

**Czech University of Life Sciences Prague**  
**Faculty of Economics and Management**

Department of Systems Engineering



MASTER OF SCIENCE (M.Sc.) THESIS

**Startup Software Company providing Database Services and Agile  
Methodologies**

Supervised by

Doc. Ing. Vojtěch Merunka, Ph.D.

Submitted by

Simranjit Kaur Virk

© 2022, CULS

## DIPLOMA THESIS ASSIGNMENT

B.Sc. Simran Virk, BSc

Systems Engineering and Informatics  
Informatics

Thesis title

**Start-up software company providing database services and agile methodologies.**

---

### Objectives of thesis

The aim is to design main organisation structure and agendas of a small start-up software company providing database services and agile database techniques and methodologies. Expected result will be a model of to-be processes, patterns and identification of process participant (e.g. employee roles, organisation units...) from IT and SW engineering aspect.

### Methodology

The work consists of two main parts: Theoretical and practical. Theoretical part specifies various aspects of the business and an overview of used tools, techniques and knowledge. The practical part is based on a case-study of management consulting services within a database development company. The methodologies that are devised for coping with the constant change during the system development are agile methodologies, where practitioners and researchers are showing keen interest to use agile strategies for databases projects development. The case study will aim to find out how well the agile strategies and database systems suitable for the development of projects. The thesis considers a case study of a medium scale database-development centric organization.

## The proposed extent of the thesis

Approx 60-80 pages

## Keywords

database application development, business process modelling, agile technology

---

## Recommended information sources

Scott W.Ambler, "Agile Database techniques", Wiley, 2003

Scott W.Ambler, Process patterns: building large scale systems using object technology, SIGS books/  
cambridge university press, July 1998

Scott W.Ambler, Process patterns: Delivering large scale systems using object technology, SIGS  
Books/Cambridge university press, September 1998



---

## Expected date of thesis defence

2019/20 SS – FEM

## The Diploma Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

## Supervising department

Department of Information Engineering

Electronic approval: 19. 2. 2020

**Ing. Martin Pelikán, Ph.D.**

Head of department

Electronic approval: 19. 2. 2020

**Ing. Martin Pelikán, Ph.D.**

Dean

Prague on 18. 03. 2021

# Declaration

I certify that the paper entitled “Startup software Company providing Database Services and Agile methodologies” submitted as a partial requirement for the degree of Master of Science (M.Sc.) in Informatics is the result of my own research, except where otherwise acknowledged, and this project report in whole or in part has not been submitted for an award, including a higher degree to any other University or Institution.

Name : Simranjit Kaur Virk

Signed:

Date:



## **Acknowledgements**

At first I would like to express gratitude to the almighty for the blessing guidance and immeasurable grace for successful completion of the thesis and submission. It is a great pleasure in expressing enormous gratefulness and sincere gratitude to my respected supervisor, Doc. Ing. Vojtěch Merunka, Ph.D. Department of Systems Engineering (FEM), Czech University of Life Sciences Prague (CULS), for his valuable advice and continuous help throughout the period of my research work.

I am extremely grateful to my erudite professor and course coordinator Ing. Marek Píčka, Ph.D. for study guidance, preparation of supporting documents on demand for training and scientific workshop.

I would like to express my heartfelt thanks to my father, mother and friends of CULS and specially my beloved friend Alam, my beloved friend Josie, my Team lead Miroslav and other colleague of the company I am working, for their enthusiastic support.

## Abstract

The main objective of this thesis is modelling and analyzing business processes of my start up company providing Database services by using Agile methodology. The company has 5 teams which are doing their own roles and responsibilities independently. There are 100 employees working in the company. The working hours in the company are not flexible as its a start up. The main aim of the company is to provide agility in there processes. As Today's modern Enterprises deals with a highly dynamic environment which requires management to integrate available knowledge gained from the data and information. This mainly focused on the comparison between relational and non relational database and why non- relational database is effective and more used. Non- relational management system enhances the limitations which relational database management system face while storing large volumes of unstructured, user generated text based data in distributed environment, their capacity to store large volume of data over multiple servers and programming interface they provide. Also it focused on the highlights of business processes of the organisational structure of various departments of company and how they work and coordinate with each other by Agile methodologies using CRAFTCASE tool. Business process is collection of activities which produces output with one or more inputs along customer satisfaction. Business process is mainly customer oriented. It is a set of clearly defined and manageable task or activities in an organisation. It includes monitoring, auditing, performance analysis and improvement in processes. Agile and scrum have become more and more popular. Most of the higher management team in an organisation are preferring agile way of working as a key to success. Agile and scrum are the mostly used methodologies in my thesis where scrum is an agile framework for completing complex projects.

This paper describes the business processes modelling structure of a start up company which is providing databases focusing on the comparison between relational and non relational database using agile way of working and organising team for efficiency and productive output.

**Keywords:** Database application development, Business process modelling, Agile technology.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Statement . . . . .	2
<b>2</b>	<b>Objective and Methodology</b>	<b>5</b>
2.1	Objective of the Study . . . . .	5
2.2	Methodology of the Study . . . . .	5
<b>3</b>	<b>Literature Review of Company FxNet</b>	<b>7</b>
3.1	Organisational structure of a start-up company . . . . .	7
3.2	Agile Methodology . . . . .	8
3.2.1	What is Agile? . . . . .	8
3.2.2	Comparison of agile and waterfall . . . . .	9
3.2.3	Agile Methodologies and Comparison Between Extreme Program- ming and SCRUM . . . . .	11
3.2.4	Why it's better? . . . . .	13
3.2.5	SCRUM Planning And Collective Commitment . . . . .	13
3.2.6	XP And Embracing Change . . . . .	14
3.2.7	How to implement it? . . . . .	16
3.3	Relational Database Management Systems (RDBMS) . . . . .	20
3.3.1	Introduction . . . . .	20
3.3.2	What is Relational Database Management System . . . . .	21
3.3.3	Relational Data Modelling . . . . .	24
3.3.4	Limitations of Relational Model . . . . .	24
3.3.5	Implications of Constraints of Relational Database . . . . .	26
3.4	Non- Relational Databases . . . . .	28

3.4.1	Column- oriented database . . . . .	30
3.4.2	Graph Database . . . . .	31
3.4.3	MongoDB . . . . .	33
3.4.4	Other Application Areas . . . . .	35
3.5	Business Process Modelling . . . . .	35
3.5.1	What is business process modelling? . . . . .	35
3.5.2	Where it is used? . . . . .	36
3.5.3	Implementation of BPM . . . . .	37
<b>4</b>	<b>Application of the proposed Agile Methodology in Tech support and Development Team</b>	<b>39</b>
4.1	Process flow of Projects . . . . .	39
4.1.1	State Diagram Process workflow . . . . .	40
4.1.2	Activity Diagram for Help desk Management system . . . . .	41
4.1.3	Login Activity Diagram of Help desk . . . . .	42
4.1.4	Merging Activity diagram with release goals . . . . .	43
4.2	Implementing Agile in Development Teams . . . . .	44
4.2.1	Product creation Framework: MILESTONES . . . . .	45
4.2.2	The structural hierarchy in scaling agile methods . . . . .	46
4.2.3	BPR and BPM . . . . .	48
4.3	The Project and cross-Project Tasks of OOSP . . . . .	48
4.3.1	People Management . . . . .	49
4.3.2	Training and Education . . . . .	49
4.3.3	Quality Assurance . . . . .	49
4.3.4	Metrics . . . . .	50
4.3.5	Success Factors . . . . .	50
<b>5</b>	<b>Case Study</b>	<b>51</b>
<b>6</b>	<b>Practical Application</b>	<b>54</b>
6.1	Scenarios . . . . .	54
6.1.1	Development . . . . .	54
6.1.2	Operation . . . . .	54
6.1.3	Testing . . . . .	55

6.1.4	Transport Support Team . . . . .	55
6.2	Simulation Report . . . . .	55
6.3	Business Architectures . . . . .	59
6.3.1	Transport Support Team . . . . .	59
6.3.2	Participants . . . . .	60
6.3.3	Basic modelling cards . . . . .	61
6.3.4	Detail modelling cards . . . . .	62
6.3.5	Data Flow . . . . .	62
6.3.6	Development Team . . . . .	63
6.3.7	Development Simulation steps . . . . .	64
<b>7</b>	<b>Conclusion</b>	<b>68</b>
	<b>Bibliography</b>	<b>69</b>

# List of Figures

3.1	Organisational structure of start up IT company . . . . .	7
3.2	Stages of Waterfall software development life cycle model . . . . .	10
3.3	Overview of Extreme Programming . . . . .	19
3.4	The Attributes and Tuples Relationship . . . . .	25
3.5	Shared Client connection . . . . .	34
3.6	Framework of Business process Model . . . . .	37
4.1	Support system . . . . .	40
4.2	Process Workflow state diagram . . . . .	41
4.3	Help Desk Management system . . . . .	42
4.4	Login Activity Diagram . . . . .	43
4.5	Activity diagram with release goals . . . . .	44
4.6	PCF ( Milestones) . . . . .	46
4.7	Hierarchy in agile methods . . . . .	47
4.8	BPR and BPM . . . . .	48
4.9	Support Pattern . . . . .	49
5.1	Decision Process Pattern . . . . .	52
5.2	Decision Process Pattern . . . . .	53
6.1	Development Team . . . . .	54
6.2	Operation Team . . . . .	54
6.3	Operation Team . . . . .	54
6.4	Testing Team . . . . .	55
6.5	Transport Support Team . . . . .	55
6.6	Transport Support Team . . . . .	55

6.7	This is is a simulation part1 . . . . .	56
6.8	This is is a simulation part2 . . . . .	56
6.9	This is a simulation part3 . . . . .	57
6.10	This is a simulation part4 . . . . .	57
6.11	This is a simulation part5 . . . . .	58
6.12	This is a simulation part6 . . . . .	58
6.13	This is a simulation part7 . . . . .	59
6.14	Help Desk Management System . . . . .	59
6.15	Transport Support Team . . . . .	60
6.16	Transport Support Team . . . . .	60
6.17	This is a report image part1 . . . . .	61
6.18	This is a report image part2 . . . . .	61
6.19	This is a report image part3 . . . . .	61
6.20	This is a report image part4 . . . . .	61
6.21	This is a report image part5 . . . . .	62
6.22	This is a report image part1 . . . . .	62
6.23	This is a report image part2 . . . . .	62
6.24	This is a report image part3 . . . . .	62
6.25	This is a report image part1 . . . . .	63
6.26	This is a report image part2 . . . . .	63
6.27	Development Team . . . . .	64
6.28	Development Simulation 1 . . . . .	64
6.29	Development Simulation 2 . . . . .	64
6.30	Development simulation 3 . . . . .	65
6.31	Development simulation 4 . . . . .	65
6.32	Development Simulation 5 . . . . .	65
6.33	Development simulation 6 . . . . .	66
6.34	Development simulation 7 . . . . .	66
6.35	Development simulation 8 . . . . .	66
6.36	Development simulation 9 . . . . .	67
6.37	Development simulation 10 . . . . .	67

# 1 Introduction

The thesis mainly focuses on the start up software company providing database services using agile methodologies and business process modelling by craft-case tool. By assessing the results collected from the online data the case study generalises the results. The output and results for the research will give an insight of the best practices that is considered while in agile strategies and also the practical problems that we may encounter on the journey. The findings have strong impact for both business and technical managers who wants to consider agile strategies for the development of database management projects.

There are 100 employees working in this start up company. Modern environment in an enterprises, business, IT and financial organizations, banking systems and almost all the private and public services are dynamic which is becoming more complex day by day. Organisations are preferring object technologies for building large scale systems and business applications. Today organisations need Tested techniques for managing complex large scale, object oriented software development projects called Process patterns. Process patterns describes one or more inputs used to describe one or more outputs whereas patterns defines the repetition of same task over and over again. Business Process modelling represent processes of an Enterprise, for the improvement and to analyze the current processes. The object oriented software process depicts process patterns dealing with medium to large size organisations developing software hence supporting main ares of business. The main focus is on Object oriented software process (OOSP) that provides a framework of successfully delivering large applications, developing those applications which are easy to maintain and Handel ensuring high quality development efforts. OOSP consists of four main phases: Initiate, Construct, Deliver, Maintain and Support which are further divided into iterative stages.

According to 'Christopher Alexander'(Martin, 2002) states patterns exists at all scales.



As repetition of pattern is different than a part as different parts will be unique but patterns will be the same.

As the organisation's main focus is providing Database services and training on Agile methodologies. The main Agile approach to software development is extreme programming and SCRUM. Agile development promotes adaptive planning, development and delivery which encourages rapid and flexible response to change. In Agile development individuals can interact over processes and tools, customers can collaborate over contract negotiation, responding to change over following a plan. And the Database will be used in the relational database management system (RDBMS).

The Relational database model was developed in 1970's. A relational database is relational model of data and is a digital database management system which uses SQL (structured query language) for querying and maintaining the database. In Relational database the data is displayed to the user as relations as collection of tables where each table consists of rows and columns. The company needs to save the information of its employees, departments and salaries which is called Data management (Ambler, 1998).

According to S.Sumathi, Database is a well organised collection of data that can be accessed in different logical orders. Interpretation and storage of information is primary goal of Database Management System (Sumathi and Esakkirajan, 2007).

In DBMS user can input, share, edit, display and manipulate data. Also it enhances more than one user to share the data and the complexity extends to its design and implementation. According to 'Stonebraker'(2008) Relational data model is best suitable for transactions based business applications which is not ideal for all data storage processes. Data in the relational Database Management system is accessed by query language called SQL.

## 1.1 Problem Statement

The main problem focused on is why it is important to consider post relational database or non relational database management system in present scenario? These databases were developed after the relational databases have come into practise. Post or non relational databases includes multi-dimensional databases, object relational databases, column databases , No SQL etc. It is used for storing large amount of unstructured data

which the relational database cannot. Relational database is used for traditional based business applications, which is not ideal for all data storage cases (Stonebraker 2008). In the modern era , mobiles and internet has made it easier to store large volume of information which is not best suited for traditional models.

Another problem in database management system is management of large amount of data as to extract information from large set of unstructured data is very difficult process. Non - relational database management system provides scalability over multiple servers of database on the other hand relational database faces troubles in dealing with large traffic with their high end clustering solutions. Document oriented databases like NoSQL, MongoDB successfully stores large amount of unstructured data in a distributed environment.

Usually the data generated by the users in non- relational database which is not considered by relational database management system. Alternative data models are needed because of high volume of data being displayed (Hewit 2011:5).

One more problem is the distribution of data over multiple servers. The heavy amount of data cannot be stored on a single server which has to be addressed in some way. Hence, the internet, VPN (virtual private network), LAN (local area network) and WAN (wide area network) have made it easier and feasible to store data in different geographic locations by having users at remote site which requires data access. In order to eliminate the communication overhead of the object driven , a copy of data is stored and synchronisation of data is achieved through replication (Wiesmann, Pedone, Schiper and Alonso 2000).

A further area problem is the discrepancy in presentation of data and programming language abstraction in the database. Advanced programming language towards object oriented language mean data manipulation and data representation needs abstraction. Data is accessed through SQL (structured query language) in RDBMS which is subset of relational algebra (Codd 1971). Abstraction of data model is hence being done by the programming model.

A mechanism required to test the veracity of promises that non- relational database system have made to overcome the shortcomings of relational database management system. To check if there is any improvement in non-relational database management system we need to understand how each system works and which data model is used for:

1. The level of programming abstraction provided.

2. Storing large volume of unstructured data.
3. Scaling data over multiple servers with high requirement in distributed environment.
4. Ensures data warehousing like analysis, searching, querying and filtering of data.

Now a days, with the advancement in CPU, network speed, memory and secondary storage devices the cost of hardware has become very cheap (Stonebraker 2008). Apparently, by adding more machines by data over multiple database servers has made the commodity hardware convenient and easier to use. In the non- relational database programmers can directly access and manipulate data without adding abstraction layers also by reducing programming cost at a very low price. Unstructured user daily generates text data on internet web pages which consists of articles, news to blog entries which is view-able to users in web browsers. By storing this data provides accurate data for data mining applications which is beneficial for the Enterprise to find out the latest trends. Due to large volume of data its impossible to store information on a single server. Hence, the reason for this research is to figure out how we can store data in RDBMS.

## 2 Objective and Methodology

### 2.1 Objective of the Study

Independently, Database application, business process modelling and agile methodologies are well-developed research areas. The main objective of this study is to integrate all this three processes and develop a well organised start up software company providing database services. The main objective can be split into following parts:

- Introduce current technology and tool like craft case to create such business process modelling structure.
- Identify the processes of integrating various departments in a company, their roles, activities and their authorised responsibilities in an organisation using agile methodologies and how all the departments co-ordinate with each other.
- Finally Create a Database application which includes the comparison of relational and non relational database in an Enterprise.

### 2.2 Methodology of the Study

According to (Χρυσομάλλη, 2018) As start up valuation is more art than science. It developed a well organised start up software company providing database services based on RDBMS and business processes modelling. By creating the whole application with respect to enterprise environment and equipped with the latest technology and services using agile and scrum. The Company's main focus is to build a database application which is providing database services to other companies. For business process modelling craft case tool describes the entire workflow process in an organisational structure of the company.

- Software development life cycle processes including Planning, Analysis, design, Implementation and Maintenance.
- Preparing a database application based on relational and non relational databases using SQL.
- Use advertising tools to promote by monitoring the web for interesting new content.
- Business process modelling tools using craft case describing the workflow of various departments their employees and roles.
- Combine all result and create the well structured start up company providing database services using agile methodologies.

