

Guideline of Test Suite Construction for GUI Software Centered on Grey-Box Approach

Mengqing TanLi¹, Jiyi Xiao¹, Ying Zhang²

¹School of Software, University of South China, Hengyang, China

²School of Mechanical Engineering, University of South China, Hengyang, China

Email: TLMQ-TEAM@163.com

How to cite this paper: TanLi, M.Q., Xiao, J.Y. and Zhang, Y. (2023) Guideline of Test Suite Construction for GUI Software Centered on Grey-Box Approach. *Journal of Software Engineering and Applications*, 16, 113-143.

<https://doi.org/10.4236/jsea.2023.165007>

Received: March 30, 2023

Accepted: May 28, 2023

Published: May 31, 2023

Copyright © 2023 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, the test suite construction for GUI (Graphical User Interface) software may be executed centered on grey-box approach with the prior test design of window access controls for unit testing, including front-end method of white box and follow-up black box method for integration testing. Moreover, two key opinions are proposed for the test suite construction for GUI software, the first one is that the “Triple-step method” should be used for unit testing with the prior disposing of data boundary value testing of input controls, and another one is that the “Grey-box approach” should be applied in integration testing for GUI software with necessary testing preparation in the precondition. At the same time, the testing of baseline version and the incremental testing should be considered for the test case construction to coordinate with the whole evolution of software product today. Additionally, all our opinion and thought are verified and tested with a typical case of GUI software—PQMS (Product Quality Monitoring Software/System), and results indicate that these methods and specific disposing are practical and effective.

Keywords

Test Suite Construction, GUI Software, Triple-Step Method, Grey-Box Approach, Guideline

1. Introduction

GUI (Graphical User Interface) software has become the majority of current software and played a dominant role for mobile and desktop software in update engineering system [1], and the GUI software has covered a wide area for engineering application as shown in **Figure 1**. Further, we believe that this trend will be sustainable for a long period in the future. However, the testing of GUI

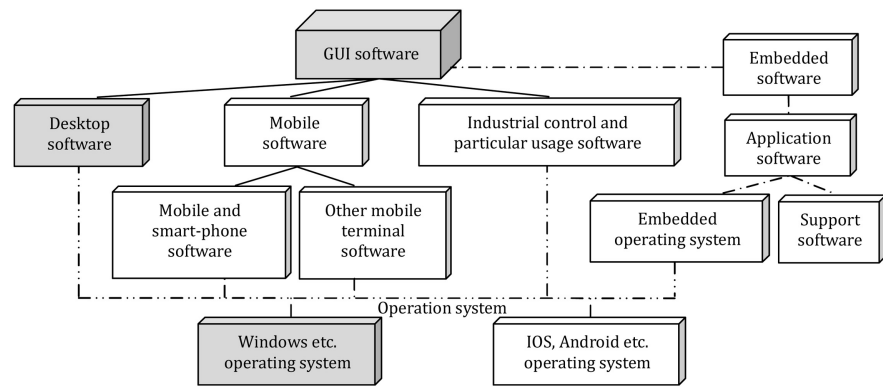


Figure 1. Popular composition of GUI software.

software has also faced some difficulties especially for the large scale software system with many communication interfaces and data scenarios. Of course, the difficulty of test suite construction for GUI software is the main aspect including the problem of how to improve the testing efficiency and how to assure testing quality. Generally, for the former, test engineer must consider how to do software testing with less time, which the grey-box approach has been proposed for GUI software testing in our study and testing practice [2], and for another problem, it will refer to accurately testing and BUG location for GUI software, which FTA (Fault Tree Analysis) has used in regression testing to deal with the software change [3]. However, concerning the core problem with test suite construction, this study will mainly focus on the grey-box approach which applied in GUI software testing.

This guideline is prepared for software companies and relative application areas, and it is a technical guidance which its thought and essence is scientifically summarized from testing practice and it also is derived from software programming technique in fact [4] [5] [6], in which there is not a bit of practical joke and baloney. As a result, this guideline will be illustrated and verified by a set of factual software testing examples [7]. However, this guideline must be fine-tuned in terms of the actual situation of software companies and relative application areas to assure the effectiveness of method application [8].

2. Related Literature and Work

Emil Alegroth, Robert Feldt [8] imply that test suite construction is done all through the process of software testing in the case company—Spotify, but the importance of testing data is ignored or omitted in their observation. Ron Patton in his writings “Software Testing” discussed the specific testing of GUI by using one chapter [5], and proposed seven aspects about it including 1) directly perceiving through the senses, 2) consistency, 3) nimbleness, 4) comfort, 5) correctness, and 6) usefulness, however the systemic approach for GUI software testing is lack especially for specific testing phase e.g. unit testing and integration testing. Fu Bing [6] puts forward that the test case constructing is the core matter of software testing as an effective measure for improving controllability and

software should be checked and tested with test case according to the requirement from user. At the same time, the difficulty of software testing for GUI was conscious in the writings of Fu Bing, and Fu Bing *et al.* thought that the study of GUI testing is in the initial phase at present and existed technology cannot assure the quality of factual GUI software. Additionally, the magnitude of test data is ignored too. Li Fan [7] proposed the main content of test design for test suite construction, and has considered the magnitude of testing sequence in software testing activity.

As a consequence, our study and testing practice [9] [10] [11] [12] has demonstrated the magnificence of data-driven in engineering application software, and the grey-box approach is proposed for dealing with the difficulty of software testing for GUI based on window frame in our recent research [2].

Further, the test suite construction, as the key aspect of software testing activity, should be paid more attention, so previous work [12] has investigated the test suite construction of smoke test. Moreover, the recent study [2] has proposed the grey-box technique of integration testing for GUI software, and more recent research [3] has provided the constructing method of test suite, including the testing method of data boundary value of input controls and the typical disposing method of function and state testing using the improved STD (State Transform Diagram).

Consequently, this study gives the guideline of test suite construction for GUI software centered on grey-box approach, and deeply discusses the constructing method and process including unit testing and integration testing by a typical case of GUI software—PQMS (Product Quality Monitoring Software/System).

3. Background and Definition of Terminology

When we start a task of software testing, regardless of any testing type, the first matter must be the construction of test case or test suite. In fact, test suite construction has become the key and most important work in software testing. However, how to construct test case or test suite for GUI software with engineering method? It probably is not a very easy-to-answer problem. As such, we would build some references in the following.

3.1. Research Methodology

Generally, software testing can be divided into four testing phases including unit testing, integration testing, system testing and validity testing. Furthermore, white box method and black box method are usual testing methods in software testing. And white box method is the traditional method previously in unit testing especially for logic testing, but black box method is more usually used in function testing [5] [6] [7].

For the view of testing phase, test suite construction should generally include test suite construction of unit testing and test suite construction of integration testing. However, in some situations, test suite construction of system testing

and test suite construction of validation testing must be considered too [6].

For the view of software production, test suite construction generally has two aspects, i.e. test suite construction of baseline version and test suite construction of incremental change [3].

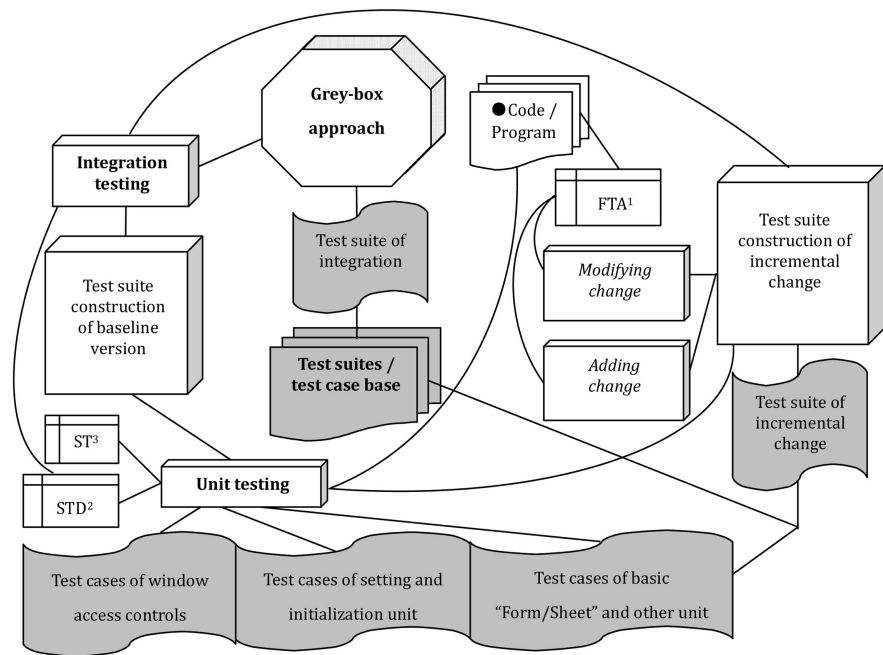
In actual software testing, all above objects should be considered in terms of factual requirement. In order to deeply investigate the test suite construction, we firstly outlined a whole view of test suite construction centered on grey-box approach as shown in Figure 2 [9] [10] [11].

3.2. Software Testing for GUI Software

For GUI software, the GUI testing is a very important context and should be taken as a high-level testing, especially in large scale GUI software system. Consequently, we gave a general overview about the level of GUI testing in software testing as shown in Figure 3.

In contrast to previous DOS software, such as the runtime software of industrial control device, the GUI testing in large scale software system faced some challenges [13] as follows:

- Functionalities of software system are much more.
- States of software system are much more and transformed with more branch and loop.
- Data organizing type and saving mode are more complex.
- Information interchanging is done by more widely and rapidly mode such as internet.
- Application scenarios will be happened with more layers and more entities.



¹FTA-Fault Tree Analysis, ²STD-State Transform Diagram, ³ST-Sampling Test

Figure 2. Constructing scheme of test suite centered on grey-box approach.

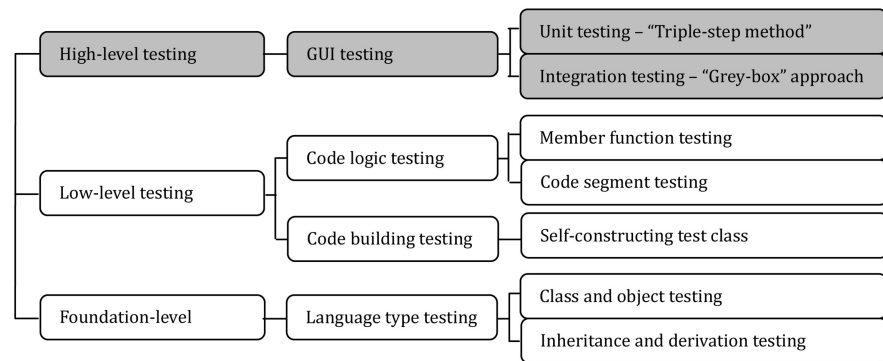


Figure 3. The level of GUI testing in software testing.

Facing to these challenges, the test suite construction of GUI software should consider more concerns from more requirements [14] [15] especially to strengthen the usability of user. Firstly, the GUI testing is taken as a more important part in GUI software testing including the testing of GUI controls and components and the general testing of function and state, besides the traditional logic testing.

