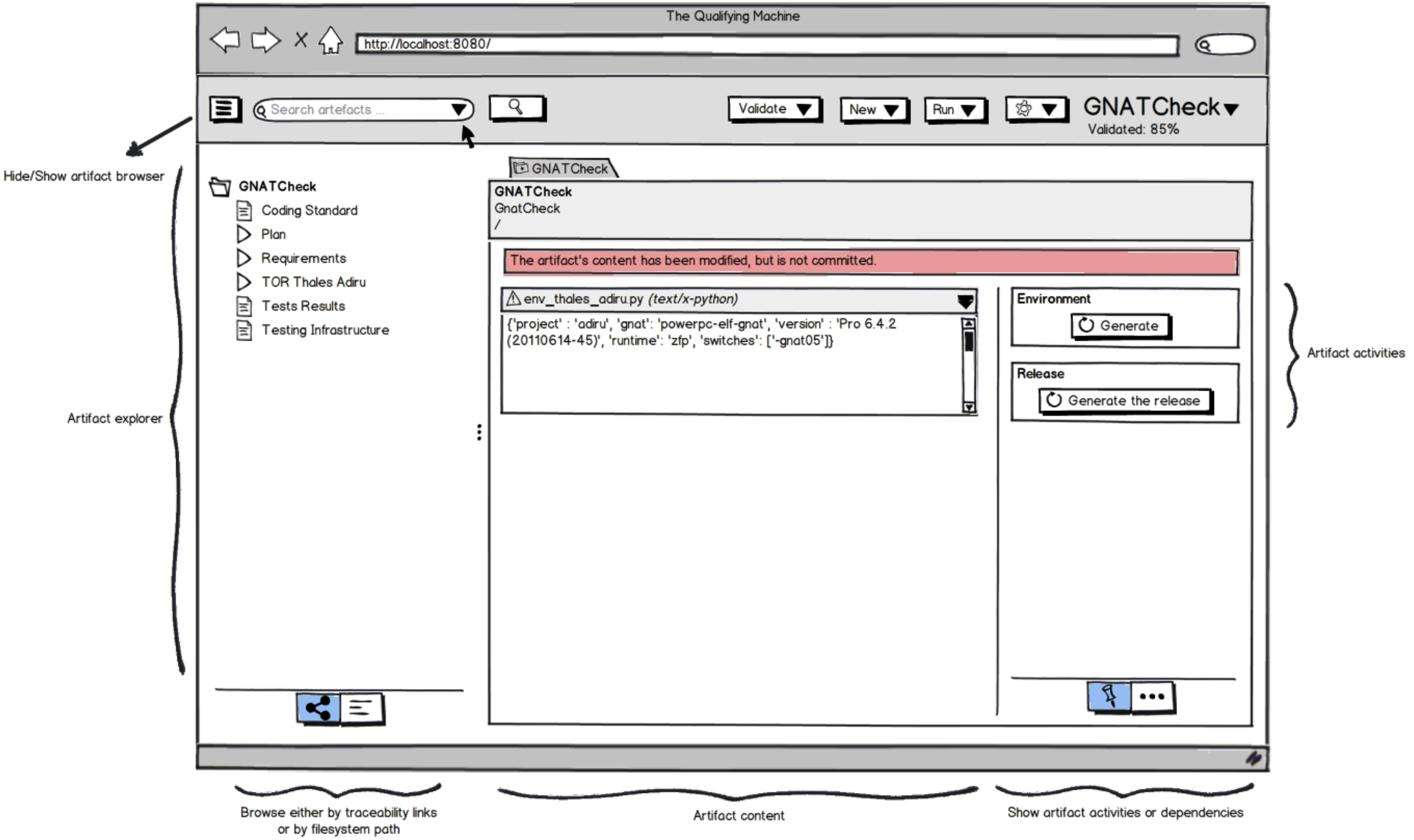
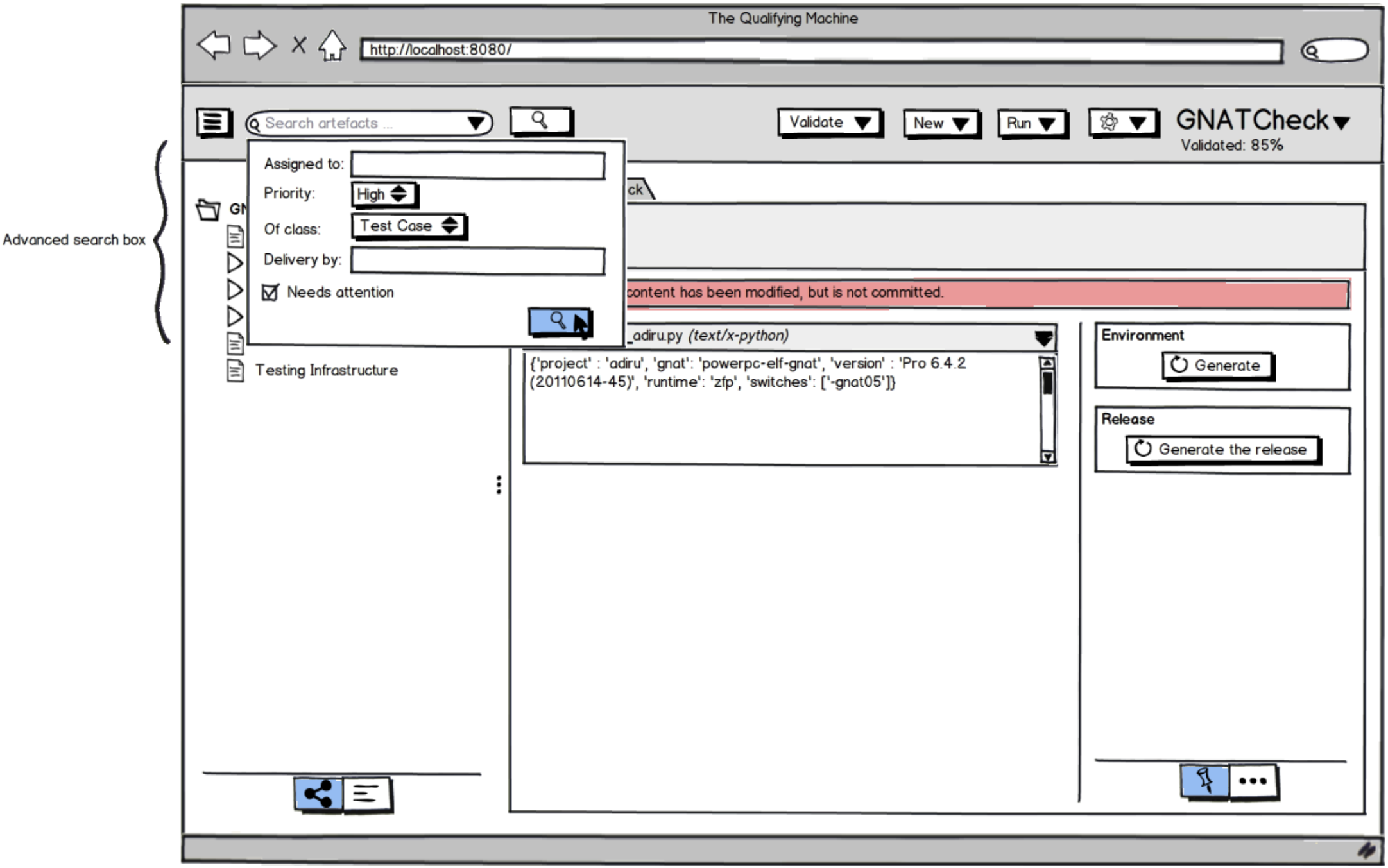
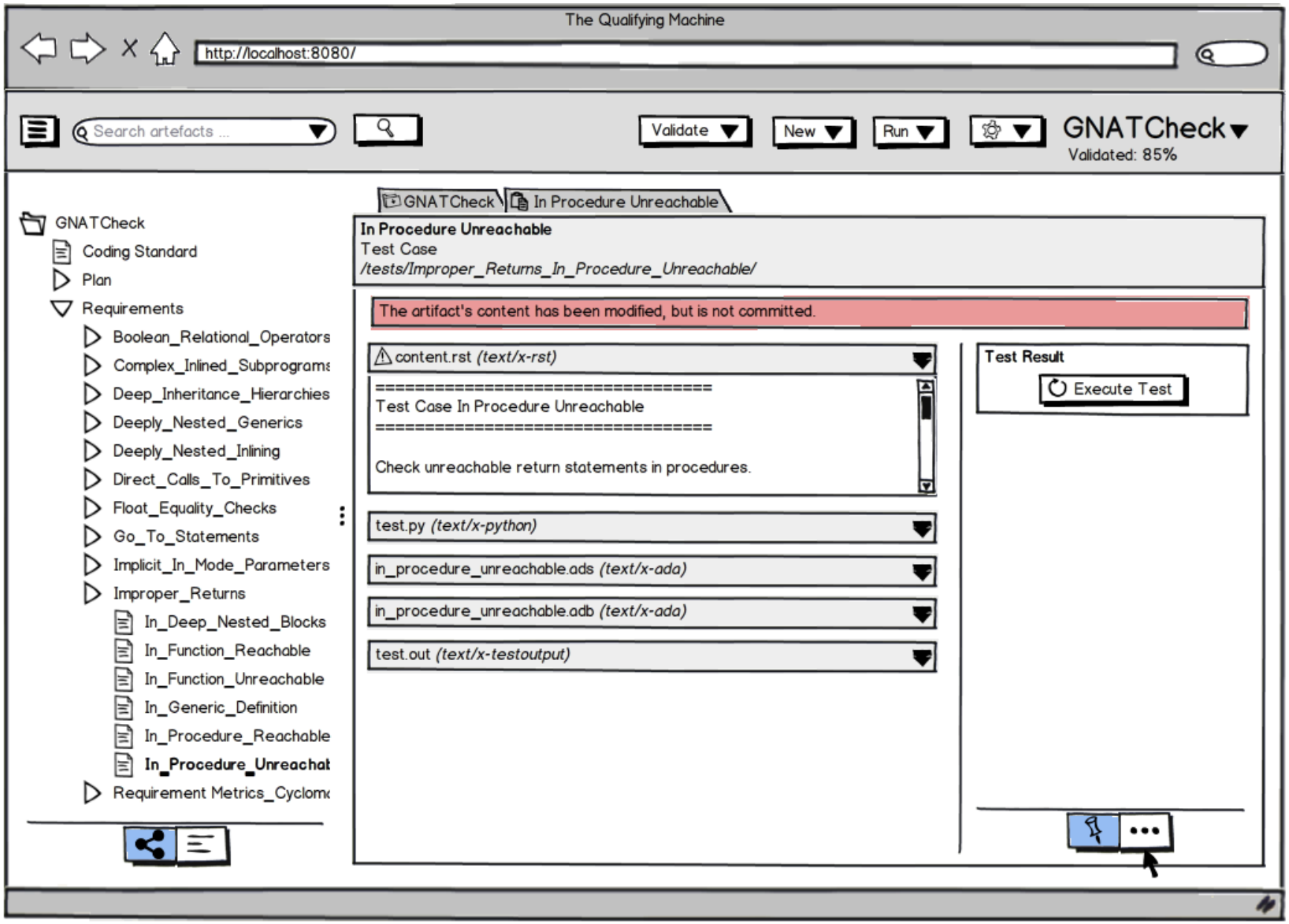

