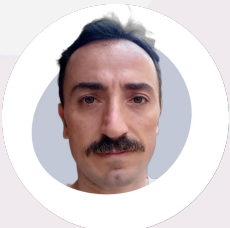


BASIL an open source tool that supports requirements traceability with design SBOM

Luigi Pellecchia
Principal Software Quality Engineer - Red Hat

Who I am



Luigi Pellecchia
Principal
Software Quality Engineer
Quality Engineering
In-vehicle OS
Red Hat



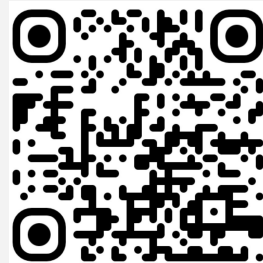
Agenda

- What is BASIL
- Traceability in a SDLC: V-Model
- BASIL applied to the V-Model
- BASIL SBOM with SPDX Model 3
- BASIL Test Infrastructure and Test Results traceability

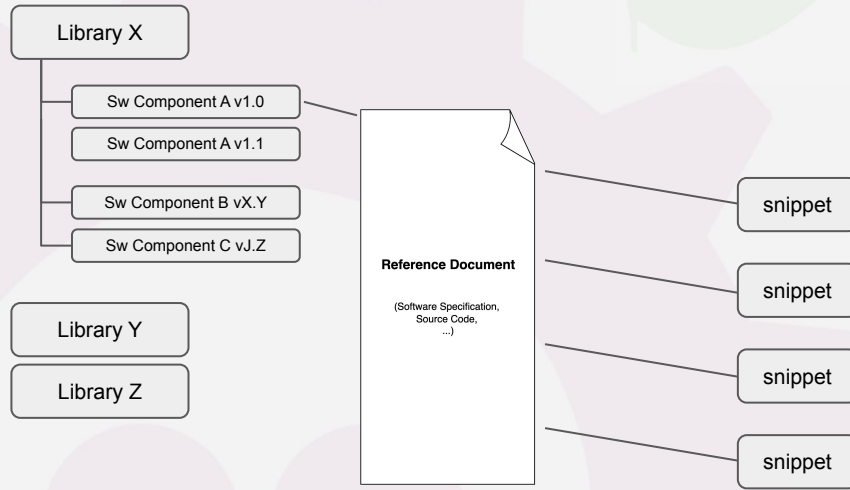
BASIL The FuSa Spice

Tool developed to manage software related work items, design their traceability towards specifications and ensure completeness of analysis

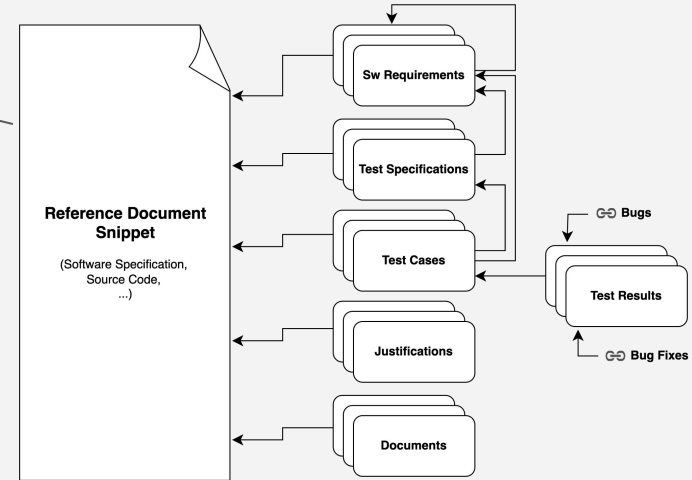
- Born at Red Hat to support RHIVOS Functional Safety ISO 26262 Compliance Certification
- BASIL name comes from ASIL B
- Presented to ELISA Project on June 2023 during the [Berlin Workshop](#)
- Open Sourced and hosted at [ELISA github](#)



