2014

Project Management Document

FINANCIAL MARKET SIMULATOR

GRAPE

Contents

Software Development Process

Software development processed used

The **Agile** software development process was used. The main reason for using this approach was because the chosen project did not have a strict predefined scope. The Agile method was adopted at a later stage in the development process, because research needed to be conducted to have a broader understanding of the requirements of the project. The project also required a strong background knowledge in stock trading as well as mathematical statistics to reason about the financial markets.

Architecture patterns and strategies

After gathering substantial knowledge of the requirements of the project we had to reason about which architecture patterns and strategies to use. We researched on which are the best type of architectures to use for Financial Market Simulators. We decided to use a Layered Architecture pattern and did not use any reference architectures. The lowest layer of the architecture would be the Matching Engine[1] which uses a Message Queue strategy to receive message from traders (in the context of the project the traders would be algorithms). The higher layer would be where the traders would listen for updates on the Matching Engine using an Event Listening strategy and would then trade based on the updates received.

Technology used

Programming Language

After deciding on the high-level architecture being used by us, we had to decide on the best suited technology to use for the application. The three candidate programming languages being considered was Java, C#, and C++. We decided to use Java because most of the group members have the most experience in it and performance and scalability were important requirements.

Application

We had to decide on what application type to use to deploy the project as. Performance was an important requirement and we decided to deploy the project as Standard Java Application to reduce performance overhead.

[1] A matching engine is an algorithm that operates on an order book and matches and determines prices at which orders are matched.

Team Profile

Madimetja Shika

Personal Information

Gender: Male

Languages: English, Sepedi, Afrikaans

Contact Information

Mobile: 076 312 7907

Email: mmjshika@gmail.com

Education

Field of Study: BSc IT in Applied Mathematics

Modules Completed: Academic information management, Academic writing skills, Communication in organizations, Program design: Introduction, Software modeling, Imperative Programming, Introduction to computer science, Computer architecture, Discrete structures, Calculus, Concurrent systems Mathematical statistics, Numerical Analysis, Linear algebra, Data structures and algorithms, Operating systems, Net centric computing, Informatics, Information science: Social and ethical impact

2014 Modules: Software engineering, Computer security and ethics, Computer networks, Programming languages, Multimedia: Human-computer interaction, Geometry, Numerical analysis

Skills

Programming Languages: Java, C++, C#, Delphi

Web Development: HTML, XML, JavaScript, JQuery, AJAX, JSON, PHP, CSS

Mobile Development: Android

Databases: MySQL, MySQLi, Microsoft SQL Server

Work Experience

Volunteer math and physics tutoring at Doasho High School in Polokwane Limpopo.

Strengths

- Dedicated
- Hard-working
- Honest
- Self-motivated
- Team-player
- Leader.

Interests:

- Mathematics
- Sport
- Nature
- Philanthropy
- Leadership
- Music
- Processes of cognition.

Short Bio

I am an ambitious, motivated and goal driven young man who is passionate about developing robust solutions to presented problems. Although my work experience is limited, I have a willingness to learn and an open mind to new things.

Moeletji Semenya

Personal Information

Gender: Male

Languages: English, Afrikaans, Sepedi

Contact Information

Mobile: 082 314 2001

Email: mdsemenya@gmail.com

Education

Field of Study: BSc IT in Applied Mathematics

Modules Completed: Academic information management, Academic writing skills, Communication in organizations, Program design: Introduction, Software modeling, Imperative Programming, Introduction to computer science, Computer architecture, Discrete structures, Calculus, Concurrent systems Mathematical statistics, Numerical Analysis, Linear algebra, Data structures and algorithms, Operating systems, Net centric computing, Informatics, Information science: Social and ethical impact

2014 Modules: Software engineering, Computer security and ethics, Computer networks, Programming languages, Multimedia: Human-computer interaction, Financial engineering, Numerical analysis, Geometry

Skills

Programming Languages: C++, Java, Delphi

Web Development: HTML, JavaScript, CSS, PHP

Databases: MySQL, Microsoft SQL Server

Work Experience

- Teaching Assistant for COS 121(Software Modeling) July 2013 - November 2013. Marking class tests.
- Teaching Assistant for COS 121(Software Modeling) July 2013 - November 2013. Assisting during practical sessions.
- Volunteer math and physics tutoring at Doasho High School in Polokwane Limpopo.

Strengths

- Problem solving
- Communication skills
- Team player
- Creative

Interests

- Music
- Fashion
- Sports (in particular Soccer and Basketball)
- Investments

Short Bio

I'm a young, energetic individual who enjoys learning and collaborating with other people. I like being challenged as I believe it brings the best out of me and I'm member of the Golden Key International Honor Society.

