Sameer Patel, Bhargavi Reddy, Angela Gu, Luna Yang

Vision & Scope DOcument  
2 USE CASES

1. Business Requirements
   1. Background

The company uses multiple set of programs across various trading desks to calculate option prices which is confusing and not reliable. The software team is intended to build a single software product that can correctly calculate option prices over various trading desks and let traders calculate and display the results of multiple algorithms.

* 1. Business Opportunity

The company will have the strength to compete with the other companies in the industry with an improved, consistent and integrated Option Pricer system (OPS), which is more efficient and accurate.

* 1. Business Objectives and Success Criteria

A complete solution will provide the traders with an integrated Option Pricer System to help them calculate and compare price options using several algorithms. The new OPS will potentially improve the efficiency of the company’s transactions.

The product should generate results in 0.5 sec with 1% error margin.

* 1. Customer or Market Needs

The new Option Pricer System will satisfy the following requirements:

* The system will correctly calculate option prices over different regions and generate volatility graphs.
* The system will allow our desk quants to select from the possible algorithms to calculate an option price and compare their results.
* The system will let our traders calculate price by adding their own algorithm into the system.

    1.5 Business Risks

The project has to be delivered in 5 weeks which might be a demanding task.

There is a possibility that the trader will not use the OPS effectively and would go back to their previous system which would lead to a significant loss of investment.

1. Vision of the Solution
   1. Vision Statement

For traders who need a consistent, more efficient and more accurate option price system, the OPS will calculate the price using different algorithms, display a consistent result quickly, compare the results with each other and analyze the result more accurately. Unlike our competitors’ system that will produce inconsistent results with different algorithms and let traders take advantage of the inaccuracy, our OPS will be consistent, more accurate and more efficient.

* 1. Major Features

**Functional Requirements:**

The new OPS will be able to calculate price options for different regions, support several different algorithms, add new algorithm to the system and compare results from different algorithms.

**Non-functional Requirements:**

The new OPS will use object-oriented design and will be implemented as an application.

* Adaptability

Must support stock options from several regions like Asia, Europe and USA

Must support algorithms input from users

* Maintainability

Easy to use and maintain

The results must be readable and understandable

* 1. Assumptions and Dependencies

The OPS will replace the current pricing systems.

The OPS will require real time stock price from the stock exchange service.

1. Scope and Limitations
   1. Scope of initial release

In the initial release, the program will be able to

* Correctly calculate option prices over various trading desk (FX, Fixed Income, Derivatives, etc.) and be consistent over them
* Value simple stock options like European, American and Asian
* Exhibit an easily understandable user interface and compare the result from different algorithms with each other
* Display graphs of volatility smiles
* Allow traders to “drop in” their own algorithm into the system
* The program is able to display graphs of volatility smiles.
  1. Scope of subsequent release

In the subsequent release, following features will be added.

* With the help of the latest algorithms, for a given option type, the user is able to select one or more of several algorithms, such as binomial trees, simulation, or numerical integration, to calculate and display the result.
* Be extensible to various types of products, such as futures, swaptions and etcetera.
  1. Limitations and exclusions
* The algorithms other than binomial trees, simulations and numerical integration are not used in the software.
  + These three algorithms are the most widely used, accurate, stable, and efficient algorithms.
  + Traders are allowed to add their own algorithms.
* The program cannot calculate the Underlying Price larger than 1000000.
  + This level of price is rarely reached. This limitation will largely decrease the complexity of the program.
* The program cannot display more than 3 digits of decimals.
  + This level of precision is of little usage. This limitation will largely accelerate the calculation speed.

1. Business context

4.1. Stakeholder profiles

The traders need a more accurate, more efficient, and more flexible option prices calculator as they communicate with the stockholders/investors.

* List of stakeholders: software team, investors, traders
* Major value: improved calculation speed and accuracy, reduced rework, increased customer satisfaction
* The product is very promising and with the experienced programmers, its rate of accomplishing is very high. Thus, shareholders can be very positive about the product.
* Advanced algorithms which improve efficiency and accuracy.

4.2. Stakeholder profiles

In case of emergencies like delayed timelines/development issues, this project will be put as top priority. All departments would be required to shift to the project and contribute to it.

4.3. Operating environment

Windows, iOS

**Use case 1: Calculate Option Prices and display Volatility graph**

**Primary Actor:**

Trader

**Stakeholders and Interests:**

Trader- wants to accurately calculate option prices across multiple stock options

Investor (User/Stock holder): wants to make a choice to call/put depending upon the option prices

Stock Exchange Service: provides stock data to the trader

**Pre-conditions:**

* Trader has access to the Option Pricer system
* Trader has retrieved relevant data from the Stock exchange service

**Post-conditions:**

* System generates accurate and consistent option prices
* System generates volatility graph

**Main Success Scenario:**

1. **Ideal Process Flow**

1.1. Trader selects the region of Stock Option

1.2. System provides list of algorithms to choose from

1.3. Trader selects one of the algorithms from the list

1.4. Trader enters input parameters like Current Stock price, Duration, Expected growth rate, Stock price volatility,

Risk-free interest rate, Strike price, Option to call or put

1.5. System calculates the option price based on the input values

1.6. System gives an option to generate the volatility graph

1.7. System provides an option to run one of the alternative algorithms for the same input values or start a new request (Go to 1.1)

**Alternative Flows:**

2.    **Data Inconsistency in input variables**

2.1. Trader enters input variable in incorrect format

2.2. System highlights the conflicting field and prompts trader to re-enter values

2.3. Trader enters new/correct value

3.    **At any time, aborting the ongoing process of calculating option price:**

3.1. Trader selects cancel button to abort the process of calculation from any stage after 1.1

3.2. System goes back to new request (1.1)

4.    **At any time, System fails:**

To support recovery and correct calculation, ensure the last session is restored from the same step of the scenario.

4.1. System crashes due to some anomaly

4.2. System reboots and reconstructs the last state

**Special Requirements:**

* System should generate accurate results with 1% error margin in less than 0.5 seconds.
* System should be able to convert stock option prices to multiple currencies as per the closing foreign exchange rate on the last business day
* Trader will be able to add a new algorithm into the system

**Technology and Data Variations List:**

* Currencies maybe any of USD, YEN, POUND, EURO etc.
* Trader enters input parameters through keyboard
* Trader can use device used on any of the following operating system – iOS and Windows

**Priority:**

High

**Frequency of occurrence:**

Continuous use during business hour. Trader may use it frequently throughout the day which may peak around 10 – 15 times per hour.

**Open Issues:**

* How will stock price information be retrieved from the stock exchange?
* How to validate the efficiency of the new pricing system?

**Use Case 2: Adding additional algorithm**

**Main Success Scenario:**

Trader selects option to add her/his algorithm into the system. System prompts trader to enter algorithm for one or many regions. Trader enters algorithm and additional variables if any. System compiles the algorithm and checks for syntactical errors. System gives an option to re-enter algorithm or abort or confirm the addition.  Trader confirms addition of new algorithm. System adds algorithm as an option to the list of calculating stock option prices.

**Alternate Flows:**

* If system finds errors in the input, it signals error and prompts trader to correct the errors or abort the process
* If trader aborts the process, system discards algorithm
* If system detects an error in importing the algorithm, it signals importing error to the trader and sends issue to the support team for resolution