SID – The Smart Investment Dashboard

C. Malcolm Todd – T00232792
Saloni Saluja – T00608615

SOFTWARE REUSE PLAN

PREPARED FOR:

Mr. Kevin O'Neil, TRU Computing Science
koneil@tru.ca

CONTENTS

EXECUTIVE SUMMARY

This report serves to define the security engineering practices to be employed during the development of SID, the Smart Investment Dashboard, a software solution for self-directed investors to improve the process of self-directed investing. In developing SID, we will utilize an Agile software development process. We will also implement a thorough testing strategy using a combination of Test-Driven Development to create a battery of tests, including unit tests, system tests, and release tests, and with thorough regression testing. As a small organization, we will adhere to a strong software maintenance and evolution schedule that will be defined using aspects of the 'Agile_MANTEMA' strategy, which combines the Scrum project management mechanism with the MANTEMA strategy used in the maintenance of large projects [1]. Concerning SID as dependable software, our organization's selection of Agile methods and limited developer resources result in our inability to produce sufficient documentation as required by a dependable process; however, we will seek to provide SID with a measure of dependability.

In context of software reuse, we will describe what software reuse is, and why we should consider building SID on top of an already existing software. In addition to this, we will discuss Wealthica, which is the software that we will extract various components and functions from. Next, we will move on to describe the various levels of software reuse, and those that we will be using for SID. Wealthica also provides powerful security features, so we will be using some of them in SID as well. Furthermore, we talk about two additional software applications – Google Spreadsheet Export Add-On and Wealthscope that will provide added functionality to our users.

In order to combat developer arrogance and 'Not Invented Here' (NIH) Syndrome, our organization will outline policies and strategies with which we can ensure that SID conforms to the above planned software reuse levels and within our future development. This will be a critical factor in designing and implementing SID given our circumstances as a small start up organization that must cut unnecessary costs wherever possible.

SOFTWARE REUSE

What is Software Reuse?

In most engineering disciplines, systems are designed by composing existing components that have been used in other systems. It is now recognized that to achieve better software, more quickly and at lower cost, we need a design process that is based on systematic software reuse. There has been a major switch to reuse-based development over the past 10 years [2]. Therefore, in order to utilize pre-existing software and software components for our dashboard, SID, we plan to make use of a software named Wealthica.

Why use Software Reuse?

Accelerated Development: Reusing software can speed up system production as both development and validation time may be reduced [2].
Effective use of specialists: Instead of doing the same work repeatedly, we, as developers, can develop reusable software that encapsulates our knowledge [2].
Increased dependability: Reused software, which has been tried and tested in working systems, should be more dependable that new software. Its design and implementation faults should have been found and fixed [2].
Lower development costs: Development costs are proportional to the size of the software being developed. Reusing software means that fewer lines of code must be written [2].

About the Wealthica Software System

Wealthica is the Canadian version of Personal Capital. It is also a very popular app in the United States. Wealthica syncs with more than 60 Canadian financial institutions so that the users can track all their investments, bank accounts, and other financial assets in one place. Wealthica is designed to make life easier by allowing the users to stay on top of their investment accounts by connecting with their respective financial institutions to save their balances, holdings, and transactions history. Wealthica also provides the users with access to income, gains, and other investment reports. Wealthica is an aggregation platform that allows investors to see all their investments in a single dashboard [3].

SOFTWARE REUSE LEVELS IN SID

There are several different levels or scales at which software can be reused. They are mentioned as under:

I. System Reuse

In system reuse, complete systems, which may include several application programs may be reused [2]. For our dashboard, SID, we have decided that rather than using Wealthica as the base software, we would use aspects/features or components of Wealthica that we think would be the most beneficial for our customers. Therefore, we won't be making any use of "System Reuse".

II. Application Reuse

An application may be reused either by incorporating it without change into other or by developing application families [2]. For additional third-party add-ons, we are considering two software Google Spreadsheet Export Add-On and Wealthscope [4]. These features are going to be included for free in the premium version of SID. However, customers can have one or both above-mentioned add-ons in the original version of SID for a small fee.

Google Spreadsheet Export Add-On automatically updates investment spreadsheet each day. In addition to this, it supports export and synchronization of transactions and holdings automatically [4]. The monthly fee for this addition would be only $5.00.

