

# 1 Automated Visual Verification

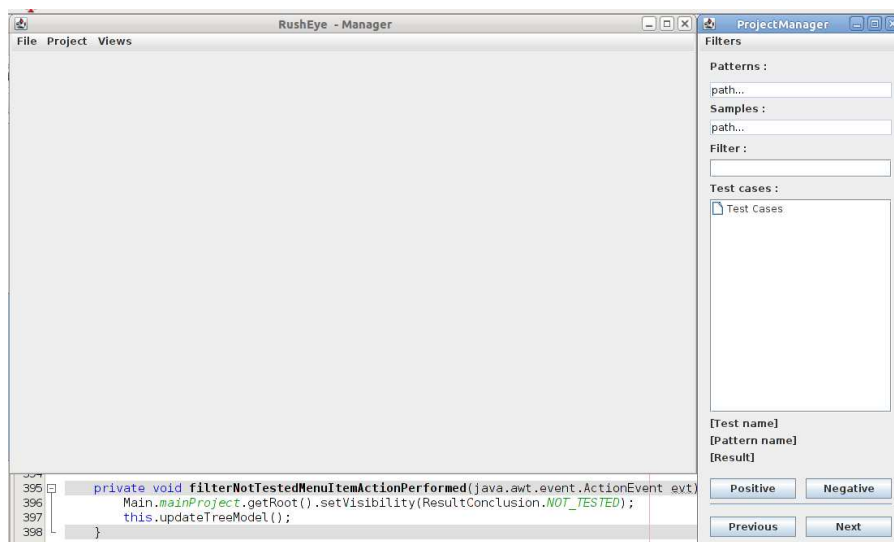
## 1.1 RushEye Manager

### 1.1.1 Brief view

RushEye Manager is graphical tool built to make using RushEye app easier.

Functionality of Manger :

- **Creating suite** - RushEye crawl option.
- **Showing results (images)** - load xml and images, to see them and manually set comparision result.
- **Parsing suite** - RushEye parse option, that can take into account our manual changes.
- **Changing result descriptor** - After parse we can still view images and do changes, that are automatically saved.



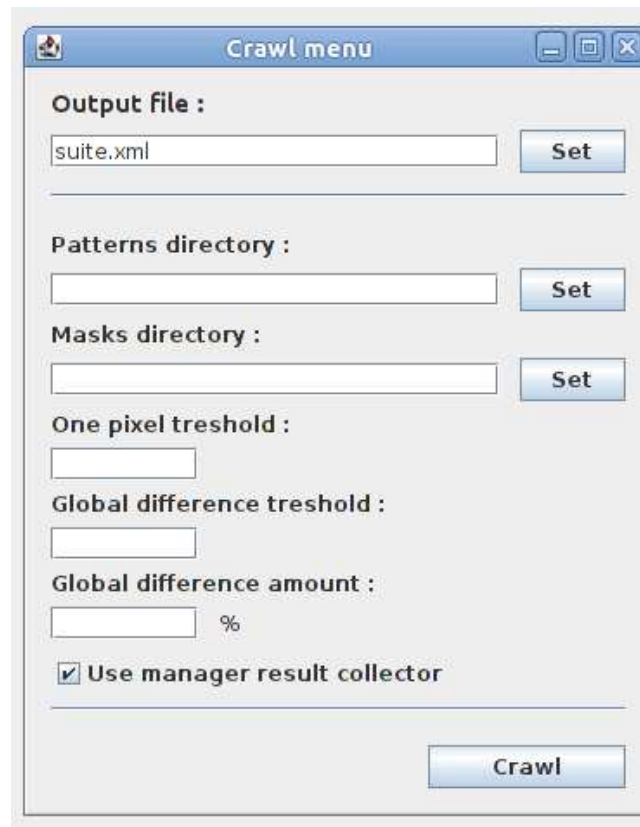
Rysunek 1: Initial view.

When we start the manager we see 2 frames, main interface frame and smaller project manager frame. Let's start with basic usage :

### 1.1.2 Crawl

Let's go to menu bar, then **Project - Generate Suite Descriptor**.

