Planning

Useful feedback in the Ampersand parser

Maarten Baertsoen and Daniel S. C. Schiavini

Open Universiteit Nederland, faculteit Informatica
T61327 - Afstudeerproject bachelor informatica

October 16, 2014

Version 0.1

Contents

1 Introduction (R-M) 4
1.1 Identification . . . 4
1.2 Goal of this document . . . 4
1.3 Document overview . . . 4

2 Project description (R-M) 5
2.1 The Ampersand project . . . 5
2.2 Current situation . . . 6
2.3 Goals of the project . . . 6
2.4 Project architecture and components . . . 6
2.5 Project environment . . . 7
2.6 Critical success factors . . . 8
2.7 Objectives and commitments . . . 8

3 Knowledge acquisition (-) 9
3.1 Research context . . . 9
3.2 Domain & technology . . . 9
3.2.1 Part Daniel . . . 9
3.2.2 Part Maarten . . . 9
3.3 Knowledge documentation . . . 9

4 Project approach (-) 10
4.1 Project methodology . . . 10
4.2 Project planning . . . 10
4.3 Project milestones and corresponding deliverables . . . 10

5 Project management (M) 10
5.1 Project governance / roles & responsibilities . . . 11
5.2 Communication . . . 11
5.2.1 Internal . . . 11
5.2.2 OU . . . 12
5.2.3 Customer . . . 12
5.3 Time keeping . . . 12
5.4 Project reporting . . . 13
5.5 Quality assurance . . . 13
5.5.1 Process quality & monitoring . . . 13
5.5.2 Deliverables quality & monitoring . . . 13

6 Assumptions and constraints (R-M) 14

7 Interferences with other projects (-) 14

8 Risk management (M) 14
8.1 Risk identification and qualification . . . 14
8.1.1 Functional risks . . . 14
8.1.2 Technical risks . . . 15
8.2 Risk mitigation plan . . . 15

9 Project realization (-) 15
9.1 Coding conventions . . . 15
9.2 Scrum approach . . . 15
9.3 Validation . . . 16

10 Testing & Validation (-) 16
10.1 Test approach . . . 16
10.2 Test methodology . . . 16
10.3 Test plan & Milestones . . . 16
10.4 Test documentation . . . 16

11 Issue management (-) 17

12 Integration & release (-) 17

13 Documentation (-) 17
13.1 Documentation plan . . . 17
13.2 Deliverables . . . 17

14 Tools, methodologies and accelerators (R-M) 18
14.1 Collaboration . . . 18
14.2 Documentation . . . 18
14.3 Design . . . 18
14.4 Development . . . 18
14.5 Testing . . . 18

Appendices 19

Glossary 19

List of Figures

1 Generated artifacts . . . 5
2 Architecture of the project . . . 7
3 Relevant data flow for the Ampersand parsing component . . . 8

1 Introduction (R-M)

1.1 Identification

This document contains the planning for the execution of the graduation project "Useful feedback in the Ampersand parser". This planning gives the high-level requirements, the risks and a timefor the project. As such, the planning provides the steps for reaching the project objectives, and provides criteria that are used to validate and accept the results of the graduation.

This document is part of the graduation project of the computer science bachelor at the Open Universiteit Nederland. The project "Useful feedback in the Ampersand parser" is assigned to the students Daniel Schiavini and Maarten Baertsoen, with support of the supervisor Dr. Bastiaan Heeren and examiner Marko van Eekelen. The assignment is given by professor Stef Joosten, who researches how to further automate the design of business processes and information systems by the development of the Ampersand project.

1.2 Goal of this document

The main goal of this document is to capture the taken decisions and agreements around the execution of the project. In order to make the targets clear, the project context is also depicted in the document.

The document describes the current situation and the issues it presents, making clear why the project has been started. The purpose is thus to describe the management approach and the describe the aimed solution in high-level, as well as chances, risks and problems that might occur.

1.3 Document overview

An introduction is given is this chapter. Afterwards, a general description of the project is given in section 2. Then, in section 3, strategies are proposed for the acquisition of knowledge. In section 4 the project approach is explained and the management strategy is given in section 5.

