

# A Comparison between Agile and Traditional Software Development Methodologies

A.K.M Zahidul Islam<sup>1</sup> and Dr. Alex Ferworn<sup>2</sup>

<sup>1</sup> Ryerson University, Toronto, Canada

*Received: 10 December 2019 Accepted: 31 December 2019 Published: 15 January 2020*

---

## Abstract

Agile and Traditional software development methodologies, both are being used in different projects of software development industry. Agile software development technology is an incremental software development process. On the other hand, Traditional software development methodologies or plan-driven software can be explained as a more formal approach to software development. These methodologies come with a fully completed set of systems requirements followed by an architectural and high level design development and inspiration. This research focuses on the software development life cycle, role and responsibilities of agile and traditional software development methodologies and their technical practices. It performs a comparison between both the software development methodologies. Here a questionnaire is used to collect data from the various experts of different IT related organizations of Bangladesh. In the questionnaire, there are three sections to bring out the individual knowledge from different organization, methodology knowledge of the respondents and software development experience of the respondents. The respondents are mainly software engineer, system analyst, software developer etc. A comparison is also performed between this survey result and a survey done by Ambler.

---

## *Index terms—*

On the other hand, Traditional software development methodologies or plan-driven software can be explained as a more formal approach to software development. These methodologies come with a fully completed set of systems requirements followed by an architectural and high level design development and inspiration. This research focuses on the software development life cycle, role and responsibilities of agile and traditional software development methodologies and their technical practices. It performs a comparison between both the software development methodologies. Here a questionnaire is used to collect data from the various experts of different IT related organizations of Bangladesh. In the questionnaire, there are three sections to bring out the individual knowledge from different organization, methodology knowledge of the respondents and software development experience of the respondents. The respondents are mainly software engineer, system analyst, software developer etc. A comparison is also performed between this survey result and a survey done by Ambler.

The analysis demonstrates the effect on software quality and cost from agile methodology and compares it with ambler (2007) survey and tries to find out correlation between the cost and quality of both the surveys. According to the respondents of the survey (Questionnaire) it is clear that what are the facilities and drawbacks of the traditional and agile software development methodologies for different size of the projects of an organization. At the end of the analysis part of this research it shows that for small scale projects more than 90% respondent response for agile methodologies and less than 10% responds for the mix software development technologies which are specific for a organization. For medium scales projects about 50% responds for agile software developments methodologies, more than 40% responds for the traditional software development methodologies and less than 10% responds for the other mix technologies for an organization. For the large scale project less than 10%

44 responds for agile methodologies, more than 80% responds for traditional methodologies and slightly more than  
45 10% responds for the other mix technologies for a specific organization.

46 The findings of this project research study also confirm the appropriateness of the use of agile methodologies for  
47 small scale projects, traditional and agile methodologies for medium scale projects and traditional methodologies  
48 for large scale projects of an organization.

### 49 1 Introduction

50 he software development industry is one of the fastest growing industries in the world. By analyzing previous  
51 20 years history of software development it is evident that a lot of brilliant ideas and methods born repeatedly.  
52 However, there was no guarantee whether those methods will last long or not though there are a good number  
53 of examples to prove this.

54 The concept of "Agile" is new. When it was introduced there was no agreement or explanation on what  
55 precisely it refers to. Despite this doubt agile methods became very popular among the industry within a very  
56 limited period. Agile was born after introducing extreme programming also known as XP. There are different  
57 methodologies comes under agile such as Dynamic Systems Development Method, FDD, TDD, SCRUM and etc.

58 "Agile" has the high reputation and interest in the industry but still there is no clear agreement on how  
59 "Agile" can be distinguish from more "Plan-driven" methods which are also known as the traditional methods.  
60 So it cannot identify any boundaries or limitations ??Boehm and Turner).There is no any systematic check on  
61 agile methods. However, there are some studies to identify the suitability of agile methods for different software  
62 project natures. Due to that there are no current events or guidelines for practitioners to select the best method  
63 to bring the maximum benefits to their projects.

64 "Agile" is becoming more renowned in the software industry. Agile methods are overtaking tradition methods  
65 in projects where requirements are changing frequently. In agile software development there is a series of software  
66 behaviors which is conventional as well as controversial. As a result, in the near future the software development  
67 industry will find ways to carefully use either the traditional or the agile methods or a hybrid of these two  
68 methods.

69 To get highest result and to achieve the goals, a software development team needs to understand and select  
70 the most suitable methodologies and techniques for their project. When acquiring the understanding that they  
71 can find answers to these questions:

72 "What natures of project they have in hand the possibility of changes while the project in progress?" "What is  
73 an appropriate balance of effort between documenting the work and getting the product implemented?" (Lindvall  
74 et al., 2002) "When does it pay to spend major effort on planning in advance and avoid change, and when is it  
75 more beneficial to plan less rigorously and embrace change?" (Lindvall et al., 2002) In order to answer properly  
76 to above questions and to make the correct decision proper knowledge should be implemented and should be  
77 disseminated within the industry. This research aims to develop a set of guidelines to help an organization in  
78 their decision making, when selecting the best software development methodology to a given nature of a project  
79 or projects, by doing a review on the different traditional and agile methods.

### 80 2 a) Aims of the Research

81 The aims of the research project are:

82 1. Review a number of different software development methods, both traditional and Agile. 2. "Can agile  
83 methods be used in any type of software development project?" find out the answer of this question. 3. Come up  
84 with a set of guidelines for a software organization to select the most suitable software development methodology  
85 for their software projects.

### 86 3 b) Objectives of the Research

87 The objectives are:

88 1. Carry out a literature survey on different software development methodologies. 2. Understand the lifecycles,  
89 roles and practices of these development methodologies. 3. A comparison for agile and traditional development  
90 methodologies to understand the similarities and differences. 4. Carry out a survey in the software industry  
91 with practitioners and professional in software engineering. 5. Analyze the gathered data from the survey and  
92 summaries them to fulfill the final aim with the help of the knowledge from the literature.

### 93 4 c) Research Question

94 What are the significant factors for a project to consider the most appropriate type of process model, after  
95 comparing agile and traditional software development methods?

### 96 5 d) Structure of this Research

97 The first chapter introduces what is the aim and objective of this research and what is the research question  
98 of this research. The second chapter introduces the literature review of this research to answer the research  
99 question. The third chapter introduces the research design and makes a questioner for the target audience of this

---

100 research. After a survey from the audience the result of this research is discussed in chapter four. Basically this  
101 questioner helps to collect data for this research. Chapter five analyzes the research result and tries to bring out  
102 proper methodology for specific software. The final chapter tries to bring out limitation of this study and future  
103 aspect of this research.

## 104 6 II.

### 105 7 Literature Review a) Outline

106 The Manifesto for Agile Software Development (MAD) was published in 2001 by a group of seventeen  
107 methodologists. This group of experts agreed on a common set of guiding principles and practices around effective  
108 software development. The focus was for modeling and documentation of software development projects. The  
109 methodologists introduce the guidelines which are: (Fowler and Highsmith, 2001) ? Individuals and interactions  
110 over processes and tools

111 The main concern in this section is the relationship and communication between the software developers and  
112 any other persons involve in the software development process. The dependency on just tools and processes will  
113 be minimal.

114 ? Working software over comprehensive documentation

115 The main purpose here is to keep the documentation as small as possible and thus concentrating more on  
116 building and delivering tested and quality products. Different teams can handle the deliveries differently. Some  
117 may deliver hourly or per week while others releases product every two weeks or once a month.

### 118 8 ? Customer collaboration over contract negotiation

119 The main concern of this section is the relationship between the development team and the client. The relationship  
120 has to be very high. However, the importance of having a contract and changing it accordingly is important as  
121 well. Agile starts to release functional program modules as soon as the development process starts and thus it  
122 effectively minimizes the risk and disappointment of not meeting the actual requirement at the far end of the  
123 project.

124 ? Responding to change over following a plan The people who are involved in the software development like  
125 programmers, clients and any other should be well knowledgeable about the progress and any changes. Any party  
126 have the authority to consider possible changes to the product When it is been developed.

127 The founders of MAD say "while we value the items on the right, we value the items on the left more"  
128 (Fowler and Highsmith, 2001), so there are different debates on these values. There are other practitioners  
129 including Steven Rakinin (2001) who thinks that items on the left are just an excuse for hackers with no regard  
130 for engineering discipline.

131 Traditional software development methodologies or plan-driven software can be explained as a more formal  
132 approach to software development. These methodologies come with a fully completed set of systems requirements  
133 followed by an architectural and high level-design development and inspiration. However, during mid 1990's some  
134 practitioners found some steps such as full documentation frustrating and unnecessary time wasting (Highsmith,  
135 2002). Due to these heavy aspects, this methodology is known as heavyweight development methods.

136 Traditional development methodologies all include with the following ??Williams & Heckman, 2008):

137 ? Repeatability and predictability ? A defined incremental process The Personal Software Process (PSP),  
138 Team Software Process (TSP), and Rational Unified Process (RUP) are the three of the most popular and widely  
139 used plan-driven methodologies. Among these plan driven methodologies waterfall model and spiral model are  
140 well-known.

141 According to Davis and Sitaram (1994) waterfall model have the ability to capture the gross state of the project.  
142 Using this model therefore a project manager can track the progress through all major phases of development of  
143 major intermediate products. On the other hand spiral model captures the iterative nature of software versions  
144 and helps the project manager to isolate the key decision points to select a development strategy. They further  
145 argue "Neither of these two models, nor any other published model, provides a project manager with a picture  
146 of the true state of the project. Project managers who track project status in terms of the major phases have no  
147 idea of the status of their projects."

148 The following table which was published by Abrahamsson et al., ??2002) demonstrates the differences of  
149 privileged and marginalized methodological information systems development process. These were a collection  
150 of views from different authors in the field. The marginalized methods have much more things in familiar with  
151 the original agile development methods. The privileged method projects use more of a process oriented software  
152 development methods. These methods also called plan-driven methods.

153 McCauley ??2001) argues that the underlying philosophy of Traditional methods which is referred to as  
154 process-oriented methods in the article, is that the functional requirements of a project is utterly frozen or in  
155 other words sealed before move in to the next phases such as the design and development. The article also states  
156 that this approach is not feasible for most of the software projects. So the need of a flexible and agile development  
157 methods is necessary for developers to make changes or amendments to the specifications while it is been built.  
158 Further according to McCauley (2001) there is no software model that suits any nature of software project. It is

159 the project management who should be able to select the best suitable methodology according to the project in  
160 hand. There are different other experts in the field who support this argument.

161 The origin of Agile methods go back a long way even though they were properly introduced and started to gain  
162 interest in the software industry during the last few years. As mentioned earlier, as a result of built up frustration  
163 within the software developers on structured and planed methods in the mid-1990s, development teams started to  
164 use early versions of some of the agile methodologies such as Extreme Programming (XP), SCRUM and Dynamic  
165 Systems Development Method (DSDM).