We should see new frame :



The image shows a window titled "Crawl menu" with a standard Windows-style title bar. Inside the window, there are several configuration options:

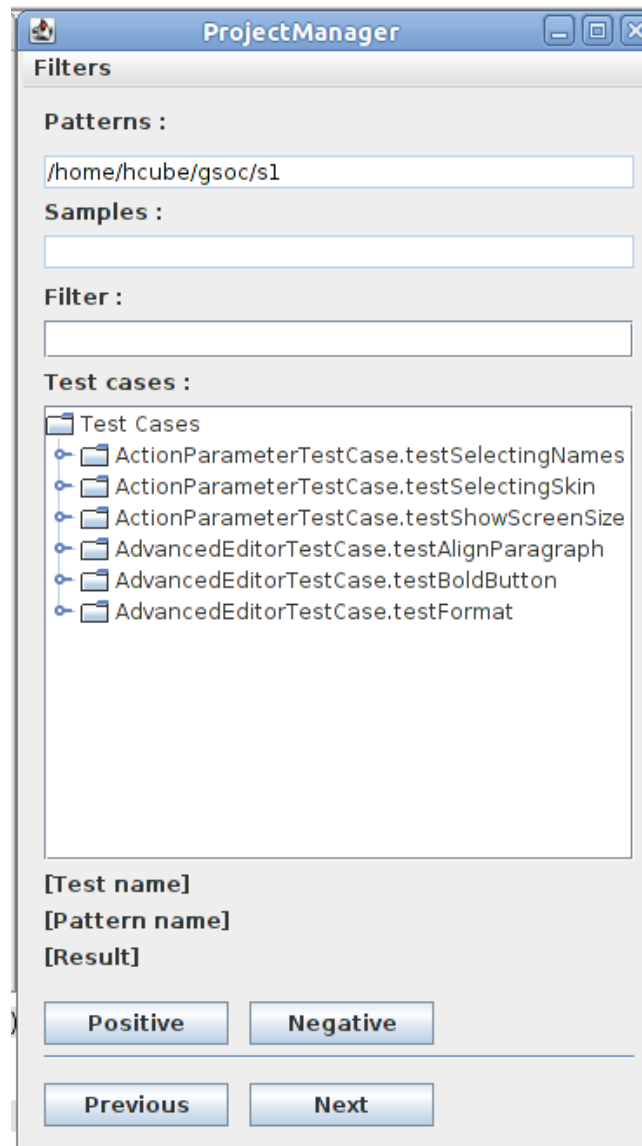
- Output file :** A text box containing "suite.xml" and a "Set" button to its right.
- Patterns directory :** An empty text box and a "Set" button to its right.
- Masks directory :** An empty text box and a "Set" button to its right.
- One pixel threshold :** An empty text box.
- Global difference threshold :** An empty text box.
- Global difference amount :** An empty text box followed by a "%" symbol.
- ☒ **Use manager result collector**
- Crawl** button at the bottom right.

Rysunek 2: Crawl menu.

Here we can set all parameters like patterns/masks directory, perception, and output filename. When we are ready we can click **Crawl** button. If some parameters are missing, we will be informed by alert window. The xml was saved to file that we specified before, and also automatically loaded to Manager.

### 1.1.3 Loading project from xml file

When we generate suite, this step is done automatically, but if we already have one, we can load it too. Let's go to the menu bar, then **File - Project From Descriptor**. In file chooser dialog we should set our xml file. That should generated tree of tests in Project Manager :

