visibility and recognition, as visibility applies for both hosted and endorsed projects, but recognition (e.g. of compliance) only applies for the latter.
2. Some White Papers also point out that **HSF membership is an acknowledgement of the value of diversity**, in the sense that small and large projects should be equally able to identify themselves with the HSF goals [3]. In other words, the HSF should recognize that “diversity fosters innovation” [7].
3. Implicitly or explicitly, **better access to knowledge** [3] is also widely seen as a benefit of HSF membership.
4. Many White Papers also imply or explicitly point out that **members benefit from support from the HSF in various areas** (although the level of support depends on resources).

Recommendations regarding HSF membership:

- *The HSF is first and foremost a bottom-up organization of individuals interested and engaged in its objectives. Membership by individuals should be open to all without restriction. A list of self-declared HSF member individuals should be maintained and public.*
- *The concept of project membership in HSF should be pursued, recognizing the benefits as enumerated in the white papers.*
- *Different levels of membership, with different requirements and levels of endorsement, review etc., should be considered.*
- *Projects cannot self-declare as HSF members. Criteria -- reasonable, inclusive, non-coercive criteria -- should be established for basic membership. Projects should be able to self-declare as interested followers of the HSF, and should have a simple means of requesting membership.*
- *In this initial startup period, the startup team should be empowered to accept projects as members, with one membership tier only: startup members. Membership in the future is subject to the future organization of HSF governance and membership tiers.*
- *Users and projects are only a subset of the stakeholders, which also include institutions, funding agencies, experiments, community organizations. These other stakeholders should be recognized and appropriately welcomed/integrated in the HSF.*

Governance

Governance of the HSF is the area where the largest differences between the various White Papers can be seen. The very detailed proposals made in several White Papers will not be repeated here and should be looked up in each paper. The following very generic points can nevertheless be noted:

1. The Geant4 White Paper [12] points out the existence of a **Collaboration Agreement** between its members. The need for a similar agreement for the HSF is explicitly pointed out by a few White Papers, but is probably taken for granted by the others.
2. Most White Papers suggest that the HSF should be a **non-prescriptive** [7] organization, i.e. one where projects retain control [9] and one that is more concerned with supporting projects than with managing them [3]. Interestingly, however, the Openlab White Paper [12] seems to go in the opposite direction by suggesting that “for the HSF to have high impact, governors might need to have an influence over the members”.
3. Most White Papers suggest that the HSF governance should be **lightweight**.
4. At the same time, all White Papers suggest that **one or more Boards** would be needed, with different roles (e.g. technical management and resource management) and varying degrees of representation of the relevant stakeholders in each type of Board. The idea of a secretariat and/or executive team has also been suggested. Most White Papers agree that software projects should be represented on these Boards. Resource providers and users are also often mentioned for some of these Boards.
5. Some White Papers explicitly suggest to have **non-HEP partners in HSF Boards**, or more specifically in one of the many Boards, for instance in the “Advisory Board” [8].

6. Each type of Board is generally associated to a **chairperson** of that Board.
7. Some White Papers advocate setting up **Areas or Technical Domains** within the HSF.
8. Some White Papers seem to suggest that **the HSF would probably not fund software projects**, or at least not initially, but this question is generally not addressed explicitly.
9. Conversely, several White Papers seem to imply that **the HSF would probably fund and staff common services** (depending on available resources), but this question is generally not addressed explicitly too.

Recommendations regarding governance:

- *The IFB -- wisely, we believe -- sidestepped the governance issue with the pragmatic, lightweight approach of establishing a startup team to get the HSF off the ground, setting aside the governance question until the HSF begins to take shape and some experience is gained.*
- *The governance question should be taken up soon only if the startup team approach is seen to be performing poorly or failing.*
- *CHEP may be the right time for a first look at governance, when some observations and lessons might be taken from HSF startup, and real needs for governance bodies may have begun to emerge.*
- *Governance should be lightweight, and should be consistent with the HSF continuing as a bottom-up community organization.*
- *HSF is more concerned with supporting projects than with managing them.*
- *There is no near term likelihood of the HSF funding and staffing software projects and/or common services, other than by leveraging the voluntary resources and efforts of HSF participants (institutions and individuals).*
- *There will surely be technical areas/domains within a successful HSF; how to reflect that in the organization should be left to experience.*

Potential activities and deliverables

Many of the recommendations above imply activities and deliverables for the HSF if they are accepted. (For some obvious high priority activities such as a website the activity is already underway.) Part of the HSF’s initial work in gathering input and talking to the various communities and projects will be to determine which garner sufficient interest and support that they should be pursued. A concrete list will be assembled (the list in Richard Mount's white paper is a good starting point) to document and track the considerations and decisions.

