SID – The Smart Investment Dashboard

C. Malcolm Todd – T00232792
Saloni Saluja – T00608615

SOFTWARE PROCESS MODEL

PREPARED FOR:

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

CONTENTS

EXECUTIVE SUMMARY

This report serves as an introduction to our intended software project which we have named the Smart Investment Dashboard or SID. SID will function as a resource within the Self-directed investment strategy to help mitigate or remove the strategy's disadvantages, and to provide them with some of the advantages of the Passive investment strategy.

Throughout the development of this software, we have chosen to employ the Agile software design process. The Agile process is well suited to small teams such as our organization which makes it valuable. Also, during the research we conducted on dashboard-based software, we found multiple resources which stated that the Agile process can be applied well in the development of dashboard type software [1]. This is as a result of the incremental releases which allows the dashboard to be improved or to release new functionality continuously over the software's lifespan [2].

INTRODUCING SID, A DASHBOARD

SID, The Smart Investment Dashboard

For our design project, we have chosen to create the Smart Investment Dashboard or SID. This project will explore the world of Self-directed investment and where software supports can improve investors results. This improve will stem from the software's ability to make investors more effective in their efforts using a dashboard of useful information and investment tools. These tools will allow for the evaluation their holdings and provide better clarity on cost implications to their activities. Before we begin to explore the focus of these tools, we will explore the types of investor strategies that exist in today's environment.

Types of Formal Investor

Within the retail investor space, a user can be described as employing one of two strategies when investing. These two strategies are the Passive investment and Self-directed investment strategies. For the purposes of SID's design, we will focus on Self-directed investors.

The passive investment strategy is the strategy which is supported by most financial institutions [3]. In this strategy, the investor typically follows all the basics of sound financial planning and the strategy is good for investors with busy lives, families, and outside interests [3]. Conversely, the Self-directed investment, or Do-it-Yourself (DIY) strategy, involves the investor taking a more direct and active role as described below.

When using the DIY strategy, the investor is responsible for the management of their own investment portfolios, typically through a discount brokerage [4]. The DIY approach comes with several advantages over the passive approach, but includes disadvantages as well. With regards to advantages, an investor may reduce their investment fees, gain the convenience of direct online trading, and have greater control over their portfolios [4]. Unfortunately, with these advantages comes an increase in responsibility, due diligence needs, and opportunity for errors on the part of the investor. To counteract these disadvantages, the investors needs to conduct their own research, track their trading costs, and ensure they don't make trading errors which can become expensive to correct [4].

SID AS A BUSINESS OPPORTUNITY

The Opportunity for SID

The increased workload demanded by the DIY strategy can prove challenging for many investors; however, we believe that the addition of software like SID can greatly streamline and provide greater clarity to investor efforts. Depending on the functionality provided, SID could also enable DIY investors to get some of the benefits offered under the Passive investment strategy.

By reducing and mitigating the disadvantages of this strategy, SID may attract a larger number of investors to switch investment strategies so as to enjoy the DIY advantages. To greater define the domain in which SID will operate, we will explore the responsibilities of the Self-directed strategy in greater detail.

Under the Self-directed strategy the investor becomes responsible for the following [4]:

- **Developing their own investment plan**
- **Learn the basics of trading**
- **Understanding fees**
- **Regularly monitor and manage investments**
- **Deciding on their investment holdings**

These aspects present the main cause of investor errors in the DIY approach through a lack of sufficient understanding or planning. Specific instances of these errors as incurring too many fees from excessive trades or picking investing holdings that do not align with their goals [4]. To be of value in preventing these errors SID could provide fee tracking and estimation, as well as portfolio evaluation services which DIY investors typically do not receive. SID could also be structured to include an education aspect to speed the investor's learning curve for new entrants to the investment environment.

AGILE SOFTWARE DEVELOPMENT

What is Agile Software Development?

Agile Software Development is a process that arose during the 1980s and 1990s as a result of dissatisfaction in the amount of process overhead in plan-driven software development processes [5]. Another significant different in Agile over plan-based processes was the emphasis on rapid development where the goal was to divide software into a series of releasable pieces allowing for earlier and incremental delivery [5]. One key difference to Agile is that it sought to focus on assisting teams to respond to the unpredictability which can arise during software development [6]. This ability to respond to unpredictability and change is made possible through the process' structure which is to allow alternating between common phases such as program specification, design, and implementation.

