University of Nottingham

UK | CHINA | MALAYSIA

COMP2043. GRP Maintenance Manual

Visualization and Validation Tool for Dynamic Models of Credit Risk

Contents

1. Summary of Quality Assurance

❖ Requirements Engineering

Understanding product requirement is the major problem in software development. Requirements are the foundation of one project. It is important to make it clear before coding. The requirements and specifications of the software have been modified and optimized several times after each formal meeting.

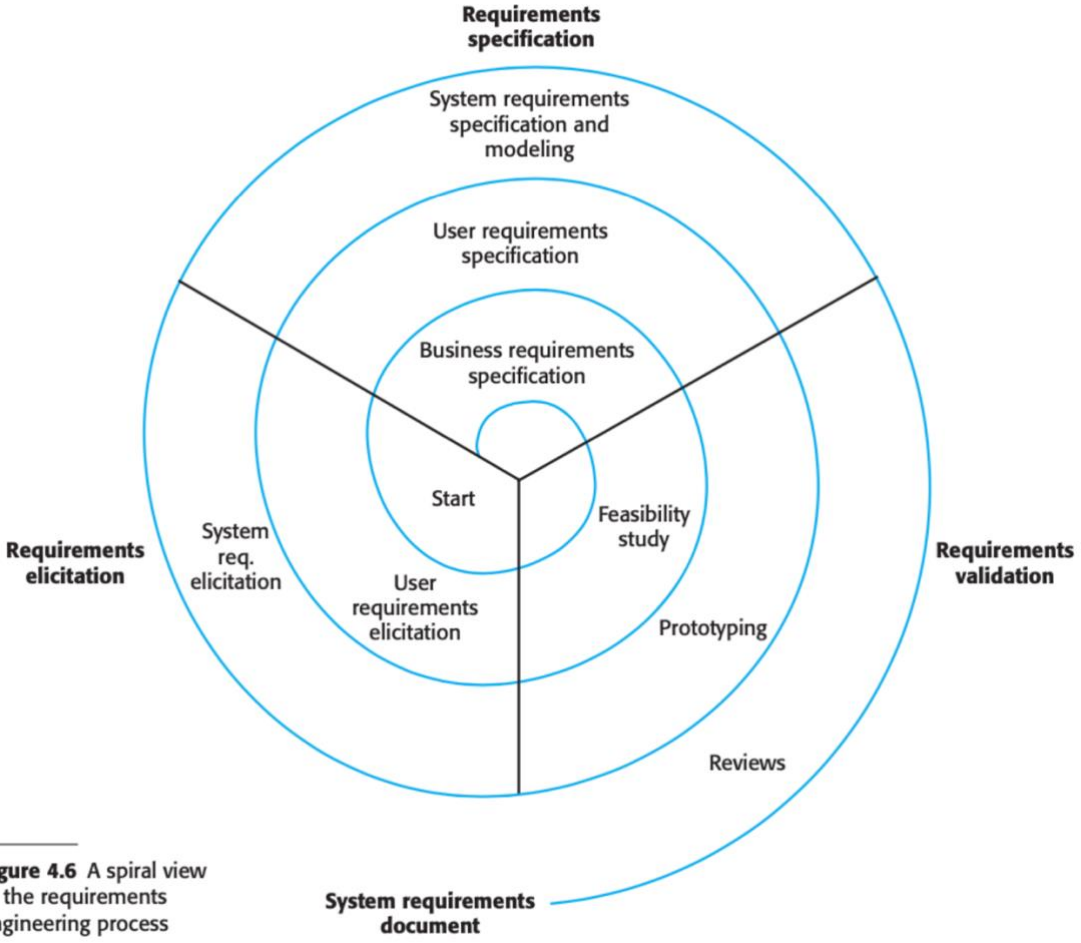


Figure 4.6 A spiral view of the requirements engineering process

Several methods have been used in requirement engineering: Stakeholder Analysis and Personas and UML Diagrams.

- Stakeholder Analysis and Personas

Before coding, all the people that will use the software have been determined. This benefits the development direction. And then, some personas are made to represent a real type of user from stakeholders.

- UML Diagrams

Use Case Diagram, Activity Diagram, and Sequence Diagram, three Behavior Diagrams were developed during the requirement engineering process.

1) Use Case Diagram

Use Case Diagram represents the people who use the software and the tasks they have to perform. This helps us to determine the code hierarchy.

