Requirements for rmtoo

Free and Open Source Requirements Management Tool

written by Andreas Florath

17. May 2011

Requirements Version: Commit [None] VCS Id [doc/topics]

Contents

Chapter 1

rmtoo

1.1 Introduction

The *rmtoo* requirements management tool is a tool to help, support and fasten up the development of software.
Therefore *rmtoo* comes with the whole set of requirements for *rmtoo* itself. This is (as far as we know) unique in the area of requirements management systems. Ask other vendors vendors for the requirements of their requirements management system. Be surprised about their answers.
This requirement description can be seen on two levels. First these are the requirements for *rmtoo*. So after reading them, you should be aware what *rmtoo* was designed for.
Second this can be seen as an example of the output of *rmtoo* itself: *rmtoo* is self-containing in the way, that all the requirements for *rmtoo* are written for *rmtoo*.

1.2 GUI

One and a half year after the first idea to create a free and open source requirements management tool, it was decided (as of 2011-05) to add a GUI to the tool.
The file format for the requirements will be kept - maybe there is the need to redefine the configuration. The creation of the GUI will be done incrementally - as the whole development of *rmtoo*. It will take at least some 10th of releases to get a good and usable GUI for the tool - which supports all the features and is well implemented and tested.
The GUI will be implemented in two parts: the major work and business logic will be implemented in a *rmtoo*-server. There will be some clients which are very thin and only render the data in different technologies (e.g. GTK, browser, Java, ...) This is the *master* requirment where all other (especially the initial requirements) depend on. There can only be one master requirment.

1.3 rmtoo

Description: *rmtoo* **must** exists.
Rationale: The world needs a good, usable and free Requirements Management Tool.
It looks that there are no such programs out.
But: it's complex!

Solved by: 1.10.3 Documentation, 1.4.19 Open Source rmtoo, 1.4.20 Processing, 1.4.24 Simplicity, 1.12.5 rmtoo Automated Testing
Test Cases: 2.1 Example Test Case, 2.2 Example Test Case 2

Id:	rmtoo	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-02-06	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4 Initial Requirements

This is the section containing all the initial requirements.

1.4.1 Analytics

Description: The requirements **must** be analyzed.
Rationale: It is hard to write good requirements - as far as possible rmtoo should support writing good requirements.
Note: Analytics are implemented using modules with a defined interface.
Depends on: 1.4.20 Processing
Solved by: 1.4.36 Analytics: Description Words, 1.4.36 Analytics: HotSpots, 1.4.36 Analytics: Requirement Topic Coherence, 1.4.36 Analytics: Topic Coherence

Id:	Analytics	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-05	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.2 Automatic Generation of Results

Description: *rmtoo* **must** support the automatic genration of outputs.
Rationale: Because rmtoo is aimed to be used in productive development environments, there is the need that all the different outputs (e.g. PDFs, graphs, ...) must be generated automatically (without user interaction).
Depends on: 1.4.20 Processing
Solved by: 1.4.3 Completed Requirement, 1.11.1 Makefile

Id:	AutomaticGeneration	**Priority:**	3.00	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.3 Completed Requirement

Description: It **must** be possible to check if a requirement is completed.
Rationale: Completed means that i.e. it and all the children are finished.
This can be used for a 'not yet finished' list as an output artifact.
Depends on: 1.4.2 Automatic Generation of Results

Id:	Completed	**Priority:**	0.30	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.4 Configuration Check

Description: The configuration **must** be checked.
Rationale: There might be typos in the configuration. Everything what is possible to check should be checked. Especially types and ranges.
Note: There are some major reasons to do this at one point and not to spread it over the different classes - especially those things which are used in multiple classes.
Depends on: 1.4.9 Configuration

Id:	ConfigCheck	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-23	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.5 Default Configuration

Description: At all places where it makes sense, a default configuration **must** be given.
Note: This simplifies the normal use of rmtoo: not all aspects must be defined always (but can).
Depends on: 1.4.9 Configuration
Solved by: 1.4.7 Configuration for maximum line length using in Topics, 1.4.8 Default Configuration for maximum line length using in Topics

Id:	ConfigDefault	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-23	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.6 Configuration of Developers

Description: The list of persons which can realize / implement requirement **must** be configurable.
Depends on: 1.4.9 Configuration, 1.5.17 Status: Assigned, 1.5.17 Status: Finished

Id:	ConfigDevelopers	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.7 Configuration for maximum line length using in Topics

Description: The maximum line length used in topic input file **must** be configurable.
Rationale: Some customers like strict length limitations where other do not like any limitation at all.
Depends on: 1.4.5 Default Configuration
Solved by: 1.4.8 Default Configuration for maximum line length using in Topics

Id:	ConfigTopicMaxLineLength	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-23	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.8 Default Configuration for maximum line length using in Topics

Description: The default maximum line length used in topics **must** be 80.
Rationale: This is the old and ever used value for this.
Depends on: 1.4.5 Default Configuration, 1.4.7 Configuration for maximum line length using in Topics

Id:	ConfigTopicMaxLineLengthDefault	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-23	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.9 Configuration

Description: Many aspects of the *rmtoo* behavior **must** be configured.
Note: The sub-requirements define which aspects must be defined.
Depends on: 1.4.17 Different Inputs
Solved by: 1.4.4 Configuration Check, 1.4.5 Default Configuration, 1.4.6 Configuration of Developers

Id:	Configuration	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-23	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.10 Requirement Constraints

Description: It **must** be possible to add constraints to a requirement.
Rationale: Typically a solution has some boundary conditions or constrains which limit the solution space.
In other requirements management resources this is often named *non functional requirements* - nevertheless mostly the definition of the difference between a *functional requirement* and a *non functional requirement* is vague.
In **rmtoo** the definition of a constraint is clear and well defined.
Depends on: 1.4.23 rmtoo must work on Requirments
Solved by: 1.4.11 Constraints Inheritance

Id:	Constraints	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-02-17	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.11 Constraints Inheritance

Description: Constrains **must** be inherited from the dependent requirements.
Rationale: A constraint for a whole solution must be valid for each part (if not: this is not a constraint!).
Depends on: 1.4.10 Requirement Constraints

Id:	ConstraintsInheritance	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-02-17	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.12 Ease of Use

Description: *rmtoo* **must** be easy to use.
Rationale: Only then it will be used.
Depends on: 1.4.24 Simplicity
Solved by: 1.13.2 Packaging

Id:	EaseOfUse	**Priority:**	9.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.13 Easy Editable

Description: The requirements used by *rmtoo* **must** be easy editable.
Rationale: It must not be a pain to change or add something.
Depends on: 1.4.24 Simplicity
Solved by: 1.9.12 Emacs Mode, 1.4.25 Traceability

Id:	EasyEditable	**Priority:**	2.70	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.14 Easy Extensible

Description: **rmtoo must** be easy extensible.
Rationale: There must no need to redisign the whole program when e.g. adding a requirements tag.
Depends on: 1.4.24 Simplicity

Id:	EasyExtensible	**Priority:**	5.40	**Owner:**	development
Invented on:	2010-02-10	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.15 Files in File System

Description: *rmtoo* **must** use files located in the file system as input.
Rationale: This can e.g. the checked out working copy of the requirements from the Version Control System. But the files can also be completely independent from any Version Control System.
Depends on: 1.4.23 rmtoo must work on Requirments
Solved by: 1.4.29 Use Filename as Requirement ID

Id:	FilesInFS	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.16 Glossary must be available

Description: *rmtoo* **must** support the possibility to define terms in a glossary.
Rationale: Typically a requirements management document used specific terms over and over. The glossary gives a possibility to define those terms once.
Different output modules can use them for e.g. automatically linking the terms to their explanation.
Depends on: 1.4.17 Different Inputs
Solved by: 1.9.3 Emacs Mode for Glossary

Id:	Glossary	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.17 Different Inputs

Description: *rmtoo* **must** support different types of input files - one type for each usage.
Note: A usage is e.g. documenting requirements or handling topics.
Depends on: 1.4.20 Processing
Solved by: 1.4.9 Configuration, 1.4.16 Glossary must be available, 1.4.18 Graphical User Interface, 1.4.23 rmtoo must work on Requirments, 1.6.6 Topics must be supported, 1.4.31 Use Txt

Id:	Input	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-16	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.18 Graphical User Interface

Description: It **must** be able to enter all data needed by a project with the help of a GUI.
Rationale: Not everybody is willing to enter the data inside a text editor.
Note: Think about the possibility of also implementing this by means of a browser interface.
Depends on: 1.4.17 Different Inputs

Id:	Input/GUI	**Priority:**	10.00	**Owner:**	development
Invented on:	2012-03-13	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.19 Open Source rmtoo

Description: *rmtoo* **must** be Open Source.
Rationale: There is the hope, that some people might use this. There is also the hope that there are some people which add the one or other feature.
Depends on: 1.3 rmtoo

Id:	OpenSource	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.20 Processing

Description: *rmtoo* **must** process the requirements by means of a defined order.
Rationale: First the requirements must be read in (and checked for syntactic - and if possible for semantic) problems.
Then the requirements must be analyzed.
As a last step the output artifacts must be generated.
Note: Please see the depended requirements for more details.
Depends on: 1.3 rmtoo
Solved by: 1.4.1 Analytics, 1.4.2 Automatic Generation of Results, 1.4.17 Different Inputs, 1.8.20 Output of Different Artifacts

Id:	Processing	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-05	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.21 Requirement generics: Order of Tags

Description: The order of tags in a requirements description **must** be independent of the semantic.
Rationale: Order doesn't matter.
Depends on: 1.4.23 rmtoo must work on Requirments

Id:	ReqGenTagOrder	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-16	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.22 Requirement ID

Description: Each requirement **must** have an unique id .
Note: This means that there is a 1:1 correlation between an id and a requirement.
Depends on: 1.4.23 rmtoo must work on Requirments
Solved by: 1.4.29 Use Filename as Requirement ID

Id:	RequirementId	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-09-27	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.23 rmtoo must work on Requirments

Description: *rmtoo* **must** work on requirements.
Rationale: That's is what's about.
Depends on: 1.4.17 Different Inputs
Solved by: 1.4.10 Requirement Constraints, 1.9.7 Emacs Mode for Requirements, 1.4.15 Files in File System, 1.4.21 Requirement generics: Order of Tags, 1.5.13 Requirement Tags, 1.4.22 Requirement ID, 1.4.35 Version Control System

Id:	Requirements	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.24 Simplicity

