H2020 FETHPC-1-2014

Enabling Exascale Fluid Dynamics Simulations
Project Number 671571

D5.2 – Quality Assurance Plan

WP5: Management

Document Information

Deliverable Number	D5.2
Deliverable Name	Quality Assurance Plan
Due Date	31/12/2015 (PM3)
Deliverable Lead	KTH
Authors	Erwin Laure, KTH
Responsible Author	Erwin Laure, KTH, erwinl@pdc.kth.se
Keywords	Project Management
WP	WP5
Nature	R
Dissemination Level	PU
Final Version Date	22/12/2015
Reviewed by	David Moxey, IC Jan Hesthaven, EPFL
MGT Board Approval	22/12/2015

Document History

Partner	Date	Comments	Version
KTH	11/11/15	Frist draft	0.1
KTH	1/12/15	Added software process information and KPIs	0.2
KTH	4/12/15	Version for internal review	0.3
KTH	14/12/15	Final version for PMB review	0.4
KTH	22/12/15	Final version for submission	1.0

Executive Summary

This deliverable describes the quality plan, key performance indicators used to measure the progress of the ExaFLOW project, as well as data management plan and lists the internal procedures and metrics that each participant will follow throughout the project to assure high-quality results.

The quality plan addresses all Work Packages, while particular attention is given to the deliverable and publications review processes. It covers the quality assurance strategy, the deliverable, reports and publication review process, the list of key performance indicators for each of the Work Packages and the monitoring, risk management, and software quality process. We also describe the software quality procedures applied in the project and the project's data management plan.

This document is mainly intended for internal reference and use, i.e. by all project participants; however it could also act as a reference for other projects to define their quality plans.

The following points have been agreed among the project partners:

- ✓ A quality procedure for the project's deliverables and for monitoring the project's metrics has been defined.
- ✓ The procedures for ensuring software quality have been detailed for all code developments in the project.
- ✓ An initial data management plan has been devised.
- ✓ The quality plan is mainly used by the Executive Board (EB) comprising the Work Package leaders and coordinated by the project manager. Items related to quality assurance are dealt with as part of the EB meetings.
- ✓ The document builds on D5.1 – Management plan that details the management structure of the project.
- ✓ Each deliverable is planned in advance, and reviewed by two other partners not involved in the writing process; the project deliverables should be ready roughly three weeks before the submission deadline, so as to allow for the review and approval process.
- ✓ Metrics for each Work Package have been specified in this report. Metrics will be monitored monthly by the EB and reported in the project reports.
- ✓ A risk management strategy summary is also incorporated in this report, which is based on the initial one provided in the DoW, and will be periodically reviewed.

Contents

1 Introduction

The objective of the quality plan deliverable is to define the processes, plans and metrics that shall apply throughout the ExaFLOW project in order to monitor the activities, to identify and eliminate potential risks, and to ensure the successful execution of the project. Deliverable review procedures, reporting timelines and templates are also defined to guarantee the quality aspired for. The guidelines involve all Work Packages.

The remainder of this document is organized as follows:

- Section 2 specifies the quality assurance action plan.
- Section 3 summarizes the risk management aspects identified in the project plan.
- Section 4 lists the Key Performance Indicators (KPIs) identified for the project.
- Section 5 details the project's data management plan.

2 Quality Assurance Action Plan

Quality Assurance coordination is a dedicated task within WP5 (T5.3) in order to ensure monitoring and assessment of project progress, possible risks, and timely implementation of milestones. It also guarantees that common practices and procedures are adopted by all Consortium members.

The quality assurance activities are clearly identified in the DoW and focus on the quality of deliverables and quality of software developed within the project. The Project Manager (PM) is responsible for the overall coordination of quality assurance activities and each WP leader is responsible for implementing the quality process in their WP.

2.1 Deliverables and Publications Quality Process

All publications of the project, particularly deliverables and periodic reports, but also scientific and other publications need to be correct and of the highest quality.

2.1.1 Deliverables and official reports

For deliverables, and other official reports that need to be submitted to the EC, quality will be ensured through an independent review process. Each deliverable will be reviewed by *two consortium members not directly involved in the work* reported in the deliverable, or in the writing of the deliverable.

