# Hansoft Test Case Management

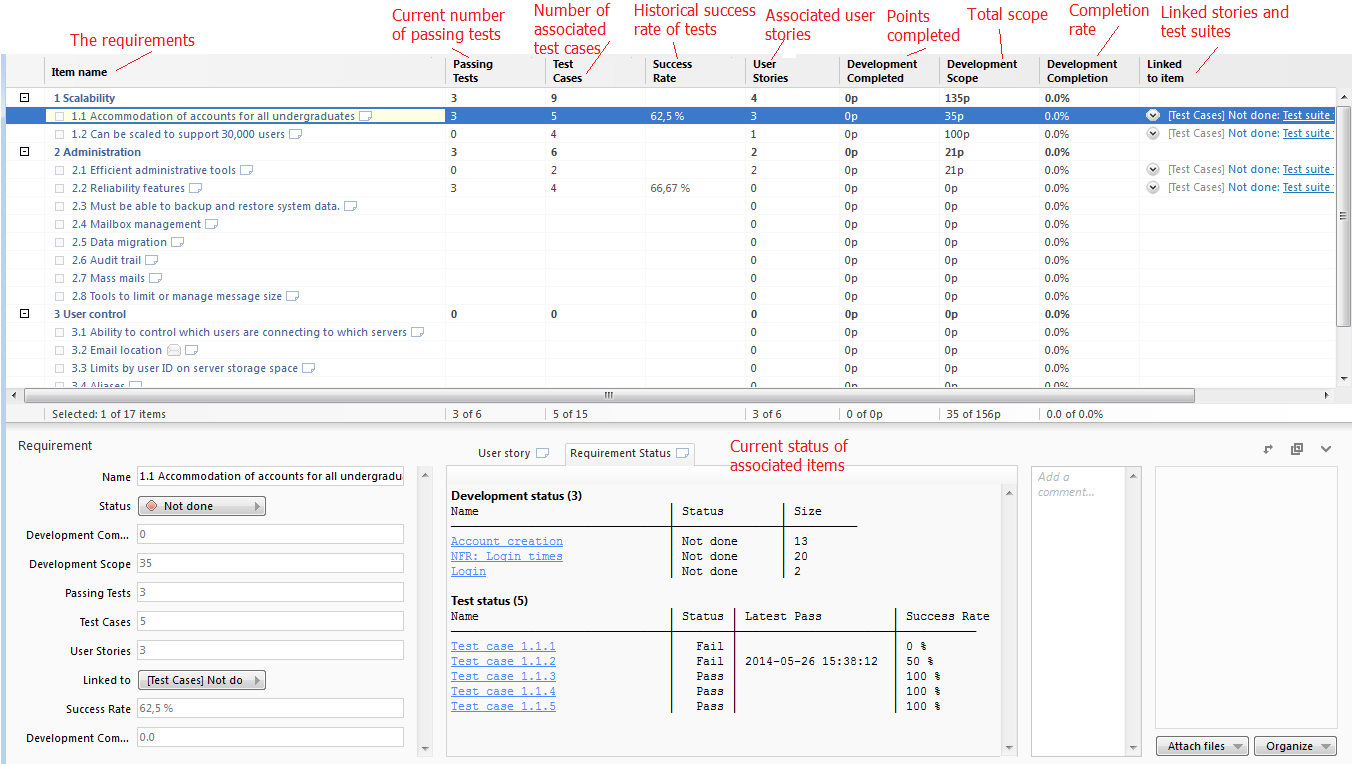
Before reading this document it’s a good idea to follow the instructions in the Readme to set up the application for either testing it or continuing developing it.

## Introduction

Hansoft-TCM is a sample project that shows how you can use the SDK to wire Hansoft in a way that you can manage test cases towards a long list of requirements. This can be particularly useful in environments where the development is run in an agile manner, but the financial contracts are created in a waterfall manner.

The database contains three projects; Requirements, Development and Test Cases. The design in this case is that each Requirement will be associated to one test suite and any number of user stories. A test suite can contain any number of test cases and user stories can be broken down further.

## Requirement project



Above is description of the components managed by the plugin. All these components will be automatically filled out, based on the data in the Development project and the Test Case project.