Description: *rmtoo* **must** be simple.
Rationale: To get started, concentrate on the major things, which are really needed.
Use techniques which are available.
Depends on: 1.3 rmtoo
Solved by: 1.4.12 Ease of Use, 1.4.13 Easy Editable, 1.4.14 Easy Extensible, 1.4.30 Use Python

Id:	Simplicity	**Priority:**	9.00	**Owner:**	development
Invented on:	2010-02-08	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.25 Traceability

Description: When a requirement changes, all dependency requirements **must** be marked.
Rationale: Maked means e.g. colorized in the graph. There is the need for a tag 'last changed' and 'last checked'.
Depends on: 1.4.13 Easy Editable
Solved by: 1.9.10 Emace Mode to Support Traceablility, 1.7.9 No Directed Circles Allowed

Id:	Traceability	**Priority:**	1.35	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.26 Txt Comment Semantics

Description: Comments (resp. empty lines) **must** be treated as comments which describe the next tag.
Rationale: This definition is needed to have the possibility to re-arrange or extend existing requirements.
Depends on: 1.4.31 Use Txt
Solved by: 1.4.27 Txt Comments, 1.4.28 Txt Empty Lines

Id:	TxtCommentSemantics	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-27	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.27 Txt Comments

Description: *rmtoo* **must** support comments in the text input files.
Rationale: Sometimes there is the need to comment tags.
Depends on: 1.4.26 Txt Comment Semantics

Id:	TxtComments	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-27	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.28 Txt Empty Lines

Description: *rmtoo* **must** support empty lines in the text input files.
Rationale: This makes it easier to read requirements.
Depends on: 1.4.26 Txt Comment Semantics

Id:	TxtEmptyLines	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-27	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.29 Use Filename as Requirement ID

Description: The file name of the requirement **must** be used as the unique id.
Rationale: It's there and it's unique.
Note: Possible slashes ('/') which are used as path separator are part of the unique requirement id.
Depends on: 1.4.15 Files in File System, 1.4.22 Requirement ID

Id:	UseFilenameAsId	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-09-27	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.4.30 Use Python

Description: *rmtoo* **must** use python as language for scripts.
Rationale: python applcations are fast to develop.
Depends on: 1.4.24 Simplicity
Solved by: 1.12.3 Test Tool: python-nose

Id:	UsePython	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-10	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.31 Use Txt

Description: *rmtoo* **must** use txt-files for documenting requirments.
Rationale: There is no need for a special editor - normal editors (such as emacs and vi) can be used.
Depends on: 1.4.17 Different Inputs
Solved by: 1.7.7 Checks, 1.4.26 Txt Comment Semantics

Id:	UseTxt	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-02-08	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.32 Version Control System: History Interval

Description: It **must** be possible to define a version interval for *rmtoo* operations.
Rationale: This gives the user the possibility to exactly reference one specific version and re-create the documents for a historic version.
Note: The specification of the version identification is Version Control System dependent. It should be possible to use symbolic constants like tag names or constants for specifying the latest version (e.g. 'HEAD').
Depends on: 1.4.35 Version Control System
Solved by: 1.4.33 Version Control System: History Interval Usage, 1.4.34 Version Control System: Latest Version

Id:	VCSInterval	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.33 Version Control System: History Interval Usage

Description: The specified history interval **must** be used for statistics.
Depends on: 1.4.32 Version Control System: History Interval

Id:	VCSIntervalUsage	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.34 Version Control System: Latest Version

Description: The latest version specified is the version *rmtoo* **must** work on.
Rationale: This means that all documents and other artifacts which only allow one version as base must use the latest given.
Depends on: 1.4.32 Version Control System: History Interval

Id:	VCSLast	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.35 Version Control System

Description: *rmtoo* **must** use a Version Control System for history handling like baselining.
Rationale: A major use case is working with different sets of requirements: e.g. develop version 2 while adding features to version 1.
Depends on: 1.4.23 rmtoo must work on Requirments
Solved by: 1.4.32 Version Control System: History Interval

Id:	VersionControlSystem	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.4.36 Analytics

Analytics: Description Words

Description: The description **must** be analyzed by means of a heuristic which evaluates regular expressions.
Rationale: This is a easy and fast way to get some hint of the quality of the description itself.
Depends on: 1.4.1 Analytics

Id:	AtcsDescWords	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-05	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

Analytics: HotSpots

Description: There **must** be an analytics module which checks if there are requirements where the sum of the incoming plus outgoing links is too high.
Rationale: If there are requirements with too many links, typically there is the need to add one level of indirection: Divide et Impera.
Depends on: 1.4.1 Analytics

Id:	AtcsHotSpot	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-05	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

Analytics: Requirement Topic Coherence

Description: There **must** be an analytics module which compares the coherence of one requirement inside one topic to the outside of a topic.
Rationale: If there are too many links to the outside, the requirement might be in the wrong topic.
Depends on: 1.4.1 Analytics

Id:	AtcsReqTopicCohe	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-05	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

Analytics: Topic Coherence

Description: There **must** be an analytics module which compares the coherence inside one topic to the outside of a topic.
Rationale: If there are too many links to the outside, a topic is mostly not well chosen.
Note: This means that all the links inside a topic are counted and compared to the ones which point to the outside.
Depends on: 1.4.1 Analytics

Id:	AtcsTopicCohe	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-05	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.5 Requirement Tags

Each requirment has a couple of tags and values. This section describes all the tags.

1.5.1 Class

Description: The class tag **must** be one of: 'implementable' or 'detailable'.
Rationale: This is a short string (headline) for the requirment. 'implementable' means that the requirment is that detailed, that is can be implemented. 'detailable' means that the requirment is not directly implementable but must be broken down and elaborated more to be implementable.
Depends on: 1.5.2 Requirments Class
Solved by: 1.8.62 Priority Output Order By Class

Id:	Class	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.2 Requirments Class

Description: Each requirement **must** have a class tag.
Rationale: The class decides which class type the requirment is.
Depends on: 1.5.14 Requirement Tags: Basics
Solved by: 1.5.1 Class

Id:	ReqTagClass	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.3 Requirements Description

Description: Each requirement **must** have a description tag.
Rationale: This is the description what the requirment must fulfill.
Depends on: 1.5.13 Requirement Tags

Id:	ReqTagDescription	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.4 Requirement Tag Effort Estimation

Description: Each requirement **may** have a tag 'Effort estimation'.
Rationale: This gives a hint how much effort is pending.
Depends on: 1.5.14 Requirement Tags: Basics
Solved by: 1.5.16 Effort Estimation Measure, 1.8.59 Priority output Include Effort Estimation

Id:	ReqTagEffortEst	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.5.5 Requirements Invented By

Description: Each requirement **must** have a 'invented by' tag.
Rationale: This names the original (initial) author of the requirment.
Depends on: 1.5.13 Requirement Tags

Id:	ReqTagInventedBy	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.6 Requirements Invented On

Description: Each requirement **must** have a 'invented on' tag.
Rationale: This is the date when the requirement was written.
Depends on: 1.5.13 Requirement Tags

Id:	ReqTagInventedOn	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.7 Requirements Name

Description: Each requirement **must** have a name tag.
Rationale: This is a short string (headline) for the requirment.
Depends on: 1.5.15 Requirement Tags: Basics

Id:	ReqTagName	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.8 Requirements Owner

Description: Each requirement **must** have a owner tag.
Rationale: The ower is the stakeholder who defines the requirment (sometimes different from the one, who writes it down — which is the 'Invented by' person).
Depends on: 1.5.15 Requirement Tags: Basics

Id:	ReqTagOwner	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.9 Requirement Priority

Description: Each requirement **must** have a priority.
Rationale: A priority is a number. The higher the number is, the higher prioritized is the requirment.
Depends on: 1.5.14 Requirement Tags: Basics
Solved by: 1.8.37 Output of Priority List, 1.5.17 Priority Computation, 1.7.8 Requirement Priority Check

Id:	ReqTagPriority	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-02-13	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.10 Requirements Status

Description: Each requirement **must** have a status tag.
Depends on: 1.5.13 Requirement Tags
Solved by: 1.5.17 Status

Id:	ReqTagStatus	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.11 Topic

Description: Each requirement **should** have a topic tag.
Rationale: The topic tag defines the relation of requirement to a special topic.
Depends on: 1.5.13 Requirement Tags

Id:	ReqTagTopic	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.12 Requirements Type

Description: Each requirement **must** have a type tag which must be one of "master requirement", "initial requirement", "requirement" or "design decision".
Depends on: 1.5.15 Requirement Tags: Basics

Id:	ReqTagType	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.13 Requirement Tags

Description: Each requirement description **must** include a set of tag-value pairs.
Rationale: This can be easily implemented.
Depends on: 1.4.23 rmtoo must work on Requirments
Solved by: 1.5.3 Requirements Description, 1.5.5 Requirements Invented By, 1.5.6 Requirements Invented On, 1.5.10 Requirements Status, 1.5.11 Topic, 1.5.14 Requirement Tags: Basics, 1.5.15 Requirement Tags: Basics

Id:	ReqTags	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-16	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.14 Requirement Tags: Basics

Description: Requirement properties which are needed for agile development **must** be supported.
Note: The requirement properties are the one which depend on this requirement.
Depends on: 1.5.13 Requirement Tags
Solved by: 1.5.2 Requirments Class, 1.5.4 Requirement Tag Effort Estimation, 1.5.9 Requirement Priority

Id:	ReqTagsAgile	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-04	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.15 Requirement Tags: Basics

Description: The basic requirements properties **must** be supported.
Note: The basic requirements properties are the one which depend on this requirement.
Depends on: 1.5.13 Requirement Tags
Solved by: 1.5.7 Requirements Name, 1.5.8 Requirements Owner, 1.5.12 Requirements Type

Id:	ReqTagsBasic	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-04	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.5.16 Effort Description

This section descibes the format of the effort estimation input fields and their semantic.

Effort Estimation Allowed Measure Unit

Description: The effort estimation **must** be one of 0, 1, 2, 3, 5, 8, 13, 21.
Rationale: This is the typically used start of the fibonacci numbers for this. 0 is no effort, 1 is tiny effort and so on until 21 is huge effort.
Depends on: 1.5.16 Effort Estimation Measure

Id:	EftAllowedMeasure	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

Effort Estimation Measure

Description: The effort estimation **must** be given as (abstract) effort points.
Rationale: This given an independence between 'real' effort (e.g. days) and estimated which can be easy re-based to a 'real' effort after some (SCRUM) Sprints.
Depends on: 1.5.4 Requirement Tag Effort Estimation
Solved by: 1.5.16 Effort Estimation Allowed Measure Unit

Id:	EftMeasure	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.5.17 Format Desctiption

This sections describes the special formats of the different tags.

