

FOSDEM'22

Testing Requirements Documents

Kristoffer Nordström
5 & 6 February 2022
`info@sltoo.dev`

2022-01-21

FOSDEM'22
Testing Requirements Documents

FOSDEM'22
Testing Requirements Documents

Kristoffer Nordström
5 & 6 February 2022
`info@sltoo.dev`

- Motivation
- Theory
- Practical Aspects
- Demonstration

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Welcome

- Welcome everyone; it's an honour to talk on this stream.
- When talking about requirements, we should also define the requirements of the problem we're trying to solve.
- Storytime

Welcome

- Motivation
- Theory
- Practical Aspects
- Demonstration

- Fully automated traceability matrix
- Consistent and up-to-date documents from the source
- Store (requirements') meta-information with code

└ Motivation

- It's fine manually. Requirements change find minimal delta-test? Change more than once. Hence we want this fully automated.
- Review multiple documents for a different contractor. No consistent set of documents were provided
- Jon Holt's term in introduction to MBSE: Documents are a live *view* of the system (not pretty pictures). CI/CD for software, why not for documents?
- Every ALM tool will do this for you so far
- It came with one document that contained all requirements and use-cases. It appeared to have been exported from some web-based tool. All the information therein is lost
- Track the motivation for the changes → same as with source code
- Also: Too technical, interface with management/business side
- myself: (FPGA) test-engineer for various space/railway applications 8 yrs
- 4'

- Fully automated traceability matrix
- Consistent and up-to-date documents from the source
- Store (requirements') meta-information with code

2022-01-21

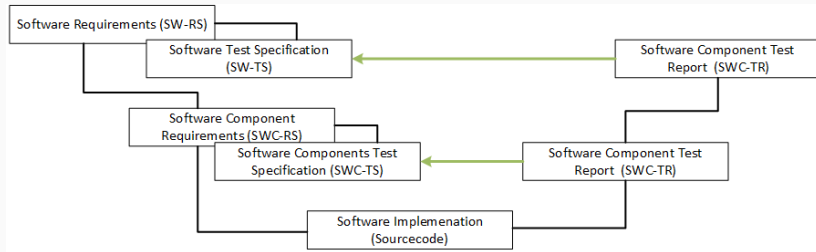
FOSDEM'22
Testing Requirements Documents
└ Theory

Theory

Theory

Requirements and Traceability

- Requirements across system hierarchies
 - Implies the need for traceability
- *Traceability* from and to specification items
- Directions
 - Forward (*Impact*) from requirements specification to dependant documents
 - Backwards from verification artefacts to specification



2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Theory

└ Requirements and Traceability

Here Traceability only from items/issues

Requirements and Traceability

- Requirements across system hierarchies
 - Implies the need for traceability
- Traceability from and to specification items
- Directions
 - Forward (*Impact*) from requirements specification to dependant documents
 - Backwards from verification artefacts to specification



- *Requirement A*
 - Red button to shut down system
- *Implementation a* says implemented A
 - Traceability can be automated
 - Machine-readable
- What if *A* changes?
 - *A* knows nothing of *a*



- *Requirement A*
 - Red button to shut down system
- *Implementation a* says implemented A
 - Traceability can be automated
 - Machine-readable
- What if *A* changes?
 - *A* knows nothing of *a*



- *Requirement A*
 - Green button with large friendly letters: don't panic
- *Implementation a* says implemented A
 - Traceability can be automated
 - Machine-readable
- What if *A* changes?
 - A knows nothing of *a*
 - Traceability isn't given anymore



- *Requirement A*
 - Green button with large friendly letters: don't panic
- *Implementation a* says implemented A
 - Traceability can be automated
 - Machine-readable
- What if *A* changes?
 - A knows nothing of *a*
 - Traceability isn't given anymore



- *Requirement A-1.0*
 - Red button to shut down system
- *Implementation a says implemented A-1.0*
 - *A-1.0* (red button) changes to *A-2.0* (green button)
 - Use hashes instead of semantic versioning
 - Calculated automatically



Why hashes: no tool or manual changes required, it's all derived

Let's see how it looks on an example

- *Requirement A-1.0*
 - Red button to shut down system
- *Implementation a says implemented A-1.0*
 - *A-1.0* (red button) changes to *A-2.0* (green button)
 - Use hashes instead of semantic versioning
 - Calculated automatically



Example Requirement

Name: VCD Writer Inputs
Topic: ReqsDocument
Description: The output from ...
Rationale: Make the process as ...
Status: external
Owner: development
Effort estimation: 1
Invented on: 2020-05-30
Invented by: default
Type: requirement

*Hash is calculated over Name,
Description and Verification
Method*

```
$ sha256sum ${Name} \  
${Description} \  
${VerifMethod}
```

VCD Writer Inputs SW-AS-501

The output from *AggregatorInterface* can be fed into the *VCD Writer* to create an identical output file.

