openMDM® 5 development status with live demo

by

Angelika Wittek,openMDM® EWG toolkit manager

openMDM® Eclipse Working Group
General Members Meeting
July 25th, 2017

Who we are

The Team

Angelika Wittek
Toolkit Management
Independent, 50%
Munich

Ganesh Deshvini
Canoo, 50%
Mumbai

Florian Schmitt
Atos, 20%
Munich

Matthias Koller
Peak Solution, 100%
Nuremberg

Alexander Nehmer
Atos, 80%
Tuebingen

Basic conditions

- 4 developers from 3 companies (2,5 full-time resources)
- time & material based
- 3 sprints per 42 days (~6 weeks)
- 75 person days planned per sprint

Sprint Planning

- Sprint 1: 12.06.-21.07.2017 -> Release V0.7
- Sprint 2: 24.07.-01.09.2017 -> Release V0.8
- Sprint 3: 04.09.-13.10.2017 -> Release V0.9

Goals:

- Stabilize and improve codebase
- Stabilize deployment
- Refactor / Extend / Improve the API
- Improve the performance
- Write / improve the documentation

What are we doing?

past / present / future

Sprint 1

12.06.-21.07.2017 -> released V0.7 last Friday

- planned: 75 person days (less days consumed)

DONE:

- REQUs
- Building, Encoding, Quality issues
- Documentation (Setup Guide, Deployment Guide)
- Setup the environment (every developer)
- Analysis of the requirements (thanks for the support from Sebastian Diersch and Victor Stöhr, committers from Gigatronik)

Release V0.7

- Release V0.7 in JIRA: https://openmdm.atlassian.net/projects/REQU/versions/12907/tab/release-report-all-issues
- Codebase: Tag V0.7 in Git on master branch

DONE:

- REQU-48, REQU-92, REQU-67
- Building, Encoding (UTF-8), Quality issues

PENDING:

- publish deployable artefacts on the mdmbl project-page
- link the Setup Guide and the Installation Guide on the mdmbl project-page

Lessons learned from Sprint 1

- good collaboration in team
- communication is important -> 2 dev telcos a week
- setup per developer between 3-5 days
- having just 1 active committer slowed us down in sprint 1
- to understand certain REQUs, support is necessary
- development environment and build processes is not set up optimal, more investigation necessary
- availability was less than planned

And thanks to the support we got, especially from Audi, BMW, Gigatronik and Peak Solutions!

Sprint 2

24.07.-01.09.2017 -> started yesterday

planned: 75 person days (may be shifted due to German holiday season)

Planned:

- release V0.8
- next REQUs (order by priority and maturity)
- setup continuous build and static code analysis
- improve documentation (Setup Guide, Deployment Guide)
- update the mdmbl Eclipse Project Page
- technical issues: hot deployment, improved build scripts
- establish closer contact to AC, QC

Support needed from the EWG

- JIRA REQU issue type / workflow has to be extended:
  - status: “TO DO”, “IN PROGRESS”, “DONE” is not sufficient
- provisioning of test data (real world examples)
- replanning of the sprints in respect to availability of the developers
- creation of new requirements
  - user stories
  - acceptance criterias
  - stakeholder (responsible person)
- testing testing testing ...

How we work

We are developing OSS

- openness
- transparency
- Eclipse Process and Tools
  (Eclipse Bugzilla, Eclipse Repositories, Code Review Process)
- openMDM(R) processes
  (JIRA, Requirements, SC, AC, QC)

Communication

- Mailing List: mdmbl-dev
  - we post periodically: meeting minutes, decisions of the team
  - everybody can subscribe [1] and interact
  - emails are archived [2]
  - ask questions and remarks -> we will answer
- Eclipse Bugzilla Issues [3]: add a comment or create a new one
- JIRA REQU Issues [4]: add a comment

