

User Manual

TestLink version 1.7

Copyright © 2004,2005,2006 TestLink Development Team

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. The license is available in ["GNU Free Documentation License" homepage](#).

1 General information

TestLink is web based **Test Management** system. This manual should serve as source for users to understand processes, terms and organization of testing with TestLink. See to **Installation manual** for more information about system requirements, installation steps and configuration. The latest documentation is available on www.teamst.org or testlink.sourceforge.net.

See an example of TestLink workflow:

1. Administrator creates a Test Project "Fast Food" and two users Adam with rights "Leader" and Bela with rights "Senior tester".
2. Leader Adam imports Software Requirements. He generates empty Test Cases for a part of these requirements.
3. Tester Bela describes a test scenario of these Test cases that are organized into two Test Suites "Hot" and "Cold".
4. Adam creates Keyword: "Regression" and assigns this keyword to ten of these Test Cases.
5. Adam creates a Test Plan "Fish & Chips", Build "Fish 0.1" and add Test Cases with keywords "Regression".
6. Adam and Bela execute and record the testing with result: 5 passed, 1 failed and 4 are blocked (because a bug causes that some functionality is not available).
7. Developers make a new build "Fish 0.2" and Bela tests the failed and blocked test cases only. Exceptionally all these remaining five Test cases passed.
8. A top manager of this team "Brown Bear" would like to see results. Administrator explains him that he can create account himself on the login page. Manager do it. He has default "Guest" rights and could see results and Test cases. He can see that everything passed in overall report ;-) and problems in build "Fish 0.1" in a report for particular Build. But he can change nothing (so he cannot cause any harm).

1.1 Overall structure

There are three cornerstones: **Test Project**, **Test Plan** and **User**. All other data are relations or attributes for this base. First, definition of a couple of terms that are used throughout the documentation and testing world.

1.2 Terminology

Test Case describes a testing task via testing steps and expected results. Test cases are the fundamental piece of Test Specification.

Test Suite organizes Test cases to units. It structures Test Specification into logical parts. You can imagine it as folders, containers or chapters of Test Specification.

*Test Suite is known as both **components** and **categories** in TL 1.6 and earlier.*

Test Plan are created when you'd like to execute test cases. Test Plan includes Builds, Testing Set, Milestones, test assignment to users and Test results.

Test Project is basic box of the TestLink structure. Test project will undergo many different versions throughout their life times. Test project includes Test Plan with Results, Test Specification with Test Cases, Requirements and more. Users within a project has defined role.

*Test project is what TL 1.6 knows as **Product**.*

Test Specification include Test cases organized into Test Suite and via Keyword. Chosen set of test cases are assigned to a particular Test Plan for execution.