Proposed startup plan

The recommendations embedded in the white paper synthesis above embody the startup team’s view of how the white paper input should guide where the HSF puts its attentions and efforts in the beginning and near term. Here we summarize and prioritize the main early steps and objectives in a proposed startup plan to establish the HSF and begin to carry out the recommendations above.

- Establish an inclusive, representative startup team that will operate in an open, transparent way, as an organizationally lightweight way to get the HSF up and running. (This has been done.)
- Establish communication tools facilitating discussions between all stakeholders, including users, and provide a system for facilitating information exchange. (Initial tools have been established, in the google group forums and the website.)
- Define the initial software domain scope that will be the focused target of the core HSF effort to build software project participation. The proposed initial focus area is the analysis chain from TDAQ and generators through to analysis tools. Particularly, optimizing software for parallel software & computing and future technology challenges are priorities.
- Similarly for science communities, define the initial focus areas to target.
- Meet with as many communities in the focus software domains and science communities as possible, to gather input on how the HSF could be useful to them, and hear what they would like to bring to it.
- At the same time, be receptive to early proactive input from the wider community also. Promote and foster broad involvement in the HSF such that community engagement amplifies the core effort, and allows the community itself to tailor the scope and activities of the HSF to actual interests.
- Document the discussions, with meeting notes and with conclusions and actions arising for the HSF.
- Based on the input, develop objectives and deliverables for the HSF tailored to the declared interests of the involved communities. Maintain the evolving plan and work program as a ‘living document’ as the community interactions lead to changes and additions to the plan. Decide based on interest and support which of the potential activities and deliverables the HSF should proceed with. And/or the discussions may lead to introducing new ones.
- Organize an early face-to-face workshop (with support for remote participation) to assess progress and plans a few months into the startup, and plan the way forward. This will be the SLAC workshop in Jan 2015. The workshop will be an information gathering venue also; “listening” and not “telling”.
- Make plans to use CHEP as a second venue for a face-to-face meeting (with support for remote participation).
- Consider the longer term, post startup team organization and governance once some real experience has been accrued. CHEP might be the right timescale to discuss.

Finally, a request and appeal to communities, projects, developers, users, and individuals: please get involved and make the HSF your own. See http://hepsoftwarefoundation.org for more information and to get started.

References

[1] I. Bird and J. Harvey, Proposal for an HEP Software Collaboration, Jan 2014
[2] HEP Software Collaboration meeting, CERN, Apr 2014
[3] G. Quast et al, View of the German community on the proposed HEP Software Foundation, Apr 2014
[4] O. Smirnova et al, HEP Software Foundation – the Grid view, May 2014
[5] J. Templon et al, HEP Software Foundation [Nikhef input], May 2014
[6] R. Mount, Foundation for HEP Software, May 2014
[7] M. Jouvin et al, HEP SW Collaboration: a few ideas [IN2P3 input], May 2014
[8] P. Spentzouris et al, HEP Software Consortium [US input] , May 2014
[9] D. Britton et al, HEP Software Foundation [UK GridPP input], May 2014
[10] D. Menasce et al, HEP Software Foundation [INFN input], Jun 2014
[11] M. Asai and M. Verderi, Experiences of the Geant4 Collaboration, Jun 2014
[12] A. Nowak, The Openlab Platform Competence Center thoughts on a future HEP Software Collaboration, Jun 2014
[13] First meeting of the HSF Interim Foundation Board, Jul 2014.
[14] Fourth meeting of the HSF Interim Foundation Board, Oct 2014.
[15] HEP Software Foundation workshop, SLAC, Jan 2015.