Mantid SSI review

# Responses to the Prioritised Recommendations:

## Introduction

### Purpose of this Document

This document contains the response from the mantid project, changes and actions in response to the Technical [TR] and Collaborative Reviews [CR] conducted by the Software Sustainability Institute. This document will be structured as a response to the prioritied recommendations [PR] provided by the Software Sustainability Institute.

### References

|  |  |  |
| --- | --- | --- |
| Document | Ref | Details and Version |
| Collaborative Review | [CR] | <https://github.com/mantidproject/documents/blob/master/Project-Management/reports/SSI2017_CollaborationReview.pdf> |
| Technical Review | [TR] | <https://github.com/mantidproject/documents/blob/master/Project-Management/reports/SSI2017_PrioritisedRecommendations.pdf> |
| Prioritised Recommendations from the reviews | [PR] | <https://github.com/mantidproject/documents/blob/master/Project-Management/reports/SSI2017_PrioritisedRecommendations.pdf> |

### Structure

For the rest of the document we will use colour to differentiate sections extracted from the prioritised recommendations [PR] which will be in purple, from our responses in black.

## Collaborative Review

1. The stability of Mantid (in terms of how often the software unexpectedly exits from a task) needs significant improvement and is currently the greatest barrier to better integration with working processes. The Collaborative Review identified a number of areas for improvements in stability and addressing it, including postprocessing using Mantid algorithms, a need for robust fitting routines and improved system tests. The commissioning of a survey or similar activity to identify the precise ways in which this aspect needs improvement and the demand for each aspect could be undertaken initially.

Stability is one of the highest priorities given from the users workshops and other user groups. This has been prioritised within the development team, we have or are in the process of implementing the following improve the stability of Mantid:

* All code changes are reviewed and tested by another developer
* For hard to reproduce problems we have installed development versions of Mantid with specific instrument scientists to allow diagnostics of any crashes.
* We are recruiting a member of the team to focus on support and reliability testing.
* The highest proportion of stability problems originate in the GUI layer, the current plan to redevelop this as part of Mantid 4 includes designing for and building automated testing into the interface.
* Long term development branches will be used for significant refactoring work, ensuring that code that is not considered complete is not merged into the deliverable code base (There may be considered exceptions were benefits outweigh the risks).
* We are now offering early access to the beta test period for instrument scientists that want more time to test the release than the standard 2 weeks.

2. Increase the level of communication between the Scientific Steering Committee and end users, including greater transparency through public dissemination of request and decision making procedures and the publishing of meeting minutes is strongly recommended. This should increase the awareness and involvement of frontline scientists and other users in key decisions, and working towards a public roadmap for future releases should also be considered.

We have appointed a Chair of the Scientific Steering Committee, to oversee this area, this has already led to several changes in the structure and running of the annual users workshop.

3. Increase the effort assigned to managing and resolving technical issues. Instituting regular ‘bug scrub’ sessions, and/or issue reprioritisation meetings between each scientific group and assigned Mantid developer(s), would help to clear older issues that are no longer relevant whilst highlighting those that are still important. This could be once a month or quarterly, with representatives from each group attending. These representatives could also be responsible for monitoring outstanding issues within their local group and reminding people to clear them up.

One of the tasks for every developer after each release is to clear older issues that are no longer relevant. Furthermore the move to the new GUI with Mantid 4.0 will allow many of the issues associated with the previous interface to be archived.

Most developers are associated with a specific group of instrument scientists and work closely with them to prioritise the work to be completed.

4. Provision of better documentation and support, particularly for new users. This should include more clearly structured and discoverable documentation, particularly on the website. More technique specific documentation, as well as increased detail on the expected behaviour of algorithms, is required. The development of more worked Mantid examples and tutorials that can be readily applied to scientific activities, are strongly recommended, perhaps leveraging the existing use of Sphinx for these aspects. Inapplicable, outdated, unused and deprecated documentation should also be identified and pruned, to avoid conflicts and make it easier to find the right documentation. It should also be made clear to which software/component versions each documentation applies.

We agree with the need for more technique specific documentation, and have identified two areas to focus initial development of courses on, Powder Diffraction and Direct Inelastic spectroscopy, These are being followed up on in the Mantid PMB.

Pruning of old out of date documentation is done as it is discovered by developers or reported by users. In the future if resources allow we could dedicate some time to a wider documentation review.

Documentation provided with each version of Mantid is specific to that version, and on the website the correct documentation set for each version is also available. However if you go to the web documentation it is not currently visible which version the documentation applies to, other than in the url. This is something we would like to address.