Rationale: Make the process as transparent as possible.

Depends on: SW-AS-500

Id:	SW-AS-501	Priority:	0.0	Owner:	development
Invented on:	2020-05-30	Invented by:	default	Status:	passed
Class:	detaillable	Hash:	f8d68d11		

Example from pymergevcd's architecture specification

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Theory

└ Example Requirement

- Every requirement is in its own file
- A from previous slides is now SW-AS-501
- Version n.0 is now F8D68D11

Example Requirement

```
Name: VCD Writer Inputs  
Topic: Requirements  
Description: The output from ...  
Rationale: Make the process as ...  
Status: external  
Owner: development  
Effort estimation: 1  
Invented on: 2020-05-30  
Invented by: default  
Type: requirement  
  
Hash is calculated over Name,  
Description and Verification  
Method  
$ sha256sum ${Name} \  
${Description} \  
${VerifMethod}
```

```
VCD Writer Inputs SW-AS-501  
The output from AggregatorInterface can be fed into the VCD Writer to create an identical  
output file.  
Rationale: Make the process as transparent as possible.  
Depends on: SW-AS-500  
  
Id: SW-AS-501 Priority: 0.0 Owner: development  
Invented on: 2020-05-30 Invented by: default Status: passed  
Class: detaillable Hash: f8d68d11
```

Example from pymergevcd's architecture specification

Testing the Example Requirement

- Requirement ID: SW-AS-501
- Hash: F8D68D11

Test Code

```
def test_read_write_engines(record_property, dummy_vcd_file):  
    """Write-back from read file, equal output"""  
    record_property('req', 'SW-AS-501-f8d68d11')  
    record_property('req', 'SW-AS-500-4c1a395a')  
    ...  
    assert filecmp.cmp(dummy_vcd_file, ofile)
```

xUnit Output

```
<testcase line="20" name="test_read_write_engines" time="2.830">  
  <properties>  
    <property name="req" value="SW-AS-501-f8d68d11"/>  
    <property name="req" value="SW-AS-500-4c1a395a"/>  
  </properties>  
</testcase>
```

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Theory

└ Testing the Example Requirement

Testing the Example Requirement

- Requirement ID: SW-AS-501
- Hash: F8D68D11

Test Code

```
def test_read_write_engines(record_property, dummy_vcd_file):  
    """Write-back from read file, equal output"""  
    record_property('req', 'SW-AS-501-f8d68d11')  
    record_property('req', 'SW-AS-500-4c1a395a')  
    ...  
    assert filecmp.cmp(dummy_vcd_file, ofile)
```

xUnit Output

```
<testcase line="20" name="test_read_write_engines" time="2.830">  
  <properties>  
    <property name="req" value="SW-AS-501-f8d68d11"/>  
    <property name="req" value="SW-AS-500-4c1a395a"/>  
  </properties>  
</testcase>
```

Table A.1: Traceability Matrix Table

Req. ID	Status	UT
SW-AS-1	passed	passed
SW-AS-100	open	open
SW-AS-101	open	open
SW-AS-102	open	open
SW-AS-200	failed	failed
SW-AS-201	passed	passed
SW-AS-202	passed	passed
SW-AS-300	passed	passed
SW-AS-301	open	open
SW-AS-302	open	open
SW-AS-500	passed	passed
SW-AS-501	passed	passed

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Theory

└ Traceability Matrix

Table A.1: Traceability Matrix Table

Req. ID	Status	UT
SW-AS-1	passed	passed
SW-AS-100	open	open
SW-AS-101	open	open
SW-AS-102	open	open
SW-AS-200	failed	failed
SW-AS-201	passed	passed
SW-AS-202	passed	passed
SW-AS-300	passed	passed
SW-AS-301	open	open
SW-AS-302	open	open
SW-AS-500	passed	passed
SW-AS-501	passed	passed