# 3 Literature Review of Company FxNet

The company has 5 teams BA Team, HR Team, Development Team, PM Team, Tech Support Team each has different responsibilities and roles with 100 employees working in it. There are no flexible hours in my company as its a start up. The employees working in my company are very dedicated and focuses on improvement. Also we satisfy our employees with fair wages so that they get motivated to work and give good results. The employees are very fair and honest with the work related tasks. Hence the main focus in on the agility of the work flow process. Both the theoretical and practical aspects has been discussed briefly.

## 3.1 Organisational structure of a start-up company

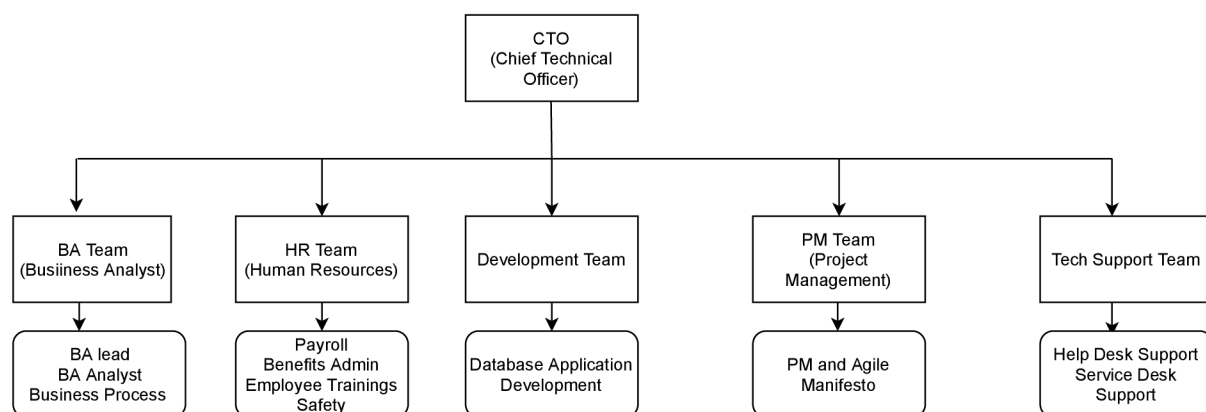


Figure 3.1: Organisational structure of start up IT company

Above is the organisational structure of a startup IT company which has 100 employees working in different teams where each team consist of 20 employees providing database services using agile methodologies and techniques. There are basically 5 teams working in my company with 20 employees in each team as a start up Business Analyst

team, Human Resource team, Development Team, Project Management Team, Technical Support Team. Where BA team focuses on the business process modelling by describing the process patterns and workflow of my company. HR team is responsible for the financing, Payroll, benefits and safety etc. Development team is responsible for my database application development where the comparison is done the difference between relational database (SQL, Oracle) and non- relational database (NoSQL, MongoDB). Also it has mentioned that why we prefer using non- relational database rather that relational database. PM team mainly focused on Agile manifesto which includes agile methodology like eXtreme programming and SCRUM. Lastly the technical support team is working as Help-desk/service desk analyst which is to provide customer satisfaction to the users by resolving technical issues that our users are facing in day to day life. Therefore, all the teams co-ordinate together to achieve a common goal which is a key to success.

## **3.2 Agile Methodology**

### **3.2.1 What is Agile?**

Agile as a concept is introduced in late 1990's through an effort to overcome the difficulties with an existing solution of development process. On the other hand, the discovery of agile had already been started in 1980's with the rise in new technologies. Agile is a time bound, iterative approach for delivering the software from the start of the project till the end. According to (Stellman and Greene, 2014) Instead of delivering it all at once to the end, it divides the project into different user functionalities called user stories. Then, prepare a proper plan to deliver the tasks in short two week cycle which is called iterations. Supporters of agile accepted the fact that project completion uncertainties should be accepted and there has to be proper solution to avoid it by planning and control with execution and feedback. Due to which, agile project manifested open communication among self- organising teams with proper training's and learning's which resulted in iterative development and incremental delivery of software's. Also it emphasise in customer collaboration, working solutions over comprehensive documentation, individual interactions which results in change. With this they discovered that with proper interaction among project team members and customer will result in more working solutions in a flexible manner. Therefore agile is Adaptive, Value driven, Collaborative and empowered.

With this there resulted in proper output of projects with adequate time management and proper utilisation of resources.

### **Principle of Agile:**

1. Foremost principle is customer satisfaction through early and continuous delivery of software's.
2. Accept changing requirements in development. Agile process uses effectively the changes for customer's competitive advantage by hiring telepathic and clairvoyant team members.
3. Delivering working software frequently from weeks to months within short time period.
4. Excellent Face-to-face conversation for passing information within the development team.
5. Developers and business people should work together daily throughout the project.
6. Build project with motivated individuals and trust them to get the job done.
7. Working software is primary goal of progress.
8. Agile processes enhances sustainable development.
9. Continuous focus to technical excellence and perfect design promotes agility.
10. Simplicity
11. Best design and requirements emerge from self- organising teams.
12. Teams regularly focus on becoming more effective and the adjusts itself accordingly.

### **3.2.2 Comparison of agile and waterfall**

The first and foremost approach introduced in SDLC (Software development life cycle) is the Waterfall which is widely used in software engineering to ensure successful delivery of projects. It is also called linear- sequential life cycle model which is very easy and simple to understand. Basically, in waterfall outcome of one phase results in input of next phase sequentially.

- Gathering requirement and analysis: All the information and requirement of a system to be developed are documented in this phase.



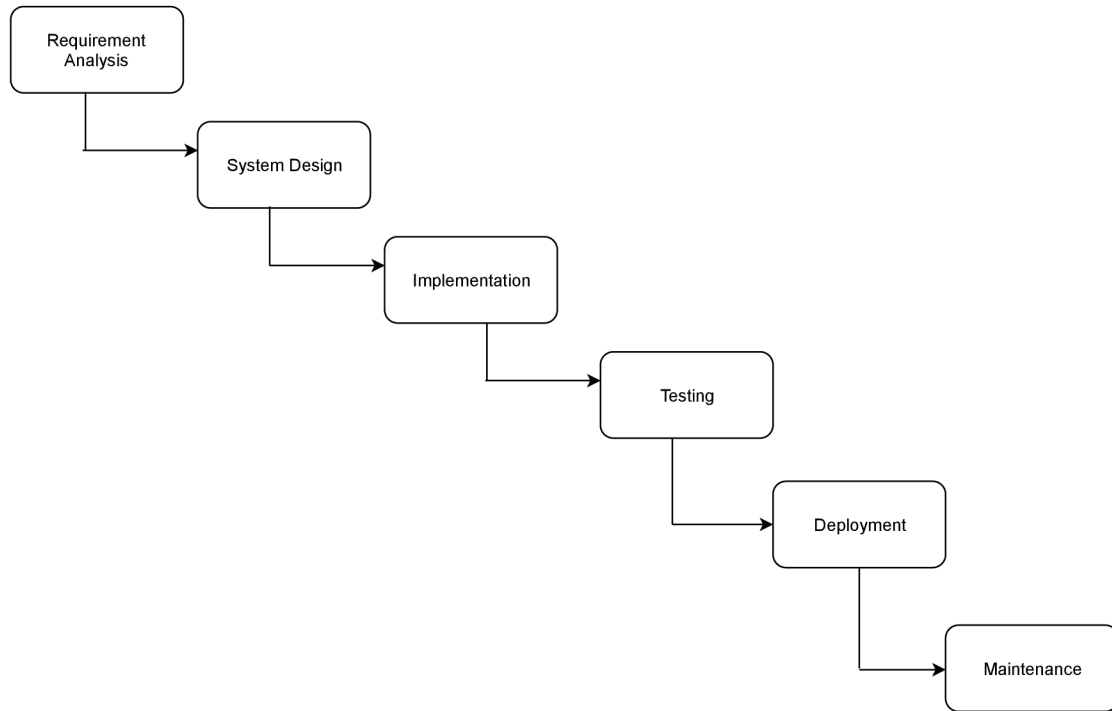


Figure 3.2: Stages of Waterfall software development life cycle model

- **System Design:** All the hardware and system requirements which helps in defining overall system architecture is covered in this phase.
- **Implementation:** With inputs from system design, system is developed in small programs called units which is combined in next phase. Each unit is developed and tested which is referred to as unit testing.
- **Integration and testing:** All the units developed in implementation phase are integrated and tested. Hence, the entire system is tested for any faults and failures.
- **Deployment of the system:** After the functional and non functional testing the product is deployed in the customer sphere or released into the market.
- **Maintenance:** It is done to make changes in the customer environment. As many issues comes up in the customer environment, to fix those issues patches are released. Hence, through maintenance these changes are delivered very easily.

All these phases are torrent to each other that is flowing downward. The next phase starts only after the defined set of goals of first phase are achieved hence called waterfall. Ample resources with required expertise are needed to support the product.

Drawbacks:

- At testing stage it is very difficult to go back and make changes in an application.
- Not suitable for long and ongoing projects.
- High risk and uncertainty.
- Difficult to make progress within the changes.

### **3.2.3 Agile Methodologies and Comparison Between Extreme Programming and SCRUM**

#### **Extreme Programming (XP)**

The software development methodology for improving the software quality of the products it focused more on the engineering aspects of the development and to meet the needs of changing customer requirements. According to (Moran, 2016) Being agile software development it enhances frequent " releases" which improves productivity and introduces upcoming customer requirements. There is frequent communication with customers and programmers which creates better understanding the needs and problem of clients. In XP developers pays more attention to changing customer requirements even late in the life cycle. It mainly focus on team work as managers, customers and developers co-ordinate with each other. The team collaborate equally to solve the problem together. XP team improves their software project in 5 ways: communication, simplicity, feedback, respect and courage. As said by (Moran, 2016) XP first collects user stories from customers which system needs to do for them. Then they follow acceptance test to make sure that the story has been implemented correctly. After the story has been made we used the release plan meeting to create a release plan where customer chooses iteration planning meeting, these stories are then implemented into different programming tasks for a specific iteration to be completed. The acceptance test is created from the user stories. This tests makes sure that the customer requirements has been met and the system is acceptable. QA is also a very important part of extreme programming. Hence, in extreme programming there is daily communication within the team for better output and delivery of products as per customer satisfaction which is far better than waterfall model of agile development.

## SCRUM

SCRUM methodology is the combination of two concepts ” iterative and incremental development ”. According to (Satpathy, 2013) Its main focus point is towards the management side of the development. Its a product development strategy where development team works together to reach a common goal. In the product development the team works back and forth. SCRUM teams usually have 6-10 members. If it exceeds 10 people multiple SCRUM meetings is formed to work on a project. The convene Scrum of scrums, process facilitates coordination among scrum teams which enables effective implementation of large projects. Large teams may have multiple scrum teams working together and co-coordinating which facilitates an easy flow of information and enhances flexible communication.

There are 6 scrum principles of SCRUM: Empirical process control, Self- organisation, collaboration, value- based prioritization, iterative development, time - boxing.

1. Empirical process control: It focuses on clarity, adaptation and security check.
2. Self- organisation: It focuses on self- organised workers delivering greater value which resulted in innovative ideas and creative environment which is more conducive for growth.
3. Collaboration: It focuses on collaborative work.
4. Value- based prioritization: delivering maximum business values.
5. Time - boxing: Time- boxed elements in the scrum includes sprint, Daily stand up meetings, sprint planning meetings and sprint review meetings.
6. Iterative development: It mainly focus on managing changes and build software satisfying customer’s needs.

### **SCRUM Process:**

1. Initiate
2. Plan and estimate
3. Implement
4. Review and retrospect
5. Release

Hence, with the introduction of SCRUM

### 3.2.4 Why it's better?

Both SCRUM and eXtreme programming (XP) are very organised methodologies that support each other. It's very difficult to describe whether you have worked in a scrum team or an XP team.

The main differences between both are:

1. Scrum team typically works in iterations (called Sprints) which is from two weeks to one month longer on the other hand XP team also works in iterations which is one to two weeks longer.
2. Scrum teams is unchangeable into their sprints. Once a sprint planning meeting is done and a commitment is made to deliver a set of uncompleted product items, that items remains unchanged till the end of the sprint. On the other hand XP team is much more open and responsive to suggestions and is adaptable to change within their iterations.
3. eXtreme programming works in a strict priority order. Features are prioritized by the customer (Scrum's product owner) to be developed and the entire team works on it whereas the scrum product owner focuses on the incomplete product where the team finds out the different methods to develop backlog items.
4. Scrum is mostly silent on engineering practices whereas xp is most likely silent on management practices. That is why SCRUM and XP fits so well together and works in co-ordination to each other.

### 3.2.5 SCRUM Planning And Collective Commitment

According to (Sumathi and Esakkirajan, 2007) the main focus is on the practices that scrum team uses for planning their sprints and the work done in real life. As there is a huge difference between scrum theory and the teams building software on a live project and being successful as it is all about self- organisation and collective commitment. This is all about how user stories can make you understand what exactly is the requirement of the users from the software. There are two visualisation tools burn-down charts and task boards to keep everyone on the same page.

As users and stakeholders hate unpredictability even if you build the most valuable software if its not upto users expectations they will be disappointed. There are two ways were we can avoid damages. Firstly, the team needs to do good job upto the expectations

at the beginning of the sprint. Secondly, On the basis of daily scrum meetings the team has to be updated as the sprint progresses. Which is why its very important to have users and stakeholders present in the daily scrum meeting to observe and not interfere. As said by 'Ken Schwaber' If you don't get collective commitment you don't get scrum.

**Collective Commitment** means to make your software useful by understanding the requirement of your users. Building the working software that really helps the users get their work done by customer collaboration. Hence, what are agile teams doing that allows developers to build useful software? They are actually trying to understand user's need and requirements. By writing a user story in a simple description of specific way of how the user will use the software which is 3\*5 index card. Hence by writing the story which is an effective tool for removing gold plating (unnecessary use of extra features.) By this they can avoid spending time on developing software with features that the users don't need. By writing the story its easier for product owner to review the story and figure out which one is the most valuable. One technique that scrum teams can use is **Story Points** which will help to understand how much efforts you need to build a specific story by comparing current user stories to other stories which are being written in the past.

**Burndown Chart** actually helps to find how the sprint is progressing when compared with the team's past velocity.

### 3.2.6 XP And Embracing Change

As we all know that people hate change. Developers always complain about the users that they don't know what they want and change their minds frequently which make the life of developers miserable as they don't know what to build. Scrum has a solution as it says work with the user to know their needs to understand the change over time. This makes the business owners and project manager the ability to know the goal of the project. After this team has to make frequent changes to code which the developers knows very well when their is need to change the code, it causes bugs. More changes you make makes the codebase complicated.

This is one of the problem that XP (Extreme-programming) solve being an agile methodology. Its same like scrum full of values, practices and principles. This thesis describes about the primary practices of XP and their how they are applied by the software team. Also it focuses on how they help every team member to have the right

mindset to build better code and instead to hating change all the members embrace change and accepts it happily.

There are primarily 10 practices of XP programming which are further divided into 4 categories - programming, integration, planning and team just to make it easy to understand.

Programming is further divided into two sub categories Test First Programming and Pair Programming. In the test first programming also called test driven development the developer builds an automate test before he writes code. That helps him to know that if code works or not as when the test fails means that the product code is not written on the overhand if test passes shows that the code is working. It helps to prevent defects by writing failing tests, build code to make them pass and finding problems and fixing them by discovering different ways to solve them. Again writing more failing tests which are called Unit Test. As in every programming language code is broken down into units (classes, methods, functions, sub-routines and modules) and there is always one way of building and running test. It makes sure each individual unit of code works. When a programmer writes unit test before the code builds those codes helps in immediate failing , and test helps the programmers to find the problem before its too difficult to remove.

On the other hand, in **Pair Programming** two developers sits together on a single workstation when writing code. In pair programming two developers sits together one writes the code while the other watches which make easier for them to work as if one gets tired other takes handover which reduces fatigue. As discovery says that pairs build more better code than working separately.

In the **Integration** category we have two categories. First one is 10- MINUTE BUILD which means automatic running of entire code base in 10 minutes. Build includes automatically running of all unit tests and making reports of passed and failed tests. If it takes more than 10 minutes the team members don't run it. Secondly, its **Continuous Integration** where a lot of people are working on a single source code file simultaneously. As if team members works on a single files will create mess as they will constantly overwrite one another changes. Continuous integration means each team member keeps his or her own copy of codebase.

PLANNING practices is based on long term planning of Extreme programming which is broken into iterations. In a **Weekly Cycle**, XP team uses one week iterations where

they focus on user's stories, work with customers to divide the story for iterations and then breaks in into tasks that are assigned to developers. Once planning is done in the first part of iteration the team writes automated tests for stories and second part of iteration teams spend on writing code to make the test pass.

Moreover the teams uses **Quarterly Cycle** practise to do long term planning. Here XP teams will focus on the themes or larger ideas which they can apply in the real world scenario which they can use to tie their project stories together. Hence , by discussing themes helps the team to find out the stories that needs to be added into the project and keeps them connected in real world business problems which they are solving with the software (Martin, 2002).

TEAM practices explains the fact that teams works well when they sit together as they can interact with each other about their problems and ideas by helping each other. But the programmers need a private work-space which is without distractions so that they can work efficiently and give productive output. Then second team practise is Informative Work-Space by using large task-board and burn down chart.