3.3. Construction Process of Software Test Suite

The workload of test suite construction almost has the ratio of 60% in software testing. The testing organizing of test suite construction will determine the quality and efficiency of software testing. Some basic and preparing work must be done before starting the test suite construction, such as familiarizing the code submitted from programmer, analyzing the requirement of testing environment and the dependency of units, and determining the general testing strategy and method, etc. If it is necessary, all preparing work should be organized by skilled tester or manager. In general, the procedure of test suite construction should be executed according to **Figure 4**.

3.4. Definition of Terminology

In this paper, the following terminology is used to understand the accurate meaning of our study. For our study opinion, new software testing method should be explored because of some old traditional methods are not fitted for the *GUI software* testing, which mainly includes the “*Triple-step method*” in unit testing and “*Grey-box approach*” in integration testing.

In order to deeply discuss the problem of the GUI software testing, the *GUI testing* is defined that is taken as a high-level testing and usually implies the software testing from GUI aspect for general GUI software. The GUI testing based on “Sheet/Form” is factually adopted as a strategy of software testing, which the “Sheet/Form” is taken as a single unified entity with all controls/components and internal function disposing. *Test suite construction* of GUI software implies the constructing processing of test suite for GUI software, and it should include the dependency analysis with FTA, the writing and assigning of test case, and

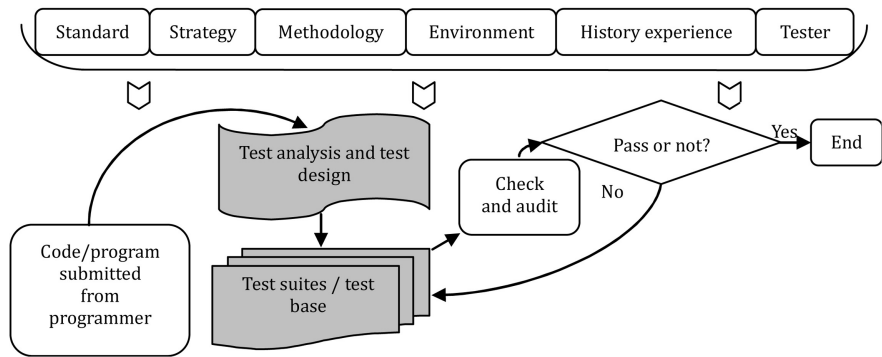


Figure 4. Procedure of test suite construction.

the modifying and updating of test suite, etc.

In the procedure of test suite construction for GUI software, “Triple-step method” should be adopted for unit testing in terms of the testing sequence of “data testing → function testing → state testing”, and *data boundary value testing* of GUI input controls must be done firstly to assure the correctness and accuracy of data input, and *process boundary value testing* must be executed to avoid various leaks e.g. the leakage in the “getting the best deal”, and the *data interface and format testing* must be strengthened to get the good usability. Function testing is the core task in the unit testing, and the *limitation testing* is generally done in the process of function testing, which limitation testing include all obliged limitation e.g. limitation of data input, limitation of function running, etc. At the same time, the state testing should be finished in unit testing to improve the software quality really, and “0-switch state testing” may be executed for general GUI software which every effective single-step transformation must be completely tested [7]. Consequently, the “Grey-box approach” should be adopted for integration testing. In general, integration testing has distinguished requirement with the unit testing, data boundary value testing of GUI input controls should not be repeated, and function testing only occurred among units regardless of internal function disposing for GUI software, and state testing should be executed in terms of factual situation of software/system.

4. Grey-Box Approach [2]

For GUI software, the grey-box approach can be mainly used for constructing the test suite of baseline version in integration testing and it will achieve minimum of testing time for dealing with combinatory explosion of GUI testing. The principle of grey-box approach is shown in **Figure 5**.

In **Figure 5**, the front part is interface controls of Windows GUI, *i.e.* “Windows control 1”, “Windows control 2”, ..., “Windows control m”. And in the center part of **Figure 5**, “Message handling” and “Map function” are the core part of message disposing based on event from C_1, C_2, \dots, C_m . As a consequence, the function disposing unit, *i.e.* “Handling unit 1”, “Handling unit 2”, ..., “Handling unit n”, is the actual function handling part by message map route, *i.e.* $H_1,$

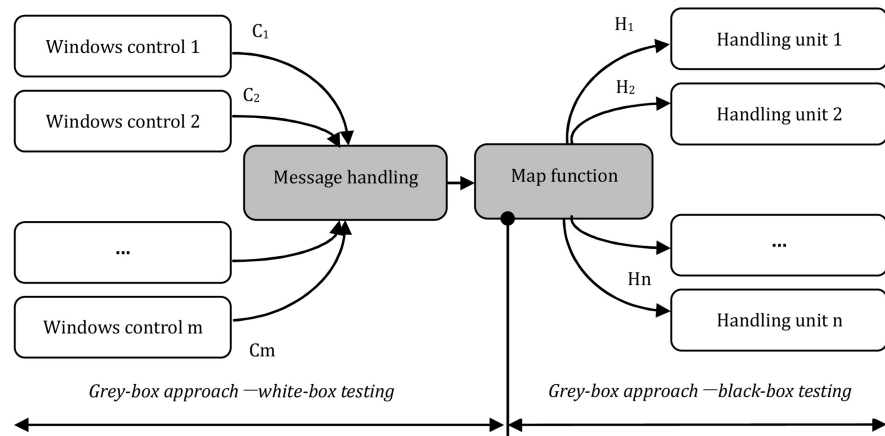


Figure 5. The principle of grey-box approach for GUI software.

H_2, \dots, H_n . We insert a pole in the beginning of mapping function which is marked with the point illustrated in **Figure 5**, for the section in the front of the pole, the white-box method is used to test, and for the section behind the pole, the black-box method is applied for actual function testing. Using this approach, we can only test the message path $\{C_1 \rightarrow H_1\}, \{C_1 \rightarrow H_2\}, \{C_1 \rightarrow \dots\}, \{C_1 \rightarrow H_n\}$, it will cover all message path $\{\{C_1 \rightarrow H_1\}, \{C_1 \rightarrow H_2\}, \{C_1 \rightarrow \dots\}, \{C_1 \rightarrow H_n\}\}, \{\{C_2 \rightarrow H_1\}, \{C_2 \rightarrow H_2\}, \{C_2 \rightarrow \dots\}, \{C_2 \rightarrow H_n\}\}, \{\{\dots \rightarrow H_1\}, \{\dots \rightarrow H_2\}, \{\dots \rightarrow \dots\}, \{\dots \rightarrow H_n\}\}, \{\{C_m \rightarrow H_1\}, \{C_m \rightarrow H_2\}, \{C_m \rightarrow \dots\}, \{C_m \rightarrow H_n\}\}$.

Generally, the selection of the pole position has several methods, details are specified as follows: 1) The path aggregation point by white-box analysis. 2) The point that message route must go through. 3) The entrance point of map function. 4) The entrance point of the initial member function.

By analysis and computation with factual example [2], we conclude that, in integration testing for GUI Software, the application of grey-box technique will greatly improve the testing efficiency with about four times according to testing executed time.

5. Software Test Suite Construction of Unit Testing

In a large sense, the majority of application software at present is GUI software, and the unit testing of GUI software is the indispensable phase. Thus, test suite construction of unit testing is an impassible and basic work.

Generally, the unit testing of GUI software should execute “Triple-step method” as shown in **Figure 6**, including 1) step 1—data testing, 2) step 2—function testing, and 3) step 3—state testing. For data testing, there are boundary value testing, data format and interface testing, and data safety testing, in which boundary value testing should include data boundary value testing and process boundary value testing. For function testing, it not only includes function-self run testing but also may have the limitation testing, such as the limitation of unpermitted empty, the input format limitation, the comparing and computing limitation, etc. In some cases, the limitation testing can be incorporated into

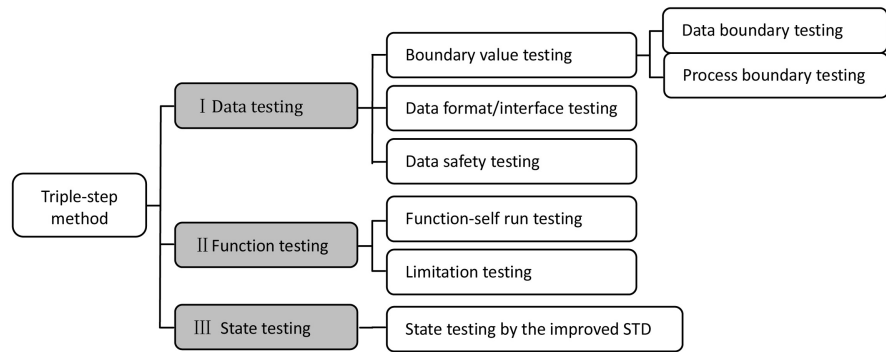


Figure 6. The composition of “Triple-step method”.

data testing, e.g. the testing for the control limitation is given independently. For state testing, the improved STD should be applied with “0-switch” requirement for general situation.

In the test suite construction of data testing, some matters should be considered as follows: 1) Sampling testing method in data boundary value testing may be adopted in terms of the situation of factual software. 2) Data format and interface testing has several types including text file, database, etc. 3) Data safety testing is generally delayed to integration testing and system testing. 4) Usual items of test case include “ID”, “Input”, and “Expected output”, etc., and test environment may be given if necessary.

In the test suite construction of function and state testing, some matters should be also noticed as follows: 1) Function-self run testing should be constructed firstly, particularly noticing the construction for the window access controls and white-box testing in grey-box approach. 2) Limitation testing must be completely conducted as careful as possible. 3) Before state test suite constructing, all states and state-transform should be depicted correctly and completely in the improved STD. 4) Items in the table of test cases should contain “ID”, “Start state”, “End state”, “Input”, “Expected output” etc., and precondition should be given if necessary. 5) The ID must be unified by specification. 6) The description of “Input” and “Expected output” must be clear and easy-to-understand.

5.1. Test Suite Construction of Baseline Version

5.1.1. Window Access Controls or Window Controls of Function Access

Window access controls or window controls of function access is one kind of GUI controls, which the interface control locates and is contained in the window interface and user can access the software function by this control. For Windows system, usual controls have the menu item of main menu, the menu item of pop-up menu, shortcut key, toolbar, hotkey, etc.

The test suite construction of window access controls should be prior to do in unit testing especially in preparation of integration testing for grey-box approach [2] [3], and it must be considered in specific terms including testing organizing e.g. “cross-testing” [3] [11]. The specific format of the writing of test

case and test suite may be executed according to the actual situation of software companies, and we give a basic reference format for guidance in this paper. Consequently, a typical example of test case of the menu item testing is given in **Table 1**, which is a “Basic Setting” menu of PQMS2 (Product Quality Monitoring Software 2.0) [16].

Matters needed attention in writing test case of window access controls are: 1) all test cases for a main menu should be together, 2) only highlighting the menu item is required rather calling the function, 3) including the content checking of window access controls e.g. the spelling of words.

5.1.2. Basic Setting Unit

For a software system, the basic setting is generally needed for the change of user status and application status. Being different from the window access control, the basic setting unit has the interaction of data, and its test suite construction should include data testing and function and state testing.

1) Data testing

In general, test suite construction of the basic setting unit may be similar to general basic “Sheet/Form” [3] including the testing of data boundary value, process boundary value, and data format and interface etc. Here, the data boundary value testing is a new testing project and testing method possibly applying sampling testing, details may be referred to [3]. For the “Basic Setting” sheet of PQMS2, as a typical small “Sheet/Form”, the testing of data boundary value should be prior to be conducted for GUI software, and the specific format of test cases is given in **Table 2**.