Priority Computation

Description: The priority **must** be computed as the normalized average of all stakeholder priorities multiplied by the maximum dependent weight.
Rationale: This makes sure that it is possible to zero out things that are not needed and that important things have a higher priority than not so important things.
Depends on: 1.5.9 Requirement Priority
Solved by: 1.5.17 Priority Format

Id:	PrioComputation	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-13	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

Priority Format

Description: Priority **must** be specified in a format, that each stakeholder can give a priority for each requirment.
Depends on: 1.5.17 Priority Computation

Id:	PrioFormat	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-13	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

Status

Description: Status **must** have one of the following states: 'not done', 'assigned', 'finished'.
Depends on: 1.5.10 Requirements Status
Solved by: 1.5.17 Status: Assigned, 1.5.17 Status: Finished, 1.5.17 Status: Not done

Id:	Status	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	assigned (Florath, 2011-04-27)
Class:	implementable				

Status: Assigned

Description: The status **must** be set to 'assigned' when a person works on this requirement.
Rationale: Work is typically not atomic. The assigned state means, that one persons works on this. This could be seen as a mutex: only the one which successfully checks in the assigned with his name is allowed to work on this requirement.
Note: There are extensions available - please consult the dependent requirements.
Depends on: 1.5.17 Status
Solved by: 1.4.6 Configuration of Developers, 1.8.22 Output of Assigned List, 1.5.17 Status: Assigned Value

Id:	StatusAssigned	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

Status: Assigned Value

Description: The status 'assigned' **must** be followed by the name of a person (separated with a colon) which works on the requirement and the date when the work was assigned to the person.
Note: The name of the person should be checked against a configured list.
Example: Status: assigned:Mustermann:2011-04-27
Depends on: 1.5.17 Status: Assigned

Id:	StatusAssignedValue	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 2 h)
Class:	implementable				

Status: Finished

Description: The status **must** be set to 'finished' when a person finished work on this requirement.
Note: There are extensions available - please consult the dependent requirements.
Depends on: 1.5.17 Status
Solved by: 1.4.6 Configuration of Developers, 1.8.28 Output of Finished List, 1.5.17 Status: Finished Value

Id:	StatusFinished	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 2 h)
Class:	implementable				

Status: Finished Value

Description: The status 'finished' **may** be followed by the name of a person (separated with a colon) which worked on the requirement, the date when the work was finished and the time in hours how long it took to finish the requirement.
Rationale: These information enable rmtoo to create a couple of different statistics, e.g. how many hours were spend in the whole project and an estimation how much is open and how long this will take.
The information is optional because of compatibility reasons.
Note: The name of the person should be checked against a configured list.

Example: Status: finished:Mustermann:2011-04-29:5h
Depends on: 1.5.17 Status: Finished

Id:	StatusFinishedValue	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

Status: Not done

Description: The status **must** be set to 'not done' when the requirement should be implemented but the work (design / implementation) has not yet started.
Note: There are no extensions available for the 'not done' value.
Depends on: 1.5.17 Status

Id:	StatusNotDone	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.6 Topic

1.6.1 Topic generics: Format independent

Description: The format of the topic **must** be output independent.
Rationale: It must be possible to create a wide range of possible formats, e.g. least LaTeX and HTML.
Depends on: 1.6.6 Topics must be supported
Solved by: 1.8.53 Output of Text Document Use Same Base

Id:	TopicGenFormat	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.6.2 Topic generics: Order of Tags

Description: The order of tags in a topic description **must not** be independent of the semantic.
Rationale: Order does matter.
Depends on: 1.6.6 Topics must be supported

Id:	TopicGenTagOrder	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.6.3 Topic Name

Description: Each topic **must** have a name tag.
Rationale: This is a short string (headline) for the topic.
Depends on: 1.6.5 Topic Tags

Id:	TopicTagName	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.6.4 Topic SubTopics

Description: Each topic **might** have one ore more sub-topics.
Note: The subtopic must point to another topic. The subtopic is included in the next hierarchy level with the subtopic's name as the headline - and possible sub-sub-topics as sub-sections.
Depends on: 1.6.5 Topic Tags

Id:	TopicTagSubTopic	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.6.5 Topic Tags

Description: Each topic **must** include a set of tag-value pairs.
Rationale: This can be easily implemented.
Depends on: 1.6.6 Topics must be supported
Solved by: 1.6.3 Topic Name, 1.6.4 Topic SubTopics

Id:	TopicTags	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.6.6 Topics must be supported

Description: *rmtoo* **must** support the possibility to define topics which can be referenced to create different output documents.
Rationale: This gives the possibility to link a requirement itself to a topic / theme / chapter.
Depends on: 1.4.17 Different Inputs
Solved by: 1.9.9 Emacs Mode for Topics, 1.6.1 Topic generics: Format independent , 1.6.2 Topic generics: Order of Tags, 1.6.5 Topic Tags

Id:	Topics	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-05-16	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7 Input

1.7.1 Graph Checks

Description: *rmtoo* **must** check the requirements for graph errors.
Rationale: Graph errors are errors which result in a broken requirement dependency graph, e.g. a circular dependency.
Depends on: 1.7.7 Checks
Solved by: 1.7.2 Graph Check for Connected Component, 1.7.3 Graph Check for Exact One Master Requirement, 1.7.4 Graph Check for Strongly Connected Component

Id:	CheckGraph	**Priority:**	8.10	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.2 Graph Check for Connected Component

Description: *rmtoo* **must** check that the requirements dependency graph consists of only one component.
Rationale: There must maximal one component - each node must be reachable by the master requirement.
Depends on: 1.7.1 Graph Checks

Id:	CheckGraphOneComponent	**Priority:**	8.10	**Owner:**	development
Invented on:	2010-06-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.3 Graph Check for Exact One Master Requirement

Description: *rmtoo* **must** check that there exactly one master requirement.
Rationale: The master requirement does not have any successors. It is the only requirement without successors. There must be only one master requirement — which is the base or root of everything.
Depends on: 1.7.1 Graph Checks

Id:	CheckGraphOneMaster	**Priority:**	8.10	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.4 Graph Check for Strongly Connected Component

Description: *rmtoo* **must** check that there is no strongly connected component in the requirement dependency with a size of two or larger.
Rationale: Directed circles are strongly connected components. A lot of other algorithms and assumptions do not work any more, if there are circles in the requirement graph.
Depends on: 1.7.1 Graph Checks

Id:	CheckGraphSCC	**Priority:**	8.10	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.5 Semantic Checks

Description: *rmtoo* **must** check the requirements for semantic errors.
Rationale: Semantic errors are errors which affect the meaning of a set of requirements.
Depends on: 1.7.7 Checks

Id:	CheckSemantic	**Priority:**	8.10	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.6 Syntax Checks

Description: *rmtoo* **must** check the requirements for syntax errors.
Rationale: Syntax errors are parse errors or other 'low level' errors.
Depends on: 1.7.7 Checks
Solved by: 1.7.8 Requirement Priority Check

Id:	CheckSyntax	**Priority:**	8.10	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.7 Checks

Description: *rmtoo* **must** check requirements.
Rationale: Because there is no database or other implicit check, *rmtoo* must do consistency checks every time it reads in the requirements.
Depends on: 1.4.31 Use Txt
Solved by: 1.7.1 Graph Checks, 1.7.5 Semantic Checks, 1.7.6 Syntax Checks

Id:	Checks	**Priority:**	9.00	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.8 Requirement Priority Check

Description: The priority **must** be a number between (and including) 0 and 10.
Rationale: Note that for internal computations the number is normalized.
Depends on: 1.7.6 Syntax Checks, 1.5.9 Requirement Priority

Id:	ReqTagPriorityCheck	**Priority:**	6.48	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.7.9 Checks

No Directed Circles Allowed

Description: Directed circles in requirements dependencies **must** be rejected.
Rationale: Directed circles make no sense in requirements management and make traceability impossible.
Depends on: 1.4.25 Traceability

Id:	NoDirectedCircles	**Priority:**	1.22	**Owner:**	development
Invented on:	2010-02-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8 Output

1.8.1 Evaluate Pricing Information: Checks

Description: Pricing information **must** be checked.
Depends on: 1.8.7 Evaluate Pricing Information
Solved by: 1.8.2 Evaluate Pricing Information: Completeness Checks, 1.8.3 Evaluate Pricing Information: Compliance Checks

Id:	EvalPriceChecks	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-11	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.2 Evaluate Pricing Information: Completeness Checks

Description: Pricing information **must** be checked if they are complete.
Rationale: For each requirement of the used requirement set there must be exactly one set of parameters. Therefore there is the need to check if all requirements are available and if all parameters are complete and correct.
Depends on: 1.8.1 Evaluate Pricing Information: Checks

Id:	EvalPriceChecksCompleteness	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-11	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.3 Evaluate Pricing Information: Compliance Checks

Description: Compliance information **must** be checked.
Rationale: There are some cases to check if the given compliance statements make sense. It is e.g. not possible to have a 'fully compliant' statement if one of the dependent requirements is not or partially compliant.
Note: It might be possible to put these information into the graph.
Depends on: 1.8.1 Evaluate Pricing Information: Checks

Id:	EvalPriceChecksComplicance	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-11	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.4 Evaluate Pricing Information: Graph

Description: There **must** be the possibility to evaluate the pricing information by using the pricing sheet to generate a graph.
Depends on: 1.8.7 Evaluate Pricing Information
Solved by: 1.8.5 Evaluate Pricing Information: Graph Color, 1.8.6 Evaluate Pricing Information: Graph Costs

Id:	EvalPriceGraph	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-09	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.5 Evaluate Pricing Information: Graph Color

Description: The pricing evaluation graph **must** be dye by using the compliance statement.
Rationale: This helps when evaluating incoming filled in pricing sheets.
Depends on: 1.8.4 Evaluate Pricing Information: Graph

Id:	EvalPriceGraphColor	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-09	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.6 Evaluate Pricing Information: Graph Costs

Description: The pricing evaluation graph **must** include the dependent pricing as well as the local requirement pricing.
Depends on: 1.8.4 Evaluate Pricing Information: Graph

Id:	EvalPriceGraphCosts	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-09	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.7 Evaluate Pricing Information

