



Lessons Learned ³

In the EICUG Quarterly Meeting on March 31, we have shared "Lessons Learned":

- From ATHENA and ECCE,
- From the EIC Software effort of the last years,
- Including our workshop on "Future Trends in Nuclear Physics Computing".

We are now proceeding with work on **one software stack:**

"Statement of Software Principles": Shared with EICUG SWG. Comments by Friday, April 29.

Today's discussion: How to organize the transition period? In particular: Cross-validation.

Planned discussions on:

- How to organize geometry and the exchange of geometry between detector simulation and reconstruction?
- Community goals and tools for software development.
- Workflow tools / frameworks for heterogeneous computing.
- **Many more (planning ongoing).**

Planned
topics



EIC Project Detector-1
"We need a real name"

Goal



1. Get consensus on the elements of the Single Software Stack
2. Allow everyone a chance to suggest specific elements
3. Concrete schedule and procedure for deciding on elements to include
 - *All major decisions completed by end of July*
 - *Usable Single Software Stack in place by October*



Procedure to follow the Critical Path

1. Publicize schedule of topics with dates of discussion and decision
2. Assign chair for each topic. Chair will be POC for the topic. Responsibilities are:
 - a. Organize discussion session agenda to include presentations from interested parties wishing to present on the topic
 - b. Publish draft list of requirements for the software being discussed at least 1 week in advance of meeting
 - c. Form list with at least one choice for the software to adopt to address the topic
 - d. Collect suggestions for modifications to the requirements list and/or the software choices list
 - e. Lead discussion on topic, starting with requirements list and the list of options
 - f. Summarize discussion in form of brief proposal to be voted on online by community
3. Presentations may be made regarding a specific decision topic, but should be communicated to discussion lead in advance for purposes of scheduling
4. Formal presentations are not required for all decision topics, however, group discussion is
5. Topic chairs will be assigned by the joint CompSW and SimQA WG conveners



EIC Project Detector-1
"We need a real name"

Critical Path

		Discussion topic(s)	Decision topic(s)
May	4	A/WG	
	11	Transition Period	Present procedure. Decide on list and order of decision topics
	18		<i>No meeting (Streaming Readout X Workshop)</i>
	25	Repository Geometry	Repository: - Location (GitHub, GitLab+Host) - Admins - Access
Jun	1	Geometry Documentation	Documentation: - Location of User documentation (wiki, repository,...) - Who will set up skeleton with list of topics (e.g. "Getting Started") Geometry: - Package (e.g. DD4HEP)
	8	Data Format	Data format - Generated events - Simulated data - Processed data (e.g. ROOT w/ specific tree format)
	15	Reconstruction Framework	
	22	Reconstruction Framework	Reconstruction Framework - Package
	29	Containerization Official buids	Containerization - platform (Singularity, Docker, multi, ...) - Supported OSES - Location of images (e.g. cvmfs) Official builds - Location (e.g. cvmfs, container image, ...)
Jul	6	Calibration DB Conditions DB	Calibration / Conditions DBs - Package - Server/Host - Access
	13	Continuous Integration	Continuous Integration
	20		
	27	Data preservation	Data Preservation - What is preserved (simulated, DSTs, ...) - Location(s) - Access (S3, xrootd, rucio, ...)
Aug	3		
	10	Distributed Campaign Workflow	Distributed Campaign Workflow - Package (DIRAC, PanDA, STAR(?), ...)



Software/Computing WG
"clever phrase here"

Google Sheet Link:

<https://docs.google.com/spreadsheets/d/1AHBCeXObBhfRj-hEQfk5Cjg1de2ljG3yPnDQWplt0lQ/edit?usp=sharing>

Exact topics and order needs to be discussed.



EIC Project Detector-1
"We need a real name"



Software/Computing WG
"clever phrase here"

We understand that this could be done much faster by fiat without the formality surrounding community input and consensus.

We understand that DWGs and PWGs would like to develop software algorithms *now* in a way that will not need to be refactored later for a different software framework.

We believe the benefit of community involvement exceeds the cost of the delay in transitioning to a Single Software Stack.