Fintechs like Wealthscope can enhance the features of SID using the add-ons marketplace. Wealthscope powerful portfolio analytics can optionally be added to SID for a small subscription fee [4]. This time, we plan to charge $13.00 per month from our customers. To access Wealthscope, the user would first need to create a free account on SID. Once signed in, it would be mandatory to link accounts on SID and Wealthscope. The user must then subscribe to the Wealthscope Add-on which would ultimately allow the user to access their portfolio scorecard and other valuable Wealthscope tools.

III. Component Reuse

Components of an application from sub-systems to single objects may be reused [2]. Majority of the components in SID are going to be derived from Wealthica. As mentioned earlier, we are not going to use Wealthica as the base application in order to develop SID. However, we are going to extract and use certain components and feature set provided by it. Some of the components provided by Wealthica that will be reused in SID are as follows:

SOFTWARE REUSE CONT'D

Bank-level security

Wealthica uses the latest technologies and has strict processes in place to protect the customers' financial information. The credentials are encrypted, and even the employees cannot access them. In addition to this, the Database Administrators also do not have access to the encryption keys used [5].

Enhanced login protections

Wealthica uses a two-factor authentication in the user's account for enhanced security. When enabled, a security PIN is sent to the user's smartphone to authorize login from a new device [5]. However, it becomes a matter of concern if the user doesn't have his/her smartphone around. So, for SID, we would provide users with the option of either sending the security PIN to their cell phone or their email ID, whichever is more convenient for them at that moment. This way, even if a hacker gains access to the user's Wealthica credentials at hand, it will be impossible for him/her to access the user's financial information and account balance.

No one can move money

Wealthica only reads and saves the user's brokerage data to build a visual dashboard of all their financial investment. Trading, funds withdrawal, money transfer, or other transaction is impossible through the platform to keep the users' money safe [5].

Identity verified

Network Solutions validated Wealthica as an organization and verified their identity. In addition, SiteLock verifies their site security every day and protects them from span, viruses, and scams [5]. For SID, in addition to SiteLock, we would also be using nsProtect (Network Solutions Protected) seal which would provide our users with an assurance of our legitimacy which has passed nsProtect's identity validation requirements. This guarantee will increase the user's trust and confidence.

Fraud Detection

After the users connect their investment accounts to Wealthica, they can easily review new transactions across all their accounts and detect any suspicious activity. They can also choose to receive a daily review of new transactions by email, making it easy to monitor their accounts for suspicious activity [5]. To enhance security, we would enable users to always receive an email and/or a text message on their phone whenever there is a login from a new device which has never been used before, or a new location, and a user's login history will always be available within SID.

Security Questions and Transaction PINs for protection

In most cases, the user's credentials are needed to establish a secure connection between Wealthica and their financial institution. In the process of adding financial institutions to Wealthica, the user will be asked to enter their credentials and answer security questions. Wealthica will only securely store the user's credentials, and never store the answers to their security questions (or transaction PINs). Therefore, if someone steals the users' credentials and attempts to login from a new device, they would be asked to provide those answers, thus preventing them from logging in [5]. We would also be using these social engineering techniques for SID.

IV. Object and Function Reuse

Small-scale software components that implement a single well-defined object or function may be reused. [1] One function provided by Wealthica that particularly impresses us the most is the use of cloud services from Amazon Web Services (AWS) located within the Canadian region, in Montreal [5]. The users' personal data is stored in Canada itself. However, Wealthica still uses some services provided by AWS in the US to process some of the users' data. These services were and are still not available in Canada [5]. For SID, we will modify this function by trying to migrate all services in Canada as soon as possible.

OVERCOMING NIH SYNDROME IN SID

One of the primary barriers to software reuse stems from software developers themselves through what has come to be known as 'Not Invented Here' (NIH) Syndrome which afflicts many programmers [2]. The main cause for this syndrome is the arrogance that often develops in programmers as a result of their personal pride, and from the stream of compliments they often receive due to the problem-solving nature of their positions [6]. In this section we will examine this barrier and how it can be prevented from forming.

What is NIH Syndrome?

NIH Syndrome refers to the tendency of individuals and organizations to reject a viable external component or solution that fulfills their need in favor of creating their own new custom tool [7]. NIH syndrome can also be described as a strong resistance to, or automatic rejection of, any ideas and concepts that are externally generated [8]. The source of this syndrome typically stems from a developer's nature as developers would prefer to write their own code, rather than read someone else's [7]. In looking for a scientific origin for NIH, some believe that it's origin can be tied to a person's interest in preserving their mental resources which creates this natural resistance to new ideas so as to not require energy to expand them [8].

