Astronomical Data Analysis Software and Systems XXVI
ASP Conference Series, Vol. 521
Marco Molinaro, Keith Shortridge, and Fabio Pasian, eds.

Applicability of Agile Scrum to BepiColombo MPO Science Ground Segment Development

F. Perez-Lopez, S. Martinez, J. Gallegos, M. Casale, M. Cuevas, R. Gill, I. Ortiz de Landaluce, N. Fajersztejn, D. Galan, I. Serraller, and M. Freschi

European Space Astronomy Centre (ESAC), Villanueva de la Cañada, Madrid, Spain; `Fernando.Perez-Lopez@esa.int`

Abstract. BepiColombo is an interdisciplinary ESA-JAXA mission to explore the planet Mercury. The Science Ground Segment (SGS), located at the European Space Astronomy Centre (ESAC), will be in charge of the ESA's Mercury Planetary Orbiter (MPO) scientific operations including data processing, preliminary analysis, archiving and distribution to the instrument teams and the science community. This paper describes the SGS development methodology and how it has been progressively migrated into an Agile Scrum, but maintaining the information consistency and the adherence to ECSS standards.

1. Introduction

BepiColombo (see Benkhoff et al. 2010) is an interdisciplinary European Space Agency (ESA) mission to explore the planet Mercury, in cooperation with the Japan Aerospace Exploration Agency (JAXA), that will be launched from Kourou in 2018. The mission consists of two separate Mercury orbiters: ESA's Mercury Planetary Orbiter (MPO) and JAXA's Mercury Magnetospheric Orbiter (MMO), which are dedicated to the detailed study of the planet and its magnetosphere. The two orbiters are combined into a stack configuration (Mercury Composite Spacecraft, MCS) which will be used during approximately 6.5 years for the interplanetary cruise phase. The MPO science payload (11 instrument packages plus a radiation monitor) will investigate the interior, the surface composition and morphology, the intrinsic magnetic field, the composition of the exosphere and the coupling between all of these aspects of the innermost planet. During cruise, BepiColombo MCS will be operated and controlled from the MPO Operations Ground Segment (OGS) and after Mercury arrival, the two spacecraft will be independently operated from the OGS and JAXA. The MPO science operations will be controlled by the SGS which will be in charge of the scientific payload operations planning, the science operations, the science data processing and distribution to the instrument teams, the preliminary analysis of the scientific data and their central archiving accessible to the science community (Macfarlane et al. 2015; Barbarisi et al. 2015).

2. The development approach

Due to the different needs during the different phases, the SGS development was organized around the phased deployment of "Launch" and "Mercury" versions of the system, with the Launch System (LaS) to be operational by launch capable of supporting the SGS activities from launch (April 2018) through the first few years of the cruise phase, while the Mercury System (MeS), to be deployed incrementally after launch will have as main scope to support the nominal science operations of the mission (2025-2026) but as well some other later-on activities during the cruise phase that the LaS would not support. Initially the SGS adopted an incremental development approach, the modified waterfall, which means to deliver a working part of a total product or solution slicing the system functionality into increments. However, this could not be strictly applied, and, we believe, this was due to several reasons:

- **System under-specification and users' involvement**: The complexity of a system which provides the variety of functionalities as described above, combined with highly interactive components, is a big challenge which requires the involvement of the final users (11 instrument teams) during development and validation phases. The mission lifetime is quite long (almost 10 years from launch until post-operations) and the system must cope with the users' needs at Launch and, in addition, must accommodate new requirements that will cover the mission at Mercury science phase. So, the challenge is to anticipate by 10 years the system and users' needs, survive the technology changes, and translate all of these in concise and clear requirements with the right level of detail for implementation; jeopardizing the specification of good subsystem requirements which provide the appropriate context of system functionalities necessary for their implementation.

- **Lateness in Testing**: Following strictly a waterfall approach the whole system is only tested at the end of each cycle (six months between testing campaigns according to the SGS plans). This does not facilitate the continuous involvement of the final users, during development and validation phases, which is considered fundamental for the SGS success, and reduces the time to apply corrections, if bugs and requirements misunderstandings are found, making their fixing more complicated (or even late).

2.1. Transitioning to Agile

Based on the previous, the SGS decided, at beginning of 2016, to change progressively the development methodology to an Agile Scrum approach. By going in this *Agile direction*, the main inconveniences found are expected to be corrected, guaranteeing the involvement of the users during system development & verification & validation and ensuring the continuous adaptation of the system with respect to the real needs during all mission phases. The unique condition which was imposed for the transition was to reuse as much as possible all the existing SGS knowledge in terms of documentation and processes; and, in consequence, the adherence to ECSS. This transition has been easier than expected, taking advantage of several good practices already adopted by the SGS: the use of JIRA/Confluence[1] (Gill et al. 2015) as a configuration control system allows us to centralize the project information (actions, SPRs, user/system/subsystem requirements, risks, test cases/reports, ...) and facilitates the tracking of the work and the consistency of the information; the SW development environment was designed to be a continuous integration system and uses several tools (Git, Nexus, Maven, Jenkins,

[1] https://www.atlassian.com/software

Sonar) which fits with an Agile methodology; the SGS concept and architecture documentation is quite consolidated and provides a good starting point for the transition to Agile. In addition, some system changes were required:

- **Definition of roles**: The Scrum master role was taken over by the SGS Quality advisor, ensuring the correctness of good practices (testing, metrics, etc.) from the very beginning and the early detection of problems. Considering that the SGS involves several disciplines, it is impossible that only one person could have a deep knowledge on all matters; so several product owners were identified, depending on the expertise area, but only one acts as contact point to the SW team.