Assumptions and limitations are given in **??**. Afterwards, possible interferences with other projects are given in section 7. Strategies for risk management are then proposed in section 8.

Details of the development techniques are given in the project realization strategies of section 9. Testing and validation plans are in section 10, while issue management is explained in section 11. The strategy for integration of the released software is given in section 12, while the documentation strategy is given section 13 and the used tools and methodologies are given in section 14.

Finally, a glossary of terms, definitions and abbreviations is given in subsection 14.5.

2 Project description (R-M)

2.1 The Ampersand project

In November 2003, the Business Rules Manifesto[1] was written, with the main purpose of declaring independence for business rules in the world of requirements. The manifesto supports the vision of business rules as equivalent to requirements. This is considered a radical change on how people see the world of business architecture.

In December 2010, Stef Joosten, Lex Wedemeijer and Gerard Michels published the paper "Rule Based Design", presenting the Ampersand approach. The approach puts the rules in the center, using them to define the business processes. Ampersand is named after the & symbol with the desire of realizing results for both business and IT, in an efficient and effective way.

In 2011, the Ampersand compiler was created as an open source project. Since then, the compiler has been improved and applied in both business and academic contexts. The Ampersand end-users write business rules in a specific language (ADL), and compile that specification into functional specification, documentation and working software prototypes. These rules are based on agreements between the different stakeholders.

The theory behind Ampersand has been throughly studied, and is based on mathematical concepts, e.g. Relational algebra and Tarski's axioms. Using this compiler, users write the requirements in ADL and generate all the system specification independent on the platform. The main advantage is that the requirements consistency and traceability are always correct (and even provable), from the lowest level up to the front-end. The requirements are presented to stakeholders in natural language, guaranteeing that any business expert who knows the context can validate the requirements. Figure 1 depicts the artifacts generated by the Ampersand compiler.

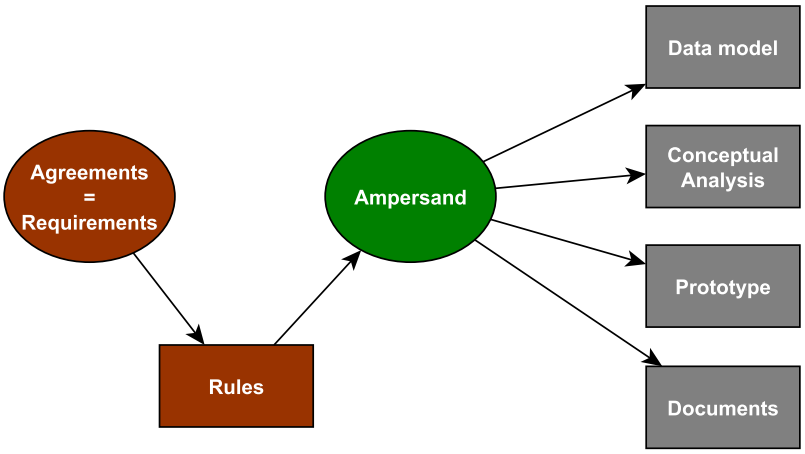

Figure 1: The Ampersand approach generates different artifacts based on the business rules

[1] http://www.businessrulesgroup.org/brmanifesto.htm

2.2 Current situation

The compiler developed for the Ampersand research project runs in several steps (see subsection 2.4 for the architecture). The first step is the parsing of the input scripts. One of the main complaints from users is the quality of the errors generated by the Ampersand parser. Since the beginning of the project, special attention has never been given to the parser subcomponent, and it has not been analyzed for improvements.

In order to generate better error messages, it is assumed that a complete refactoring of the parser will be necessary. The main challenge is to choose the correct kind of architecture and libraries in order to generate the most user-friendly messages possible.

Besides the main project task of improving the parser's feedback, a list of user wishes has accumulated over the years. The main customer and other users would appreciate if as many wishes as possible could be fulfilled.

