Telerik Academy Learning System

Release 5.7

Master Test Plan

Version 1.2

**Draft**

December 2015

Submitted by

Team Giant

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision # | Revision Date | Description of Change | Author |
| 1.0 | 11/27/2015 | First Draft | Team Giant |
| 1.1 | 4/12/2015 | Fixed risks, metrics test types and levels | Team Giant |
| 1.2 | 11/12/2015 | Revised Introduction, test Strategy,  Updated risks, metrics, test types and levels, tools | Team Giant |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Test Identifier 3](#_Toc437595491)

[2. Introduction 4](#_Toc437595492)

[2.1. Scope 4](#_Toc437595493)

[2.2. Test Objectives 4](#_Toc437595494)

[3. Features to be Tested 4](#_Toc437595495)

[3.1. Admin panel – Courses and Lectures 4](#_Toc437595496)

[3.2. Front End 5](#_Toc437595497)

[4. Risk Analysis 5](#_Toc437595498)

[5. Test Strategy 6](#_Toc437595499)

[5.1. Test Approach 6](#_Toc437595500)

[5.2. Test level and types 6](#_Toc437595501)

[*5.2.1.* *Functional Testing* 6](#_Toc437595502)

[*5.2.2.* *Non-functional Testing* 7](#_Toc437595503)

[*5.2.3.* *Regression Testing* 7](#_Toc437595504)

[5.3. Measurement and Metrics 8](#_Toc437595505)

[5.4. Entry/Exit Criteria 9](#_Toc437595506)

[*5.4.1.* *Entry Criteria* 9](#_Toc437595507)

[*5.4.2.* *Exit Criteria* 9](#_Toc437595508)

[6. Test Deliverables 9](#_Toc437595509)

[7. Test Environment 9](#_Toc437595510)

[7.1. Client-side Environment 9](#_Toc437595511)

[7.2. Server-side Environment 9](#_Toc437595512)

[8. Test Tools 10](#_Toc437595513)

[9. Test Schedule and Responsibilities 10](#_Toc437595514)

[10. Approvals 11](#_Toc437595515)

[11. Glossary 11](#_Toc437595516)

# Test Identifier

This Master Test Plan has the ID:

TALS v5.7(build20151119.44a964b)-MTPv1.2

# Introduction

## Scope

This document details the activities that will be performed by team Giant for testing Telerik Academy Learning System (TALS), web site located at <http://stage.telerikacademy.com/>. Our team is assigned to test Admin panel GUI, section “Courses and Lectures” and navigation and interaction between them in End User GUI.

It defines the overall testing requirements and provides an integrated view of the project test activities. Its purpose is to determinate:

* What will be tested
* How testing will be performed
* What resources will be used

This document will be updated for each build.

## Test Objectives

This test plan for testing functionality of TALS supports the following objectives:

* Details the activities required to prepare and support the test
* Communicate between all responsible persons, the tasks which they are assigned to complete and the schedule to be followed in executing the tasks.
* Define the sources of information used to prepare the test plan
* Define the test tools and environment needed to conduct the test

# Features to be Tested

Test scope for TALS contains two different parts of whole system Admin Panel and End User GUIs. For each of them will be performed tests for part of their features:

## Admin panel – Courses and Lectures

* Courses and Lectures - [Курсове и лекции](http://stage.telerikacademy.com/Administration_Courses/CoursesInstancesAndLectures)
* Course Categories - [Категории курсове](http://stage.telerikacademy.com/Administration_Courses/CourseInstanceCategories)
* Category Hierarchy - [Категории - йерархия](http://stage.telerikacademy.com/Administration_Courses/CourseInstanceCategoriesHierarchy)
* Students in Courses - [Студенти в курсове](http://stage.telerikacademy.com/Administration_Courses/UsersInCourses)
* Groups - [Групи](http://stage.telerikacademy.com/Administration_Courses/CoursesGroups)
* Homework - [Домашно](http://stage.telerikacademy.com/Administration_Courses/Homework)
* Instances - [Курсове, инстанции и лекции](http://stage.telerikacademy.com/Administration_Courses/CoursesAndLectures)
* Single Courses (Kids Academy) - [Единични курсове (Детска академия)](http://stage.telerikacademy.com/Administration_Courses/CoursesInstancesAndLectures/CourseInstancesWithoutCourse)
* Licenses - [Лицензи](http://stage.telerikacademy.com/Administration_Courses/CourseLicenses)
* Students Statistics - [Статистики за студентите](http://stage.telerikacademy.com/Administration_Courses/CourseInstancesStatistics)
* Students Statistics (by age) - [Статистика за студентите по години](http://stage.telerikacademy.com/Administration_Courses/CourseInstancesYearsStatistics)
* Course's Surveys - [Анкети за курсове](http://stage.telerikacademy.com/Administration_Courses/Polls)
* Statistics for course's surveys - [Статистика анкети за курсове](http://stage.telerikacademy.com/Administration_Courses/PollsStatistics)
* Course’s unenrolled students - [Отписани студенти от курсове](http://stage.telerikacademy.com/Administration_Courses/UnenrolledUsersInCourses)

