



Requirements Management Tool rmtoo

Overview & Feature Set

by flonatel GmbH & Co. KG

Version 7

Content

- 1.The idea behind rmtoo
- 2.Features
- 3.Introduction
- 4.10 dos and 10 don'ts
- 5.Future

1.The idea behind rmtoo

Or: How to solve problems

Idea

- Requirements Management is about solving problems.
- This process is iterative: one solution may result in new problems which in turn form the basis for new solutions.
- Problem solving is mostly a *Divide et Impera*: divide the problem into smaller ones and solve them.

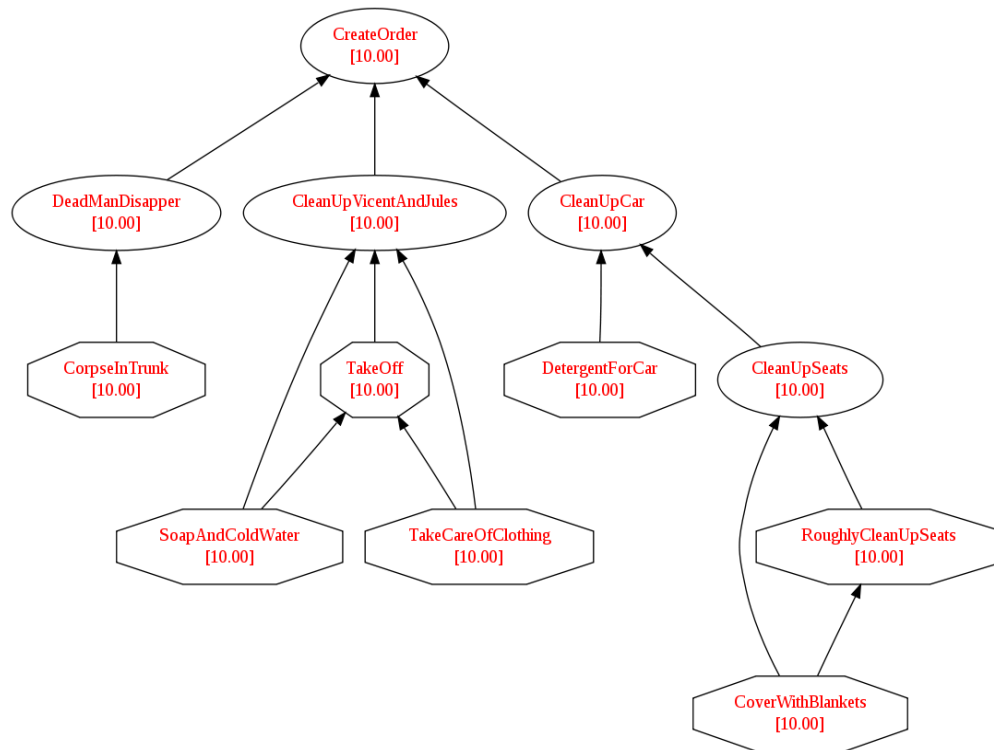
Idea: Example (1/3)

- Mr. Wulf (from the movie Pulp Fiction) solves problems.
- He divides his 'problem' into three parts:
 - **Clean up car**
 - **Clean up Vincent and Jules**
 - **Make dead man disappear**

Idea: Example (2/3)

- Then he solves these problems – one by one by dividing them up into smaller ones.
- Example: Clean up Vincent and Jules
 - take off clothes then
 - clean with soap and cold water.

Idea: Example (3/3)



- This is the problem – solution tree.
- Each problem can be seen as a requirement.
- Many solutions mean new problems – solutions are also requirements.

rmtoo

- Created for problem solving.
- It's a tool to support you. Not the other way round!

2.Features

Features (1/3)

- *rmtoo* works on plain text files.
- *rmtoo* supports version control systems.
- *rmtoo* supports a wide range of output formats.
- *rmtoo* supports a unique way of structuring requirements: topics.

Features (2/3)

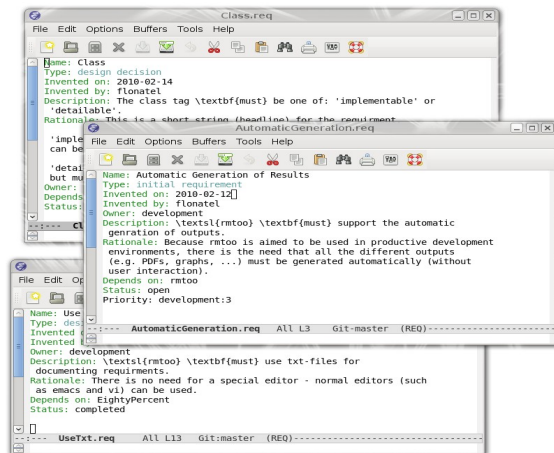
- *rmtoo* can be fully integrated into existing development environments.
- *rmtoo* checks the quality of requirements with the help of heuristics.
- *rmtoo* works on local files (you have the control).
- *rmtoo* has minimal system dependencies.

