Developers’ perspective and proposals

DUNE framework taskforce and developer workshop

27 June 2024

This talk is about establishing the process and setting context

Included in this talk:

- Framework requirements evaluation summary
- Prioritized discussion topics
- Workshop discussion format
- Some observations and proposals
  - **Proposal** for developers as requirement custodians
  - **Proposal** for maintaining requirements with Jama Connect
  - **Proposal** for approving requirements
  - **Proposal** for collaborative development model
  - **Proposal** for establishing milestones

This talk is about establishing the process and setting context

Included in this talk:

- Framework requirements evaluation summary
- Prioritized discussion topics
- Workshop discussion format
- Some observations and proposals
  - **Proposal** for developers as requirement custodians
  - **Proposal** for maintaining requirements with Jama Connect
  - **Proposal** for approving requirements
  - **Proposal** for collaborative development model
  - **Proposal** for establishing milestones

To make progress, the developers need a decision from DUNE on each proposal.

Those decisions should be made before the end of this workshop.

Framework requirements evaluation

- Developers have been evaluating requirements and discussing ideas since February.
- Many developer meetings with hundreds of pages of transcripts, ideas, and questions.
- The first iteration of evaluating the requirements is complete:

  https://docs.google.com/document/d/1GlnENatRxzk8ewMapL7ewCOS_OPTqWg5QUpR27_K3QE
- The next step is to "rediagonalize" the requirements according to the information learned in the first iteration.
- To refine the requirements, we are targeting specific areas:
  - Services and algorithms (meeting already held)
  - FPUs, data families, data hierarchies, etc. (meeting already held)
  - Data products and persistency, etc. (this workshop)
  - More in the future

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units In good enough shape
2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms
6. Resource requirements
7. Configurability and configuration language
8. Resource monitoring

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units

In good enough shape

2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

Workshop focus

6. Resource requirements
7. Configurability and configuration language
8. Resource monitoring

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE’s requirements regarding:

1. Flexible processing units

In good enough shape

2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

Workshop focus

6. Resource requirements
7. Configurability and configuration language
8. Resource monitoring

If there’s time

Prioritized discussion topics

Before starting development in earnest, the developers need to understand DUNE's requirements regarding:

1. Flexible processing units

In good enough shape

2. Nature of data products
3. Persistency of data products
4. Checkpointing, failures and errors
5. Backwards compatibility wrt. existing art/ROOT files and existing algorithms

Workshop focus

6. Resource requirements
7. Configurability and configuration language
8. Resource monitoring

If there's time

Strong preference: finish discussing one topic before starting another.

Workshop discussion format

The developers will share (via Zoom) discussion points for each topic.

Discussion is intended to be unstructured and organic; we'll go where we need to.

- Aim for high-level discussions except when the details become necessary.

Both on-site and remote participants should raise their hands and wait to speak until acknowledged by the moderator.

On-site participants should refrain from "side conversations".

All participants are encouraged to contribute to the live notes.

Some observations

1. DUNE task force members stayed clear of implementation details (thank you!)
2. Substantial effort toward establishing a common language to convey ideas.
   - The definitions we settle on will likely change over time…like the requirements.
3. Exchanging responses between developers and DUNE is slow. Workshops like these are best for understanding needs and clarifying concepts.
4. It is important for developers to tie requirements to documented use cases/user stories.
   - Version 0 of the requirements document includes a few use cases, but we would like to see many more.
   - At the last CM, DUNE stated its intention of collecting more use cases.

Some observations

1. DUNE task force members stayed clear of implementation details (thank you!)
2. Substantial effort toward establishing a common language to convey ideas.
   - The definitions we settle on will likely change over time…like the requirements.
3. Exchanging responses between developers and DUNE is slow. Workshops like these are best for understanding needs and clarifying concepts.
4. It is important for developers to tie requirements to documented use cases/user stories.
   - Version 0 of the requirements document includes a few use cases, but we would like to see many more.
   - At the last CM, DUNE stated its intention of collecting more use cases.
5. To facilitate progress, the developers **propose** being the *custodians* of the requirements (maintaining) with DUNE as the *owner* of the requirements and the use cases.
   - We have identified a tool (Jama Connect) that will facilitate the maintenance and tracking of requirements and their use cases.

Jama Connect

Over the past weeks we've evaluated a few options for managing requirements (Google sheets, Smartsheet, CAIRIS, and Jama Connect).