The Qualifying Machine
http://localhost:8080/
Search artefacts ...
Validate
New
Run
GNATCheck
Validated: 85%
Hide/Show artifact browser
GNATCheck
Coding Standard
Plan
Requirements
TOR Thales Adiru
Tests Results
Testing Infrastructure
Artifact explorer
GNATCheck
GNATCheck
GnatCheck
/
The artifact's content has been modified, but is not committed.
env_thales_adiru.py (text/x-python)
{'project' : 'adiru', 'gnat': 'powerpc-elf-gnat', 'version' : 'Pro 6.4.2 (20110614-45)', 'runtime': 'zfp', 'switches': ['-gnat05']}
Environment
Generate
Release
Generate the release
Artifact activities
Browse either by traceability links or by filesystem path
Artifact content
Show artifact activities or dependencies

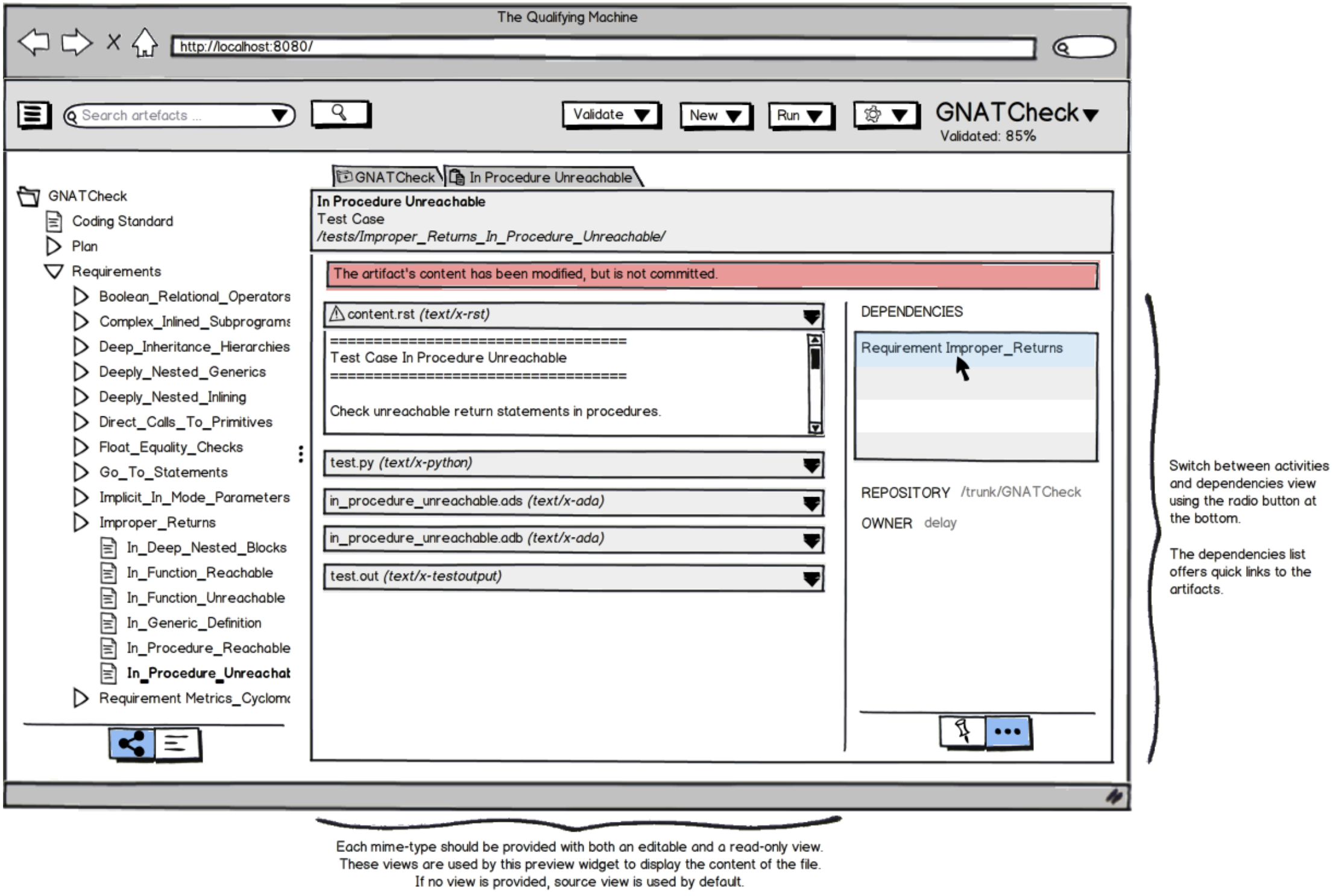

The Qualifying Machine
http://localhost:8080/
Search artefacts ...
Validate
New
Run
GNATCheck
Validated: 85%
Advanced search box
Assigned to:
Priority:
High
Of class:
Test Case
Delivery by:
Needs attention
content has been modified, but is not committed.
_adiru.py (text/x-python)
{'project' : 'adiru', 'gnat': 'powerpc-elf-gnat', 'version' : 'Pro 6.4.2 (20110614-45)', 'runtime': 'zfp', 'switches': ['-gnat05']}
Testing Infrastructure
Environment
Generate
Release
Generate the release



Opening an artifact through search reveal it in the browser

Opened artefact are registered in the MDI (Multiple Document Interface) to be easily re-opened

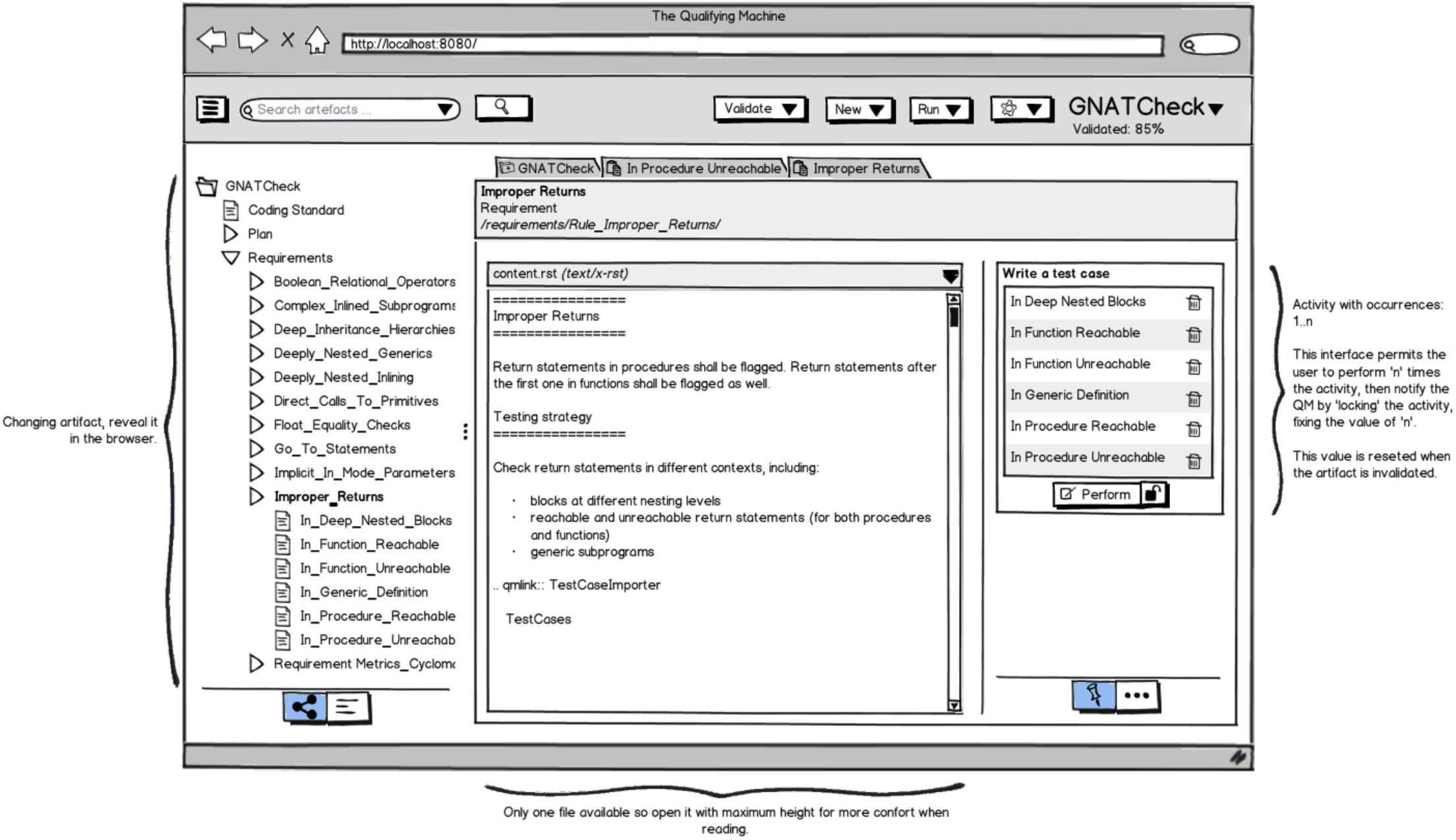
Artifact with multiple files as content.
The content.rst is opened by default.
Clicking on the filename roll in/out the file preview.
Preview uses the correct widget corresponding to the file mime-type.