2) Function and state testing [3]

The function and state testing of the “Basic Setting” sheet is relatively simple because less type and quantity of controls are used, and the specific format of test cases of function testing is given as shown in **Table 3**.

For the state testing of this basic setting unit, the improved STD should be applied, and details are shown in **Figure 7**, which the event e_i and action a_i are given with an attached list and all of symbol description are omitted here. Consequently, the construction of test cases can be done according to this diagram,

Table 1. Test cases of the window access controls for baseline version.

ID	Input	Expected output
PQMS2-MEF-UNI-TC010	Starting the main menu from windows main interface, and click and highlight the menu item of “Basic Setting - System User - Set Authority”.	This menu item is correct and it can be activated for highlighting.
PQMS2-MEF-UNI-TC011	Starting the main menu from windows main interface, and click and highlight the menu item of “Basic Setting - System User - Modify Password”.	This menu item is correct and it can be activated for highlighting.
PQMS2-MEF-UNI-TC017	Starting the main menu from windows main interface, and click and highlight the menu item of “Basic Setting - Basic setting of monitoring”.	This menu item is correct and it can be activated for highlighting.

Table 2. Test cases of the window access controls for baseline version.

ID	Input	Expected output
PQMS2-MBB-UNI-TC010	Starting the basic setting sheet of monitoring from menu item “Basic Setting - Basic setting of monitoring”, and in the ComBoBox of “Accuracy of inspection data” input character “T...” by keyboard with limited seven times, and stop when “Da” voice is listened.	Only six characters are permitted for input in the ComBoBox of data accuracy.
PQMS2-MBB-UNI-TC011	In the ComBoBox of “Moving mode”, input continuously character “T...” by keyboard with limited 51 times, and count actually if width is shorted, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the ComBoBox of moving mode.
PQMS2-MBB-UNI-TC012	In the ComBoBox of “Sampling limit of counting chart”, input continuously character “T...” by keyboard with limited 51 times, and count actually if width is shorted, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the ComBoBox of sampling limit of counting chart.
PQMS2-MBB-UNI-TC013	In the ComBoBox of “Division or department”, input continuously character “T...” by keyboard with limited 51 times, and count actually if width is shorted, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the ComBoBox of division or department.
PQMS2-MBB-UNI-TC014	In the ComBoBox of “Name of testing operator”, input continuously character “T...” by keyboard with limited 51 times, and count actually if width is shorted, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the ComBoBox of name of testing operator.

and the specific format of test cases of state testing is given as shown in **Table 4**.

5.1.3. Initialization Unit

Because the initialization unit is driven directly by a menu item with the member function rather than driven by controls in a “Sheet/Form”, this unit has not data testing of input controls.

For the function and state testing, it is necessary to analyze with the improved STD even though it may be not complex. The improved STD of initializing to delete all inspection data from category is shown in **Figure 8**. Consequently, test cases can be designed according to this improved STD with less difficulty according above method. As such, test cases of function testing in this initialization unit are given in **Table 5**, and test cases of state testing are given in **Table 6**.

5.1.4. Basic “Sheet/Form” Unit

By a lot of testing practice, in our opinion, the GUI testing of software system may be executed based on “Sheet/Form” [3]. In detail, all controls and components in a “Sheet/Form” should be taken as a unified unit if the function is implemented within the “Sheet/Form”. For the function of software is directly driven by a control and component in the window interface, the testing should be similarly done in terms of the member function driven by event of the control in

Table 3. Test cases of the function testing for basic setting unit^a.

ID	Input	Expected output
PQMS2-SMF-UNI-TC010	Starting “Basic setting of monitoring” sheet from the menu item of “Basic Setting - Basic setting of monitoring”, and input “0.001” in the ComBoBox of “Data Accuracy”, and input “Single data - Automatic” in the ComBoBox of “Moving mode”, and input “More than or equal to 10 and less than 1000” in the ComBoBox of “Sampling limit of counting chart” while the information of tester is kept invariant, and click the “Setting” button finally.	Correctly displaying the prompt of “The setting of control chat is saved.” and no prompt of error information.
PQMS2-SMF-UNI-TC011	In the “Basic setting of monitoring” sheet, input “ ” in the “ComBoBox” of “Data Accuracy”, and input “Single data - Automatic” in the “ComBoBox” of “Moving mode”, and input “More than or equal to 10 and less than 1000” in the “ComBoBox” of “Sampling limit of counting chart” while the information of tester is kept invariant, and click the “Setting” button finally.	Displaying the prompt of “The setting of data accuracy is error, and only two values, i.e. 0.01 and 0.001.”
PQMS2-SMF-UNI-TC012	In the “Basic setting of monitoring” sheet, input “0.001” in the “ComBoBox” of “Data Accuracy”, and input “ ” in the “ComBoBox” of “Moving mode”, and input “More than or equal to 10 and less than 1000” in the “ComBoBox” of “Sampling limit of counting chart” while the information of tester is kept invariant, and click the “Setting” button finally.	Displaying the prompt of “The setting of moving mode is error, and only giving the mode of ‘Single data-Automatic’”.
PQMS2-SMF-UNI-TC013	In the “Basic setting of monitoring” sheet, input “0.001” in the “ComBoBox” of “Data Accuracy”, and input “Single data - Automatic” in the “ComBoBox” of “Moving mode”, and input “ ” in the “ComBoBox” of “Sampling limit of counting chart” while the information of tester is kept invariant, and click the “Setting” button finally.	Displaying the prompt of “The setting of sampling limit of counting chart is error, and the number must be more than 0 and less than 10,000”.

a. In writing test case of function testing for general situation, it is noticed that all correct and error possibility of function running must be considered.

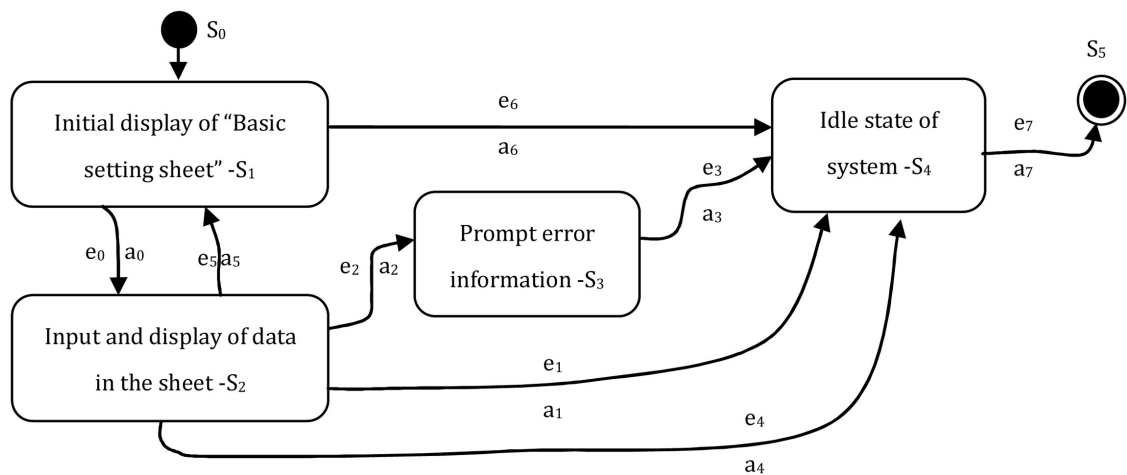


Figure 7. The improved STD of “Basic setting” sheet unit.

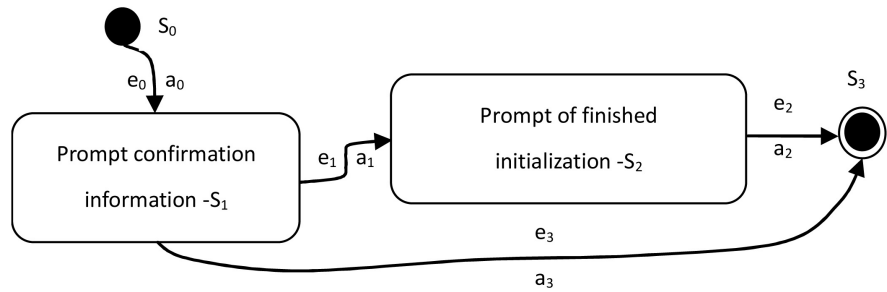


Figure 8. The improved STD of initialization unit.

Table 4. Test cases of the state testing for basic setting unit^b.

ID	Start state	End state	Input	Expected output
PQMS2-SMS-UNI-TC010	S ₀	S ₁	Click the menu item of “Basic Setting - Basic setting of monitoring” from the main interface.	Display the sheet of basic setting of monitoring.
PQMS2-SMS-UNI-TC011	S ₁	S ₄	No inputting operation, click the button of “Cancel” or “Shut-off” button of up-right corner in the sheet.	Cancel basic setting, and enter the system idle.
PQMS2-SMS-UNI-TC012	S ₁	S ₂	Enter the sheet again, and input data in the ComBoBox.	Display the data in the sheet.
PQMS2-SMS-UNI-TC013	S ₂	S ₄	Input correct default data in the sheet, and click the button of “Setting”.	No prompt of error information, and save the setting data.
PQMS2-SMS-UNI-TC014	S ₂	S ₃	Enter the sheet again, and input “” in all ComBoBoxes, and click the button of “Setting”.	Prompt error information.
PQMS2-SMS-UNI-TC015	S ₃	S ₄	Click the button of “OK” or “Shut-off” button of up-right corner in the prompt dialogue.	Enter the system idle.
PQMS2-SMS-UNI-TC016	S ₂	S ₄	Enter the sheet again, and click the button of “Cancel” in the sheet when inputting data.	Cancel basic setting, and enter the system idle.
PQMS2-SMS-UNI-TC017	S ₂	S ₁	Enter the sheet again, and click the button of “Default” in the sheet.	Restore default setting.
PQMS2-SMS-UNI-TC018	S ₄	S ₅	When system is idle, click “Shut-off” button of up-right corner in the system interface.	Exit.

b. In writing test case of state testing for general situation, it is noticed that the correct and completed STD is needed and 0-switch state testing is passable for most situation in general software system.

the “Sheet/Form”, *i.e.* the testing of this situation should be disposed in the level of member function.

The “Product and class” sheet of PQMS2 [16], as shown in Figure 9, is a typical popular “Sheet/Form” including basic GUI controls and components, and popular functions of data adding, data modifying, data deleting. As such, the test suite construction of data testing, function and state testing for this basic sheet is

Table 5. Test cases of function testing in initialization unit.

ID	Input	Expected output
PQMS2-SSF-UNI-TC011	Starting the main interface, and click the menu item of “Basic Setting - Initialization - System User Authority - Initializing to delete all inspection data from category”, and click button “OK” in the confirming dialogue.	Correctly displaying the prompt of “All inspection data are deleted.” and no prompt of error information.

Table 6. Test cases of the state testing for basic setting unit.