5. The project should put more emphasis on community building. At the moment, there is not much external evidence of an active user community. The website could include successful use cases, quotes from satisfied users and success stories. This would not only increase engagement of the current users, but also potentially make the project more attractive to prospective users (as well as potentially providing a foundation from which to collect data on the impact of Mantid, for justifying future funding). Other options could include organising regular webinars, Q&A or other online sessions for users. Once option would be to assign the task of undertaking community building to an individual with an outreach and dissemination skillset.

At the users workshop this year we included talks from Instrument scientists about Work they have done with Mantid and the science that has made possible. This might be a good starting point to capture some of this form of case study, together with examples from non-partner facilities such as ANSTO and HZB.

6. Consider embedding Mantid developers on experiments in situ for a period of time. This would help them increase their understanding and appreciation for the underlying barriers faced by beamline scientists, and to provide in situ training on using Mantid for their experiments to supplement existing training Mantid courses. Having Mantid developers present on courses to determine where their users struggle with their software, which is often incredibly valuable. If this were done for each experiment, this would have the added benefit of developing an even better foundation of scientific operational understanding across the Mantid developer team to inform design and other development decisions, and potentially allow instrument specific support liaisons between the Mantid team and the scientists.

Since the review was conducted we have made significant progress in this area. For developers that are aligned with an specific technique we encourage them to visit or join experiments to see the real life usage of Mantid. This is backed up with visits to the instrument hall during experimental cycles.

7. Improved support for easier and simpler plot generation and exporting of quality plots in Mantid. The Collaborative Survey also identified the need for easier integration of various plots, easier line zooming, and more direct access to plot inspection options. The commissioning of a survey or similar activity to identify the precise ways in which this aspect needs improvement and the demand for each aspect could be undertaken initially.

This feedback is being fed into the design and user interaction planning of the new Mantid workbench that will form part of Mantid 4.0.

8. Effort should be made to improve the data access and processing speeds of Mantid, to reduce time to results. The Collaborative Review identifies faster on the fly analysis, improved speed for legacy instruments, speed of Catalog when selecting multiple files, and faster SANS reduction. The commissioning of a survey or similar activity to identify the precise ways in which this aspect needs improvement and the demand for each aspect could be undertaken initially.

Performance is a significant concern within the development of Mantid. We have developed and use a suite of performance tests that highlight any performance regressions within Mantid, and those are shared with the developer responsible on a timely basis. Further work on this is planned to make the data easier to view and manage over time.

Work to improve performance farther is prioritised alongside other work to fit within the available development resources.

Fundamental changes within the framework to better support distributed heterogeneous computing is being led by the ESS team and may lead to a path for performance improvement for certain areas.

9. Consider instituting official end user or scientific champions across the wider collaboration, for specific techniques, algorithms or documentation, for example. Acting as a partially ‘devolved’ spokesperson and coordinator for their local group, they collate, channel, and regularly reprioritise requests and issues at a technical level.

This is an interesting option that we would be very open to, and would certainly support anyone that offered their time in this regard. We do have some examples already with for example Duc Le championing the mSlice and crystal field areas and documentation.

10. Consider a general annual user meeting across the facilities (or within each facility) to discuss development progress, promote networking and a sense of shared community, identify ways the collaboration could improve, and collect and discuss requirements for road mapping at a wider scale. A single yearly PMB meeting that is open to a greater number of representatives could also be considered, perhaps coupling it with, or holding it prior to, the annual user event. Prioritised Recommendations:

We considered involving / inviting users to the Users Workshop this year. The view gathered from the instrument scientists was that the take up and interest of the end users would be patchy at best, and that those that did attend would be mostly interested in getting “free” development resource for their own personal area of development, rather than the wider usage at facilities. We plan to reassess this in future years. We would also need to consider the costs to the facility hosting the workshop for that year as well.

## Technical Review

1. At a higher level and for the medium to longer term, investigate how the team’s development operations could be improved or restructured to continue to efficiently address the changing needs and scale of work arising from end users. Common challenges with projects growing in capability and demand are to remain responsive to requirements as the operational, communication and management overheads increase, and to maintain a clear, cohesive, planned approach to dealing with the needs of each end user group, as opposed to a piecemeal issue by issue approach. As such, the survey comment that recommends that Mantid should be recognised as a programme of work and not just a project is fully endorsed and should be investigated.

Since the review we have implemented a restructuring of the ISIS development team into sub teams aligned with experimental approaches. This has provided clarity in the communication paths with instrument groups as well as a local resource pool for their requirements. Larger tasks such as the work towards Mantid 4.0 lead to the creation of a further sub team focussed on that particular goal.

