Design and Rationale of a Quality Assurance Process for a Scientific Framework

Hanna Remmel
Institute of Computer Science
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
remmel@informatik.uni-heidelberg.de

RUPRECHT KARL UNIVERSITY OF HEIDELBERG
Motivation and DUNE - Distributed and Unified Numerics Environment

- Focus in our research: quality assurance of scientific frameworks
- Applying Software Product Line Engineering (SPLE)
- DUNE: solving partial differential equations
  - Grid-based methods
  - Supports parallelism
- DUNE applications include
  - Fluid mechanics
  - Heat transport
  - Flow and transport processes in porous media
  - …and many more

More information: www.dune-project.org
Content

- Software Product Line (SPL) Test Strategy
- Characteristics of Scientific Software as Rationale for a Quality Assurance (QA) Process
- Contribution and Future Work
Variable Test Application Strategy for Frameworks (VAF)

- Criteria (CR) for a SPL test strategy for a framework:
  - CR1: Both commonality and the variability are tested in domain testing
  - CR2: Application testing is supported with reusable test artifacts
  - CR3: Product line applications still need to be tested in application testing

- VAF: reusable system test applications

VM = Variability Model
Characteristics of Scientific Software Development Relevant for the Design of a QA Process

- Manual literature review with over 200 papers

<table>
<thead>
<tr>
<th>C1</th>
<th>Different possible sources for a software problem. Need for Code Verification, Algorithm Verification and Scientific Validation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>Lack of test oracles.</td>
</tr>
<tr>
<td>C3</td>
<td>Most software requirements, except for high-level ones, are not known at the beginning of a software project. RQs stem from science.</td>
</tr>
<tr>
<td>C4</td>
<td>The cognitive complexity, the difficulty in understanding a concept, thought, or system, is high.</td>
</tr>
</tbody>
</table>
### Characteristics of Scientific Software Development Relevant for the Design of a QA Process

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5</td>
<td>Need for shared, centralized computing resources; high performance computing, parallelism.</td>
</tr>
<tr>
<td>C6</td>
<td>Calculations include rounding errors and machine accuracy.</td>
</tr>
<tr>
<td>C7</td>
<td>Most developers are domain scientists or engineers, not software engineers.</td>
</tr>
<tr>
<td>C8</td>
<td>There is a high turnover in the development team.</td>
</tr>
</tbody>
</table>
Carver et al.: the most highly ranked project goals

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C9</td>
<td>The most highly ranked project goals: 1. Correctness</td>
</tr>
<tr>
<td>C10</td>
<td>The most highly ranked project goals: 2. Performance</td>
</tr>
<tr>
<td>C11</td>
<td>The most highly ranked project goals: 3. Portability</td>
</tr>
<tr>
<td>C12</td>
<td>The most highly ranked project goals: 4. Maintainability</td>
</tr>
</tbody>
</table>
Design of a QA Process for a Scientific Framework

1. Planning
2. Review (desk-check)
   - Failure found? yes → 3. Unit and Integration Testing
   - Failure found? no → yes
3. Unit and Integration Testing
   - Failure found? yes → Failure found? yes → Regression Testing
   - Failure found? no → no
4. System Testing
   - Failure found? yes → Failure found? yes → Reporting
   - Failure found? no → no
5. Scientific Validation
   - Failure found? yes → Failure found? yes → Reporting
   - Failure found? no → no

Code Verification
Algorithm Verification
Scientific Validation

VAF – Rationale – QA Process – Future Work
Each developer is responsible for preparing tests for own source code

- Add/adjust/remove unit test cases
- Developers personal responsibility: thoroughly understands the source code (C4), might leave the team soon (C8)

- Advisable: Test Driven Development, since specifications mostly do not exist in advance (C3)

If mathematical requirements change

- Add/adjust/remove variability models and system test applications
QA Process Step 2: Review

- Earliest possible point to find failures
- Review all created artifacts, e.g. code, unit tests
- Review code structure and readability
  - Understandability for complex code (C4)
  - Benefit for new colleagues (C8)
  - Improves maintainability (C12)
- No structured inspection or review to keep it simple (C7)

<table>
<thead>
<tr>
<th>Rationale:</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4</td>
</tr>
<tr>
<td>C8</td>
</tr>
<tr>
<td>C12</td>
</tr>
<tr>
<td>C7</td>
</tr>
</tbody>
</table>
QA Process Step 3: Unit and Integration Testing

Together with review build the code verification part in V&V (C1)

Importance of unit tests is high
- In contexts, where system tests only run on HPC (C5)
- Alleviate the problem with missing test oracle (C2)

Rationale:
C1 Different possible sources for a software problem. Need for Code Verification, Algorithm Verification and Scientific Validation.
C5 Need for shared, centralized computing resources; high performance computing, parallelism.
C2 Lack of test oracles.
QA Process Step 4: System Testing

- Output for Algorithm verification (C1)
  - Expected output is determined analytically, if possible, and includes a tolerance range for rounding errors (C6)
  - Together with testing on different platforms significant for correctness (C9) and portability (C11)
- Suitable step for performance testing (C10)
- Together with unit and integration testing implement our SPL test strategy VAF

Rationale:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Different possible sources for a software problem. Need for Code Verification, Algorithm Verification and Scientific Validation.</td>
</tr>
<tr>
<td>C6</td>
<td>Calculations include rounding errors and machine accuracy.</td>
</tr>
<tr>
<td>C9</td>
<td>The most highly ranked project goals: 1. Correctness</td>
</tr>
<tr>
<td>C11</td>
<td>The most highly ranked project goals: 3. Portability</td>
</tr>
<tr>
<td>C10</td>
<td>The most highly ranked project goals: 2. Performance</td>
</tr>
</tbody>
</table>
QA Process Step 5: Scientific Validation

- Third step in V&V for scientific software (C1)
- How accurate is the simulation (C9)
- Mostly no analytical solution available (C2)
  - Developers decide based on domain knowledge (C4), whether the simulation result is as expected
- System test environment compares graphical simulation output files
  - Consider rounding errors and machine accuracy (C6)

Rationale:

<table>
<thead>
<tr>
<th>C1</th>
<th>Different possible sources for a software problem. Need for Code Verification, Algorithm Verification and Scientific Validation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C9</td>
<td>The most highly ranked project goals: 1. Correctness</td>
</tr>
<tr>
<td>C2</td>
<td>Lack of test oracles.</td>
</tr>
<tr>
<td>C4</td>
<td>The cognitive complexity, the difficulty in understanding a concept, thought, or system, is high.</td>
</tr>
<tr>
<td>C6</td>
<td>Calculations include rounding errors and machine accuracy.</td>
</tr>
</tbody>
</table>
Contribution and Future Work

**Contribution**
- VAF – a SPL test strategy for frameworks
- Special characteristics of scientific software as rationale for the
- Design of a QA Process for a scientific Framework

**Future Work**
- Fully implement QA Process
- Make reusable test applications available for DUNE users
- Evaluate the feasibility and acceptance of the QA process with a Case Study
Hanna Remmel
Institute of Computer Science
Chair of Software Engineering
Im Neuenheimer Feld 326
69120 Heidelberg, Germany
http://se.ifi.uni-heidelberg.de
remmel@informatik.uni-heidelberg.de

RUPRECHT KARL UNIVERSITY OF HEIDELBERG