Therefore, effective XP teams have good balance among themselves to keep everyone up-to-date.

### 3.2.7 How to implement it?

Now, the focus is on the real live implementation of these methodologies in the start up company where in the **Project Management Team** there are 20 employees working by using effective agile methodologies where we have Project manager, Scrum master, Team lead and Stakeholders. On the other hand, In the **Development Team** there are developers working together in pairs using pair programming, back-end developer works hand in hand with front- end developers and creates the logic so that the web app should function properly by using server side scripting language like SQL, PHP, RUBY, PYTHON etc. Apart from this back end developer creates data storage solution using a database like MySQL, MongoDB, Oracle and PostgreSQL, Database administrator uses special software for storing and organizing data, designing database, Project manager, Testers and many more. Business people and developers must work frequently together and should have flexible communication for sustainable development and promotes growth.

## **IMPLEMENTATION IN PROJECT MANAGEMENT WITH SCRUM**

As per (Stellman and Greene, 2014) Here in project management team has mainly focused on the DSDM (Dynamic System Development Method) framework has been best understood. The DSDM philosophy says that business value appears through communication and collaboration within an incremental and iterative approach that delivers well defined quality of solutions in specific time-frame. According to DSDM, the stakeholders need to focus on the business needs, goals and priorities. Also it provides the framework for identifying responsibilities of the team. The role of team lead is not permanent or hierarchical in nature. Restructuring new responsibilities and the familiar change in the mindset is one of the biggest challenge. Hence, the dynamic responsibilities within the team which are new to agile should be taken into consideration by DSDM coach by that time the team gets mentally prepared to take such responsibilities. In the DSDM the project plays the role of risk manager who ensures that risk is collectively discussed and addressed. It also includes business visionary who must keep check on the business risks and the technical coordinator who keeps check on the technical risks in the technical architecture. Finally the business analyst assist in the identification of risk and their respective domains.

Now all the employees are trying to do a good job and the supervisors are watching that everyone is good in performing the tasks assigned to them and working very enthusiastically. And now they are expecting someone to come and help them to adapt to entirely new ways of doing things. And the agile coaches are spending most of their time helping people on the teams change the way that they work which is challenging for both coach and team as only the coach can see the big picture. Teams are trying to adopt many practices like they often turn daily scrum into daily status meeting. The most important task of daily scrum is to replace command-and-control project management with self organisation. The scrum master updates the team about the daily changes in the project and assigns work to them.

## **IMPLEMENTATION BY USING EXTREME PROGRAMMING**

In this start up company the first step in Agile using extreme programming is to collect all the user stories then organising spike solutions. The main aim of spike solutions is to find out the solutions for technical problems. According to (Ambler, 1999) Also it reduces



the risk of technical problems and also to increase the reliability of user stories. When the technical problem holds up the system's development pushes the pair of developers to work together for a week or two to reduce potential risk by working on a technical problem. Keep the system accessible with extra stuff as we are not going to use it in future as per experience, it will only slow down the process and vanishes the resources.

Keeping the code ready to avoid unexpected changes is about simple design. Adding extra flexibility makes the design more complex and difficult to understand. Hence, following the five rules of code :

1. **Planning** where the user stories are written , making frequent small releases, dividing the project into iterations and iteration planning starts each iteration.
2. **MANAGING** by giving the team an open work space and make sure the iteration ends on time. If you think you will not be able to finish everything by iteration end immediately plan an iteration planning meeting.
3. **DESIGNING** by keeping it simple. It should be simple so that the people can understand it without facing any difficulty. Secondly, it should make naming classes and methods consistent. Create spike solutions to reduce risk. Refactor whenever and wherever possible.
4. **CODING** as the major requirement of extreme programming is to make the customer available not only to help the development team but to be a part of it as well. All the phases requires communication with the customers. Code must be written to agreed standards, code the unit test first. All production code is par programmed only one pair integrates code at a time. Developers should integrate the code after every few hours and integration should happen very often by using collective ownership.
5. **TESTING** says all codes must have a unit test which should be passed just to make sure that the code works before it is released. When a bug is found tests are created. Frequently run the acceptance test again and again.

As conflicts and problems arises everyday. A good scrum master spends most of his time in resolving problems and issues. The coach recognizes the opportunity and understands why the scrum teams uses iteration. He is well aware of the fact that iteration can be ineffective if the team is allowed to include the work that isn't done. As the working software is the primary goal of the progress and delivering incomplete

software at the end of the sprint gives the customers and users false hope. The agile team values customer collaboration which doesn't mean that customer is always satisfied and happy but if the customer has any problem then the entire team should work together to resolve that problem and provides the service of good customer satisfaction.

An Extreme programming coach understands embracing change and incremental design. Hence, XP and scrum are backward looking at the project and finds ways to improve it. This is the way to help teams learning from their failures and focusing on solving problems more efficiently. One should have a right mindset which is very important aspect of efficiency.

### EXTREME PROGRAMMING AT GLANCE

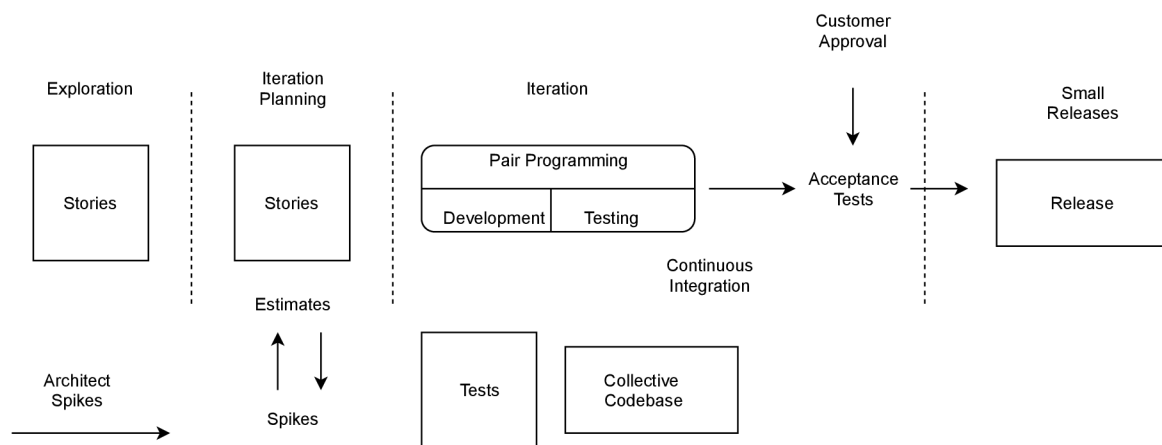


Figure 3.3: Overview of Extreme Programming

### SCRUM vs Extreme programming (XP)

1. Extreme programming teams always works in strict priority order Features that needs to be developed are prioritised by the customers (Scrum's product owner) and the team has to follow it without any deviation whereas scrum teams don't work on highest priority order.

2. Scrum teams work on iterations called sprint which is two weeks to one months whereas XP teams works on iterations that are one to two week long.

3. Scrum teams are less flexible in allowing changes into their sprints whereas XP teams are much more amenable to make changes within their iterations.

4. Scrum team don't authorize any engineering practices on the other hand XP team does like test driven development which focuses on automated testing, pair programming, simple design and refactoring.

5. In Scrum the scrum masters communicates with the customer whereas in XP customer is available and communicates with the team directly.

6. In XP customer itself defines the order to fulfil development tasks whereas in scrum developers decides themselves what they need to develop based on the priorities.

7. Scrum focuses mainly on the management side of agile as it emphasizes on the activities done but not the coding part as how work is actually done and product is actually built that is done by XP.

8. XP sprint focuses on creating a working bug free system whereas scrum sprint focuses on product release and their result in the working product.

9. Both the agile software development frameworks focuses on delivering high quality product to customers as fast as possible.

10. Lastly, Scrum values include openness, focus and commitment whereas XP values are communication, simplicity and feedback. Courage and respect values are common for both the methodologies.

Hence, both these methodologies are very important and useful in their own perspectives and ways. And both works one after another and co ordinate together to serve common purpose and are highly productive frameworks.

In this start up company their is implementation of both these methodologies that works simultaneously.

## **3.3 Relational Database Management Systems (RDBMS)**

### **3.3.1 Introduction**

According to (Χρυσομάλλη, 2018) In 1970, Codd published a paper which became the foundation of modern relational database management system (RDBMS'S). The main mostly used RDMS's are (MySQL, Oracle, Microsoft SQL Server). Codd presented the relational data model which allows data independence from applications, by which data can be modified without affecting the applications. Which is mostly used in Banks where the same data is used by different applications in different ways. Data is stored

in tree structure in files or in network model where data was access difficult because it was dependent on data access procedures. The relational database displays data in a very natural and simple way with the help of mathematical notions of sets with data viewed as tuples by creating a table structure for each relation or domain. This is defined as relational algebra which is set of operations in SQL. SQL uses join, where, select statement to access data and also to manipulate it. It doesn't make a DBMS but a medium to communicate to DBMS. SQL commands are mainly used to query, insert, delete and update data. It is a non- procedural database language. The first DBMS that supported SQL was ORACLE. The main benefits of this language was it enhanced productivity, reduced the cost of training and it stays for longer time which reduces the pressure to rewrite older applications again and saves time. SQL commands id divided into 3 categories are: Data Definition language (DDL), Data Manipulation Language (DML), Data Control Language (DCL). Data in relational model is stored in the form of tables which further consist of rows and columns. Oracle provides number of data types and its categories of user defined types. Some of the built in data types are string datatype to store character, number datatype to store numerical value and date, time datatype to store event happened which are further explained in sub categories very well and detailed. Therefore, SQL and PLSQL is used for communicating with the database like ORACLE, Postgre sql and many more. Through these commands you don't have to ask how to communicate but what data you need from that database.

### **3.3.2 What is Relational Database Management System**

Data is the heart of database management system. As all the data will not convey useful information we can only get useful information from processed data. And relevant information is the key for decision making which is further key to organisational survival. As data is the most important part of an organisation information system. Every company needs to save information about their employees, departments and salaries allotment to each employee which is data (piece of information).

Database systems stores and interprets the data which is its primary goal. A database contains huge data which is needed by the organisation and also its used for accessing data by large number of users called database systems. Many users access this data in an organisation which maintains the integrity of the data. Integration of data occurs when

same information is not recorded in two places.

DBMS is a set of interrelated data which is called database management system referred to as database. According to (Ambler, 1999)The DBMS like oracle , postgresQL contains huge amount of data which is useful for library management, university as it stores and access data. It is complex system which allows the users to input, share, edit, manipulate and display the data in database. Also it allows more than one users to access the complex data and the complexity extends to design and implementation.

### **The main objectives of DBMS are:**

1. **DATA AVAILABILITY** - which means availability of data to number of users as reasonable cost so that the users can easily access data.

2. **DATA INTEGRITY** - refers to reliability of data.

3. **DATA SECURITY** - means only authorised users can access the data if two users are working together on a same data the DBMS should not allow them to make conflicting changes.

4. **DATA INDEPENDENCE** - Allows the users to store, retrieve and update the data various complex data structures are used to represent data. Also they hide the specific information on how the data is stored and maintained efficiently.

**DBMS** helps in defining, accessing and manipulating data. By using this approach same data is shared by different application programs which reduces the data redundancy.

### **Advantages of DBMS**

There are 3 main advantages of DBMS;

1. Centralized data management
2. Data independence
3. System integration

### **Centralized Data Management**

In DBMS files are integrated in one system which reduces redundancies hence makes data management more efficient.

### **Data Independence**

Data independence can be physical data independence or logical data independence. Physical data independence means the application don't have to worry about how the

data is stored and physically structured. Application work with logical data model by using query language. If there are any changes which needs to be made in data, the application program has to be rewritten.

Therefore, Physical data independence modifies physical schema without affecting the conceptual schema or application programs to be rewritten. Logical data independence modifies the conceptual schema without changing the external schema or application programs.

### **Data Inconsistency**

Data inconsistency means same data should have different copies which will have different values. For example the details of an employee working in an organisation should be stored in branch office and main office. If an employee changes his address then the changes has to be made in main office and branch office as well.

### **People Interacting With Databases**

People who manages database- Database Administrator, people who design the Application program- Database designer and people who interacts with the database- Database users.

**DATABASE ADMINISTRATOR** - A person who has central control over data and programs who access that data. It focuses on the management of technical aspect of database system. Also it supports the development and maintenance of database application. The main responsibility of DBA is to maintain the integrity, security and availability of data.

**DATABASE DESIGNER** - They are logical database designers and physical database designers. Logical database designers should understand the organisation data and its business rules. The physical database designer uses the logical data model and decides how it can be physically implemented.

**DATABASE MANAGER** - It ensures integrity enforcement, security enforcement, backup and recovery.

**DATABASE USERS** - They are application programmers and end users. Applications programmers writes application programs and interacts with data through Cobol, Pascal and C.

### 3.3.3 Relational Data Modelling

According to (Sumathi and Esakkirajan, 2007) As we learned that the relational model presents the data in mathematical form, it presents the data as a set of relations. Relations can be in the form of table, rows each representing a collection of related data types. To interpret data in the form of relational model, tables and columns have names. Table is called as relation, column name is an attribute which forms a part of specific domain. Values appearing in each columns is called the domain of an attribute. A row is called as a tuple. (Elmasri Navate 1994).

As per (Sumathi and Esakkirajan, 2007) Data type is defined as the domain of an attribute, a name and a format, presenting the smallest element of the relational model. Each attribute is described by the domain. Before database schema domain needs to be defined. Schema defines the relationship stored as well as definition of each relationship in the database. Schema defines the set of relations and each relation is defined through set of attributes.

Sometimes the data is not available when we are defining schema which can be controlled by NULL element allowing the storage of an attribute if value is not available.

Each tuple represents a unique relation in the schema. Relational model have different types of unique identifiers called keys. Primary keys are the main reference key of a tuple. As each tuple have a unique or primary key to describe it (Elmasri Navate). It is the main reference key of a tuple.

Relational algebra also helps in manipulating relational data which consist of two types of operations one derived from mathematical operations and second is specific to relational database whose simplest operations are select, project and join.

### 3.3.4 Limitations of Relational Model

This section discusses the constraints imposed by the relational model when by implementing a relational database management system. Relational models are designed to store data for business processing and meeting the needs to support transactions which means changing of data from one system to another. For better transitional integrity in RDBMS's transactions should be frequent which is called ACID properties. While updating the transactions all viewers must view all the changes, it should occur one after

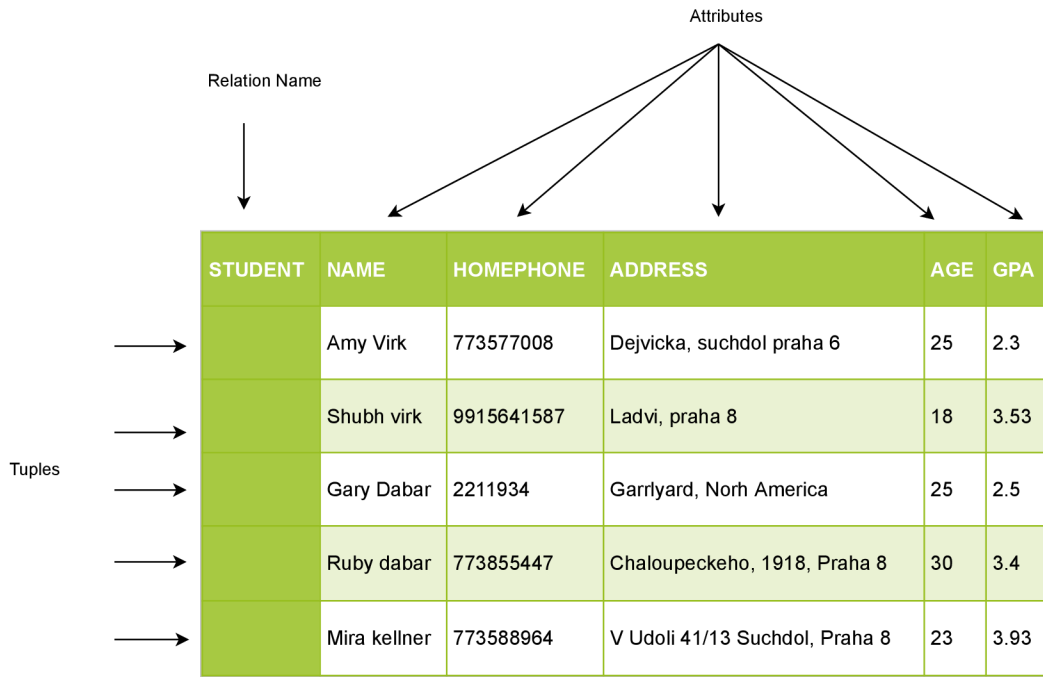


Figure 3.4: The Attributes and Tuples Relationship

another and once its completed changes cannot be lost. Due to growing amount of data the relationship between the data becomes complicated due to which information cannot be shared from one system to another which is a huge drawback. Also the maintenance cost is very high as highly trained staff is needed to handle huge amount of databases. If the database is changed or modified then the entire database has to be changed or converted to new form then the cost even exceeds the cost of database creation and management. Hence, mostly organisations prefer to work with their old databases just to avoid the wastage and maintain high productivity and efficient utilisation of resources.

Transaction must transform the data in such a manner that it follows to ACID properties. The implementation of ACID properties in RDBMS's has made the data over multiple servers difficult. ACID properties don't fulfil the requirement of consistency, availability and tolerance to network partition in distributed environment you can satisfy only two requirements which is called CAP theorem.