ID	Start state	End state	Input	Expected output
PQMS2-SMS-UNI-TC010	S ₀	S ₁	Click the menu item of “Basic Setting - Basic setting of monitoring” from the main interface.	Display the sheet of basic setting of monitoring.
PQMS2-SMS-UNI-TC011	S ₁	S ₄	No inputting operation, click the button of “Cancel” or “Shut-off” button of up-right corner in the sheet.	Cancel basic setting, and enter the system idle.
PQMS2-SMS-UNI-TC012	S ₁	S ₂	Enter the sheet again, and input data in the ComBoBox.	Display the data in the sheet.
PQMS2-SMS-UNI-TC013	S ₂	S ₄	Input correct default data in the sheet, and click the button of “Setting”.	No prompt of error information, and save the setting data.

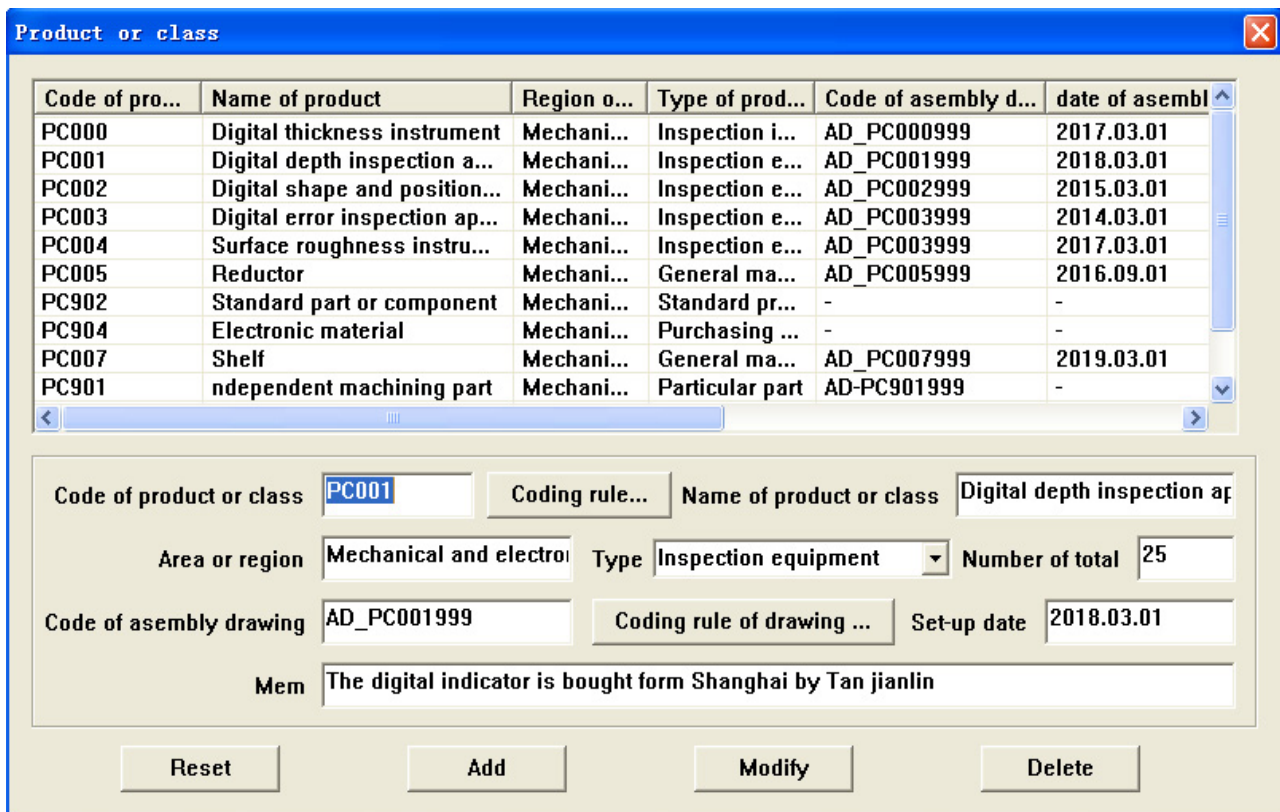


Figure 9. The GUI of “Product and class” sheet unit in PQMS2.

discussed in detail in the following.

1) Data testing of the “Product or class” sheet

a) Data boundary value testing of “input controls”

Similarly, for the “Product and class” sheet, as a typical popular “Sheet/Form”, the testing of data boundary value should also be prior to be conducted while the limitation testing is incorporated into function testing for GUI software, and the specific format of test cases is given in **Table 7**.

2) Data testing of the “Product or class” sheet

The function and state testing of the “Product or class” sheet is a typical example because functions and types of controls are representative in some extent, and the specific format of test cases of function testing of this basic “Sheet/Form” unit is given as shown in **Table A1** of the **Appendix**.

Similarly, the improved STD should be applied to construct the state test case of the “Product or class” sheet, and details are shown in **Figure 10**. Consequently, the test case construction of the state testing can be accomplished according to this diagram, and the specific format of test cases of state testing is given as shown in **Table A2** of the **Appendix**.

5.1.5. Complex “Sheet/Form” Unit

In various “Sheet/Form” units, there is a kind of sheet that there are many kinds of control or component and it has data exchange with other sheet or component, and the “Inspection process” sheet unit is a typical example, as shown in **Figure 11** [16]. In the “Inspection process” sheet unit, the function of “Add to the monitoring category” with “Button control” will do data exchange with the “Tree control” component in main window.

In **Figure 11**, as a more complex “Sheet/Form” unit, the sight of typical sheet of data disposing is presented. Moreover, the “Tree control” on the left of the sheet is to display the data chain of “product or class - part/component - inspection process”, and detail information of inspection process taken in the “List control” can be found on the up-right of panel, and the area on the down-right of panel is to accomplish the input of inspection process data including popular EditBox control, ComboBox control, Button control, etc.

Table 7. Test cases of the data boundary value testing for the “Product or class” sheet.

ID	Input	Expected output
PQMS2-PDB-UNI-TC010	Start the product or class sheet from the menu item of “Basic quality data - Product or class”, and continuously input character “T...” by keyboard with limited 21 times in the EditBox “Code of product or class”, and stop when “Da” voice is listened.	Only 20 characters are permitted for input in this EditBox.
PQMS2-PDB-UNI-TC011	In the product or class sheet, continuously input character “T...” by keyboard with limited 51 times in the EditBox “Name of product or class”, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in this EditBox.

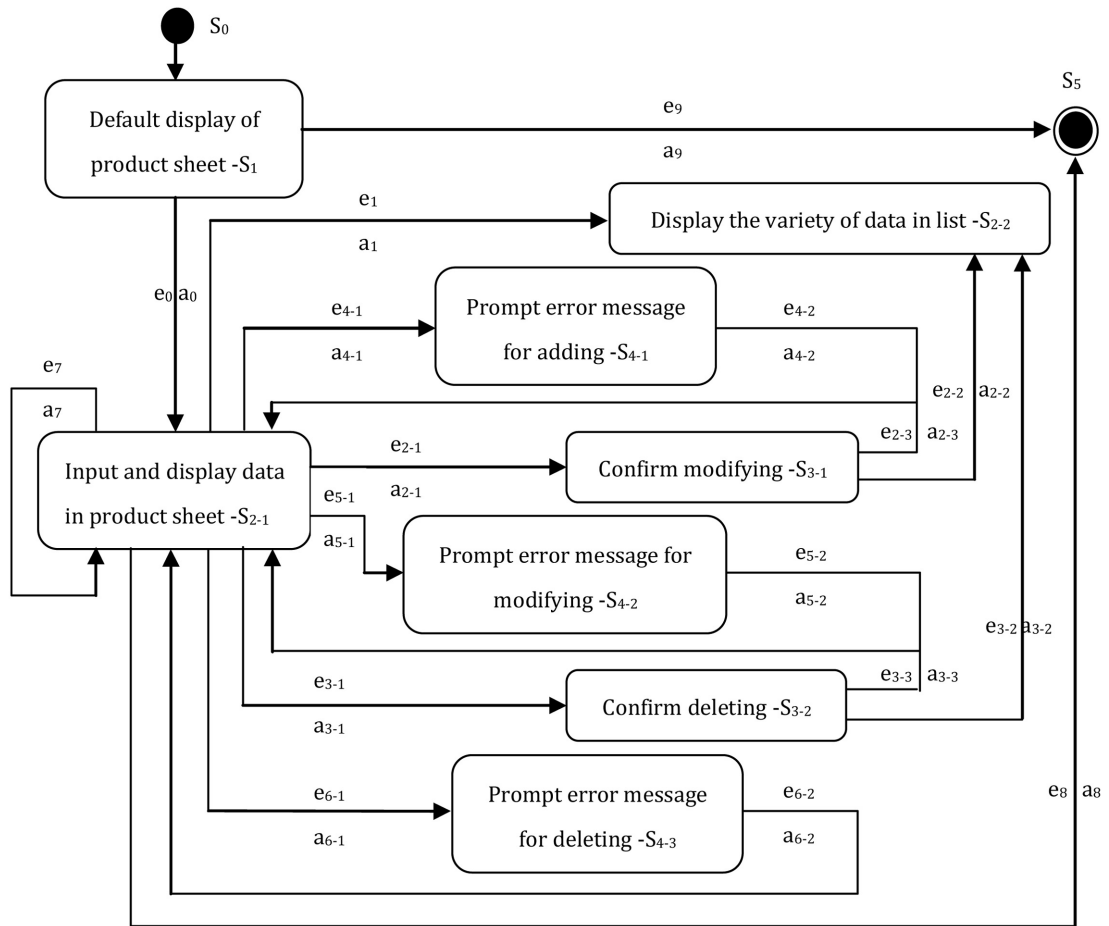


Figure 10. The improved STD of “Product or class” sheet unit.

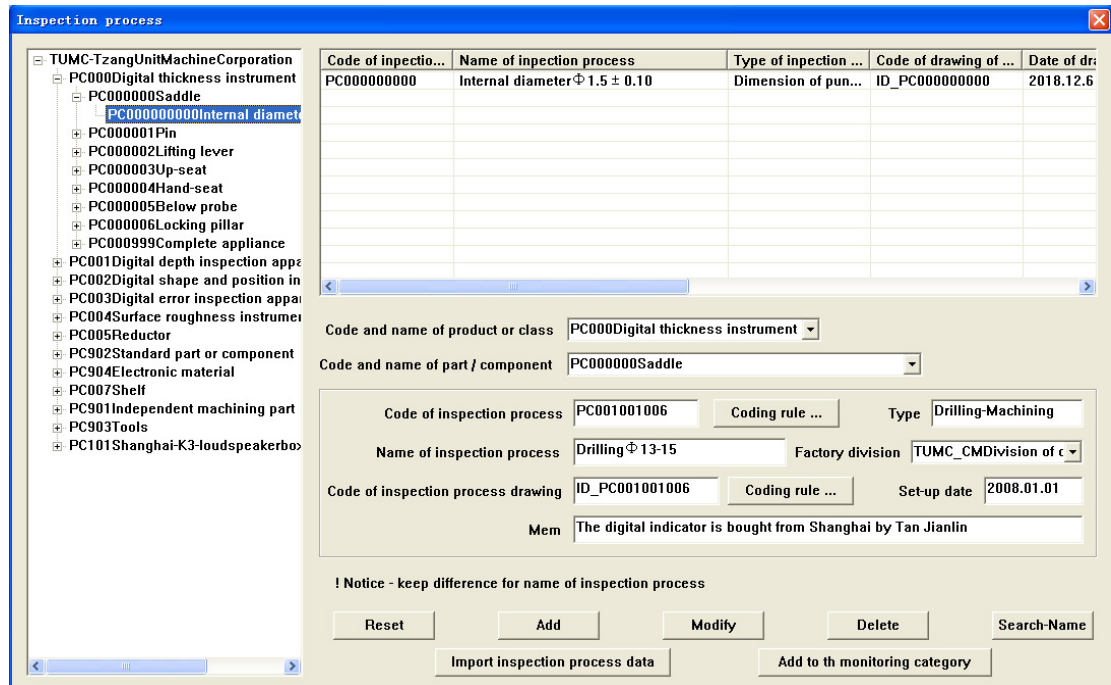


Figure 11. The GUI of “Inspection process” sheet unit.