2) Activity Diagram

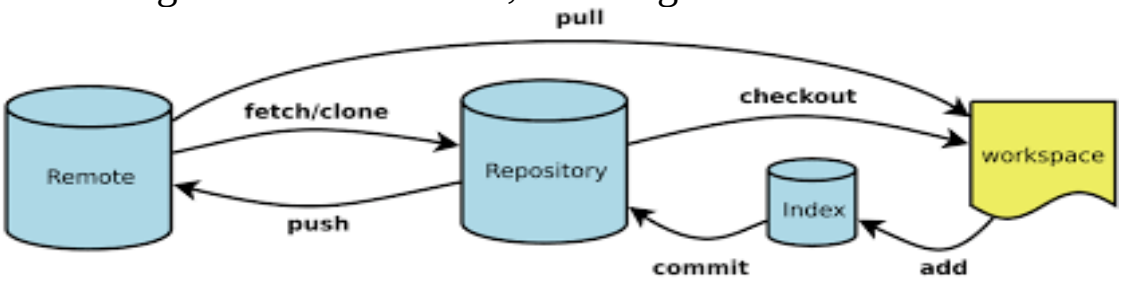
Activity Diagram is used to define one Use Case in more detail. This helps us to determine the specific code function in the coding process.

3) Sequence Diagram

This diagram intuitively displays the time sequence of the user-system interaction and system-database information transfer. It can be used to specify function calls in Java objects, determine the interaction logic between different classes.

❖ Source Code Control

To support multi-person collaborative development, git is used, to achieve version control. It can keep a history of everything and support work on branches. It helps us during the code development. For example, it allows all team members to work at the same time, greatly saving the development time. Moreover, code conflicts between our team members can be easily resolved. Besides, if there are some problems at a certain stage of software development, it is easy to go back to the original correct version, reducing the cost of errors.



❖ Build Files – Maven

Maven is a build automation tool for software management and comprehension. Project Object Model (POM) can manage the use of

dependencies and a project's build. POM is a basic work unit of a Maven project. POM is an XML file, which includes the basic information and dependency of the project. Maven handles the conflicts between Jar files. Maven can also convert the project code into an executable Jar file. The Jar file can be executed on any local machine with proper environment settings. In our project, before the software is finally converted into an exe installation package, Maven publishes and generates executable Jar file. Besides, some plugins and dependencies are managed by Maven.

❖ Design Pattern

We have used several design patterns (e.g. Model - View - Controller (MVC), Singleton) to make our code more organized and easy for all team members to develop and maintain. These kinds of design patterns could help developers write code faster by providing a picture of how you are implementing the design.

- Singleton Design Pattern

Singleton restricts the number of instantiation of a class, which means only one instance of the class exists in the java virtual machine. During our development process, it helps to save memory and reuse code.

- MVC Design Pattern

MVC separates user interface and related program logic into three interconnected elements: Model, View, and Controller. Because all team members need to develop the code simultaneously, and each of us has different area of expertise (some are good at the back-end Python model development, some are skilled in user interface front-end), it benefits us to utilize our strengths. Moreover, it greatly facilitates code integration and interaction.

❖ Testing

Testing is very important for both developers and users. For developers, it is a good way to debug and optimize the code. Besides, it makes developers think about how the code is used, before building it. For users, software testing can provide objective, intuitive information about the quality of software.

Several kinds of testing have been used in the testing process: Unit test, Functional test, and User Acceptance test.

- Unit test

Junit test and TestFX have been used during the Unit test process.

1) Junit Test

For the InputField class, we adopt Junit test, because this part of the code is designed with clear input and output value. All the Junit cases of this class are passed.

2) TestFX

TestFX is an automatic test platform, which is simple and clean testing for JavaFX. Because the interface of the software is mainly developed using JavaFX, TestFX is a suitable method that can be treated as a simple robot to simulate user interactions. It can be used to test whether each control is executed according to the intention of developers. All the TestFX cases are passed.

- Functional Test

The purpose of Functional Test is to test each function of the software application by providing appropriate input, verifying the output against the functional requirements.

First prototype：
System test time:2021/3/16

Part 1: Sign in interface:

Clicking terms view button, whether it will jump to terms viewing page or not. True

Clicking Data Policy view button, whether it will jump to Data Policy viewing page or not. True

Clicking About Us view button, whether it will jump to About Us viewing website or not. True

Clicking "Can' t access your account?" button, whether it will jump to Verification interface or not. True

Clicking "sign up now!" button, whether it will jump to sign up page or not. True

Entering correct user information and clicking sign in button, whether it will jump to main menu. True

Part 2: Verification interface

Entering appropriate user email address and clicking "send code" button, whether the verification email will be sent. True

Entering correct Verification code, appropriate new password and confirming password after getting verification code in email, then clicking "Create new password" button, whether successful reset interface will show up and user can use new password to login. True

Clicking "Back to sign in" button, whether it will jump back to sign in interface. True

Part3: main menu

Double clicking head portrait in the upper right corner, whether user information interface will show up. True

Clicking the bin sign in the frame of Uploaded data sets, whether the confirmation dialog interface will show up. True