The DoW defines the due dates and responsible partner for each deliverable. The partner responsible for a deliverable has to prepare the deliverable in due time to allow for the independent review process. The following timeline lists the steps involved in the deliverable and report quality process (see also Figure 1):

- The PM assigns two independent reviewers 1 month before the due date.
- The partner responsible for the deliverable submits the deliverable to the PM and reviewers at least 3 weeks before the due date. In case of problems meeting this deadline, the partner has to inform the PM as soon as possible to define a contingency plan and, if needed, initiate a discussion with the project officer.
- The reviewers submit their comments at least 2 weeks before the due date.
- The partner responsible will duly consider the comments from the reviewers and submits the final draft to the PMB for approval at least 1 week before the due date. The PMB may raise comments within 4 days; silence is considered approval of the deliverable.
- The Coordinator prepares the final version and submits the deliverable to the project officer on the due date.

Figure 1 - Deliverable Process

A recommended timeframe for developing deliverables is to start with a table of contents around two months before the deadline. It is also recommended that the partner responsible identifies an editor who will manage the writing of the deliverable.

Templates in Microsoft Word and Latex are made available and must be used to prepare the deliverables and reports.

2.1.2 Scientific publications and additional output

For scientific and other publications the Consortium Agreement (CA) defines the applicable procedures. We specifically require partners to provide notice of planned conference or journal submissions in due time before submission deadline. Any objection to the planned publication shall be made in accordance with the CA in writing to the Coordinator and to any Party concerned. If no objection is made within the time limit stated above, the publication is permitted. Further details can be found in Section 8.3 of the CA.

All publications need to acknowledge the funding through EC by adding the following statement in the acknowledgements:

- *This work was co-funded by the European Commission through the ExaFLOW project (grant agreement no 671571).*

2.2 Milestone quality process

The DoW lists a number of milestones with clearly identified due dates and partners responsible. When a milestone has been reached, the partner responsible needs to inform the EB and provide sufficient documentation that proves that the milestone has been reached. This documentation will subsequently be included in the next periodic report.

2.3 Software quality process

Software development takes place in the context of the pilot codes used in the project. All of these codes have already established software quality processes, which are described below, and ExaFLOW intends to re-use those processes, rather than defining and implementing new ones. These processes follow best practices of software development and include particularly the following aspects:

- All software needs to be available to all project partners in repositories using a version control system.
- All software needs to be documented within the code, contain release notes and basic test cases verifying the correctness of the software for most common cases.
- Sufficient installation and usage documentation needs to be provided such that every partner can install and use the software.

We now provide further details on the software processes applied by each package:

2.3.1 Nek5000

Code Access

The code is available for download and installation from either a Subversion (SVN) Control Repository or Git. Links to both repositories are given at:
http://nek5000.mcs.anl.gov/install/

Contributor Access

The Git repository always mirrors the SVN. Official releases are not in place since the Nek5000 community users and developers prefer immediate access to their contributions. However, since the software is updated on constant basis, tags for stable releases as well as latest releases are available, so far only for the Git mirror of the code. The reason for this is that SVN is maintained mainly for senior users who already have their own coding practices, and will be maintained at Argonne National Laboratory (using respective account at ANL); the Git repository is maintained at GitHub. A similar procedure is followed for the documentation to which developers/users are free to contribute by editing and adding descriptions of features, and these are pushed back to the repository by issuing pull requests. These allow the Nek5000 team to assess whether publication is in order. All information about these procedures are documented on the homepage http://nek5000.msc.anl.gov/.
KTH maintains a close collaboration with the Nek5000 team at ANL.

Quality Control
The code is daily run through a series of regression tests via the continuous integration platform buildbot (to be transferred to Jenkins). The checks range from functional testing, compiler suite testing to unit testing. So far not all solvers benefit of unit testing but work is ongoing in this direction. Successful runs of buildbot determine whether a version of the code is deemed stable.

A suite of examples is available with the source code, which illustrate modifications of geometry as well as solvers and implementations of various routines. Users are encouraged to submit their own example cases to be included in the distribution.

The use cases within ExaFLOW, which involve Nek5000, will be packaged as examples and included in the repository for future reference.

2.3.2 Nektar++

Code Access
Stable code releases for Nektar++ can be found on the Nektar++ website (www.nektar.info). However, since Nektar++ is usually under heavy development, use of the most recent version from the Git repository is recommended. The following resources outline getting access to the repository and the code:

- Detailed instructions on getting access to the repository: https://www.nektar.info/getting-started/.
- As a quick start, the code can be checked out anonymously from the Git repository: https://gitlab.nektar.info/nektar/nektar.git
- The main project timeline detailing commits, open merge requests and issues: https://gitlab.nektar.info/nektar/nektar