## Front End

* Navigation between different courses
* My Courses
* Archive

# Risk Analysis

Our risk analysis will contain only project risks, because TALS is already in use and for product risks we need requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Likelihood | Impact | Mitigation plan | Owner |
| RESOURSES | | | | |
| Web testing software is not available/does not work. | Low | High | Prepare spare web testing software and be sure that there is a way for easy migration between main and spare resources. |  |
| DEFECTS | | | | |
| In late stage of testing is found critical defect, which need much time to be fixed. | Medium | High | Defect management plan is in place to ensure prompt communication and fixing of issues. |  |
| Defects discovered are many more than expected. | Low | Medium |  |
| TEAM | | | | |
| Team has poor communication. | Medium | High | Strictly defined scrum meetings |  |
| Team member lack of necessity skills | Medium | Medium | Check at the beginning of each sprint if all team members have base knowledge for current tasks |  |
| Team member get sick and need time to recover. | Medium | High | All team members are agreed work additional hours to cover tasks of sick team member |  |
| ENVIROMENT | | | | |
| Web site becomes unavailable for a period of time. | Low | High | Build back up instance of web site if the main go down. |  |
| Non-availability of Independent Client- side test environment and accessibility. | Medium | High | Prepare all hardware and software need for environment at the beginning and be sure they are well maintained during project. |  |

# Test Strategy

Because we are lack of requirements we will perform risk-based strategy for our test project. We will prioritize all TALS functionality and choose top-down approach to test them. Prioritization of stories and test cases will be done according:

* Frequency of usage
* Importance
* Risk of failure
* Impact

Test cases and stories will have four level of priority in our test case management system.

## Test Approach

In abstract perspective our test project is very specific, so for realizing it we need specific approach. First we do not have raw code for system, so we will perform only black-box testing approach for it functionality. Second TALS is already functioning system, so our approach will be perfectly reactive.

## Test level and types

System test level will be implement for TALS, as a fully build system. Acceptance tests are not planed as they are not required.

### *Functional Testing*

Testing of the application will focus on expected behavior of the modules of testing system and their interaction based on our judgment and experience as users of similar system. The goals of these tests are:

* to verify data acceptance, processing, and retrieval.
* to ensure that the User Interface for Admin Panel and End user provides them with felicitous access and navigation through rich TALS (APPLICATION) functionality
* Accuracy

This type of testing is based upon black box techniques, that is, verifying the application (and its internal processes) by interacting with the application via the GUI and analyzing the output results. Identified below is an outline of the testing recommended for each system module.

**Test Objective**:

* Ensure correct application navigation, data entry, processing and retrieval.
* Ensure receiving accurate data from statistic modules

**Technique:**

Execute each use case, use case flow, or function, using valid and invalid data, to verify the following:

* The expected results occur when valid data is used.
* The appropriate error / warning messages are displayed when invalid data is used.
* Each link navigates to correct window

**Completion Criteria**:

* All high priority tests successfully executed.
* All identified defects have been addressed.

### *Non-functional Testing*

Testing of the application will focus on expected end user behavior based on our knowledge and logical conclusions. The goals of these tests are:

* Compatibility
* Performance Testing
* Load Testing

### *Regression Testing*

Testing the application will be executed after each new build is code freeze. If time is running out, only smoke testing will be applied for regression.

**Completion Criteria:**

There are very few known medium or low-priority defects that don't affect the usage of the product.

## Measurement and Metrics

Basic metrics will be kept for test effort (i.e. hours), test cases executed, incidents happens and defect reported. Metrics to evaluate:

* Number and severity of defect found
* Number of test cases designed and executed given by priority.
* Number of automated tests.
* Percentage of total application functionality covered by tests.

***Definition:***

**Defect** is a flaw in a system that can cause it to fail to perform its required function. Defect will be grouped by their severity.

