Fermilab U.S. DEPARTMENT OF ENERGY | Office of Science

HEPCloud Technical coordination

Marco Mambelli

HEPCloud Stakeholders meeting

17 February 2021

Marco Mambelli new Technical Lead

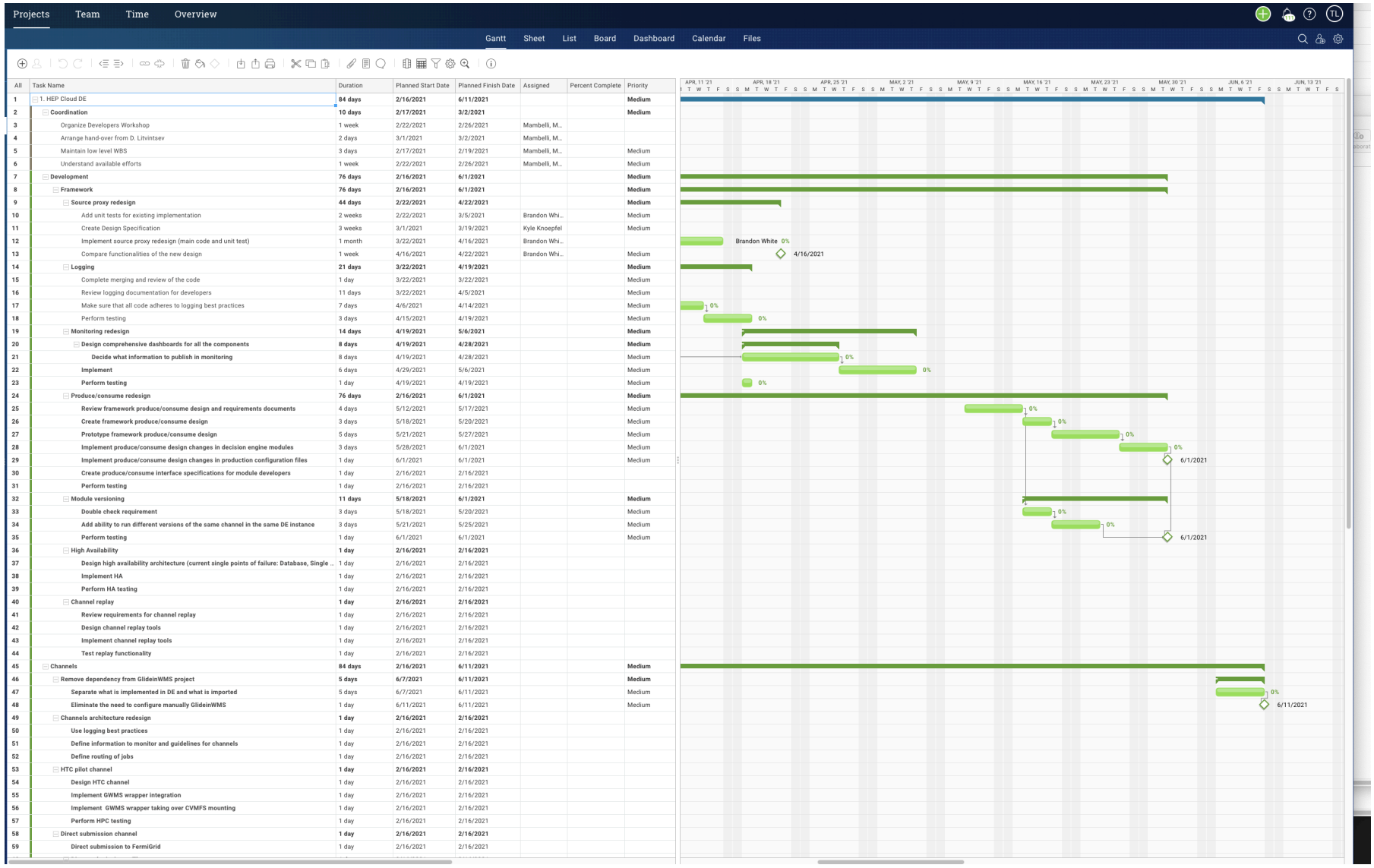
- Coordinating the hand-over to maintain continuity and proceed with the original schedule
- Operation is stable and led by Steve Timm
- Solid group of developers
- Established software development
  - Decision Engine on GitHub
  - Clear development practices
  - CI/CD pipelines
  - Improved code quality and coverage