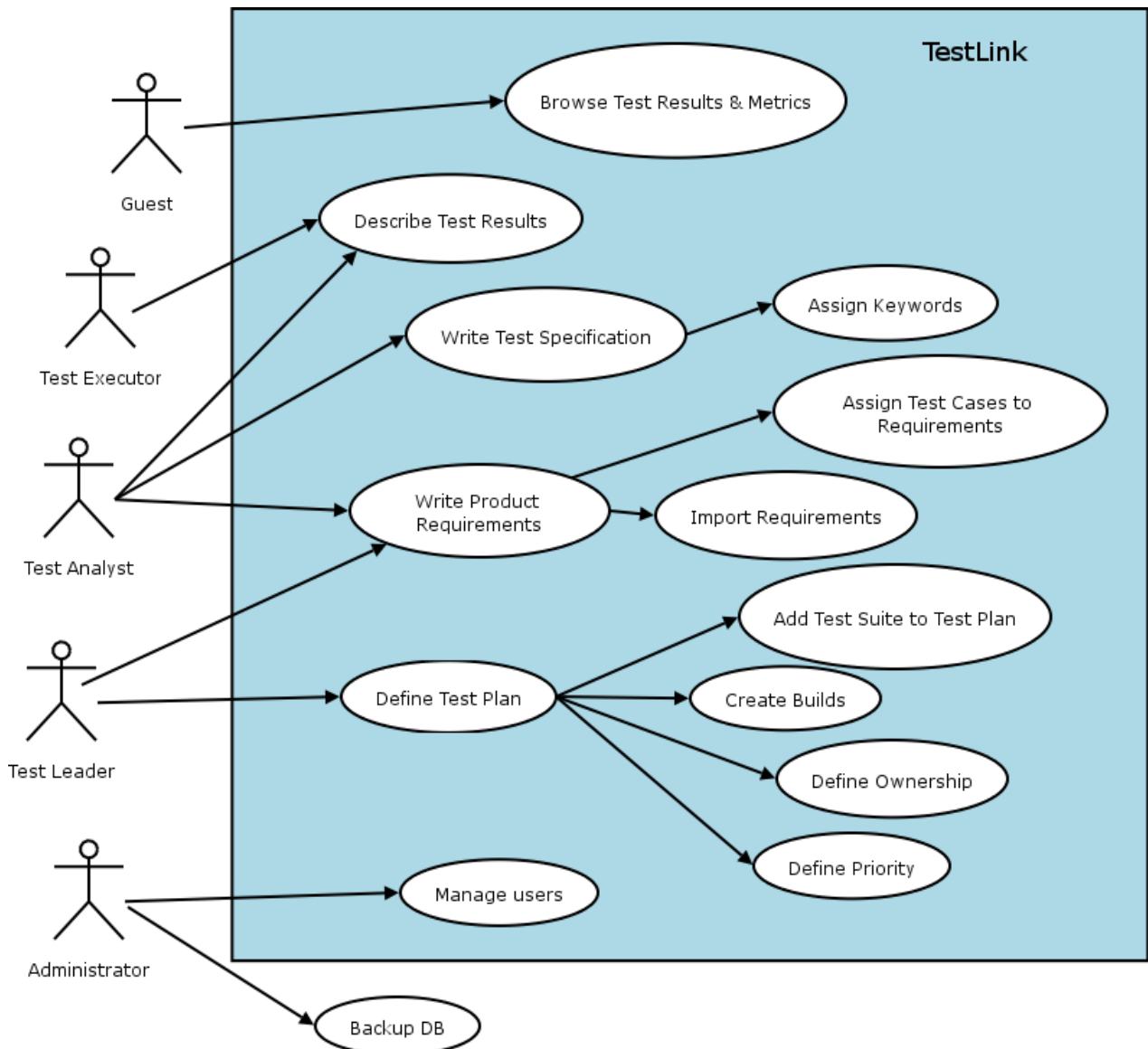


Illustration 1: User roles

User: each TL user has a Roles, that defines available TestLink features. See more in chapter User Administration. The Illustration 1 shows a common activity according to user roles.

2 Test Specification

TestLink organizes amount of Test Cases into one or more levels of Test Suites. These levels are persisted throughout the application. Keywords could be assign to a particular Test Cases to achieve the next level of Test Case organization.

2.1 Test Suite

Tester follows this structure: Test Specification -> Test Suites -> Test Case. At first you create Test Suite(s) for your Test project. You can fill a description. Test Suite can includes one or more level of itself.

2.2 Test Cases

Test Cases has next parts:

- **Title:** could include either short description or abbreviation (e.g. TL-USER-LOGIN)
- **Summary:** should be really short; just for overview.
- **Steps:** describe test scenario (input actions); can also include precondition and cleanup information here.
- **Expected results:** describe checkpoints and expected behaviour a tested Test project or system.

User can also copy or move Test Cases through a Test Specification.

2.2.1 Test Case Versions

The numeric ID of a test case is assigned automatically by Test Link, and can not be changed by users. This ID is system wide, means when a test case is created a global counter is used, independent of test project where test case is created.

The 'holes' you will see in the ID (i.e. Missing numbers) are due to implementation details.

2.2.2 Test Case - Active Attribute

If several versions of a Test Case can exist, it would be useful to have a new attribute Active/Inactive, to use in this way:

- Every Test Case version is created ACTIVE
- An Inactive Test Case Version is not possible to assign to any Test Plan (will not be available in "Add Test Cases to Test Plan"). This can be useful for Test Designers. They can edit or change the TC Version and only when he/she decide is completed => change Status to ACTIVE, will be available to be included an used in a Test Plan.
- Once a TC Version has been linked to a Test Plan, and have results, it can't be turned INACTIVE.

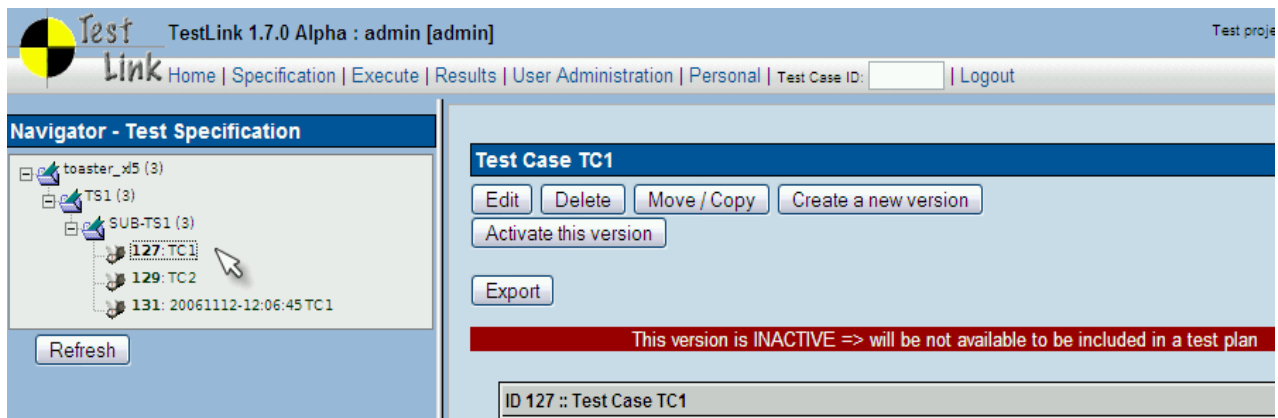


Illustration 2: Test Case menu

2.3 Test Case relation to Software Requirements

Test cases could be related with software/system requirements as n to n. The functionality must be enabled for a Test project. User can assign Test Cases and Requirements via link Assign Requirements in the main screen.