Of the options, Jama Connect meets our needs best (e.g.):

- Connect use cases to requirements
- Relate requirements to each other
- Preserve version history of requirements
- Automatically render document based on requirement entries
- Real-time analysis of requirement wording

The ACORN project at Fermilab uses this and has permitted the developers to use their license. Stakeholder/reviewer license registration can be provided for many people.

We have entered some requirements into Jama Connect to explore its capabilities (see demo).

Jama Connect

Over the past weeks we've evaluated a few options for managing requirements (Google sheets, Smartsheet, CAIRIS, and Jama Connect).

Of the options, Jama Connect meets our needs best (e.g.):

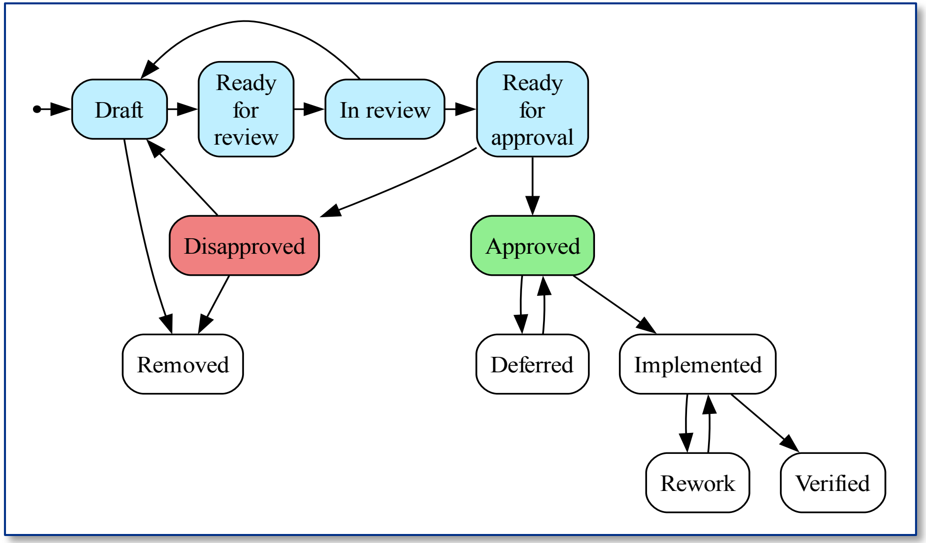
- Connect use cases to requirements
- Relate requirements to each other
- Preserve version history of requirements
- Automatically render document based on requirement entries
- Real-time analysis of requirement wording

The ACORN project at Fermilab uses this and has permitted the developers to use their license. Stakeholder/reviewer license registration can be provided for many people.

We have entered some requirements into Jama Connect to explore its capabilities (see demo).

We **propose** that the developers manage the requirements using Jama.

Proposed requirement approval process



We are developing a requirement lifecycle that should work for the long-term.

We **propose** that each requirement travels through this flow chart.

Any requirement reaching “Ready for approval” is either “Approved” or “Disapproved” by DUNE.

In the design phase, much of the time is spent in the light blue boxes, which anticipates contributions outside of the current development team.

Collaborative framework development

Fermilab has stated its intention in providing DUNE with a framework and will lead its development.

The development model is not finalized. However, we **propose** that:

1. Source code will be hosted on GitHub in an organization yet to be determined.
2. We will use a pull-request model for accepting contributions. Each pull request
    1. Must be evaluated and approved by experts of the relevant system
    2. Must pass a series of checks (style-guide conformance, unit and integration tests, etc.)
    3. Must not violate any DUNE-approved requirements
3. Contributions from all of DUNE will be especially welcome in areas where core framework designers cannot dedicate much effort (e.g. IO explorations by ANL)
4. New releases of the framework will be packaged using Spack.

Proposal for how to create and use milestones

- A baseline is a snapshot of approved requirements at a particular time, with a specific version for each requirement.
  - A baseline is a prerequisite to a milestone
- Milestones specify the set of required functionalities that are met
  - Each milestone must be associated with a baseline
  - Each milestone is verified by successful execution of tests intended to verify the requirements
- Milestones in the near future are clearer and more detailed than those in the far future
  - As time marches forward, later milestones are continually refined
  - This gives DUNE the flexibility to control the design and development process based on continually improving information and project needs.

Proposal list

By the end of the workshop we need a “yes” or “no, here’s a better option…” for each proposal:

1. Proposal for developers as requirement custodians
2. Proposal for maintaining requirements with Jama Connect
3. Proposal for approving requirements
4. Proposal for collaborative development model
5. Proposal for establishing milestones