Model Driven Paediatric European Digital Repository

Call identifier: FP7-ICT-2011-9 - **Grant agreement no**: 600932

Thematic Priority: ICT - ICT-2011.5.2: Virtual Physiological Human

Deliverable 17.3

Test Report on MD-Paedigree Release

Due date of delivery: 31th May 2017

Actual submission date: 26th June 2017

Start of the project: 1st March 2013
Ending Date: 31th may 2017

Partner responsible for this deliverable: GNUBILA

Version: 0.3

Dissemination Level: Public

Document Classification

Title	Test Report on MD-Paedigree Release
Deliverable	17.3
Reporting Period	4
Authors	Sebastien Gaspard
Work Package	17
Security	PU
Nature	RE
Keyword(s)	Testing Infrastructure Beta Prototype

Document History

Name	**Version**	**Date**
MDP_D17.2_ Test Report on MD-Paedigree RElease _v0.1	0.1	24 May 2017
Reviewed version	0.2	31st May 2017
Final version	0.3	26th June 2017

List of Contributors

Name	**Affiliation**
Sebastien Gaspard	MAAT
David Manset	MAAT

List of reviewers

Name	**Affiliation**
Rod Hose	USFD
Harry Dimitropoulos	ATHENA
Bruno Dallapiccola	OPBG

Table of Contents

1 Introduction

The purpose of this deliverable is to report on the activities of task T17.1 “MD-Paedigree Infrastructure testing and validation” which started on M18 and ended on M51. Task T17.1 is focused on ensuring the timely and efficient completion of the activities necessary to test and validate the MD-Paedigree Alpha and Beta Prototypes and the Final Platform developed under WP14.

It is the last of three deliverables associated with this task (D17.1, D17.2 & D17.3), and in particular this deliverable reports on the testing of the MD-Peadigree Release on M51 (see also deliverable D14.4).

2 Testing and agile methodology

With a distributed team both in terms of location and business centres, the standard scrum methodology could not fully and naturally apply. In this situation, the methodology has consisted in pragmatically keeping “agile” concepts with less formalism, and picking up every month the emergencies and tasks with the most concerned users to then implement one by one the functionalities that were missing or bugging. A “quick in production” process has been used to make the functionalities usable (and so testable) as soon as possible.

3 Release test report

3.1 Repository Unit tests

The infrastructure repository that manages the data sharing now consists of 13381 lines of code of unit tests for a total of 1752 assertions. (in comparison with the 1584 lines of code in Alpha version)

The tests are distributed on different topics:

Lines			Elements of Tests	
Y3	**Y4**		**Y3**	**Y4**
720	466	**general**	24	66
3743	3825	**metadata**	505	558
550	528	**semantics**	161	73
5177	5324	**data**	837	825
423	3238	**rights**	57	230
10613	13381	**Total**	1584	1752

Basic tests about access have been moved to Rights specific tests, refactoring of existing code has proceeded and lots of tests have been added to check Access Rights.

Each compilation of the code runs the 118 procedures of testing and does not proceed in case of error.

```
Tests run: 118, Failures: 0, Errors: 0, Time elapsed: 53.646 sec
```

3.2 Functional tests and use

3.2.1 D17.1 postulate

The MD-Paedigree project is a federation of technologies. In this context, integration is a key feature and needs to be thoroughly tested. For this, depending on the integration need of each application, specific attention will be paid to the following tests:

- Application is able to connect to the infostructure repository through web services
- Application is accessible through the portal
- Application is using MD-Paedigree access control (login password)
- Application is using MD-Paedigree rights (groups)

3.2.2 Web service real condition tests

Web services are the main interfaces to the repository, to exchange data with other applications including importers. All data is sent to the repository via web services.

Different partners use the web services to retrieve/push data from/to their application. All kind of data can be exchanged by different organisations using this. The real use that has taken place over the two last years has demonstrated the usability and the stability of this interface.

3.2.3 Interfaces tests

Since the Alpha version, graphical user interfaces are available to enable the end users to upload data. Over the last two year these interfaces have been used by different people with different skills. As of today, all the participants have gone through the process of using the interfaces in their everyday work and no more malfunctions have been identified.

3.2.4 Insertion tests

All data has been produced. Data sources have been identified and more or less specific Importers have been developed. For all data inserted, different tests have been implemented and pass with success. All data formats are now managed and the database is filled.

3.2.4.1 Manual Development Tests

For each data imported, a manual verification of samples has been processed by developers. One by one, fields of a small set of data have been checked to ensure the availability and the completeness of each importation.

3.2.4.2 Systematic comparison tests

When possible on the importer machine inside the secured network of the data centre, in order to ensure the absence of unplanned alteration in data, some systematic comparison tests have taken place. For simplicity, these tests have been processed by the exportation of data of a kind into CSV format and compared to the same set of data converted also to CSV. The two sheets were compared using an Excel utility, and it has been verified that only anonymization rules had altered the data in the importation process.

3.2.5 Access Rights

This last year, the main element of testing the infrastructure was about the access rights. A huge set of tests have been developed to ensure the security of data and its concordance with the awaited behaviour. Access rights tests have been treated with a special attention involving some key users that have confirmed the availability of their data and verifying they are not seeing other groups data anymore.

3.3 Bug report and correction

Testing is not only happening when tests are done but also whilst the system is being operated in its functional mode. Users have different channels to provide feedback and to log problems: some have direct access to the reporting tool, whilst others use Skype or mail. All problems, evolutions or bug reports are logged into the development management system.

During the last year gnùbila has managed around 400 fixes and updates tasks in direct relation with the repository or its importers.

4 Conclusions

All applications are integrated together. The central repository is filled with all kinds of data that is used by modellers with different tools. Web applications are integrated into the portal with each at the chosen level of interactions. Several users are testing the use cases as part of their usual work and stabilisation is taking place. The combination of asset of unit tests and the real use of the system ensures a good level of functionality and reporting of the major part of the bugs.