2.4 Keywords

Keywords were created to give users another level of depth when categorizing test cases. Keywords serve as a collection of Test cases with some attribute within a Test specification. You can use it to define e.g.

- Regression or Sanity set
- Reviewed Test cases
- Set of test cases valid for one platform

2.4.1 Keyword Creation

At this time keywords can only be created by users with the *mgt_modify_key* rights. These rights are currently held only by Leaders. Once a keyword or grouping of keywords have been created users may assign them to test cases.

2.4.2 Assigning Keywords

Keywords may be assigned to test cases either from the assign keyword screen (in batch) or via the test case management (individually).

2.4.3 Filter by Keyword

Users have the ability to filter by Keywords for:

- Search Test Cases in Test Specification.
- Add groups of Test cases in a Test case Suite (Test plan).
- Execute test screen.

3 Requirement based testing

3.1 Introduction

To prove that a system is built as specified, testers use requirement based testing. For every requirement, they design one or more test cases. At the end of the test execution a test manager reports on the tests that are executed and the requirements that are covered. Based on this information the client and the various stakeholders decide whether a system can be transferred to the next test phase or can go live. To ensure that a system is built as specified, test managers use a combination of risk and requirement-based testing to ensure that a system is built as specified from the customer and stakeholders perspective. As a result, this complete testing delivers the following advantages:

- Linking risks and requirements will reveal vague or missing requirements. This is especially interesting for risks with a high priority.
- Testing can be focused on the most important parts of an information system first: covering the risks with the highest priority.
- Communicating in the same language as the client and the stakeholders. This makes it easier to report on the status of the test project. Beside that a better founded decision can be made whether to invest more in testing or take the risk.
- The risks and their priority make negotiating on the test project in times of pressure easier. What risks have to be covered within this test project and which ones can be postponed. Risk and requirement-based testing results in a better controlled test project. The communication with the client and the stakeholders improved. The test manager begins testing with risks with the highest priority. The process is streamlined and the end result is higher quality.

3.2 Availability

The functionality is available on Test project level. I.e. Administrator should enable it for a specified Test project (Edit Test project link in Main window). Otherwise links are not shown.

There are two user levels for this feature. The most of roles can view requirement but not modify. Refer to User section for more.

3.3 Requirements Specification

Requirements are grouped to one or more System/Software/User Requirement Specifications.

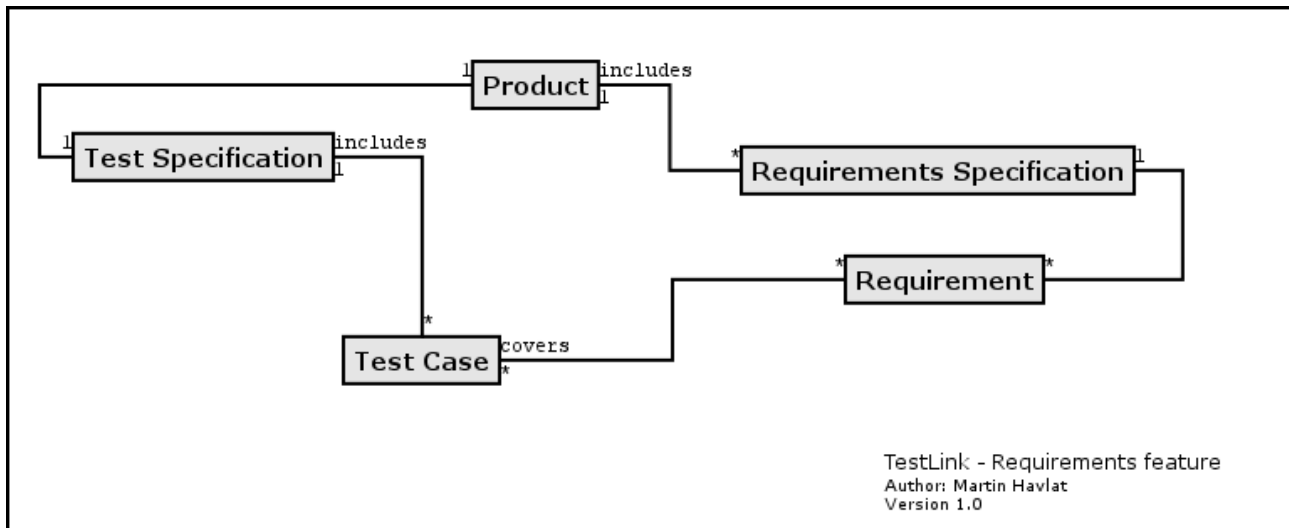


Illustration 3: Dependencies between requirement related objects

Create a document with Requirements:

1. Click Requirements Specification in Main window. The List of Requirement Specification window is shown.
2. Press Create button to create a document.
3. Adjust Title, Scope and eventually Count of Test cases. The last parameter is used for statistics. Use only if you have a valid Requirement document but not all requirements are available at the moment in TestLink. Default value 'n/a' means that the current count of requirements in a specification is used.
4. Press Create button to add data to database. You can see the title of your new created document in the table of List of Requirement Specification window.
5. Click the title of document for next work. The Requirement Specification window is shown.