### 3.3.5 Implications of Constraints of Relational Database

#### Scale-ability of Data Storage

According to (Ambler, 1999) the data cannot be stored on a single database server there is a huge requirement of scale-ability of data storage. Scale-ability is the state where large volume of data is stored on multiple servers. There are different approaches discussed to solve the data storage scaling issues. Whenever this kind of issues takes place an architectural approach of scaling data storage is used for partitioning of data. The first method is vertical scaling by adding better hardware to solution more servers are added and solution is clustered and catching through the use of alternative database implementation. Data partitioning splits the applications across different data partitions.

For spreading data over multiple servers, clustering is used. It is having a cluster of database servers, where each stores different partitions of data. The disadvantage is that data on different nodes cannot be joined easily.

Catching ensures the storage of frequently accessed data in the main memory. In order to speed up the data queries Mem-cached catching system is mostly used. When a user sends a request, an in-memory cache is queried first to check if the data is present before accessing it.

Still there are difficulties which can be solved using third approach which is the use of alternative database implementation which is operated in distributed environment.

#### Managing Distributed Database

As per (Martin, 2002) Distributed database is a collection of data which belong to same system logically spread across various sites of computer network. The data in the distributed database is spread across different computers of a computer network. In the distribute database there is a hierarchical control structure which is purely dependent on global database administrator who has the central responsibility of the entire database and the local database administrators who is responsible for only localised database. It has the following advantages:

1. Data Independence: The main advantage of the data independence is programs are not affected by the changes in the physical organisation of data.
2. Low Redundancy: Redundancy is reduced for two possible reasons. Firstly, On

the contrary of having several copies of same logical data there is only one copy of data being used. Which is increase in the localisation of applications if data is distributed at all sites where applications need it. Secondly, the site failure don't stop the execution of applications which increases the availability of systems.

3. Complex physical structure and efficient access: In distributed data complex accessing structure are not the perfect tool for efficient access. Following software components are used for building a distributed database are:

1. Database communication component
2. Database management component
3. Distributed database component
4. Data Dictionary

### **Programming abstraction**

As per (Sumathi and Esakkirajan, 2007) Here the accessing data for manipulation and client request pro grammatically is being discussed. As we all know that the RDBMS's is very popular database management system and has numerous access in data centres around the world. Hence, to access data access and manipulation we use SQL (Structured query language) to manage data in database. SQL is used to manage, extract and manipulate information. It is designed by IBM as an interface to system R which was called SEQUEL.

Apart from managing relational database, SQL is used to manipulate and retrieve data. Retrieving is done through SELECT query and manipulation is done through update statements like INSERT and UPDATE.

The advancement in programming language to an object oriented approach created a data representation mismatch between relational data and programming objects. Each property of an object should be mapped to a column to be stored in the database.

Manipulating and managing data at application level by using programming language there is a significant performance advantage in data structures by using RDBMS. In a study by Jacob a 100 GB sample data was used to represent sample data which consisted of 6.75 billion records further representing a person and columns for age, gender, language, ethnicity, language and religion the result is extracted in less than a minute. Hence, to compare these results with relational database performance, the data set was imported

by commonly used PostgreSQL database. Non-relational database implementations provided data access at API (Application programming interface) level.

### **New approach to data management and analysis in distributed environment**

According to (Martin, 2002) As we know distributed data needs to be managed efficiently. There is the need to analyse distributed data for data warehousing applications, which is a very costly task to be performed on data. Apart from relational databases there is a need to analyse data in distributed environment using an alternative methods.

The analysis is done on dynamic data set by updating the existing documents or adding new documents continuously. Processing the volume of data means that data and content needs to be stored in distributed database. All the output sets are grouped together and passed to a reduced function which is created by programmer or a user. The reduced function accepts an intermediate key, and hence these values merges together to produce a smaller set of values which will produce either zero or one result.

For internet related data, analysis is based on large amount of unstructured data. Extracting information from unstructured content is a computation intensive process which involves character level operations such a Reorganisation. Processing of the volume of data means that data and content needs to be stored in distributed database. The analysis is done on dynamic data set by adding new documents or updating existing documents continuously.

## **3.4 Non- Relational Databases**

According to (Ambler, 1999) Non- Relational Databases is an alternative DBMS's in detail. A non-relational databases like other traditional database systems does not use tabular schema of rows and columns. Instead, it uses storage model for storing the data for specific requirements. Like, NoSQL, MongoDB, DocumentDB, Couchbase which are further grouped into 4 categories: Key-value stores, graph stores, document stores, column stores. According to (Celko, 2013) The main focus is on the alternative DBMS'S in detail which is NoSQL to handle large data workloads with high scalability. Different data models are used by this alternative DBMS's explaining how they are implemented and what their features are, introduction of alternative databases and different database fam-

ilies like column oriented databases, key/ value databases, document- oriented databases and graph databases in addition there are also various application areas of non- relational databases.

NoSQL is a non- relational database which does not follow the traditional method which relational database uses. Data access in NoSQL is provided by simple query API. Storing user generated text based data in distributed environment requires the need for alternative data models. In addition to data models there is also a need for scalability of data storage over multiple database servers. There is also a need to investigate the level of programming abstraction provided by these database systems. User generated text based data is less dependent on a pre- defined schema and is prone to complex queries in data warehousing applications. There is a need to investigate the programming abstraction provided by these database systems.

According to Cattell, NoSQL you can scale your system horizontally by taking advantage of cheap commodity servers. The classification of NoSQL databases are as follows;

1. An efficient criteria for managing distributed indexes.
2. Dynamic schema or data record attribute changes.
3. Weak consistency model.
4. Partitioning of data over multiple servers.
5. Scalability of simple operations like retrieving, inserting and updating data should be executed over multiple servers.

High availability over multiple servers makes scalability more useful and is one of the main goal of NoSQL systems. (Cattell 2010b).

According to (Ambler, 1999) The concept of Horizontal scalability achieved by NoSQL databases requires terminologies. The concept of "clusters" recapitulate in distributed computing environments, collection of interconnected computers which are working together on a single integrated computer resource known as clusters. Clusters are interconnected computers connected to network called node. A cluster consist of many nodes. By using high speed networks, multiple database servers or nodes are connected to one another. A cluster refers to a collection of database nodes working together to present clients with a uniform database whereas data may be distributed over several nodes. The client is only aware that it is updating a database while accessing the database.

According to NoSQL website, the alternative database model that differ from rela-

tional database are column- oriented database, key value database, document database and graph database.

### 3.4.1 Column- oriented database

In 1969, TAXIR was the first built column oriented database storage system. Column-oriented databases initiates from data analysis and business intelligence by addressing the scalability problem over multiple distributed servers. Applications related to data warehousing need fast access to data which returns the query results from different columns.

A cluster refers to a database nodes working parallel to each other and presenting clients with a uniform database, although data is distributed in several nodes. Data Tables are stored by column rather than by row.

Column oriented database provides a data model closely related to relational model. The main difference between relational database and column oriented database is that relational database stores null values for each column which even don't have a null value whereas column oriented database only stores available data in the database. Column oriented databases store data in a row, which can store an arbitrary number of key/value pairs.

#### How it works?

Although all the values are of same type and drawn from same domain, calculating the value of nth row is easy. The columns are in the same order as in the original row, so to present the ith row, go to the ith position in the relevant column store and concatenate them.

Lets take the example of a phone number, go to area\_codes, phone\_exchange and phone\_nbr column stores and find each ith record in parallel. As the area codes are smaller they comes back first after exchanges and finally the phone numbers.

Many columnar databases are row- based optimizer which reflects many benefits of columnar storage. They materialize "rows" first in the query execution and process them with a row based optimizer. We can add table name in the columnar descriptor by changing it to domain descriptor: table\_name, start\_position, end\_position, data\_value.

## Doing an ALTER statement

ALTER statement changes the structure in the schema. In the columnar model ADD COLUMN and DROP COLUMN are very easily accessible. New columnar structure can be created and old one can be removed very easily from physical storage. In the row oriented model each row can be expended with the alteration.

### 3.4.2 Graph Database

These databases are useful to model the relationships rather than traditional structured data. Like relational databases are based on sets, graph databases are based on graph theory they are not network based. A Graph has two things in it there are edges (arcs) and nodes (vertices); edges are drawn as lines that connects nodes that are further drawn as dots or circles. These are the special kind of databases for efficiently managing heavily linked data. They are mostly used where there are many relationships between data exists. The simple example is FlockDB, which is twitters implementation of storing relationships between users. They are mostly useful in location based implementation, navigation systems and recommendation systems.

#### Nodes

Nodes are abstractions. The most important characteristics of a graph theory is that a node can model a sub-graph. According to, (Celko, 2013) A node is not an object. Objects have methods and local data inside them. In a complex graph query, the focus is mostly on an unknown and nonexistent node. For example a bus stop with a Romanian barbeque stand might not exist. But a bus stop with a barbeque in a Romanian neighbourhood might exist and hence we only get to know it until we get many factors together (e.g. riders getting off at the Romanian center bus stop, restaurants or romanian churches within n blocks of the bus stop etc). There are various types of graphs as such:

1. SCHEMATIC Maps: the nodes are the bus stop, town and so forth.
2. Circuit diagrams: the nodes are electrical components.
3. State transitions: the nodes are the states ( this can be modeled in SQL)

## Edges

Edges are arcs connecting nodes. In schematic maps, the edges are the roads having distance and time on it. In the circuit diagram, the edges are the wires that have resistance, voltage etc. Edges are connecting the abstract state transition also model the legal transition paths. Edges are more interesting than nodes in graph theory. Therefore, in graph database we can have multiple edges of different kinds between nodes.

## Graph Structures

According to (Celko, 2013) since graph theory is newly introduced by mathematical standards which means before 500 years ago, there were lots of open problems and authors were using different terminology like :

1. A Null Graph is a set of nodes without any edges. A complete graph has an edge between every pairs of nodes. Both these extremes are very rare in graph databases.
2. A Walk connects nodes without repeating an edge.
3. A Connected graph is set of nodes were two nodes can be reached by a walk.
4. A Path is a walk passing through each node once. If you have  $n$  nodes, you will have  $(n-1)$  edges in the path.
5. A cycle or circuit returns to where it started.
6. A Tree is a connected graph having no cycles.

According to, (Du Toit, 2014) The research focuses on the storage of user generated text based documents in non relational databases where graph databases are useful in storing relationships between objects. That's why they have been excluded from the research. There is an introduction of graph database, but the storage model falls outside the scope of this research.

Hence, the development of alternative database management system is the result of limitations defined in the relational model while storing distributed data. The scalability and alternative data models used create opportunities for application in different areas.

## RDBMS Versus Graph Database

Due to generalization, the graph database worry about relationships, while RDBMSs worry about data. RDBMSs faces difficulties with complex graph theoretical analysis.

Its very easy to make use of graph theory where the length of every path is one which is just a three column table (node, edge, node). By using self- joins, paths of two lengths can be constructed called a breadth- first search.

### **3.4.3 MongoDB**

According to (Celko, 2013) Mongodb is a document oriented databases as alternatives to relational database. It provides a document querying mechanism. It provides indexes on document collection also its a lockless implementation.

This thesis discusses about the data model, scalability of data storage and the programming abstraction it implements.

#### **Data Model**

MongoDB stores document also it recognises the structure of each document. As we know document oriented databases use JSON- like structures to present document. MongoDB is case sensitive which means that if the type of the value of a key/value pair is different than the type of the key/value pair in the same key, the documents are different. Documents are viewed in the row format in relational database. MongoDB contains high level containers information in the databases. An instance on a server can have zero or more databases.

#### **Scalability**

MongoDB allows data storage to be measured over multiple servers in distributed environment. Multiple databases are present in MongoDB instance where each database is separate from another having its own set of permissions. By automatic sharing, MongoDB can distribute documents over servers in distributed environment.

In the above figure, in order to share data over multiple database nodes in cluster an instance is required. Also there is a need to have config servers for storing the metadata of clusters which is hosted in mongod instance. As result through the routing service called mongs queries from the clients are then directed to appropriate shard on mongod instance.



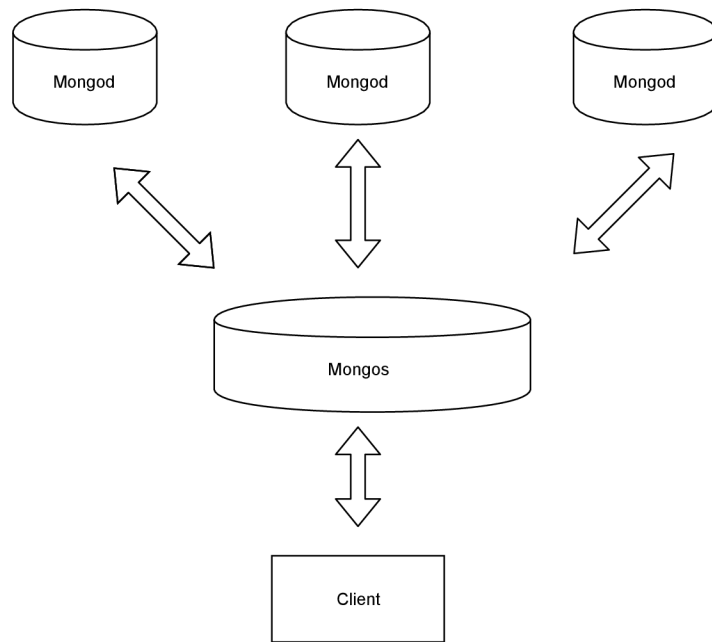


Figure 3.5: Shared Client connection

### Programming Abstraction

As per (Du Toit, 2014) MongoDB is a document-oriented database which adopts JSON-style document model. This is written in C++ and client access is provided in JAVA also the interactive shell is provided to access data using MongoDB's own query language. As MongoDB stores JSON documents data stored is based on JSON representation which supports all the basic data type supported by JSON. It serializes the JSON document to binary presentation. Binary large objects (BLOBS) are also supported in MongoDB, that allows the server to store images and videos.

INSERT method of collection command is used to insert data. Keys are added automatically. For removing data, REMOVE method is used. MongoDB allows document to be updated through UPDATE command of a statement. This command is atomic, the UPDATE that reaches the server first will occur first. MongoDB has modifiers to UPDATE few parts of the documents instead of updating the whole document. Apart from update MongoDB has an UPSERT command, that inserts the record even if it does not exist otherwise the existing record will be updated.

FIND command is used to retrieve record from the database. If no query document is provided then the entire document will be displayed or returned.

Hence, MongoDB is a document oriented database that provides horizontal scalability

capabilities. It allows queries and functions on the values stored also provides a programming interface with a large set of functions and features to access and manipulate data over multiple database servers.

### **3.4.4 Other Application Areas**

In the above section we have discussed various types of non- relational databases and their usages with languages.

According (Sumathi and Esakkirajan, 2007) NoSQL database system is basically used in cloud- based systems where other traditional database transaction requirements makes the implementation difficult. Various cloud serving systems have the following characteristics like scale- out, elasticity and high availability. Scale- out in database system is achieved when the databases are running on different commodity servers. And an elasticity allows the servers to be added and removed.

High scalability needs to be provided by the cloud system, as commodity hardware can be destroyed easily. However, the traditional RDBMS implements ACID properties that makes implementation in cloud environments much more difficult mostly for elasticity and scaling.

The scalability capabilities in NoSQL databases, makes much more easier and suitable for these types of implementations and provides high flexibility and efficiency.

Therefore, there are various application areas storing RDBMS but for modern internet scale data implementation, that requires high level of scalability for large volume of data. Alternative data models and data stores have been used and implemented in order to access scalability issues and handle large volumes of data which the RDBMS cannot. This is mostly used by multinational companies for supporting big data applications.

## **3.5 Business Process Modelling**

### **3.5.1 What is business process modelling?**

According to(Xρυσομάλλη, 2018) A Business process is a collection of activities or tasks that produce a specific service or a product of a particular customer. A process is usually represented through diagrams, flow charts. In each and every process an input produces

an output of a particular situation. Business process modelling represents processes of an enterprise, just to analyse and improve and understand current process in a flow through various diagrams. Hence, the output of business process model is Business model. Business model describes the principle of how an organisation creates, delivers and captures value with efficient and proper utilisation of resources with proper productivity.

Business process re-engineering is a fundamental rethinking and redesigning of business process to achieve improvement in critical measures of performance such as cost, quality, service and speed. The main areas of improvement in processes are effectiveness, efficiency, internal control and compliance of various policies.

Business process management is focused on aligning all aspects of an organisation with the wants and needs of clients. BPM attempts on improving business processes frequently. There are various phases of business processes such as Vision, Define, Model, analyze, improve, control and then re-engineer.

### **3.5.2 Where it is used?**

As per (Ambler, 1998) in today's world organisations needs proven techniques for managing the complexities of large scale, object oriented software development projects. Hence, a process patterns describes a collection of general techniques, actions and tasks for developing object oriented software. The object oriented software process presented is a collection of process patterns which are focused towards medium to large size organisations which needs to develop software to support their main line of business.

According to Scott W. Ambler (Ambler, 1999) the OOSP provides a framework which addresses issues such as how to:

- Successfully deliver large applications using object technology.
- Easily maintainable development of application and are hence enhanced.
- managing projects.
- ensuring the high quality of development efforts.

Above are all the related issue it could relate and face in the development process. It is used to measure the processes just to identify potential weaknesses and improvements.