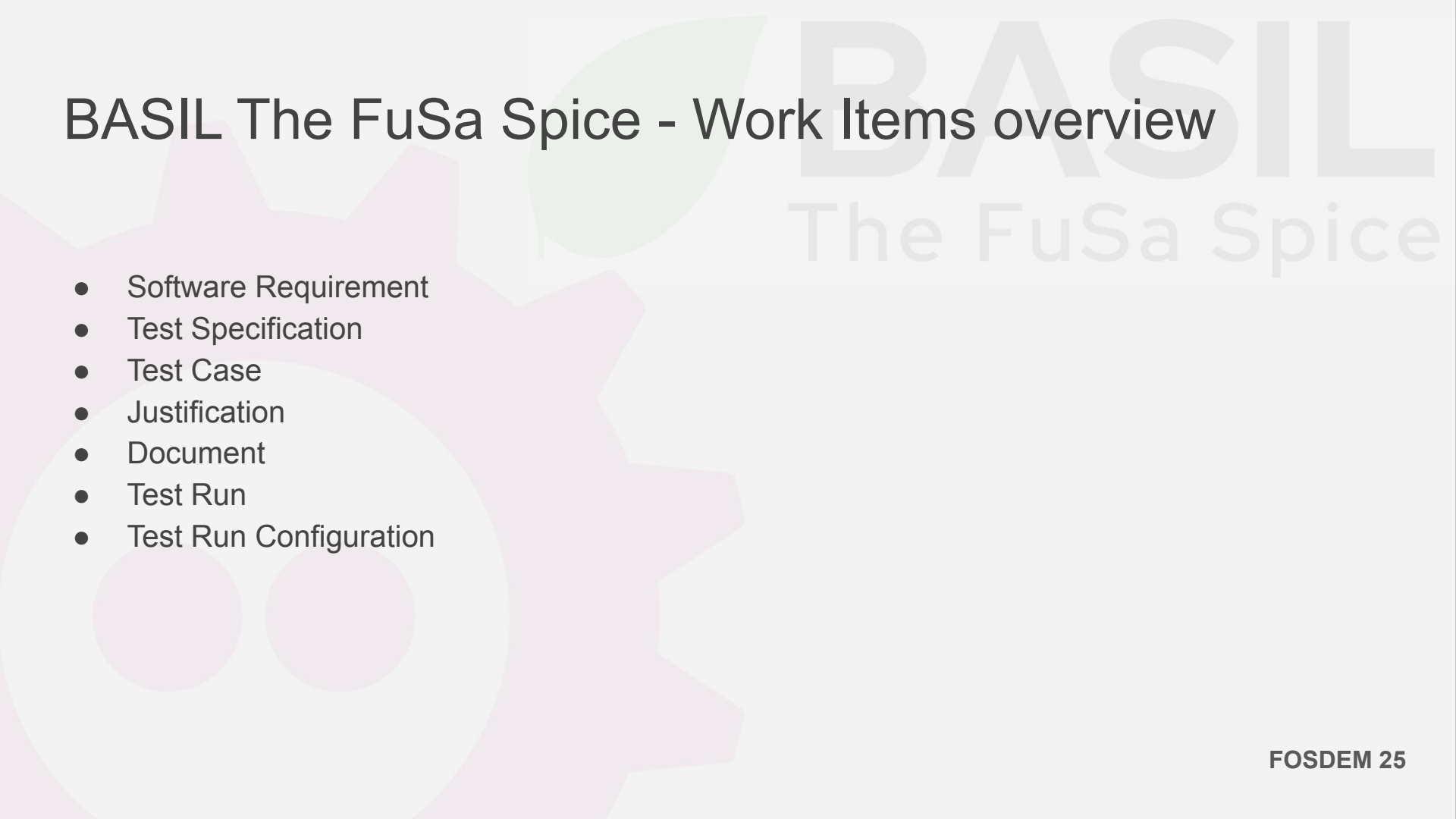
BASIL The FuSa Spice



Define the traceability matrix by creating the work items



BASIL The FuSa Spice - Work Items overview



- Software Requirement
- Test Specification
- Test Case
- Justification
- Document
- Test Run
- Test Run Configuration

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- Test Specification
- Test Case
- Justification
- Document
- Test Run
- Test Run Configuration

Key points:

- Hierarchical mapping

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- **Test Specification**
- Test Case
- Justification
- Document
- Test Run
- Test Run Configuration

Key points:

- Describes how to test a software functionality.
The preconditions, the maneuver that a tester should perform and the expected behavior.

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- Test Specification
- **Test Case**
- Justification
- Document
- Test Run
- Test Run Configuration

Key points:

- It is the test implementation
- Can link to a remote file in a git repo or to a local file in the machine running the BASIL instance

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- Test Specification
- Test Case
- **Justification**
- Document
- Test Run
- Test Run Configuration

Key points:

- Completeness of analysis

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- Test Specification
- Test Case
- Justification
- Document
- Test Run
- Test Run Configuration

Key points:

- Types: File, Text
- Text document supports snippet definition and automatic validation
- SPDX Model 3 based Relationship Type to the Reference Document
- Can be used to trace the source code to the specification

Next Steps:

- Hierarchical mapping
(e.g. Ref Doc ← AI Model ← Training Dataset)

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- Test Specification
- Test Case
- Justification
- Document
- **Test Run**
- Test Run Configuration

Key points:

- It is related to a Test Case Mapping as we can reuse a test case multiple times inside a Software Component (configurable test cases: e.g. ltp syscall test in BASIL examples)
- We can use the same Test Case with different Test Run Configuration (environment variables, SUT, ...)
- Link to bugs, fixes, artifacts
- Can refer test runs executed on external test infrastructures

BASIL The FuSa Spice - Work Items overview

- Software Requirement
- Test Specification
- Test Case
- Justification
- Document
- Test Run
- Test Run Configuration

Key points:

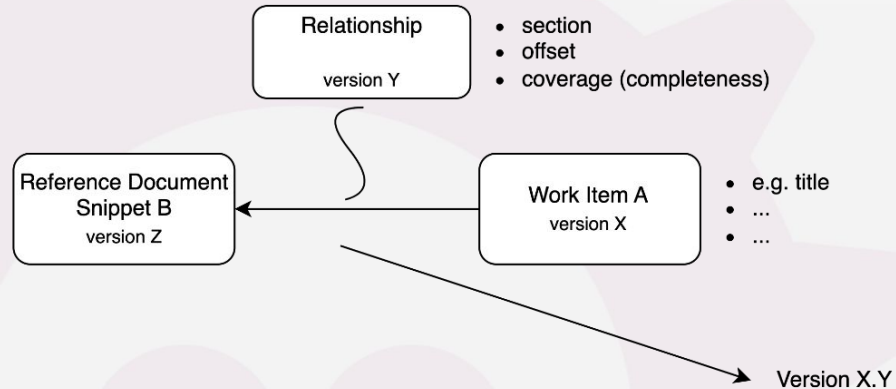
- Can change the test behavior and the SUT
- Can leverage external test infrastructures
- Reusable
- Can use preset configuration defined by the BASIL Admin in a yaml file

BASIL The FuSa Spice - Work Items Mapping

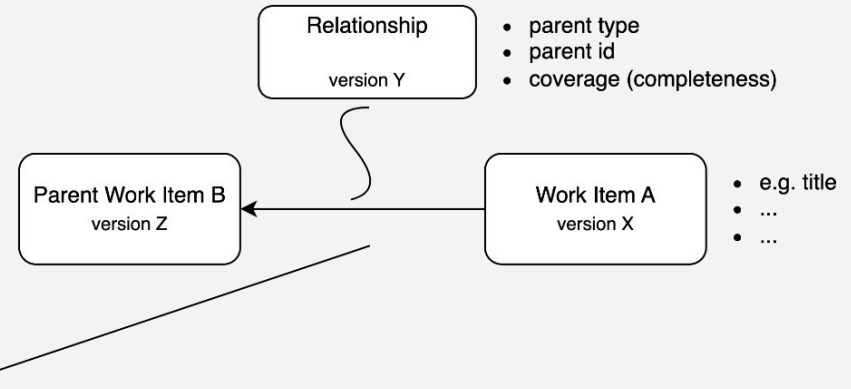
- Direct Mapping
 - Multiple views based on work item type
 - Link to a snippet of the reference document using section and offset
 - Completeness percentage (basil coverage)
- Indirect Mapping
 - Waterfall propagation of completeness percentage
- Broken Mapping
 - Changes of the Reference document can lead in broken mapping
 - Broken mapping are displayed in a dedicated section
 - Can be automatically fixed by the tool
 - Prediction of broken mapping analyzing a different version of the reference document

BASIL - Work Items Version Control

DIRECT MAPPING



INDIRECT MAPPING



BASIL The FuSa Spice - key points

- Web App with user management
- Clarifies the gaps
- Support collaboration through comments, notifications and work item workflow
- Multiple mapping views to parallelize teams work
- Follow the project evolution
- Allow integration in CI and automated workflows via REST API
- Simplified deployment via containers