For this “Inspection process” sheet unit, its unit testing should similarly include data testing, function testing and state testing.

1) Data testing

As mentioned above, test suite construction of data testing of the “Inspection process” sheet unit should include the testing of data boundary value, the testing of process boundary value, and the testing of data format and interface etc., while the testing of data limitation is incorporated into function testing.

a) Data boundary value testing of “input controls”

At first, the testing of data boundary value should be similarly executed as a new testing project for GUI software, and the specific result of test design is given in **Table A3** of the **Appendix**.

b) Data format testing of data file

In general, the complex “Sheet /Form” unit has the processing of many kinds of data I/O including the data type of controls, the data type of text file, and the data type of database. The data type of inputting from controls has been discussed as above. Here, the data interface and format testing for the data type of text file will be investigated in this “Inspection process” sheet.

In the testing of data format and interface for the “Inspection process” sheet unit, the format of file name should be tested firstly, and the format of every data may be checked and verified in sequence. Test cases of the data format testing for the “Inspection process” sheet is given in **Table A4** of the **Appendix**. Additionally, **Table A5** of the **Appendix** is the specification of data format with the checking of “the tail of file cannot be empty” for “Import inspection process data” in the “Inspection process” sheet. And other test cases of data format testing are accomplished with code of “PQMS2-PDI-UNI-TC010~TC016”, details are omitted here.

2) Function and state testing

As a complex “Sheet/Form” unit, the function and state testing of the “Inspection process” sheet unit is relatively difficult not only due to many controls and components but also because of a lot of function disposing.

For the function testing of the “Inspection process” sheet unit, it must include 1) initialization displaying of the sheet with default data—test case “PQMS2-IDF-UNI-TC010”, 2) clicking and displaying of inspection process data in monitoring category—test case “PQMS2-IDF-UNI-TC011”, 3) adding of inspection process data—test case “PQMS2-IDF-UNI-TC030~TC044” and “PQMS2-IDF-UNI-TC003-AD~TC004-AD”, 4) modifying of inspection process data—test case “PQMS2-IDF-UNI-TC012~TC029” and “PQMS2-IDF-UNI-TC001-AD~TC002-AD”, 5) deleting of inspection process data—test case “PQMS2-IDF-UNI-TC050~TC053” and “PQMS2-IDF-UNI-TC005-AD~TC006-AD”, 6) searching of inspection process data according to name—test case “PQMS2-IDF-UNI-TC054~TC058” and “PQMS2-IDF-UNI-TC007-AD, and 7) resetting of data input—test case “PQMS2-IDF-UNI-TC059”.

As a guidance of test suite construction, we only give some test case examples for adding of inspection process data as shown in Table A6 of the Appendix, and remaindering is omitted here. Consequently, Table A6 has given the function-self run testing with “PQMS2-IDF-UNI-TC030” and the limitation testing of “unpermitted overlapping” and “unpermitted empty” with “PQMS2-IDF-UNI-TC031~TC034”. As such, other test case constructing of limitation testing can be done in a similar way.

For the state testing of the “Inspection process” sheet unit, the improved STD should be drawn before constructing the state test suite. In consequence, Figure 12 has shown the details of the improved STD of the “Inspection process” sheet

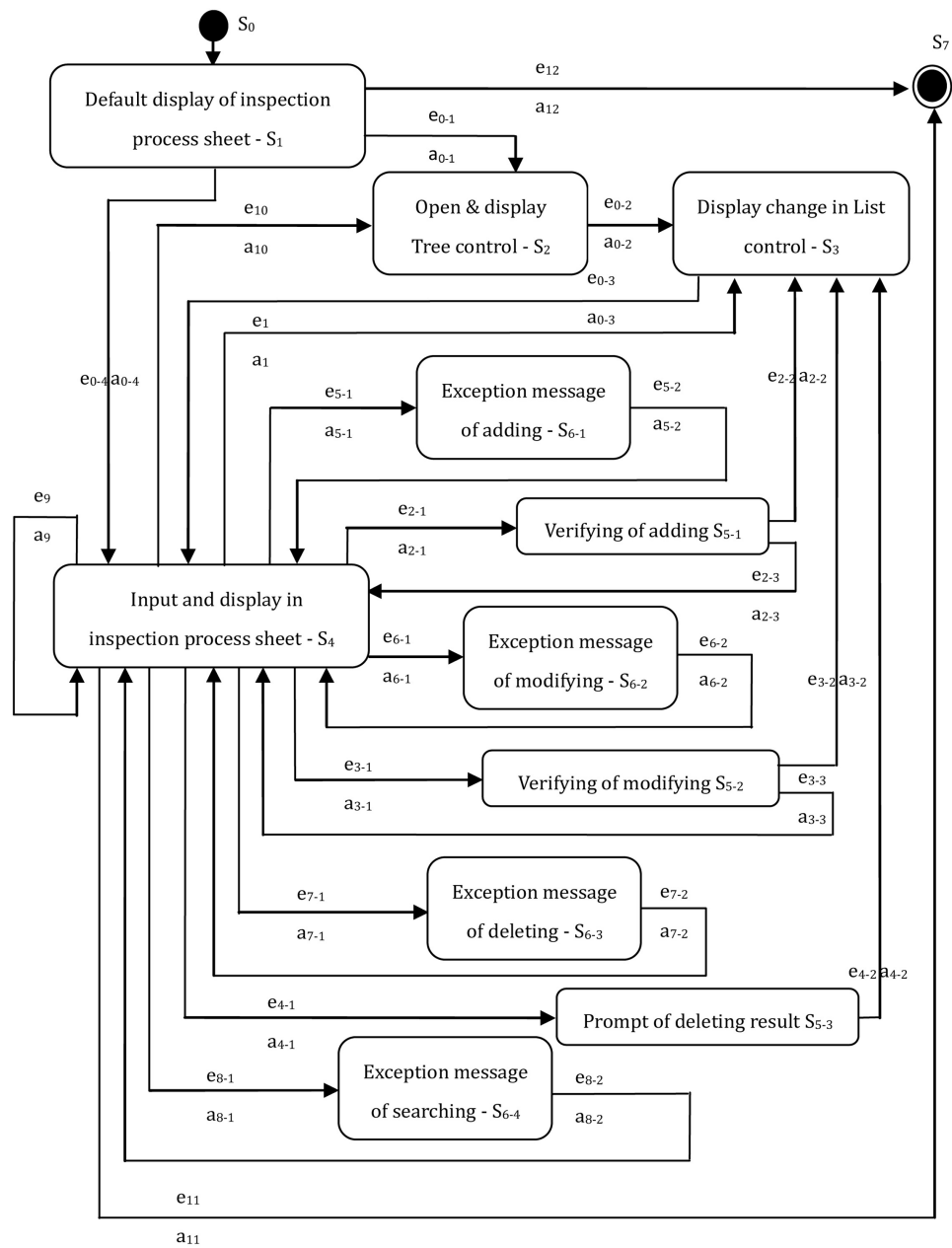


Figure 12. The improved STD of the “Inspection process” sheet unit.

in PQMS2. As such, we can construct the test suite of state testing for the “Inspection process” sheet unit in PQMS2 in terms of the improved STD. Consequently, the specific format of test cases of state testing is given as shown in **Table A7** of the **Appendix**.

5.2. Test Suite Construction of Incremental Unit Testing

In the cycle of software developing and maintenance, all composing parts of software will change to fitful the varied requirement of customers, e.g. GUI, data and data organizing etc. For this kind of variety, incremental construction of test suite must be done in regression testing [9]. At the same time, we must know whether the modified or added part has influenced the other unit of the software [9]. Generally, because the unit testing based on “Sheet/Form” has been independent of other units for specified requirement, the dependency analysis of unit testing could be usually omitted instead of integration testing, but the testing of the member function must be concerned with distinguished consideration. As a consequence, the incremental construction is demonstrated in terms of grey-box approach [2] in the following.

5.2.1. Incremental Unit Testing of Window Access Controls

Usually, window access controls are changing including its content and layout according to requirement of users and manager. In the view of quality assurance, this variety must be tested and controlled in terms of the specification of regression testing.

1) Modification of window access controls

If the content of a menu item or main menu is modified, this modification must be tested for regression testing. Consequently, the test case of this window access control must be constructed again. One typical example is shown in “ID—PQMS2-MEF-UNI-TC017-MF” of **Table 8**, which it has tiny difference from PQMS2-MEF-UNI-TC017 of **Table 1**.

2) Addition of window access controls

Table 8. Test cases of window access controls for incremental unit testing.

ID	Input	Expected output
PQMS2-MEF-UNI-TC017-MF	Starting the main menu from windows main interface, and click and highlight the menu item of “Basic Setting - System User - Basic setting of control chart”.	This menu item is correct and it can be activated till highlighted.
PQMS2-MEF-UNI-TC025	Starting the main menu from windows main interface, and click and highlight the menu item of “Quality Monitoring Report - Monthly Report”.	This menu item is correct and it can be activated till highlighted.
PQMS2-MEF-UNI-TC026	Starting the main menu from windows main interface, and click and highlight the menu item of “Quality Monitoring Report - Yearly Report”.	This menu item is correct and it can be activated till highlighted.

When a set of functions are added for a software system, all additions must be tested for regression testing. Consequently, test cases of all added window access controls must be constructed. One typical example is the adding of “Report output” in an application, the construction of test cases in PQMS is shown in “ID-PQMS2-MEF-UNI-TC025-AD~TC026-AD” of **Table 8**.

5.2.2. Incremental Unit Testing of “Sheet/Form” Unit

1) Data testing

Modification of basic “Sheet /Form” unit

If the control of a basic “Sheet/Form” is changed, this change must be tested for regression testing in terms of this change. Consequently, the test case of this basic “Sheet/Form” must be constructed again. Similarly, if the data boundary of controls is changed, the test case of data boundary value must be conducted again. One typical example is the change of data length in two EditBoxes of “Product or class” sheet as shown in “PQMS2-PDB-UNI-TC010-MF~TC011-MF” of **Table 9**, which it has some difference from **Table 7**.

2) Function and state testing

Addition of basic “Sheet/Form” unit

When an interface control and implement function are added for a “Sheet/Form”, this adding must be tested for regression testing. Consequently, test cases for this adding change must be constructed. One typical example is the adding of “Search-Name” in the data processing sheet, the construction of test cases is similar to the data testing above, but the function testing of member function of “Search-Name” button must be added, and state testing of this button must be inserted into the state test suite.