Each Requirement Specification has own statistics and report related to included data.

All Specifications can be printed using the "Print" button in the "Requirement Specification" window. Administrator can define company, copyright and confident text via configuration files.

3.4 Requirements

Each requirement has Title, Scope (html format) and Status. *Title* must not be unique and has max. 100 characters. *Scope* paramter is text in HTML format. *Status* can have vale VALID or NOT_TESTABLE. A NOT_TESTABLE requirements are not counted to metrics.

Requirements could be created/modified or deleted manually via TestLink interface or imported as CSV file.

3.4.1 Import requirements

TestLink support two types of CSV. The first 'simple' is composed from title and scope in each row. The second 'Export from Doors' try to detect header and chooses correct fields. Import compare titles and allow to solve conflicts. There are three ways: update, create requirements with same title and skip adding the conflicted ones).

3.4.2 Requirements to Test Case relation

Test cases are related with software/system requirements as * to *. I.e. you can assign more

Test cases to one Requirement and more requirements could be covered by one Test Case. User can assign Requirements to Test Cases via the Assign Requirements link in the Main window.

A coverage of Test Specification could be view via pressing the button Analyse in the Requirement Specification window.

3.4.3 Requirement based Report

Navigate to Reports and Metrics menu. There is Requirements based Report link. Requirements in current Requirement Specification and Test Plan are analysed for this report. All the latest result of test cases (available in Test Plan) are proceeded for each requirement. The result with the highest priority is applied for the requirement. Priority from the highest are: Failed, Blocked, Not Run and Passed.

Example of requirement coverage

A requirement is covered by three Test Cases. Two of them are included in the current Test Suite. One passed and one was not tested for the Build 1. Now Requirement has overall result: Not Run. Second test case was tested with Build 2 and passed. So Requirement passed too.

4 Test projects

Test projects are the cornerstone of TestLink. Test projects are releases of your company that may change their features and functionality over time but for the most part remain the same. Test project includes requirements documentation, test specification, test plans¹ and keywords.

4.1 Creating new Test projects

Creating a new Test project requires "admin" rights. Each Test project must have a unique name. Background colors can be assigned to Test project templates to visually distinguish them.

Things to note when creating a new product:

- Deleting Test projects from the system is not recommended as this either orphans a large number of test cases or deletes the test cases from the system.
- Test plans represent the testing of a Test project at a certain point in time. Consequently, test plans are created from Test project test cases.
- TestLink supports importing CSV data into a Test project. This is explained further in the Import section, below.

4.2 Edit and delete Test projects

Edit Test projects requires "admin" rights. User can change Test project name, color of background and availability of requirements functionality.

User with the privileges also can **inactivate** the Test project if it's obsolete. This cause that the Test project is not visible in list within the top navigation bar (except admin who see such Test project in the list marked by '*').

User can also **delete** a Test project. This action delete also all related data from database. **This action is not reversible.** We strongly recommend to use inactivate instead of delete.

¹ Older version TL have independent Test projects and Test Plans. This is configurable for back compatibility.

5 Test Plans

Test plan contains name, description, collection a chosen test cases, builds, test results, milestones, tester assignment and priority definition.

5.1 Creating a new Test Plan

Test Plans may be deleted from the "Create test plan" page (link "Create Test Plan") by users with lead privileges.

Test plans are the basis for test case execution. Test plans are made up of test cases imported from Test projects at a specific point of time. Test plans can only be created by users with lead privileges. Test plans may be created from other test plans. This allows users to create test plans from test cases that at a desired point in time. This may be necessary when creating a test plan for a patch. In order for a user to see a test plan they must have the proper rights. Rights may be assigned (by leads) in the define User/Project Rights section. This is an important thing to remember when users tell you they can't see the project they are working on.

5.2 Builds

An user with lead privileges could follow the link "Build management" in the main page.

Builds are a specific release of software. Each project in a company is most likely made up of many different builds. In TestLink execution is made up of both builds and test cases. If there are no builds created for a project the execution screen will not allow you to execute. The metrics screen will also be completely blank.

Builds currently cannot be edited. A build could be deleted by click on the appropriate "bin" icon in the table of existing builds.