Daniel Makgonta

Personal Information

Gender: Male

Languages: English, Afrikaans, Tswana, Swati

Contact Information

Mobile: 071 319 0206

Email: danmakk@gmail.com

Education

Field of Study: BSc IT in Software Development

Year: 3rd

Modules Completed: Academic information management, Academic writing skills, Communication in organizations, Program design: Introduction, Software modeling, Imperative Programming, Introduction to computer science, Computer architecture, Mathematical statistics, Introduction to web, Introduction to XML, Data structures and algorithms, Operating systems, Net centric computing, Informatics, Information science: Social and ethical impact Concurrent systems, Calculus, Discrete structures

2014 Modules: Software engineering, Computer security and ethics, Computer networks, Programming languages, Multimedia: Human-computer interaction, Database Systems, Informatics

Skills

Programming Languages: C#, Java, C/C++

Web Development: XML, HTML/XHTML, JavaScript (incl. jQuery and AJAX), CSS, PHP, and ASP.NET

Mobile Development: Windows Mobile, Android

Databases: MySQL, Microsoft SQL Server

Frameworks: C#.NET, Python Django

Work Experience

Volunteer computer skills development project for the Stanza Bopape community.

Strengths

- Good time management
- Good problem analysis and problem solving abilities
- Able to adapt to different environments with reasonable ease

Areas of interests:

Sports (rugby, football, cricket, and athletics) and movies

Short Bio

A current final year student at the University of Pretoria who was born in Mamelodi, Pretoria and raised in a small town in Mpumalanga called Barberton where I attended Hoerskool Barberton for 5 years and did various sports including rugby, cricket, chess, cross country, athletics, and korfball. I enjoy the problem solving abilities that programming gives a person and apply it to other aspects of my life with great success.

Issue Management

The meetings were conducted every Monday. The meetings took place in the form of a semi-structured Scrum meeting. The Sprint period was to do so many deliverables by the next Monday.

Issues within the group and issues regarding the project was dealt with before every Sprint cycle. The scrum meetings held were conducted in a semi-structured fashion and issues or concerns regarding the project or group members were conducted at the beginning of the meeting and dealt with accordingly before the information regarding the next Sprint would commence.

The team leader would be the Scrum Master and the other members would simply take notes or raise concerns. A style adopted for meetings when it came to creative thinking was that one member should always only have positive ideas and one member would think of ways it wouldn't work and this helped us to develop only components that are creative but also more fault tolerant. This style of creative thinking works best when the roles each member plays changes from Scrum Meeting to Scrum Meeting.

All decisions regarding the project were decided on democratically. The only exception to the rule is that when a group member has been assigned to do a certain task then the group member has to follow the *Grape* principle of "do what you believe is right". This methodology was adopted to allow creative thinking and not limit the team member's ability to solve problems. The methodology also helped to improve the team member's confidence when it came to applying knowledge.

Project progress and status

Progress

The project progress was assessed on a weekly basis and a set of goals were put in place from Monday to Monday that needed to be achieved. The project status was not static, it was dynamic and changed quite often as unforeseen problems or circumstances may delay the overall progress of the project and the group had to cater for that. *Grape* believes in starting with something small and letting it grow. The project did not have exact requirements regarding the scope of the project, hence why the agile methodology was adopted for development. All basic components were built as simple units and grew and evolved with time depending on the state of the project in which we wanted to achieve.

Sprint Stories

Sprint	Story	Estimate
1	Jun-16	
	Research on stock trading	3
	Research on financial market simulators	3
	Research on commercial trading software	2
2	Jun-23	
	Research on trading architectures	6
	Research on trading technologies	4
3	Jun-30	
	Create Minimalistic Requirements Specification	8
	Create High Level Architecture	9
	Draw up USE Cases	5
4	Jul-7	
	Research more on technology	8
5	Jul-14	
	Create project	13
6	Jul-21	
	Design Matching Engine	3
	Create 3 Trading Strategies	9
7	Jul-28	
	Implement Matching Engine	4
	Create OrderBook for Matching Engine	4
	Create Stock Manager	3
8	Aug-4	
	Create Market Participant	7
	Create Technical Indicators	13
9	Aug-11	
	Create minimalistic GUI	4
	Create 3 Trading Strategies	4
	Restructure the Matching Engine and StockManager	3
10	Aug-18	
	Research on concurrency	2
	Implement Threading for MarketParticipant	3
	Implement Threading for StockManager	3
11	Aug-25	
	Break	0
12	Sep-1	
	Review documentation	7
	Research on new strategies	5
	Create 4 strategies	9
13	Sep-8	
	Research on concurrency	2
	Implement Threading for MarketParticipant	3
	Implement Threading for StockManager	3