166 The Agile methodologies describe a number of principles which in summary put the human factor (customers  
167 and developers) first over processes and plans. The highest priority principle is to satisfy the customer through  
168 early and continuous delivery of software. According to Miller (2001) there are a number of characteristics of  
169 agile methods from a fast delivery view, which ultimately shortens the software project lifecycle:

170 1. Modularity -This is on the process level of development 2. Iterative -Consider short development cycles  
171 which enables to clear error faster and more accurate 3. Time bound -iterative cycles ranging from one to six  
172 weeks 4. Parsimony -remove all the unnecessary activities in the development 5. Adaptive -Take faster action  
173 against possible new emerging risks 6. Incremental -A functioning application software, build up in smaller steps  
174 7. Convergent -Minimizes risks 8. People-oriented -Agile favour people who are involved over the process and  
175 technology 9. Collaborative -Active communication.

176 In Barry Boehm's IEEE computer article (2002) it is mentioned that according to there are several critical  
177 people-factors which agile highlights, such as amicability, talent, skill, and communication. further describes,  
178 what is new in agile is not the behaviors or practices they use but the recognition of users or any other people  
179 involved as the primary sources which drive the project to a success.

180 Agile does not require highly-capable people to execute its practices in a software project environment.  
181 However, it requires tacit knowledge and lot of expertise to function successfully. Due to this reason agile  
182 has a minimum use of fully completed documents. Boehm warned that there is a possible risk that this situation  
183 may lead to architectural mistakes, which are hard to find and correct by any external party.

## 184 9 c) Definition of Agile

185 Agile cannot be given with a constant definition. Different practitioners have different wordings according to  
186 their experience and understanding. But agile can be explained in few characteristics that are considered as the  
187 core characteristics.

188 ? Iterative and incremental process ? Simple and easily adoptable ? Collaboration of all the parties such as  
189 users, customers, developers, project managers, etc. ? Produce high quality software within the requirements,  
190 budgets and the time scale.

191 Following are different definitions from different expert practitioners.

192 "Agile is an iterative and incremental (evolutionary) approach to software development which is performed in  
193 a highly collaborative manner by selforganizing teams with 'just enough' ceremony that produces high quality  
194 software in a cost effective and timely manner which meets the changing needs of its stakeholders." ??Amblor,  
195 2001) "Agile is a conceptual framework generally centred on iterative and incremental delivery of working software,  
196 driven by the customer. The iterative part suggests that we are repeating, or iterating, a complete lifecycle of  
197 development over a short, fixed span of time. With each of these iterations, we ship some working subset, or  
198 increment, of features." (Langr, 2006)

## 199 10 d) SDLC for Agile

200 According to Amblor agile SDLC composed of four phases Iteration0, Development, Release and production.

201 Iteration 0:

- 202 1. Initial time of the agile project.
- 203 2. Modeling and initial architecture of the project. 3. Setting up the environment of the project.

## 204 11 Development Phase:

205 Incrementally deliver high quality software which meets the changing needs of the use.

## 206 12 Release Phase:

207 In this phase agile practitioners transition the system into production.

## 208 13 Production Phase:

209 The fundamental goal of this phase is to keep the system running and help users to use the software. Year 2020

## 210 14 ( ) C

211 © 2020 Global Journals

---

## 212 15 e) Agile Methodologies

213 Agile manifesto provides an ideological environment to modern so called "agile" software development with its  
214 defined values and principles. A survey conducted by Cutter Consortium with regard to the methods been used  
215 in the software development field revealed that 54% of the users use their own inhouse development methods,  
216 which can be explained within the agile boundaries. Among the defined methodologies in agile the most popular  
217 methods were Extreme programming, Feature Driven Development and Adaptive Software Development. The  
218 purpose of this section is to introduce few of the widely used agile methodologies identifying the roles, process,  
219 responsibilities and practices. The following methods will be included for discussion: Extreme Programming  
220 (XP), Dynamic Systems Development Method (DSDM) and SCRUM.

221 i. Extreme Programming a. Outline Extreme programming (XP) evolved from the frustrations and the  
222 problems caused from traditional plan-driven methods, which were the only development solutions in the software  
223 industry for a long time (Beck, 1999). XP was developed and brought in to practice in the mid 1990's by Kent  
224 Beck, Ward Cunningham and Ron Jeffries (Paulk, 2001) as a result of a project they been working. The main  
225 features which XP emphasizes are those that they identify as the prerequisite for effective software development  
226 which are improving communication, getting feedback, simplicity and proceeding with courage (Cockburn, 2001).  
227 Even though these practices started as just a better ways of development rather than traditional methods with  
228 time they showed success. This was the root for XP. XP has widely influence on the principles in the agile  
229 manifesto (Kalermo & Rissanen, 2002).

230 There are different theories and arguments about XP whether it is actually a method and how extreme this  
231 methods is. Paulk (2001) argues that these practices are actually just commonsense practices that any discipline  
232 method would have and not something extreme. Beck (1999) who is one of the founders of XP states that XP is a  
233 fresh and new methodology and the term "Extreme" comes from taking these commonsense practices to extreme  
234 levels.

235 XP is based on the following five important values.

236 ? Communication -"Problems with projects can invariably be traced back to somebody not talking to somebody  
237 else about something important." (Beck, 2000). XP focuses lot on face to face or oral communication and its  
238 techniques encourages in maximizing interaction. This is valued on the observation that most project difficulties  
239 occur because individuals or teams have not spoken with other parties to clarify questions, to collaborate, or  
240 to obtain help. ? Simplicity -Rather than try to capture all features and complicate, Design the project in  
241 the simplest way to meets the customer's needs. The value highly stresses on the point, only design and code  
242 the current requirements obtained rather than to anticipate and plan for unstated requirements. ? Feedback  
243 -The development team(s) obtain feedback from the customers at the end of each iteration and release. The  
244 next iteration drives with the consideration of this feedback. There are very short design and implementation  
245 feedback loops built into the methodology via pair programming and test-driven development (Williams, 2003).  
246 ? Courage -The best thing about XP is that the other three values give the team to have courage in their actions  
247 and decision making. The team decides which parts will be done at which stages. Further, this encourages the  
248 team to avoid any pressure for unrealistic deadlines or requirements. ? Exploration phase -Story cards are used  
249 by customers to express the features they want in the system. In each story card they have to write a feature  
250 they wish to have in the system. Mean while the technical teams focus on the tools and technologies they are  
251 going to use in the project. They get familiar with those tools as well. They test the technologies and the  
252 proposed architecture possibilities by building a prototype of the system. Depending on the project scope and  
253 the teams' familiarity with the technologies this phase spans from few weeks to few months. ? Planning phase  
254 -Considering all the stories, prioritize the features to be delivered in the first set of the release of the system. The  
255 development teams estimate the time required for different features and then agree upon the deliveries for the  
256 first release. The first release of the system can take up to two months and the planning phase may take few days.  
257 ? Iteration to release phase -The schedule set up for the first release is divided into small iterations before the  
258 actual first release. The first iteration builds system architecture for the whole system by selecting and analyzing  
259 the stories which includes the features. The customers decide which story to include in each of the iterations.  
260 Further the customers can create functional test for the system. These will be used to check the accuracy of the  
261 system and may use in the future. Iteration is around one to four weeks each for implement. Once the iterations  
262 are done the system is ready for production. ? Product ionized phase -This phase runs faster than the others,  
263 which means that the iterations can be reduced to one week instead of three weeks. The system has to be extra  
264 tested for performance before release to the customers. New changes found here has to be decided before start  
265 working on them. Postponed ideas will be documented to build later. ? Maintenance phase -After the product is  
266 product ionized and released for customer use, teams have to make sure that system in the production running  
267 and also produces new iterations. This phase need an effort for customer support tasks In order to maintain  
268 these operations. Thus, the maintenance phase may require new people into the team and also changes in the  
269 development structure. ? Death phase -The project comes to this phase when there are no more requirements  
270 from the customers. But there are other concerns such as reliability and performance before reaching this point.  
271 Since there are no more requirements to be added to the system all the documents been written at this stage.  
272 On the other hand when the project does meet the requirements and it is expensive for further development, it  
273 can reach death phase.

## 16 c. Responsibilities and Roles of XP

There are specific roles in XP for different tasks. This makes work much easier to handle as they are divided with clear roles. The following describes these roles according to Beck (2000) and Brahmsson et al. (2002).  
? Manager -Makes all the decisions and is responsible for the team and its issues. He or she has the right to form the team, obtain and allocate resources, manage people and problems. In order to do all above, he or she communicates with the team to understand the present situation. The manager interfaces with external groups as well including the customers. Selects stories for a release and for an iteration.

One individual from the customer organization or a group of customers can be involved in the sections, or a customer representative can be chosen from within the development organization that is external to the development team.

## 17 d. Technical Practices

The initial version of XP had defined programmer-centric technical practices. This was published in 2000 by Beck.

? Planning game? Short releases ? Metaphor ? Simple design ? Testing ? Refactoring ? Pair programming ? Collective ownership ? Continues integration ? 40 hour week ? On-site customer ? Coding standards ? Open workspace ? Just rules

XP practices were changed to include 13 primary practices and 11 corollary practices in 2005 (Beck, 2005). The primary practices are intended to be useful independent of each other and the other practices used, though the interactions between the practices may amplify their effect (Williams, 2007).

ii. SCRUM a. Outline "The relay race approach to product development may conflict with the goals of maximum speed and flexibility. Instead, a holistic or 'rugby' approach -where a team tries to go the distance as a unit, passing the ball back and forth -may better serve today's competitive requirements." (Takeuchi and Nonaka, 1986) SCRUM is also a member from the agile development processes family. Scrum is a process skeleton that includes a set of practices and predefined roles. It provides you a set of guidelines to develop software from its design stage to its completion. Scrum is best suited for the projects with rapidly changing or highly emergent requirements. It is a Simple and scalable method which means easily combined with other methods and doesn't prescribe engineering practices. According to the article on scrum by Clifton and Dunlap (2003b) there are few software development issues scrum addresses for a better software production.

? Chaos due to changing requirements -The real or perceived requirements of a project usually change drastically from the time the product is designed to when it is released. Under most product development methods, all design is done at the beginning of the project, and then no changes are allowed for or made when the requirements change. ? Unrealistic estimates of time, cost, and quality of the product -The project management and the developers tend to underestimate how much time and resources a project will take, and how much functionality can be produced within those constraints. In actuality, this usually cannot be accurately predicted at the beginning of the development cycle. ? Developers are forced to lie about how the project is progressing -When management underestimates the time and cost needed to reach a certain level of quality, the developers must either lie about how much progress has been made on the product, or face the indignation of the management.

## 18 b. SCRUM Lifecycle

Scrum has a process which has to be followed by any organization or team that adopt this methodology. As figure 2 illustrates the projects Year 2020

## 19 ( ) C

© 2020 Global Journals development happens via a series of month-long iterations called Sprints. Scrum is ideally suited for projects with frequently changing or highly emergent requirements. The Product Backlog lists the work to be done on a Scrum project. It lists all desired changes to the product. A Sprint Planning Meeting is held at the start of each sprint during which the Product Owner prioritizes the Product Backlog and the Scrum Team selects the tasks they can complete during the coming Sprint. These tasks are then moved from the Product Backlog to the Sprint Backlog.