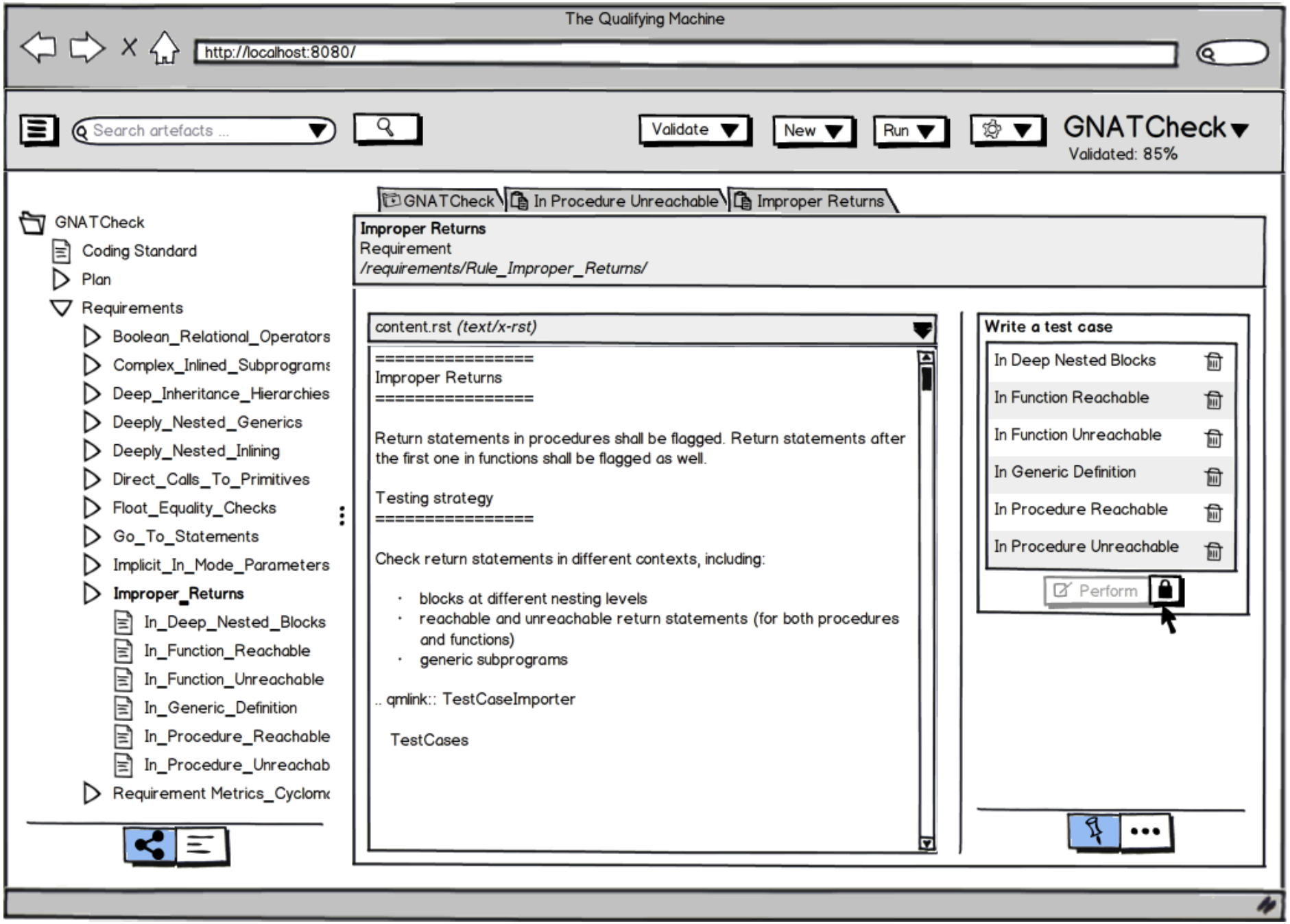


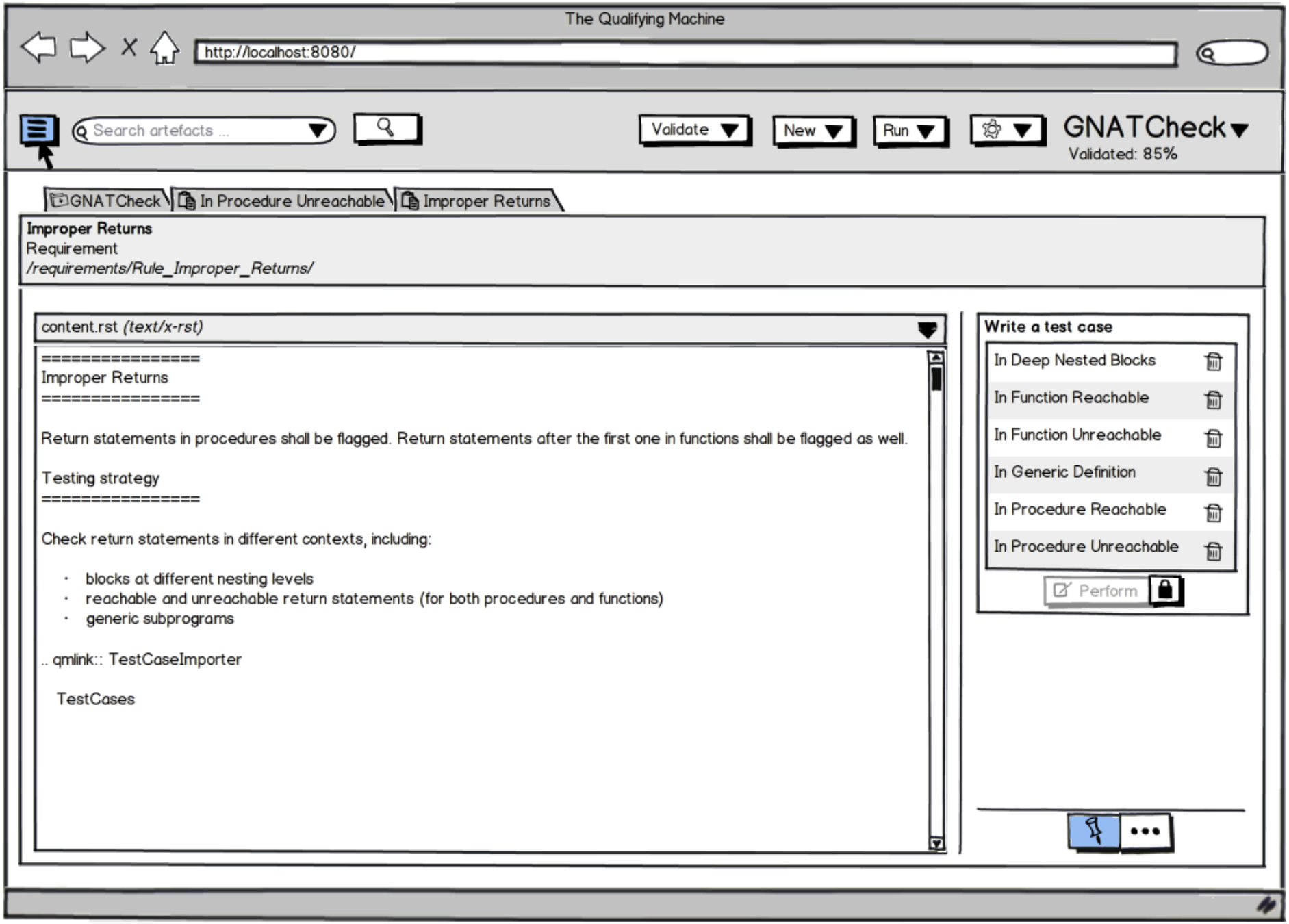
Switch between activities and dependencies view using the radio button at the bottom.

The dependencies list offers quick links to the artifacts.

Each mime-type should be provided with both an editable and a read-only view.
These views are used by this preview widget to display the content of the file.
If no view is provided, source view is used by default.


The Qualifying Machine
http://localhost:8080/
Search artefacts ...
Validate
New
Run
GNATCheck
Validated: 85%
GNATCheck
Coding Standard
Plan
Requirements
Boolean_Relational_Operators
Complex_Inlined_Subprograms
Deep_Inheritance_Hierarchies
Deeply_Nested_Generics
Deeply_Nested_Inlining
Direct_Calls_To_Primitives
Float_Equality_Checks
Go_To_Statements
Implicit_In_Mode_Parameters
Improper_Returns
In_Deep_Nested_Blocks
In_Function_Reachable
In_Function_Unreachable
In_Generic_Definition
In_Procedure_Reachable
In_Procedure_Unreachab
Requirement Metrics_Cyclom
GNATCheck
In Procedure Unreachable
Improper Returns
Improper Returns
Requirement
/requirements/Rule_Improper_Returns/
content.rst (text/x-rst)
==========
Improper Returns
==========
Return statements in procedures shall be flagged. Return statements after the first one in functions shall be flagged as well.
Testing strategy
==========
Check return statements in different contexts, including:
blocks at different nesting levels
reachable and unreachable return statements (for both procedures and functions)
generic subprograms
.. qmlink:: TestCaseImporter
TestCases
Write a test case
In Deep Nested Blocks
In Function Reachable
In Function Unreachable
In Generic Definition
In Procedure Reachable
In Procedure Unreachable
Perform
Changing artifact, reveal it in the browser.
Activity with occurrences: 1..n
This interface permits the user to perform 'n' times the activity, then notify the QM by 'locking' the activity, fixing the value of 'n'.
This value is reseted when the artifact is invalidated.
Only one file available so open it with maximum height for more confort when reading.