2.3 Goals of the project

The main objective for the graduation project is to implement useful feedback in the Ampersand parser. In order to achieve this goal, some of the following activities can take place:

- Analysis of user-friendly messages in compilers;
- Comparison of different Haskell parsing libraries (also for pretty-printing);
- Researching tools and techniques in Haskell for improving the software quality (e.g. testing and error messages);
- Analysis of the current development environment in relation with software engineering principles such as continuous delivery/integration;
- Recommending improvements for the overall software quality;

In case the new parser is successfully implemented and accepted, while the project members still have time budget available, the list of open user wishes issues can be addressed. Some of these wishes are substantial, so that most of them cannot be fulfilled during the graduation project. The current list of open issues has been provided[2], although it must be clear that the issues are strictly seen as lower priority.

2.4 Project architecture and components

The Ampersand compiler is divided in several subcomponents:

Parser: This component receives the ADL code as input, and parses that code into a parse-tree (also known as P-structure).

Type checker: The Ampersand type checker receives the P-structure as input and converts it into a relational algebra format, suitable for manipulation (also known as A-structure or ADL-structure). The semantics of ampersand are expressed in terms of the A-structure.

[2] `https://github.com/AmpersandTarski/ampersand/tree/ABI_Parser/ArchitectureAndDesign`

Calc: The Calc component receives the A-structure as input, and manipulates it according to the research rules, generating the functional structure (also known as F-structure). The F-structure contains all design artifacts needed to write a specification and generate the output.

Output components: All design artifacts present in the F-structure are ready to be rendered. Several components use this data structure to generate the wished output. Outputs currently implemented (and their output formats) are the following: Atlas (HTML interface), revert (Haskell source), Query (prototype generation) and Documentation Generator (Pandoc structure).

The complete architecture is depicted in Figure 2. The part of this architecture relevant for this project is depicted in Figure 3.