Description: There **must** be the possibility to evaluate the pricing information.
Depends on: 1.8.36 Output of Pricing Table
Solved by: 1.8.1 Evaluate Pricing Information: Checks, 1.8.4 Evaluate Pricing Information: Graph

Id:	EvaluatePricing	**Priority:**	5.60	**Owner:**	development
Invented on:	2010-09-09	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.8 Modular Output

Description: The output generation **must** be modular on a requirement tag level.
Rationale: This means, that the class for handling (i.e. parsing) is also responsible for the output.
Depends on: 1.8.20 Output of Different Artifacts

Id:	ModularOutput	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-03-10	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.9 Output of Pricing Table: Computation of Dependent Costs

Description: The document **must** compute the costs of all the dependent requirements.
Rationale: This must be based on the customer's input of dependent requirements.
Depends on: 1.8.11 Output of Pricing Table: Computation

Id:	OPCDependent	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.10 Output of Pricing Table: Computation of Sum

Description: The document **must** compute the sum of costs of all the dependent requirements plus local requirement costs.
Rationale: This must be based on the customer's input.
Depends on: 1.8.11 Output of Pricing Table: Computation

Id:	OPCSum	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.11 Output of Pricing Table: Computation

Description: The generated document **must** support automatic computation of the results needed to compare different biddings.
Rationale: These are the fields which the document must automatically compute.
Depends on: 1.8.36 Output of Pricing Table
Solved by: 1.8.9 Output of Pricing Table: Computation of Dependent Costs, 1.8.10 Output of Pricing Table: Computation of Sum

Id:	OPComputation	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.12 Output of Pricing Table: Input Cell Comment

Description: It **must** be possible that the vendor can comment each requirement.
Depends on: 1.8.18 Output of Pricing Table: Input Cells

Id:	OPICommment	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.13 Output of Pricing Table: Input Cell Compliant

Description: It **must** be possible that the vendor can define the compliant of each requirement.
Note: This is typically done with the help of *full*, *partially* or *none*.
Depends on: 1.8.18 Output of Pricing Table: Input Cells

Id:	OPICompliant	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.14 Output of Pricing Table: Input Cell Day Count

Description: It **must** be possible to define the number of days for each requirement.
Rationale: Different requirements might need different amount of effort.
Depends on: 1.8.18 Output of Pricing Table: Input Cells

Id:	OPIDayCnt	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.15 Output of Pricing Table: Input Cell Dayrate

Description: It **must** be possible to define the dayrate for each requirement.
Rationale: Different requirements might need different people which might be payed different.
Depends on: 1.8.18 Output of Pricing Table: Input Cells

Id:	OPIDayrate	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.16 Output of Pricing Table: Input Cell Dependent On

Description: It **must** be possible to define the dependent on relation for each requirement.
Rationale: Different vendors might want to split up the costs to different departments.
Depends on: 1.8.18 Output of Pricing Table: Input Cells

Id:	OPIDependentOn	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.17 Output of Pricing Table: Input Cell Material

Description: It **must** be possible to define the costs of the material for each requirement.
Rationale: Different requirements might need different amount of money.
Note: Material is everything which is not day or day rate, e.g. hardware costs, license costs, ...
Depends on: 1.8.18 Output of Pricing Table: Input Cells

Id:	OPIMaterial	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.18 Output of Pricing Table: Input Cells

Description: The generated document **must** support input fields for pricing information.
Rationale: These are the fields which the vendors must fill in.
Depends on: 1.8.36 Output of Pricing Table
Solved by: 1.8.12 Output of Pricing Table: Input Cell Comment, 1.8.13 Output of Pricing Table: Input Cell Compliant, 1.8.14 Output of Pricing Table: Input Cell Day Count, 1.8.15 Output of Pricing Table: Input Cell Dayrate, 1.8.16 Output of Pricing Table: Input Cell Dependent On, 1.8.17 Output of Pricing Table: Input Cell Material

Id:	OPInput	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.19 Output: Include Version Identifier of Requirements

Description: All *rmtoo* output modules **must** include the Version Control System identifier of the used requirements.
Rationale: This can be one identifier for e.g. documents or an interval for statistic output.
Depends on: 1.8.29 Generic Output Requirements

Id:	OutGenVersionId	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.20 Output of Different Artifacts

Description: *rmtoo* **must** support generation of different outputs.
Rationale: It's not very easy to e.g. visualize the dependency graph. Also typically for the testing department a document is needed that decribes the requirements (features) of the product.
Depends on: 1.4.20 Processing
Solved by: 1.8.8 Modular Output, 1.8.26 Document Output, 1.8.27 Output of Elements, 1.8.29 Generic Output Requirements, 1.8.39 Output Requirements, 1.8.40 Output Statistics about Persons, 1.8.44 Output of Number of Requirements, 1.8.45 Output Statistics about Work, 1.8.55 Output XML File

Id:	Output	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.21 Output Diff of Two Versions

Description: *rmtoo* **must** support document output generation where the differences of two versions are marked.
Rationale: This feature is often needed: one document is read and will change over time. To be up to date, you need only read the changes - which should me market / highlighed somehow.
Depends on: 1.8.52 Output of Text Document

Id:	Output/DiffOfTwoVersions	**Priority:**	6.40	**Owner:**	development
Invented on:	2012-03-08	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.22 Output of Assigned List

Description: It **must** be possible, to output a list of all *assigned* requirements sorted by date.
Rationale: This gives information about the currently work in progress.
Depends on: 1.8.27 Output of Elements, 1.5.17 Status: Assigned

Id:	OutputAssigned	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

1.8.23 Text Base Description Requirement References

Description: The text base description for the different output formats **must** be organized in a way that requirements can reference to one specific part.
Rationale: During writing a requirement it is (mostly) clear to which topic the requirement belongs to. So it makes sense to define it there.
Note: The 'specific part' will by typically a chapter or section in an document.
This is implemented using 'topics'.
Depends on: 1.8.52 Output of Text Document

Id:	OutputBaseReqRefs	**Priority:**	5.12	**Owner:**	development
Invented on:	2010-03-12	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.24 Text Base Description Choose Base Tags

Description: The text base description for the different output formats **must** be organized in a way that it is possible to specify for each output document the tags which will be outputted.
Depends on: 1.8.52 Output of Text Document

Id:	OutputBaseTags	**Priority:**	5.12	**Owner:**	development
Invented on:	2010-03-12	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.25 Output of Text Document must be configurable

Description: The output of the created document **must** be configurable.
Note: This can be done e.g. by supporting LaTeX macros or HTML CSS tags.
Depends on: 1.8.52 Output of Text Document

Id:	OutputConfgbl	**Priority:**	5.12	**Owner:**	development
Invented on:	2010-05-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.26 Document Output

Description: *rmtoo* **must** support genration of different types output documents.
Rationale: Different output documents might display different aspects of the requirements, Some might e.g. include the text, others the dependency graph.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.30 Output of Dependency Graph, 1.8.31 Output of Dependency Graph: Topics based, 1.8.36 Output of Pricing Table, 1.8.52 Output of Text Document

Id:	OutputDocument	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-09-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.27 Output of Elements

Description: *rmtoo* **must** support genration of different elements.
Rationale: Elements are only one small aspect of the whole set of information which are typically used in other output documents because only these elements make no sense at all. Example: file with version number of used requirements set.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.22 Output of Assigned List, 1.8.28 Output of Finished List, 1.8.37 Output of Priority List, 1.8.54 Output of Version Number

Id:	OutputElements	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-09-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.28 Output of Finished List

Description: It **must** be possible, to output a list of all *finished* requirements sorted by date.
Rationale: This gives information about the finished work.
Depends on: 1.8.27 Output of Elements, 1.5.17 Status: Finished

Id:	OutputFinished	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

1.8.29 Generic Output Requirements

Description: All *rmtoo* output modules **must** support a common set of functionality.
Rationale: There are some functionalities which must be implemented for all requirements.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.19 Output: Include Version Identifier of Requirements

Id:	OutputGeneric	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.30 Output of Dependency Graph

Description: *rmtoo* **must** support genration of a requirements dependency graph.
Rationale: A graph says more than thousand words.
Depends on: 1.8.26 Document Output
Solved by: 1.8.32 Output of Dependency Graph Configuration

Id:	OutputGraph	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.31 Output of Dependency Graph: Topics based

Description: *rmtoo* **must** support genration of a requirements dependency graph which is topic based.
Rationale: A graph says more than thousand words.
Depends on: 1.8.26 Document Output

Id:	OutputGraph2	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.32 Output of Dependency Graph Configuration

Description: The graph output module **must** be configurable in the way that different node attributes can separately switched on.
Rationale: For internal use the more information the better. For external use only some information is typically needed.
Depends on: 1.8.30 Output of Dependency Graph

Id:	OutputGraphConfig	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.33 Output of HTML

Description: *rmtoo* **must** support genration of HTML output.
Rationale: This gives a fast possibility to check requirements.
Depends on: 1.8.52 Output of Text Document, 1.8.53 Output of Text Document Use Same Base

Id:	OutputHTML	**Priority:**	6.40	**Owner:**	development
Invented on:	2010-03-10	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.34 Output of PDF

Description: *rmtoo* **must** support genration of PDF output.
Rationale: Typically the PDF output holds all the requirements and additional explanations.
Note: This might be implemented in the way of an additional layout layer, e.g. LaTeX.
Depends on: 1.8.52 Output of Text Document, 1.8.53 Output of Text Document Use Same Base
Solved by: 1.8.35 Output of PDF Configuration

Id:	OutputPDF	**Priority:**	6.40	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.35 Output of PDF Configuration

Description: The output PDF module **must** support configuration options in the way that different attributes can separately switched on.
Rationale: Often there is the need that different information should be written in different documents,
Depends on: 1.8.34 Output of PDF

Id:	OutputPDFConfig	**Priority:**	5.12	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.36 Output of Pricing Table

Description: *rmtoo* **must** support generation of a pricing table which can be easily filled in by vendors.
Rationale: This pricing table can be used to get the commercial bidding of different vendors.
Depends on: 1.8.26 Document Output
Solved by: 1.8.7 Evaluate Pricing Information, 1.8.11 Output of Pricing Table: Computation, 1.8.18 Output of Pricing Table: Input Cells

Id:	OutputPricing	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-08-15	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.37 Output of Priority List