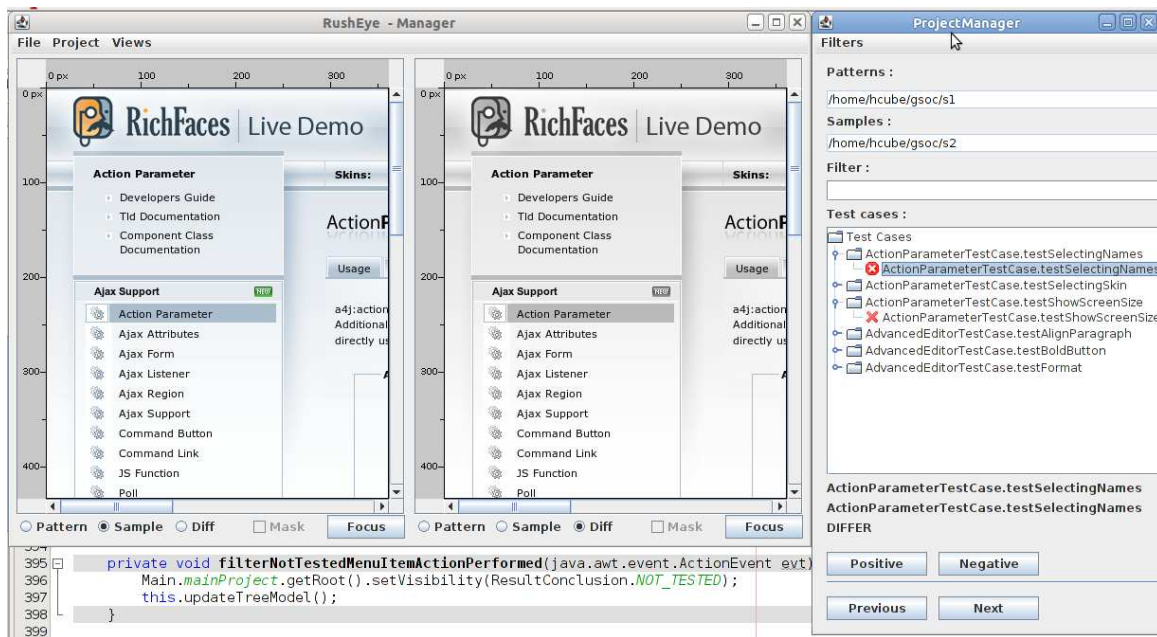


Rysunek 3: Project manager.

To see some images, we need to click on leaf in the tree (pattern of suite descriptor). But there is one thing to remember. To actually see images, we need to define path to directories, where those images are - Patterns path, and Samples path. In example above, we can see that **Patterns** are already defined - because it is project loaded automatically after crawl (path was stored). If we don't define both path we will see warning, when we open a test. The easiest way to load a **path** is through menu bar **Project - Set samples path** and **Project - Set patterns path**.

### 1.1.4 Double View

When we have both params set we can click on a leaf in tree.



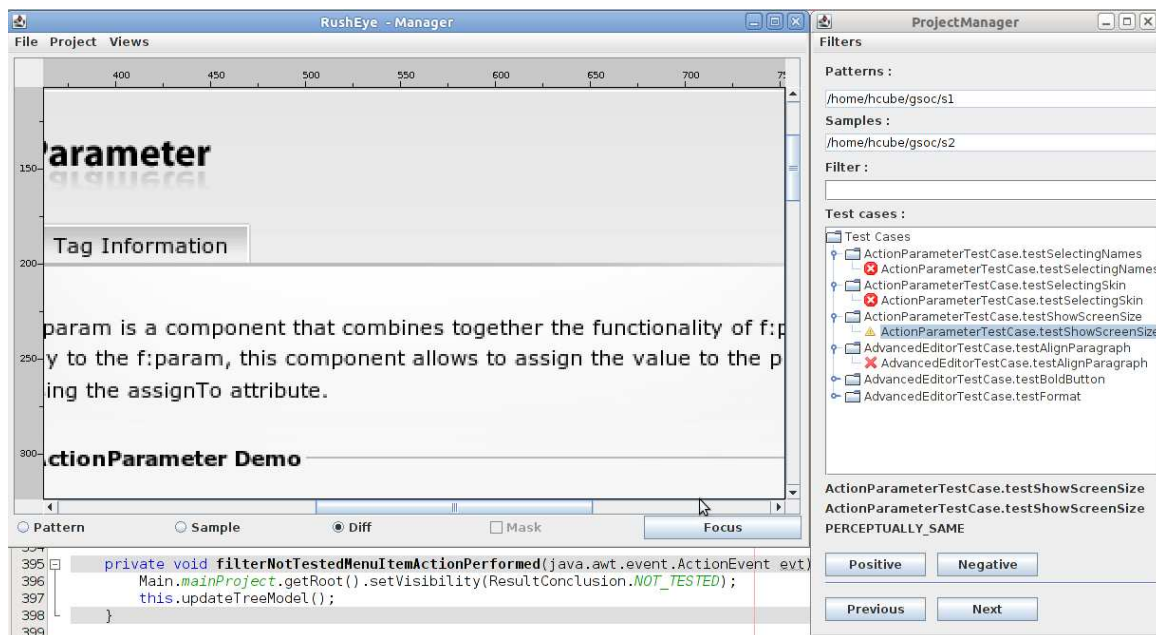
Rysunek 4: After clicking on test.

