

rmtoo – Traceability

Kristoffer Nordström

February 14, 2019

Introduction

Purpose and Outline

This slideshow provides an overview over the new features not mentioned in the presentation below.

- Provide an introduction into the traceability features
- Export and import of *xlsx* files
- Show missing features and how to solve them

An introduction into *rmtoo* is available here and in more detail here.

Traceability

Imagine your customer has provided you with *the perfect* requirements document. You've written your code and tests and your traceability matrix is perfect.

Enter the *change request*:

- some requirements have changed, and
- some other code has to be changed as well.

- Keep requirements and code synced
- Update test specification and keep tests synced
- Traceability matrix must remain correct
 - Filled out completely

What you want

- Code and requirements belong together
 - Same repository
 - CI runs unit-tests
 - CI creates traceability matrix automatically
- Unit-tests point to test specification
 - Changes to tests must be reflected in test specification
 - Manual work
 - No silver bullet (only golden ones)
 - *Backwards* arrow in V-model
- Changes to test specification must fail CI toolchain
 - For unchanged unit-tests
 - Link not obvious from specification
 - CI tool must validate
 - *Forwards* arrow in V-model

Available Solutions

- Web based solutions
 - We've just separated our requirements from our code
 - Does it fail your build?
 - Probably not and you're late, so ship anyways
 - you'll be in trouble
 - Your web-page isn't working in three years when the customer comes around
- DOORS
 - They're over there →
- Manual Reports
 - Manually verify a traceability matrix twice and your engineers will use that golden bullet against you.
 - Processes will not be followed, unless customer insists, e.g., ECSS or EN50128 is required.
 - YMMV, but questionable correctness

rmtoo Traceability

Every specification-item has a name, e.g., SWC-TS-102, and a unique hash (more later). Every unit-test lists the specifications it solves.

The following unit-test will test the aforementioned *software component test specification* item 102.

```
def test_adding_req(self, record_property):
    record_property('req', 'SWC-TS-102-96ac8522')
    assert True
```

Backwards — Traceability Matrix Input

- Running `pytest` will yield a `xunit` file `result.xml`
- This file is used to generate the traceability matrix
- Any unit testing framework can be used if the XML is equivalent

```
<testcase time="0.048" name="rmttest_adding_req" line="89"  
file="rmtoo/tests/RMTTest-Output/RMTTest-Xls.py"  
classname="rmtoo.tests.RMTTest-Output.RMTTest-Xls.RMTTestC  
  <properties>  
    <property name="req" value="SWC-TS-102-96ac8522"/>  
  </properties>  
</testcase>
```

The previously test requirement SWC-TS-102 will change and with it its hash-value.

Hence the test on the previous page will fail the traceability matrix because the hash *96ac8522* has changed.

Time for your engineer to ensure if/what needs to change.

```
def test_adding_req(self, record_property):  
    record_property('req', 'SWC-TS-102-96ac8522')  
    assert True
```

- SHA256 hash calculated over sum of
 - Description,
 - Titel, and
 - Verification Method (if available)
- Rationale is only informative

From the test specification in *rmtoo's* testspe folder.

Appendix A

Traceability Matrix

Table A.1: Traceability Matrix Table

Req. ID	Status	UT
SWC-TS-1	finished	
SWC-TS-100	open	open
SWC-TS-101	passed	passed
SWC-TS-102	passed	passed
SWC-TS-103	passed	passed
SWC-TS-110	passed	passed
SWC-TS-120	passed	passed
SWC-TS-140	open	open
SWC-TS-141	passed	passed
SWC-TS-142	passed	passed
SWC-TS-143	passed	passed
SWC-TS-144	passed	passed
SWC-TS-145	passed	passed
SWC-TS-146	passed	passed

- Write Parser for *Test Reports*
 - Documents with the correct identifier automatically solve the specification
 - Document Formats:
 - docx (maybe with pandoc)
 - L^AT_EX
- Cross-Document References
 - *Solved by external*
 - *Depends on external*
 - Same principle applies

Excel Support

- Good-enough GUI
- Suits love it
- The *Truth* is still in your repository

Final Thoughts

```
pip3 install git+https://github.com/kown7/rmtoo.git@master
```

At the moment the traceability features haven't been merged. Pull request is pending.

- Never test against your requirements
 - Always write some form of test specification
 - Consider cucumber for acceptance testing