Description: It **must** be possible, to output a list of all *open* requirements sorted by priority.
Rationale: This are typically those requirments which must be worked on first in an agile development process.
Depends on: 1.8.27 Output of Elements, 1.5.9 Requirement Priority
Solved by: 1.8.59 Priority output Include Effort Estimation, 1.8.60 Priority Output in Graph, 1.8.61 Priority Output LaTeX, 1.8.62 Priority Output Order By Class

Id:	OutputPrio	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-02-13	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.38 Output Requirements: Preserve Everything which is Possible

Description: When changing an already existing requirement, everything which is possible **must** be preserved.
Rationale: This is especially true for comments and empty lines.
Depends on: 1.8.39 Output Requirements

Id:	OutputReqsPreserve	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-27	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.39 Output Requirements

Description: If **must** be possible to output requirements in the text file format.
Rationale: This is needed by different other (automatic) tools which work on the rmtoo input files. For example a (possible and not yet planned) GUI can write new requirements or changes to existing requirements with the help of this functionality.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.38 Output Requirements: Preserve Everything which is Possible

Id:	OutputRequirements	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-12-27	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.40 Output Statistics about Persons

Description: *rmtoo* **must** support generation of statistics about each person which works on the project.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.41 Output Statistics about Persons: Relation, 1.8.42 Output Statistics about Persons: Time, 1.8.43 Output Statistics about Persons: Units

Id:	OutputStatsPersons	**Priority:**	8.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.41 Output Statistics about Persons: Relation

Description: The relation between the abstract EfE units and the real spent working hours **must** be included in the person amount statistics.
Depends on: 1.8.40 Output Statistics about Persons

Id:	OutputStatsPersonsRelation	**Priority:**	8.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.42 Output Statistics about Persons: Time

Description: The amount of time which were finished by a person **must** be included into the person statistics.
Depends on: 1.8.40 Output Statistics about Persons

Id:	OutputStatsPersonsTime	**Priority:**	8.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.43 Output Statistics about Persons: Units

Description: The number of abstract EfE units which were finished by a person **must** be included into the person statistics.
Depends on: 1.8.40 Output Statistics about Persons

Id:	OutputStatsPersonsUnits	**Priority:**	8.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.44 Output of Number of Requirements

Description: *rmtoo* **must** support genration of requirements statistics where the number of requirements related to the point of time.
Depends on: 1.8.20 Output of Different Artifacts

Id:	OutputStatsReqCnt	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.45 Output Statistics about Work

Description: *rmtoo* **must** support generation of statistics about the amount of work.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.46 Output Statistics about Work: Assigned, 1.8.47 Output Statistics about Work: Estimated End Date, 1.8.48 Output Statistics about Work: Finished, 1.8.49 Output Statistics about Work: Not Done, 1.8.50 Output Statistics about Work: Relation, 1.8.51 Output Statistics about Work: Start Date

Id:	OutputStatsWork	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.46 Output Statistics about Work: Assigned

Description: The work amount statistics **must** include the number of assigned work units.
Depends on: 1.8.45 Output Statistics about Work

Id:	OutputStatsWorkAssigned	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-28, 1 h)
Class:	implementable				

1.8.47 Output Statistics about Work: Estimated End Date

Description: The work amount statistics **must** include an estimated end date which is computed based on the finished work and the relation of the EfE units and real spent hours.
Depends on: 1.8.45 Output Statistics about Work

Id:	OutputStatsWorkEstEndDate	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.8.48 Output Statistics about Work: Finished

Description: The work amount statistics **must** include the number of finished work units.
Depends on: 1.8.45 Output Statistics about Work

Id:	OutputStatsWorkFinished	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

1.8.49 Output Statistics about Work: Not Done

Description: The work amount statistics **must** include the number of not done work units.
Depends on: 1.8.45 Output Statistics about Work

Id:	OutputStatsWorkNotDone	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

1.8.50 Output Statistics about Work: Relation

Description: The relation between the abstract EfE units and the real spent working hours **must** be included in the work amount statistics.
Depends on: 1.8.45 Output Statistics about Work

Id:	OutputStatsWorkRelation	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-27, 1 h)
Class:	implementable				

1.8.51 Output Statistics about Work: Start Date

Description: The start date of the work amount statistics **must** be configurable.
Depends on: 1.8.45 Output Statistics about Work

Id:	OutputStatsWorkStartDate	**Priority:**	10.00	**Owner:**	development
Invented on:	2011-04-27	**Invented by:**	flonatel	**Status:**	finished (Florath, 2011-04-28, 1 h)
Class:	implementable				

1.8.52 Output of Text Document

Description: *rmtoo* **must** support genration of different types text documents.
Rationale: Different types mean, that it must be possible to convert one document to a LaTeX / PDF document or e.g. to a HTML document.
Depends on: 1.8.26 Document Output
Solved by: 1.8.21 Output Diff of Two Versions, 1.8.23 Text Base Description Requirement References, 1.8.24 Text Base Description Choose Base Tags, 1.8.25 Output of Text Document must be configurable, 1.8.33 Output of HTML, 1.8.34 Output of PDF, 1.8.53 Output of Text Document Use Same Base

Id:	OutputTextDocument	**Priority:**	6.40	**Owner:**	development
Invented on:	2010-03-12	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.53 Output of Text Document Use Same Base

Description: The output of the text document **must** use the same description for all different outputs.
Rationale: I.e. with the help of one document description, it must be possible to generate different formats.
Note: This is implemented using so called 'topics'.
Depends on: 1.8.52 Output of Text Document, 1.6.1 Topic generics: Format independent
Solved by: 1.8.33 Output of HTML, 1.8.34 Output of PDF

Id:	OutputTextSameBase	**Priority:**	8.00	**Owner:**	development
Invented on:	2010-03-12	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.54 Output of Version Number

Description: *rmtoo* **must** support genration of a file which contains the Version Control System unique identifier for the used set of requirements.
Rationale: This is typically a version number; for git this is the hex hash string.
This is very important to automatically include the version number in different output documents.
Depends on: 1.8.27 Output of Elements

Id:	OutputVersion1	**Priority:**	6.40	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.55 Output XML File

Description: *rmtoo* **must** support genration of XML output.
Rationale: XML is used in many projects to import data or exchange data. This should be an implementation which supports output of one set of requirements.
Depends on: 1.8.20 Output of Different Artifacts
Solved by: 1.8.56 Output XML Example Implementation, 1.8.57 Output XML for GanttProject, 1.8.58 Output XML for GanttProject Second Generation

Id:	OutputXML	**Priority:**	10.00	**Owner:**	customers
Invented on:	2010-04-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.56 Output XML Example Implementation

Description: *rmtoo* **must** implement an example XML output class.
Rationale: To show how to implement (customer) specific XML output modules, this example XML output can be used as a reference implementation.
Depends on: 1.8.55 Output XML File

Id:	OutputXMLExample	**Priority:**	2.00	**Owner:**	customers
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.57 Output XML for GanttProject

Description: *rmtoo* **must** support generation of XML output which can be read in by the GanttProject.
Rationale: For experimentation there should be a output module implementation which supports output of the GanttProject XML file.
Note: This was suggested by a customer - but currently some details are unknown. So the first version will be a implementation to get the requirements sorted.
There are no more requirements for this feature. So it is seen as finished.
Depends on: 1.8.55 Output XML File

Id:	OutputXMLGanttProject	**Priority:**	2.00	**Owner:**	customers
Invented on:	2010-04-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.58 Output XML for GanttProject Second Generation

Description: *rmtoo* **must** support generation of topic based XML output which can be read in by the GanttProject.
Rationale: For experimentation there should be a output module implementation which supports output of the GanttProject XML file.
Note: This was suggested by a customer - but currently some details are unknown. So the first version will be a implementation to get the requirements sorted.
This is similar to the GantProject output (without a number) - but is now based on the topics.
Depends on: 1.8.55 Output XML File

Id:	OutputXMLGanttProject2	**Priority:**	10.00	**Owner:**	customers
Invented on:	2010-09-11	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.8.59 Priority output Include Effort Estimation

Description: The priority output list **must** include the effort estimation for each requirement. If no effort estimation is given, the output must be an empty field.
Rationale: This gives a good overview which requirements need which effort.
Depends on: 1.8.37 Output of Priority List, 1.5.4 Requirement Tag Effort Estimation

Id:	PrioOutputEft	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.60 Priority Output in Graph

Description: The priority output **must** be included in graph for all open requirements.
Rationale: This gives a great overview what must be done.
Depends on: 1.8.37 Output of Priority List

Id:	PrioOutputInGraph	**Priority:**	6.40	**Owner:**	development
Invented on:	2010-03-01	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.8.61 Priority Output LaTeX

Description: The priority output **must** have LaTeX format.
Rationale: There were first experiments using plain files (*.txt) — but this was not really usable — especially when classes were added.
Depends on: 1.8.37 Output of Priority List

Id:	PrioOutputLaTeX	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.8.62 Priority Output Order By Class

Description: The priority output **must** be ordered by class.
Rationale: It makes no sense to have one big common list of all prioritized requirments.
For real daily live use the lists must be separated and ordered by the class. So there is one list for the developer (e.g. the backlog) and one list for the SCRUM master who can then elaborate the highest prioritized requirements.
Depends on: 1.5.1 Class, 1.8.37 Output of Priority List

Id:	PrioOutputOrderByClass	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.9 Emacs Mode

1.9.1 Emace Mode Auto Fill Mode

Description: The Emacs mode *must* switch on auto-fill-mode by default.
Depends on: 1.9.8 Emacs Mode: Support Author

Id:	EMAutoFill	**Priority:**	0.79	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.9.2 Emace Mode Flyspell Mode

Description: The Emacs mode *must* switch on flyspell-mode by default.
Depends on: 1.9.8 Emacs Mode: Support Author

Id:	EMFlyspellMode	**Priority:**	0.79	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.9.3 Emacs Mode for Glossary

Description: There **must** be an Emacs mode which supports the glossary file format.
Depends on: 1.9.6 Emacs Mode for all Inputs, 1.4.16 Glossary must be available

Id:	EMGlossary	**Priority:**	7.00	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.9.4 Emacs Mode Highlight Tags

Description: The Emacs mode **must** provide tag highlighting.
Depends on: 1.9.12 Emacs Mode

Id:	EMHighLightTags	**Priority:**	0.00	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.9.5 Emacs Mode Indentation

Description: The Emacs mode **must** provide functionality for automatic indentation.
Depends on: 1.9.8 Emacs Mode: Support Author

Id:	EMIndentation	**Priority:**	0.79	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.9.6 Emacs Mode for all Inputs