Traceability in Safety Critical Application

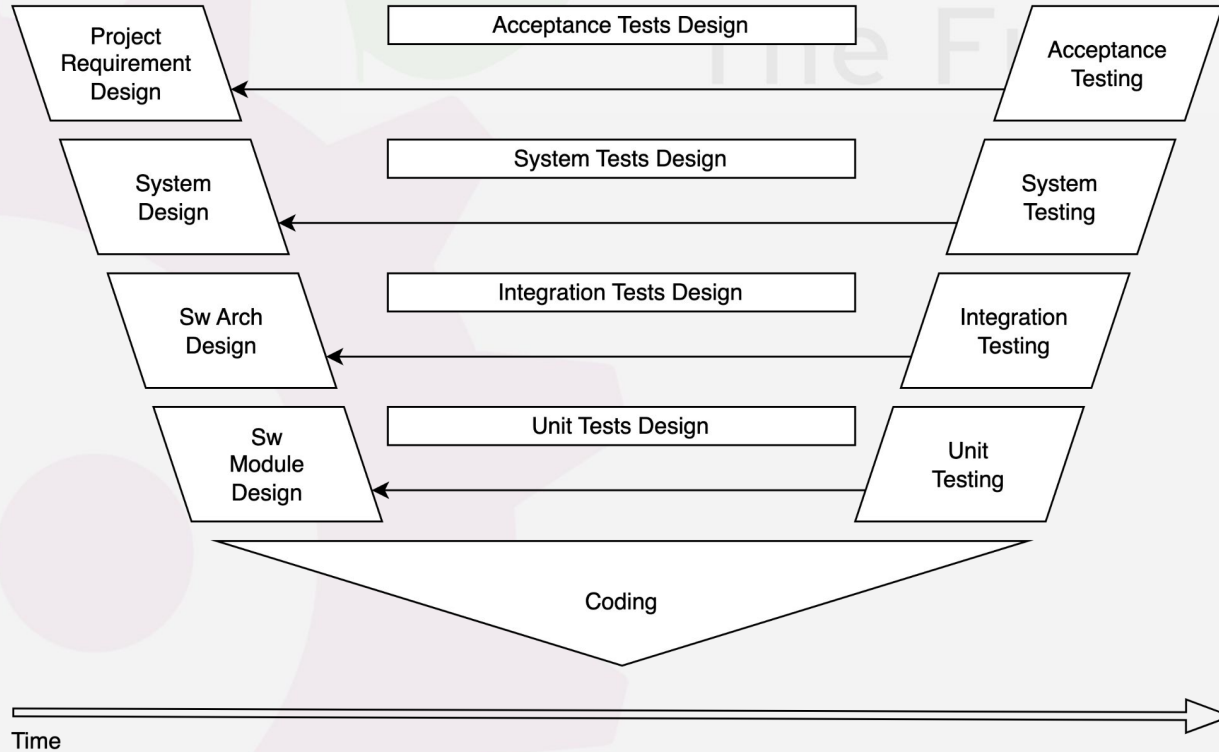
A **safety-critical system** or **life-critical system** is a system whose failure or malfunction may result in one (or more) of the following outcomes:

- death or serious injury to people
- loss or severe damage to equipment/property
- environmental harm

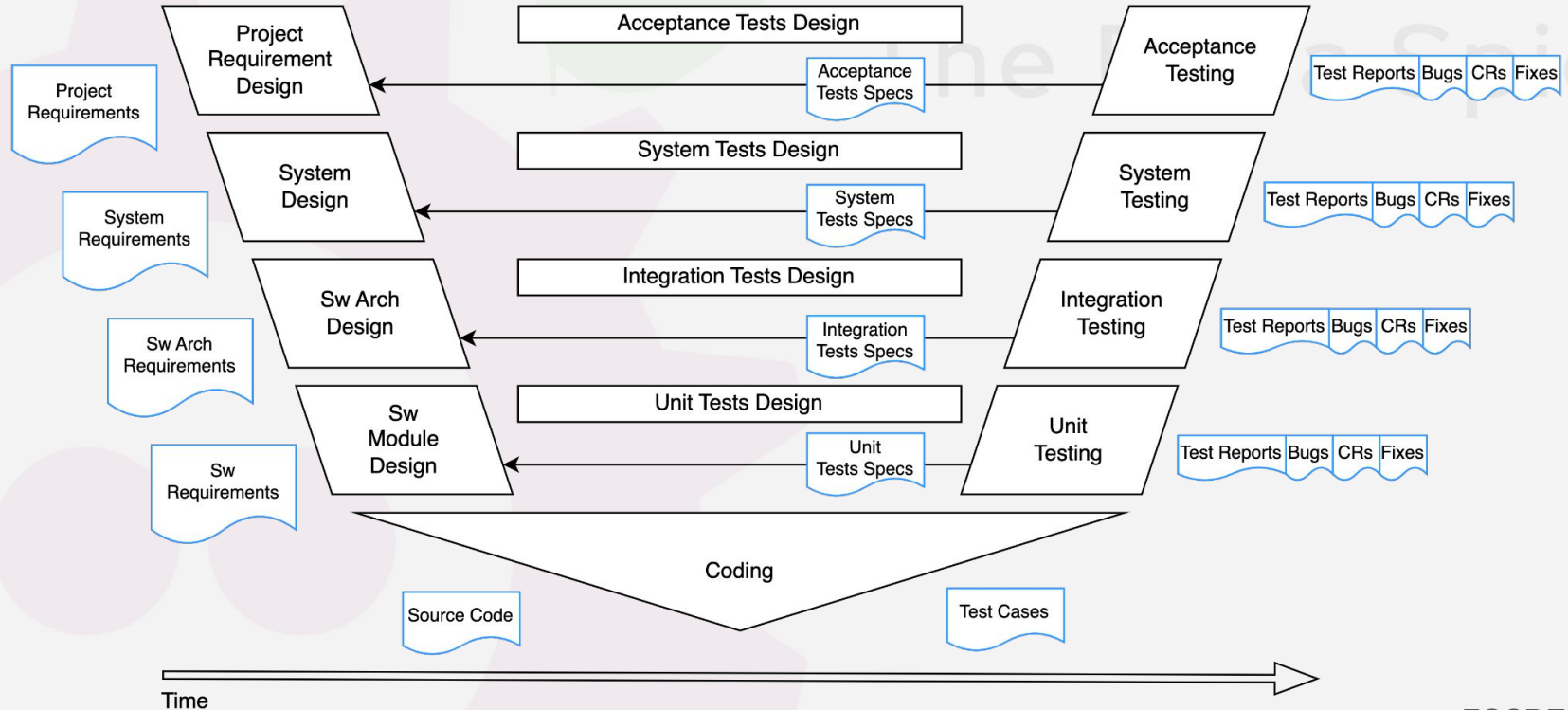
[https://en.wikipedia.org/wiki/Safety-critical_system]