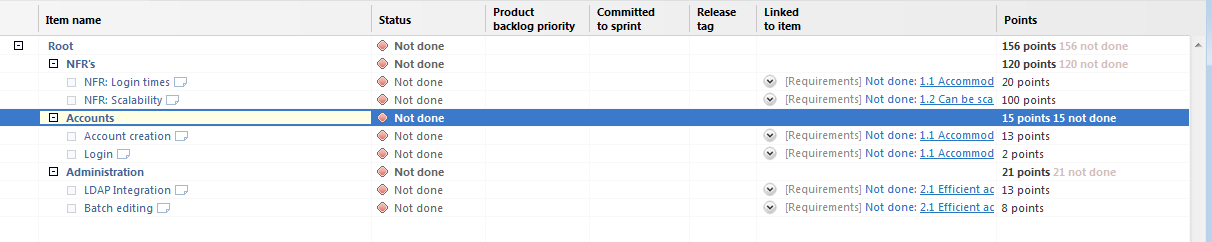
If you right click a requirement you will be given the following two options in the menu:



If you choose “Add Test Suite” a dialogue will be brought up that allows you to create a new Test Suite. After creating it, it will be added to the Test Case project’s backlog. Per design it can only manage one test suite per requirement, but it will allow you to add more. In the case of several test suites, column summaries will not behave very well. A link will show up both in “Linked to item” and in the Requirement status box that will allow you to easily go to the newly created suite.

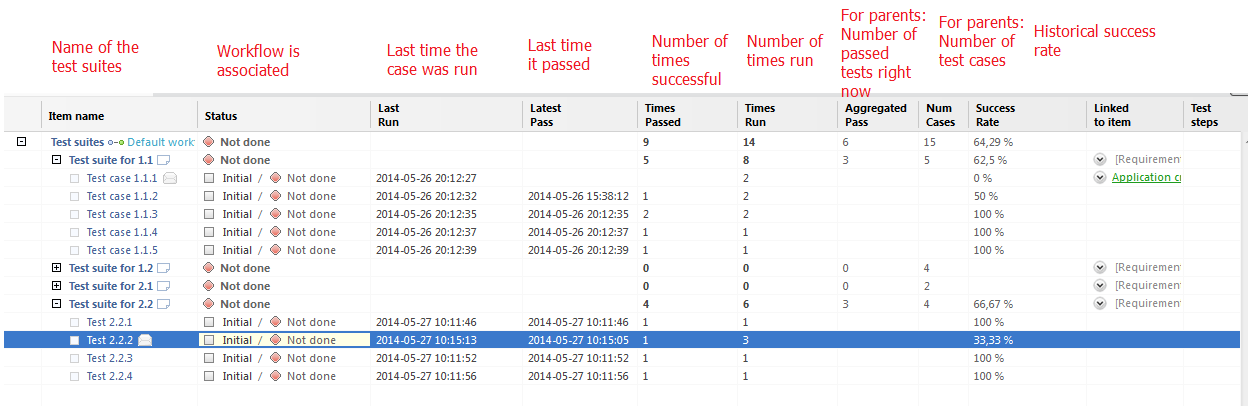
If you choose “Add User Story” a dialogue will be brought up that allows you to create a new User Story. After creating it, it will be added to the Development project’s backlog. You can add any number of user stories. A link will show up both in “Linked to item” and in the Requirement status box that will allow you to easily go to the newly user story.

## Development project

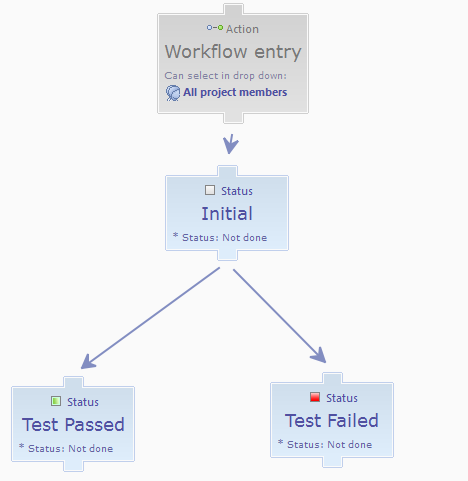


There are no automatic calculations in the development project. New user stories will be added under the root node, so don’t rename or delete that node unless you change the Client side plugin.