### 3.5.3 Implementation of BPM

In this thesis the Business process modelling is a graphical representation of company's business processes and workflows for the identification of potential improvements by flow charts, data flow diagram and graphical methods. It helps in communicating various processes and information about the processes with their decision rules which further helps business managers to quickly communicate their ideas using craftcase tool.

In the implementation part the main focus is on the Development team and Tech support team. All the process pattern has been described in detail with diagrams using craftcase tool. In both the parts of an organisation there are 20 employees working parallel and coordinating with each other. As identifying, defining and communicating the roles and responsibilities is the fundamental aspect of a team to deliver. By clearly defining roles and responsibilities allows each team member in an organisation to focus on their work and provides the structure needed to perform better roles and responsibilities in an organisation.

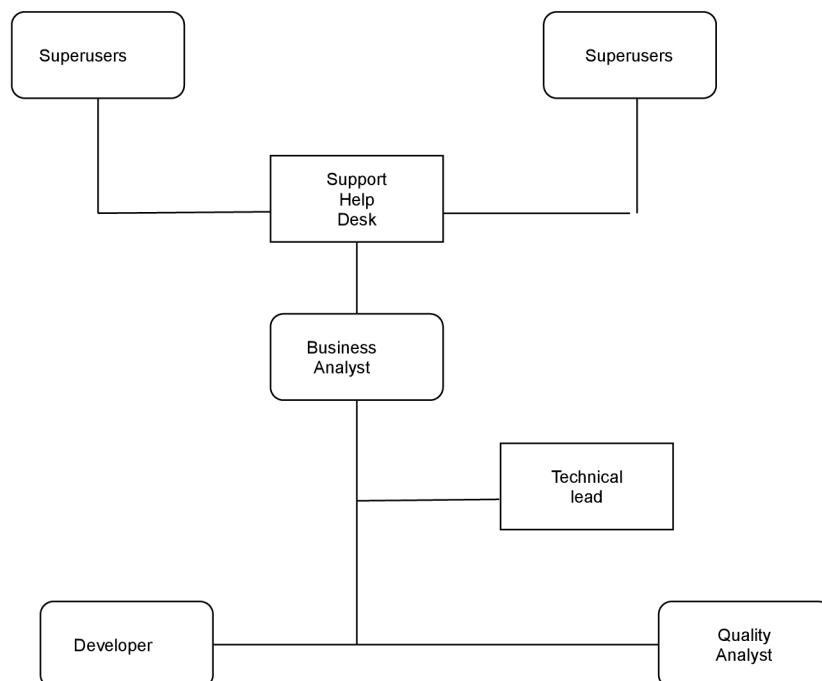


Figure 3.6: Framework of Business process Model

In the development team there are developers working in pairs using pair programming and also there is an implementation of extreme programming whereas on the other side there is tech support team working as Help-desk agent, IT support supporting various

applications and resolving technical issues.

Using agile methodologies the SCRUM team working on the functional side of an organisation which represents the collection of individuals working together to deliver a request or a service. It mainly consists of three roles: Product owner, Scrum master and Development team.

# 4 Application of the proposed Agile Methodology in Tech support and Development Team

Being a database oriented and agile start up company an application part consist of an architecture of process flow of start up company focusing on development team and tech support team. The main point of focus here is on Help-desk Agents and developers designing database.

## 4.1 Process flow of Projects

The process flow of projects mainly focuses on the Help Desk support team which is providing assistance to users technical problems by receiving phones, emails and resolving their technical issues also a good customer service approach. And the development team where developers are working in Co- ordination to each other mainly designing a database in an agile environment.

In figure 4.1 there is description of business concept map of Help desk support system which shows how the help desk agent provides support services to customers by using various service channels like phone calls, emails and ticketing system etc. How they resolve technical issues by various tools and provides the users good satisfaction with proper guidance. Their are various means of helping the customers but the one which consumes less time to resolve issues is preferred by the Help Desk Agent.

There are various teams resolving issues L1, L2 and L3. All these are the levels of the teams resolving technical issues according to assigned levels. L1 team resolves basic issues like Password Reset, Account Unlock, Software installation and update. L2 team

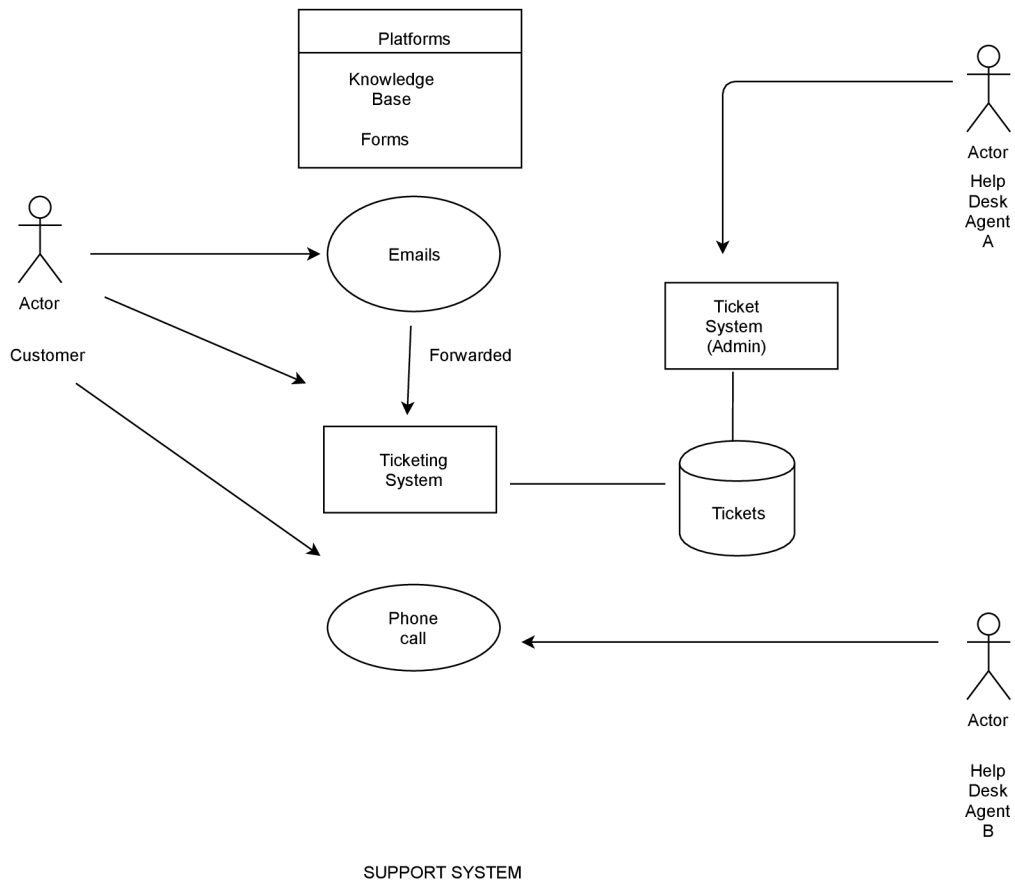


Figure 4.1: Support system

is the similar as that of L2 team and does the same level work as compared to L1 team. L3 team resolves the most difficult level tasks like java error, application error, page not displayed which is mostly been done by the developers. When the L1 support team is unable to resolve any issue they do the escalation process to the next higher level team who resolves the task assigned and also make changes and progress in the application.

#### 4.1.1 State Diagram Process workflow

Figure 4.2 state diagram describes the process workflow in IT Help Desk support system which explains the workflow from logging in to the system with credentials after log in we can create requests, update requests and delete customers and various technicians then the system needs to be logged out.

There are various tasks needs to be performed by the Help desk agent as they do all the maintenance of users issues and help them to use the application if they are not able to do that. They raise requests, update them and manage users related to escalation

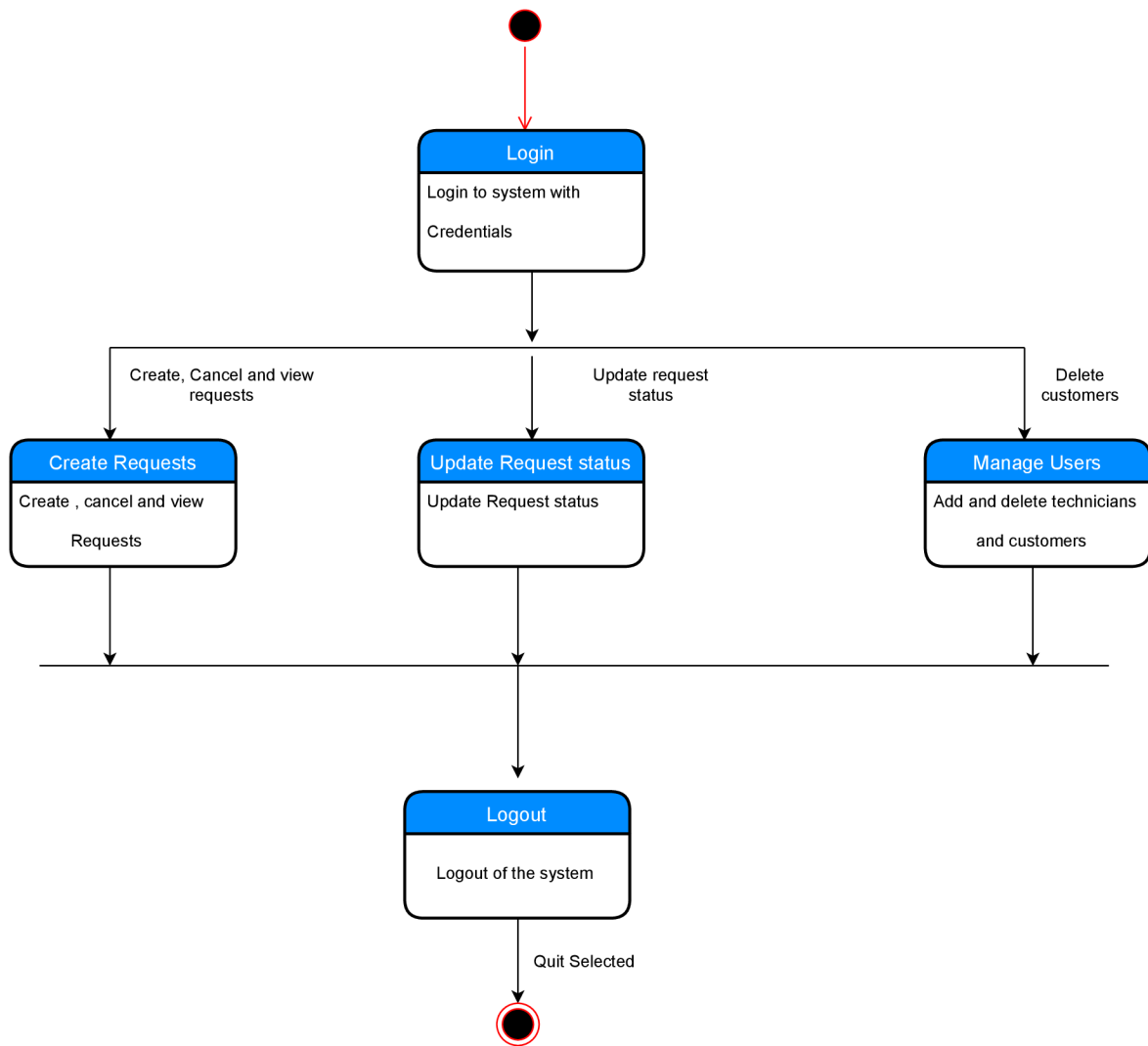


Figure 4.2: Process Workflow state diagram

process. And then at the end of all the resolutions they logout from the system.

Users also provides their feedback according to the service provided to them. If they are not satisfied with any held desk agent they can even complain to the manager as he/she is not satisfied with the service. The resolution of issues is also done via emailing process by receiving application and finding issues related to its usages by using SQL querying language.

#### 4.1.2 Activity Diagram for Help desk Management system

Activity diagram or Help desk management system describing flow of login activity, where admin login to the system using their username and password. After login user is able to manage all the operations on client, Ticket, ticket Type and issue.



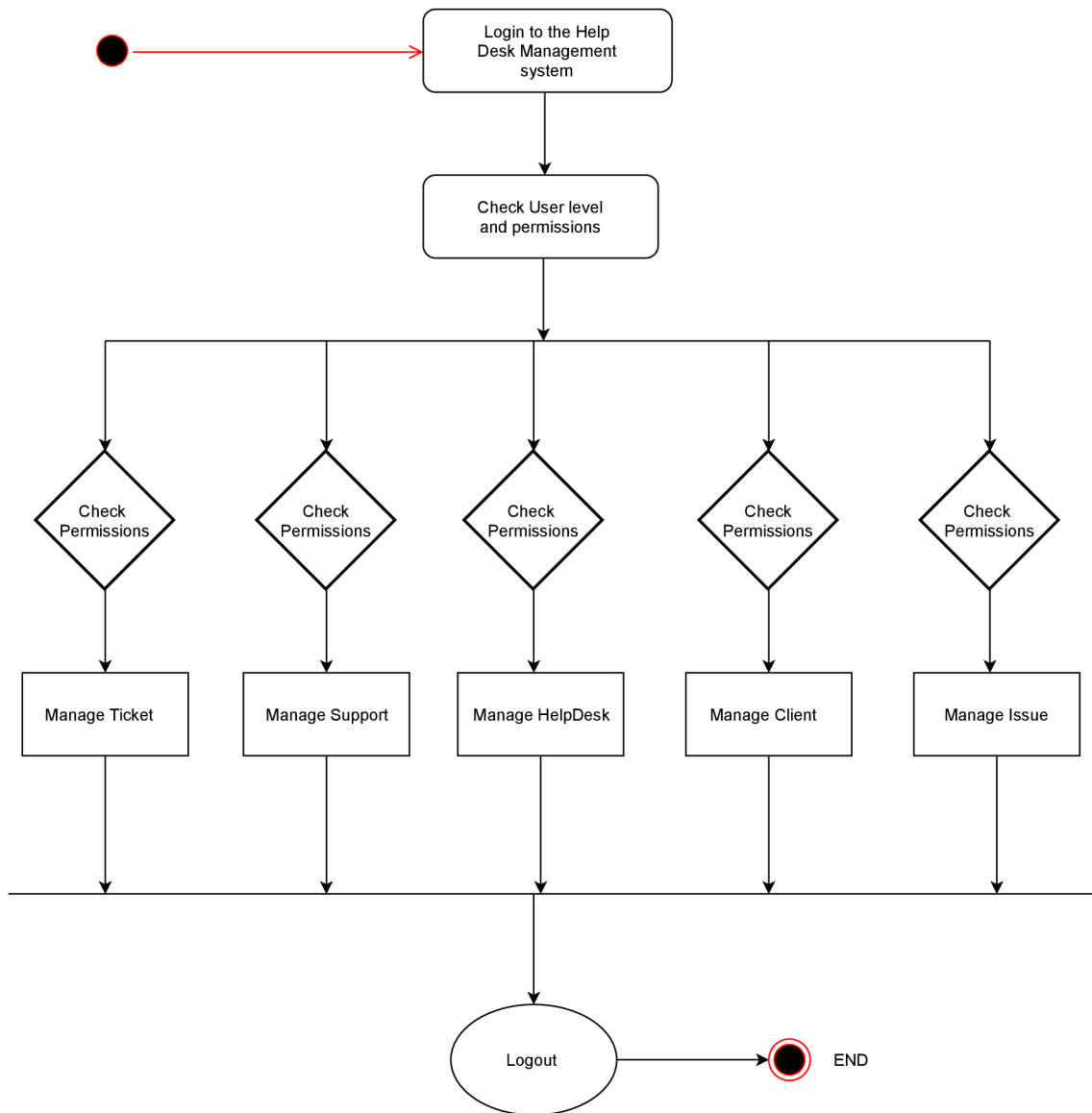


Figure 4.3: Help Desk Management system

### 4.1.3 Login Activity Diagram of Help desk

This activity diagram explains how the login page works in Help Desk management system where admin log in using their username and password. After user is logged in to the system can manage all the operations Ticket, Ticket type, issue, Help Desk and client by helping them resolve the issue. The diagram explains how the login page works in Help Desk Management System. Different objects in the issue Client, Ticket, Ticket type and Help desk page interact with each other over the entire course of activity process and without verifying their identity user will not be able to access this page.

