

Design and implementation of test record management based on ALM

Jiao Li, Yuan Yang, Zhangbiao Fan

Yunnan Open University, Kunming 650500, China

Abstract: Test records are detailed records of the execution process of test cases. Effective management of test records can facilitate rapid tracking and tracing of the test execution and defects of each case. This paper introduces the process of optimizing test execution record management based on ALM innovation to achieve dual tracking control of test record “online” + “offline”, effectively reducing the execution time of testers, improving the timeliness of developers to locate and solve problems, and facilitating managers to quickly and efficiently trace the test execution process.

Key words: Test management system ALM; Test cases; Test case execution; Test record sheet; Defects

Introduction

In the process of software testing, the test record is the effective data to prove the execution process of the tester. Through the test record, the test execution information can be effectively viewed, and it is also an important basis for the test repetition, development and positioning to find problems. However, in the past testing process, there was a lack of effective and complete record of the test execution process, which could not effectively and quickly track the test execution. The test records submitted by the tester are in different forms and diverse contents. Only one test screenshot, TXT file, or Excel file is delivered, and the test execution record cannot be effectively associated with the test cases and defects. In view of the problems encountered in the above testing process, this paper introduces the secondary development of test execution records based on HP ALM (Application Lifecycle Management) test management system to achieve unified management of test execution records.

1. Test record table design idea

Based on the problems of different forms and non-standard test records in the testing process, we first designed a standardized and unified “test record form” template to collect various data types needed to record in the testing process.

The test record form contains the following test elements:

- (1) Project name: The name of the project
- (2) Test round: the stage round to which the current test belongs
- (3) Tester: Information about the person performing the test
- (4) Test time: the execution time of the test
- (5) Use case number: the number compiled by designing test cases to distinguish test cases
- (6) Test data: test data used in the test process
- (7) Preconditions: test conditions required before the test is executed
- (8) Detailed steps and screenshots: name of the test steps, description of the test steps, expected test results and screenshots
- (9) Description of test results: Whether it is consistent with the expected results according to the execution of the test
- (10) Defect number: the defect number found during the execution of the test case
- (11) Defect description: The description of the defect found
- (12) Processing records: records and remarks of developers, business personnel, testers and other relevant stakeholders dealing with the defect
- (13) Processing person: the information of the person who handled the defect
- (14) Processing time: the latest time when the defect was processed
- (15) Verification record: After the defect is repaired, the tester shall verify the process record and screenshot
- (16) Verification result: verify whether the result after the defect repair passes
- (17) Validator: the information of the person verifying the defect
- (18) Verification time: the time to verify the defect

2. Test record management design based on ALM

The whole function of test record management based on ALM includes the back-end Web application framework based on ASP.NET, the use of C# language to develop WEB applications, the establishment of IIS to provide services and the call of ALM end of the test management system.

- (1) Test record table automatically associated with test case data

By dynamically configuring HP ALM database information, the IIS back-end obtains test cases and defect data from the ALM Oracle database, sets Bookmarks for the test record table Word template document, and writes the corresponding test information into the test record table Word document. Automatically read the ALM database project corresponding to the project name, test rounds, testers, test time,

use case number, use case name, preconditions, detailed steps and expected result data, and insert into the corresponding fields, testers do not need to manually write the corresponding relevant data, greatly reducing the execution time of testers.

(2) Automatic generation of the test record table

In order to ensure that testers use a unified test record template and effectively shorten the time for testers to write test records, ALM system exports test record Form in batches, automatically packages it into a zip package and returns it, and deploys the WEB application through the IIS server for the use of the test management system. Test management system ALM side through the custom workflow development control, and the test instance, defect development event code call back-end service, to achieve the automatic generation of test record table and batch download function, testers do not need to manually create “test record table”, convenient testers according to each test instance report for offline testing, saving the tester execution time. Effectively improve the efficiency of testers.

(3) Test record form batch upload

After the tester completes the test execution, update the status of the test record table file name. In the test lab, you can specify the file directory and upload all files to the test lab in batches. According to the file name of the uploaded file, the system will automatically parse out the test instance ID and find the corresponding test instance according to the test ID. And the test record table file as an attachment to the test instance, if the uploaded file already exists, it will automatically overwrite the latest file. The upload operation is simple and convenient, and the tester does not need to manually upload each test record form, which greatly reduces the tester time, improves the test efficiency, and allows the tester to spend more time on other more meaningful work such as test case design.