Now it should be straightforward to integrate it into any CI pipeline

- Integration for every output document
- Match *open* and/or *failed* issues
 - Left as an exercise for the reader
- Example for *failed* issues

```
$ bash -ec 'test "$(grep -c failed \
arch/artifacts/tracematrix.tex)" -eq "0"'
```

- Integration for every output document
 - Match open and/or failed issues
 - Left as an exercise for the reader
 - Example for failed issues
- ```
$ bash -ec 'test "$(grep -c failed \
arch/artifacts/tracematrix.tex)" -eq "0"'
```

2022-01-21

FOSDEM'22  
Testing Requirements Documents  
└─ sltoo in Practice

sltoo in Practice

## sltoo in Practice

---

- *s/too* (currently) a fork of *rmtoo*
  - text-file based req. tracking tool
- Tracking requirements in text files with git
  - might be ideal for engineering department
- Defining system behaviour is a team effort
- Solution not for everyone / command-line unfamiliar UI

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ sltoo in Practice

└ sl... what?

- Guess: most people aren't familiar with the tooling/console →  
Which tool is "everyone" familiar with?

- *sltoo* (currently) a fork of *rmtoo*
  - text-file based req. tracking tool
- Tracking requirements in text files with git
  - might be ideal for engineering department
- Defining system behaviour is a team effort
- Solution not for everyone / command-line unfamiliar UI

## Excel Workflow (I)

- Defining system collaborative effort
- Familiarity / Ease-of-use
- Consistency of Documents
  - The *Truth* is always in your repository
  - Templating for branding
- Works if all you've got is Office and E-Mail
- Getting Started: Edit example Excel-Sheet

[illegible]

2022-01-21

## FOSDEM'22

## Testing Requirements Documents

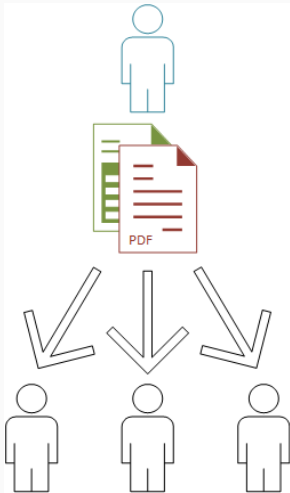
## └ Excel Workflow (I)

Simply use the excel sheet as long as no documents or verification tasks are necessary

## Stepping stone for requirements tracking

- Defining system collaborative effort
- Familiarity / Ease-of-use
- Consistency of Documents
  - The Truth is always in your repository
  - Templating for branding
- Works if all you've got is Office and E-Mail
- Getting Started: Edit example Excel-Sheet





2022-01-21

FOSDEM'22

Testing Requirements Documents

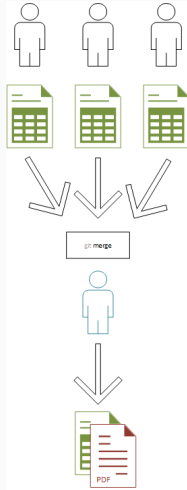
└─ sltoo in Practice

└─ Excel Workflow (II) – Distribution

Excel Workflow (II) – Distribution







2022-01-21

FOSDEM'22

Testing Requirements Documents

└─ sltoo in Practice

└─ Excel Workflow (III) – Merging

- Commit for every author individually -> also put name in commit for easier lookup
- Truth stays in repository
- `git tag` makes new document releases as easy as possible



Every document has a its own version tag

```
$ git tag -a RS/1A
```

```
$ git describe $(git log -n 1 --format=%H -- docs/reqs)
```

The output from `git describe` will be used as document baseline

```
RS/1A - 0aec3ad0 # good
RS/1A-8-g76b3ffe - 76b3ffe4 # tainted
```

Example excerpt from page 7:

---

Version SWC-RS/1A-8-g76b3ffe — 76b3ffe4

7

2022-01-21

FOSDEM'22

Testing Requirements Documents

└─ sltoo in Practice

└─ Document Baseline

Exercise to reject tainted documents in referenced documents

Document Baseline

```
Every document has a its own version tag
$ git tag -a RS/1A
$ git describe $(git log -n 1 --format=%H -- docs/reqs)
The output from git describe will be used as document baseline
 RS/1A - 0aec3ad0 # good
 RS/1A-8-g76b3ffe - 76b3ffe4 # tainted
Example excerpt from page 7:

Version SWC-RS/1A-8-g76b3ffe — 76b3ffe4
7
```

- Merge changes with *rmtoo*
- Transform *sltoo* into a GUI
  - Provide a GUI to show relations between items
  - Provide cross-documents editing capabilities
  - Update hashes in sources
- How much can AI help?