## 20 c. Responsibilities and Roles of SCRUM

Scrum implements its iterative and incremental process through three roles. All management responsibilities are divided between these three roles (Schwaber, 2007).

? Product Owner -The product owner is responsible for the project, managing, controlling and creating and prioritizing the Product Backlog. He or she is selected from the other parties such as management, customers and the scrum master. The Dynamic Systems Development Method (DSDM) was first developed in the United Kingdom around the mid to late 1990s by a group of people from a business background. It was totally not related with technical perspective. This can be said as one of the heavier Agile approaches available (Coffin and Lane, 2007). It was initially developed as an addition to Rapid Application Development (RAD), incorporating best practices from the business-oriented environments.

---

332 DSDM is a well ordered, commonsensical process focused on delivering business solutions quickly and efficiently.  
333 It has similarities to SCRUM and XP in many ways, but it has its best uses where the time requirement is fixed  
334 (CliftonandDunlap, 2003a). DSDM focuses on delivery of the business solution, rather than just team activity.  
335 It ensures the feasibility and business sense of a project before it is created. The cooperation and collaboration  
336 between all interested parties is an important fact in DSDM. This method makes heavy use of prototyping to  
337 ensure all the involved parties have a clear picture of all aspects of the system.

338 Unlike in traditional development methodologies where functionality is fixed, and time and resources are  
339 variable, in DSDM, time is fixed, and functionality is variable (CliftonandDunlap, 2003a). The following figure  
340 best illustrates this scenario. ? Ambassador -The person who acts as intermediate between the users and the  
341 development team. He manages the development team, and usually has a good overall understanding of how the  
342 system will work.

343 ? Visionary -This role is the driving force behind the project. This role keeps the project steered on course  
344 towards the business goals. Often is the person who started/thought of the project. ? Advisers -People who  
345 have practical knowledge in areas of the business that need to be automated, and/or in the technologies needed  
346 to automate these areas.

## 347 21 d. Technical Practices

348 There are nine principles at the core of the DSDM methodology. Some clearly overlap with XP and similar  
349 approaches. However, DSDM's principles are sufficiently robust to minimize damage to schedules and resources  
350 when a business process radically changes or a major component's design is faulty-problems that could cripple  
351 an XP project (Robinson, 2002).

352 ? Active user involvement is a must.

353 ? Design groups are empowered to make system development decisions. ? Frequent and regular delivery of  
354 components is a priority. ? The primary acceptance criterion for a system or component is its fitness for business  
355 purposes-the design driver is business benefit. ? The business solution is the goal, and iterative and incremental  
356 development is necessary to converge on that solution. © 2020 Global Journals methods used in the engineering  
357 world in the software development world.” ??Georgiadou, 2001).

358 Traditional software development methodologies are the first methods of software development. They are also  
359 known as heavyweight methodologies. They are considered to be the classic way of developing software. These  
360 methodologies are mostly based on a series of sequential steps, such as requirements definition, solution building,  
361 testing and deployment.

362 Traditional software development methodologies require defining and documenting a stable set of requirements  
363 at the beginning of a project.

364 ii. Waterfall Model a. Outline The Waterfall model is known as the classic model of software development.  
365 The Waterfall model also known as the "top down" approach, was proposed by ??oyce (1970). Until the mid  
366 80's it was the only model with a level of general acceptance. It was derived from models used in traditional  
367 engineering activities with the objective of establishing an order in the development of large software products.  
368 It is more rigid and less manageable compared with other software development models.

369 The Waterfall Model is one of the most important models ever published. It is a reference to others, and  
370 serves as the basis for many modern projects as well. Its original version was improved over time and is still  
371 frequently used today (Peters and Pedrycz, 2000). A great part of the success of the Waterfall Model is due to  
372 the baseline management, which identifies a fixed group of documents produced as a result of each phase of the  
373 life cycle (Peters and Pedrycz, 2000). The produced documentation includes more than text files, it has graphical  
374 representations of the software and even simulations.

## 375 22 b. Waterfall Model Life Cycle

376 Waterfall model phases are executed systematically in a sequential order. The model usually has the following  
377 phases: Analysis, Design, Implementation, Testing, Deployment and Maintenance. Requirement gathering and  
378 Analysis -This is the phase where all the requirements to be developed are captured. This is done by conducting  
379 consultations, interviews, observation and so on. A document called requirement specification is created including  
380 all the gathered requirements at the end of this phase (Parekh, 2005a).

381 System design -Looks at the overall system in a design and architectural level before starting actual coding.  
382 This is to get an idea how the system look like at the end of the project. All hardware, software and resource  
383 requirements are considered here and finally create the system design specification to start the next phase. Year  
384 2020( ) C © 2020 Global Journals

385 Implementation and unit testing -The actual coding begins in this phase. According to the system design spec  
386 system is built in small units. Each of these units are tested to ensure that it servers the purpose that unit is  
387 built (Parekh, 2005a).

388 Integration and system testing -In the previous phase the system is built in units. This phase focuses on  
389 getting these units together. The system is build by putting the units together. Units are tested with each other  
390 to ensure that they work and communicate with each other and give the final outputs which are expected from  
391 the whole system (Parekh, 2005a).

392 Operations and maintenance -This phase is normally considered the longest of all. Issues and errors of the  
393 system which were not found during the development stages come alive once the system starts to operate in a  
394 live environment. This will normally happen time to time. So this phase is called maintenance (Parekh, 2005a).

395 iii. Spiral Model a. Outline The spiral model was introduced by Barry Boehm in 1980s, based on experience  
396 with various refinements of the waterfall model as applied to large software projects. This method combines  
397 elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-  
398 up concepts (Boehm, 1988). There are four main phases of the spiral model (Boehm, 1988):

399 ? Objective setting -Specific objectives for the project phase are identified. ? Risk assessment and reduction -  
400 Key risks are identified, analyzed and to reduce these risks information is obtained. ? Development and Validation  
401 -For the next phase of development an appropriate model is chosen. ? Planning -For the next round of spiral the  
402 project is reviewed and plans are drawn up.

### 403 **23 b. Spiral Model Lifecycle**

404 As shown in figure 6 there are four main phases in spiral model. They are Planning, Evaluation, Risk Analysis  
405 and Engineering. These phases follow one after another in an iterative manner. The objective is to eliminate  
406 the problems occurred in the waterfall model. In an article by Parekh (2005b) mentions that even though the  
407 iterative approach became a solution to waterfall model issues, spiral model requires people with high skills in  
408 the area of planning, evaluation, risk and customer relations. The project becomes more costly than planned  
409 due to the demand for more than one iteration cycle. Following describes the main phase in spiral model. Year  
410 2020 Plan phase -This phase gather and finalize the objectives and constraints of the project and documented.  
411 These are kept locked in order to decide on the approaches and strategies of the project. Risk analysis -This is  
412 considered as the most important phase of the model. All the approaches and strategies are analyzed for risk  
413 factors. Prototyping is used to find solutions and to develop a low cost and quality system if there are any  
414 indications of risk. Engineering -This is the development phase. Development outputs are carried through all  
415 the phases iteratively for improvements. Customer evaluation -The built product is passed on to the customer in  
416 order to receive feedback. This phase is expected to come across possible errors and/ or changes. This is similar  
417 to system testing. iv. Unified Process a. Outline Unified process is actually not a process rather it can be called  
418 as an extensible process which can be customized according to the nature of different projects or organisations.  
419 Every approach such as modeling is organized into workflows in the Unified Process (UP). UP is performed in  
420 an iterative and incremental manner and some of the key features of the UP are described below (Booch, 1994):

421 ? UP consists with an architecture based on components which creates a system that is easily extensible,  
422 supports software reuse and intuitively understandable. The component commonly being used to coordinate  
423 object oriented programming projects. ? It uses modeling software such as UML to represent its code graphically  
424 as a diagrammatic notation to allow less technically capable individuals, but with a better understanding of the  
425 problem to come up with a greater input. ? The use of use-cases and scenarios to manage requirements seems to  
426 be very effective at both capturing functional requirements and help in keeping sight of the anticipated behaviors  
427 of the system. ? Since the design is done in an iterative and incremental manner it helps reduce project risk  
428 profile. Further it allows greater customer feedback and help developers stay focused. Year 2020

### 429 **24 ( ) C**

430 © 2020 Global Journals

431 ? Verifying software quality is very important in a software project. UP assists in planning quality control  
432 and assessment built into the entire process involving all member of the team.

433 b. UP Lifecycle The above diagram indicates the four phases in UP lifecycle. These four phases are described  
434 below (devdaily).

435 ? Inception -This phase creates a business case at the end of the process. The feasibility of the system is  
436 measured and the scope of the system is defined. ? Elaboration -The basic architecture of the system have  
437 been produced and a construction plan is agreed. Furthermore a risk analysis takes place and major risks are  
438 addressed. ? Construction -The system is produced and released for testing. This is not a full functioning system.  
439 A working system should be available and sufficient enough for testing under realistic conditions. ? Transition  
440 -The system is finally up to the standard to go in a live environment. So it is introduced to the stakeholders and  
441 intended users. Once the customers and the project team agreed that the intended target is met and the user is  
442 satisfied the system is completed.

443 There are approximately 50 work steps that has to be completed in UP during the process (Larman, 2004).  
444 All this documentation and this rigid approach add a lot of complexity to UP. UP has predefined roles to the  
445 project team making it less flexible in working.

### 446 **25 g) Comparison of Agile and Traditional Methods.**

447 In the previous section some discussions were there on both agile and traditional methods to identify the  
448 characteristics of these methods. It is important to do a comparison on these two methods in order to understand  
449 the differences that will affect different projects. Year 2020



---

## 26 ( ) C

© 2020 Global Journals It is mentioned in the early sections that traditional methodologies were ruling the software industry for a long time until practitioners begin to understand some of the drawbacks largely affecting the software projects. Extreme programming became popular in the industry when it was introduced in late 90's by Kent Beck. Then agile was introduced based on the concepts used in XP. Agile handle projects mostly in a volatile and uncertain environments. But with the passage of time practitioners came to realize that agile cannot handle all types of software projects as it has some limitations as well. Both of these methodologies have their strengths and weakness. Now the organizations tend to use the strengths of both together in their projects. There are three main factors which need to be considered when selecting a methodology. They are people, project size and risk.

### 27 i. People

This is one of the main important factors considered in software development. Especially agile methodologies strongly believe in human factor. Bohem and Turner (2003) believe that "In essence, software engineering is done 'of the people, by the people, and for the people.'" The agile manifesto stresses about the importance of the human interactions and customer collaboration in their basic values of agile methodologies (Fowler and Highsmith, 2001).

Developers and customers are the most important categories in people needed for software development. When using agile methodologies the people factors for developers were identified as skill, talent, communication and amicability (Bohem and Turner, 2003). Agile unlike traditional methodologies encourage working closely with the customers. This is important for a successful development environment.

The organization's culture has an impact on the people factor. If the developers are under the tight rules of the organization, it is hard to adopt agile since the developers will not get the maximum out of agile methodologies.