5.3 Deleting TestPlans

Test Plans may be deleted from the "Edit test plan" page (link "Edit / Delete Test Plan") by users with lead privileges. **Deleting Test Plans permanently deletes both the test plan and all of its corresponding data, including test cases, results, etc.** This should be reserved only for special cases. Alternatively, Test Plans may be deactivated on the same page, which supresses display on selection menus in the "main" page.

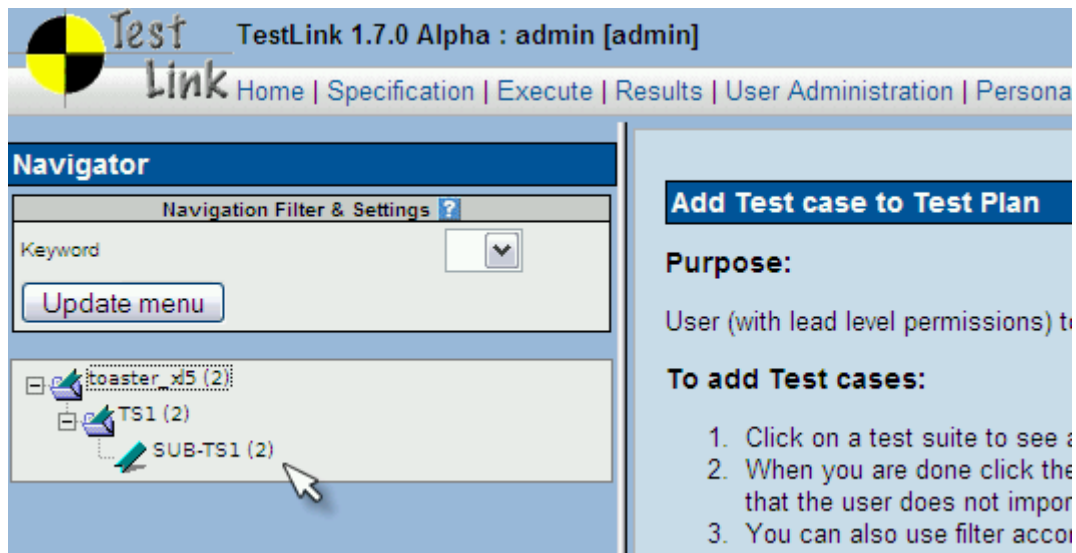
5.4 Testing Set

5.4.1 Assign Test Cases into Testing Set

Test Case Suite is set of test cases which are defined to be run within Test Plan. Test case Suite is created via Add Test Cases from Test Specification to Test Plan. Test cases are added including Steps and Expected result. So, you must use 'Update modified Test Cases' page to update test scenario (version of test case).

Data from multiple Test projects can be added into one test plan. Test Specification data can be filtered by keywords (adjusted in navigation pane).

Once data has been imported into a test plan it will be marked with checkmark. If a test case has already been imported it will be ignored if it is imported again.



As you can note, the number near test project name (in this example: **toaster_xl5**) is 2, but the test project has 3 test cases. Test case TC1 is not counted, because is inactive.