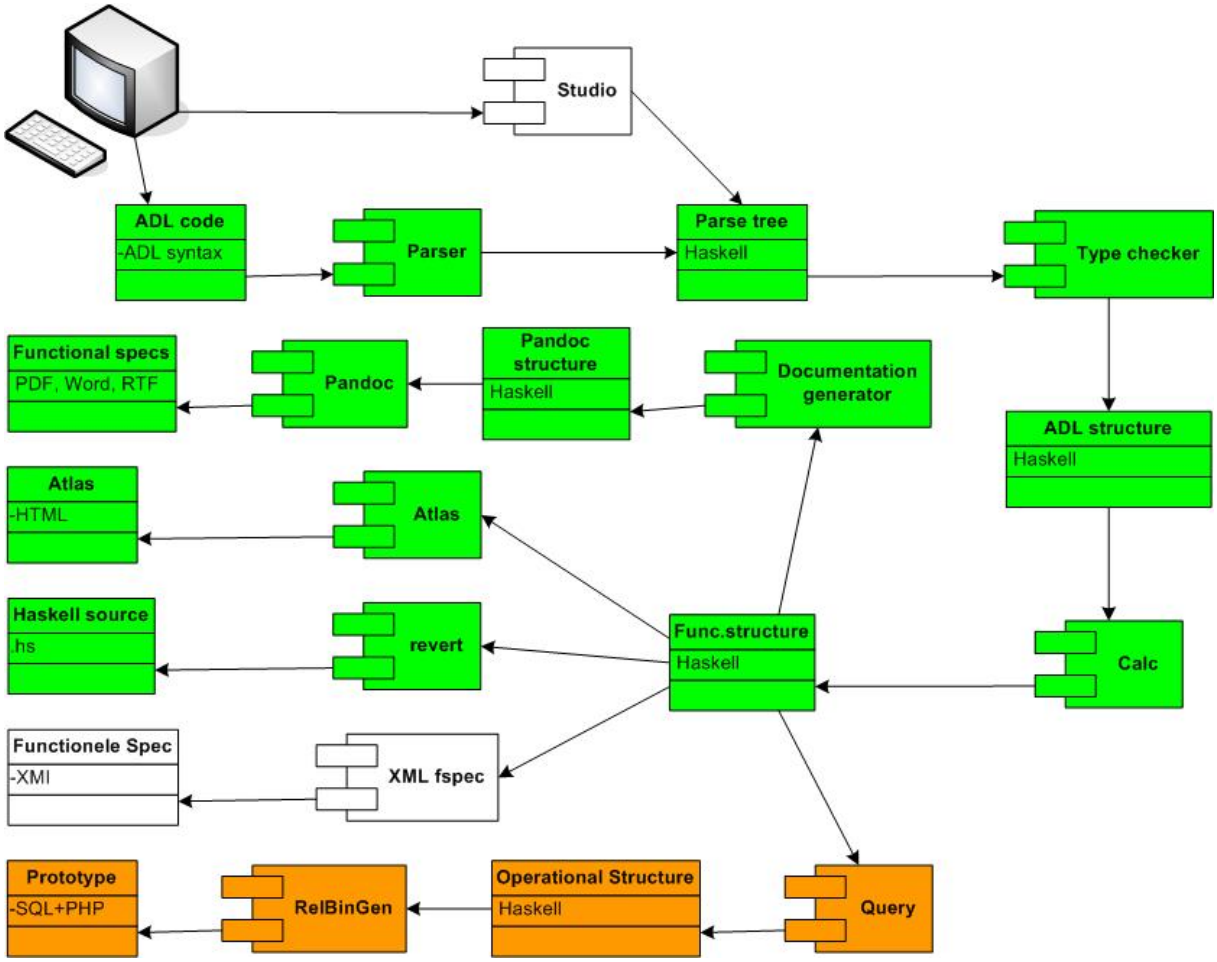

Figure 2: Architecture of the project, showing where the parser fits in the Ampersand system

The components in green background are part of the Ampersand compiler. Components in orange are part of the Ampersand Prototype compiler. Finally, components in white background are future components, not yet implemented.

2.5 Project environment

The Ampersand project is used in the following environments, for different users:

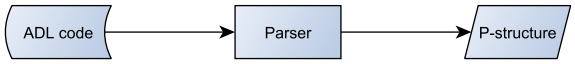

Figure 3: Relevant data flow for the Ampersand parsing component

Research: The Ampersand project is part of a research effort on the use of business rules for software design;

Academic: Ampersand is used as main tool in the course "Ontwerpen met bedrijfsregels" (code T18321) from the Open Universiteit Nederland;

Business: The compiler is used in business environments to design and develop real world business software.

2.6 Critical success factors

The following factors are critical for the project's successful completion:

Maintainability: the code shall at least be as maintainable as the current Ampersand code. It is known however that maintainability is hard to measure (see also section 8).

Production code: the implemented functionalities shall be integrated into the master branch for production use;

Users context: in order to provide useful feedback, the user context wherein the Ampersand compiler is used needs to be well understood, and the improvements must have extra user value;

2.7 Objectives and commitments

Besides the project goals described in subsection 2.3 and the customer goals described in subsection 2.6, the project members declare herewith to have the following objectives and commitments fulfilled by the end of this graduation project:

Customer: The primary objective is to deliver a well-working and maintainable piece software that will help the customer.

Users: Another important goal is to make the software more user friendly, and so improve the usability.

Knowledge: Building up knowledge is the main reason why one starts a bachelor study. As such it is important do learn more about functional programming, Haskell, compilers, business rules and research in general.

University: Hopefully the final thesis will be of use for the university and other students.

Graduation: As this is a graduation project, it is natural to have the graduation as an important objective.

3 Knowledge acquisition (-)

3.1 Research context

Some ideas by Bastiaan:

- more info on ampersand users
- user-friendly error messages (specific to compilers or not)
- parser-libraries
- Stef's research (3a)
- Haskell parsers (e.g. Helium), monadit or combinators

3.2 Domain & technology

3.2.1 Part Daniel

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.2.2 Part Maarten

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.3 Knowledge documentation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna

fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4 Project approach (-)

4.1 Project methodology

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.2 Project planning

Milestones & phases

4.3 Project milestones and corresponding deliverables

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5 Project management (M)

This sections descirbes the project management approach and used techniques to guarantee project succes. As the project team consists only out of 2 team members, special attention is given to make the project management approach holistic but light. Holistic to assure that all important aspects with regards to the management of this project are properly covered and lightweight to avoid that the project team loses time in handling unnecessary tasks which only have added value in larger project teams.