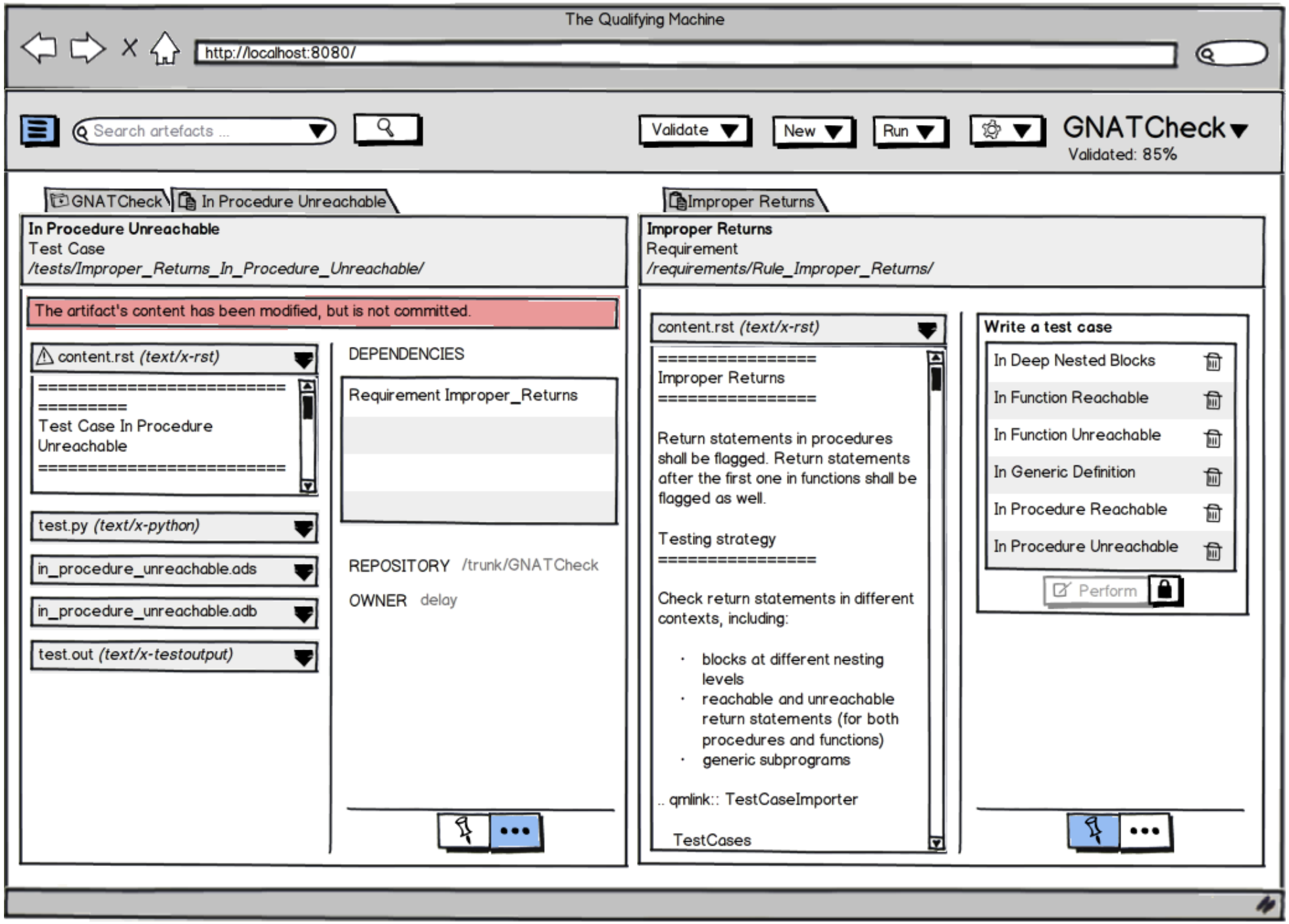

The Qualifying Machine
http://localhost:8080/
Search artefacts ...
Validate
New
Run
GNATCheck
Validated: 85%
GNATCheck
In Procedure Unreachable
Improper Returns
GNATCheck
Coding Standard
Plan
Requirements
Boolean_Relational_Operators
Complex_Inlined_Subprograms
Deep_Inheritance_Hierarchies
Deeply_Nested_Generics
Deeply_Nested_Inlining
Direct_Calls_To_Primitives
Float_Equality_Checks
Go_To_Statements
Implicit_In_Mode_Parameters
Improper_Returns
In_Deep_Nested_Blocks
In_Function_Reachable
In_Function_Unreachable
In_Generic_Definition
In_Procedure_Reachable
In_Procedure_Unreachab
Requirement Metrics_Cyclom
Improper Returns
Requirement
/requirements/Rule_Improper_Returns/
content.rst (text/x-rst)
==========
Improper Returns
==========
Return statements in procedures shall be flagged. Return statements after the first one in functions shall be flagged as well.
Testing strategy
==========
Check return statements in different contexts, including:
blocks at different nesting levels
reachable and unreachable return statements (for both procedures and functions)
generic subprograms
.. qmlink:: TestCaseImporter
TestCases
Write a test case
In Deep Nested Blocks
In Function Reachable
In Function Unreachable
In Generic Definition
In Procedure Reachable
In Procedure Unreachable
Perform
The activity is now locked.
No more test case can be created.



Hides the artifact browser using the button at the top. The artifact viewer is now using the full window.

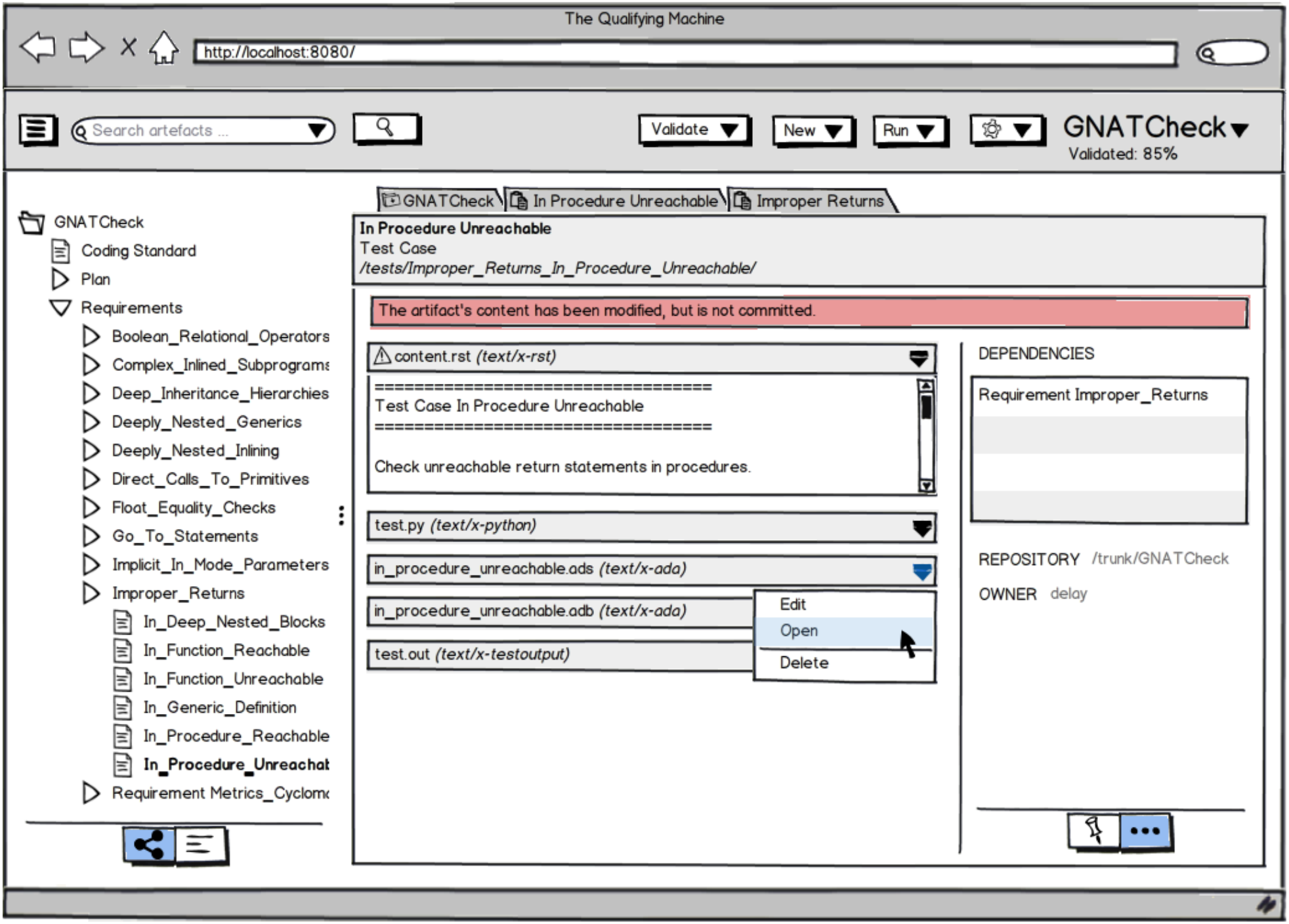
This is useful for small screens.

Opened artifacts can still be easily accessible using the MDI's tabs.



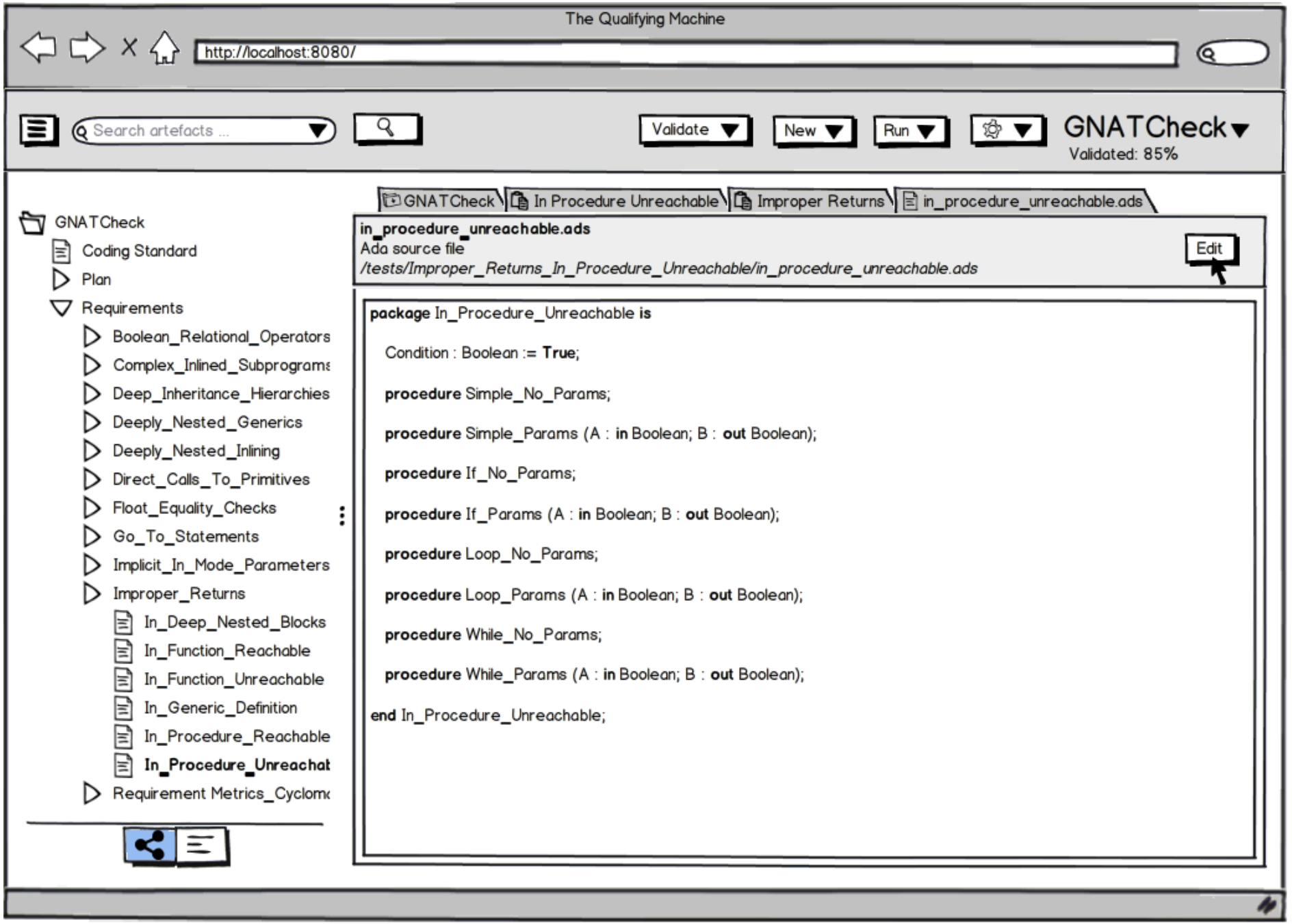
This 'no-artifact-browser' view is also useful when confronting two entries of the MDI.

This is needed when working on a requirement and one of its test cases, or between the test case source file and its output.



The preview offers edition capabilities. This is goog for quick fixes.

