Software
Sustainability
Institute

SSI/Mantid - Prioritised Recommendations

Project Title	SSI/Mantid
Authorship	Steve Crouch
Contributors	Mario Antonioletti
Document Version	1.0
Date	28/02/17
Distribution Classification	Project-Internal
Distribution List	*Project Management Board; Project Team*
Approval List	*Project Management Board*

Revision History

Date	Comment	Author	Version
17/02/17	Initial complete draft	STC	0.1
28/02/17	Incorporate review comments from MAA	STC	1.0

Contents

Prioritised Recommendations: 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 areas for improvements in stability and addressing it, including post-processing 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.
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 front-line scientists and other users in key decisions, and working towards a public roadmap for future releases should also be considered.
3. Increase the effort assigned to managing and resolving technical issues. Instituting regular ‘bug scrub’ sessions, and/or issue re-prioritisation 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.
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.
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.
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.

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