Another defining characteristic of the Agile process is to employ incremental and iterative work sequences which are commonly known as 'sprints' [6] Sprints are defined as a period of time allocated to a particular phase of a project's development and are considered complete when the period expires at which point development moves on to a new phase of the project [6].

Within the Agile process there also exist many varieties through varying methodologies with a focus on different aspects. Some of the better-known varieties are the Agile Scrum Methodology, Lean Software Development, Extreme Programming, and Kanban. [6]
This ability to respond to unpredictability and change is made possible through Agile's structure which is to allow alternating between common phases such as program specification, design, and implementation.

The Key Principles of Agile

While each of the varieties of Agile do vary in their implementation, they do share commonality in certain key principles which are listed below:

- Reprioritization: Satisfy the ever-changing needs of the client during development [5].
- Customer Involvement: Developers should work along with the customer throughout the entire project [6]
- People over Process: Self-organizing teams usually produce the best designs [6] [5]
- Collective Ownership: Pairs of developers work on all aspects of a system, "anyone can change anything [5]
- Refactoring: Maintain simplicity and work to eliminate system complexity [5]
- Incremental delivery: Rapid and frequent delivery of software to the customer [6]

AGILE PROCESS (CONT)

The Strengths of Agile

While there are some critics of the Agile Method, this method is still employed in the majority of development projects. One reason for this is that Agile often produces the best overall results in small to medium sized projects [5].

The key strengths of the Agile process are its adaptability and speed. This adaptability permits teams to correct errors earlier on in the design process which reduces costs. It also helps to correct the project when the delivered software may not turn out to be exactly as the client had originally envisioned [6]. Often during development, the requirements of the client keep on varying throughout the process and it is the responsibility of the developers to accommodate those changes to ensure that the clients are ultimately happy with the results [6].

Agile also allows for shorter time to delivery by breaking the software development into phases which can result in earlier revenues with which to fund further development reduces the project's risk to the software company [5].

The Weaknesses of Agile

Agile can prove very strong in many project environments; however, it is not without weaknesses as not all projects can employ the Agile process. Two notable weaknesses are scaling up Agile in larger companies and projects, and the challenge of implementing Agile in existing companies with a focus on strong bureaucratic processes [5].

Some outcomes that can occur when the Agile process is implemented poorly can be seen below:

- **Agile pushes the developers to rush in order to do their job as there is an extremely high amount of pressure on them, less time to do the work, and demands to "go faster".**
- **Agile does not focus much on the product design [7]. In turn, there is no room for developer creativity. The primary focus of Agile lies on requirement elicitation from the customer and developing code based on more requirements.**
- **Using Agile can pose a challenge if the team members have mixed levels of experience. If Scrum is not understood and enacted upon properly, even though it preaches cross functionality, having a hierarchy may lead to division between who gets to do what [8].**

WHY WE CHOOSE TO USE AGILE

Our reasons for selecting the Agile software development process was due to the nature of our current organizational structure and the type of software in which SID will be implemented. Our use of the Agile process will use a combination of Scrum's sprint structure, and Extreme Programming's incremental delivery concept minus the multiple daily builds rather than one specific variety of Agile.

Using Agile in Dashboard Development

As a dashboard, SID will be a GUI based information management tool that provides relevant information and a progress report to a user at a glance [2]. The steps involved in dashboard design also closely mirror the design phases in Agile by repeatedly iterating through cycles of eliciting client requirement, designing the dashboard, and evaluating the impact [2]. "Dashboards should always be developed iteratively in increments with the customer's commitment to be in close collaboration" [2], which aligns to a key principle of the Agile process. This iterative approach is key to the planning and feedback which is necessary to deliver quality dashboard software that reflects the needs of the customers [2].

Creation of a dashboard is often implemented over a series of releases in which new functionality is added or improved. The Agile Method empowers optimization of the release during its development. "It preserves relevance of the critical market and ensures that a team's work doesn't wind up collecting dust on a shelf." [6]. A dashboard supports the business needs through a visual representation of the data contained within a database which needs to be maintained [7]. This requirement of successive releases, regular maintenance, and incremental releases is also well accommodated by the Agile process [5].