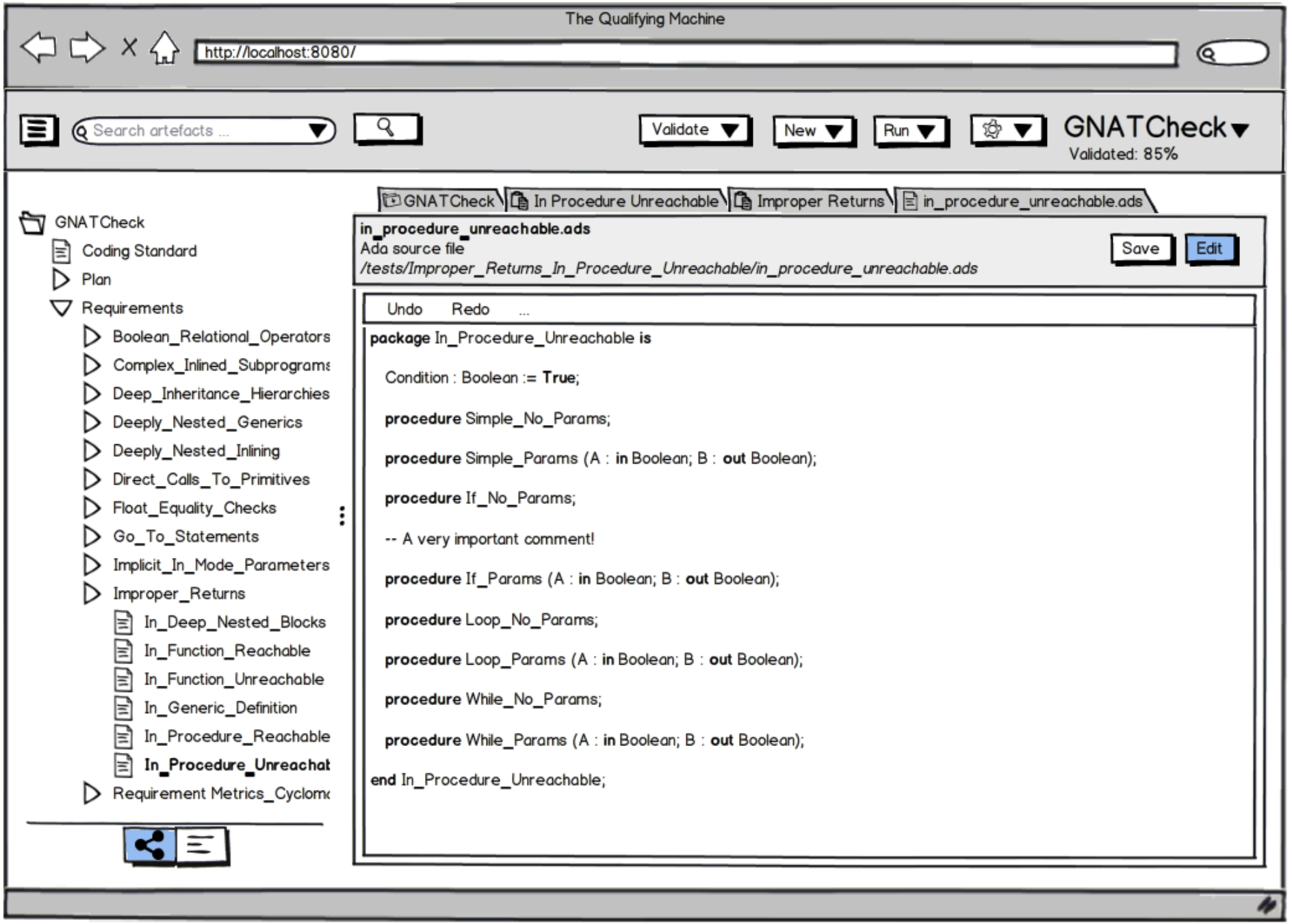
For a more confortable view, one can open the file as a new MDI's tab.



Opening a file as a tab permits the user to confront different type of file/artifact when using the dual view.

For example, one can edit both the body and the spec of an Ada package.

It is then easier to read/review...



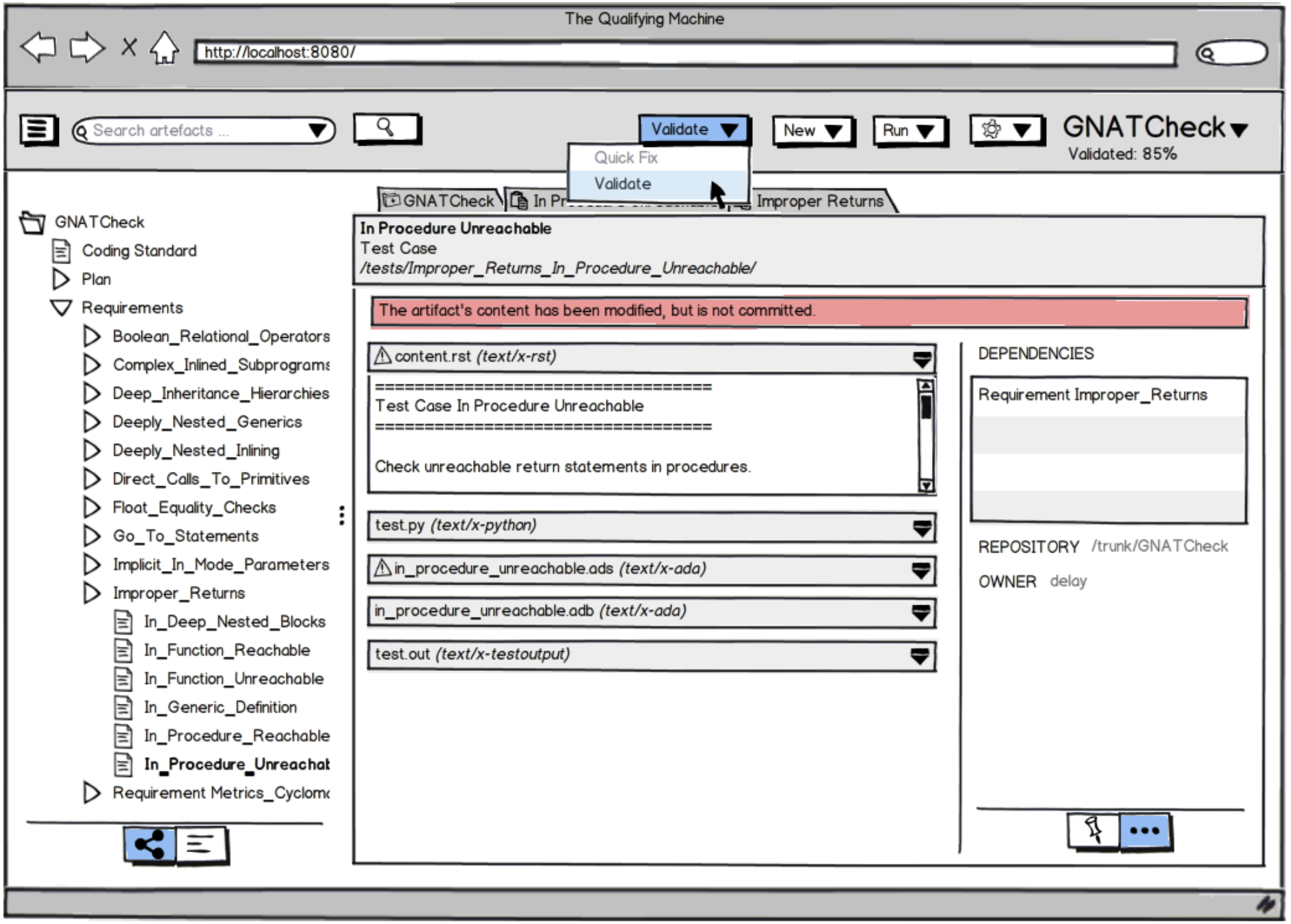
The save button appears when changes occured.

It is no use to weighten the interface with widget we don't need at a particular moment.

We only show them when we need them.

Both read-only and editable view are using the same modules associated with the mime-type of the file being manipulated than the one used by the preview panels.

... and edit!

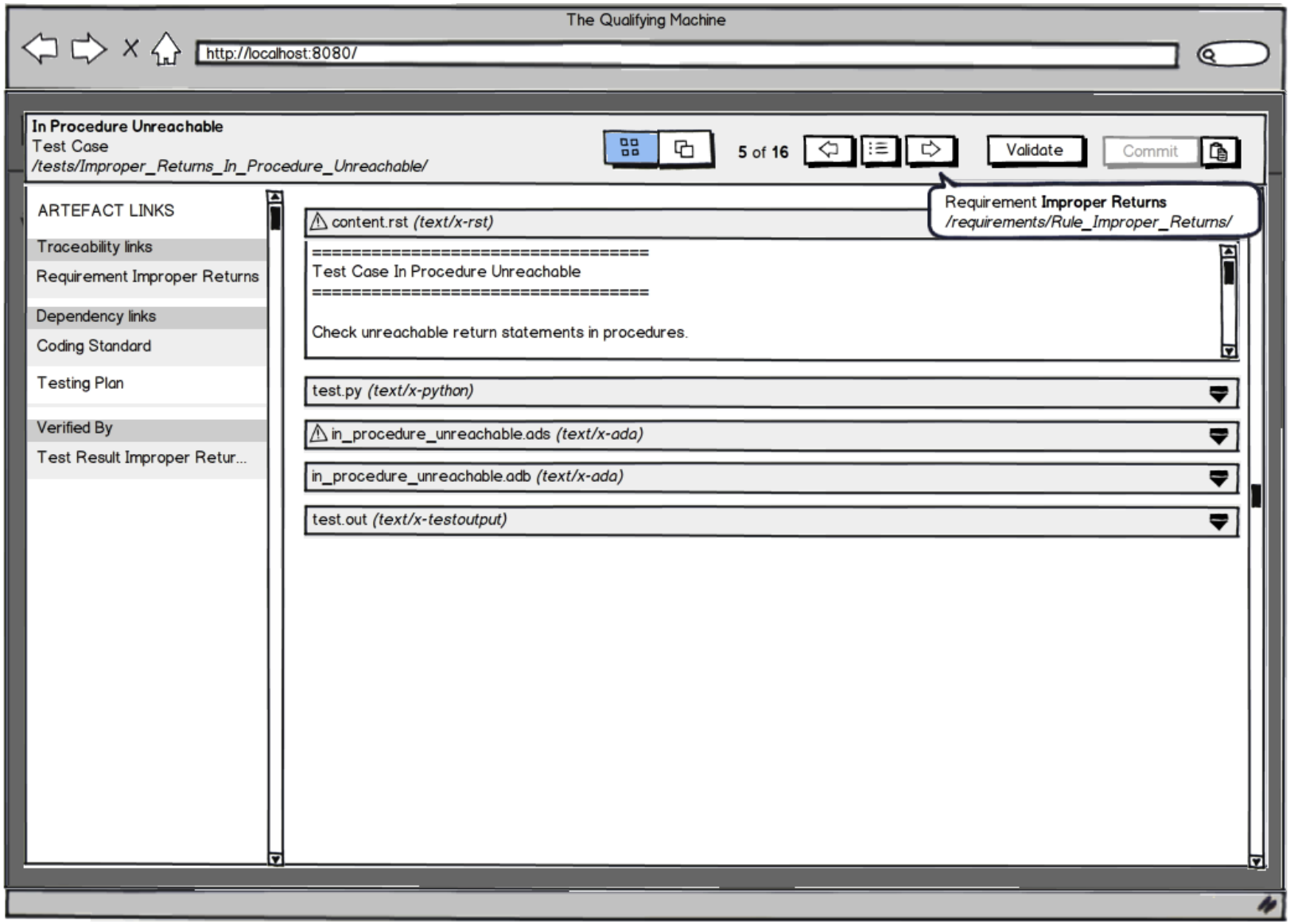


Time to validate our project.

One can choose between Quick Fixing the current opened artifact. This is useful when you have just fixed a typo, and do not want to go through the entire validation process again (since modifying an artefact can invalidate its children).