***Severity levels:***

* **Critical:** The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.
* **High**: The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.
* **Medium:** The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.
* **Low**: The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

***Test case priority levels:***

* **Priority-1**: Allocated to the tests, which must pass, otherwise the delivery date will be affected.
* **Priority-2:** Allocated to the tests, which must be executed before the final delivery.
* **Priority-3**: Allocated to the tests which can be executed, only when time permits.
* **Priority-4**: Allocated to the tests, which can be executed even after the delivery date or have remote probability of execution ever.

## Entry/Exit Criteria

### *Entry Criteria*

* The testing environment should function properly
* All of the testing tools, which are going to be used, should be available
* All of the team members should be familiar with all of the testing tools used.
* Telerik Academy Learning System should be code freeze.

### *Exit Criteria*

* 100% of test cases executed
* 100% of critical severity faults fixed
* 80% of low & medium severity faults fixed
* maximum of 10 medium severity known faults remain
* maximum of 40 low severity known faults remain
* time has run out

# Test Deliverables

The following documents will be generated as a result of these testing activities:

* Master test plan (MTP - this document)
* Combination incident/test summary reports for each build
* Test log for each build
* Automated tests and supporting test data

# Test Environment

There are essentially two parts to the Telerik Academy Learning System Web APPLICATION in production:

## Client-side Environment

* ***OS***: Windows 7, 8.1, 10
* ***Browser***: Internet Explorer 8, 9, 10, Mozilla Firefox 27.0, Google Chrome 32.0, Opera 27 and later should be available to the tester**.**

## Server-side Environment

* ***OS***: Windows Server 2012 R2
* ***Web Server***: MS IIS 8.5
* ***SQL Server***: MS SQL Server 2012
* ***CPU:*** Intel Xeon CPU @ 3.10 GHz (Quad Core)
* ***RAM***: 8GB
* ***Storage***: 100GB

# Test Tools

The testing will use a combination of manual and automated tests.

* Test automation - Test Studio
* Test case management system– Microsoft Test Manager
* Source control - GitHub
* Bug tracking tool - TeamPulse
* Link validation - LinkChecker
* Performance – Visual Studio

# Test Schedule and Responsibilities

For the project schedule we will use form of our academy schedule:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint | Features to be done |  | Roles and Responsibility | | |
| Team Leader | | Scrum Master | Product Owner |
| Week 47 | Organize a Party | Dushka Dragoeva | | Blago Shockov | Ventsi Ivanov |
| Week 48 | Prepare a Test plan for Telerik Academy Learning System | Biser Hristov | | Ventsislav Ivanov | Asia Georgieva |
| Week 49 | Prepare Test Cases, Update the Test Plan | Blagoy Shokov | | Dushka Dragoeva | Asia Georgieva |
| Week 50 | Prepare a automation test with Test Studio | Ventsislav Ivanov | | Plamen Kostadinov | Asia Georgieva |
| Week 51 |  | Dushka Dragoeva | | Biser Hristov | Asia Georgieva |
| Week 52 | Christmas Holiday |  | |  |  |
| Week 1 | Christmas Holiday |  | |  |  |
| Week 2 |  | Plamen Kostadinov | | Blagoy Shokov | Asia Georgieva |
| Week 3 |  | Biser Hristov | | Ventsislav Ivanov | Asia Georgieva |
| Week 4 |  | Blagoy Shokov | | Dushka Dragoeva | Asia Georgieva |
| Week 5 |  | Ventsislav Ivanov | | Plamen Kostadinov | Asia Georgieva |
| Week 6 | Final Release | Dushka Dragoeva | | Biser Hristov | Asia Georgieva |

Whole team members have responsibilities to present on these team meetings:

* + Scrum Planning Meeting – Every Friday -20.30 o’clock, live
  + Daily Scrum Meetings – every Evening at 20.00 o’clock
* Monday, Wednesday – live at Telerik
* Tuesday, Thursday, Saturday and Sunday – on-line, via Skype
  + Scrum Sprint Review Meeting – Friday 15.00 o’clock
  + Scrum Retrospective Meeting - Every Friday -20.00 o’clock, live
  + Brainstorming – any time it necessary via Skype

During these meetings all tasks and obscurity problems issues will be resolved and assigned

# Approvals

Telerik QA trainers

Product Owner must approve this plan

# Glossary

GUI - Graphical user interface,

TALS - Telerik Academy Learning System

# Team Conventions

Team cases naming – [section name] test case name.