- Required by international standards (ISO 26262, 15504, DO-178C...)
- Establishes and demonstrates control over the process
- Simplifies impact analysis
- Highlights gaps (helps estimate effort)

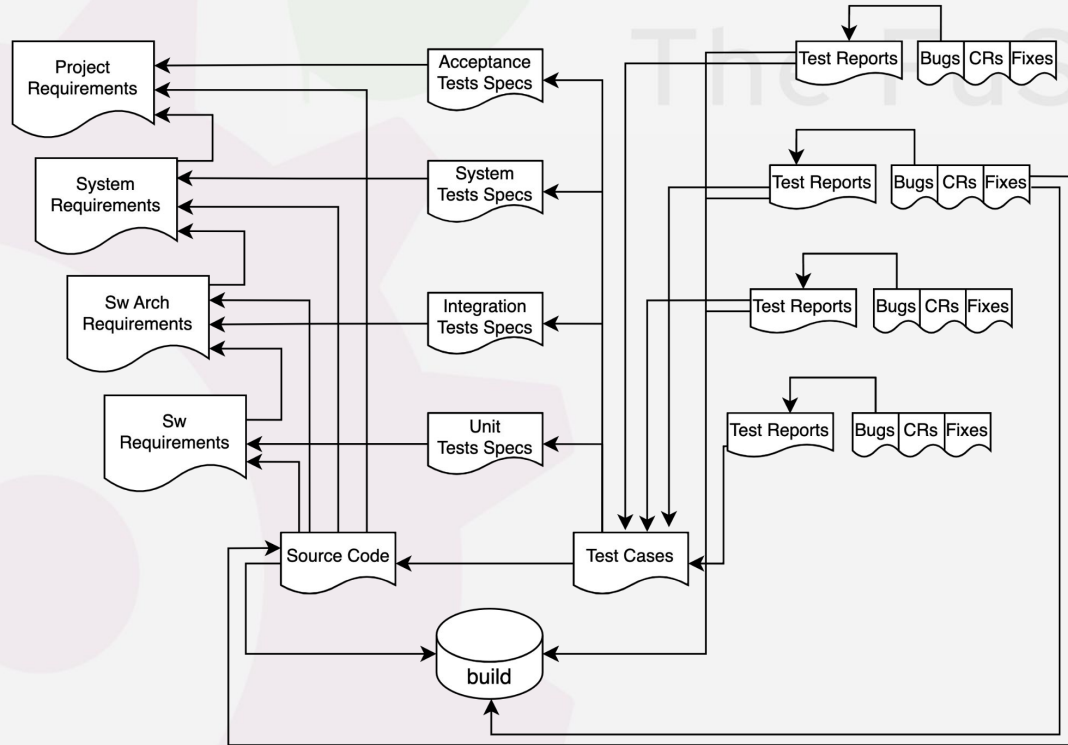
SDLC V-Model



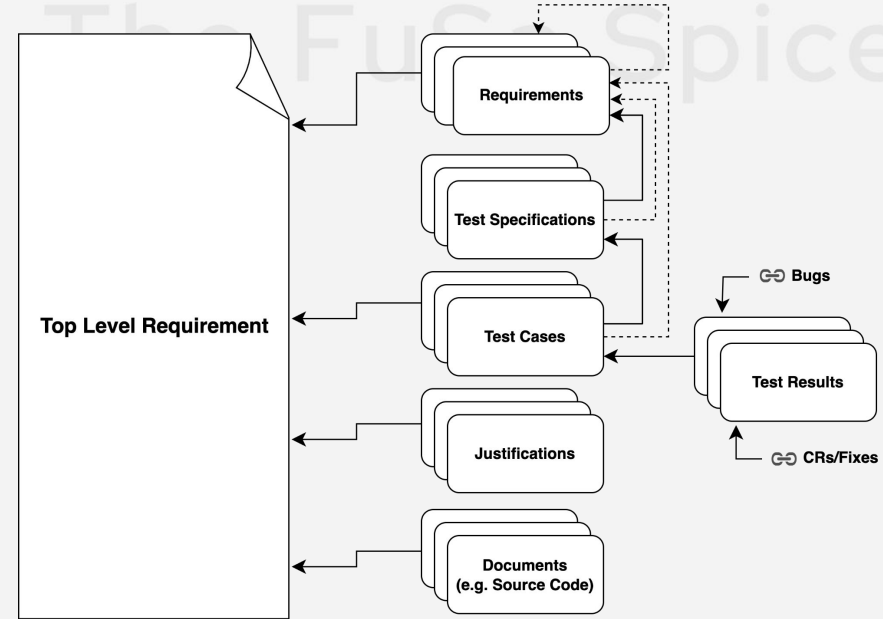
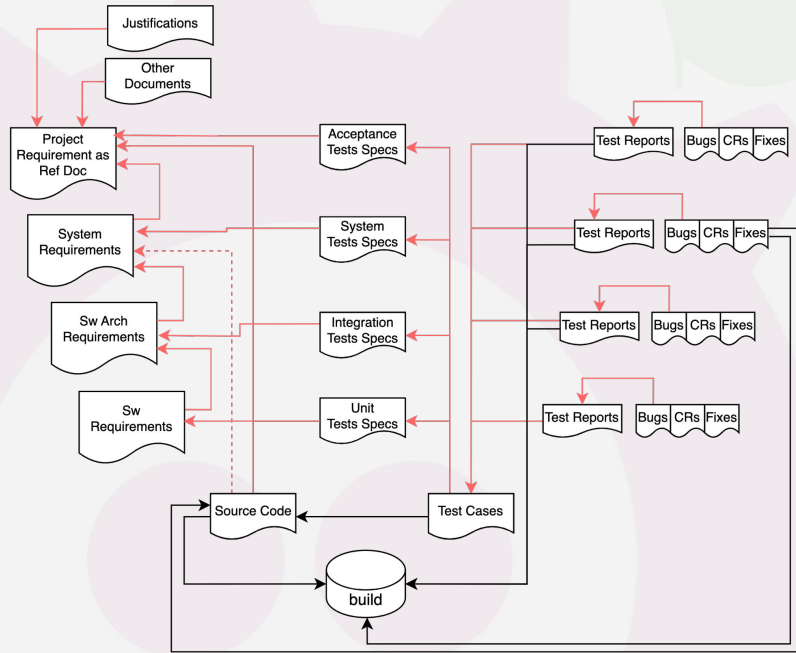
SDLC V-Model Artifacts