[1] https://dev.eclipse.org/mailman/listinfo/mdmbl-dev
[2] https://dev.eclipse.org/mhonarc/lists/mdmbl-dev/
[3] https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=mdmbl
[4] https://openmdm.atlassian.net/secure/RapidBoard.jspa?rapidView=57&projectKey=REQU&view=planning.nodetail

JIRA Requirements & Bugzilla Issues (1)

- Requirements are in the JIRA REQU project [1], prioritized by BMW
- work is done ordered by priority and maturity
- for every REQU issue one or more Eclipse Bugzilla issues [2] are created
  - a comment with the link to the Bugzilla issue is added the the REQU issue
  - the Bugzilla issue summaries follow the naming convention:
    [REQU-xx] <summary>
- dev team works on the Bugzilla issues

[1] https://openmdm.atlassian.net/secure/RapidBoard.jspa?projectKey=REQU&rapidView=57&view=planning

[2] https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=mdmbl

JIRA Requirements & Bugzilla Issues (2)

- Bugzilla issue is implemented -> set to resolved with an comment (summary)
- Release week / code freeze -> Tests
- Tests done -> Bugzilla issue set to done with comment (version)
- JIRA REQU is set to done with an comment and set the version attribute
- JIRA REQUs are automatically assigned to the release
- all done: set the release to “released” in JIRA

Eclipse Bugzilla

- created from REQU issues
- “real” bugs from internal or external
- technical requirements / enhancements from the team

Codebase - Git repositories

- structure, see [1]
- master: lastest stable release
- dev-branch: current development
- Tags in Git relate to the releases in JIRA

[1] http://git.eclipse.org/c/?q=mdm

Why are active committers important?

Committer Privileges

These tasks can only be done by a committer:

- Assign Eclipse Bugzilla Bugs
- Contribute code and documentation directly to the repositories (Git)
- Code reviews via Gerrit (Gateway to Git)
- Edit the Eclipse Project Pages
- Configure and maintain the systems for continuous build and static code analysis (Jenkins, Sonar)
- Start committer elections, elect the project lead
- and more….

Committer in the Eclipse ecosystem

From the Eclipse website:

*(...) Becoming a committer is a **privilege** that is **earned** by contributing and showing discipline and good judgment. It is a **responsibility** that should be neither given nor taken lightly, nor is it a right based on employment by an Eclipse member company or any company employing existing committers.*

There are only three requirements around nominating and electing:

- Define Trust
- Employment Neutral
- Public and Archival Election

URL: https://wiki.eclipse.org/Development_Resources/HOWTO/Nominating_and_Electing_a_New_Committer

Current committers

Project Lead and Committer: Stefan Beese (inactive)

Committers:

- Matthias Koller
- Andres Almiray
- Sebastian Dirsch
- Viktor Stoehr
- Christian Weyermann (inactive)
- Alexander Nehmer (new)
- Angelika Wittek (new)

URL: https://projects.eclipse.org/projects/technology.mdmbl/who

Committer work for the EWG

Decision from the SC Meeting, 29.7.2017:

As of now, work contributed to the openMDM related projects (e.g., Eclipse mdmbl) as committer can be considered as a service package as described in the openMDM charter. The decision if a service qualifies shall be on a case-by-case base at the discretion of the Toolkit Manager in close communication with the Steering Committee. Proposals from the members will be very appreciated.

see: JIRA Issue ORGA-256

Contribution is easy

Be a contributor

- everybody can do it
- sign the ECA [1]
- contribute code to the code review system (Gerrit)

Create a bug

- something is missing, does not work
- create a bug [2]

Communicate with us via the mdmbl-dev mailing list

[1] https://www.eclipse.org/legal/ECA.php

[2] https://bugs.eclipse.org/bugs/buglist.cgi?quicksearch=mdmbl

Questions?

Contact me directly:

angelika.wittek.openmdm@gmail.com

or via the mailing list

subscribe: https://dev.eclipse.org/mailman/listinfo/mdmbl-dev

Demo Time!