## Test Case project



The test case project includes a very simple workflow:



All calculations are built upon this workflow. If you want to change the names or the structure of the workflow you need to modify the following conditions in JeanSettings.xml.

<Condition Expression="$WorkflowStatus == 'Test Passed'">

<Condition Expression="$WorkflowStatus == 'Test Failed'">

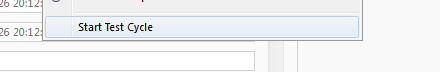
Whenever a test moves to “Test Passed” the columns; Last Run, Latest Pass, Times Passed and Times run will be updated.

When a test moves to “Test Failed” the columns; Last Run and Times Run will be updated.

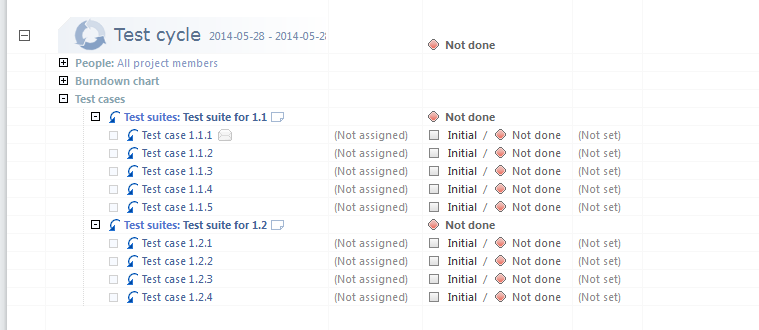
Aggregated Pass is calculated on parents, which is the number of test cases beneath where Last Run is the same as Latest Pass.

Success Rate is the same as Times Passed through Times Run.

The user can start a test cycle by selecting any number of items in the backlog, right clicking them and selecting:



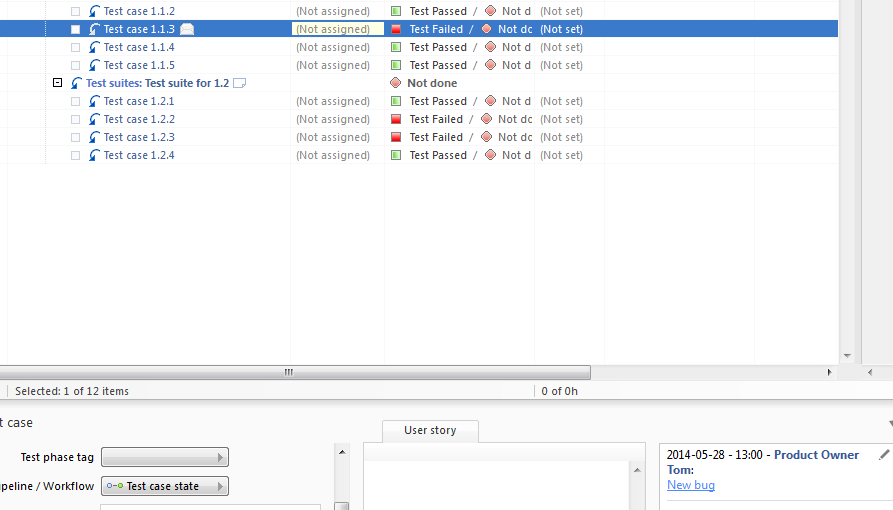
This will create a new test cycle with the selected element. Right now the test cycle length is always 1 day.



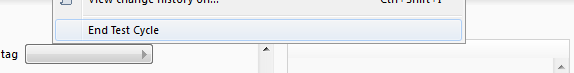
When a test fails and the user wants to report a bug he can right click the test case and select:



This will bring up a dialogue where the user can enter the details. After the bug has been created a link will be created between it and the test case and a comment will be posted with a link to the bug:



When a test cycle is completed it can be ended by right clicking the header and select:



This action will delete the test cycle, bring back all test cases to the backlog and reset the state for them to initial.

## JeanSettings.xml

This file configures all the calculation behaviors. If you want to change for example column names you need to modify that file and restart Jean.