V-Model - Traceability

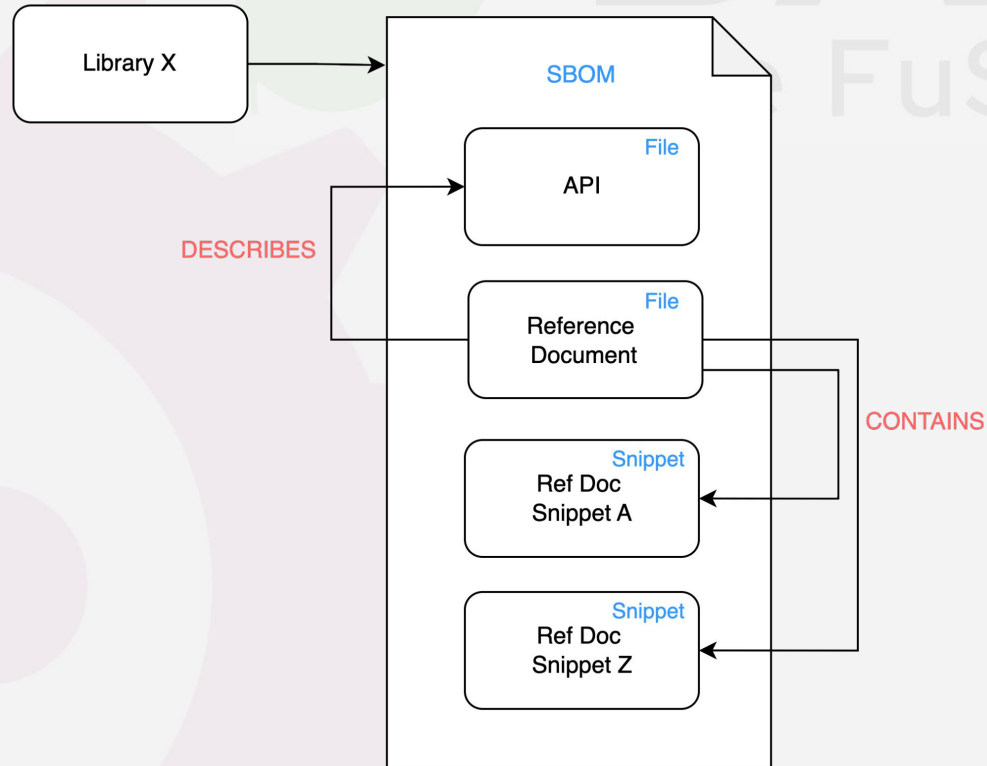


V-Model - System Requirements - BASIL example

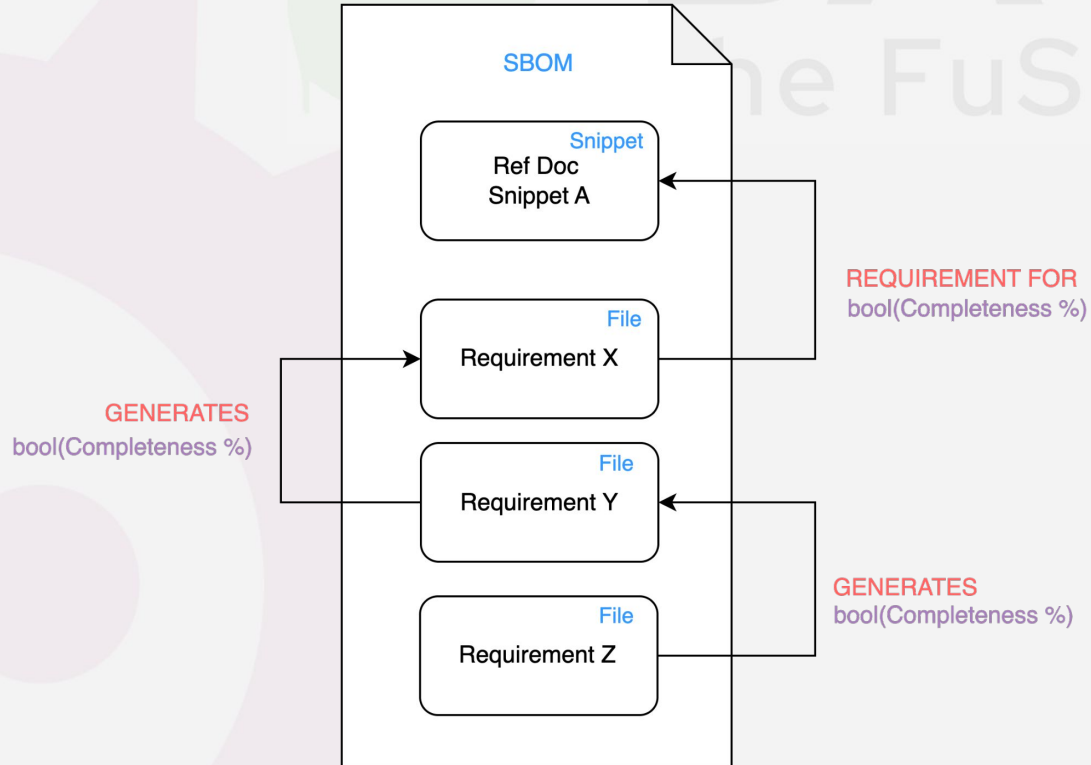


NOTE: the same traceability can be established focusing on other types of requirements