6. Software Test Suite Construction of Integration Testing

6.1. Test Suite Construction of Baseline Version

In general, the test suite construction of integration testing for baseline version should be done using grey-box approach [2] for GUI software, except that it is very simple program or non-GUI software. As mentioned in Section 4, the

Table 9. Test cases of the data boundary value testing for incremental unit testing.

ID	Input	Expected output
PQMS2-PDB-UNI-TC010-MF	Start the product or class sheet from the menu item of “Basic quality data - Product or class”, and continuously input character “T...” by keyboard with limited six times in the EditBox “Code of product or class”, and stop when “Da” voice is listened.	Only five characters are permitted for input in this EditBox.
PQMS2-PDB-UNI-TC011-MF	In the product or class sheet, continuously input character “T...” by keyboard with limited 21 times in the EditBox “Name of product or class”, and stop when “Da” voice is listened.	Only 20 characters are permitted for input in this EditBox.

grey-box approach is a better method in integration testing for GUI software based on window frame with efficiency improved about four times.

For the constructing of test suite of integration testing based on grey-box approach, the general procedure can be demonstrated as follows.

- For one function disposing, several test cases of fore-end white-box testing should be constructed according to all kinds of control types in window frame.
- In terms of factual software, using black-box testing method, test cases of function testing are conducted in sequence and distinguishing with various running situation, generally ignoring finished parts in unit testing.
- Conducting test case for all follow-up function disposing with black-box testing method.
- In the process of all follow-up construction of function test case, the mapping function should be chosen with the most rapid route.
- If it is necessary, the test case of fore-end white-box testing must be conducted for all follow-up construction including controls activated in other units such as “Hotkey”.
- Necessary description should be given in the process of test case construction.

Here, the constructing method and writing format are given for the integration testing with the case software, it is noticed that the precondition is generally needed for grey-box approach.

Without loss representative and typicality for GUI software, the integration testing of “Division and department” sheet in PQMS2 is investigated for baseline version according to the grey-box approach.

6.1.1. Test Suite Construction for “Division and Department” Sheet

The “Division and department” sheet is a basic “Sheet/Form” for getting basic data of factory division and department in PQMS2, which the GUI is similar to **Figure 9**. For shortening aim, the test suite constructing of integration function of “Add to monitoring category” in the “Division and department” sheet is only given here. Using the grey-box approach, test cases of the front-end white-box testing are given in **Table 10**, and test cases of the black-box testing are given in **Table 11**.

6.1.2. Modification of Window Access Controls

As we all known, window access controls and its implementation functions usually are needed to modify sometimes, e.g. for fulfilling the supervision of key sampling or UI check. If the window access control changed, the test case of integration testing must be constructed again in terms of grey-box approach. As a typical example, in PQMS2, the menu item of “Output of inspection data” is modified to “Backup of inspection data output”, and test cases of fore-end white-box were changed with ID “P QMS2-ENT-INT-TC368-MF~TC369-MF” as shown in **Table 12**, but test cases of the black-box testing are unnecessary to

Table 10. Test cases of front-end white-box for the “Division and department” sheet.

Precondition—Insert the pole of <MessageBox (“Testing output.”);> at the beginning of member function of <BOOL CDivisionDIALOG::OnInitDialog (>.		
ID	Input	Expected output
PQMS2-ENT-INT-TC300-AD	In the monitoring category of main window interface, click the root item, choose “Add division and department” using the right key of mouse, and activate the “Division and department” sheet.	Prompt “Testing output.”
PQMS2-ENT-INT-TC301-AD	In the main window interface, click the menu item of “Basic setting - Division and department”, and activate the “Division and department” sheet.	Prompt “Testing output.”
PQMS2-ENT-INT-TC302-AD	Click the shortcut key “Alt-S” and “Alt-D” from the menu item of “Basic setting (S) - Division and department (D)”.	Prompt “Testing output.”
PQMS2-ENT-INT-TC303-AD	In the main window interface, click the toolbar item of “Division”, and activate the “Division and department” sheet.	Prompt “Testing output.”

Table 11. Test cases of black-box for the function of “Add to monitoring category”.

Precondition—(a) Delete the pole of <MessageBox (“Testing output.”);> at the beginning of member function of <BOOL CDivisionDIALOG::OnInitDialog(>. (b) “TUMC_MC-Division of machining and cutting” and “TUMC_AS-Division of assembly” do not exist in the monitoring category, but two items have been input and saved from the “Division and department” sheet unit.		
ID	Input	Expected output
PQMS2-ENT-INT-TC307-MF	Using the authority of system user, from the toolbar item “Division” of main window interface, start the “Division and department” sheet, choose the recorder of “TUMC_MC-Division of machining and cutting” in the list and display all data in below input controls, and click the button “Add to monitoring category” finally.	Prompt the information of finished adding, and can find this added item in monitoring category.
PQMS2-ENT-INT-TC308-MF	Using the authority of system user, from the toolbar item “Division” of main window interface, start the “Division and department” sheet, choose the recorder of “TUMC_AS-Division of assembly” in the list and display all data in below input controls, and click the button “Add to monitoring category” finally.	Prompt the information of finished adding, and can find this added item in monitoring category.

Table 12. Test cases for the menu item of “Backup of inspection data output”.

Precondition—Insert the pole of <MessageBox (“Testing output.”);> in the front of member function of <BOOL CExportDataDialog::OnInitDialog(>.		
ID	Input	Expected output
PQMS2-ENT-INT-TC368-MF	In the main window interface, click the menu item of “Data I/O—Backup of inspection data output”, and activate the backup of inspection data.	Prompt “Testing output.”
PQMS2-ENT-INT-TC369-MF	Click the shortcut key “Alt-T” and “Alt-I” from the menu item of “Data I/O (T)—Backup of inspection data output (I)”.	Prompt “Testing output.”

modify again if without change in implementation function.

6.1.3. Addition of Window Access Controls and Disposing Functions

“Batch restoring of inspection data” is important function for factual requirement in PQMS, which was not developed in the past. In order to improve the safety of inspection data, this function is arranged to add. Consequently, “Batch restoring of inspection data” is added to the main menu “Data I/O” and the actual function is implemented by programmer. For the “cross-testing”, after the testing task arrangement of manager, tester should do incremental regression testing. As a consequence, testing engineer as the tester analyzed the original code from programmer with FTA tool. During this period, it is noticed that the testing engineer should not only apply FTA tool with careful manner but also be familiar with the code and check it. As a result, **Figure 13** has shown the result of FTA which has been accomplished by testing engineer. [9]

In **Figure 13**, in order to implement the requirement change of “Batch restoring of inspection data is necessary to add”, “Adding coordination-offset coefficient for R chart is necessary respectively in XAve-R chart” is conducted as the top event, and it is noted with mark A. Consequently, $A_{i,j,\dots}$ presents the middle event of fault-tree, and $X_{i,j,k,\dots}$ is the final event of fault-tree, while C_i is the additional condition. Correspondingly, details are demonstrated in **Table 13**. At the same time, the test case choice and adding for “Batch restoring of inspection data”, derived from final events, are shown in **Table 14**.

In terms of the result of FTA as above, we can construct test case of incremental testing. In general, **Table 14** can be directly use to execute the test design *i.e.* test suite construction. As such, the front-end test cases of white-box testing is firstly constructed using grey-box approach as shown in **Table 15**, and PQMS2-ENT-INT-TC145-AD is the added test case for incremental function

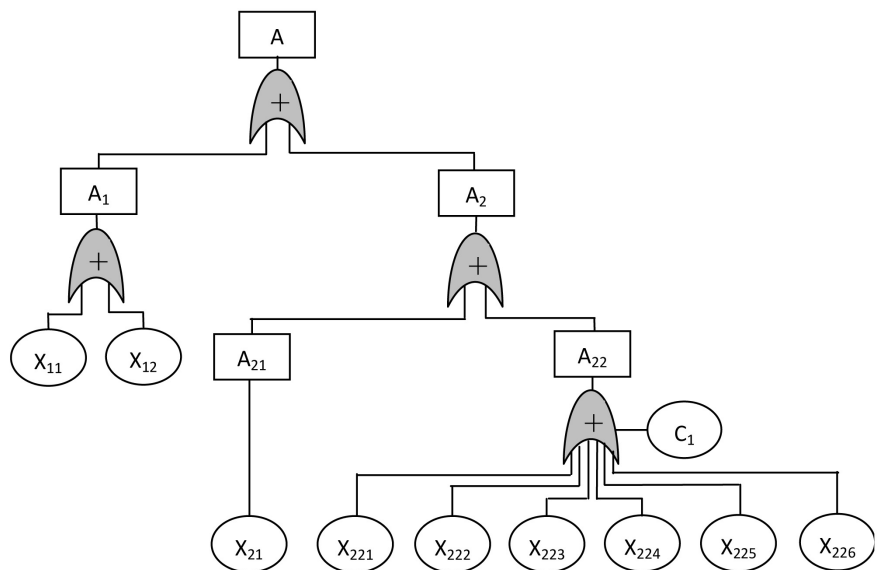


Figure 13. FTA for batch restoring of inspection data in PQMS.

Table 13. The events of FTA for batch restoring of inspection data.

Code	Event name	Code	Event name
A ₁	Add controls from main/pop-up menu and event map	A ₂	Add member function statement and body
A ₂₁	Add member function statement	A ₂₂	Add member function body
X ₁₁	Add controls from main menu/pop-up	X ₁₂	Add event map
X ₂₁	Add member function statement in “MainView.h”	X ₂₂₁	Attributes statement and initialization
X ₂₂₂	Judge if inspection process is exist, and get basic information e.g. inspection process, part, etc.	X ₂₂₃	Searching all target file names
X ₂₂₄	Judge if this inspection data is exist	X ₂₂₅	Add inspection data ID into INSPECTIONDATADIRECTORYNAME
X ₂₂₆	Add filename and all data of inspection data file into SAVINGDIRECTORY	C ₁	Sequence should be “X ₂₂₁ →X ₂₂₂ →X ₂₂₃ →X ₂₂₄ →X ₂₂₅ →X ₂₂₆ ”

Table 14. Choice and adding of test case for batch restoring of inspection data.

ID	Testing content	Note
PQMS2-ENT-INT-TC372-AD	White-box testing for “Popup Menu-menu item”	X ₁₁ , X ₁₂ , X ₂₁
PQMS2-ENT-INT-TC373-AD	White-box testing for “Main Menu - menu item”	X ₁₁ , X ₁₂ , X ₂₁
PQMS2-ENT-INT-TC374-AD	White-box testing for “Shortcut key”	X ₁₁ , X ₁₂ , X ₂₁
PQMS2-ENT-INT-TC145-AD	Integration testing of batch restoring inspection data	X ₁₁ , X ₁₂ , X ₂₁ , X ₂₂₁ , X ₂₂₂ , X ₂₂₃ , X ₂₂₄
PQMS2-FRD-UNI-TC001	Testing if output of starting from monitoring category is correct	X ₁₁ , X ₁₂ , X ₂₁ , X ₂₂₁ , X ₂₂₂
PQMS2-FRD-UNI-TC002	Testing the output of starting from “Main Menu-menu item” for correct inspection process	X ₁₁ , X ₁₂ , X ₂₁ , X ₂₂₁ , X ₂₂₂
PQMS2-FRD-UNI-TC003	Testing the output of starting from “Main Menu-menu item” for incorrect inspection process	X ₁₁ , X ₁₂ , X ₂₁ , X ₂₂₁ , X ₂₂₂

Table 15. Test cases of front-end for adding function of batch restoring inspection data.