(4) Using the same test record form to automatically generate defects

1. Fill in defect description field information automatically

When the tester creates a new defect in ALM, in order to save the time for the tester to submit the defect, standardize and unify the defect submission information, the secondary development realizes the system to automatically fill in the required content of the defect description information, including “problem description”, “test environment”, “test data”, “preconditions”, “detailed steps”, “expected result” and “actual result”. When submitting defects, testers only need to fill in the corresponding defect information, there is no need to manually enter the detailed steps of the field set in the corresponding format, standardize the defect description information, and shorten the time for testers to write the defect description information.

2. Automatically generate defect reports using the same test log sheet

In the past software testing process, the test execution record only records the evidence of successful test execution, and the defects found in the test process are recorded separately. As a result, multiple documents need to be searched to check and track the execution and defects in the complete life cycle of a test case, which is cumbersome and time-consuming, and it is impossible to effectively manage the execution of a case. Through the Test Record Form, the execution and defects of the same test case are recorded in a unified and orderly manner. One document can check whether the test case has defects, whether the defects are repaired, and the defect verification status, etc., which is convenient for testers and managers to control the execution of test cases online and offline. After the developer has repaired the defect, the tester can directly download the “Test Record Form” in the ALM defect, the system will read the latest test record form attachment information, and automatically read the defect information submitted in the system ALM. Including “defect number”, “defect name”, “defect description information”, “defect cause”, defect “system”, “planned repair time”, “actual repair time”, “Note”, “processor” and “processing time” inserted into the latest test record table, the tester does not need to manually write the defect related processing data during the process of verifying the defect. Only need to attach verification screenshots and verification result information, saving test manpower.

3. Test record management based on ALM

(1) Automatic generation of test record table

1. The IIS back-end uses C# development to read the Oracle database of ALM, obtain test case data, insert specific object Bookmarks into the Word template file, write the corresponding information read into the Word document, and automatically package into a zip package return.

2. Add batch import test record table through script editor in ALM front-end custom process.

Develop code in the script editor Function Template_TestLab_ActionCanExecute(ActionName) function to implement the call to the back-end service. Part of the core code is as follows:

```
projectUID = ""
```

```
for each project in TDConnection.GetAllVisibleProjectDescriptors
```

```
    if TDConnection.DomainName = project.DomainName and TDConnection.ProjectName = project.Name then
```

```
        projectUID = project.UID
```

```
    end if
```

```
next
```

```
CreateObject (" WScript. Shell ".) Run "http://127.0.0.1/Report/AutoCaseReport.aspx? t=auto&puid=" & projectUID & "&ts=" & testSetID
```

(2) Test record sheets are uploaded in batches

1. In the test management system ALM custom workflow, through the script editor, add batch import test record table control.

2. In the script editor, develop the event handling script in the function: Template_TestLab_ActionCanExecute to realize the automatic batch upload of test record table to the corresponding test instance in the test lab. Part of the core code is as follows:

```

Set tsf = TDConnection.TSTestFactory
Set tFilter = tsf.Filter
tFilter.Filter("TC_CYCLE_ID") = testSetID
Set lst = tFilter.NewList()
Set fs = CreateObject("Scripting.FileSystemObject")
Set folders = fs.GetFolder(objPath)
For Each file In folders.Files
    Set tstID = Split(file.Name, "_")(0)
    For Each aTest In lst
        If aTest.ID = Split(file.Name, "_")(0) Then
            AddAttachmentOnAlm atest, file
        End If
    Next
Next

```

(C) Use the same test record sheet to automatically generate defects

1. Auto-fill defect description field implementation

In the test management system ALM custom workflow, through the script editor, develop code to automatically fill in the defect description field for new defects.

2. Automatically generate defect reports using the same test log sheet

The IIS back end uses C# development to read the ALM Oracle database, obtain the defect data, insert Bookmarks into the test record table template file, and write the corresponding defect information read into the latest submitted test record table.

4. Summary

Innovate and optimize the test execution record management process, fully tap the test management system ALM, and realize dual tracking control of "online" + "offline" in secondary development, effectively reduce the execution time of testers, improve the timing of developers to locate and solve problems, and facilitate managers to quickly and efficiently trace the test execution process. This topic comes from the Web Front-end Development Research Center of Yunnan Open University, and discusses the feasibility of achieving compliance, efficiency and authenticity through standardized process, end-to-end traceability, audit trail and security measures.

References:

- [1] Yong Wang,Zhenzhen Lu. The Application of HP ALM Tool in Software Test Management [J]. Electronic Technology and Software Engineering.2016(06)
- [2] Feifei Wang,Jiatao Shi,Meng Wang,Xiaofei He. Embedded Software Platform Management Scheme Based on ALM system [J]. Electronic Technology and Software Engineering.2022(14)