14	**Sep-15**	
	Create Phantom MarketParticipant	1
	Begin with construction of final GUI	12
	Add more technical technical indicators	5
15	**Sep-22**	
	Add graphs	9
	Add persistence in the form of CSV	2
16	**Sep-29**	
	Review documentation	2
	Change entire GUI to cater for Threading	3
17	**Oct-6**	
	Fix bugs with MarketParticipant	2
	Add latest graps	6
	Add more trading strategies	8
18	**Oct-13**	
	Fix bugs with MarketParticipant	9
	Review documentation	9

Release October 18

Total	220
Completed	220
Remaining	0
% Complete	100%
Avg Velocity	12
Estimated # Iterations Remaining	0

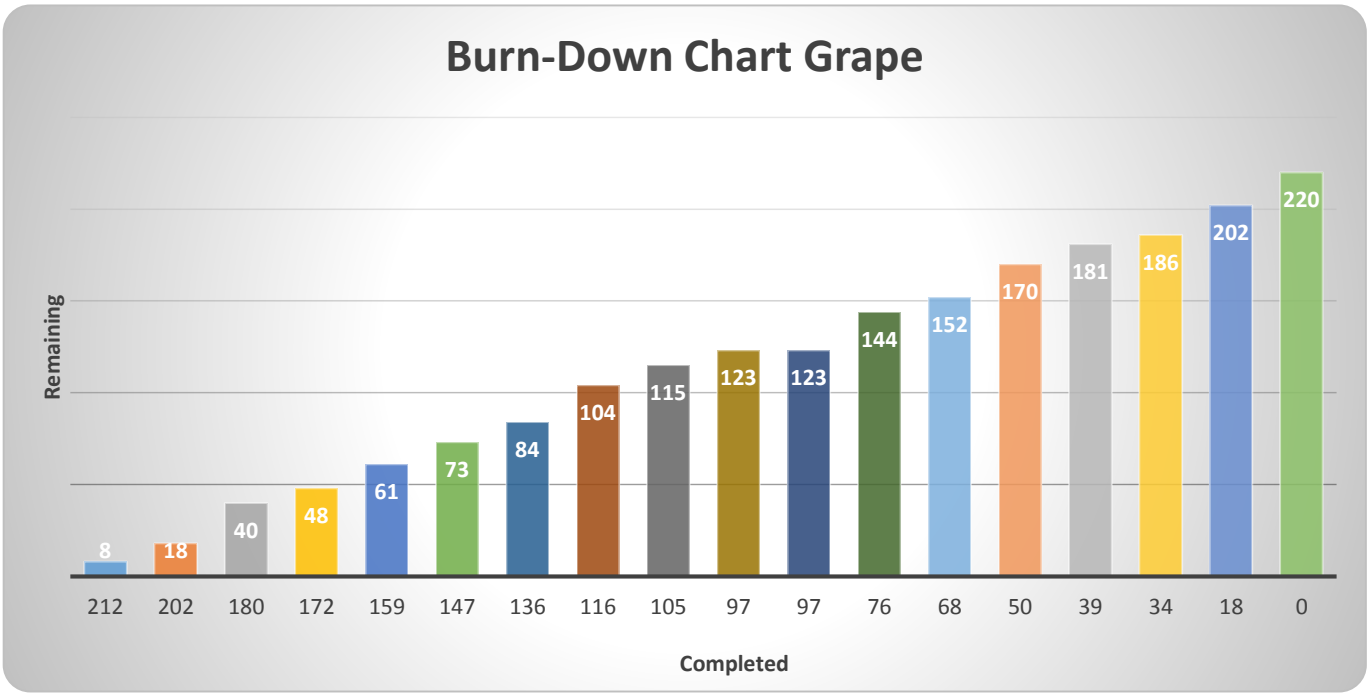
Parking Lot

Did not implement the Chart Patterns and Chart analyis
Did not implment the Market Depth view

Dropped

View Internal State of Matching Engine
Show process of Matching Occurring

Sprint Graph



Functionality not implemented

From our understanding of the project scope all the requirements according to the requirements specification were met. All functionality not implemented were “nice to have features”.

Not implemented:

- Chart patterns
- Chart pattern analysis
- XStream persistence storage of state
- Dynamic trading strategy creation
- How to use the program online guide
- Market Depth analysis

Risks and challenges faced

One of the main challenges was learning about financial markets within the given timeframe. It is a very specialized field of expertise and extensive background knowledge was required in order to convert that knowledge into a working program.

Concurrency was also a major concern. The application relied heavily on Thread Management and ensuring that the application is thread-safe and correct required extensive knowledge on concurrency that took a lot of time to figure out.

Working on the project itself was also challenging in the sense that every group member has a better understanding of their implementation within the project and always trying to have uniformity while also having your own “coding style” was a very daunting task.

Another major problem faced is understanding the scope of the project. The project did not have a final look at the beginning and this made it difficult to define to what extend should the project grow.

Integrating different internal components and making into one project was challenging in the sense that system has to be as open source as possible for other developers to use.