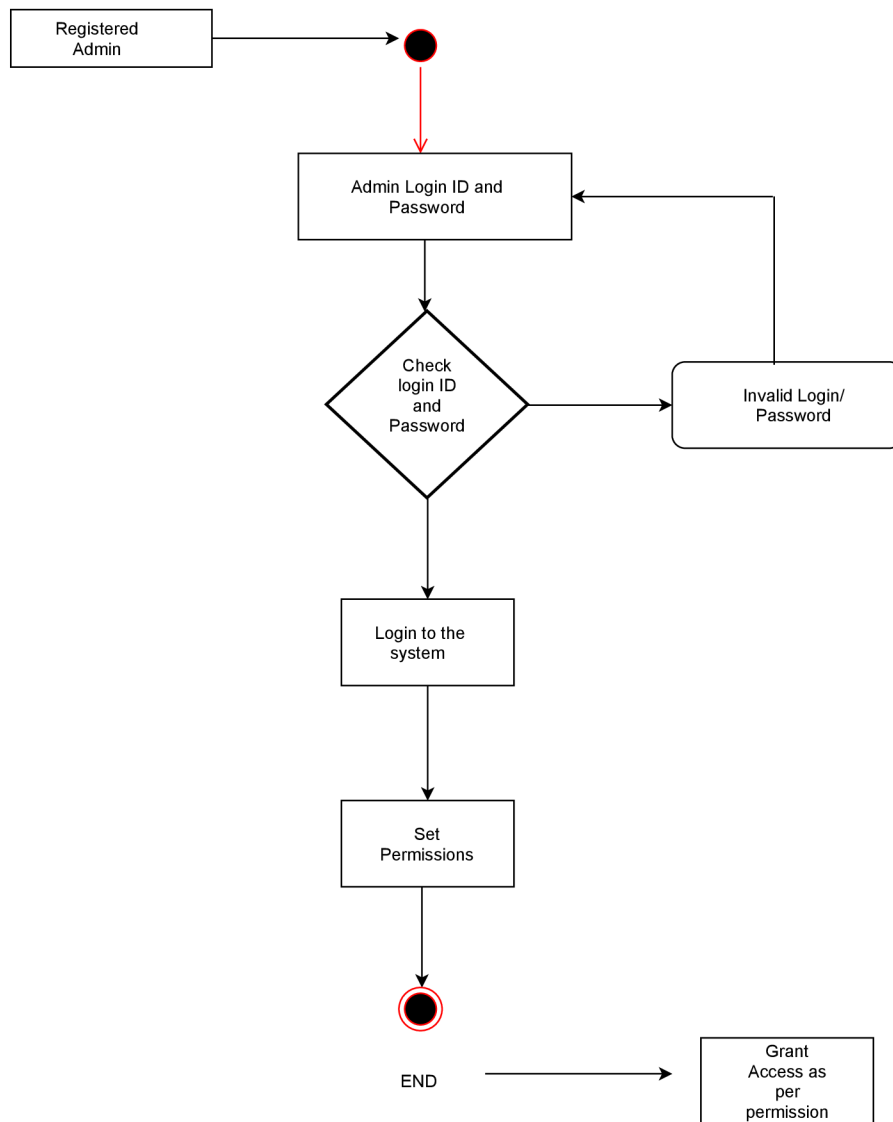


Figure 4.4: Login Activity Diagram

#### 4.1.4 Merging Activity diagram with release goals

Activity diagram is a flowchart to represent flow from one activity to another. The main aim of the release goal is to make sure what is the value and helps to guide what should be built. A story map is built to achieve the goals which are dependent between stories and business flow processes. In case the issue is not resolved by the L2 support team then it is further passed on to the L3 support team and resolved by them.

Below is an Activity diagram:

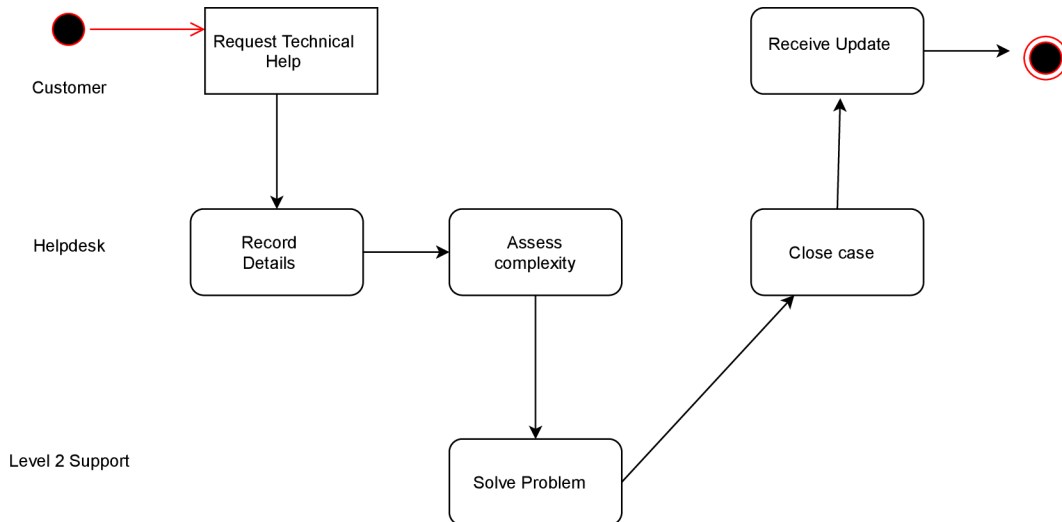


Figure 4.5: Activity diagram with release goals

## 4.2 Implementing Agile in Development Teams

As we have discussed about improving productivity in software development by using Agile methodologies now the focus is on the implementation of these methodologies practically. However, this process is applied to smaller teams.

The average team size is 11 which are highly dependent on other teams and hence better dependency management can improve productivity. Training has a positive impact on perception which makes agile development work better. There were 9 members in the team using Extreme programming Techniques which observed scientific gains in productivity. Three other critical factors are good project management, an agile friendly team environment and intense involvement of customers.

On the other hand the main focus is on the adoption of SCRUM on customer satisfaction.

The organisation is using models to guide management and deployment to improve the software process improvement (SPI). The paper guides lightweight model by using SCRUM methods to do some modifications for the base process specializing the company.

The customers and developers agree on the assigned tasks for each iteration. Hence, changing requirements can be accommodated. Solution are developed by the development team in collaboration with customers over sprint. After that there are frequent releases. Proper feedback refines and finally delivers a complete software system.

### 4.2.1 Product creation Framework: MILESTONES

Figure 4.6 Describes that every product manages its backlog for a full year which in 365 days. PCF is based on the release level which is made up of features. Each release is comprised of the features for which design solution needs to be implemented. Features are broken down into epics. After that user stories are developed and functionality Tested. The release is functionally complete and moved to staging so that it can be functionally tested. Regression testing as well as Performance testing is performed here as well. The release is then moved to production where user acceptance testing takes place. Hence, PC6 and PC7 could be a separate date depending on weather or not the release release goes through a pilot phase first. At the end, the release is on the production all the necessary relevant customer documentation and information is launched as well and the field and operation staff are ready to support the customer.

#### **RELEASE PLANNING**

- Milestone 1: Defining the delivery goals.
- Milestone 2: Designing the optimum solutions.

#### **QUARTERLY PLANNING**

- Milestone 3: Detail planning of the delivery.
- Milestone 4: Releasable complete features.
- Milestone 5: Certifiable release components.
- Milestone 6: Releasable components are ready for the customer.
- Milestone 7: Customer ready solution delivered to the customer.

Development release work in Scrum is done in cycles. It is iterative. The fixed length sprints are usually for two weeks which can be anywhere between one to four weeks. Each sprint reviews the previous sprint, prioritization and estimation, implementation, Testing and delivery. Focus is to review on finding what went well and what did not in the last sprint.

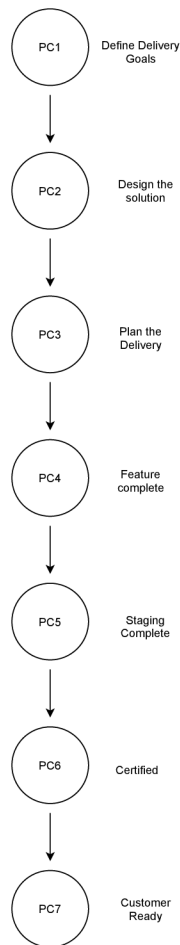


Figure 4.6: PCF ( Milestones)

## 4.2.2 The structural hierarchy in scaling agile methods

**TEAMS:** Teams consist of developers, Testers, Scrum master and Product Owner. Team consist of five to nine people. The team backlog consist of User stories.

**PROGRAMS:** Multiple teams form a program which have five to fifteen teams. Programs integrating components from different teams to form an entire product. Features and Epics are defined at this level.

**Portfolio:** A portfolio consist of multiple programs and strategic decisions are made by the people working at this level.

**User-Stories:** They are the smallest units of work and deliver particular value to the customer. User stories are written by the product owner in a very simple language. Later team together writes more detailed requirement.

**Epics:** Epics are larger units of work. These are development components further divided into user stories. User stories can be completed in biweekly sprints. Epics are

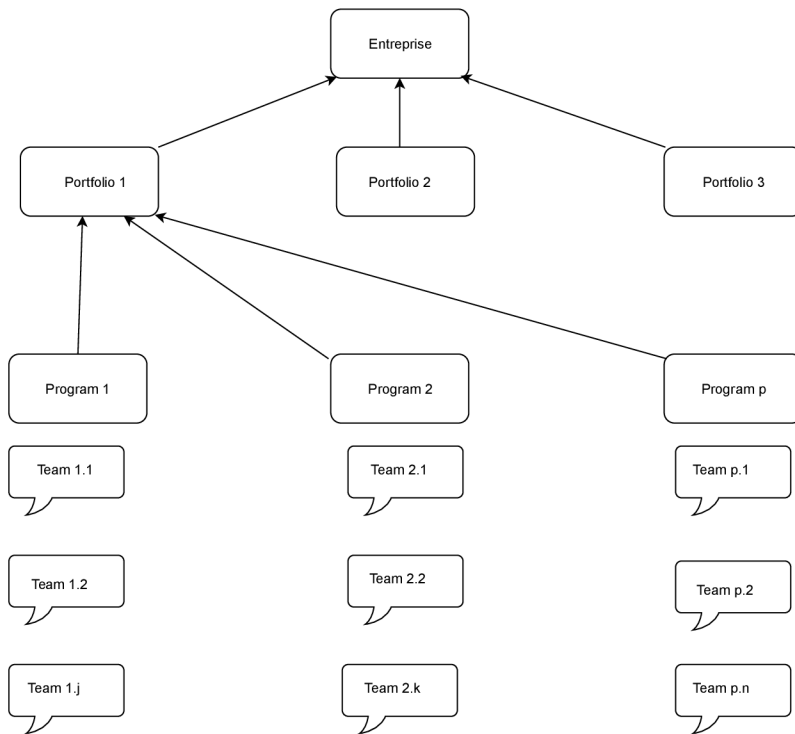


Figure 4.7: Hierarchy in agile methods

delivered over set of sprints. As a team gets to know more about epic through development and customer feedback, user stories are added to teams backlog and epic burndown chart helps to visualise epics which further keeps stakeholders informed about how the team is progressing and encourages open conversation about the evolution and completion of the product.

**Features and capabilities:** The teams are divided into features and capabilities which are deliverable functionalities. They are delivered into quarterly releases hence decomposed into epics.

**Investment Themes:** These are very strategic decisions explained at the portfolio level. Here products are developed at the budget, market requirements and several other factors by the stakeholders which are semi annual decisions determining workflow for the organisations. It comprises of functional target like redesigning a product or non-functional goals like migration from windows to Linux based servers. The figure below describes the structural hierarchy of scaled agile developed organisation.

### 4.2.3 BPR and BPM

Business process management is the process of managing processes its not an individual task for improving business performance outcomes and operational agility.

Business process re-engineering as compared to business process modelling is more specific methodology focused on integration of process management and radical improvement of processes. It is defined as integrated set of management policies, project management procedures and modelling, analysis, design and testing techniques for analyzing existing business processes.

Typically re-engineering project consist of analysis of "as-is" state of process, proposal of "to-be" state and implementation of "to-be" state. It is basically designed to involve changes in key areas of process involving business structure, communication, process-design, strategy and policies, organisational culture, technology and people.

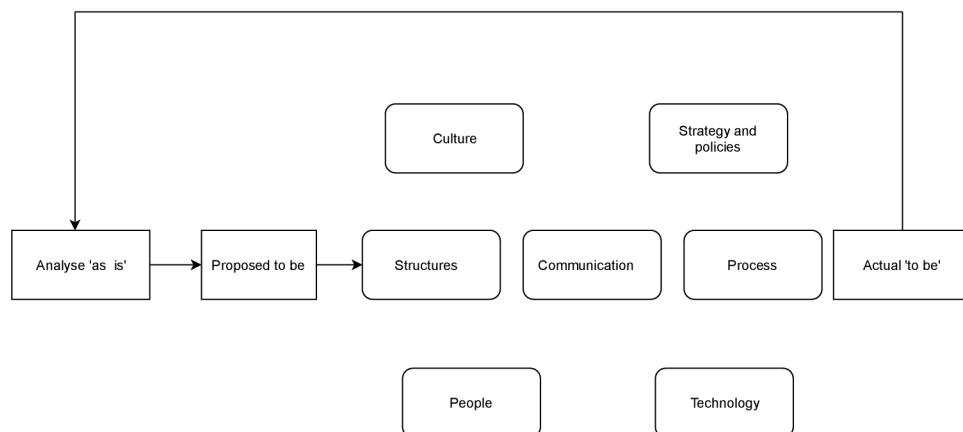


Figure 4.8: BPR and BPM

## 4.3 The Project and cross-Project Tasks of OOSP

The main objective of the support stage is to check and respond the incoming support request from users, to identify the resolution for the request and then implementation of that resolution. And the Developers put an application into into production.

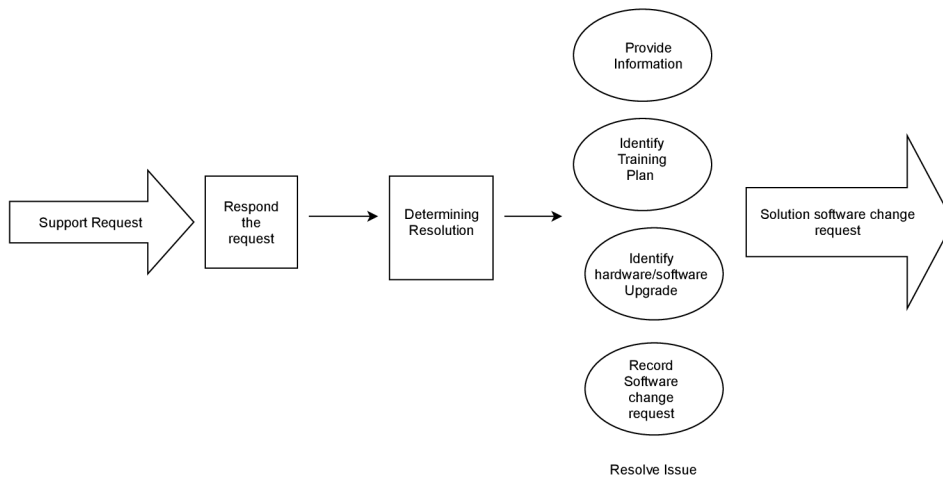


Figure 4.9: Support Pattern

### 4.3.1 People Management

The key of people management is focused during the support stage is one of maintaining the morale of your support staff. As lets say the nature of the support engineer’s job they deal with problems all the day, also they are abused by the people whom they are helping in an organisation. Effective techniques that i have used in my company is as follows

- Acknowledging support engineers who receive daily feedback from their customers.
- Online software games played after hours over your organisation’s network.
- Group outdoor activities such as football, basketball, chess etc.

### 4.3.2 Training and Education

Support engineers needs to be constantly learning a wide range of new skills even while they are at job.

### 4.3.3 Quality Assurance

The foremost quality insurance issue is good customer support to ensure that your user community is receiving good customer service. This can be done in many ways. Firstly, your support manager may choose to randomly monitor support calls as monitoring can provide support engineers with productive feedback for improving their support skills.



### 4.3.4 Metrics

There are several metrics available to the support stage which is taken for each specific application supported by, the type of support request like Phone calls, emails or other.

1. AVERAGE TIME RESPONSE. This is average time between when a support request is submitted and when support engineers begin working on it. Lets suppose the time taken by support engineers to complete on task.

2. AVERAGE RESOLUTION TIME. This is the average time when the support request is reported and the time when solution is accepted by the customers. Lets sat it is the length of time the user expect to have their support request resolved.

3. SUPPORT REQUEST VOLUME. This is the number of support request submitted, and closed over a given period of time.

4. SUPPORT BACKLOG. This is the number of support request currently being worked on either by individual or by the entire support team.

5. SUPPORT ENGINEER EFFICIENCY. It is the number of support request resolved by the support engineers over a given period of time. For this metrics, i need to perform a trend analysis- which states that during that period an engineer may get lucky to have simple requests to resolve while on the other side can stuck in difficult requests to deal with.

### 4.3.5 Success Factors

- create an image that every support request is critical.
- Support engineers are the sales team for your application.
- Main objective is to support your user community.
- Support request is not resolved until the requester is satisfied.
- To provide good customer service.

## 5 Case Study

The main purpose of my case study is to represent my own practical experience working as customer service specialist where my customers were commercial carriers, fulfilment centers. Our main objective was to avoid customer impact by handling the cases on time and within SLA (Service level agreement) which leads to good customer service. There were various tools involved in the service which were caps dashboards for measuring the capacity of volume, skeds viewer which was used to view the CPT trucks for our transportation, truck fill assumption for measuring the fill rate. There were lots of blurbs utilized in the service criteria. We were calling our customers which are carriers for negotiating the price of our bids which we are selling at our online bidding system platform called relay load board. The teams are basically divided into two parts inbound and outbound. Being from an outbound team I was taking care of the capacity where we were mainly focusing on the caps reduction, replacements, cancellations and providing trucks.

The teams were divided into classes and parts based on the division working in an agile manner with proper co-ordination with each other. As there were specific agents allotted to each queue and all of them were responsible for maintaining SLA along with handling the queues. The higher authorities were mainly responsible for checking the injection flow of the volume and hence works accordingly. There should be mainly 40 resolved cases everyday for better performance. There was an online bidding system platform where we were selling bids by negotiating the prices with the carriers as per their request. Now i m going to represent it diagrammatically for better clarity.