Precondition—Insert the pole of <MessageBox (“Testing output.”);> in the front of member function of <BOOL CMainView::CBatchRestoringInspectionData (>.>		
ID	Input	Expected output
PQMS2-ENT-INT-TC372-AD	In the monitoring category of main window interface, click the item “Division of machining and cutting - CM_Digital inspection apparatus of depth_Pillar - Thickness 10(-0.10 0)”, and click the menu item “Batch restoring of inspection data” using the right key of mouse.	Prompt “Testing output.”
PQMS2-ENT-INT-TC373-AD	In the main window interface, click the menu item of “Data I/O—Batch restoring of inspection data”.	Prompt “Testing output.”
PQMS2-ENT-INT-TC374-AD	Click the shortcut key “Alt-T” and “Alt-P” from the menu item of “Data I/O (T)—Batch restoring of inspection data (P)”.	Prompt “Testing output.”

Table 16. Test case of black box for adding function of restoring inspection data.

ID	Input	Expected output
PQMS2-ENT-INT-TC145-AD	<p>Precondition—(1) Delete the pole of <MessageBox (“Testing output.”);> in the front of member function of <BOOL CMainView::CBatchRestoringInspectionData()>, (2) the inspection process item “Division of machining and cutting - CM_Digital inspection apparatus of depth_Pillar - Thickness 10(-0.10 0)” has existed in monitoring category and its three inspection data—“2019_09_28 InSpDaVa1_1 3, 2019_09_29 InSpDaVa1_1 3, 2019_09_30 InSpDaVa1_1 3” has been just deleted.</p> <p>In the category of main window interface, click the item “Division of machining and cutting - CM_Digital inspection apparatus of depth_Pillar - Thickness 10(-0.10 0)”, and click the menu item “Batch restoring of inspection data” using the right key of mouse, start the restoring.</p>	<p>Three batches of inspection data—“2019_09_28 InSpDaVa1_1 3, 2019_09_29 InSpDaVa1_1 3, 2019_09_30 InSpDaVa1_1 3” can be correctly restored.</p>

testing of integration testing, detail is shown in **Table 16**. Of course, it is noticed that the state testing should be finished in unit testing.

In conclusion, the test suite construction of integration testing has some differences from that of unit testing. On the one hand, the data boundary value testing of input controls is omitted in the data testing of integration testing, except that the testing of data accessing/visiting safety and the testing of obliged data interface and format must be accomplished in sequence. On the other hand, the function integration testing should focus on the function interface among units while limitation testing and state testing should be executed in terms of actual situation of software.

7. Summary

Test suite construction is the most important work in software testing, because workload of test design has the ratio of 60% in software testing activity. Test suite construction is not the same as mechanical design etc., and it can be taken as a kind of process design which directly is projected from the factual software product.

Test suite construction is generally linked to software testing phases, mainly focusing on unit testing phase and integration testing phase in this study. For GUI software testing, in unit testing phase, the test suite construction should generally be done in terms of “Triple-step method” including data testing, function testing and state testing. As a consequence, in integration testing, the “Grey-box approach” is an effective and useful testing methodology for GUI software with prior disposing of window access controls.

This paper aims to provide a referring guidance for researchers and industrial practitioners when conducting test cases and test suite for GUI software in software engineering, and it is based on software testing activity of the case GUI software—PQMS and the author’s own experience of test design. Consequently, it is noticed that tiny tune is needed for software testing practice.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Runeson, P. and Höst, M. (2009) Guidelines for Conducting and Reporting Case Study Research in Software Engineering. *Empirical Software Engineering*, **14**, 131-164. <https://doi.org/10.1007/s10664-008-9102-8>
- [2] TanLi, M.Q., Zhang, Y., Wang, Y.L., et al. (2021) Grey-Box Technique of Software Integration Testing Based on Message. *Proceedings of 3rd International Conference on Artificial Intelligence and Computer Science*, Beijing, 29-31 July 2021, 198-206.
- [3] TanLi, M.Q., Zhang, Y. and Wang, Y.L. (2022) Architecture and Methodology of Unit Testing Embedding Pair-Wise Mode for Small Team. *Journal of Software Engineering and Applications*, **15**, 111-133. <https://doi.org/10.4236/jsea.2022.1511022>
- [4] Boehm, B.W. (1979) *Classics in Software Engineering*. Yourdon Press, New Jersey.
- [5] Patton, R. (2006) *Software Testing*. Pearson Education Inc., London.
- [6] Fu, B. (2014) *Course of Software Testing Technology*. Tsinghua University Press, Beijing.
- [7] Li, F. (2016) *Software Testing Technology*. Mechanical Industry Press, Beijing.
- [8] Alégroth, E. and Feldt, R. (2017) On the Long-Term Use of Visual Gui Testing in Industrial Practice: A Case Study. *Empirical Software Engineering*, **22**, 2937-2971. <https://doi.org/10.4236/jsea.2022.1511022>
- [9] TanLi, M.Q., Zhang, Y. and Wang, Y.L. (2020) Research on Fault Tree Technique in Software Regression Testing. *Computer Engineering and Software*, **41**, 5-8, 25.
- [10] TanLi, M.Q., Zhang, Y. and Wang, Y.L. (2020) System Testing Based on Software Performance. *Computer Engineering and Software*, **41**, 1-4, 25.
- [11] Tang, D., TanLi, M.Q. and Li, T. (2021) Software Test Organizing for Small Team Based on “Pair-Wise” Mode. *Proceedings of 2022 International Conference on Smart Transportation and Future Mobility-CSTFM 2022*, Changsha, 2-4 September 2021.
- [12] TanLi, M.Q., Zhang, Y., Jiang, Y., et al. (2021) Baseline Test Suite Construction of Smoke Test for Extreme Programming. *Proceedings of 2021 International Conference on Communication Engineering and Logistics Management*, Changsha, 24-26 July 2021.
- [13] TanLi, M.Q., Jiang, Y., Wang, Y.L., et al. (2020) Infrastructure Building of Software Testing for Engineering Software Based on Cooperation of University and Company. *Proceedings of the 10th International Workshop on Computer Science and Engineering-WCSE2020*, Shanghai, 19-21 June 2020, 18-26.
- [14] Xu, Y.Y. (2015) A Study of Test Case Reuse Based on CBR. *Computer Engineering and Software*, **36**, 117-120.
- [15] Chen, Z.H. (2005) Research and Implementation of Test Method in Task Arrangement of Resource Satellite. *Radio Engineering*, **35**, 62-64.
- [16] TanLi, M.Q., Jiang, Y., Wang, Y.L., Wang, X. and Peng, R.S. (2018) Digital Inspection of Cutting and Machining Based on Manufacturing Quality for Shop Floor. *2018 International Conference on Mechanical, Electronic and Information Technology (ICMEIT2018)*, Shanghai, 23-24 April 2018, 1-7. <https://doi.org/10.4236/jsea.2022.1511022>

Appendix

Table A1. Test cases of the function testing for “product or class” sheet unit.

ID	Input	Expected output
PQMS2-PDF-UNI-TC011	Start the product or class sheet from the menu item of “Basic quality data - Product or class”, and input “PC101” in the EditBox of “Code of product or class”, and input “Shanghai-K3-Loudspeaker” in the EditBox of “Name of product or class”, and keep “Mechanicaland electronic” in the EditBox of “Area or region”, and input “Pure electronic product” in the ComBoBox of “Type”, and input “15” in the EditBox of “Number”, and input “AD-PC101999” in the EditBox of “Code of assembly drawing”, and input “2020.03.01” in the EditBox of “Set-up date”, and input “MUSIC EQUIPMENT” in the EditBox “Mem”, and click the button “Add” finally.	Correctly displaying the prompt of “New data of product is added.” and no prompt of error information.
PQMS2-PDF-UNI-TC012	After TC011, input “ ” in the EditBox of “Code of product or class”, and click the button “Add”.	Displaying the error prompt of “The code of product or class cannot be empty.”
PQMS2-PDF-UNI-TC013	After TC011, input “ ” in the EditBox of “Name of product or class”, and click the button “Add”.	Displaying the error prompt of “The name of product or class cannot be empty.”
PQMS2-PDF-UNI-TC014	After TC011, input “PC10” in the EditBox of “Code of product or class”, and click the button “Add”.	Displaying the error prompt of “The number of code character of product or class is 5 and the beginning is ‘PC’, please refer to coding rule.”
PQMS2-PDF-UNI-TC015	After TC011, input “PC1010” in the EditBox of “Code of product or class”, and click the button “Add”.	Displaying the error prompt of “The number of code character of product or class is 5 and the beginning is ‘PC’, please refer to coding rule.”

Table A2. Test cases of the state testing for “product or class” sheet unit.

ID	Start state	End state	Input	Expected output
PQMS2-PDS-UNI-TC010	S ₀	S ₁	Start the product or class sheet with default.	Display this sheet.
PQMS2-PDS-UNI-TC011	S ₁	S ₅	No inputting operation, click the button of “OK” or “Shut-off” button of up-right corner in the sheet.	Exit.
PQMS2-PDS-UNI-TC012	S ₂₋₁	S ₂₋₁	Start the product or class sheet again, and input data in this sheet.	Display the data of inputting.
PQMS2-PDS-UNI-TC013	S ₂₋₁	S ₂₋₂	After data was input, click the button “Add”.	Save data, and display data in the “List control”.
PQMS2-PDS-UNI-TC014	S ₂₋₁	S ₄₋₁	Input “ ” in the EditBox of “Code of product or class”, and click the button “Add”.	Display the error message prompt.
PQMS2-PDS-UNI-TC015	S ₄₋₁	S ₂₋₁	Click the button of “OK” or “Shut-off” button of up-right corner in the prompt dialogue.	Return.
PQMS2-PDS-UNI-TC016	S ₂₋₁	S ₃₋₁	Modify data in the sheet, and click the button “Modify”.	Display the confirm dialogue.
PQMS2-PDS-UNI-TC017	S ₃₋₁	S ₂₋₁	Click the button of “Cancel” in the confirm dialogue.	Return.
PQMS2-PDS-UNI-TC018	S ₃₋₁	S ₂₋₂	Click the button “Modify” again, and click the button of “OK” in the confirm dialogue.	Save data and display data in the list.
PQMS2-PDS-UNI-TC019	S ₂₋₁	S ₄₋₂	Input “ ” in the EditBox of “Name of product or class”, and click the button “Modify”.	Display the error message prompt.
PQMS2-PDS-UNI-TC020	S ₄₋₂	S ₂₋₁	Click the button of “OK” or “Shut-off” button of up-right corner in the prompt dialogue.	Return.
PQMS2-PDS-UNI-TC021	S ₂₋₁	S ₃₋₂	Input the data for deleting, and click the button “Delete”.	Display the confirm dialogue.
PQMS2-PDS-UNI-TC022	S ₃₋₂	S ₂₋₁	Click the button of “Cancel” in the confirm dialogue.	Return.
PQMS2-PDS-UNI-TC023	S ₃₋₂	S ₂₋₂	Click the button “Delete” again, and click the button of “OK” in the confirm dialogue.	Complete the deleting of data.
PQMS2-PDS-UNI-TC024	S ₂₋₁	S ₄₋₃	Input the code with less character than specification, and click the button “Delete”.	Display the error message prompt.
PQMS2-PDS-UNI-TC025	S ₄₋₃	S ₂₋₁	Click the button of “OK” or “Shut-off” button of up-right corner in the prompt dialogue.	Return.
PQMS2-PDS-UNI-TC026	S ₂₋₁	S ₅	In the product or class sheet, when inputting data, click the button of “Cancel”.	Exit.