Clicking "confirmation" button after confirmation dialog of user information showing up, whether the corresponding datasets will be deleted. True

Clicking "cancel" button after confirmation dialog of user information showing up, whether it will jump back to user information interface. True

Clicking "help" button, whether user guide, about us and about beaver option button will show up. True

Clicking user guide, about beaver and about us button after clicking help button, whether user guide interface and about beaver interface will show up or it will jump to about us website. True

Selecting one data set, one model and one scenario/no stress test, then clicking model application button to model fit, whether progress bar will load successfully to show the loading progress of model fit and whether the DR_chart and Loglike_chart will show up clearly. True

Model fitting successfully, whether the accurate attribute valued in the statistic table will be showed. True

Clicking "view" button, whether show toolbar, show content as tree and show chart symbols option button will show up. True

Clicking show toolbar, show content as tree and show chart symbols option button after clicking view button, whether the toolbar/ content in tree form/ chart symbols will show up/hide. True

Double clicking the option button (attached with specific graphs) below my models, my data sets, my scenarios, and history records in toolbar, whether interface containing DTSM python model, specific data sets interface, scenario information interface and recent records showed in DR_chart will show up. True

Clicking with right mouse on specific model, datasets, scenario and records button, whether open, rename, delete and favour button will show up. True

Clicking open button after clicking specific model item with right mouse, whether it will jump to the interface containing DTSM python model. True

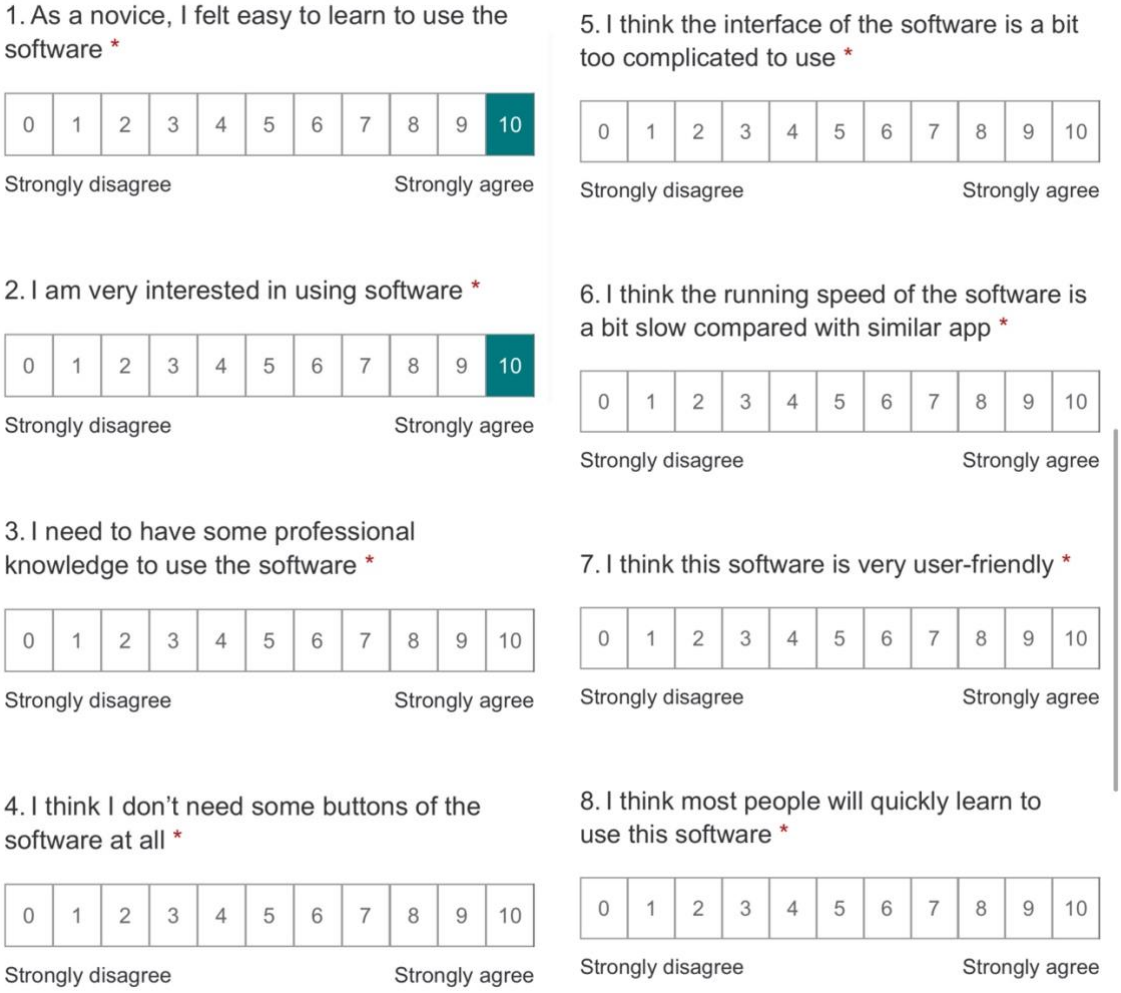
Clicking open button after clicking specific datasets item with right mouse, whether specific data sets interface will show up. True