Features (3/3)

- *rmtoo* comes with a complete specification in *rmtoo* format itself (unique!).
- *rmtoo* supports you: fast and structured working is a guaranteed.
- *rmtoo* enforces you to 'Think before you write'.

3.Introduction

Overview



rmtoo



3 Initial Requirements

This is the section containing all the initial requirements.

3.1 rmtoo must work on Requirements

Description: rmtoo must work on requirements.

Rationale: That's what it's about.

Depends on: 2.1 rmtoo

Id	Requirement
2.1	Requirements
2.2	Requirements
2.3	Requirements
2.4	Requirements
2.5	Requirements
2.6	Requirements
2.7	Requirements
2.8	Requirements
2.9	Requirements
2.10	Requirements

3.2 Agile Development Process

Description: The development process which is mainly supported by rmtoo must be agile. Rationale: This gives a hint for which features to add to rmtoo. Also the development process of rmtoo is also agile. (Rmtoo will be because this feature is not implemented in the process.)

Depends on: 2.1 rmtoo

Id	Requirement
2.1	Agile Development
2.2	Agile Development



Backlog

Priority	Requirement Id
10	2.1 Document all the...
9	2.2 Documentation: Main Page
8	10.1 Makefile
7	10.2 Makefile Dependencies
6	7.5 Feature: Make Auto Fill Mode
6	7.6 Feature: Make Format Mode
6	7.7 Feature: Make Evaluation

Requirements Elaboration

Priority	Requirement Id
11	2.1 Documentation
7	7.2 Feature: Make to Support Traceability
7	7.3 Feature: Make
6	3.5 Easy Extensible
6	3.7 Easy Editable
5	3.3 Traceability
5	3.2 Agile Development Process
3	3.1 rmtoo must work on Requirements
3	2.1 rmtoo
2	3.6 Automatically Generation of Results
2	3.3 Higher Percent Risk
2	3.2 Output of Different Artifacts
1	3.2 Documentation of the Graph output

Input Data

Command line tool

Output Artifacts

Basic Facts (1/4)

- *rmtoo* is a minimalistic non-interactive requirements management tool.
- *rmtoo* works on data stored in the file system (plain text files).
- *rmtoo* is a command line tool which reads in files and creates output.

Basic Facts (2/4)

- *rmtoo* supports different output formats and artifacts. For example:
 - PDF / LaTeX
 - XML / HTML
 - GanttProject
 - Requirements Dependency Graph
- *rmtoo* can be easily extended by adding new output modules.

Basic Facts (3/4)

- *rmtoo* data files can be handled by standard *nix commands (emacs, vi, grep, awk, streplace, sed, ...).
- *rmtoo* runs largely on the same hardware and operating system where the development is done – no need for a dedicated machine.
- *rmtoo* baselining can be handled by a version control system.

Basic Facts (4/4)

- *rmtoo* needs minimal resources
 - Netbook CPU
 - 10MByte disk space for installation and documentation
 - RAM usage dependent on size and duration of project
- *rmtoo* comes with a complete template project. Using this makes it simple to start own projects using *rmtoo*.

rmtoo does not

- *rmtoo* has no GUI.
- *rmtoo* does not come with a database.
- *rmtoo* does not have an integrated editor.
- *rmtoo* does not provide a UML editor.
- *rmtoo* does not have import capabilities (e.g. from a spreadsheet or a word processing document).

License

- *rmtoo* is Open Source.
- *rmtoo* is free.
- *rmtoo* is licensed under GPLv3.
- Commercial support is available.

4. 10 dos and 10 don'ts

reasons for using rmtoo

reasons for not using rmtoo

Use rmtoo

- when you want to be fast.
- when you want to have full control over the input files.
- when you need one set of requirements to be converted to various different output formats.
- when you need structured requirements.
- when you need a tool that supports you.

Use rmtoo

- when you need functionality which is easily extensible.
- when you want a single source of information for all requirement related materials.
- when you want the requirements for the requirements management tool you are using.

Use rmtoo

- when you want to store the requirements with the implementation source code.
- when you want to include building requirements in your continuous build environment.

Do not use rmtoo

- when a GUI is needed.
- when a web-based approach is needed.
- when you do not need full control over your data.
- when you want to adapt your behavior to a tool's processes.
- when you want to put your internal data into another companies computer.

Do not use rmtoo

- when you want to work with a tool which tries to do everything.
- when you want to be dependent on one vendor.
- when you need a traceability matrix.
- if you want to make dozens of mouse-clicks to get one output document.
- if you want to spend money.

5. Future

Future / Plans

- Some features are currently missing but planned for the upcoming releases
 - Traceability
 - Better support writing requirements in Emacs mode
- Community, User and Customer driven

Thank you!



Copyright

This document is distributed under the creative commons license 'Attribution-Noncommercial-No Derivative Works 3.0 Germany'

© 2010 flonatel GmbH & Co. KG

<http://www.flonatel.de/projekte/rmtoo>