As we can see, images are displayed, and icon near test was changed. **Red cross** means that the pattern is not tested. When we click on a pattern, ad hoc comparison is done. In case above *testSelectingNames* was marked as Diff.

Let's focus on images. What we see is called **Double View**. Here we can compare 2 images, looking for changes. **Radio buttons** at the bottom of images allow us to change between sample, pattern and diff image. **Focus** button allows us to move to changed part of diff image. Scrollbars are synchronized - when we move one, the other moves automatically.

### 1.1.5 Single View

Using menu **Views - Single View** we can switch to another view. It is similar, but instead of 2 images we have one. Also we can zoom in/out image using mouse wheel.

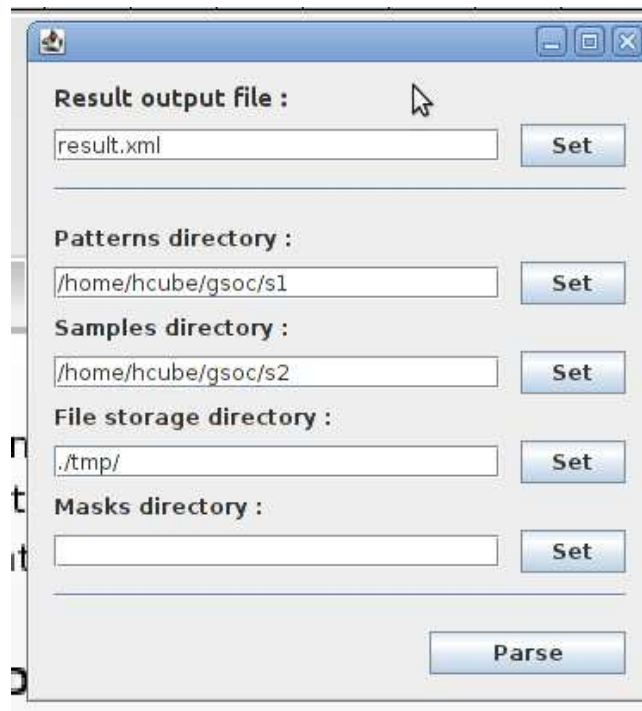


Rysunek 5: Single view.

On image above we can see another icon in tree - we marked *testShowScreenSize* as **PERCEPTUALLY\_SAME** using **Positive** button. That is how we can make changes manually.

### 1.1.6 Parse

Parse option is most important for Manager. We can run it through **Project - Generate Result descriptor**. We should then see new frame.



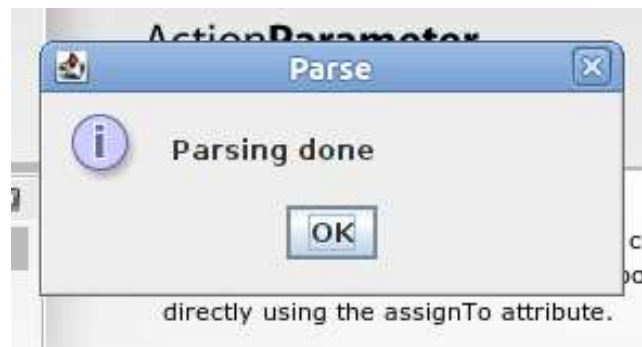
Rysunek 6: Parse frame.

Like with crawl button we need to set all parameters and then simply hit the **Parse**.

Parse functionality is strongly determined by configuration set in xml suite. Normally we use result collector *org.jboss.rusheye.result.collector.ResultCollectorImpl*. If that one is defined in our xml file, parsing will be done normally.