Clicking open button after clicking specific scenario item with right mouse, whether scenario information interface will show up. True

Some functional test results are shown in the figure above.
In Software Testing, Functional testing is a process of testing functionalities of the system and ensures that the system is working as per the functionalities specified in the business document. The goal of this testing is to check whether the system is functionally perfect! All functional tests are passed.

- User Acceptance Test

User Acceptance Test was taken place when the whole development process is completed. It is the final state before the software is accepted. The test was done by some target users. After users finish testing, they need to fill out a questionnaire like follows for subsequent analysis by developers.



2. Environment Requirements

❖ Operating System

Beaver is designed based on Windows X64 operating system and has not been deployed on Linux or Macintosh OS systems.

❖ Setup Environment Variables

To successfully run Beaver, the following environment variables need to be installed:

- Java Runtime Environment version1.8

Beaver is developed as a Java project and needs to be run on Java Runtime Environment 1.8 (JRE 1.8) or later versions. Or users can install the Java Development Kit 11 (JDK 11) or later version as JRE 1.8 is included in it. Users can download the above from https://www.oracle.com/java/ and choose the required products.

- Python

Because dynamic models are built in Python, which means the core part of credit risk analysis is implemented in Python. Java Runtime method is used to execute Python script, which makes code productive and dynamic. The main idea of the Java Runtime method is to execute Python script through the command line and get the execution result from the script output stream that should have been printed in the console. However, some compulsory third-party libraries are not available. As a result, you need to install the required libraries manually. To do this, carry out the following steps:

```
pip install numpy
pip install sklearn
pip install pymysql
```

Attention: If you use other third-party libraries in Python model script, install these libraries in the same way.

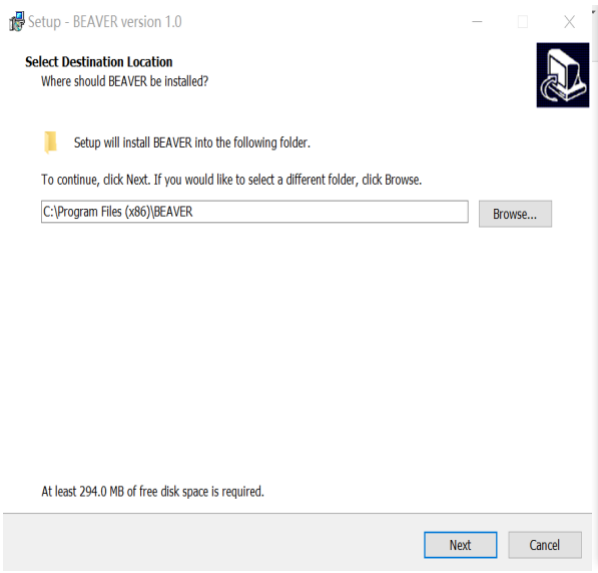
3. Installation Instructions

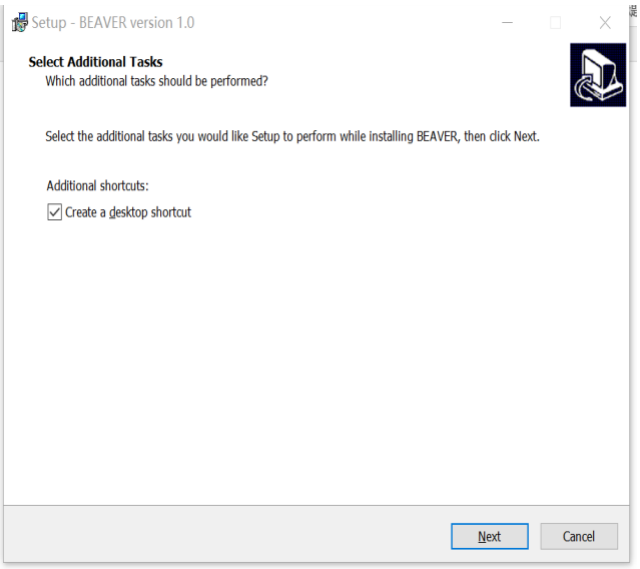
❖Download Beaver

Download the software from:
http://cslinux.nottingham.edu.cn/~Team202001/

❖Install Beaver

Open setup.exe you just downloaded, select the path you want to store the software. And then, follow the installation guide step by step.





❖Finish installation

When the installation steps are as shown in the figure below, you can launch Beaver software.