- **Definition of epics/stories/tasks**: As mentioned, the main pitfall found was the lack of user involvement. In order to interact with the final users, providing *context* to the requirements and also a better scheduling of the development work, *use cases* have been incorporated to the system. These *use cases*, assumed as *epics*, were split in *user stories* which are tangible and implementable according to the Agile methodology. Taking the advantage of using JIRA, we incorporated the user stories and the epics in JIRA maintaining the traceability with the requirements and the rest of the existing SGS issue types. For this purpose, we have made use of the Jira Agile plugin which has proven to be a very powerful tool. The following figure provides a summary of the different issue types and their traceability:

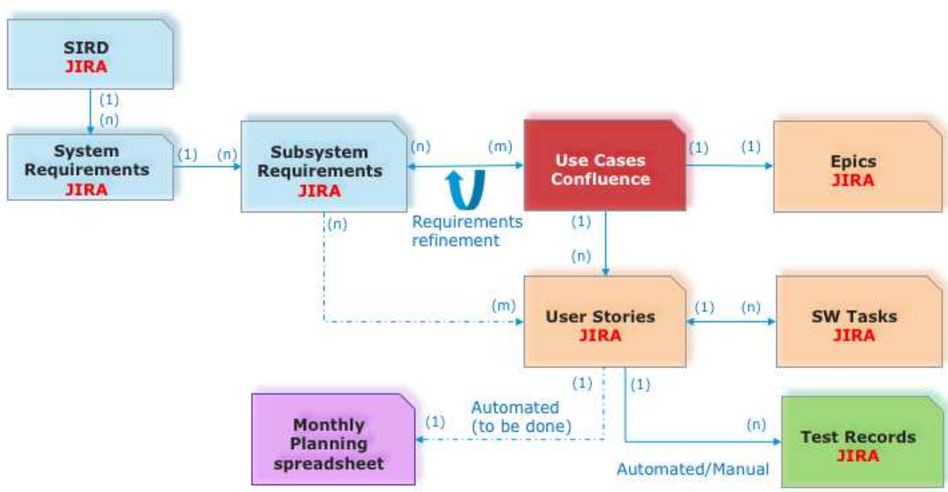

Figure 1. Description of the issue types and traceability details.

- **Scrum cycles**: We adopted 4-weeks as the nominal duration of each sprint with a release at the end. Every six sprints, a formal version of the system is delivered for acceptance testing according to the SGS general schedule. Apart from the Scrum sprint review, sprint planning and sprint retrospective (especially useful) meetings; at the beginning of each version cycle a ***version scope*** meeting with the whole team is organised to define the goals and general priorities of the six upcoming sprints. This provides visibility of the full scope of the planned features and allows us to anticipate the needs from the different SGS and final users' perspectives; responding to questions on the next system version like: what are the system version goals, the features for each sprint, the documents to be provided and how these will be validated. This defines what does *Done* (or expected results) mean for each user story, as a key aspect.

- **Testing**: Apart from the unit testing activities carried out as part of the development, automatic regression tests have been included in the system: on the daily deployment of the development branch and on each sprint release. This practice is considered extremely useful by the developers because it ensures continuous visibility of the system consistency. In addition, every six sprints, a formal system testing campaign is carried out. Note that the late sprint of each cycle is used to consolidate the tests to be automated as part of the regression testing. In this way, once the user stories have been declared as *Done* by the developers, they are automatically tested and test reports are ingested in JIRA closing the loop between definition, specification, implementation and testing and maintaining all the information centralized in JIRA/Confluence. A set of tools installed facilitates the automatic generation of the documentation required by the standards.

3. Conclusions

After some development sprint cycles applying this methodology we have noted a significant improvement of the system, in terms of definition and performances. Now the **system does what it is expected to do** and the SGS is able to react in time to possible system changes. The involvement of the final users (i.e. instrument teams) has been improved by having a product available from early stages and incorporating users' feedback into each monthly sprint. In parallel, the communication between product owners and developers has been largely improved and now the understanding of requirements has been improved.

References

Barbarisi, I., et al. 2015, AGU Fall Meeting Abstracts

Benkhoff, J., et al. 2010, Planetary and Space Science, 58, 2. Comprehensive Science Investigations of Mercury: The scientific goals of the joint ESA/JAXA mission BepiColombo, URL `http://www.sciencedirect.com/science/article/pii/S0032063309002840`

Gill, R., et al. 2015, in ADASS XXIV, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of ASP Conf. Ser., 371

Macfarlane, A. J., et al. 2015, in ADASS XXIV, edited by A. R. Taylor, & E. Rosolowsky, vol. 495 of ASP Conf. Ser., 547