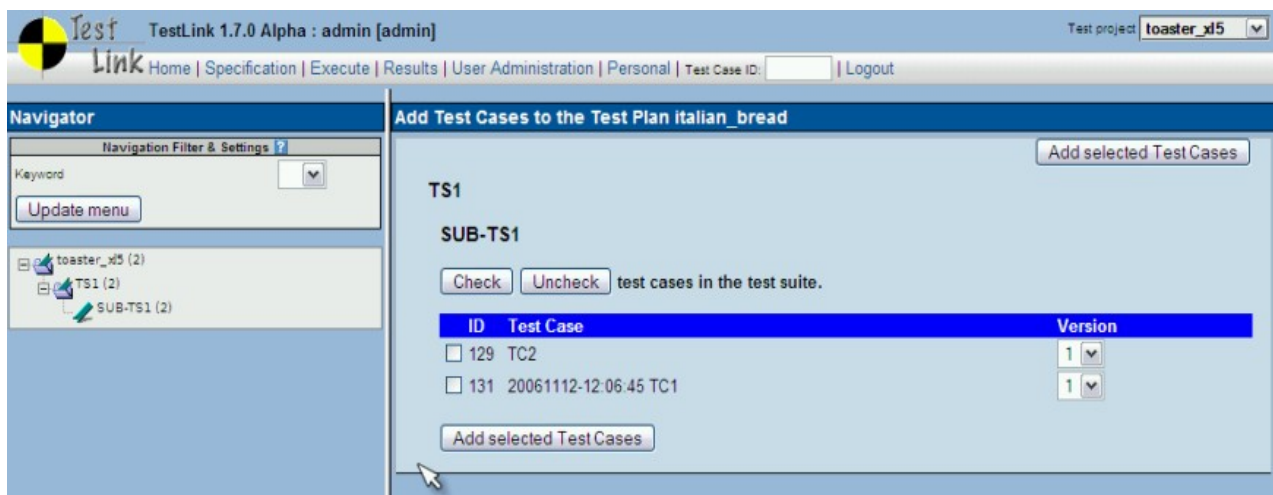
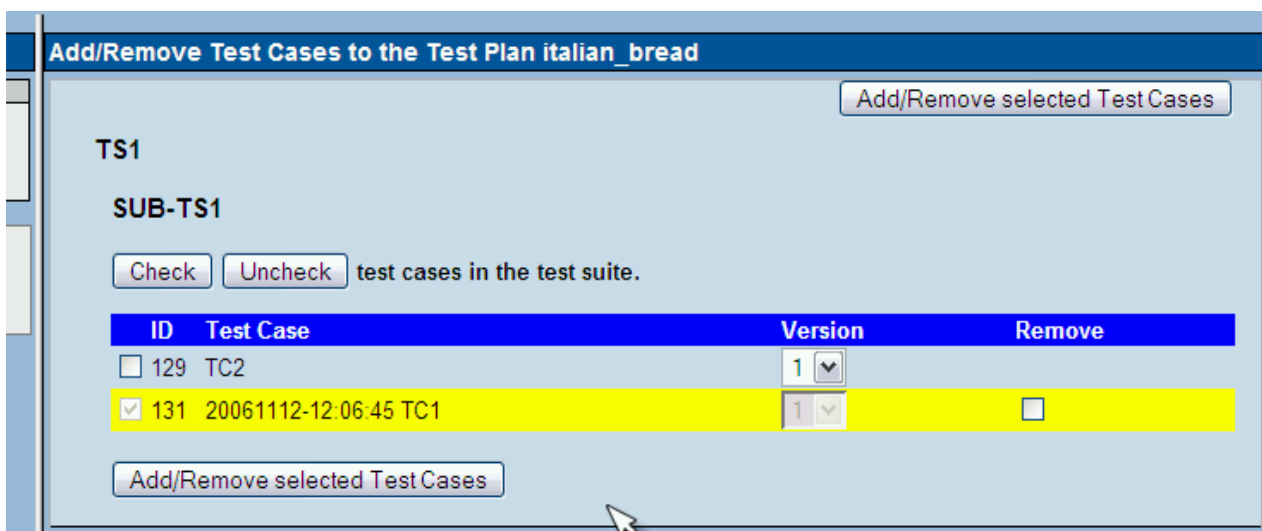


Illustration 4: Assign Testing Set of Test Plan



5.4.2 Removing Test Cases from Testing Set

Test Cases and Test Suites may be deleted from a test plan by users with Leader permissions from the "Remove test cases" page. Deleting data may be useful when first creating a test plan since there are no results. However, Deleting test cases will cause the loss of all results associated with them. Therefore, extreme caution is recommended when using this functionality.

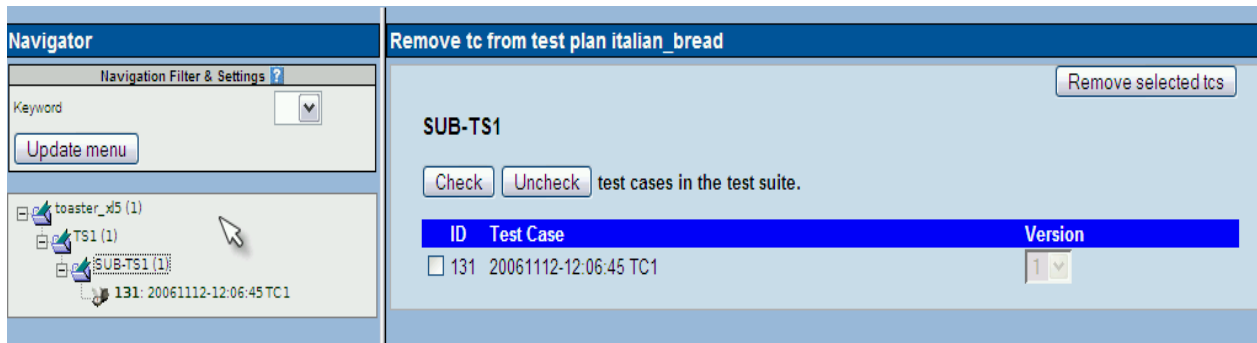


Illustration 5: Removing Test Cases

Another way to remove test case is using the menu option that allows only the remove operation. Tree on left pane, show only the test cases present in test plan.

5.5 Priority

TestLink gives users with Leader rights the ability to assign ownership and priority to test cases. General Risk/Ownership is done at the Category level. TestLink currently does not allow users to assign risk or ownership at the individual test case level.

Risk levels are low, medium, high and Importance levels are 3, 2, 1. Users can rank the combinations of risk and importance (L1, L2, L3, M1, H2, M3, H1, H2, H3) as priority A,B,C.

Assigning risk, importance, ownership, and priority are all optional and will default to priority B in the metrics screen.

5.6 Ownership

Ownership affects both the execution and metrics screens. In the execution screen users have the ability to sort the executable test cases by the ones they have ownership of. In the main metrics screen there is a table that shows the remaining test cases by ownership. If there are no test case owners it defaults to none.