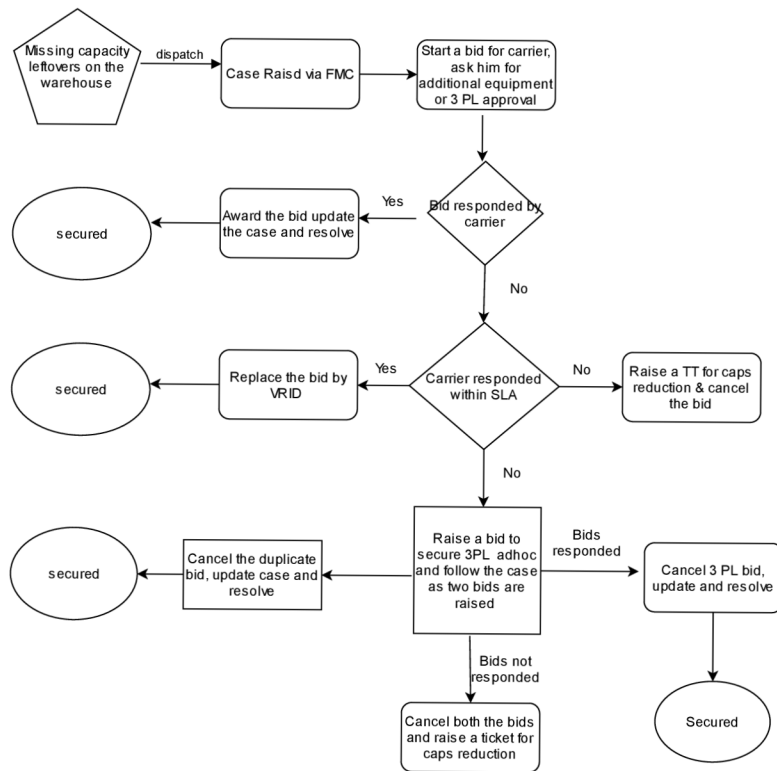


Figure 5.1: Decision Process Pattern

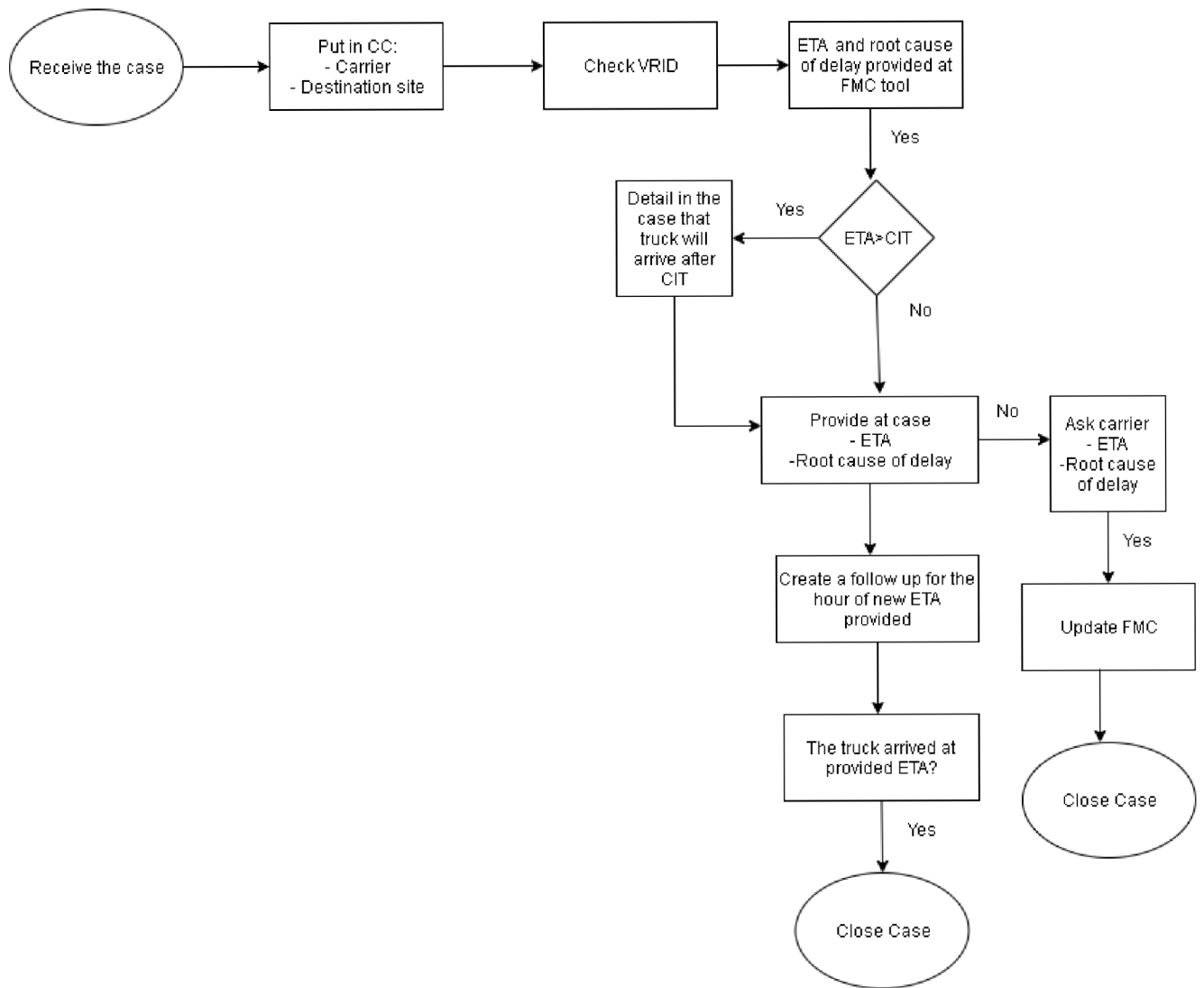


Figure 5.2: Decision Process Pattern

# 6 Practical Application

Final Report along with scenario, simulation and Business Architecture using craft case.

## 6.1 Scenarios

### 6.1.1 Development

Development	
<i>initiation:</i> Team develops code to build software.	<i>roles:</i> <b>Tester</b> performs
<i>action:</i> Programmers are working in pairs or individually.	
<i>result:</i> Software is build.	
<i>Integrated functions:</i> App SW Development	
<i>Is followed by:</i> Testing	
<i>Derived diagrams:</i>	

Figure 6.1: Development Team

### 6.1.2 Operation

Operation	
<i>initiation:</i> Teams are operating in an agile manner.	<i>roles:</i> <b>Tech support specialist</b> cooperates
<i>action:</i> All the workflow is in process.	
<i>result:</i> Customers are satisfied.	
<i>Integrated functions:</i> Help Desk	

Figure 6.2: Operation Team

<i>Follows:</i> Testing
<i>Is followed by:</i> Transport Support Team
<i>Derived diagrams:</i>

Figure 6.3: Operation Team

### 6.1.3 Testing

Testing	
<i>initiation:</i> Application is ready to be tested.	<i>roles:</i> <b>Customer</b> cooperates
<i>action:</i> Tester and programmer are performing tests.	<b>Programmer</b> cooperates
<i>result:</i> Information how application has been tested.	<b>Tester</b> performs
<i>Integrated functions:</i> App SW Development	
<i>Follows:</i> Development	
<i>Is followed by:</i> Operation	
<i>Derived diagrams:</i>	

Figure 6.4: Testing Team

### 6.1.4 Transport Support Team

Transport Support Team	
<i>initiation:</i> Team needs support in an agile manner.	<i>roles:</i> <b>Customer</b> cooperates
<i>action:</i> All the agents in the team are supporting hand to hand.	<b>Programmer</b> performs
<i>result:</i> The customers are satisfied.	<b>Project Manager</b> is responsible

Figure 6.5: Transport Support Team

<i>Integrated functions:</i> Help Desk
<i>Follows:</i> Operation
<i>Derived diagrams:</i> Transport Support Team

Figure 6.6: Transport Support Team

## 6.2 Simulation Report

This report is all about the list of participants and how they are working in the coordination with accomplished tasks.

### Step: 1

Here customer is raising a ticket which needs to be resolved.

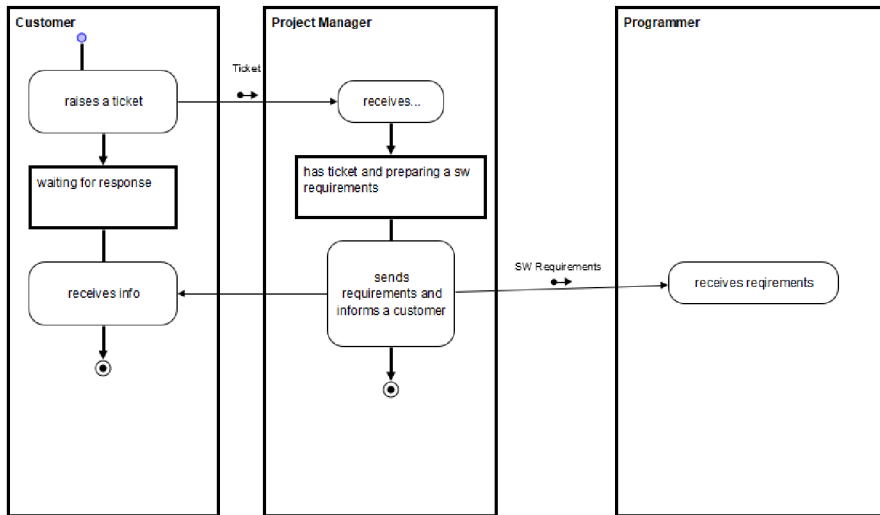


Figure 6.7: This is is a simulation part1

**Step: 2**

Here the customer reviews the ticket and pass it to the project manager.

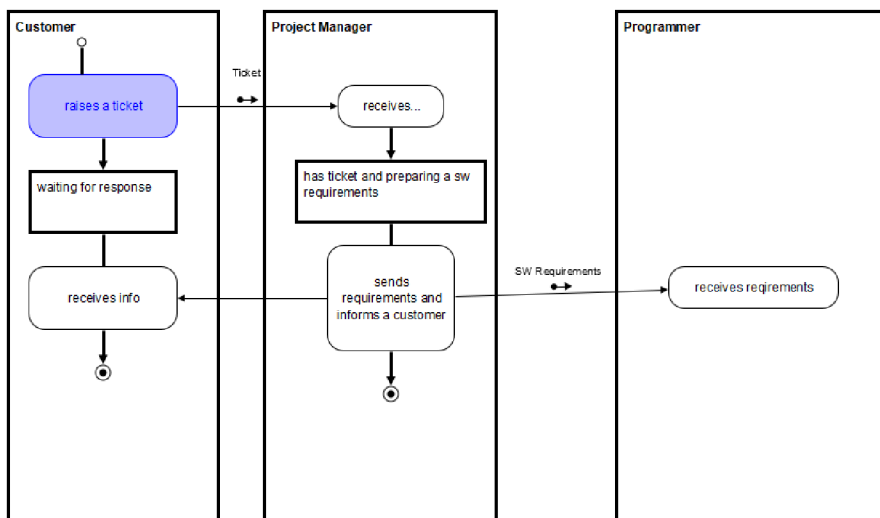


Figure 6.8: This is is a simulation part2

**Step: 3**

Here the project manager receives the ticket and reviews it as how it can be resolved.

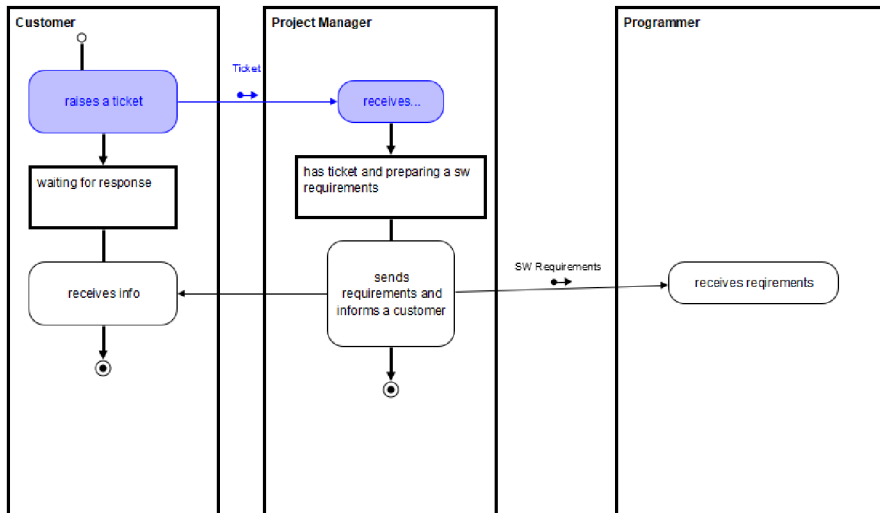


Figure 6.9: This is a simulation part3

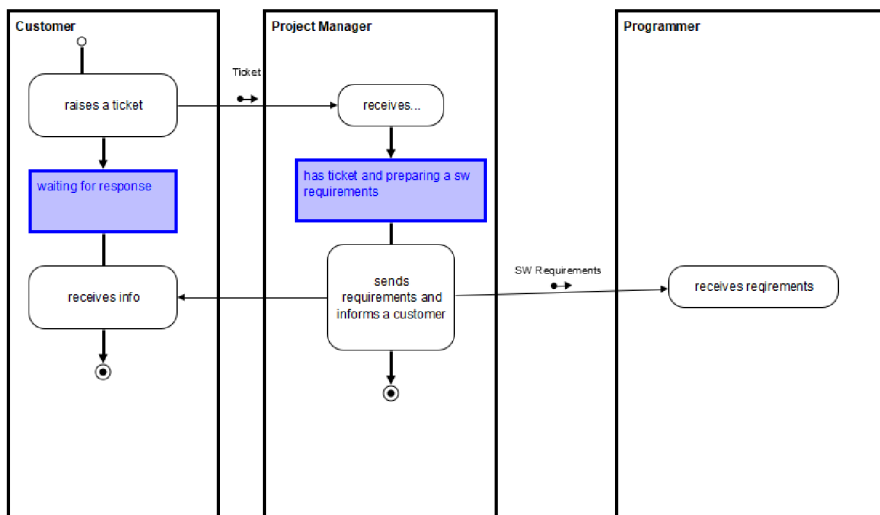


Figure 6.10: This is a simulation part4



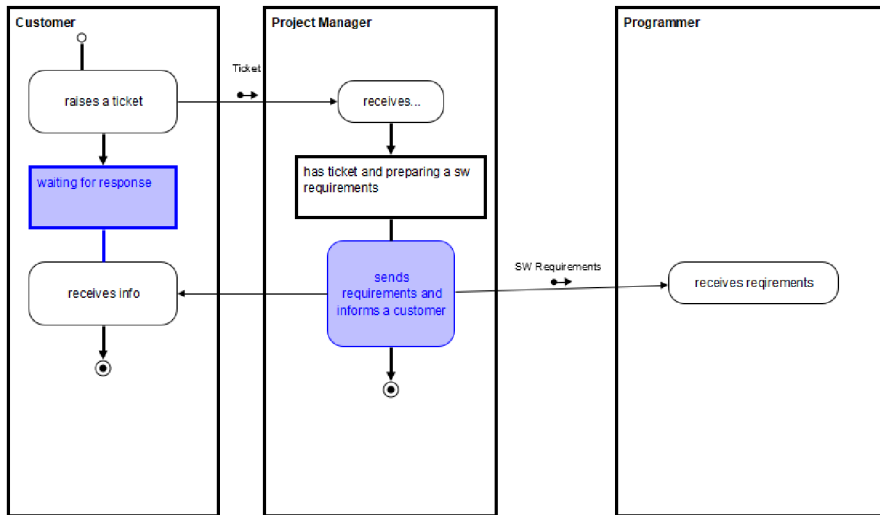


Figure 6.11: This is a simulation part5

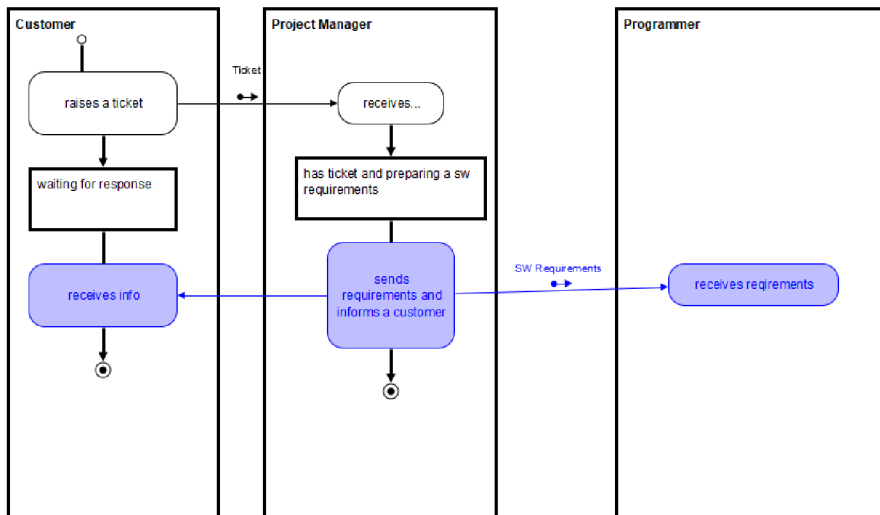


Figure 6.12: This is a simulation part6

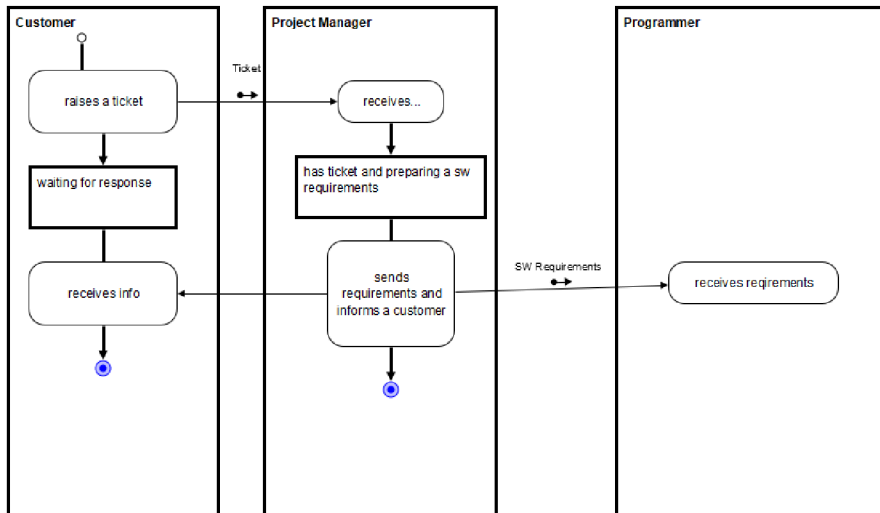


Figure 6.13: This is a simulation part7

## 6.3 Business Architectures

In a start up company there is a help desk management team working in co-ordination among themselves and supporting the requests of the customers by resolving there problems using various tools and technologies.