This option should be available or not depending on the project type. This may be a variable set in the QM model which allows or not quick fixing.

The other option, 'Validate' is always available (from the moment a change occurs, and one or more artifacts need to be re-validated).



Provide a view of the input and output links of the artifact.

This can be useful when revalidating an artifact which has not changed, but was invalidated by a modification done in an other artifact.

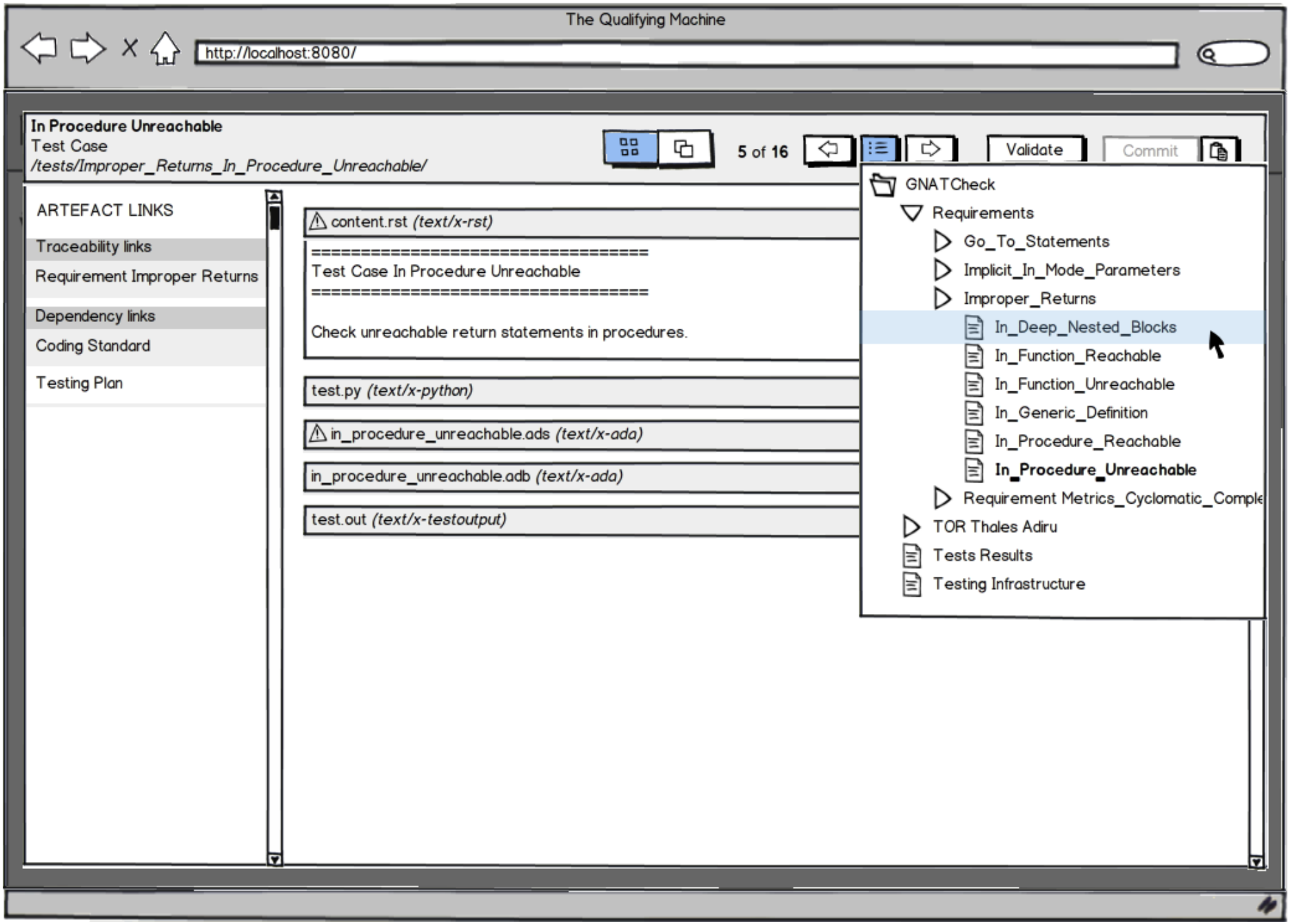
Provides two different views:
- the current state of the artifact
- a diff view with a previous version

Provides a mechanism to switch between artifacts that need to be validated.

Provides a toggle button to tag the artifact as 'Validated'.

Provides a 'Commit' button available when at least one artifact as been validated.

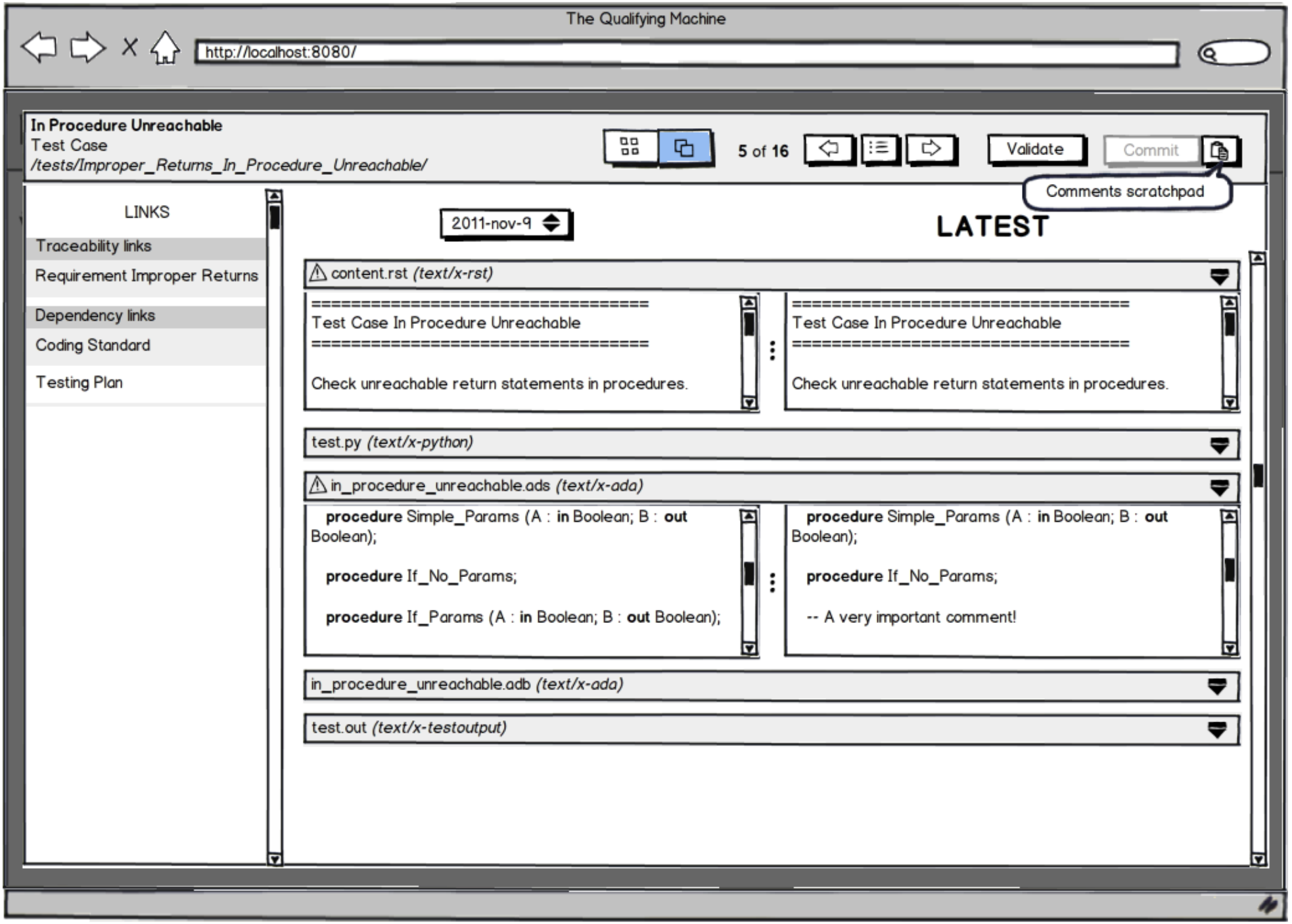
Displays the current state of the artifact's content.



Provides a tree view of artefacts in this validation set.

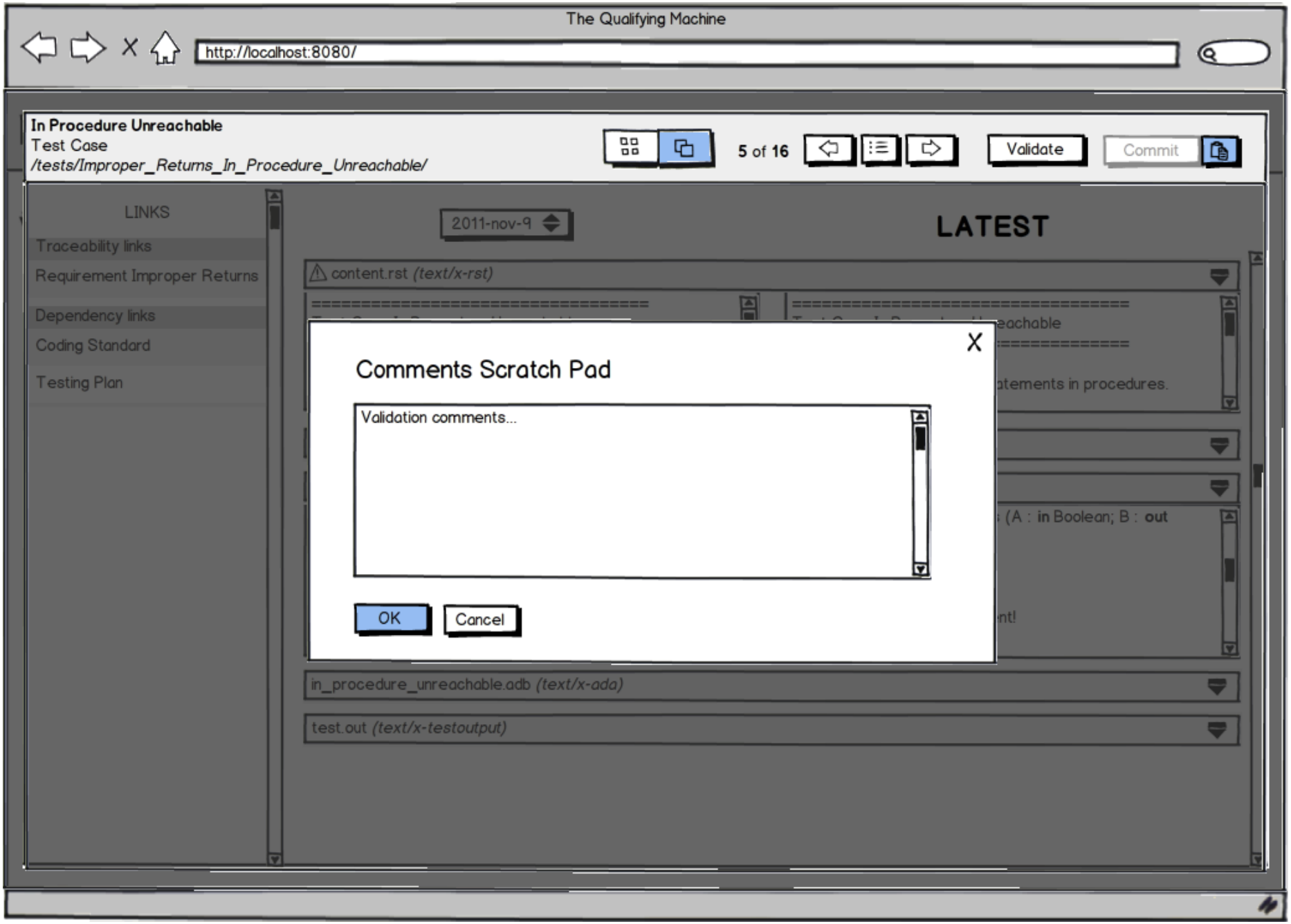
One can switch between artefact using two different ways:

- Through the left/right arrows, which navigate in the artefacts list which is sorted in a topological way.
- Through the tree view widget, which displays the artifacts respecting the traceability links.