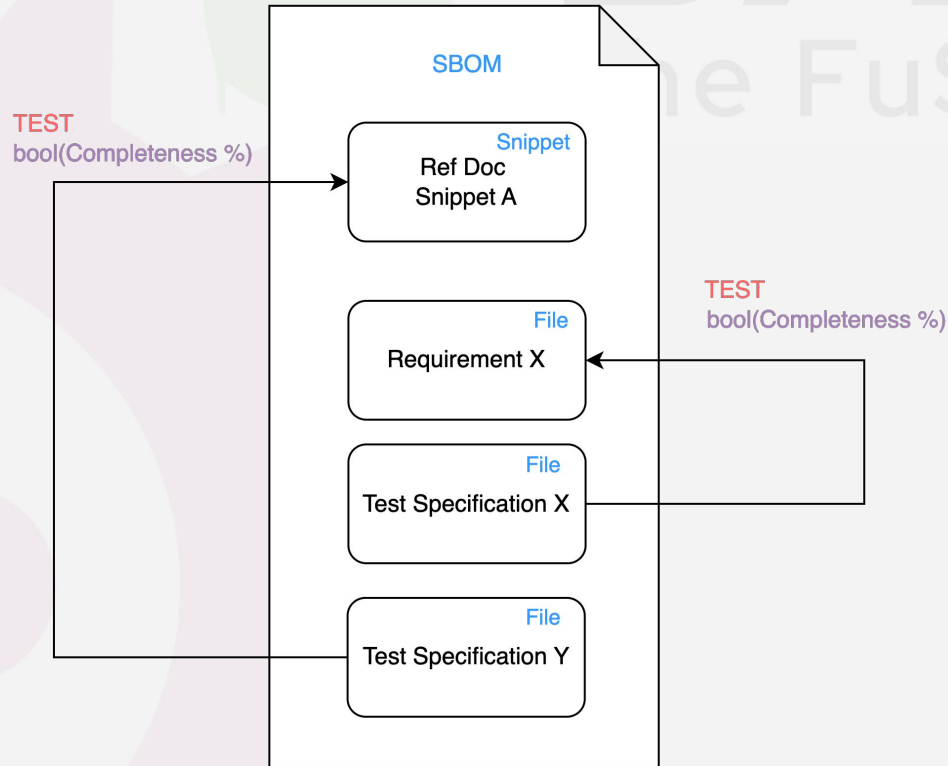
BASIL - Reference Document Snippets



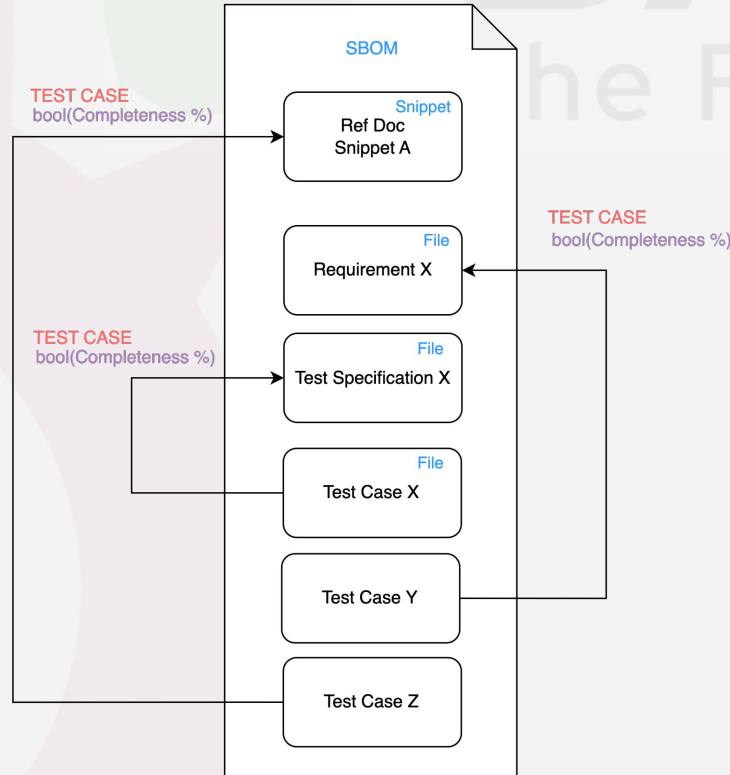
BASIL - Sw Requirements



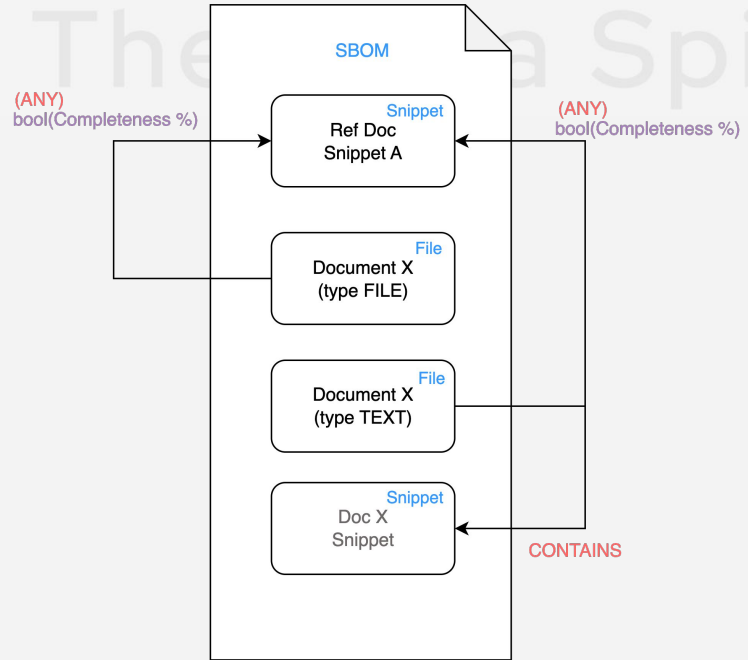
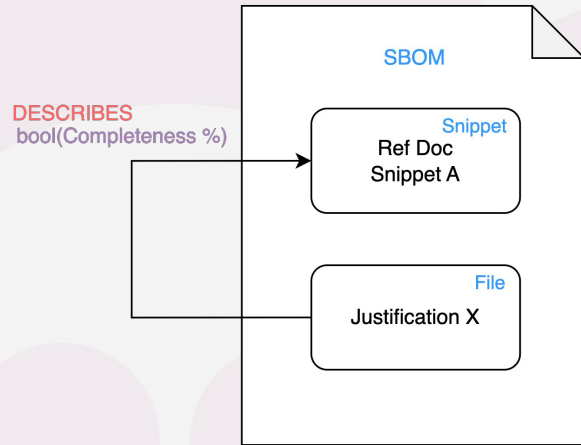
BASIL - Test Specifications