Description: For each input format ther **must** be an Emacs mode which supports the file format.
Depends on: 1.9.12 Emacs Mode
Solved by: 1.9.3 Emacs Mode for Glossary, 1.9.7 Emacs Mode for Requirements, 1.9.9 Emacs Mode for Topics, 1.9.10 Emace Mode to Support Traceablility

Id:	EMInput	**Priority:**	1.32	**Owner:**	development
Invented on:	2010-08-06	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.9.7 Emacs Mode for Requirements

Description: There **must** be an Emacs mode which supports the requirements file format.
Depends on: 1.9.6 Emacs Mode for all Inputs, 1.4.23 rmtoo must work on Requirments

Id:	EMRequirements	**Priority:**	7.00	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.9.8 Emacs Mode: Support Author

Description: The Emacs Mode **must** be implemented in the way that it supports the author / writer of a requirement.
Note: This can be done by e.g. switching on some different commonly used minor modes like flyspell and auto-fill.
Depends on: 1.9.12 Emacs Mode
Solved by: 1.9.1 Emace Mode Auto Fill Mode, 1.9.2 Emace Mode Flyspell Mode, 1.9.5 Emacs Mode Indentation

Id:	EMSupportAuthor	**Priority:**	1.32	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	detailable				

1.9.9 Emacs Mode for Topics

Description: There **must** be an Emacs mode which supports the topics file format.
Depends on: 1.9.6 Emacs Mode for all Inputs, 1.6.6 Topics must be supported

Id:	EMTopics	**Priority:**	7.00	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.9.10 Emace Mode to Support Traceablility

Description: The Emacs mode *must* support tracability.
Rationale: Maybe this might include automatic timestamp update.
Depends on: 1.9.6 Emacs Mode for all Inputs, 1.4.25 Traceability

Id:	EMTraceability	**Priority:**	0.95	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.9.11 Emace Mode Value Highlighting

Description: The Emacs mode *must* highlight values which are limited to a set of possible values.
Rationale: E.g. in the Status line 'not done' and 'finished' should be highlighted — but these words should not highlighted in e.g. the Description section.
Depends on: 1.9.12 Emacs Mode

Id:	EMValueHighlight	**Priority:**	1.13	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.9.12 Emacs Mode

Description: There **must** be an Emacs mode which supports the text file format used by *rmtoo*.
Depends on: 1.4.13 Easy Editable
Solved by: 1.9.4 Emacs Mode Highlight Tags, 1.9.6 Emacs Mode for all Inputs, 1.9.8 Emacs Mode: Support Author, 1.9.11 Emace Mode Value Highlighting

Id:	EmacsMode	**Priority:**	1.89	**Owner:**	development
Invented on:	2010-02-11	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10 Documentation

1.10.1 Documentation Man Page

Description: *rmtoo* **must** come with a (*nix) man page describing the basic behaviour.
Rationale: This typically describes the input and output and all the parameters needed (but not the ideas behind).
Depends on: 1.10.3 Documentation
Solved by: 1.10.17 Man Page Emacs Mode, 1.10.22 Man Page Overview, 1.10.24 Documentation Man Page for Analytics, 1.10.27 Documentation Man Page for Artifacts, 1.10.28 Documentation Man Page for File Formats, 1.10.29 Documentation of Generic Man Page

Id:	DocManPage	**Priority:**	4.12	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.2 Documentation Slides

Description: For documentation purposes there **must** exists a slide show introducing the major features.
Rationale: Software not only needs to be good — also the 'marketing' aspect should be considered: the more people / companies *rmtoo* using, the more bug reports and comments there will be, the better *rmtoo* will be.
Depends on: 1.10.3 Documentation

Id:	DocSlides	**Priority:**	4.68	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.3 Documentation

Description: *rmtoo* **must** be documented.
Depends on: 1.3 rmtoo
Solved by: 1.10.1 Documentation Man Page, 1.10.2 Documentation Slides, 1.10.31 Version

Id:	Documentation	**Priority:**	5.50	**Owner:**	development
Invented on:	2010-02-12	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.4 Man Page Analytics Description Words

Description: *rmtoo* **must** come with a man page which describes the Description Words analytic module.
Depends on: 1.10.24 Documentation Man Page for Analytics

Id:	ManAnaDescWords	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-08-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.5 Man Page Analytics HotSpot

Description: *rmtoo* **must** come with a man page which describes the HotSpot analytic module.
Depends on: 1.10.24 Documentation Man Page for Analytics

Id:	ManAnaHotSpot	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-08-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.6 Man Page Analytics Requirement Topic Coherence

Description: *rmtoo* **must** come with a man page which describes the Requirement Topic Coherence analytic module.
Depends on: 1.10.24 Documentation Man Page for Analytics

Id:	ManAnaReqTopicCohe	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-08-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.7 Man Page Analytics Topic Coherence

Description: *rmtoo* **must** come with a man page which describes the Topic Coherence analytic module.
Depends on: 1.10.24 Documentation Man Page for Analytics

Id:	ManAnaTopicCohe	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-08-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.8 Man Page Artifact Backlog

Description: *rmtoo* **must** come with a man page which describes the backlog output.
Depends on: 1.10.26 Documentation Man Page for Artifacts Elements

Id:	ManArtBacklog	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.9 Man Page Artifact Elaboration List

Description: *rmtoo* **must** come with a man page which describes the output of the elaboration list.
Depends on: 1.10.26 Documentation Man Page for Artifacts Elements

Id:	ManArtElabList	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.10 Man Page Artifact LaTeX

Description: *rmtoo* **must** come with a man page which describes the LaTeX output.
Rationale: The man page should also give some hints how to use the LaTeX file and the configuration options for this modules.
Depends on: 1.10.25 Documentation Man Page for Artifacts Documents

Id:	ManArtLaTeX	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.11 Man Page Artifact Requirement Pricing

Description: *rmtoo* **must** come with a man page which describes the output of the pricing module.
Rationale: This documentation must describe the different input and output fields. It must also contain an overview over the process behind the pricing.
Depends on: 1.10.25 Documentation Man Page for Artifacts Documents

Id:	ManArtPricing1	**Priority:**	1.51	**Owner:**	development
Invented on:	2010-08-16	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.12 Man Page Artifact Requirement Dependency Graph

Description: *rmtoo* **must** come with a man page which describes the output requirement dependency graph.
Rationale: This documentation must describe the different colors (circle, font), lines, ...
Also this must include the configuration options.
Depends on: 1.10.25 Documentation Man Page for Artifacts Documents

Id:	ManArtRDG	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.13 Man Page Artifact Requirement Dependency Graph - version 2

Description: *rmtoo* **must** come with a man page which describes the output requirement dependency graph version 2.
Depends on: 1.10.25 Documentation Man Page for Artifacts Documents

Id:	ManArtRDG2	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.14 Man Page Artifact Requirements History Count

Description: *rmtoo* **must** come with a man page which describes the output of the requirements history count graph.
Depends on: 1.10.26 Documentation Man Page for Artifacts Elements

Id:	ManArtRHC	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.15 Man Page Artifact Version1

Description: *rmtoo* **must** come with a man page which describes the version1 output.
Rationale: The man page should include the configuration options for this modules.
Depends on: 1.10.26 Documentation Man Page for Artifacts Elements

Id:	ManArtVersion1	**Priority:**	1.16	**Owner:**	development
Invented on:	2010-08-18	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.16 Man Page Config

Description: *rmtoo* **must** come with a man page which describes the rmtoo configuration.
Depends on: 1.10.29 Documentation of Generic Man Page

Id:	ManConfig	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.17 Man Page Emacs Mode

Description: *rmtoo* **must** come with man pages describing the Emacs Mode.
Depends on: 1.10.1 Documentation Man Page
Solved by: 1.10.18 Man Page Emacs Mode Glossary, 1.10.19 Man Page Emacs Mode Requirements, 1.10.20 Man Page Emacs Mode Topic

Id:	ManEmacsMode	**Priority:**	2.06	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.18 Man Page Emacs Mode Glossary

Description: *rmtoo* **must** come with man pages describing the Emacs Mode for glossary.
Depends on: 1.10.17 Man Page Emacs Mode

Id:	ManEmacsModeGlo	**Priority:**	1.03	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.19 Man Page Emacs Mode Requirements

Description: *rmtoo* **must** come with man pages describing the Emacs Mode for requirements.
Depends on: 1.10.17 Man Page Emacs Mode

Id:	ManEmacsModeReq	**Priority:**	1.03	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.20 Man Page Emacs Mode Topic

Description: *rmtoo* **must** come with man pages describing the Emacs Mode for topics.
Depends on: 1.10.17 Man Page Emacs Mode

Id:	ManEmacsModeTic	**Priority:**	1.03	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.10.21 Man Page Invoking rmtoo

Description: *rmtoo* **must** come with a man page which describes the invoking of rmtoo.
Rationale: This must include the command line parameters.
Depends on: 1.10.29 Documentation of Generic Man Page

Id:	ManInvoking	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.22 Man Page Overview

Description: *rmtoo* **must** come with an overview man page.
Depends on: 1.10.1 Documentation Man Page

Id:	ManOverview	**Priority:**	2.06	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.23 Man Page Requirements Format

Description: *rmtoo* **must** come with a man page which describes the requirement input file format.
Depends on: 1.10.28 Documentation Man Page for File Formats

Id:	ManReqFormat	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.24 Documentation Man Page for Analytics

Description: *rmtoo* **must** come with (*nix) man pages describing the analytics modules.
Depends on: 1.10.1 Documentation Man Page
Solved by: 1.10.4 Man Page Analytics Description Words, 1.10.5 Man Page Analytics HotSpot, 1.10.6 Man Page Analytics Requirement Topic Coherence, 1.10.7 Man Page Analytics Topic Coherence

Id:	ManSecAnalytics	**Priority:**	3.09	**Owner:**	development
Invented on:	2010-08-06	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.25 Documentation Man Page for Artifacts Documents

Description: *rmtoo* **must** come with (*nix) man pages describing the generated output documents.
Depends on: 1.10.27 Documentation Man Page for Artifacts
Solved by: 1.10.10 Man Page Artifact LaTeX, 1.10.11 Man Page Artifact Requirement Pricing, 1.10.12 Man Page Artifact Requirement Dependency Graph, 1.10.13 Man Page Artifact Requirement Dependency Graph - version 2

Id:	ManSecArtDocuments	**Priority:**	2.32	**Owner:**	development
Invented on:	2010-09-10	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.26 Documentation Man Page for Artifacts Elements