HEPCloud Software

- Robust Decision Engine running at Fermilab
  - Production quality software
  - Reliable
  - Redesign to incorporate changed requirements and production experience
  - Flexible design to plan for future requirements

- Available to everybody
  - Code on GitHub
  - Release with working base configuration
  - Working out of the box
  - Evaluating distribution channels

HEPCloud Facility Development WBS



All	Task Name	Duration	Planned Start Date	Planned Finish Date	Assigned	Percent Complete	Priority
1	1. HEP Cloud DE	84 days	2/16/2021	6/11/2021			Medium
2	Coordination	10 days	2/17/2021	3/2/2021			Medium
3	Organize Developers Workshop	1 week	2/22/2021	2/26/2021	Mambelli, M...		
4	Arrange hand-over from D. Litvintsev	2 days	3/1/2021	3/2/2021	Mambelli, M...		
5	Maintain low level WBS	3 days	2/17/2021	2/19/2021	Mambelli, M...		Medium
6	Understand available efforts	1 week	2/22/2021	2/26/2021	Mambelli, M...		Medium
7	Development	76 days	2/16/2021	6/1/2021			Medium
8	Framework	76 days	2/16/2021	6/1/2021			Medium
9	Source proxy redesign	44 days	2/22/2021	4/22/2021			Medium
10	Add unit tests for existing implementation	2 weeks	2/22/2021	3/5/2021	Brandon Whi...		Medium
11	Create Design Specification	3 weeks	3/1/2021	3/19/2021	Kyle Knoepfel		Medium
12	Implement source proxy redesign (main code and unit test)	1 month	3/22/2021	4/16/2021	Brandon Whi...		
13	Compare functionalities of the new design	1 week	4/16/2021	4/22/2021	Brandon Whi...		Medium
14	Logging	21 days	3/22/2021	4/19/2021			Medium
15	Complete merging and review of the code	1 day	3/22/2021	3/22/2021			Medium
16	Review logging documentation for developers	11 days	3/22/2021	4/5/2021			Medium
17	Make sure that all code adheres to logging best practices	7 days	4/6/2021	4/14/2021			Medium
18	Perform testing	3 days	4/15/2021	4/19/2021			Medium
19	Monitoring redesign	14 days	4/19/2021	5/6/2021			Medium
20	Design comprehensive dashboards for all the components	8 days	4/19/2021	4/28/2021			Medium
21	Decide what information to publish in monitoring	8 days	4/19/2021	4/28/2021			Medium
22	Implement	6 days	4/29/2021	5/6/2021			Medium
23	Perform testing	1 day	4/19/2021	4/19/2021			Medium
24	Produce/consume redesign	76 days	2/16/2021	6/1/2021			Medium
25	Review framework produce/consume design and requirements documents	4 days	5/12/2021	5/17/2021			Medium
26	Create framework produce/consume design	3 days	5/18/2021	5/20/2021			Medium
27	Prototype framework produce/consume design	5 days	5/21/2021	5/27/2021			Medium
28	Implement produce/consume design changes in decision engine modules	3 days	5/28/2021	6/1/2021			Medium
29	Implement produce/consume design changes in production configuration files	1 day	6/1/2021	6/1/2021			Medium
30	Create produce/consume interface specifications for module developers	1 day	2/16/2021	2/16/2021			
31	Perform testing	1 day	2/16/2021	2/16/2021			
32	Module versioning	11 days	5/18/2021	6/1/2021			Medium
33	Double check requirement	3 days	5/18/2021	5/20/2021			Medium
34	Add ability to run different versions of the same channel in the same DE instance	3 days	5/21/2021	5/25/2021			Medium
35	Perform testing	1 day	6/1/2021	6/1/2021			Medium
36	High Availability	1 day	2/16/2021	2/16/2021			
37	Design high availability architecture (current single points of failure: Database, Single ...	1 day	2/16/2021	2/16/2021			
38	Implement HA	1 day	2/16/2021	2/16/2021			
39	Perform HA testing	1 day	2/16/2021	2/16/2021			
40	Channel replay	1 day	2/16/2021	2/16/2021			
41	Review requirements for channel replay	1 day	2/16/2021	2/16/2021			
42	Design channel replay tools	1 day	2/16/2021	2/16/2021			
43	Implement channel replay tools	1 day	2/16/2021	2/16/2021			
44	Test replay functionality	1 day	2/16/2021	2/16/2021			
45	Channels	84 days	2/16/2021	6/11/2021			Medium
46	Remove dependency from GlideinWMS project	5 days	6/7/2021	6/11/2021			Medium
47	Separate what is implemented in DE and what is imported	5 days	6/7/2021	6/11/2021			Medium
48	Eliminate the need to configure manually GlideinWMS	1 day	6/11/2021	6/11/2021			Medium
49	Channels architecture redesign	1 day	2/16/2021	2/16/2021			
50	Use logging best practices	1 day	2/16/2021	2/16/2021			
51	Define information to monitor and guidelines for channels	1 day	2/16/2021	2/16/2021			
52	Define routing of jobs	1 day	2/16/2021	2/16/2021			
53	HTC pilot channel	1 day	2/16/2021	2/16/2021			
54	Design HTC channel	1 day	2/16/2021	2/16/2021			
55	Implement GWMS wrapper integration	1 day	2/16/2021	2/16/2021			
56	Implement GWMS wrapper taking over CVMFS mounting	1 day	2/16/2021	2/16/2021			
57	Perform HPC testing	1 day	2/16/2021	2/16/2021			
58	Direct submission channel	1 day	2/16/2021	2/16/2021			
59	Direct submission to FermiGrid	1 day	2/16/2021	2/16/2021			