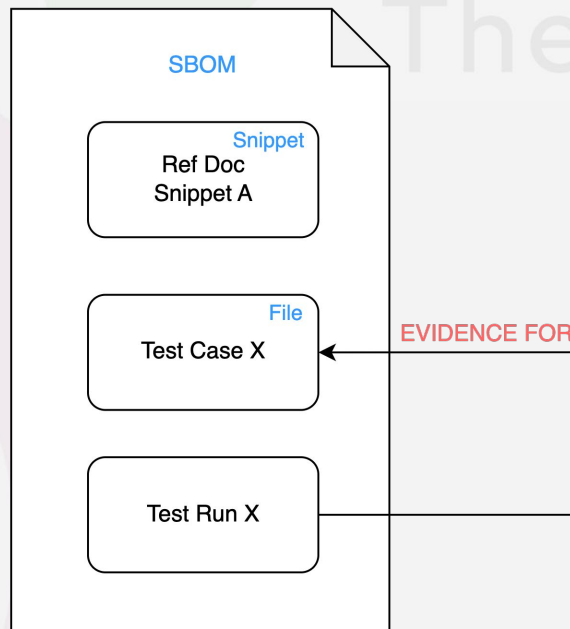
BASIL - Test Cases



BASIL - Justifications and Documents



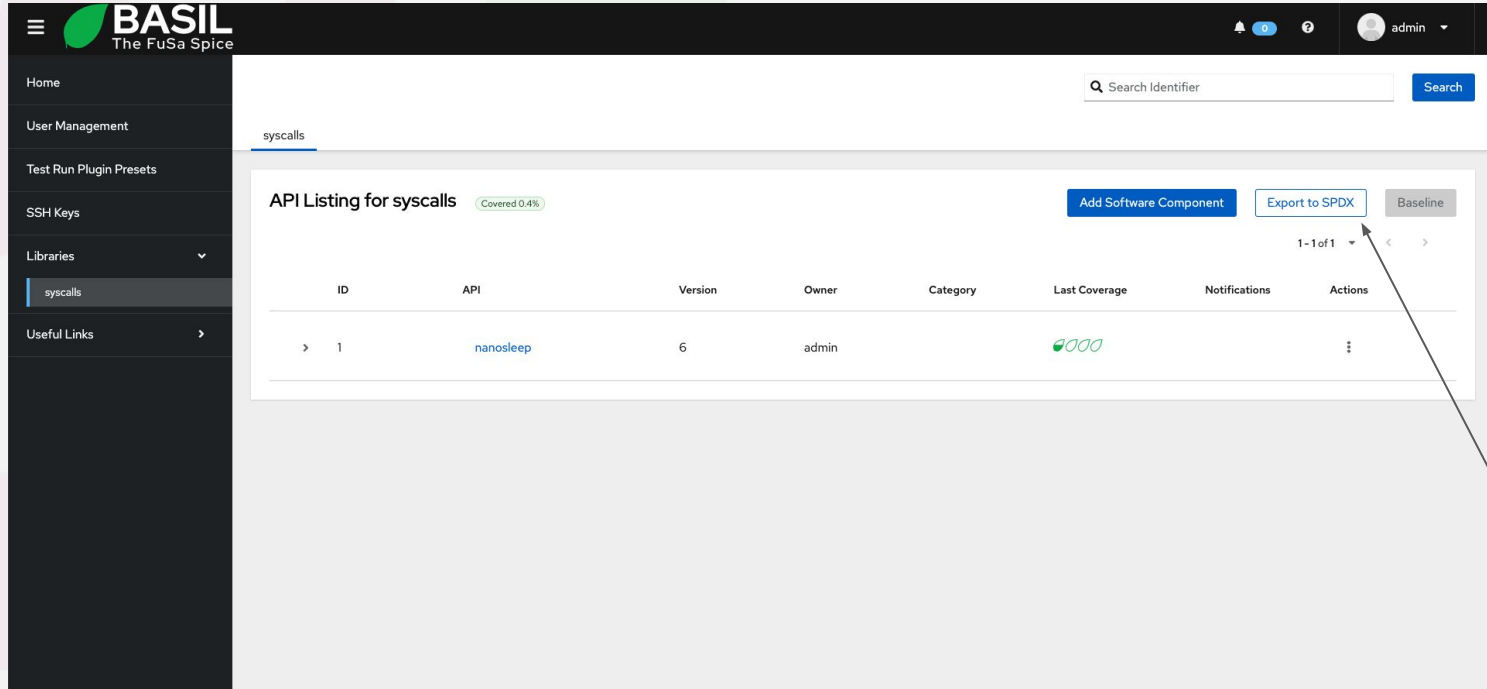
BASIL - Test Results



BASIL - Export work items data in SPDX

- Which Data
 - Stringified python dictionary of the work item
- Exported to
 - Attribution text
- Used also to
 - Calculate the hash used to populate verified_using
- Pros
 - All data collected the same way/place
 - Can iterate through the keys of the dictionary to know the data struct

BASIL - Export library



The screenshot shows the BASIL web interface. The sidebar on the left contains the following menu items: Home, User Management, Test Run Plugin Presets, SSH Keys, Libraries (expanded), syscalls (selected), and Useful Links. The main content area is titled 'syscalls' and displays 'API Listing for syscalls' with a 'Covered 0.4%' status. Above the table are three buttons: 'Add Software Component', 'Export to SPDX' (highlighted with an arrow), and 'Baseline'. The table has columns for ID, API, Version, Owner, Category, Last Coverage, Notifications, and Actions. It contains one row for the 'nanosleep' API.

ID	API	Version	Owner	Category	Last Coverage	Notifications	Actions
> 1	nanosleep	6	admin		0000		⋮

BASIL - Export library

The screenshot displays the BASIL web application interface. On the left is a dark sidebar with navigation links: Home, User Management, Test Run Plugin Presets, SSH Keys, Libraries (with a dropdown arrow), and Useful Links (with a right arrow). The 'Libraries' section is active, showing a list with 'syscalls' selected. The main content area shows the 'syscalls' library details, including an 'API L' link. A modal window titled 'SPDX Data' is open, displaying the 'Export of selected library work items and relationships'. The modal contains a JSON-LD context for SPDX, listing various namespaces and their URIs, such as 'ai', 'build', 'core', 'dataset', 'licensing', 'ns0', 'owl', 'rdfs', 'security', 'sh', 'software', 'xsd', 'AIPackage', 'Build', 'Annotation', 'AnonymousPayload', 'Organization', 'Person', 'SoftwareAgent', 'SpdxDocument', 'Dataset', 'ConjunctiveLicenseSet', 'CustomLicense', 'CustomLicenseAddition', 'DisjunctiveLicenseSet', 'ListedLicense', 'ListedLicenseException', 'NoAssertionLicense', and 'NoneLicense'. A 'Close' button is at the bottom left of the modal. In the background, the 'Export to SPDX' button is visible on the library's action bar.