Description: *rmtoo* **must** come with (*nix) man pages describing the possible generated elements.
Rationale: *rmtoo* is able to output a wide range of simple and small information which can be used in different documents but which are not a document itself.
Depends on: 1.10.27 Documentation Man Page for Artifacts
Solved by: 1.10.8 Man Page Artifact Backlog, 1.10.9 Man Page Artifact Elaboration List, 1.10.14 Man Page Artifact Requirements History Count, 1.10.15 Man Page Artifact Version1

Id:	ManSecArtElements	**Priority:**	2.32	**Owner:**	development
Invented on:	2010-09-10	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.27 Documentation Man Page for Artifacts

Description: *rmtoo* **must** come with (*nix) man pages describing the generated artifacts.
Depends on: 1.10.1 Documentation Man Page
Solved by: 1.10.25 Documentation Man Page for Artifacts Documents, 1.10.26 Documentation Man Page for Artifacts Elements

Id:	ManSecArtifacts	**Priority:**	3.09	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.28 Documentation Man Page for File Formats

Description: *rmtoo* **must** come with (*nix) man pages describing the different (input) file formats.
Depends on: 1.10.1 Documentation Man Page
Solved by: 1.10.23 Man Page Requirements Format, 1.10.30 Man Page Topic Format

Id:	ManSecFileFormats	**Priority:**	3.09	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.29 Documentation of Generic Man Page

Description: *rmtoo* **must** come with a (*nix) man page describing generic parts of the rmtoo.
Note: This includes e.g. the invoking of rmtoo.
Depends on: 1.10.1 Documentation Man Page
Solved by: 1.10.16 Man Page Config, 1.10.21 Man Page Invoking rmtoo

Id:	ManSecGeneric	**Priority:**	3.09	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.30 Man Page Topic Format

Description: *rmtoo* **must** come with a man page which describes the topics input file format.
Depends on: 1.10.28 Documentation Man Page for File Formats

Id:	ManTopicFormat	**Priority:**	1.55	**Owner:**	development
Invented on:	2010-07-26	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.10.31 Version

Description: Version information **must** be available.
Note: This is valid for the documentation and all program files: see Solved by dependencies.
Depends on: 1.10.3 Documentation
Solved by: 1.10.32 Version in Application, 1.10.33 Version in Documentation

Id:	Version	**Priority:**	5.50	**Owner:**	development
Invented on:	2011-03-14	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.32 Version in Application

Description: Version information **must** be available in all applications.
Note: This includes every single script which can be executed. If there is someday a GUI there must be a function to display the version.
Depends on: 1.10.31 Version

Id:	VersionApplication	**Priority:**	5.50	**Owner:**	development
Invented on:	2011-03-14	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.10.33 Version in Documentation

Description: Version information **must** be available in all documents.
Note: Documents are: man pages, presentations, requirement documents, graphs, ...
Depends on: 1.10.31 Version

Id:	VersionDocumentation	**Priority:**	5.50	**Owner:**	development
Invented on:	2011-03-14	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.11 Automatic Creation of Artifacts

1.11.1 Makefile

Description: Automatic generation of artifacts **must** be possible by using a Makefile.
Rationale: This gives the users a simple to 'make all' thing, which in turn calls the needed commands.
Depends on: 1.4.2 Automatic Generation of Results
Solved by: 1.11.2 Makefile Dependencies

Id:	Makefile	**Priority:**	2.40	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.11.2 Makefile Dependencies

Description: Dependency support for Makefile **must** be implemented.
Rationale: The major problem handling everything within a Makefile are the needed dependencies.
Depends on: 1.11.1 Makefile

Id:	MakefileDeps	**Priority:**	1.68	**Owner:**	development
Invented on:	2010-02-14	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.12 Testing

1.12.1 Test Before Packaging

Description: The packaging procedure **must** guarantee that all tests are successfully run before packaging.
Rationale: This gives the possibility to run a set of regression test and check the whole functionality of rmtoo.
Depends on: 1.13.2 Packaging, 1.12.5 rmtoo Automated Testing

Id:	TestBeforePack	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	implementable				

1.12.2 Test Integration

Description: For each requirement there **must** be a integration test which tests the requirement in a larger context.
Rationale: This tests the interaction between the different layers of implementation and makes sure that the interaction works.
Depends on: 1.12.5 rmtoo Automated Testing
Solved by: 1.12.3 Test Tool: python-nose

Id:	TestIntegration	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-03-10	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.12.3 Test Tool: python-nose

Description: Test **must** be started with *nosetests*.
Rationale: *nosetests* is a common used test tool which integrates fine in the *rmtoo* development environment.
It also can check the test coverage.
Depends on: 1.12.2 Test Integration, 1.12.4 Unit Testing, 1.4.30 Use Python

Id:	TestTool	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-03-10	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.12.4 Unit Testing

Description: For each code path there **must** be a unit test.
Rationale: Each class, function and method must be tested. Each decision and error condition must be tested.
Depends on: 1.12.5 rmtoo Automated Testing
Solved by: 1.12.3 Test Tool: python-nose

Id:	TestUnit	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-03-10	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.12.5 rmtoo Automated Testing

Description: Each feature of *rmtoo* **must** be automatically testable.
Rationale: This gives the possibility to run a set of regression test and check the whole functionality of rmtoo.
Depends on: 1.3 rmtoo
Solved by: 1.12.1 Test Before Packaging, 1.12.2 Test Integration, 1.12.4 Unit Testing

Id:	Testing	**Priority:**	10.00	**Owner:**	development
Invented on:	2010-03-04	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

1.13 Deployment

1.13.1 Debian Package

Description: *rmtoo* **must** be distributed as a Debian package.
Rationale: This is the main development platform — and currently the most used platform for *rmtoo*.
Depends on: 1.13.2 Packaging

Id:	PackDebian	**Priority:**	7.20	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	finished (None, , None h)
Class:	implementable				

1.13.2 Packaging

Description: *rmtoo* **must** be distributed as a package.
Rationale: This makes it very easy to install and update *rmtoo*.
Depends on: 1.4.12 Ease of Use
Solved by: 1.13.1 Debian Package, 1.12.1 Test Before Packaging

Id:	Packaging	**Priority:**	9.00	**Owner:**	development
Invented on:	2010-03-06	**Invented by:**	flonatel	**Status:**	not done
Class:	detailable				

Chapter 2

Test Cases

2.1 Example Test Case

Description: This is an example test case.
Expected Result: This is the result.
Note: Just for testing.

2.2 Example Test Case 2

Description: This is an example test case.
Expected Result: This is also a result.
Note: Just for testing.

Chapter 3

Status

3.1 Selected for Sprint

Prio	Chap	Requirement Id	EfE	Sum

3.2 Assigned

Prio	Chap	Requirement Id	EfE	Person	Date
10.00	1.5.17	Status	1	Florath	2011-04-27

3.3 Backlog

Prio	Chap	Requirement Id	EfE	Sum
10.00	1.12.1	Test Before Packaging	3	3
10.00	1.5.17	Status: Not done		3
10.00	1.8.47	Output Statistics about Work: Estimated End Date		3
10.00	1.8.45	Output Statistics about Work		3
8.00	1.8.43	Output Statistics about Persons: Units		3
8.00	1.8.42	Output Statistics about Persons: Time		3
8.00	1.8.41	Output Statistics about Persons: Relation		3
8.00	1.8.40	Output Statistics about Persons		3
8.00	1.8.18	Output of Pricing Table: Input Cells	5	8
8.00	1.8.12	Output of Pricing Table: Input Cell Comment	5	13
5.60	1.8.3	Evaluate Pricing Information: Compliance Checks	8	21
5.60	1.8.2	Evaluate Pricing Information: Completeness Checks	8	29
2.40	1.11.1	Makefile	5	34
2.06	1.10.17	Man Page Emacs Mode	3	37

Prio	Chap	Requirement Id	EfE	Sum
1.68	1.11.2	Makefile Dependencies	8	45
1.55	1.10.7	Man Page Analytics Topic Coherence	3	48
1.55	1.10.6	Man Page Analytics Requirement Topic Coherence	3	51
1.55	1.10.5	Man Page Analytics HotSpot	3	54
1.55	1.10.4	Man Page Analytics Description Words	3	57
1.16	1.10.15	Man Page Artifact Version1	3	60
1.16	1.10.14	Man Page Artifact Requirements History Count	3	63
1.16	1.10.13	Man Page Artifact Requirement Dependency Graph - version 2	3	66
1.16	1.10.12	Man Page Artifact Requirement Dependency Graph	3	69
1.16	1.10.10	Man Page Artifact LaTeX	3	72
1.16	1.10.9	Man Page Artifact Elaboration List	3	75
1.16	1.10.8	Man Page Artifact Backlog	3	78
1.03	1.10.20	Man Page Emacs Mode Topic	3	81
1.03	1.10.19	Man Page Emacs Mode Requirements	3	84
1.03	1.10.18	Man Page Emacs Mode Glossary	3	87

3.4 Requirements Elaboration List

Prio	Chap	Requirement Id	EfE	Sum
10.00	1.3	rmtoo	5	5
10.00	1.4.29	Use Filename as Requirement ID	13	18
10.00	1.12.5	rmtoo Automated Testing	3	21
10.00	1.12.4	Unit Testing	13	34
10.00	1.12.3	Test Tool: python-nose	5	39
10.00	1.12.2	Test Integration	13	52
10.00	1.4.22	Requirement ID	13	65
10.00	1.4.20	Processing	21	86
10.00	1.8.58	Output XML for GanttProject Second Generation	13	99
10.00	1.8.55	Output XML File	8	107
10.00	1.8.39	Output Requirements	21	128
10.00	1.8.38	Output Requirements: Preserve Everything which is Possible	21	149
10.00	1.8.29	Generic Output Requirements	13	162
10.00	1.8.20	Output of Different Artifacts	34	196
10.00	1.8.19	Output: Include Version Identifier of Requirements	3	199
10.00	1.4.18	Graphical User Interface	34	233
10.00	1.4.17	Different Inputs	8	241
10.00	1.4.16	Glossary must be available	8	249
10.00	1.4.11	Constraints Inheritance	3	252