A Tester can also see a metrics of own executed tests in main page if these metrics are enabled (see Installation manual).

6 Test Execution

6.1 General

Test execution is available when:

1. A Test Specification is written.
2. A Test Plan is created.
3. Test Case Suite (for the Test Plan) is defined.
4. A Build is created.
5. The Test plan is assigned to testers (otherwise they cannot navigate to this Test Plan).

Select a required Test Plan in main page and navigate to the 'Execute tests' link. Left pane serves for navigation in Test Case Suite via tree menu, filtering and define a tested build.

6.2 Navigation

The navigation pane consists from a 'Filter & Settings' box and a tree menu with Test Case Suite.

6.2.1 Filtering Test Cases

This table allows the user to filter test cases for smart navigation before they are executed.

- **owner:** Users can filter test cases by their owner. Ownership is determined at the Category level, is determined by leads, and can be changed at the Assign Risk and Ownership page under metrics.
- **Keyword:** Users can filter test cases by keyword. Keywords are set either using the Create/Edit/Delete Test Cases or by the Assign Keywords To Multiple Cases. Keywords can only be created, edited, or deleted by leads but may be assigned to test cases by testers.
- **Result:** Users can filter test cases by results. Results are what happened to that test case during a particular build. Test cases can pass, fail, be blocked, or not be run.

6.2.2 Define a tested build

Users can filter test cases by builds. Builds are the basic Component for how test cases are tracked. Each test case may be run once and only once per build. Builds can be created by leads using the Create New Build page.

6.2.3 Tree menu

The tree menu in navigation pane includes Test Case Suite colored by results.

Menu colored: By default the tree will be sorted by the results for [the defined build](#) that is chosen from the dropdown box.

Example TC colored according to the build

User selects build 2 from the dropdown box and doesn't check the "most current" check box. All test cases will be shown with their status from build 2. So, if test case 1 passed in build 2 it will be colored green.

Second possibility Last result is that menu is colored according to the latest test result.

Example TC colored according to the latest result

User selects build 2 from the dropdown box and this time checks the "most current" check box. All test cases will be shown with most current status. So, if test case 1 passed in build 3, even though the user has also selected build 2, it will be colored green.

6.3 Execution

6.3.1 Test Status

Execution is the process of assigning a result (pass, fail, blocked) to a test case for a specific build. 'Blocked' test case is not possible to test for some reason (e.g. a problem in configuration disallows to run a tested functionality).

6.3.2 Insert Test results

Test Results screen is shown via click on an appropriate Component, Category or test case in navigation pane. The title shows the current build and owner. The colored bar indicate status of the test case. Yellow box includes test scenario of the test case.

The indication that the test case was updated or deleted in test Specification is not supported in 1.5 version.

Updated Test Cases: TL 1.0.4 version has indication by flag, that is missing in 1.6 version. If users have the proper rights they can go to the "Update modified test case" page through the link on main page. It is not necessary for users to update test cases if there has been a change (newer version or deleted).

7 Test Reports and Metrics

The metrics pages sum up the results of execution into reports. Metrics are broken down by both individual builds and across all builds.

7.1 General Test Plan Metrics

This page shows you only the most current status of a test plan. For instance, you have test case 1 which was executed in builds 1,2, and 3. Build 1 2 3 Status Pass Fail Blocked Since the most recent result of the test case is blocked the result on the "Across All Builds" page would be blocked. If a user would go and change the status of build 3 to something else or not run the current result would be fail.

(in TL 1.0.4: View Project Status Across All Builds)

7.2 Metrics of active Build

This report shows the detailed results for a particular build defined in navigation pane.

(in TL 1.0.4: View Status by an Individual Build)

7.3 The Overall Build Status

View The Overall Build Status This report show a high level view of each build's result.

7.4 Query Metrics

Query results for specific test cases based on keyword, Component, owner, last result, and 1 or more builds. For example if you wanted to know if a test case has failed, passed, blocked, or has not been run in builds 2, 3, and 4 you would want to use this query functionality. Additionally you could determine if test cases assigned to a tester have been executed in a set of builds. This is important because test passes may occur over multiple builds, and multiple test passes may occur during the development cycle. This is especially handy at the end of a development cycle when you want to know if certain cases have been run over the past few builds of the Test project. This functionality differs from other reports which display results on a specific build, or all builds.

Export to MS Excel is also available.

7.5 Test Report

View Status By Individual Test Cases This report shows each test case's result for every build. An user can navigate to Test Execution screen via link for each test Status.

Export to MS Excel is also available.

7.6 Blocked/Failed Test Cases

These reports show all of the currently blocked or failing test cases.

7.7 Total Bugs For Each Test Case

This report shows each test case with all of the bugs filed against it for the entire project. This metrics are available only if Bug Tracking System is connected.