Start up SW company provides help desk support.

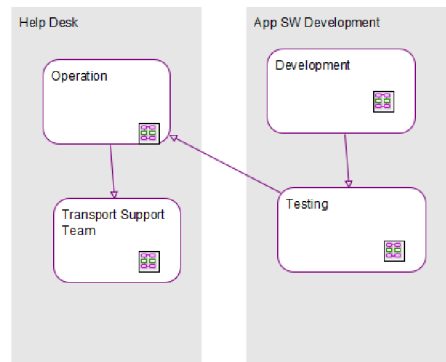


Figure 6.14: Help Desk Management System

### 6.3.1 Transport Support Team

Integrated scenarios: Transport Support Team

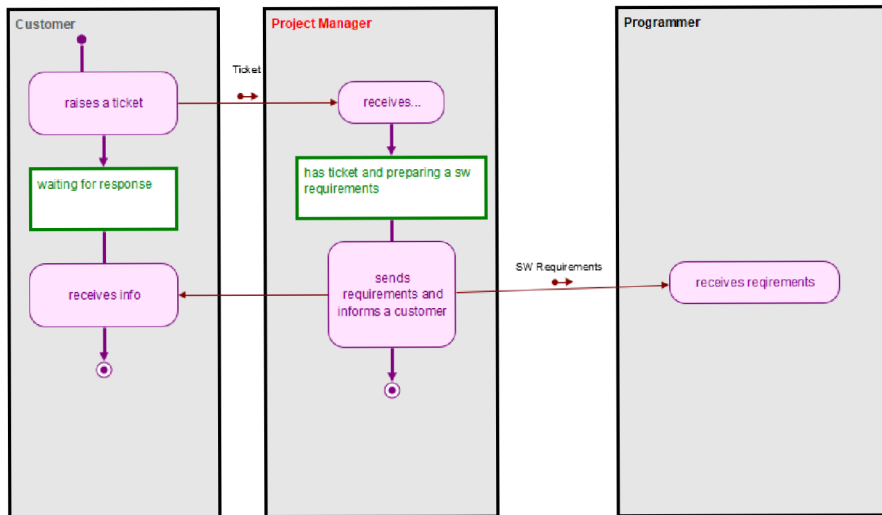


Figure 6.15: Transport Support Team

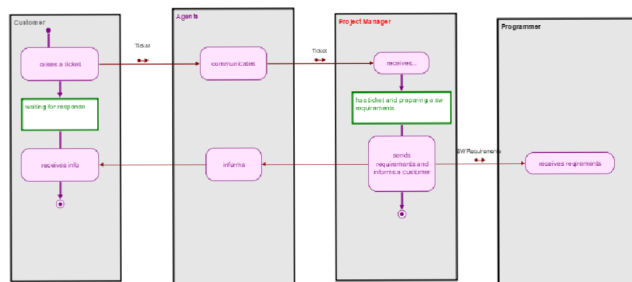


Figure 6.16: Transport Support Team

### 6.3.2 Participants

#### Business Analyst

The one who is responsible for all the business processes and activities in the organisation.

Customer

To whom the software is built.

HR

The team responsible for payroll processes, employee training and benefits.

Programmer

This is the developer of the software who develops software.

Project Manager

The team who takes control and care of company projects and manages them.

Tech support specialist

The team who provides technical support to the company such as help desk and various support activities.

Tester

It mainly focuses on testing the programmers output and result.

### 6.3.3 Basic modelling cards

#### Customer

Collaborators:	Programmer	Project Manager	Tester
Testing (cooperates)	cooperates		performs
Transport Support Team (cooperates)	performs	is responsible	

Figure 6.17: This is a report image part1

#### Programmer

Collaborators:	Customer	Project Manager	Tester
Testing (cooperates)	cooperates		performs
Transport Support Team (performs)	cooperates	is responsible	

Figure 6.18: This is a report image part2

#### Project Manager

Collaborators:	Customer	Programmer
Transport Support Team (is responsible)	cooperates	performs

Figure 6.19: This is a report image part3

#### Tech Support Specialist

Collaborators:
Operation (cooperates)

Figure 6.20: This is a report image part4

## Tester

Collaborators:	Customer	Programmer
Development (performs)		
Testing (performs)	cooperates	cooperates

Figure 6.21: This is a report image part5

### 6.3.4 Detail modelling cards

## Customer

Collaborators in diagram with name 'Transport Support Team':	Programmer	Project Manager
start: raises a ticket		>>
waiting for response: receives info		<<

Figure 6.22: This is a report image part1

## Programmer

Collaborators in diagram with name 'Transport Support Team':	Customer	Project Manager
receives requirements		<<

Figure 6.23: This is a report image part2

## Project Manager

Collaborators in diagram with name 'Transport Support Team':	Customer	Programmer
receives a ticket	<<	
has ticket and preparing a sw requirements: sends requirements and informs a customer	>>	>>

Figure 6.24: This is a report image part3

### 6.3.5 Data Flow

## SW Requirements

Business process re-engineering is a fundamental rethinking and redesigning of business process to achieve improvement in critical measures of performance such as cost, qual-

ity, service and speed. The main areas of improvement in processes are effectiveness, efficiency, internal control and compliance of various policies.

'SW Requirements' from 'Project Manager'	Programmer
sends requirements and informs a customer	receives requirements

Figure 6.25: This is a report image part1

## Ticket

Tickets has been raised regarding the information of delivery of goods and if the transportation has not reached on time, delays in unloading and loading has been recognised and taken into consideration of the agents who are taking it.

'Ticket' from 'Customer'	Project Manager
raises a ticket	receives a ticket

Figure 6.26: This is a report image part2

### 6.3.6 Development Team

In the development team there are 3 participants project manager, programmer and tester working hand to hand and coordinating the workflow. Where the project manager is sending the request to programmer, programmer receives the request analyze it and build codes and hence develops software. At the end the programmer sends the software to testers for testing.

Hence, this results in development of software in an agile manner and the models has been simulated with appropriate steps.

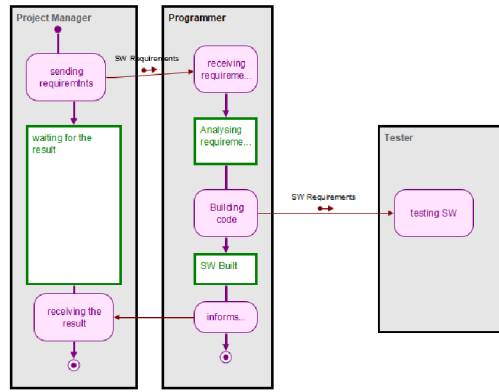


Figure 6.27: Development Team

### 6.3.7 Development Simulation steps

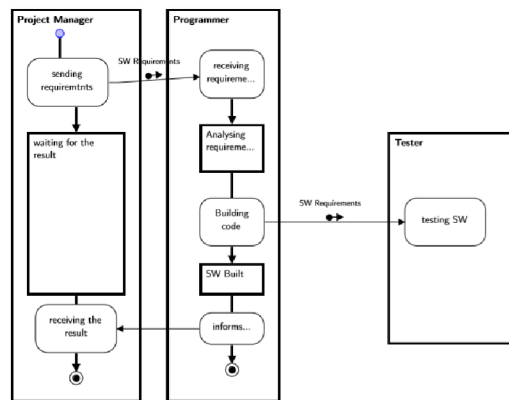


Figure 6.28: Development Simulation 1

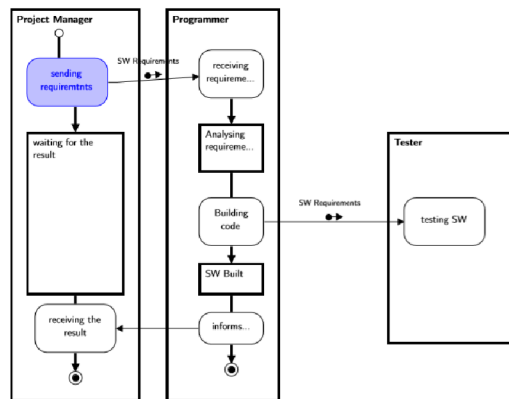


Figure 6.29: Development Simulation 2

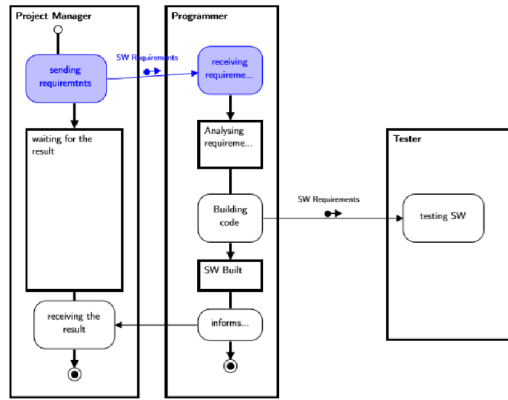


Figure 6.30: Development simulation 3

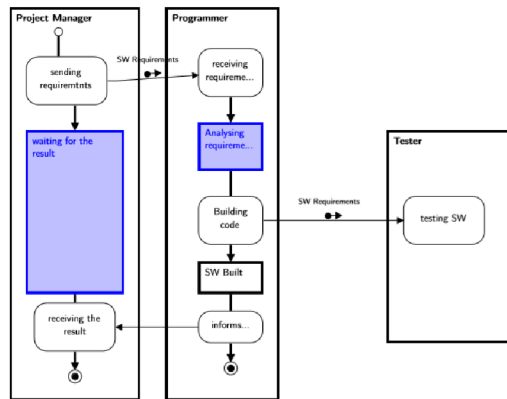


Figure 6.31: Development simulation 4

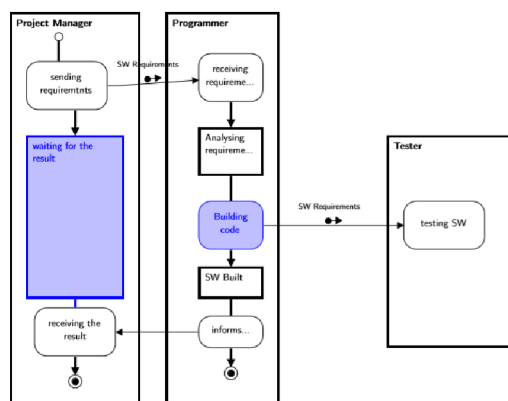


Figure 6.32: Development Simulation 5



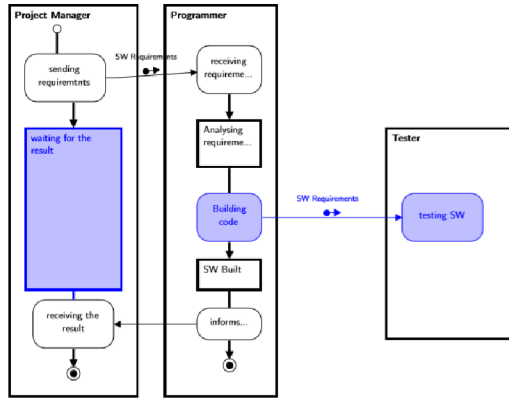


Figure 6.33: Development simulation 6

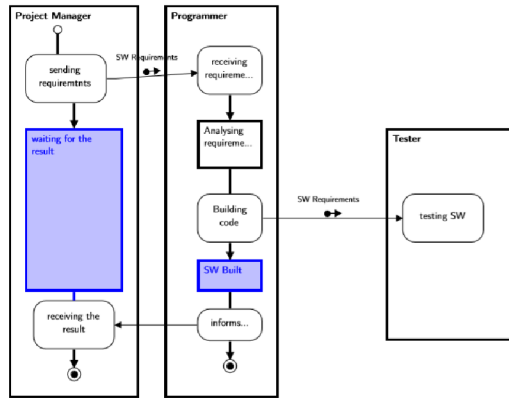


Figure 6.34: Development simulation 7

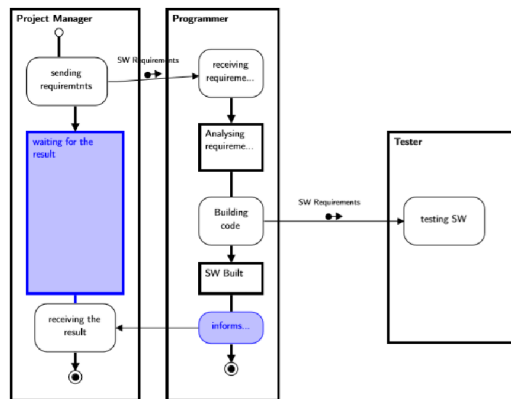


Figure 6.35: Development simulation 8

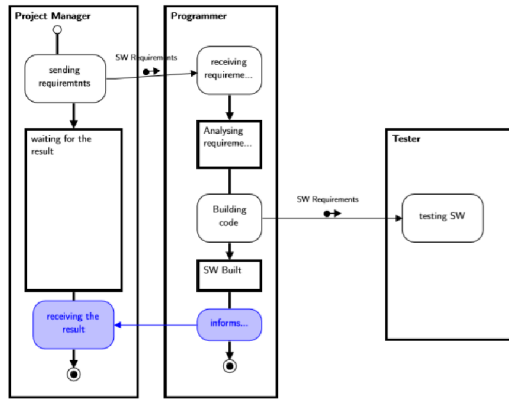


Figure 6.36: Development simulation 9

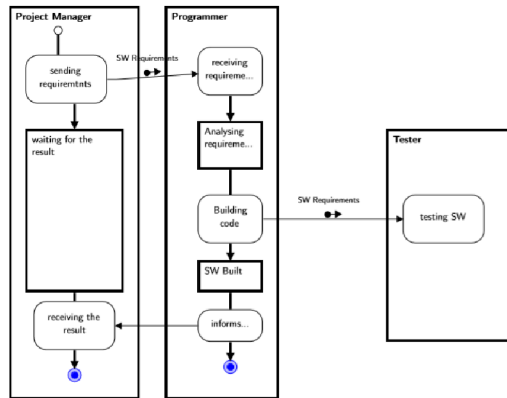


Figure 6.37: Development simulation 10

## 7 Conclusion

In my thesis i have designed the main organisation structure agenda of a small start up software company FxNet which has 5 teams and is providing database services and agile methodologies. Expected output is the model of to be processes, patterns and identification of process participant like employee roles, organisation units etc. The entire thesis is divided into two parts Theoretical and practical where theoretical part specifies various aspects of business processes and practical models to be used. The methodology focusing on constant change are agile methodology where the researchers are showing keen interest in the utilization of agile strategies and project development. In UML, numerous diagrams are used to describe the working structure of my company which makes the processes little time consuming. BORM also uses the UML concepts of communicating objects as interconnected finite-state machines efficient to model user stories. The participants are Project Manager, Programmer and Tester who are coordinating with each other in the development team. On the other side, in the transport support team the participants are customer, project manager and programmer who are providing customer service to there customers. This is our agile solution to fill the gap between requirements description and the software system features.

# Bibliography

Scott W Ambler. *Process patterns: building large-scale systems using object technology*. Cambridge university press, 1998.

Scott W Ambler. *More process patterns: delivering large-scale systems using object technology*. Cambridge University Press, 1999.

Joe Celko. *Joe Celko's complete guide to NoSQL: What every SQL professional needs to know about non-relational databases*. Newnes, 2013.

Petrus Du Toit. *An evaluation of non-relational database management systems as suitable storage for user generated text-based content in a distributed environment*. PhD thesis, University of South Africa, 2014.

Robert C Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall, 2002.

Alan Moran. *MANAGING AGILE*. Springer, 2016.

Tridibesh Satpathy. *A guide to the scrum body of knowledge*. scrumstudy, 2013. isbn: 978-0-9899252-0-4, 2013.

Andrew Stellman and Jennifer Greene. *Learning agile: Understanding scrum, XP, lean, and kanban*. " O'Reilly Media, Inc.", 2014.

Sai Sumathi and S Esakkirajan. *Fundamentals of relational database management systems*, volume 47. Springer, 2007.

Ελένη Χρυσομάλλη. *Agile methodologies & Scrum: When is a team ready to start using it?* PhD thesis, <https://nemertes.lis.upatras.gr/jspui/handle/10889/11404>, 2018.