Prio	Chap	Requirement Id	EfE	Sum
10.00	1.4.10	Requirement Constraints	3	255
10.00	1.4.9	Configuration	8	263
10.00	1.4.8	Default Configuration for maximum line length using in Topics	13	276
10.00	1.4.7	Configuration for maximum line length using in Topics	13	289
10.00	1.4.6	Configuration of Developers	8	297
10.00	1.4.5	Default Configuration	13	310
10.00	1.4.4	Configuration Check	13	323
9.00	1.4.24	Simplicity	21	344
9.00	1.13.2	Packaging	3	347
9.00	1.4.12	Ease of Use	3	350
8.00	1.8.36	Output of Pricing Table	13	363
8.00	1.8.17	Output of Pricing Table: Input Cell Material	5	368
7.00	1.9.9	Emacs Mode for Topics	8	376
7.00	1.9.3	Emacs Mode for Glossary	8	384
6.40	1.8.21	Output Diff of Two Versions	13	397
5.60	1.8.7	Evaluate Pricing Information	8	405
5.60	1.8.1	Evaluate Pricing Information: Checks	8	413
5.50	1.10.33	Version in Documentation	3	416
5.50	1.10.32	Version in Application	3	419
5.50	1.10.31	Version	5	424
5.50	1.10.3	Documentation	3	427
5.40	1.4.14	Easy Extensible	5	432
4.12	1.10.1	Documentation Man Page	5	437
3.09	1.10.29	Documentation of Generic Man Page	5	442
3.09	1.10.28	Documentation Man Page for File Formats	5	447
3.09	1.10.27	Documentation Man Page for Artifacts	5	452
3.09	1.10.24	Documentation Man Page for Analytics	5	457
3.00	1.4.2	Automatic Generation of Results	3	460
2.70	1.4.13	Easy Editable	5	465
2.32	1.10.26	Documentation Man Page for Artifacts Elements	8	473
2.32	1.10.25	Documentation Man Page for Artifacts Documents	5	478
1.89	1.9.12	Emacs Mode	8	486
1.35	1.4.25	Traceability	13	499
1.32	1.9.6	Emacs Mode for all Inputs	8	507
1.13	1.9.11	Emace Mode Value Highlighting	3	510
0.95	1.9.10	Emace Mode to Support Traceablility	3	513

3.5 Finished

Chap	Requirement Id	EfE	Person	Date	Time	Rel
1.4.23	rmtoo must work on Requirments	3				

Chap	Requirement Id	EfE	Person	Date	Time	Rel
1.4.36	Analytics: Requirement Topic Coherence	5				
1.5.3	Requirements Description					
1.10.11	Man Page Artifact Requirement Pricing	3				
1.5.15	Requirement Tags: Basics	3				
1.4.21	Requirement generics: Order of Tags	8				
1.6.4	Topic SubTopics					
1.8.24	Text Base Description Choose Base Tags	13				
1.5.5	Requirements Invented By					
1.8.10	Output of Pricing Table: Computation of Sum	5				
1.5.4	Requirement Tag Effort Estimation					
1.4.26	Txt Comment Semantics					
1.4.15	Files in File System	13				
1.8.5	Evaluate Pricing Information: Graph Color	8				
1.8.56	Output XML Example Implementation	3				
1.9.7	Emacs Mode for Requirements	8				
1.8.33	Output of HTML	13				
1.9.1	Emace Mode Auto Fill Mode					
1.5.16	Effort Estimation Measure					
1.4.34	Version Control System: Latest Version	3				
1.8.35	Output of PDF Configuration	13				
1.8.13	Output of Pricing Table: Input Cell Compliant	5				
1.5.1	Class					
1.8.15	Output of Pricing Table: Input Cell Dayrate	5				
1.7.4	Graph Check for Strongly Connected Component					
1.4.30	Use Python					
1.10.23	Man Page Requirements Format					
1.7.8	Requirement Priority Check					
1.10.16	Man Page Config					
1.7.2	Graph Check for Connected Component					
1.8.11	Output of Pricing Table: Computation	5				
1.5.16	Effort Estimation Allowed Measure Unit					
1.8.8	Modular Output					
1.8.6	Evaluate Pricing Information: Graph Costs	8				
1.5.17	Priority Computation					
1.7.7	Checks	8				
1.8.54	Output of Version Number	5				
1.8.25	Output of Text Document must be configurable	8				
1.8.30	Output of Dependency Graph					

Chap	Requirement Id	EfE	Person	Date	Time	Rel
1.6.5	Topic Tags	3				
1.4.36	Analytics: HotSpots	5				
1.9.2	Emace Mode Flyspell Mode					
1.5.14	Requirement Tags: Basics	3				
1.5.10	Requirements Status					
1.6.6	Topics must be supported	3				
1.8.32	Output of Dependency Graph Configuration					
1.8.27	Output of Elements	8				
1.7.6	Syntax Checks					
1.9.5	Emacs Mode Indentation					
1.5.12	Requirements Type					
1.8.52	Output of Text Document	8				
1.8.57	Output XML for GanttProject	8				
1.6.3	Topic Name					
1.10.2	Documentation Slides					
1.5.2	Requirments Class					
1.5.13	Requirement Tags	3				
1.7.1	Graph Checks					
1.8.9	Output of Pricing Table: Computation of Dependent Costs	5				
1.8.16	Output of Pricing Table: Input Cell Dependent On	5				
1.4.33	Version Control System: History Interval Usage	3				
1.8.23	Text Base Description Requirement References	13				
1.4.36	Analytics: Topic Coherence	5				
1.9.4	Emacs Mode Highlight Tags					
1.4.19	Open Source rmtoo					
1.4.32	Version Control System: History Interval	5				
1.7.9	No Directed Circles Allowed					
1.6.1	Topic generics: Format independent	13				
1.4.36	Analytics: Description Words	5				
1.13.1	Debian Package	8				
1.8.59	Priority output Include Effort Estimation					
1.8.61	Priority Output LaTeX					
1.8.34	Output of PDF	13				
1.4.1	Analytics	5				
1.10.22	Man Page Overview					
1.8.4	Evaluate Pricing Information: Graph	8				
1.5.8	Requirements Owner					
1.5.11	Topic					
1.8.62	Priority Output Order By Class					
1.8.60	Priority Output in Graph					
1.9.8	Emacs Mode: Support Author	8				
1.8.26	Document Output	8				
1.10.30	Man Page Topic Format					

Chap	Requirement Id	EfE	Person	Date	Time	Rel
1.8.31	Output of Dependency Graph: Topics based					
1.7.5	Semantic Checks					
1.8.53	Output of Text Document Use Same Base	13				
1.5.7	Requirements Name					
1.4.35	Version Control System	13				
1.4.31	Use Txt					
1.4.28	Txt Empty Lines					
1.4.27	Txt Comments					
1.6.2	Topic generics: Order of Tags	8				
1.8.14	Output of Pricing Table: Input Cell Day Count	5				
1.10.21	Man Page Invoking rmtoo	3				
1.7.3	Graph Check for Exact One Master Requirement					
1.8.37	Output of Priority List					
1.5.17	Priority Format					
1.5.9	Requirement Priority					
1.4.3	Completed Requirement					
1.5.6	Requirements Invented On					
1.8.44	Output of Number of Requirements					
1.8.22	Output of Assigned List	3	Florath	2011-04-27	1	3.00
1.8.50	Output Statistics about Work: Relation	2	Florath	2011-04-27	1	2.00
1.5.17	Status: Finished	5	Florath	2011-04-27	2	2.50
1.8.48	Output Statistics about Work: Finished	2	Florath	2011-04-27	1	2.00
1.8.28	Output of Finished List	3	Florath	2011-04-27	1	3.00
1.5.17	Status: Assigned	3	Florath	2011-04-27	1	3.00
1.5.17	Status: Assigned Value	5	Florath	2011-04-27	2	2.50
1.5.17	Status: Finished Value	3	Florath	2011-04-27	1	3.00
1.8.49	Output Statistics about Work: Not Done	2	Florath	2011-04-27	1	2.00
1.8.46	Output Statistics about Work: Assigned	2	Florath	2011-04-28	1	2.00
1.8.51	Output Statistics about Work: Start Date	2	Florath	2011-04-28	1	2.00

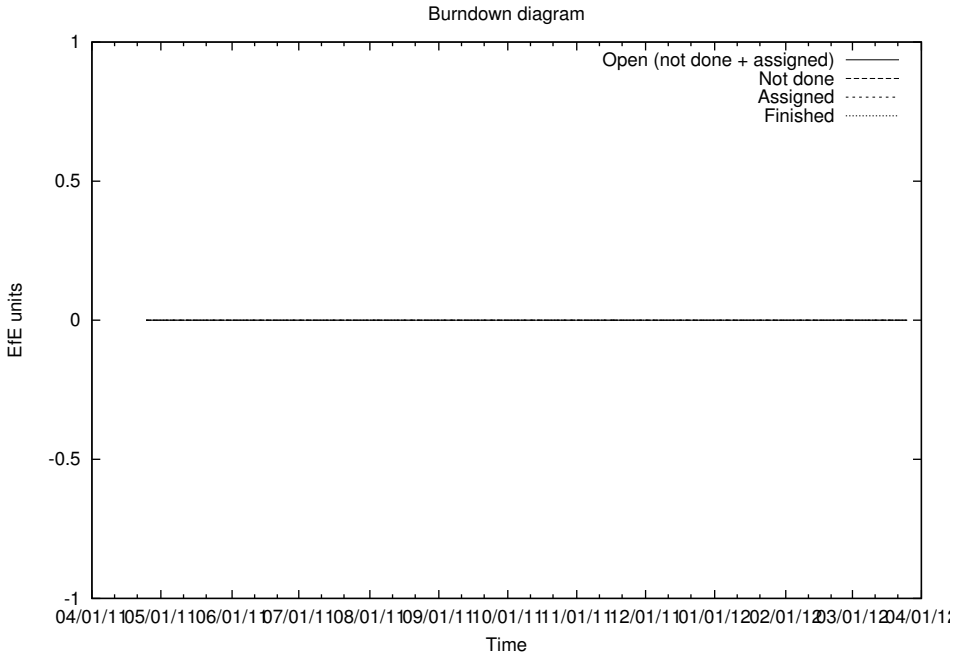
3.6 Statistics

Start date	2011-04-25	
Not done	87	EfE units
Assigned	1	EfE units
Finished	364	EfE units
Finished (duration given)	32	EfE units
	13	hours
Relation	2.46	EfE units / hour
Estimated Not done	35.34	hours
Estimated End date	2079-06-05	

Chapter 4

Statistics

4.1 Burndown Diagram



4.2 Requirements Count Statistics