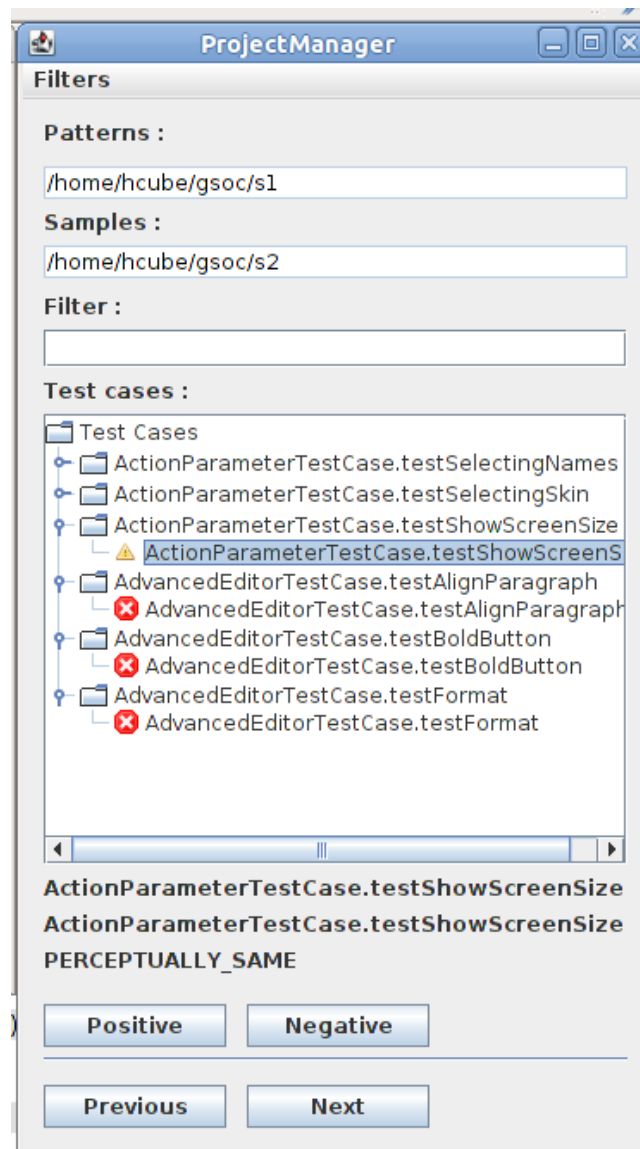
If we define there *org.jboss.rusheye.parser.ManagerResultCollector* we will run another collector, which takes our modifications in tree into account. When test case in tree is set to positive or negative - this result will overwrite result that would be set normally by parser.

Decision about collector is made when we run **Crawl** - using checkbox, or we can overwrite classpath in xml file manually.



Rysunek 7: After parsing.

When we hit parse, another thread will be run. When it finishes, alert is shown.



Rysunek 8: Results of parsing.

As we can see, all 'not tested' cases, were overwritten by parser (they are diff now). PERCEPTUALLY\_SAME case we set before is still present.

```

</test>
<test name="AdvancedEditorTestCase.testBoldButton">
  <pattern name="AdvancedEditorTestCase.testBoldButton" output="AdvancedEditorTestCase.testBoldButton.png" result="DIFFER">
    <comparison-result>
      <area height="906" width="1040"/>
      <rectangle>
        <min x="48" y="319"/>
        <max x="237" y="494"/>
      </rectangle>
      <total-pixels>942240</total-pixels>
      <masked-pixels>0</masked-pixels>
      <perceptible-different-pixels>352</perceptible-different-pixels>
      <global-different-pixels>68</global-different-pixels>
      <unperceptible-different-pixels>0</unperceptible-different-pixels>
      <same-pixels>941820</same-pixels>
    </comparison-result>
  </pattern>
</test>
<test name="ActionParameterTestCase.testShowScreenSize">
  <pattern name="ActionParameterTestCase.testShowScreenSize" output="ActionParameterTestCase.testShowScreenSize.png" result="PERCEPTUALLY_SAME">
    <comparison-result>
      <area height="1234" width="1002"/>
      <rectangle>
        <min x="219" y="713"/>
        <max x="237" y="767"/>
      </rectangle>
      <total-pixels>1236468</total-pixels>
      <masked-pixels>0</masked-pixels>
      <perceptible-different-pixels>350</perceptible-different-pixels>
      <global-different-pixels>68</global-different-pixels>
      <unperceptible-different-pixels>0</unperceptible-different-pixels>
      <same-pixels>1236050</same-pixels>
    </comparison-result>
  </pattern>
</test>

```

Rysunek 9: Results of parsing.

Here we can see that our manual change is also shown in result descriptor.

What when we want to change another test, without running parse again? We can simply change that test result in tree, and change will be automatically written into result.xml.