### 28 ii. Project Size

Project size of software is another major factor and considered as a challenging factor. In the early stages of project size estimation it was measured by predicting the number of lines of code the project may need (Dekkers, 2005). This is one of the limitations agile is facing today. For most of the large scale projects which involve more than 50 software developers agile seems to be working in a negative manner. This was shown in a study conducted by ambler (2008). Cockburn (2008) iii. Software Risk Software project risk may result in lots of problems. Budget and plan overruns and unable to meet the expectations of the uses and many more (Renhui and Fengyong, 2007). There are few categories of risk according to Renhui and Fengyong (2007), and there are; Team risk ? Environmental risk of organization ? Demand risk ? Plan and control risk ? User risk ? Complexity risk

An organization should be careful when handling these risks. Traditional methodologies are used for large critical systems with security and reliability such as military systems. However, for the systems that can be made quickly and have lots of uncertainty, Agile is the most appropriate methodology. For example a system expected lots of change of requirement during the development phase through customer involvement agile is the best methodology as it can respond to changes faster.

## 29 III.

### 30 Research Methods

#### 31 a) Introduction

This chapter discuss about the methodology used by the researcher to present a research into the statement of aim. The main purpose of this section is to evidently define the specific guidelines which will make possible the researcher to substantiate the achieved hypothesis. In brief, this section discusses about the ideas, which are used in the course of primary and secondary.

#### 32 b) Research Philosophy

Research philosophy depends on the way a researcher thinks about his/ her development of knowledge (Saunders et. al., 2003). The major research philosophy theories are Positive, Phenomenology and Realism (Maylor & Blackmon, 2005).

Positive or scientific method affirms that there is just one truth about the world. It is understood that such truth is objective and does not entail any value judgments. Finding this truth requires a process based on a deductive method for which data must be collected. In this sense positivist researchers stand that the data is not affected by the researcher opinion and that the more objective the data collection the better. ??McNeill, 1985) Usually the data is collected, interpreted and analyzed following the quantitative method and according to a statistical approach. Data collection might be achieved through surveys. The survey aim is to test the original hypothesis and therefore, to establish the truth of a specific phenomenon. The relevance of this kind of approach resides in its objectivity, since the results obtained are independent of the subjectivity of those involved in that

506 process. ?? McNeill, 1985) Phenomenology (ethno methodology), has as main principle that there is not a unique  
507 truth. According, the explanation of a phenomenon emerges Year 2020

### 508 **33 ( ) C**

509 © 2020 Global Journals from different points of view people affected and involved have in relation to the  
510 phenomena analyzed. This is an action-reaction process. Every single person has his own interpretation of  
511 the world and phenomenon. In this sense, there are different truths and realities, and sharing meanings and  
512 interpretations vary depending on the context. ??McNeill, 1985) Realism shares some philosophical views with  
513 positivism, since it is based on the impression that there is an intention reality, which is self-governing of human  
514 beings' thoughts and beliefs. However, realism also recognizes that humans are not substance to be considered  
515 in the style of natural science. On the other hand it takes social influences into account. Realism recognizes  
516 the importance of the fact that those social influences, although are independent of individuals, affect the way  
517 people make sense of their world, whether they are conscious of these forces or not. (Saunders et. al., 2003)  
518 The comparison between the characteristics of each research philosophy is summarized in the following Table  
519 3. In this research, researcher uses realism philosophy because it helps to find out the research questions more  
520 efficiently.

### 521 **34 c) Research Design**

522 According to Kerlinger (1994) "A research design is the plan, strategy and structure of exploration conceived so  
523 as to achieve answers to research questions and to control variance."

524 Sekaran (1992) states, research has been defined as:

525 "An organized, data based critical, systematic, scientific enquiry and exploration into a particular difficulty,  
526 undertaken with the intention of finding answers or solutions to it."

527 According to Saunders et. al.(2003), there are three different types of research design, which are; 1) Exploratory  
528 2) Descriptive 3) Explanatory.

529 The concept of each is discussed below.

### 530 **35 i. Exploratory**

531 Exploratory research is a kind of investigate conducted because a problem has not been evidently defined.  
532 Exploratory research helps decide the best research design, data collection process and variety of subjects.  
533 Investigative research relies on Secondary research. Though, research that is conducted with a desire to discover  
534 are called an exploratory research.

### 535 **36 ii. Descriptive**

536 Descriptive analysis describes data and characteristics about the society or phenomenon being studied. If the  
537 function of the study is to describe, the study is measured to be descriptive in character. It mainly gives the  
538 researcher a choice of aspects, perspective, levels, terms and concepts, as well as to observe, register, systemize,  
539 classify and interpret.

### 540 **37 iii. Explanatory**

541 Explanatory research is useful when the issue is previously known and has a explanation of it. The ambition  
542 to know "why" to provide details is the point of explanatory research. It builds on descriptive and exploratory  
543 research and goes on to identify the cause for something that occurs. Explanatory research looks for reasons and  
544 causes.

545 The different between exploratory, descriptive and explanatory research design In this research, the researcher  
546 has explored "Marketing strategy in fast food restaurant" in particular through his own literature view. The  
547 researcher has tried to explore the relationship between the impacts of marketing strategy in fast food restaurants  
548 and consumer intentions of coming back to the restaurant. On the beginning of this correlation the researcher  
549 has been capable to explore the various features of the marketing strategy. Consequently, the researcher has  
550 coined his research as an exploratory research.

### 551 **38 d) Data collection Method**

552 Data collection method is an important stage of a research and must be well planned to ensure that researchers  
553 will not face the problem of being overwhelmed by the data, which become a barrier rather than an aid to the  
554 research project. In order to be able to plan and organize data collection systematically, an understanding of  
555 the various types of data depending on different approaches to, methods of, and techniques of data collection is  
556 significantly required.

557 According to Saunders et. al. (2003), data comes in various shapes and forms, but can be distinguished  
558 between two main categories: 1. Secondary data, and 2. Primary data.

559 i. Secondary Data Secondary data is data which has been composed by agencies or individuals for purposes  
560 other than those of our meticulous research study. For example, if a management has carried out a review of,

---

561 say, expenditures of family food, and then a food producer might apply this information in the organization's  
562 assessments of the whole probable market for a fresh product. Similarly, statistics arranged by a ministry  
563 on agricultural production will demonstrate useful to whole lots of people and organizations, including those  
564 marketing agricultural supplies.

565 The most frequent exercise of secondary data in marketing research is to achieve familiarity and to create a  
566 background in which primary data are composed, reported and analyzed, the problem is defined, and the research  
567 is planned. This approach is a literature search -an assessment of exiting material, penetrating for information  
568 pertinent to the present marketing research project. Materials are typically scholarly magazines, journals, books,  
569 newspapers, and company records (accessed through computer data bases). ??Patzer L. Gordon, 1995) Secondary  
570 data can give information about performance and procedures for conducting marketing research. For example,  
571 these data can help learn language for communication with the research sample members, questions and topics  
572 to avoid, problems likely to be encountered, and statistical techniques to engage. ??Patzer L. Gordon, 1995)  
573 Secondary data are potentially misleading term for people not experienced with marketing research. For example,  
574 it is misleading to think of secondary data as being of second importance, minor importance, inferior value, or  
575 in any way not necessary. Their worth, like that of all data, depends instead on the marketing research project.  
576 However, it is reasonable to conclude that secondary data play a significant role in almost all marketing research  
577 projects. Another misconception is to think of secondary data as coming second in a sequence. The sequential  
578 order is just the opposite: secondary data typically are collected and analyzed first, before primary data. ??Patzer  
579 L. Gordon, 1995) ii. Secondary data sources Book reviews: The external research will be carried out through the  
580 reading and understanding of published material. This includes books and articles written on online shopping,  
581 catalogue shopping and consumer perception and satisfaction. Book and journal reviews are a very good source  
582 of collecting data as can get a wide variety of theories and authors references.

583 Internet Research: Internet research is another source of secondary data. This will be used to gather historical  
584 and present information about online shopping, catalogue shopping and consumer perception and satisfaction.  
585 This will also help to get contact details about the bottom level consumer as a whole. Helps to gather and  
586 analyses articles and journals about catalogue shopping and consumer perception and satisfaction. Collecting  
587 data from internet search is widely used now a days and is very quick and also you can get a wide variety of data  
588 through internet search.

589 Documents: Documents can be treated as a source of data in their own rights. In effect it can be an alternative  
590 to questionnaires, interviews or observation. This includes published materials of company details, like annual  
591 and financial reports of the proposed banks as well as other banks.

### 592 **39 iii. Primary Data**

593 Primary data means the data that are to be collected by the researchers themselves through a variety of  
594 data collection methods and techniques, for example, interviews, questionnaires, experiments, observations etc.  
595 Although the process of collecting primary data may have more requirements than secondary data in terms of  
596 time , effort and resources, the result is likely more relevant for answering the research question.

597 Regarding collecting data primarily, we can distinguish the type of data collected into two subcategories;  
598 1.quantitative data and 2. Qualitative data

## 599 **40 Quantitative Data**

600 Quantitative data means data which is number based or can expressed numerically as well as classified by some  
601 numerical value. In contrast, qualitative data means data which is in the form of Year 2020 Quantitative data is  
602 more objective and scientific than qualitative data. It involves the implication that what is being researched can  
603 be quantified, and, therefore, is only applicable to incidence that can be quantified and measured.

## 604 **41 Qualitative Data**

605 Qualitative data explained items in terms of some feature or category that possibly informal or may use  
606 comparatively imprecise characteristics such as benevolence and flavor. However, qualitative data can contain  
607 well-defined aspects such as gender, nationality or object type.

608 Qualitative Analysis conducted through the use of conceptualization iv. Primary Data Sources Interviews:  
609 Direct Interview is one of the major sources of primary data today. This method is would be used for the  
610 internal research. The internal research will focus on a few semis structured interviews with a few senior and  
611 top managers. The intention is to ascertain a true picture of the perceptions and satisfactions that a consumer  
612 feels when they eat in a fast food restaurant. These interviews will help to find out the secrets of their success  
613 or reasons for failure.

614 Interviews are a good source of collecting data. Also it is relatively cheap and quick to collect data through  
615 conducting interviews. But also there are some disadvantages in conducting an interview: -1. As the nature of  
616 topic suggests it will be highly impossible to contact top level officials of the company and to ask them to give  
617 information about their company. 2. The second disadvantage is that the nature of the topic is so complex that  
618 there is a chance of getting biased opinion and it will be highly risky to rely on these answers.

619 Questionnaires: Another methodology that is the questionnaires. In this research, researcher uses seventeen  
620 relevant questions to find out the findings of this research which are given in APPENDIX 1.

621 Questionnaires are more economical, easier to arrange, the answers will be standardized. In situations of  
622 difficulty to get appointments with the top-level managers this method would be used to. Postal questionnaires  
623 will be sending to top managers of the banks and the responses can be analysed.

624 Collecting data from questionnaires is often for getting information and also it is relatively cheap. But it also  
625 has got some disadvantages like: -1. Collecting data from questionnaires is a long procedure and takes long time  
626 to collect and analyze such data. 2. The second disadvantage is that people generally don't like to spend time  
627 in giving answers in writing.