Table A3. Test cases of the data boundary value testing for “Inspection process” sheet unit.

ID	Input	Expected output
PQMS2-IDB-UNI-TC010	Starting the “Inspection process” sheet from menu item “Basic quality data - Inspection process”, and in the ComBoBox of “Code and name of product or class”, input character “T...” by keyboard with limited 51 times, and count actually if width is shorted, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the ComBoBox of “Code and name of product or class”.
PQMS2-IDB-UNI-TC011	Continuously, in the ComBoBox of “Code and name of part/component”, input character “T...” by keyboard with limited 51 times, and count actually if width is shorted, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the ComBoBox of “Code and name of part/component”.
PQMS2-IDB-UNI-TC012	Continuously, in the EditBox of “Code of inspection process”, input character “T...” by keyboard with limited 21 times, and stop when “Da” voice is listened.	Only 20 characters are permitted for input in the EditBox of “Code of inspection process”.
PQMS2-IDB-UNI-TC013	Continuously, in the EditBox of “Name of inspection process”, input character “T...” by keyboard with limited 51 times, and stop when “Da” voice is listened.	Only 50 characters are permitted for input in the EditBox of “Name of inspection process”.

Table A4. Test cases of the data format testing for “Inspection process” sheet unit.

ID	Input	Expected output
PQMS2-PDI-UNI-TC010	Starting the “Inspection process” sheet from menu item “Basic quality data - Inspection process”, and keep default data, and click the button “import inspection process data”. Consequently choose the folder of “INSPECTIONDATA/EXTERNAL BASIC DATA” in the follow-up dialogue—the format of file name and data is shown in Table A5 , and open the data file—“PC001000002 Height of hollow groove.txt”, and click button “OK” finally.	Prompt “the format is error—the tail of file cannot be empty”.

Table A5. The specification of data format for “Import inspection process data”.

Items	Format																								
The format of file name	(a) The front is the code of inspection process, has 11 characters. (b) Next is the name of inspection process. (c) The last part is the file name extension “.txt”.																								
	The general format																								
	The example with error																								
The format of data	<table border="0"> <tr> <td>1) The general string</td> <td>1) Caliper File. Interface.</td> </tr> <tr> <td>2) Number of data</td> <td>2) 7</td> </tr> <tr> <td>3) Accuracy—if various types, use “0”</td> <td>3) 0</td> </tr> <tr> <td>4) Set-up date</td> <td>4) 20220520</td> </tr> <tr> <td>5) Mem</td> <td>5) This data is generated by quality control software</td> </tr> <tr> <td>6) Data No.1—Code</td> <td>6) PC001000002</td> </tr> <tr> <td>7) Data No.2—Name</td> <td>7) Height of hollow groove</td> </tr> <tr> <td>8) Data No.3—Type</td> <td>8) Dimension of machining and cutting</td> </tr> <tr> <td>9) Data No.4—Code of drawing</td> <td>9) ID_PC001000002</td> </tr> <tr> <td>10) Data No.5—Date of design</td> <td>10) 2018-01-01</td> </tr> <tr> <td>11) Data No.6—Location and site</td> <td>11) Division of machining and cutting</td> </tr> <tr> <td>12) Data No.7—Mem of inspection process</td> <td>12) Digital indicator is bought from Shanghai by Tan Jianlin</td> </tr> </table>	1) The general string	1) Caliper File. Interface.	2) Number of data	2) 7	3) Accuracy—if various types, use “0”	3) 0	4) Set-up date	4) 20220520	5) Mem	5) This data is generated by quality control software	6) Data No.1—Code	6) PC001000002	7) Data No.2—Name	7) Height of hollow groove	8) Data No.3—Type	8) Dimension of machining and cutting	9) Data No.4—Code of drawing	9) ID_PC001000002	10) Data No.5—Date of design	10) 2018-01-01	11) Data No.6—Location and site	11) Division of machining and cutting	12) Data No.7—Mem of inspection process	12) Digital indicator is bought from Shanghai by Tan Jianlin
1) The general string	1) Caliper File. Interface.																								
2) Number of data	2) 7																								
3) Accuracy—if various types, use “0”	3) 0																								
4) Set-up date	4) 20220520																								
5) Mem	5) This data is generated by quality control software																								
6) Data No.1—Code	6) PC001000002																								
7) Data No.2—Name	7) Height of hollow groove																								
8) Data No.3—Type	8) Dimension of machining and cutting																								
9) Data No.4—Code of drawing	9) ID_PC001000002																								
10) Data No.5—Date of design	10) 2018-01-01																								
11) Data No.6—Location and site	11) Division of machining and cutting																								
12) Data No.7—Mem of inspection process	12) Digital indicator is bought from Shanghai by Tan Jianlin																								

Table A6. Test cases of unit function testing of adding inspection process data.

ID	Input	Expected output
PQMS2-IDF-UNI-TC030	Start inspection process sheet from the menu item “Basic quality data - Inspection process” in main program interface, and click the branch item of “Tree control” on the left—“TUMC - PC101Shanghai Brand K3 loudspeaker box - PC101000FrontRedNet - PC101000000 Plating flaw of FrontRedNet (0 10)”. EditBox of “Product Code and Name” display “PC101Shanghai Brand K3 loudspeaker box”, EditBox of “Part Code and Name” display “PC101000 FrontRedNet”, and choose this column in the “List control”. And in below input controls, alter the code to “PC101000001”—it has not been added, name to “Elasticity of FrontRedNet”, type to “Performance inspection of raw material”, division to “TUMC_ST StoreHouse”, code of drawing to “ID_PC101000001”, date to “2020.03.01”, Mem to “Shanghai”, then click the button “Add”.	Correctly prompt “Inspection process data of PC101000001 is added”, and the data item changed in the “List control”, and no other error information prompt.
PQMS2-IDF-UNI-TC031	After TC030, click the record “PC101000001” in the list, and display all data on the below box, then click the button “Add”.	Display error prompt “Data of PC101000001 has been already added”.
PQMS2-IDF-UNI-TC032	Start inspection process sheet, and input “ ” in EditBox of “Product Code and Name”, input “PC101000 FrontRedNet” in EditBox of “Part Code and Name”, and keep others no change, then click the button “Add”.	Display error prompt “Product code and name has not been input”.
PQMS2-IDF-UNI-TC033	Start inspection process sheet, and input “PC101Shanghai Brand K3 loudspeaker box” in EditBox of “Product Code and Name”, and input “ ” in EditBox of “Part Code and Name”, and keep others no change, then click the button “Add”.	Display error prompt “Part code and name has not been input”.
PQMS2-IDF-UNI-TC034	Start inspection process sheet, and input “PC101Shanghai Brand K3 loudspeaker box” in EditBox of “Product Code and Name”, and input “PC101000FrontRedNet” in EditBox of “Part Code and Name”, and choose this record in the “List control”, and input “ ” in EditBox “Code of inspection process”, keep others no change, then click the button “Add”.	Display error prompt “The code of inspection process has not been input”.

Table A7. Test cases of the state testing of “inspection process” sheet.

ID	Start state	End state	Input	Expected output
PQMS2-IDS-UNI-TC010	S ₀	S ₁	Start the “Inspection process” sheet from main interface.	Display the “Inspection process” sheet.
PQMS2-IDS-UNI-TC011	S ₁	S ₇	No inputting operation, click the “Shut-off” button of up-right corner in the sheet.	Exit.
PQMS2-IDS-UNI-TC012	S ₁	S ₂	Start the “Inspection process” sheet again, and open the category item of four layers.	Display correctly the opened item among four layers.
PQMS2-IDS-UNI-TC013	S ₂	S ₃	Click the item of inspection process of the category.	Display correct data of clicked item in the “List control”.
PQMS2-IDS-UNI-TC014	S ₃	S ₄	Click the record of inspection process in the “List control”.	Display all data of clicked record in below input controls.
PQMS2-IDS-UNI-TC015	S ₄	S ₃	Input correct data in the sheet, and click the button “Add” finally.	Save data, and display all data in the “List control”.
PQMS2-IDS-UNI-TC016	S ₄	S ₆₋₁	When the input of code is empty, click the button “Add” finally.	Prompt error information.
PQMS2-IDS-UNI-TC017	S ₆₋₁	S ₄	Click the button “OK” or the “Shut-off” button of up-right corner.	Return.
PQMS2-IDS-UNI-TC018	S ₄	S ₅₋₁	Modify data correctly in input controls, and click the button “Modify” finally.	Display the confirming dialogue of modifying.
PQMS2-IDS-UNI-TC019	S ₅₋₁	S ₄	Click the button “Cancel” of the confirming dialogue.	Return.
PQMS2-IDS-UNI-TC020	S ₄	S ₅₋₁	Click the button “Modify” of the sheet again.	Display the confirming dialogue of modifying.
PQMS2-IDS-UNI-TC021	S ₅₋₁	S ₃	Click the button “OK” of the confirming dialogue.	Save data, and display all data in the “List control”.
PQMS2-IDS-UNI-TC022	S ₄	S ₆₋₂	When the input of code is empty, click the button “Modify” finally.	Prompt error information.
PQMS2-IDS-UNI-TC023	S ₆₋₂	S ₄	Click the button “OK” or the “Shut-off” button of up-right corner in the dialogue.	Return.
PQMS2-IDS-UNI-TC024	S ₄	S ₅₋₂	Input the data for deleting in input controls, and click the button “Delete” finally.	Display the confirming dialogue of deleting.
PQMS2-IDS-UNI-TC025	S ₅₋₂	S ₄	Click the button “Cancel” of the confirming dialogue.	Return.
PQMS2-IDS-UNI-TC026	S ₄	S ₅₋₂	Click the button “Delete” of the sheet again.	Display the confirming dialogue of deleting.

Continued

PQMS2-IDS-UNI-TC027	S5-2	S3	Click the button “OK” of the confirming dialogue.	Delete data, and display variety in the “List control”.
PQMS2-IDS-UNI-TC028	S4	S6-3	When the input of code is empty, click the button “Delete” finally.	Prompt error information.
PQMS2-IDS-UNI-TC029	S6-3	S4	Click the button “OK” or the “Shut-off” button of up-right corner in the dialogue.	Return.
PQMS2-IDS-UNI-TC030	S4	S5-3	Input data in input control for “Search-name”, and click the button “Search-name” finally.	Execute search, and display the search result in the list.
PQMS2-IDS-UNI-TC008-AD	S5-3	S4	Click the button “OK” or the “Shut-off” button of up-right corner in the dialogue.	Return.
PQMS2-IDS-UNI-TC031	S4	S6-4	When the input of name is empty, click the button “Search-name” again.	Prompt error information.
PQMS2-IDS-UNI-TC032	S6-4	S4	Click the button “OK” or the “Shut-off” button of up-right corner in the dialogue.	Return.
PQMS2-IDS-UNI-TC033	S4	S4	Click the button “Reset” of the sheet.	Initialize all input controls in the sheet.