Contributor Access
Contributor access is maintained through our GitLab instance. Contributors should visit https://gitlab.nektar.info/ and use the signup form to create an account. Log in with these credentials (via the *Standard* tab, not the *LDAP* tab), and add your SSH key(s) in the account settings. A dedicated group and a forked copy of the repository has been established for ExaFLOW development purposes, which can be found at https://gitlab.nektar.info/exaflow/nektar. This is only accessible once you have been added to the ExaFLOW group. To request contributor access to this repository, your username should be sent to David Moxey (d.moxey@imperial.ac.uk). Any development efforts and contributions made as part of ExaFLOW can then be merged back into the main repository through a GitLab merge request.
Conventions/git workflow
The latest development version of the code can be found on the master branch of the main repository. Code is developed in branches that are prefixed with specific names:

- feature/*: branch adds new functionality

- fix/*: branch fixes an issue in the code
- ticket/*: branch fixes a ticketed issue

Nektar++ also has coding styling guidelines, which can be found in the developer guide (note that this is very much a work in progress).

Compilation instructions and functionality
Instructions for compiling/installing Nektar++ can be found in the user guide (link on the project website). The user guide also outlines the functionality of each of the solvers with some guided examples. Each solver also has a *Tests* directory that contains the testing session files, that also double up as examples.

Quality Control
Nektar++ has an automated buildbot that tests the framework (i.e. its constituent library and solvers) against a number of 32- and 64-bit machines on Linux, OS X and Windows. This can be found here: http://buildbot.nektar.info/. When code is ready to be merged back into the repository, a merge request is issued by the branch author. The branch should, at a minimum, include:

- Code that adheres to the coding guidelines.
- Appropriate comments, including doxygen for new functions and variables.
- Tests that cover the new functionality and indication that all tests pass on buildbot.
- Documentation in the user and developer guides where appropriate.

All merge requests will be reviewed by 1-2 developers, depending on the size of the request, and a senior developer before it is merged into master.

2.3.3 SBLI

Code Access
The SBLI code in its current form is not an open-source code. An open-source version of SBLI is under development and ExaFLOW-related work will be incorporated into this.

Contributor Access
Over the past 3 years SBLI has been maintained using SVN in a local server at the University of Southampton. The user base is not large and significant code modifications were typically communicated by e-mails to users based outside of the University of Southampton. The SVN repository was recently converted to a private Git repository hosted on bitbucket.org, where all users have direct access to the latest releases.

Quality Control
A test suite was developed as part of an earlier software engineering project that brought the code up to the f-95+ standard. The full test suite is only applied to major releases. A subset of test cases are more routinely run during code development. Users are responsible for code verification and validation, which is a routine part of the research that is carried out (since most problems being run

are new cases that haven't been run before). A manual exists, primarily aimed at new users.

2.3.4 NS3D

Code Access

The NS3D code in its current form is not an open-source code.

Contributor Access

The source code resides in a subversion (SVN) repository at IAG, University of Stuttgart, together with a set of external I/O-libraries as well as pre- and post-processing tools. The solver and libraries constitute a self-contained software package to be run independently from other software (except for visualization). Code development, so far, takes place at IAG exclusively with each developer/user having full access to the repository.

Quality Control

For testing purposes a standard template solution (transitional flow in a ZPG flat-plate boundary layer) is checked for convergence after each submission. Extensive documentation resides at https://wiki.iag.uni-stuttgart.de/transiwiki/ (restricted access) including underlying theory of the method, best practice guides and data to run above mentioned test case as a tutorial.

3 Risk Management

All Work Packages have a suitable management structure defined, which allows for early detection and resolution of issues within the Work Package. Issues that cannot be resolved within the WP will be forwarded to the EB. Issues not resolvable within the EB will be forwarded to the PMB for resolution.

The risk management process involves four tasks with clearly identified responsible persons:

1. Risk identification (prior and during project duration) - WP leaders
2. Risk monitoring - WP leaders coordinated by the PM
3. Risk analysis, evaluation and prioritization – PM
4. Risk mitigation and damage recovery – EB and PMB

Risk identification will be performed mainly by WP leaders; however risks can be identified by other PMB and/or SAB as well, mainly in case of risks with external causes. The first set of general risks has been identified during the proposal preparation and project negotiation. Other risks might be identified during the course of the project.

The WP leaders will report on their respective risk status periodically in quarterly reports. However, risks that require immediate action will be reported during the EB bi-weekly teleconferences where a risk analysis, evaluation and

prioritization is initially performed; for important risks with high impact the PMB will be informed in order to assess the situation and take preventive measures to mitigate the risk. In case the risk cannot be mitigated, the PMB will perform damage assessment and recovery.

Table 1 lists the risks identified during the preparation stages of the project.