How we can Reduce and Prevent NIH

To prevent and reduce NIH within the organization, we will monitor our development efforts to record what external exploration was completed with each piece of functionality as this can help to detect those activities with insufficient external research. These may also be the activities in which a 'wheel is being re-invented' which results in wasted time [7]. To reduce developer arrogance, it will also be important that our developers maintain an awareness of available external solutions over time. In being aware of what is available, and by extension useful, our developers can reduce the likelihood of falling into the belief system of 'if you want something done right, do it yourself' which invites NIH [7].

Other strategies that our organization will employ include the definition of a reasonable utility threshold up to which an external component must perform; however, this threshold should also not be too stringent as to require a perfect external component [7]. For example, if a component can accomplish at least 90% of our desired functions, then it is acceptable for use. Finally, we can also implement a minimum external required ratio to be maintained within SID. This is a control that has been used by other companies such as P&G to help enforce the concept of reuse in their product development [8].

CONCLUSION

To conclude, we started with the discussion of what software reuse is, and why we should consider building SID on top of an already existing software. Additionally, we went on to discuss what the various levels of software reuse are – Systems, Application, Component, and Object and Function. Our intention is to not make use of Systems level Reuse, because we don't want to use the entire Wealthica system as a base application. However, we intend to make use of Component Reuse by using some of the security features provided by Wealthica for bank-level security, security for transactions, identity verification, and fraud detection. Regarding Application Reuse, we would provide our users with the option of integrating software applications such as the Google Spreadsheet Export Add-On and Wealthscope for added functionality. Finally, for Object and Function Reuse, we will be making use of cloud services from Amazon Web Services (AWS).

With regards to NIH Syndrome and developer arrogance, the tendency towards re-inventing the wheel with many software components, can greatly increase unnecessary costs within an organization such as ours. This is of particular concern given our limited resources as a start up company. In order to combat the likelihood of NIH occurring, our organization will mandate a minimum amount of functionality within SID that must come from external sources. Additionally, our developers will be required to conduct regular research on new software within the domain as well as document a certain amount of research relating to each new component. Through the above organizational policies, we will improve our ability to identify areas wherein 'wheels are being re-invented', so that they can be addressed before becoming to costly.

REFERENCES

[1] F. J. Pino, F. Ruiz, F. Garcia and M. Piattini, "A software maintenance methodology for small organizations: Agile_MANTEMA," *JOURNAL OF SOFTWARE MAINTENANCE AND EVOLUTION: RESEARCH AND PRACTICE,* pp. 851-876, 2012.

[2] I. Sommerville, Software Engineering (10th Edition), Pearson, 2015.

[3] Wealthica Financial Technology Inc., "We are Wealthica Financial Technology Inc.," Wealthica Financial Technology Inc., 2019. [Online]. Available: https://wealthica.com/about-us/. [Accessed 12 March 2020].

[4] Wealthica Financial Technology Inc., "Pricing," Wealthica Financial Technology Inc., 2019. [Online]. Available: https://wealthica.com/pricing/. [Accessed 12 March 2020].

[5] Wealthica Financial Technology Inc., "Organize your Financial Life Securely," Wealthica Financial Technology Inc., 2019. [Online]. Available: https://wealthica.com/security/. [Accessed 12 March 2020].

[6] M. Jones, "The Programmer's Plague: Fighting the Spread of Arrogance," Exception Not Found, 10 March 2015. [Online]. Available: https://exceptionnotfound.net/the-programmers-plague-fighting-the-spread-of-arrogance/. [Accessed 8 March 2020].

[7] M. Nash, "Overcoming "Not Invented Here" Syndrome," Developer.com, 12 April 2004. [Online]. Available: https://www.developer.com/design/article.php/3338791/Overcoming-quotNot-Invented-Herequot-Syndrome.htm. [Accessed 8 March 2020].

[8] M. E. May, "Overcoming the Not-Invented-Here (NIH) Mindset," Innovation Excellence, [Online]. Available: https://innovationexcellence.com/blog/2018/05/08/overcoming-the-not-invented-here-nih-mindset/. [Accessed 8 March 2020].