The diff view confront two versions of the artefact contents.

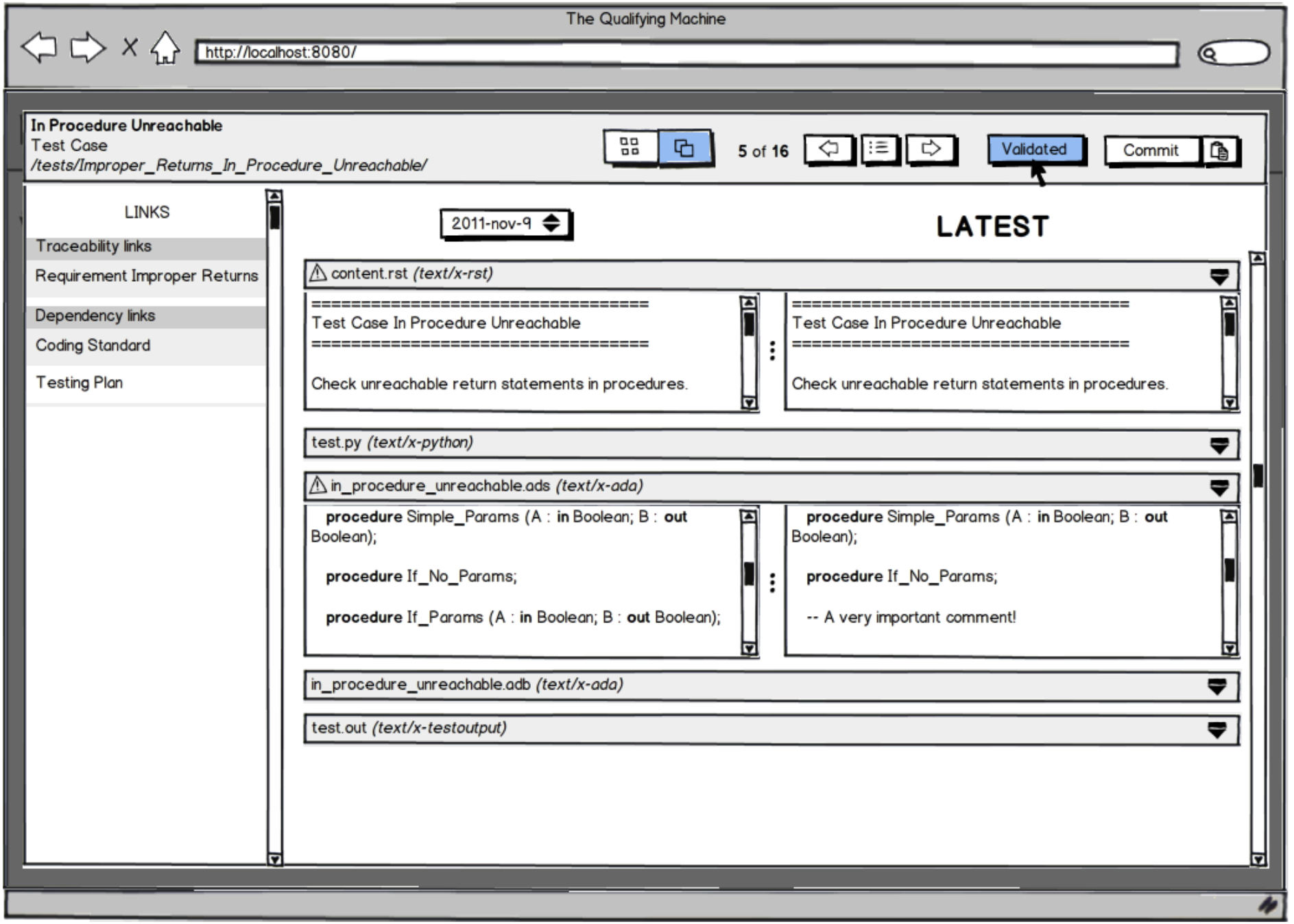
This view can confront any previous version of the artifact with the last one.



The comment scratch pad is a quick and easy way to write one's comments during the validation process, iteratively.

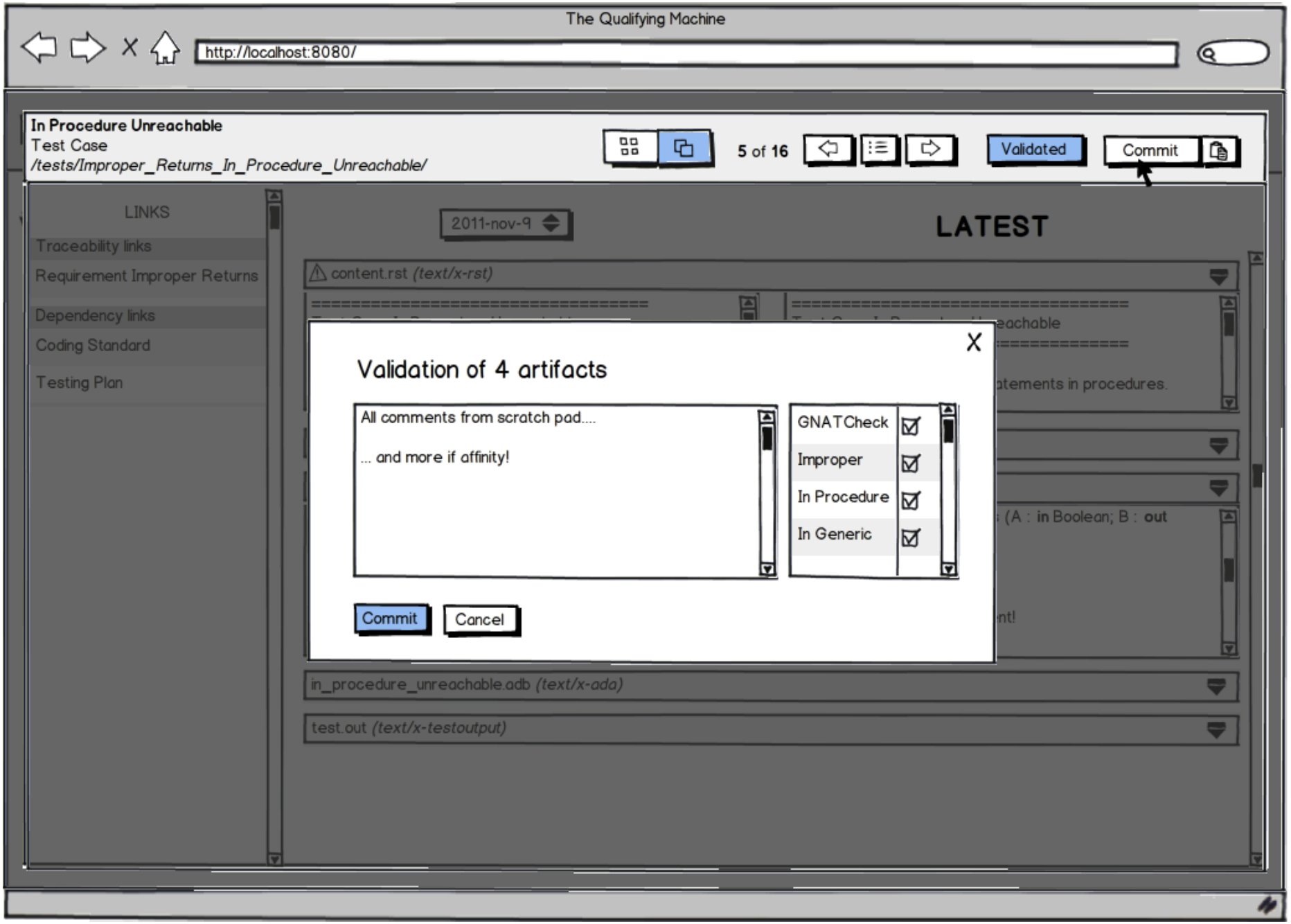
At the end, all comments written here are pasted in the final commit message.

This scratch pad is available through the toggle button close to the 'commit' button.



Once reviewed, we can tag an artifact as 'Validated' through the toggle button.

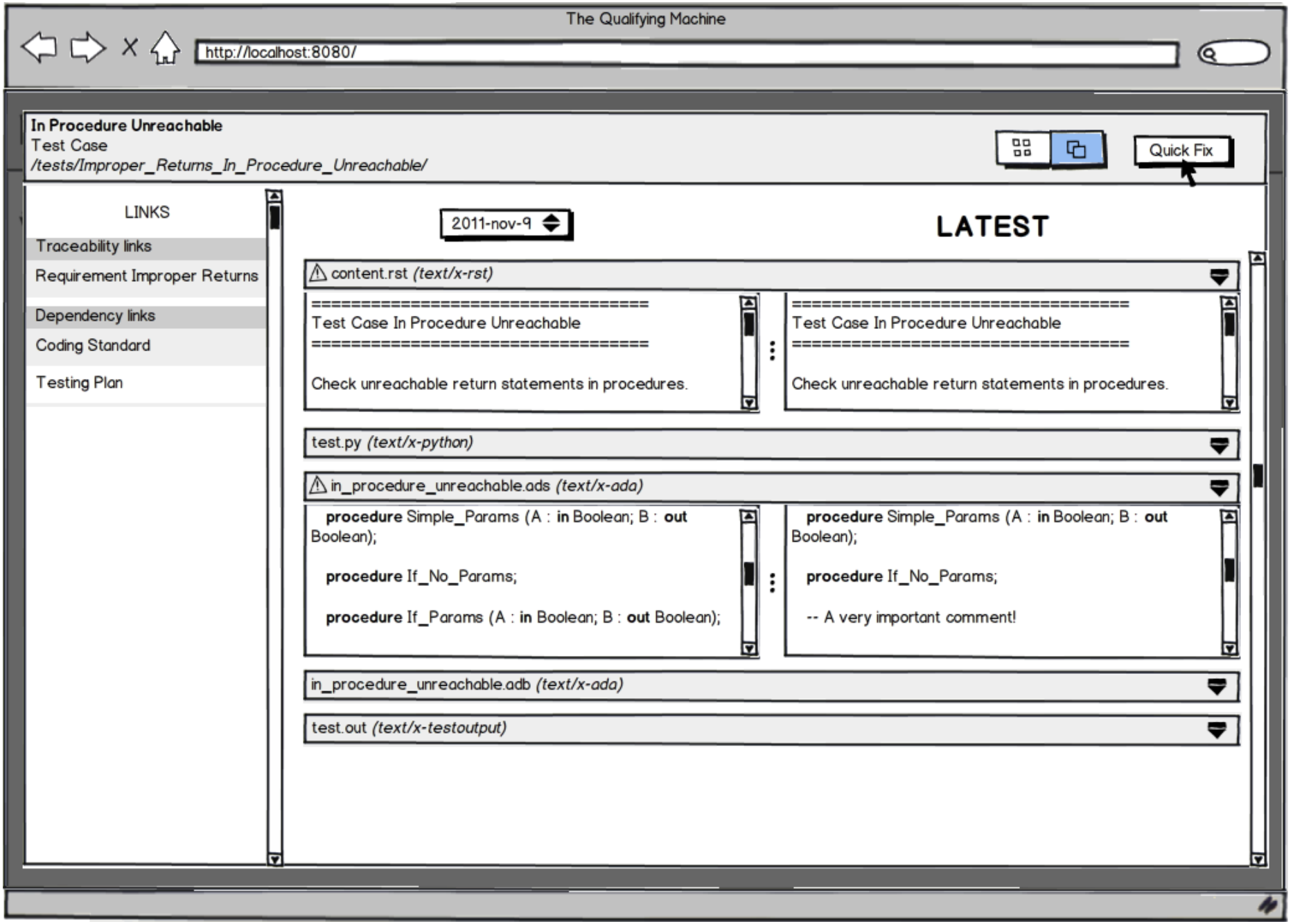
From then, at least one artifact is validated, so the commit button become available.