Fermilab

Some roadmap highlights

- Coordination
  - Arrange hand-over from Dmitry
  - Organize Developers Workshop
  - Maintain low level WBS (see below)
  - Understand available efforts
  - Evaluate tasks assignment
- Repository
  - Create HEPCloud contrib repo
  - Document how to handle security issues
- Release
  - Provide a release working out of the box
    - Provide a working base configuration
  - Evaluate release channels

Some roadmap highlights - Development

- Framework
  - Source proxy redesign
    - Add unit tests for existing implementation
    - Create detailed redesign document
    - Implement source proxy redesign (main code and unit test)
    - Compare functionalities of the new design
  - Logging
    - Complete merging and review of the code
    - Review logging documentation for developers
    - Make sure that all code adheres to logging best practices
    - Perform testing
  - Monitoring redesign
    - Design comprehensive dashboards for all the components
      - Decide what information to publish in monitoring
    - Implement
    - Test
  - Produce/consume redesign
    - Review framework produce/consume design and requirements documents
    - Create framework produce/consume design
    - Prototype framework produce/consume design
    - Implement produce/consume design changes in decision engine modules
    - Implement produce/consume design changes in production configuration files
    - Create produce/consume interface specifications for module developers
  - High Availability
    - Verify DB HA options with FNAL DB managers
    - Design high availability architecture (current single points of failure: Database, Single DE server)
    - Implement HA
    - Test HA

Some roadmap highlights – Development (cont)

- Channel
  - Dependency from GlideinWMS project (Marco)
    - Separate what is implemented in DE and what is imported
    - Eliminate the need to configure manually GlideinWMS
  - Channels architecture redesign (Marco)
    - Use logging best practices
    - Define information to monitor and guidelines for channels
    - Define routing of jobs
  - HTC pilot channel
    - Design
    - Implement
      - GWMS wrapper integration
      - GWMS wrapper taking over CVMFS mounting
    - Test
  - Direct submission channel
    - Direct submission to FermiGrid
    - Direct submission to Theta
      - Split-starter (Maria)
        - Include documentation in HEPCloud
        - Include code in HEPCloud contrib repository (under HEPCloud group)
      - Lumberjack (Maria)
        - Identify missing parts
        - Estimate work required
        - Integration (development of missing parts) and evaluation
        - Scalability test
        - Documentation in HEPCloud
      - Decide between Split-starter and Lumberjack
        - Integrate winner in production
    - Direct submission of MPI jobs

Development next steps

- Framework development
  - Logic engine and revised logging in the next release
  - Source proxy redesign
  - Producers/consumers redesign
  - High Availability
- Channels development
  - Channel Architecture Redesign
  - HTC pilot channel
    - Base already there
    - Separate GlideinWMS and DE code and configuration
  - Direct submission
    - New channel
    - Tests to Theta