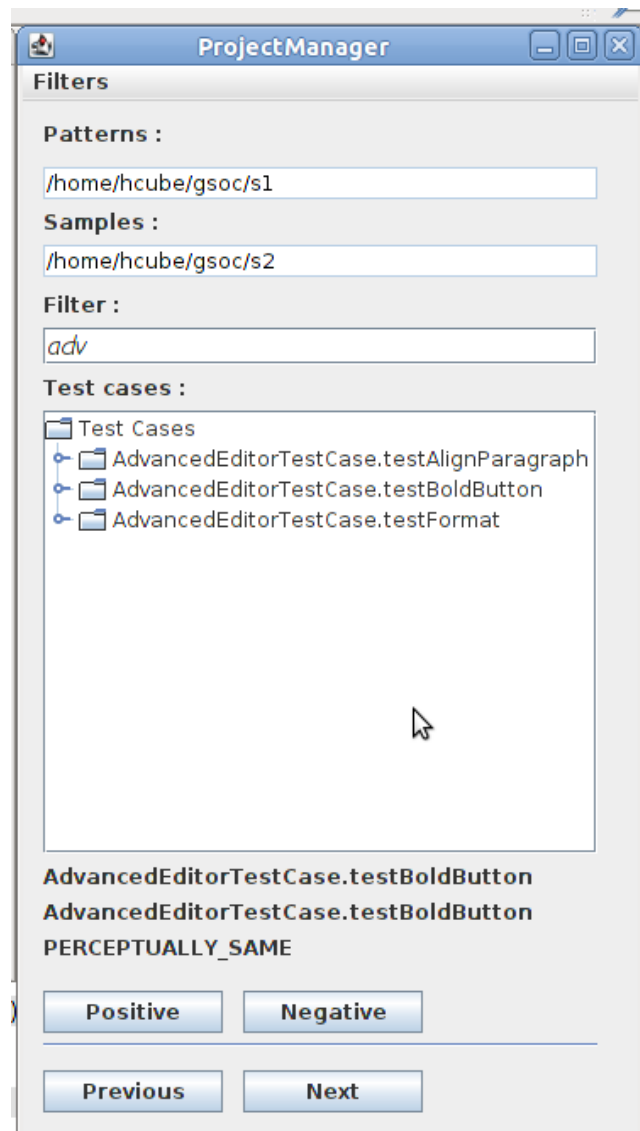
```
</test>
<test name="AdvancedEditorTestCase.testBoldButton">
  <pattern name="AdvancedEditorTestCase.testBoldButton" output="AdvancedEditorTestCase.testBoldButton.AdvancedEditorTestCase.testBoldButton.png" result="PERCEPTUALLY_SAME">
    <comparison-result>
      <area height="966" width="1040"/>
      <rectangle>
        <min x="48" y="319"/>
        <max x="237" y="494"/>
      </rectangle>
      <total-pixels>942240</total-pixels>
      <masked-pixels>0</masked-pixels>
      <perceptible-different-pixels>352</perceptible-different-pixels>
      <global-different-pixels>68</global-different-pixels>
      <unperceptible-different-pixels>0</unperceptible-different-pixels>
      <same-pixels>941820</same-pixels>
    </comparison-result>
  </pattern>
</test>
<test name="ActionParameterTestCase.testShowScreenSize">
  <pattern name="ActionParameterTestCase.testShowScreenSize" output="ActionParameterTestCase.testShowScreenSize.ActionParameterTestCase.testShowScreenSize.png" result="PERCEPTUALLY_SAME">
    <comparison-result>
      <area height="1234" width="1002"/>
      <rectangle>
        <min x="219" y="713"/>
        <max x="237" y="767"/>
      </rectangle>
      <total-pixels>1236468</total-pixels>
      <masked-pixels>0</masked-pixels>
      <perceptible-different-pixels>350</perceptible-different-pixels>
      <global-different-pixels>68</global-different-pixels>
      <unperceptible-different-pixels>0</unperceptible-different-pixels>
      <same-pixels>1236050</same-pixels>
    </comparison-result>
  </pattern>
</test>
```

Rysunek 10: testBoldButton is now PERCEPTUALLY\_SAME instead of DIFFER.



### 1.1.7 Filters and navigation

projectManager Frame has also few widgets improving work:



Rysunek 11: **Filter text field** -where we can put part of the test name, to filter out not matching tests.

There are also buttons to navigate through tree - **Previous** and **Next**



Rysunek 12: **Filter menu** - in menu bar we can choose **Filters - Show all/Not tested/Diff** to show particular tests.