### 628 **42 e) Data Analysis**

629 After the data have been composed, the researcher turns to the responsibility of analyzing them. Analysis of  
630 this data needs a number of closely connected operations such as creation of category, the importance of these  
631 categories to unprocessed data through tabulation, coding and then sketch arithmetical inferences. Scrutiny work  
632 after tabulation is mainly based on the calculation of various coefficients, percentages, etc. in brief the researcher  
633 can analyse the collected data with the assist of various numerical equipment.

### 634 **43 f) Reliability**

635 According to Joppe (2000) "The extent to which results are consistent over time and an accurate representation  
636 of the total population under study is referred to as reliability and if the results of a study can be reproduced  
637 under a similar Year 2020

### 638 **44 ( ) C**

639 © 2020 Global Journals methodology, then the research instrument is considered to be reliable."

640 In this study, the researcher used the method of qualitative research in order to explore and understand the  
641 implementation of marketing strategies in the fast food restaurants, so that the researcher was then able to  
642 compare and contrast the findings with the literature, and eventually, was able to give suggestions about the  
643 issue. The case to be explored is dynamic and complex, and, therefore, it cannot be ensured that the research  
644 can be replicated and will give the consistent result when the time and circumstances have changed.

### 645 **45 g) Validity**

646 According to Winter (2000) "The traditional criteria for validity find their roots in a positivist tradition, and  
647 to an extent, positivism has been defined by a systematic theory of validity. Within the positivist terminology,  
648 validity resided amongst, and was the result and culmination of other empirical conceptions: universal laws,  
649 evidence, objectivity, truth, actuality, deduction, reason, fact and mathematical data to name just a few."

650 Researcher built the validity by establishing correct operational measures for the concepts of study. Researcher  
651 used the structured questionnaire as the mean to obtain the data.

652 The questions were designed and pre-tested in order to minimize as much as possible the misunderstanding  
653 and problems for the respondents; meanwhile it also increased the internal validity and reliability of the data.

### 654 **46 h) Limitations of the research project**

655 Researcher found some limitation at the time of research work. These are: a) Extent of research will provide a  
656 general overview of the entire outsourcing operations rather than complete audit. b) Limited amount of time  
657 available for completing the study. c) May not be possible to conduct interview with all of the firm's clients. d)  
658 Some of the data gathered may not be totally relevant to the research topic. e) Research needs to be conducted  
659 on a very low budget. f) There could also be a problem with translating the questionnaires and interviews as the  
660 company is located in a region where English not the main language in use. So there are chances that some data  
661 corruption might occur.

### 662 **47 i) Methods for this Research**

663 The chapter describes the methodologies used in the research. The project used both qualitative and quantitative  
664 methods. Using the following methods, a detailed study of the software development methodologies were carried  
665 out. The research is in two sections. Primary research carried out with a questionnaire. It consists of a survey.  
666 The Secondary research comprises of Literature survey from various sources.

### 667 **48 i. Completing the Questionnaire**

668 The questionnaire has been created in a way so that responders can answer quickly and easily. It is divided in  
669 to three main sections and contains all closed-ended questions. The time taken to complete the questionnaire was  
670 approximately 15 minutes to 17 minutes.

---

671 Individual and organizational questions -This section contains questions on respondent's position in the  
672 industry as well as the position of the organization. It also contains question on the size of the organization  
673 including the number of employees, the projects they adopt and the likeness of adopting new technologies.

674 Methodology knowledge questions -This section focused on the knowledge of the respondent on the method-  
675 ologies Software development questions -This section contains questions on the different agile and traditional  
676 methodologies used on different projects. This is scaled on the project sizes measured in person months. The  
677 scales are selected as small scale, medium scale and large scale projects. There are questions to capture the  
678 opinion of the respondents on how effective the used methodologies were with regard to cost and quality of  
679 the software. Finally questions were included to capture their opinion on the preferred characteristics of both  
680 development methodologies from their point of view. The questionnaire is included in Appendix 1.

## 681 **49 ii. Target Audience**

682 The questions were distributed among software companies of various sizes and types. The respondents involved  
683 were mainly software architects, software engineers, and project managers. However, there were some other roles  
684 involved in software development as well.

## 685 **50 j) Research Audit**

686 Different resources were used for the research. The resources include various books on software engineering and  
687 development methodologies from different authors including Cockburn, journals related with software industry,  
688 white papers on agile and traditional methods, and websites from the internet which are related with the subject  
689 area.

690 IV.

## 691 **51 Research**

692 The following are based on the data that were collected from various companies in the software industry. A  
693 questionnaire was prepared and provided in order to collect these data.

## 694 **52 a) Data Collection**

695 Most of the respondent was from Bangladeshi 21 different organizations. Among the organization 15 organizations  
696 were Information technology related organizations, 3 organizations were from Year 2020 ( ) C

697 © 2020 Global Journals

698 Telecommunication, 2 organizations were from the Engineering and 1 was other organization.

## 699 **53 i. Organizational Characteristics**

700 When analyze the results from the sample question it was discovered that about 70% of the respondents  
701 were from organizations with an Information Technology background. There were other respondents from  
702 telecommunication, engineering and medical organizations as well. There were some cases that projects were  
703 outsourced to information technology organizations. It is shown in the table below in a ratio of 100 ii. Individual  
704 Knowledge Gathering from Different Organization Among the organization the number of the respondent was 21  
705 and majority of the respondents to the questionnaire were software developers. The other respondent were System  
706 Analysts, software engineer and software architects. It is shown in table below in a ratio of 100. When it comes to  
707 the knowledge rating for different methodologies more than 90% of the respondents have an understanding about  
708 agile and traditional methodologies in an average or higher level. 12 out of the 15 respondents have rated their  
709 knowledge of agile methodologies as average or broader, out of that 6 of the respondents rated their knowledge as  
710 broad or very broad. For traditional methodologies the rating was broad or very broad for 12 respondents. When  
711 compare the experience they have in the software industry it was revealed that with less experience in the field  
712 or in other words people who have experience less than four years have less practical knowledge in traditional  
713 methodologies. Figure 12 presents the results below. According to a survey results published by Ambler in early  
714 2006 he has found that even though more than 60% were fully or partially using agile, there is a considerably  
715 a large number of organizations who are still have no idea of adopting agile. In that survey result shown below  
716 In this survey about 75 % of the respondents are either leader or follower to adopt agile methodologies, on the  
717 other hand in Ambler survey 65% of the respondent said yes for adopting agile methodology.

718 So with the result of this survey, it proves that other than the organizations who adopted agile at the beginning,  
719 the potential of organizations adopting agile at a later stage without any assurance is minimal. A possible reason  
720 could be that these organizations are just waiting to see how agile projects will result in the future.

## 721 **54 b) Methodologies used in Organizations**

722 In this section focus on which type of agile and traditional methodology used on different organizations. If the  
723 organization is small which types of agile and traditional development methodologies they are using. In similar  
724 way it is focused for the medium and large scale organizations.

### 55 i. Use of Agile Methodologies

According to the respondents of the different types of organization indicate that Extreme programming (XP) is the most popular method used in the industry. But SCRUM also maintains a good position within the industry even though it is not up to XP level. There was a remark about XP stating that it sometimes gives bit of a fear because of the steps it includes and also the "Name itself". There is an interesting point that was found during the analysis. The next most popular was in-house build methods by organizations for their own use. In an article published by Sliwa (2002) mentions that agile methods can be mixed for different organizations purposes. The article further stated that; "Schwaber, a Scrum co-creator, said it makes sense to combine Scrum and XP because Scrum focuses on management practices and XP centers on engineering practices for building object-oriented software."

The result proves this point as organizations are already using combined methods according to their needs for a better result. Another point was that some organizations tend to mix other new techniques built for specific tasks in software development with their development methodology. For example they use scheduling techniques such as planning poker for estimating time for development tasks. Planning poker is a technique which is used in Scrum in most cases to estimate time for development tasks. It has a deck of cards with different estimates which the developers can use (Cohn, 2005) cited by planningpoker). Figure 14 represent the results of respondent. In figure 15 it is indicate that for small, medium and large scale organizations Extreme programming (XP) is most popular among the agile methodologies and SCRUM is in second position. Other methodologies are using in a very small scale in different organizations.

### 56 ii. Use of Traditional Software development Methodologies

According to the respondents of the questionnaire, for traditional methodologies more than 50% respondents use the waterfall model, 23% of respondents were interested in unified process and the rest was on in-house build methods for different type and sized projects. Figure 16 represent the results.

### 57 Analysis

In this chapter the discussion will be focused on analyzing these collected data and find out the responses from the software industry professionals. When consider the small scale software projects more than 30% of the respondents think that working software is more important. People interactions and responding to change come after respectively. All the respondents' believe that human interaction is an important fact for better software development regardless of the project size. Cockburn (2001) point out that;

### 58 a) Most Appealing Agile Values over Traditional Characteristics

"Core to agile software development is the use of light-but sufficient rules of project behaviour and the use of human and communication-oriented rules" proving the point made out from the survey results.

Respondents' have a different view about Medium and large projects. For both of these project types, customer collaboration have obtain the highest votes. This means that when the system is getting bigger more customer collaboration helps to keep the development on the track. Medium projects have a higher percentage of votes for people interaction than large projects. Even though this is outside the expected result for large projects, it may be due to the reasons that respondents think it is hard to communicate within large projects. Figure 17 represents the results obtained.

### 59 b) Factors that Influence to use Agile Methods over Traditional Methods

Cost and quality of software products are the main concerns in the industry when it comes to software engineering. It is important for both software organization as well as the customers (Krasner, 1998).

Due to this in the questionnaire, it was necessary to include questions regarding the cost and software quality. The reason was to find out how agile methodologies have affected on these two features of a software project compared with traditional methodologies.

The questions were targeted to capture the opinions of the respondents, whether they believe by adopting agile methodologies will affect the software cost and quality of a project than the traditional methodologies. Since agile is making a huge entrance to software industry I was expecting a very higher positive feedback. Even though the result was rather different from what I was expecting.

When it comes to cost of the software project 50% of the respondents agreed that there were no change in cost at all by using agile methodologies but according to Ambler (2007) it was 47 % (in Figure 19). Surprisingly 22% of the respondents have voted as the affect of the cost has slightly decreased than the traditional methodologies but according to Ambler (2007). Only about 18% of the respondents believe that agile methodologies have made a slight increase affect on cost. The rest of the respondents falls both sides to the far end of the ratings. But with regard to software quality the result I obtain was different than the cost. Overly respondents have a

---

781 positive feedback on the quality. More than 30% of the respondents believe that adopting agile methodologies  
782 have slightly increased the affect on quality compared with traditional methodologies. 13% of the votes were even  
783 higher. They believed that the affect was in a very higher state. But again there were huge number respondents  
784 who really did not believe in agile methodologies as 39% was on the no change state. The rest was in the low  
785 side of the rating.

786 Figure 18 and figure 19 show the results for the software cost and quality I discovered. So comparatively  
787 organizations believe that there is a higher effect to quality from agile than the effect to cost. In the survey done  
788 by Ambler in early 2008 the results on quality was noticeably different. In his survey 67% of the votes said that  
789 they experienced better or significantly better affect on the quality of software projects with the adaptation of  
790 agile methodologies. The difference to my results is that considerably a large number of respondents voted for no  
791 change. Since the results I got mentioned about medium size projects than other two there can be issues occur  
792 when practicing agile values such as team communication and customer feedback. Due to these reasons there  
793 may be problems when try to capture the quality of the project. The cost affect was slightly tally with the results  
794 from the Ambler's survey. Figure 18 and figure 19 show the results for the software cost and quality I discovered.  
795 This two survey result on software cost are closely similar and their correlation coefficient is 0.991039.