7.8 E-mail Test report

Email Test Plan Info This page allows users to email the results of the entire test Plan or for a particular build. It also allows users to email the status of a Component

8 User Administration

8.1 Account settings

Every user on the system will also be able to edit their own information via the Account settings window (link Personal in menu bar).

TestLink allows users with administrator rights to create, edit, and delete users within the system. However, TestLink does not allow administrators to view or edit user's passwords. If users forget their passwords there is link on the login screen, that will mail the user their password based upon their user name and the email address they entered.

8.2 Role Permissions

TestLink is built with 6 different default permission levels built in. Changing of these rights is handled by the user administration link which is accessible by admins. These permission levels are as follows:

- Guest: A guest only has permission to view test cases and project metrics.
- Test Executor: A tester outside of the company that only has permissions to run tests allotted to them. (initially in 1.0.4 – otester).
- Test Designer: A user can fully work with Test Specification and Requirements.
- Test Analyst: A tester can view, create, edit, and delete test cases as well as execute them. Testers lack the permissions to manage test plans, manage Test projects, create milestones, or assign rights. (initially tester, senior tester).
- Test Leader: A lead has all of the same permissions as a Tester but also gains the ability to manage test plans, assign rights, create milestones, and manage keywords.
- Administrator: An admin has all of the same permissions as a lead but gains the ability to manage Test projects.

Note: Test plan related features needs also assign a Test Plan to be available. See [Test Plan Assignment](#).

8.2.1 User Roles

There are predefined user roles. Administrator gives appropriate ability to modify data within TestLink. Each user has assigned just one of these roles.

If you view the table you will see rows for each of the permissions levels (guest ,tester, senior tester, leader, admin). The column next to the row holds all of the different rights levels which will be defined below. These levels have been determined as standard for the use but they are free to be edited or define a new roles (for experienced administrator). The user table contains a foreign key that points to the appropriate permission level in the rights table.

Role	List of Rights	Permissions
Guest	mgt_view_tc, mgt_view_key, tp_metrics	Browse data only.
Test Executor (Tester)	tp_execute, tp_metrics	Execute test only.
Test Analyst (Senior Tester)	tp_execute, tp_metrics, tp_create_build, mgt_view_tc, mgt_modify_tc, mgt_view_key, mgt_view_req	Edit Test Specification, execute tests, create build.
Test Designer	tp_metrics, mgt_view_tc, mgt_modify_tc, mgt_view_key, mgt_modify_req, mgt_view_req	Edit Test Specification and Requirements.
Test Leader	tp_execute, tp_create_build, tp_metrics, tp_planning, tp_assign_rights, mgt_view_tc, mgt_modify_tc, mgt_view_key, mgt_modify_key, mgt_view_req, mgt_modify_req	All Test Plan permissions, edit Test Specification and execute tests.
Admin	tp_execute, tp_create_build, tp_metrics, tp_planning, tp_assign_rights, mgt_view_tc, mgt_modify_tc, mgt_view_key, mgt_modify_key, mgt_view_req, mgt_modify_req, mgt_modify_product, mgt_users	Everything possible. Only this role can maintain Test projects and users.

Table 1: Role description

8.2.2 Rights Definitions

Next tables list keywords used for definition of role abilities.

Right	Description
mgt_view_tc	Viewing Test Specification (data of Component, Category, and test case)
mgt_modify_tc	Edit Test Specification (create, modify, delete, order, move, and copy Components, Categories, and test cases)
mgt_view_key	Viewing keywords
mgt_modify_key	Modifying keywords
mgt_modify_product	Create, edit and delete Test projects
mgt_view_req	View requirements
mgt_modify_req	Create, edit, associate and delete requirements

Table 2: Test project related Rights

Right	Description
tp_execute	Ability to execute test cases (insert test results)
tp_create_build	Ability to create builds
tp_metrics	Viewing metrics
tp_planning	create, edit, delete Test Plans, assign risk/ownership, milestones, edit Test Case Suite
tp_assign_rights	Assigning the rights to view projects

Table 3: Test Plan related Rights

8.3 Test Plan Assignment

Users can see only assigned Test Plans. In order to gain test plan permissions a user with lead or admin status must give them rights through the "Define user/project rights" link under "Test Plan Management".

All users in the system will by default not have permissions to view newly created test plans (except for the test plan creator who can give themselves permissions at creation). Zero test plan permissions means that users will not see any Test Plans in the Test Plan dropdown box on main screen.

There is a table with Test Plan rights (i.e. which users can see which Test plan). This table is made up of a combined user id and project id. The main page contains code which checks to see if the logged in user has the appropriate permissions (and then shows the allowed projects. It is not recommended that this be hacked with.

Revision History:

#	Description	Date	Author
0.x	Documents for TL 1.5 and update for TL 1.6	2005	M. Havlat A. Morsing F. Mancardi
1.0	Converted to OO2 format;	2005/03/12	M. Havlat
1.1	Minor update; FIX 372, 352	2006/02/14	M. Havlat
1.2	Updated as draft for TL 1.7	17/11/06	M. Havlát