Risk number	**Description of Risk**	**WP**	**Proposed risk-mitigation measures**
R1	Partner is not competent to carry out allocated tasks.	5	Partners have been carefully selected based on different required expertise (high performance computing, CFD, software development, scientific & industrial applications), track record in their field (number of scientific publications and citations for research partners; level of innovation; academic excellence) and balance of the consortium (mix of expertise in HPC and CFD). The consortium agreement includes measures to be taken if a partner still would not deliver for some reason, such as replacement of this partner by another one and a corresponding budget reallocation.
R2	Novel algorithms show negligible scaling benefit over current methods	1	The proposed methods have been identified to overcome the main challenges identified by the broader scientific community. It is possible that one of the algorithms to be developed in the project shows negligible gain in practice but it is highly unlikely, given the knowledge and experience of the partners, that the project will not make significant advances towards the requirements of exascale.
R3	No algorithm found that leads to substantial data reduction	1	Case- or application-dependent data reduction can be expected. Work will continue on one of these.
R4	Embedding the new algorithms requires drastic changes to the codesign application and are therefore not implemented fully	2	It is fairly likely that the new algorithms will require significant modifications in the co-design applications, which can have an impact on development time. It is important that the algorithms are implemented without compromise and the proposed effort levels take this risk into account.
R5	Disruptive technologies in hardware for which our methods are not suitable	2	ExaFLOW software will be designed and implemented to fully exploit exascale computer architectures that will resemble in large part today's most common computer architectures.

			It is possible but unlikely that new disruptive technologies are introduced during the duration of the ExaFLOW project. In order to appropriately react to them a technology watch function as been defined in WP2.
R6	Computer hardware not available to test adequately the codes for production runs on architectures representative of exascale	3	Leading European systems are already included in the proposal and applications to PRACE, XSEDE, and other programs that will have suitable systems available are planned in milestone 4.
R7	Project will not reach TLR6	1,2, 3,5	The codes targeted by the project are already now being widely used in academia and to some extent even in industry. They thus provide a perfect vehicle to implement the new concepts developed by the project and quickly reach TRL5. With the help of our industrial partners the developments will be field tested in their respective environment thus reaching TRL6. The close feedback loop maintained between WP1, WP2, and WP3 will ensure that this is indeed possible and our partners have already several years of experiences in working with the industrial settings of our commercial partners.

Table 1 - Risk Analysis

4 Key Performance Indicators

To monitor the project progress, a number of Key Performance Indicators (KPIs) have been identified, for each WP and the project overall. As ExaFLOW is concerned about algorithmic development and efficient implementations, it is for most KPIs not useful to define an absolute target but rather report the development relative to the baseline established at the start of the project. Table 2 lists those KPIs. The EB will monitor the progress of the project against these KPIs on a monthly basis and this will be reported in the project reports.

WP	KPI	Target
WP1	Reduction of simulation time due to error control and adaptivity	Up to 50% in case of non-optimized meshes
WP1	Increase scaling due to combined CG-HDG approaches while maintaining absolute performance	Up to 50%
WP1, 2	Improved I/O schemes	I/O time significantly reduced as compared to

		non-optimized (/O schemes
WP2	Data compression	Reduced amounts of I/O using data compression as compared to uncompressed data representation
WP1,2&3	Strong and weak scaling of pilot codes using pilot use-cases	Significant improvements as compared to baseline
WP2	Energy to solution	Reduced energy to solution as compared to baseline
WP4	Dissemination KPIs are defined in D4.6	
WP4	Exploitation KPIs are defined in D4.1	

Table 2 – KPIs

5 Data Management Plan

The ExaFLOW project is focusing on algorithmic and method development thus production of research data is limited to benchmark and performance data. The data produced by the use cases have, with exception of the commercial use cases, will be published using the conventions applicable by the problem owners. For the large-scale use case that is intended to be performed in the third project year, the repository and publication convention of the respective problem owner will be used as well.

The benchmark and performance data will be published likewise and made publicly available in a robust and reliable way using established e-Infrastructure services, like the ones provided by EUDAT (currently, we plan to use EUDAT's b2share service). As the production of research data is not in the centre of the project, ExaFLOW will not participate in the open data pilot.

6 Outlook

In order to assure successful project implementation and constant progress the practices described above will be pursued during the whole course of the project. The progress of the results and implementation of the main goals will be monitored through the processes described in this deliverable, particularly through the KPIs, Deliverables, and Milestones defined in this deliverable and the DoW. The detailed analysis of possible risks both before the start and during the project will help to eliminate these risks and minimize their damages. The well-defined organizational and collaborative structures between partners and Work Packages will simplify the quality assurance coordination.