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ sltoo in Practice

└ Outlook

- Replace your current ALM tool with a diverse set of existing/open tools

Outlook

- Merge changes with *rmtoo*
- Transform *sltoo* into a GUI
  - Provide a GUI to show relations between items
  - Provide cross-documents editing capabilities
  - Update hashes in sources
- How much can AI help?

2022-01-21

FOSDEM'22  
Testing Requirements Documents  
└─ Demonstration

Demonstration

# Demonstration

---

2022-01-21

FOSDEM'22  
Testing Requirements Documents  
└ Conclusion

Conclusion

## Conclusion

- Requirements shipped with code ✓
  - Including relational meta-information
  - Use commit message to document the *why*
- Traceability matrix automated ✓
- Continuously updated documentation ✓
  - Document Versioning (baselining) ✓

A familiar UI for all stakeholders included

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Conclusion

└ Storytime Revisited

Your Jira workspace will be gone

- Requirements shipped with code ✓
  - Including relational meta-information
  - Use commit message to document the *why*
- Traceability matrix automated ✓
- Continuously updated documentation ✓
  - Document Versioning (baselining) ✓

A familiar UI for all stakeholders included

2022-01-21

FOSDEM'22  
Testing Requirements Documents  
└─ Questions

Questions

---

# Questions

---

Appendix



An introduction presentation into *rmtoo* and with more details.

- Traceability for the given requirements
- Bring code and documentation into same repository
- Integrate into build-system
  - Detect upstream changes to requirements
  - Quickly identify affected code-regions
- No silver bullet for verification

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Questions

└ Traceability Rationale

Traceability Rationale

- Traceability for the given requirements
- Bring code and documentation into same repository
- Integrate into build-system
  - Detect upstream changes to requirements
  - Quickly identify affected code-regions
- No silver bullet for verification

The status *external* will yield the following results:

- *open*
  - No matching requirement ID
- *passed*
  - Matching requirement ID
  - All hashes match
  - Unit-tests passed
- *failed*
  - Matching requirement ID
  - Some/all hashes didn't match, or
  - Unit-tests haven't passed

2022-01-21

FOSDEM'22  
Testing Requirements Documents  
└ Questions

└ Results

Results

The status *external* will yield the following results:

- *open*
  - No matching requirement ID
- *passed*
  - Matching requirement ID
  - All hashes match
  - Unit-tests passed
- *failed*
  - Matching requirement ID
  - Some/all hashes didn't match, or
  - Unit-tests haven't passed

Traceability features are in the beta releases.

```
$ pip3 install sltoo>=25.1.0b3
```

```
$ wget https://kown7.github.io/pymergevcd/assets/template_1
```

2022-01-21

FOSDEM'22

Testing Requirements Documents

└─ Questions

└─ Installation

Installation

Traceability features are in the beta releases.

```
$ pip3 install sltoo>=25.1.0b3
```

```
$ wget https://kown7.github.io/pymergevcd/assets/template_1
```

- Sphinx-Needs
- Octane ALM
- Codebeamer
- Aligned elements
- See Wikipedia
- ...

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Questions

└ Alternatives

Alternatives

- Sphinx-Needs
- Octane ALM
- Codebeamer
- Aligned elements
- See Wikipedia
- ...

- Write Parser for *Test Reports* ✓
- Documents with the correct identifier automatically solve the specification
  - Document Formats: \* docx (maybe with pandoc) \*  $\text{\LaTeX}$  ✓ \* Text \* CAD Files from HW/Mechanical
- GUI with multi-documents support (RS/TS/..)
  - Simplify design process

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Questions

└ Future Developements

- Write Parser for *Test Reports* ✓
- Documents with the correct identifier automatically solve the specification
  - Document Formats: \* docx (maybe with pandoc) \*  $\text{\LaTeX}$  ✓ \* Text \* CAD Files from HW/Mechanical
- GUI with multi-documents support (RS/TS/..)
  - Simplify design process

- Never test against your requirements
- Always write some form of test specification

2022-01-21

FOSDEM'22

Testing Requirements Documents

└ Questions

└ Final Thoughts

Final Thoughts

- Never test against your requirements
- Always write some form of test specification

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.