A set of artifacts has been validated. One can commit it.

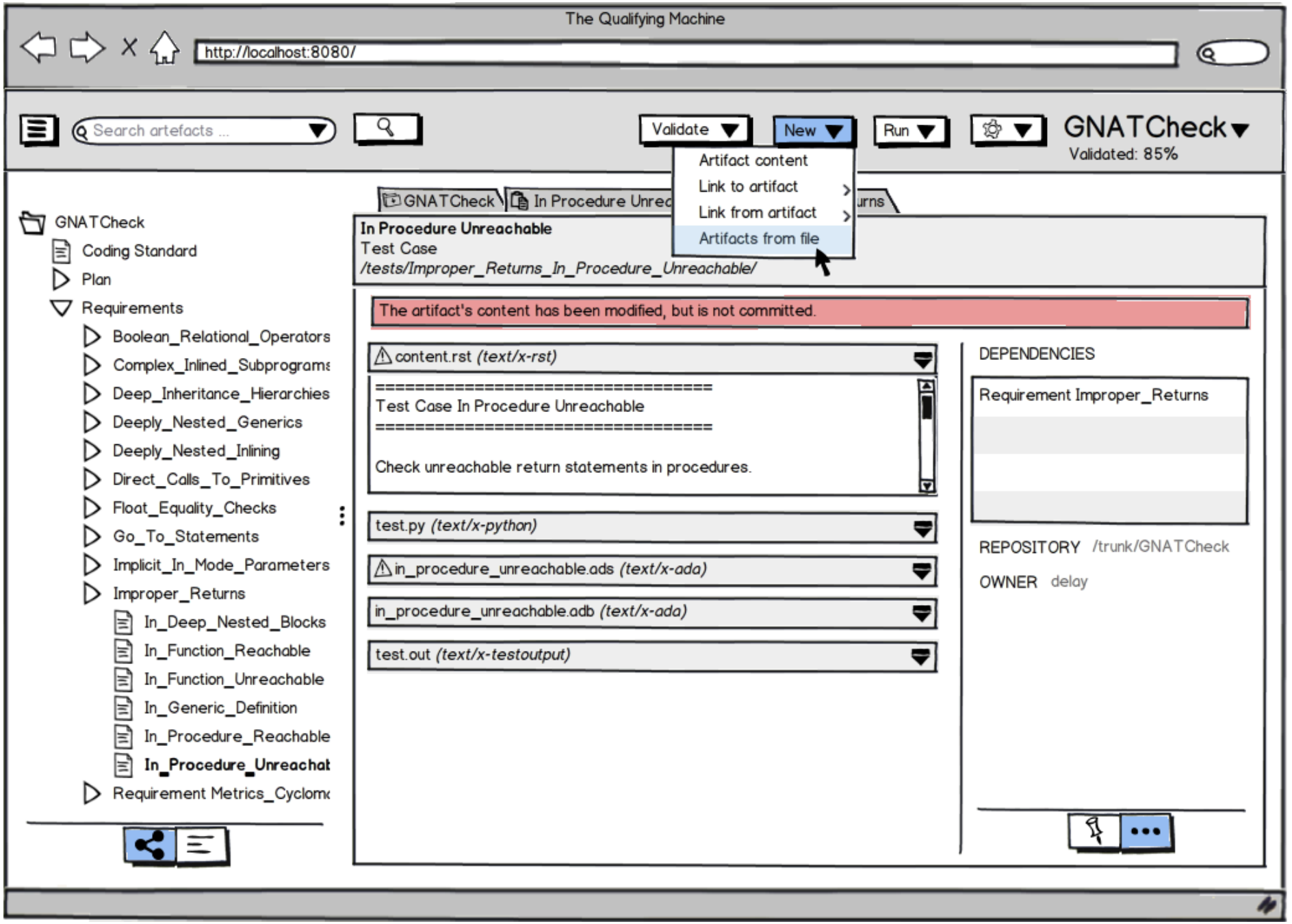
This allows multiple commits on the same set, dividing them into several atomical commits.

The comments are retrieved from the comments scratch pad. They can still be edited and completed before the final commit.



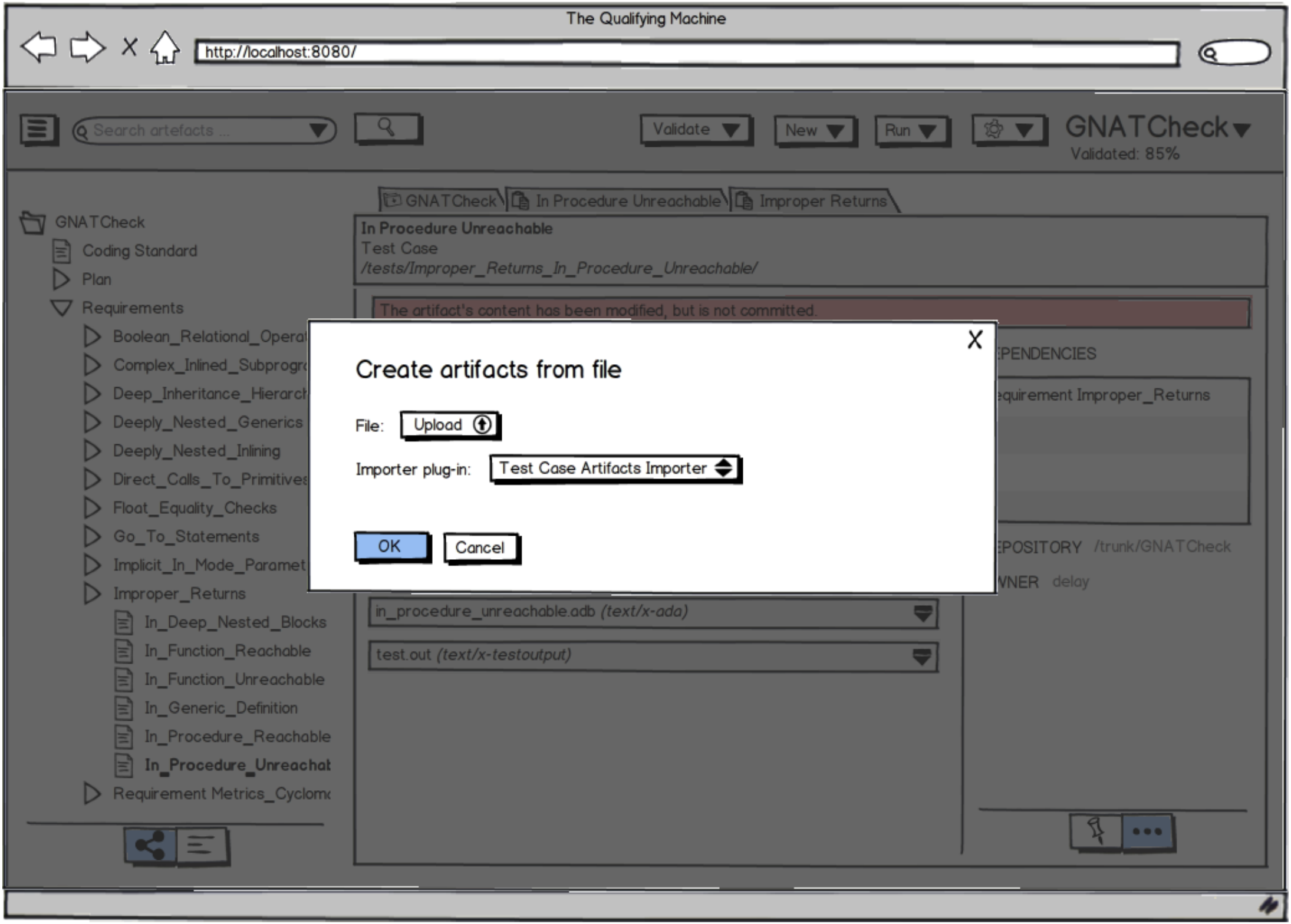
Quick Fix view with a minimum of capabilities.

Only one artifact can be reviewed and commited.



One can create artifact directly from the Qualifying Machine.

This can be done through the 'New' button.


The Qualifying Machine
http://localhost:8080/
Search artefacts ...
Validate
New
Run
GNATCheck
Validated: 85%
GNATCheck
In Procedure Unreachable
Improper Returns
GNATCheck
Coding Standard
Plan
Requirements
Boolean_Relational_Opera
Complex_Inlined_Subprogr
Deep_Inheritance_Hierarch
Deeply_Nested_Generics
Deeply_Nested_Inlining
Direct_Calls_To_Primitives
Float_Equality_Checks
Go_To_Statements
Implicit_In_Mode_Paramet
Improper_Returns
In_Deep_Nested_Blocks
In_Function_Reachable
In_Function_Unreachable
In_Generic_Definition
In_Procedure_Reachable
In_Procedure_Unreachat
Requirement Metrics_Cyclom
In Procedure Unreachable
Test Case
/tests/Improper_Returns_In_Procedure_Unreachable/
The artifact's content has been modified, but is not committed.
PENDENCIES
equirement Improper_Returns
POSITORY /trunk/GNATCheck
WNER delay
in_procedure_unreachable.adb (text/x-ada)
test.out (text/x-testoutput)
Create artifacts from file
File:
Upload
Importer plug-in:
Test Case Artifacts Importer
OK
Cancel