796 ii. Comparison of software quality from agile methodologies between Ambler and this survey This two survey  
797 result on software quality are closely similar and their correlation coefficient is 0.87579.

## 798 **60 c) Preferences for Agile and Traditional Methodologies**

799 When an organization uses a methodology, there are processes and techniques they have to follow regardless of  
800 the type of the methodology. From the past experiences in the industry I had the understanding that there were  
801 some processes which development teams think is useless for the success of the project objectives. To have a  
802 broader view in these aspects questions were included in the questionnaire to find out respondents opinion on  
803 certain characteristics in both methodologies.

804 According to the results shown in figure 22 more than 50% of the respondents' believe that low management  
805 control is a drawback for small scale and medium scale projects. In fact they believed that low management  
806 affects all sizes of projects in a considerable amount. The other major aspect was the project structure. Again  
807 all the respondents' believed "Documentation should be assigned a cost and its extent be determined by the  
808 customer.

809 Many organizations demand more than is needed. The goal should be to communicate effectively and  
810 documentation should be the last option."

## 811 **61 d) Methodology preferred**

812 When analyzing data to find out which development methodology is preferred by the respondents I have realized  
813 that agile has come a long way during the past few years after it was properly published. But on the other hand  
814 it still has to go further to take over the whole software market.

## 815 **62 i. Methodology selection for different project sizes**

816 The results discover that almost all the respondents have agreed that agile methodologies are the best for  
817 small scale projects. This means that software organizations getting to know how to get their hands on  
818 agile methodologies to manage the tasks in small scale project environments. For medium scale projects both  
819 methodologies were voted. The gap between the results for the two methodologies was very less. This shows that  
820 agile is adopted by organizations than before for medium scale projects. But respondents had a different idea  
821 about large scale projects. Nearly © 2020 Global 90% of the responds were bias to traditional methodologies.  
822 Only the remaining was for the agile and other methodologies.

823 The interesting fact was that organizations are using a mix of both methodologies when it comes to medium  
824 and large scale projects. Medium size projects are in this process more than the large size projects but it seems  
825 within the next few years large scale projects may also start to use a mix of both methodologies. Figure 24 below  
826 represents the methodology selection. The other fact was that some organizations mix other techniques also into  
827 their methodologies. Some respondents have rated for Scrum or Scrums and also planning poker which are new  
828 techniques to make the development processes more efficient. Figure 25 represents the total results.

## 829 **63 Use of agile methodologies**

830 Around 40% to 50% of the respondents' agree that for medium and large scale projects autocratic management is  
831 not necessary. This type of a management would keep the teams stick to the standard work and have no agility  
832 leaving the teams work without any innovation or creativity. Only for small scale projects some organizations  
833 use 100% of the agile methodologies. Other than that for both small and medium scale projects majority of the  
834 respondents' agree only up to 50% of agile methodologies are used. There was a comment from a respondent saying  
835 that "It is hard to stick to agile methods especially when it comes to large projects. There are other techniques  
836 been used mixing with the practices in both agile and traditional methods?" This means that organizations are  
837 tend to use their own in house methodologies created to suits the projects they handle.

838 The findings of this project research study also confirm the appropriateness of the use of agile methodologies for  
839 small scale projects, traditional and agile methodologies for medium scale projects and traditional methodologies  
840 for large scale projects of an organization.

## 841 64 VI.

## 842 65 Conclusion

843 The purpose of this research is to present a set of guide lines for a software organization to help choose the  
844 most appropriate development methodology according to most of the software projects they have in hand.  
845 The thesis starts with an overview of the software industry and explanation of the problem domain which is  
846 focused in the research. Through traditional and agile development methodologies, this discusses the different  
847 software development approaches used in the software industry. Further a discussion about the life cycles of  
848 selected approaches from both traditional and agile methodologies were carried out with identifying the roles,  
849 responsibilities and practices of each development approach. This would give the reader clear idea about the two  
850 methodologies and also the differences they have. Chapter 2 briefly presents a comparison on the methodologies  
851 and focuses on the problems in both methodologies. Finally, in order to get the professional opinions, the  
852 document presents the analyzed results from the survey conducted.

853 Throughout the research it was understood that the traditional methodologies were apparently handling a  
854 considerable portion in software industry. The basis was the complete planning, heavy documentation and  
855 extensive designs. Traditional approaches will still be useful in large, long lived projects that require special  
856 safety, reliability or security requirements. The military and defense industry gives a perfect example to prove  
857 this point. Lijek (2007) in a presentation discusses the reasons why agile methodologies are not adopted in the  
858 military and defense industry.

859 ? Defense Contractor Mentality regarding change ? Safety Critical Systems ? Long development cycles ?  
860 Large teams ? Customer Relations But in the near future with the improvements agile will be able to be adopted  
861 in these industries.

862 Agile methodologies cannot be defined by a small set of rules and practices. From the literature review and the  
863 survey results it became obvious that agile methods have the capability to respond to change faster, the ability  
864 to extract the hidden creativity and innovations out of the teams, the capability in balancing the structure  
865 and flexibility and to drive the organization through rough situations and uncertainty. Agile is more likely to  
866 dominate volatile environments with uncertainty and unpredictability where the exact customer needs are not  
867 clear. Organizations tend to respond to the market changes quickly with the customer needs. They make plans  
868 for the system but do not tie their view to it. Rather than making models they want to focus more on the working  
869 software. They focus on interaction within the team members, customers and management and individual skills  
870 as well. With all the readings and findings it is clear that there is no "one-size-fits-all" solution.

871 a) Limitations There were some obstacles on the way to the success of this thesis. At first, Gathering the  
872 information from the professionals and practitioners in the industry was a problem as it takes long time for most  
873 of them to respond to the questionnaire. There were some returned questionnaires half-completed which had  
874 to be discarded. Another barrier was the time factor. Even though there are lots of areas that can be focused  
875 under this topic it was not possible since the allocated time was limited. But within the time period a good and  
876 original piece of work was produced with great attention.

## 877 66 b) The Guidelines

878 The guide lines presented are to support an organization to select the most appropriate software development  
879 methodology for software projects they undertake. For an organization, it is hard to have more than single  
880 software methodologies operating. Generally the top management and human resources would prefer all projects  
881 to use the same method for ease of handling.

882 However, software developing is a complex and uncertain process. To cater for specific needs, Project  
883 requirements and different teams may have to produce different results. Therefore, it is important to consider  
884 adopting different methodologies or a mix and match of several techniques from different methodologies at least  
885 between two departments or two different project sections which operate independently in the same organization.

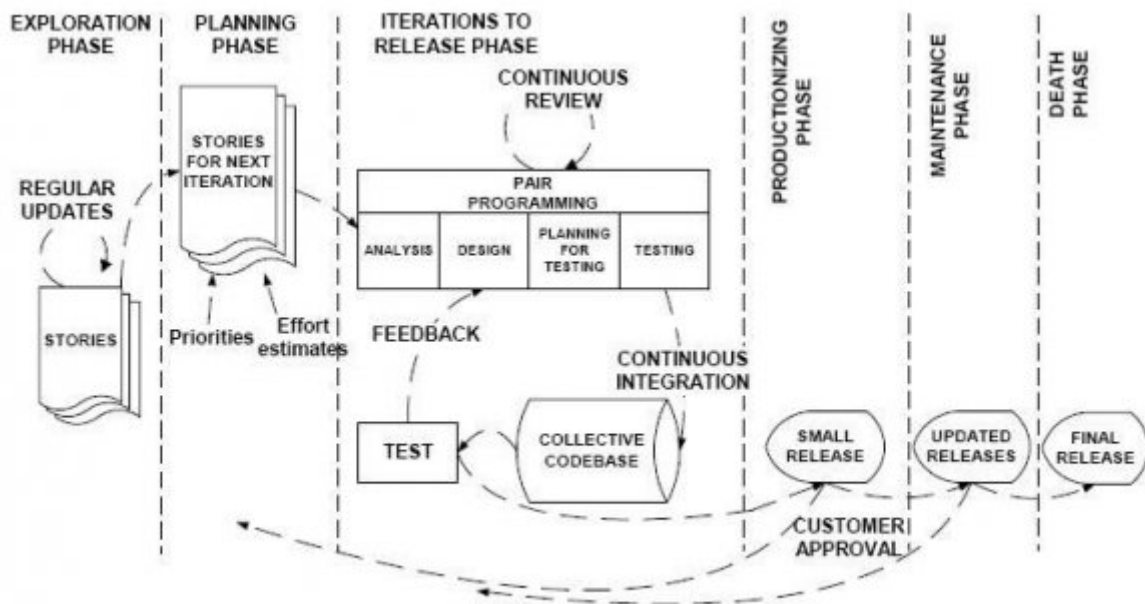
886 The following guidelines are created with the knowledge obtained from the research on the literature and  
887 the analysis and understanding gained from the survey results which involved the real software development  
888 organizations.

889 ? Flexibility -Everybody involved with software development needs to be flexible, starting from the top  
890 management. They should understand different projects have different needs and there are different ways to  
891 make them successful. ? Priority on the needs -Different projects need Different techniques and artifacts.  
892 Therefore, it is important to identify them and prioritize them. For an example the use of other techniques  
893 and artifacts outside the working methodology (e.g. planning poker) for certain types of projects may lead the  
894 project to greater success. But the management has to remember that, this may need some training to the team  
895 members as a person may have to deal with a range of methods and/or artifacts.