Agile in Our Current Structure

Based on to our organizational structure, we are well aligned to capitalize on many the strengths of Agile. Our small size also causes many of the weaknesses to Agile, such as the scaling issues, to be a non-issue, and allows us to focus on the proper implementation of the process instead. As a team of two members, the light framework provided by Agile will help us lay our focus on rapid delivery and reduction of unnecessary overheads to reduce the overall risks of software development. Also, the small nature of our team does not allow for role separation and specialization, so Agile's principle of collective ownership will help ensure good collaboration and shared knowledge between team members allowing us to better maintain SID over its lifespan [5].

CONCLUSION

In conclusion, we found that investors employ either the Passive investment strategy or the Self-directed investment strategy wherein the users take a more active role. For those self-directed investors, the strategy comes with the opportunity to gain flexibility and reduce their management costs at the expense of a higher personal burden on the investor. It is this increased burden on the investor which presents the opportunity to create SID, by providing a means for these self-directed investors to become more effective.

The Agile software development process gained in popularity in the 80s and 90s with the goal of being better suited to the fast paced and unpredictable nature of software development and to better handle changing requirements. Many varieties of Agile exist; however, they share common principles which is to provide incremental delivery of software, maintain customer involvement, and a focus on constant improvement. The Agile process is typically well suited in small to medium sized teams, but can prove challenging to scale up in larger organizations and those that emphasize bureaucratic processes.

The reason for which we chose to employ the Agile software development process was a combination of being well aligned for the development of dashboard-based software and for those organizations that had a similar structure to our own. Best practices in Dashboard software development software follows an incremental approach with phases very similar to those found in Agile processes. As a small organization, we are also well positioned to take advantage of Agile's key principles and strengths without the need to concern ourselves with many of it's weaknesses. Through incremental release, a focus on people rather than process, and no need to scale to a large project at this time, a combination of existing Agile methods provides an ideal process for the development of SID.

BIBLIOGRAPHY

[1] C.-E. Lavoie, "How to Build a Dashboard Using Agile Methodology [Part 1]," agileDSS, July 2013. [Online]. Available: https://www.agiledss.com/en/blog/how-build-dashboard-using-agile-methodology-part-1. [Accessed 19 January 2020].

[2] M. Staron, "Dashboard development guide," Chalmers University of Technology and University of Gothenburg, Göteborg, Sweden, 2015.

[3] T. Tresidder, "3 Types of Investor," Financialmentor.com, [Online]. Available: https://financialmentor.com/investment-advice/investment-strategy-alternative/types-of-investors/18150. [Accessed 19 January 2020].

[4] Ontario Securities Commission, "Self-directed Investing," Ontario Securities Commission, 16 April 2019. [Online]. Available: https://www.getsmarteraboutmoney.ca/invest/investing-basics/getting-started/self-directed-investing/. [Accessed 19 January 2020].

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

[6] Team Linchpin, "A Beginner's Guide To The Agile Method & Scrums," Linchpin SEO, 10 January 2020. [Online]. Available: https://linchpinseo.com/the-agile-method/. [Accessed 21 January 2020].

[7] A. Krush, "Why software developers (quite honestly) hate Agile," ObjectStyle, 26 August 2019. [Online]. Available: https://www.objectstyle.com/agile/why-developers-hate-agile. [Accessed 22 January 2020].

[8] C. Matias, "6 Reasons Why Scrum Fails," Medium.com, 22 October 2018. [Online]. Available: https://medium.com/bring-the-kit/6-reasons-why-scrum-fails-36b89d9e6061. [Accessed 22 January 2020].

[9] P. Kukhnavets, "Strong and Weak points of Agile," Hygger, 21 June 2016. [Online]. Available: https://hygger.io/blog/agile-strong-and-weak-points/. [Accessed 22 January 2020].

[10] Wealthica Financial Technology Inc., "Wealthica: See All your Investments in One Place," Wealthica Financial Technology Inc., 2019. [Online]. Available: https://wealthica.com/. [Accessed 19 January 2020].

[11] PW Portfolio Analytic Inc., "Wealthscope | Powerful Analytics for Holistic Wealth Management," PW Portfolio Analytics Inc., 2019. [Online]. Available: https://www.wealthscope.ca/. [Accessed 19 January 2020].