SPDX Data
Export of selected library work items and relationships

```
{
  "@context": {
    "ai": "https://spdx.org/rdf/AI/",
    "build": "https://spdx.org/rdf/Build/",
    "core": "https://spdx.org/rdf/Core/",
    "dataset": "https://spdx.org/rdf/Dataset/",
    "licensing": "https://spdx.org/rdf/Licensing/",
    "ns0": "http://www.w3.org/2003/06/sw-vocab-status/ns#",
    "owl": "http://www.w3.org/2002/07/owl#",
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
    "security": "https://spdx.org/rdf/Security/",
    "sh": "http://www.w3.org/ns/shacl#",
    "software": "https://spdx.org/rdf/Software/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "AIPackage": "ai:AIPackage",
    "Build": "build:Build",
    "Annotation": "core:Annotation",
    "AnonymousPayload": "core:AnonymousPayload",
    "Organization": "core:Organization",
    "Person": "core:Person",
    "SoftwareAgent": "core:SoftwareAgent",
    "SpdxDocument": "core:SpdxDocument",
    "Dataset": "dataset:Dataset",
    "ConjunctiveLicenseSet": "licensing:ConjunctiveLicenseSet",
    "CustomLicense": "licensing:CustomLicense",
    "CustomLicenseAddition": "licensing:CustomLicenseAddition",
    "DisjunctiveLicenseSet": "licensing:DisjunctiveLicenseSet",
    "ListedLicense": "licensing>ListedLicense",
    "ListedLicenseException": "licensing>ListedLicenseException",
    "NoAssertionLicense": "licensing:NoAssertionLicense",
    "NoneLicense": "licensing:NoneLicense",
```

Close

BASIL - Import sw requirements data in SPDX

- Filtering the @graph
 - @type == File
 - summary == 'Software Requirement'
- Import work item data from
 - Attribution text
- User can select which one to import (backend is using spdx_id)

BASIL - Import Software Requirements

The screenshot displays the BASIL web application interface. A modal dialog titled 'Software Requirement' is open, showing the 'Import' tab. The dialog contains a text input field with 'example.jsonld', an 'Upload' button, and a 'Clear' button. Below this is a table with columns 'ID', 'Title', and 'Description'. The table contains one row with a checkbox, 'SW-REQUIREMENT_1', 'Dummy Requirement SWREQ-XXX-YYY', and a placeholder description. At the bottom of the dialog are 'Submit', 'Reset', and 'Cancel' buttons. The background shows the BASIL dashboard with a sidebar menu and a main content area.

BASIL
The FuSa Spice

Home
User Management
Test Run Plugin Presets
SSH Keys
Libraries
Useful Links

library one > 1 > aaa

Mapping

Sw Requirements

SPECIFICATION

Coverage Total: 50.0%

Coverage Total: 50.0%

BASIL

A tool developed to manage software related work items, design their traceability towards specifications and source code and ensure completeness of analysis.

It comes with a web user interface to provide a simplified view of work item relationships and with a REST api to simplify the integration in other toolchains.

Map Test Case Map Justification Map Document

Indirect Test Specification Indirect Test Case

50.0% Coverage

0

Software Requirement

Work item data and mapping information (section, offset, coverage).

Sw Requirement Data Mapping Section Existing Import

example.jsonld Upload Clear

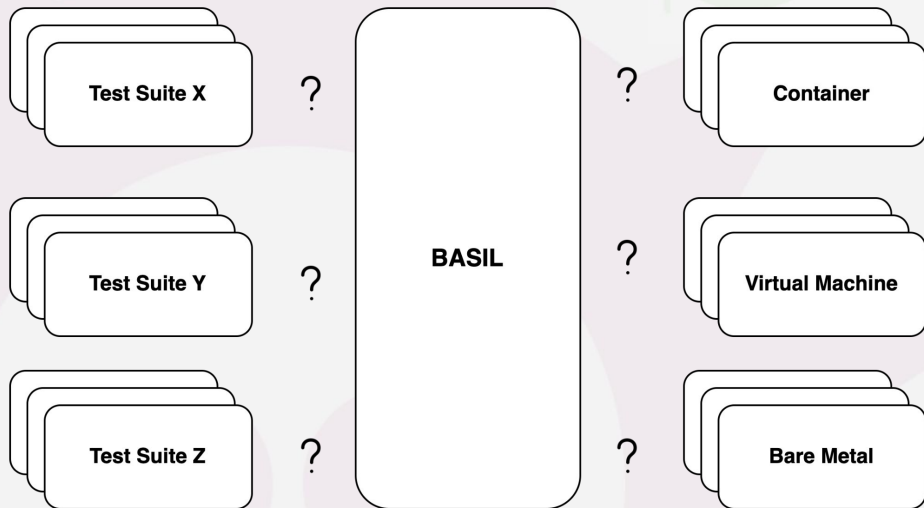
<input type="checkbox"/>	ID	Title	Description
<input type="checkbox"/>	SW-REQUIREMENT_1	Dummy Requirement SWREQ-XXX-YYY	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Data loaded

Submit Reset

Cancel

BASIL Embedded Test Infrastructure



tmt (Test Management Tool)


Python project that uses **fmf** metadata file (yaml) to abstract test case, test plans and user stories.

Can provision different test environments.

BASIL Plugin based Test Infrastructure

Test Infrastructure	Trigger and Trace	Trace pre existing runs	Test Infrastructure	BASIL Version
tmt	✓	✗	Embedded	>= 1.4
Gitlab CI	✓	✓	External	>= 1.5
Github Actions	✓	✓	External	>= 1.5
KernelCI	✗	✓	External	>= 1.5
Testing Farm	✓	✗	External	>= 1.5

BASIL - roadmap

- Import Software Requirements from BASIL export file [#82](#) - 
- Import Software Requirements from BASIL export file documentation [#86](#)
- Hierarchical Document Mapping [#81](#)
- Import Software Requirements from other tools [#83](#)
- Extend the traceability to
 - Test Run Configuration
 - Bugs, Fixes
 - Document (TEXT) Snippet

Questions?

Mine

- Which SBOM should collect Test Cases, Test Results, Bugs, MR/PR? (Runtime SBOM?)
- There is a standard on where to put custom work item data in spdx tool?

Luigi Pellecchia

Principal Software Quality Engineer - Red Hat



BASIL
The FuSa Spice

A large, faint pink gear shape serves as a background element on the left side of the slide.

Thanks

Luigi Pellecchia

Principal Software Quality Engineer - Red Hat

<https://github.com/elisa-tech/BASIL>

FOSDEM 25