896 ? Cater according to the team -For different projects, Development teams may be different in size. So it is  
897 important to use suitable methodology or mix of methodology to cater for that requirement. As an example, XP



898 and scrum are suitable for projects with small-scale to medium-scale development teams with 4 to 20 members.  
 899 However, for large and medium scale teams Unified Process can be used. ? Define targets -There are specified  
 900 artifacts for each approach in traditional development. So organizations rely on these artifacts and always try  
 901 to stick to them. Rather defining the targets with the help of the customers on what to build may be more  
 902 productive. The artifacts will be decided along with the targets which is more useful for all the parties involved  
 903 in that specific project. ? The use of methods -Organizations with large or medium scale projects can combine  
 904 subsets of different methods. SCRUM is a methodology which can be mixed with different other methodologies  
 905 including XP and waterfall. However, for organizations, who handle small scale projects can settle with a single  
 906 methodology.  
 907 Come up with a specific set of rules is not that easy in a rapidly changing field with uncertainty like software  
 908 engineering. For different organizations, these guidelines can be used in different ways. With time and experience  
 909 these can be improved more. The best way is to experiment these in a real time environment and observe the  
 910 validity and the success, which will give an understanding on how to improve them for better results. Year 2020



1

Figure 1: Figure 1 :



2

Figure 2: Figure 2 :

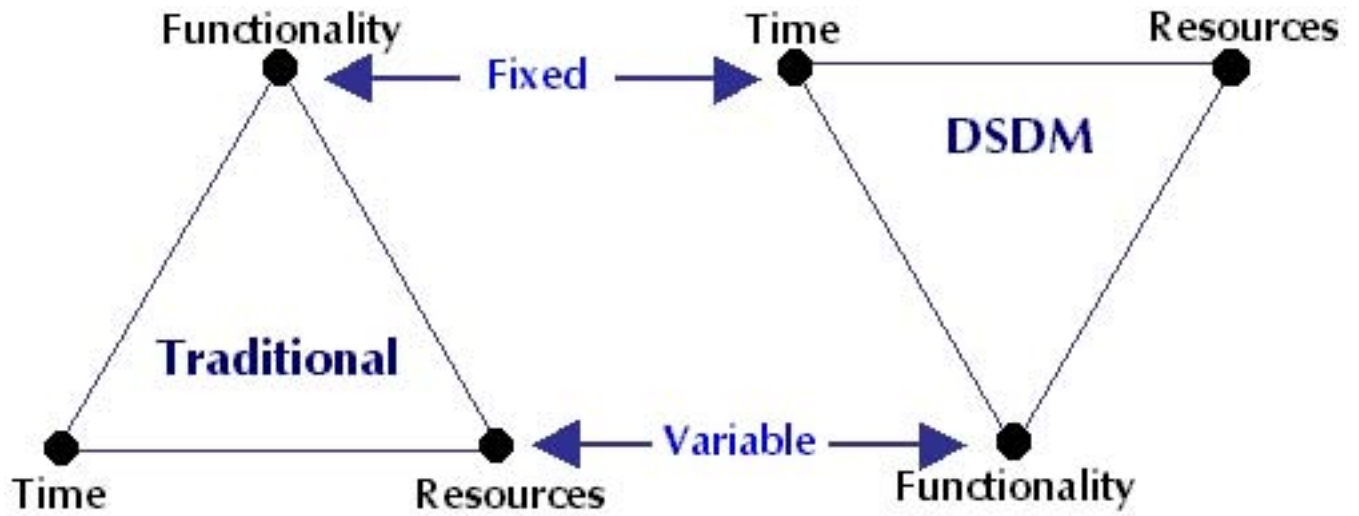
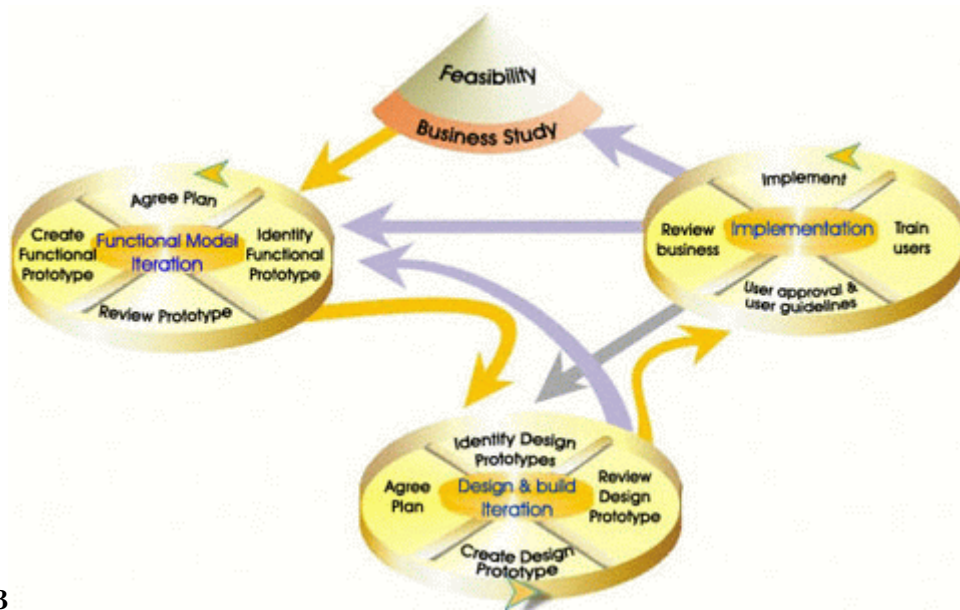


Figure 3:

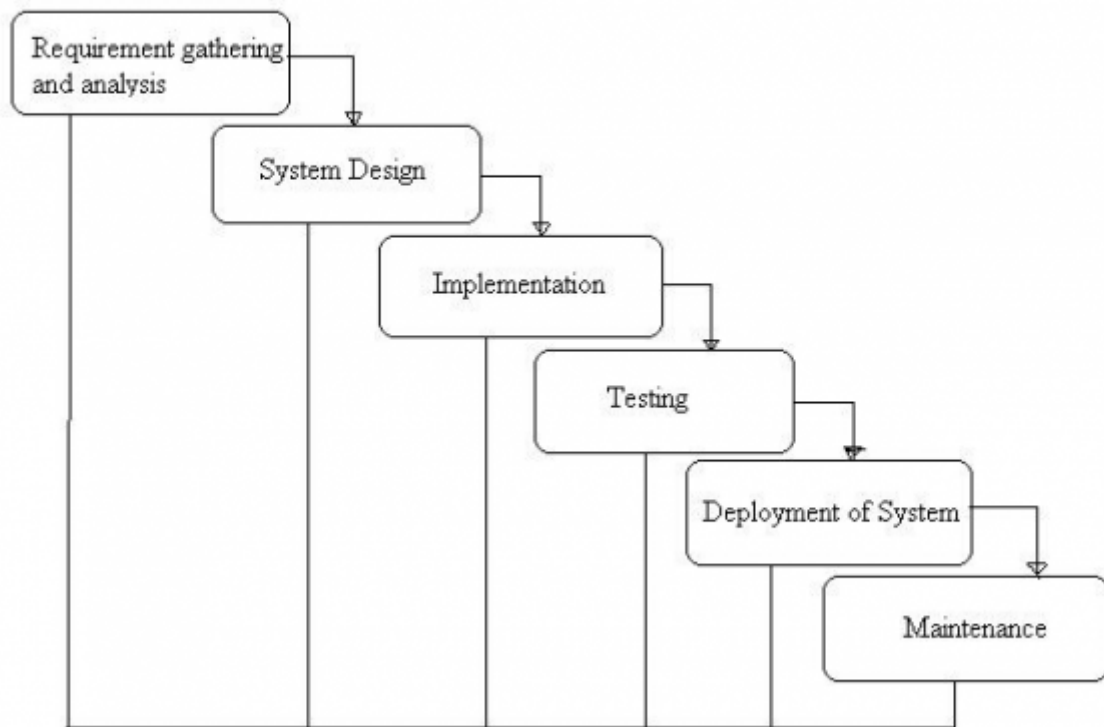


3

Figure 4: Figure 3 :

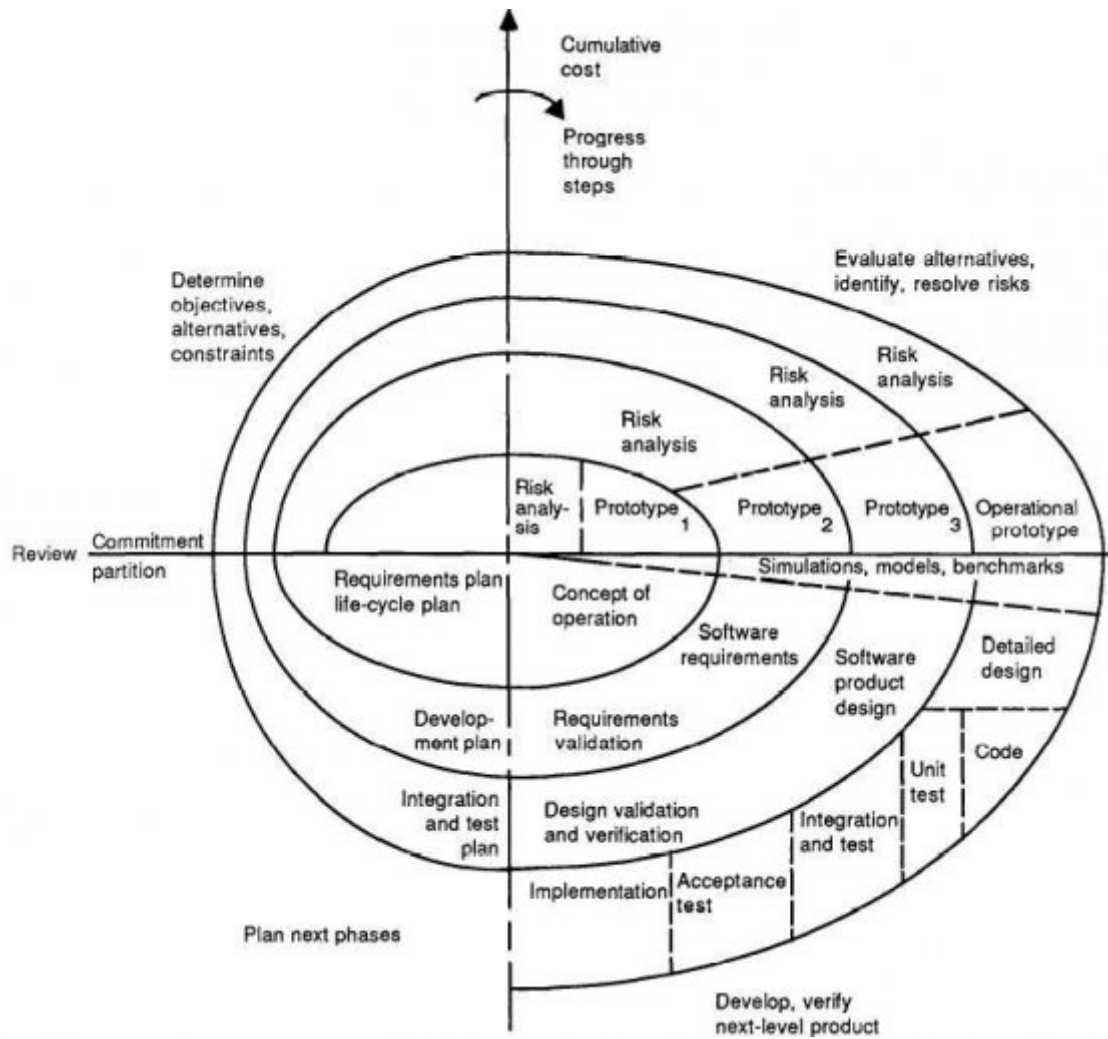
---

**General Overview of "Waterfall Model"**



4

Figure 5: Figure 4 :

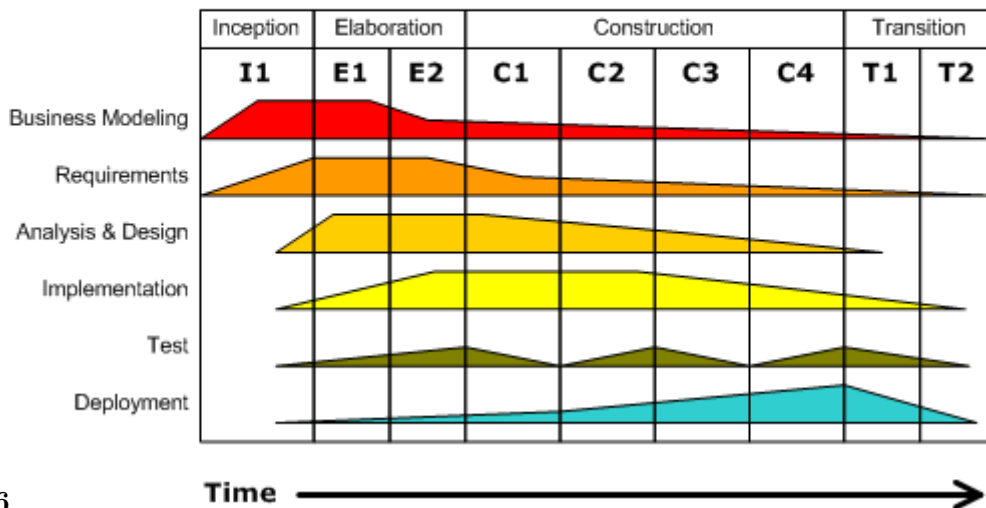


5

Figure 6: Figure 5 :

**Iterative Development**

Business value is delivered incrementally in time-boxed cross-discipline iterations.