5.1 Project governance / roles & responsibilities

We distinguish 5 different project parties in the project environment:

1. The project team The project team consists out of Daniel Schiavini and Maarten Baertsoen. This team takes the full responsibility for the actual delivery for all project work products consisting out of
   a) The actual Haskell program code in full conformity with the coding conventions
   b) Exhaustive code documentation
   c) All necessary final and intermediate deliverables to ensure project success such as: project planning and, project methodology as described within this document.

   The full list of all work products with their corresponding description is included in this document.

2. The project supervisor The project supervisor is the main contact person for the project team. It is the role of the project supervisor to monitor the project team on a regular basis as well as to provide support and guidance to the project team members both on content and approach. It is the responsibility of the project team to send project status updates towards the project supervisor; these status updates will form the basis for the regular alignment meetings. During project execution, it might occur that the effort needed to meet the customer expectations is way over the foreseen budget of the project, measured in hours. Should this situation occur, the supervisor will facilitate customer alignments to adjust the customer's requirements. All efforts spent are therefore reported on a weekly basis to the supervisor who will track the spent effort to make sure that every aspect of the project receives the needed attention.

3. The customer The project will deliver an actual product with all corresponding documentations in a controlled way. The customer's satisfaction is one of the key success factors of this project.

   Although this project is conducted in a educatinal environment, the customer will be treated just like he would be served in a professional environment.

4. End-Users The Ampersand Parser is currently used by several users groups

5. The examiner

5.2 Communication

Regular alignment sessions and communications s are foreseen withtinthe project. The communication plan described below will identify the way the project team will align and communicte with the dfferent project parties

5.2.1 Internal

The project team memebers, Dniel and Maarten will align on a weekly basis. During this meeting, following topics will be discussed:

1. Tasks done

2. Tasks due
3. Assignment of next project tasks wihtin the projetc team
4. Actions
5. Issues & Risks

Given the project size of the team, Daniel and Maarten will communicate on a regular basis between each other on a informal basis besides this recurrent weekly meeting.

5.2.2 OU

A recurrent meeting bewteen the project supervisor and the project team is planned each 3 weeks. This meeting is an offical feedback meeting regarding the full status of the projects encompassing all important project aspects such as progress, issues and risks.

During these meetings the supervisor will monitor and steer the project progress and will check the proper project management by consulting the project's statistics. All formal project delivery products which needs to be delivered towards the examinator.

All documents will be delivered to the project supervisor at least 24hrs before the actual meeting to allow the project supervisor to perform an in depth review . It is the goal of the projec team members to deliver the to be reviewed documents a couple of days earlier to respect the supervisor's agenda.

5.2.3 Customer

The meetings with he customer will be planned carefully both on timing and content to make sure that the customer is not disturbed too much with project questions not relevant for the customer.

The main principle to distinguish between the questions that can be posed to the customer or not, is by reflecting our project to a real life project. Every question or issue that should be handled by a professional project team, based on knowledge and experience, will be handled internally within this project as well. All questions, issues and status updates towards the customer will be listed by the project team and will be revised with the project supervisor to assure to customer meetings are efficient, clear and most importantly, that these meetings provide the customer a trustful feeling regarding the project.

5.3 Time keeping

The project team members will record every effort they have spent on the project an overview over:

1. Date
2. Project phase
3. Project domein
4. Executor

This overview will be shared with the project supervisor on a regular basis.

5.4 Project reporting

The project team will share periodic status updates during the project, this reporting will cover following project aspects:

1. General project status
2. Timing
3. Results achieved
4. Budget (spent hours)
5. Risks and issues
6. Quality
7. Focus for the upcoming weeks

5.5 Quality assurance

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.5.1 Process quality & monitoring

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.5.2 Deliverables quality & monitoring

Each project deliverable will be reviewed by the project team member's counterpart within the project team. Where needed, several revision iterations will be foreseen until the project team comes to an agreement regarding the quality level of the deliverable. Internally approved deliverables will be shared with the project supervisor who can perform an additional quality check before the deliverables are shared with the customer and the

project examiner. Each delivery towards the customer can be followed by a subsequent revision phase to perform a final fine-tuning to assure customer satisfaction. The necessary information to assure that the initial deliverable towards the customer is as spot on as possible will be gathered during the customer alignment meetings.

