Software Development, Deployment and Validation/Verification

Compiled by…
Giulio Eulisse (CERN)
Patricia Mendez Lorenzo (CERN)
Graeme Stewart (Glasgow)

Software Development, Deployment and Validation/Verification

This working group is concerned with:

- Software development process and methodology, that is, the way different activities and tasks (e.g., architecting, planning, coding, building, testing, issue tracking, reusing, packaging etc.) are structured and organized;
- Quality of software in ecosystem, for example, functional correctness of a code module, performance and throughput of a set of code modules, modularity and dependency in an ecosystem of APIs, codes developed in-house and adapted from open source, etc.;
- Relationship (interaction, feedback…) between software developers and operation teams, including feeding back operation monitoring data to developers
- Tools and best practices that support the above, for example, Git, code review, architectural violation warning, software synthesis and reuse, etc.
- Packaging and distribution of the software to common backend systems
- Validation and verification of the software
- Software Development (both technical and social) training and support for users
- Licensing and impact on software interoperability

Key challenges and opportunities

- Effective use of developer effort
- Knowledge sharing
- Reusing code effectively
- Validation of complex outputs
- Hardware evolution

Addressing the CWP Charge

- Strengthen and encourage the use of standard tools and methodologies
  - For development environment, build and integration
  - Continuous integration
  - Code review
- Developer training
  - In aspects of solid code design (agile based)
  - In use of common tools and techniques
  - Many opportunities for collaboration between experiments here
- Improve documentation
  - Again, use standard tools - attractive and usefully indexed ‘out of the box’
  - Try to ensure that documentation is folded into development

Addressing the CWP Charge

- Communication
  - Particularly encourage encourage communication of new ideas and tools
    - Prototype fast and fail fast
  - Community reviews give strategic directions (applies to all areas)
- Sharing
  - Good design and documentation
  - Clear copyright and license, with contribution guide
- Validation
  - Save effort by investing in standard toolkits
  - Automate to cope with platform diversity
    - Understand significance of architecture variations

Addressing the CWP Charge

- Lifecycle management
  - Ensure reproducibility of software products
  - Preserve environments (containers)
  - Planned deprecation, especially of abandoned externals
    - Refactoring tools probably help greatly
- Hardware evolution
  - Use libraries to abstract away from underlying hardware
  - Keep physics code and architecture optimisations separate
    - Need to be able to measure performance effectively
  - Track technologies - share with the community
  - Automated build systems, orchestrating many targets
- Deployment
  - CVMFS looks to remain core system
  - Containerisation favoured for more portability and independence

Practical Consideration for Progress in the WG Area

- We propose smaller focused projects that should all bring incremental improvements
- Deliver better code through better training and development practice, with documentation
    - Better code is easier to maintain
- Focus on tools and libraries that deliver improvements without making users be concurrency experts
- Enhance communication to improve sharing
    - Sharing is not free, but it can save a lot when done properly

Commonality and Leveraging S&C beyond HEP domain

- De facto starting point should be many developments and innovations outwith HEP
  - We are rarely a unique community anymore
- Challenges are to overcome
  - The ‘not made here’ philosophy
  - Share development between the interested parties
  - Sustain useful tools over the long term
  - Binding many small pieces into a working stack of software

Cross-cutting Elements

- Everything is code!
- Strong relationship between code preservation and general data preservation
- Maths libraries are a key element of our software
- Validation issues link deployment to facilities

CWP Chapter Status and Plans

What is the status of the CWP Chapter for this working group? Are the key ideas and R&D in place?

- https://paper.dropbox.com/doc/CWP-WG-Software-development-Deployment-and-ValidationVerification-3fHuGjHGETMIHv4pKQrPR
- Largely complete, would benefit from other readers and more concrete examples

What additional work is required to get the prose in good shape for a viable CWP chapter and for others outside of your WG to read and comment?

- It's ready

How do you plan to complete your chapter?

- Discuss contentions points here in Annecy, follow up as required in the next few weeks

What do you expect to accomplish by the end of this workshop?

- Clear view of the end point; review the charge and San Diego discussion to check for missing things - do we want to include case studies?

Auxiliary Material

Primary Activities

- Two group meetings
  - 18/5/2017 https://indico.cern.ch/event/639996/
  - 6/6/2017 https://indico.cern.ch/event/644485/
- A lot of collaborative work on the draft document

Software Development, Deployment and Validation/Verification

Primary organizers of the WG are:

1. Giulio Eulisse
2. Patricia Mendez Lorenzo
3. Graeme Stewart

https://groups.google.com/forum/#!forum/hsf-cwp-software-development