It is worth considering that the long term maintenance and support of Mantid are mission critical to the facilities relying on the software, and defining those aspects as a programme of work rather than a project may be more correct. We have therefore created a "Core design” team to prepare and propose terms of longer term maintenance and support work to the PMB for consideration and resourcing.

2. Increase the effort available to manage and resolve longer term technical issues (also noted in Collaborative Review recommendations). In particular, greater centralised backing and drive from management to address these issues as well as stale branches and pull requests is needed, particularly for those general issues that transcend facility boundaries and often are at risk of deprioritisation due to local facility demands. This could be achieved via facilities assigning a fraction of effort to ‘core’ development aside from facility, instrument, and other local user needs for maintenance of the core parts of the framework.

Driven by the work for the ESS we have improved our processes for handling larger scale changes within the code base.

Increasing the effort to resolve strategic and or maintenance issues will either require additional resources or come at the expense of other developments. We currently reserve 2 weeks per release for all developers to work on “maintenance” issues, and in addition longer tasks are assigned and carried out alongside other development. This currently works out as about 15% of development time not including the separately scheduled pieces of work. This is a good baseline of maintenance effort, but there will be larger tasks that need to be resourced separately. The recently created "Core design” team is tasked to prepare and propose terms of longer term maintenance and support work to the PMB for consideration and resourcing beyond this baseline.

3. Increase developer effort to improve the overall structure and maintainability of the Mantid codebase, supporting the team’s current activity to improve framework level updates. Maintainability and inherent technical debt is currently highly variable across Mantid’s constituent components and is consuming an increasing proportion of developer effort. This would inevitably require a conscious rebalancing of the development effort between new feature development and issue resolution.

As discussed in the answer to 2. We have created a process for consideration and resourcing of maintenance tasks that exceed the pre-approved baseline maintenance effort.

4. Improve the processes associated with issue management, in particular for prioritisation of longer term issues that often end up neglected, and develop well defined threshold criteria for escalation of significant issues to higher authorities across the team.

It is important to differentiate between long term issues that are neglected as they are low priority, and those that have been neglected as they require a greater level of resource. For the former we need to be clear that we are not working on those issues, but if we are correctly prioritising the work we should not worry about them further. For the latter these need to be described as significant pieces of work and proposed either within a facility or the Mantid PMB as a sub project for the benefits and costs to be assessed and discussed.

5. Improve communication between developers and end user scientists to improve understanding and appreciation of requirements, and to identify opportunities for shared development work to avoid duplication of effort and an unnecessarily divergent codebase. Increase the contact between these two groups, through short term in situ developer placements with the scientists, and more face to face discussions which are overwhelmingly seen as the most efficient means of user communication.

We already align developers with instrument scientists, and have improved this domain grouping of knowledge recently in a train restructuring at ISIS using sub teams. The developers have all given access to the experimental halls to aid with in situ placements and ad-hoc discussions.

6. Improve developer documentation, especially more tutorials and worked examples that apply to common user scenarios including documentation (also noted in Collaborative Review recommendations see for more detail).

The developer documentation is an area that has had a lesser amount of attention in recent years, with the user documentation having a greater emphasis. We should consider consolidating and improving the developer documentation as a priority for future maintenance work.

7. Improve the stability of Mantid by prioritising maintenance development effort over new feature development (perhaps periodically) to investigate and reduce such issues (also noted in Collaborative Review recommendations see for more detail)

See the answer to number 2.

8. Improve speed of data processing and data access through Mantid (also noted in Collaborative Review recommendations see for more detail).

See the answer to 8 in the collaborative review.

9. Improve the level and quality of code commenting across the codebase, prioritising areas where it is acknowledged to be most needed. The commissioning of a small poll or similar activity to identify the precise ways in which this aspect needs improvement and the demand for each aspect could be undertaken initially.

It is clear from metrics and discussions with the development team that the Mantidplot GUI code is the hardest to understand, and due to its original design hardest to test.

The redevelopment of the user interface in Mantid 4.0 is planned to address both of these issues.

10. Consider holding intermediate and/or technique specific training courses for end users, prioritising those subjects of highest interest from a commissioned end user survey.

As has been discussed at the PMB technique specific training courses are a clear requirement, but the development team would not be the best people to either write or deliver these courses. We are working with instrument scientists to provide support and help them create such training material, and excellent example of this is the Muon moodle based training material.

A good suggestion was made by a few people at the last Users Workshop for an “Advanced” training course to focus on the Python API of the workspace and how best to access and modify the data within it both using Mantid and external libraries.