6 Assumptions and constraints (R-M)

The following assumptions and constraints are considered in this project plan:

- For the main objectives, only the parser needs to be changed much. It is assumed that the changes in other parts of the software will be very limited.
- The project members will need to spend time building up knowledge on various involved parts: business rules, Ampersand, Haskell, LaTeX, etc.
- The architecture as described by the customer and documented on the Ampersand Wiki is assumed to be correct and up-to-date.
- The delivered software will be reviewed and tested by the relevant parties within a week.
- The customer and supervisor will be able to answer questions, give support, feedback and advice without much delay.
- The assumption is made that two project members will be available for an average of 12 hours a week. This can be influenced by the members' jobs or private lives.

7 Interferences with other projects (-)

(code conflicts, etc...)

8 Risk management (M)

8.1 Risk identification and qualification

8.1.1 Functional risks

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

8.1.2 Technical risks

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

8.2 Risk mitigation plan

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

9 Project realization (-)

9.1 Coding conventions

De coding conventions are described on the Ampersand wiki [3].

9.2 Scrum approach

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[3] http://wiki.tarski.nl/index.php/Coding_conventions

9.3 Validation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

10 Testing & Validation (-)

10.1 Test approach

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

10.2 Test methodology

Will we use student's data? That can be hard to do.

10.3 Test plan & Milestones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

10.4 Test documentation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero,

nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

11 Issue management (-)

12 Integration & release (-)

(acceptation, communication, etc...)

13 Documentation (-)

13.1 Documentation plan

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

13.2 Deliverables

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

14 Tools, methodologies and accelerators (R-M)

14.1 Collaboration

For the collaboration between the project members, the following tools will be used:

Dropbox For sharing time tracking and other reference documents;

GitHub For sharing git repositories of code, besides managing issues and documentation;

SourceForge For managing existing (old) issues;

Microsoft Office For writing internal documents, e.g. time tracking;

14.2 Documentation

For writing the documentation, the following tools will be used:

Haddock For annotating documentation on the code;

TeXworks For writing and compiling LaTeX documents;

Ampersand Wiki For keeping notes and information useful for users and other developers;

14.3 Design

For designing the software and its architecture, the following tools will be used:

yEd For creating diagrams and graphs;

14.4 Development

For software development, the following tools will be used:

IDE No standard integrated development environment will be chosen: the project members are free to use any IDE, e.g. Eclipse, Leksah, Notepad++;

GHC The compiler GHC (Glasgow Haskell Compilation System), version 7.8.3, will be used;

Cabal For managing Haskell packages and compilation, cabal-install version 1.18.0.5 and Cabal library version 1.18.1.3.

14.5 Testing

Finally, the following tools will be used for testing the software:

Hpc might be used for checking, recording and displaying the code coverage of tests;

Sentinel server can be used for the integration tests;

QuickCheck for automating tests on the code and property based testing (e.g. pretty-print and reparsing, random code generation);

Glossary

A-structure

The ADL code generated by the Ampersand type checker, used as input for the calculator component.. 6

ADL

Ampersand Design Language. 5

ADL-structure

See A-structure.. 6

Cabal

Library for managing Haskell builds and packages. 18

Dropbox

File hosting service that offers cloud storage, file synchronization, personal cloud, and client software. 17

F-structure

The functional structure generated by the Ampersand calculator, used as input for the different output modules.. 6

GHC

Glasgow Haskell Compilation system. 18

Git

A distributed revision control and source code management. 17

GitHub

Git repository web-based hosting service which offers all of the distributed revision control and source code management (SCM) functionality of Git. 17

Haddock

A software documentation generator for the Haskell programming language. 18

Hpc

Library for checking, recording and displaying code coverage. 18

IDE

Integrated Development Environment. 18

LaTeX

Document preparation system and document markup language for the TeX typeset. 18

Microsoft Office

Office suite of desktop applications devleoped by Microsoft. 17

P-structure

The parse-tree generated by the Ampersand parser, used as input for the type checker.. 6

QuickCheck

Library for testing Haskell code. 18

Sentinel

Test server for the Ampersand project. 18

TeXworks

Graphical user interface for editing and compiling LaTeX documents. 18

yEd

Software for editing graphs. 18