6

Figure 7: Figure 6 :

|                       | <b>Agile Methods</b>      | <b>Heavy Methods</b> |
|-----------------------|---------------------------|----------------------|
| Approach              | Adaptive                  | Predictive           |
| Success Measurement   | Business Value            | Conformation to      |
| Project Size          | Small                     | Large                |
| Management Style      | Decentralized             | Autocratic           |
| Perspective to Change | Change Adaptability       | Change Sustaina      |
| Culture               | Leadership-Collaboration  | Command-Contro       |
| Documentation         | Low                       | Heavy                |
| Emphasis              | People-Oriented           | Process-Oriented     |
| Cycles                | Numerous                  | Limited              |
| Domain                | Unpredictable/Exploratory | Predictable          |
| Team Size             | Small/Creative            | Large                |
| Upfront Planning      | Minimal                   | Comprehensive        |
| Return on Investment  | Early in the project      | End of the projec    |

Figure 8: Figure 7 :

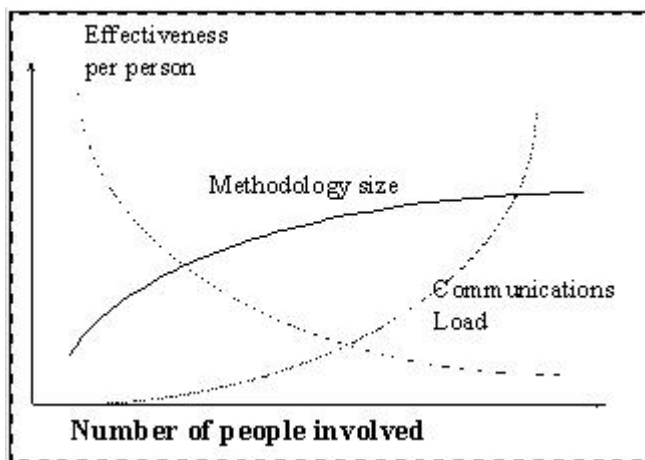


Figure 9:

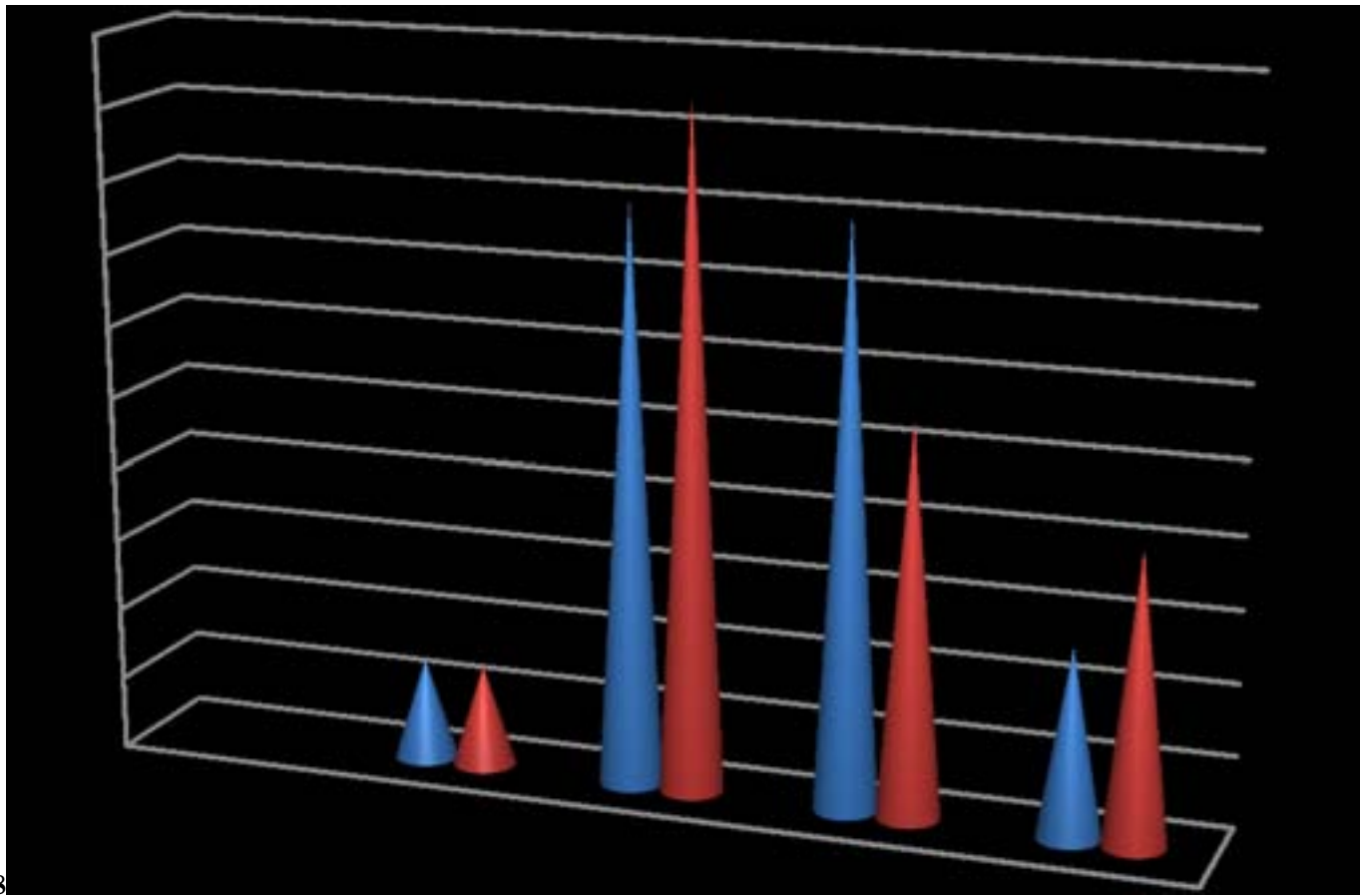


Figure 10: Figure 8 :

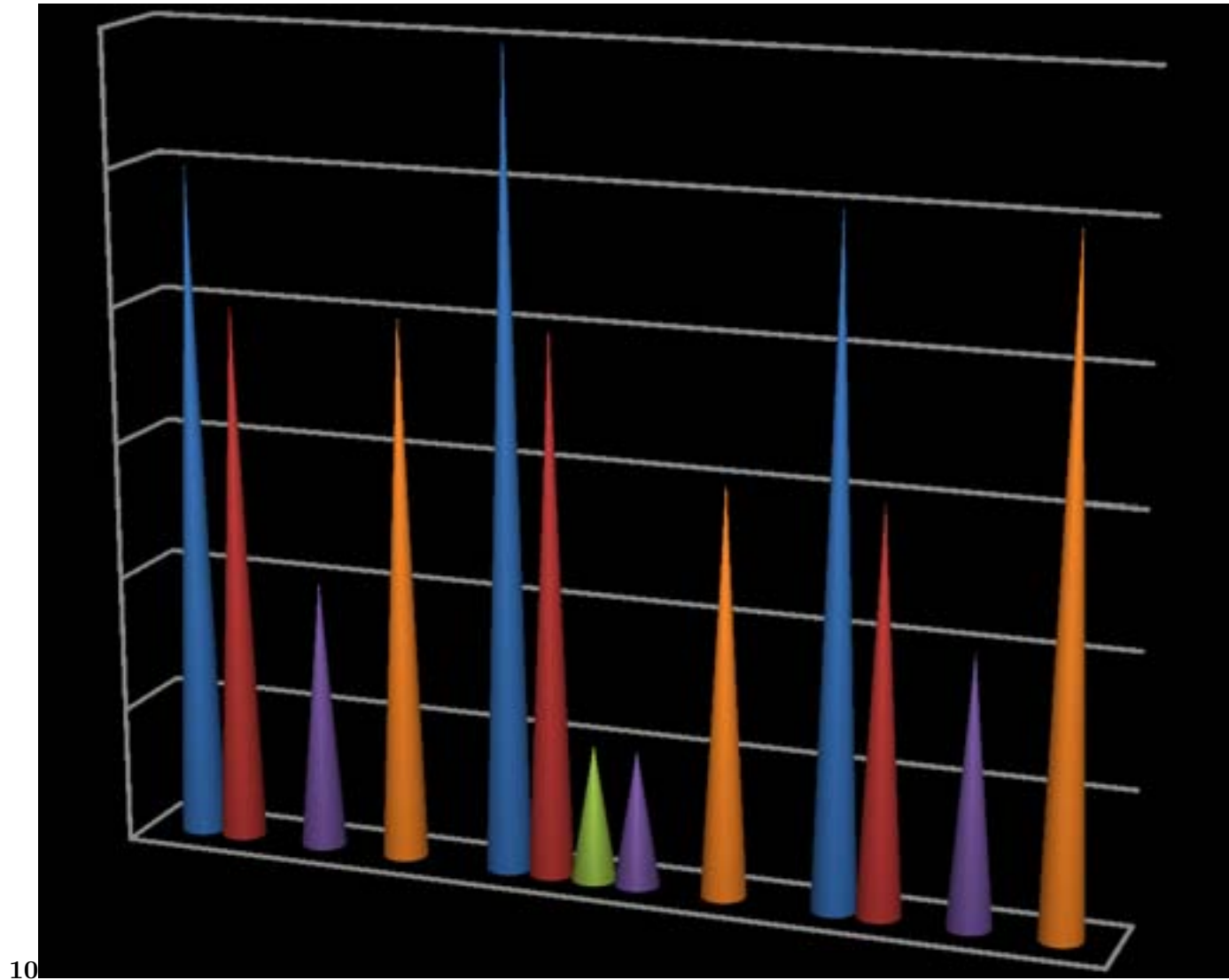


Figure 11:



Figure 12: Figure 9 represent the resultsFigure 9 :





10

Figure 13: Figure 10

12 

Figure 14: Figure 12 :

13 

Figure 15: Figure 13 :

14 

Figure 16: Figure 14 :

15 

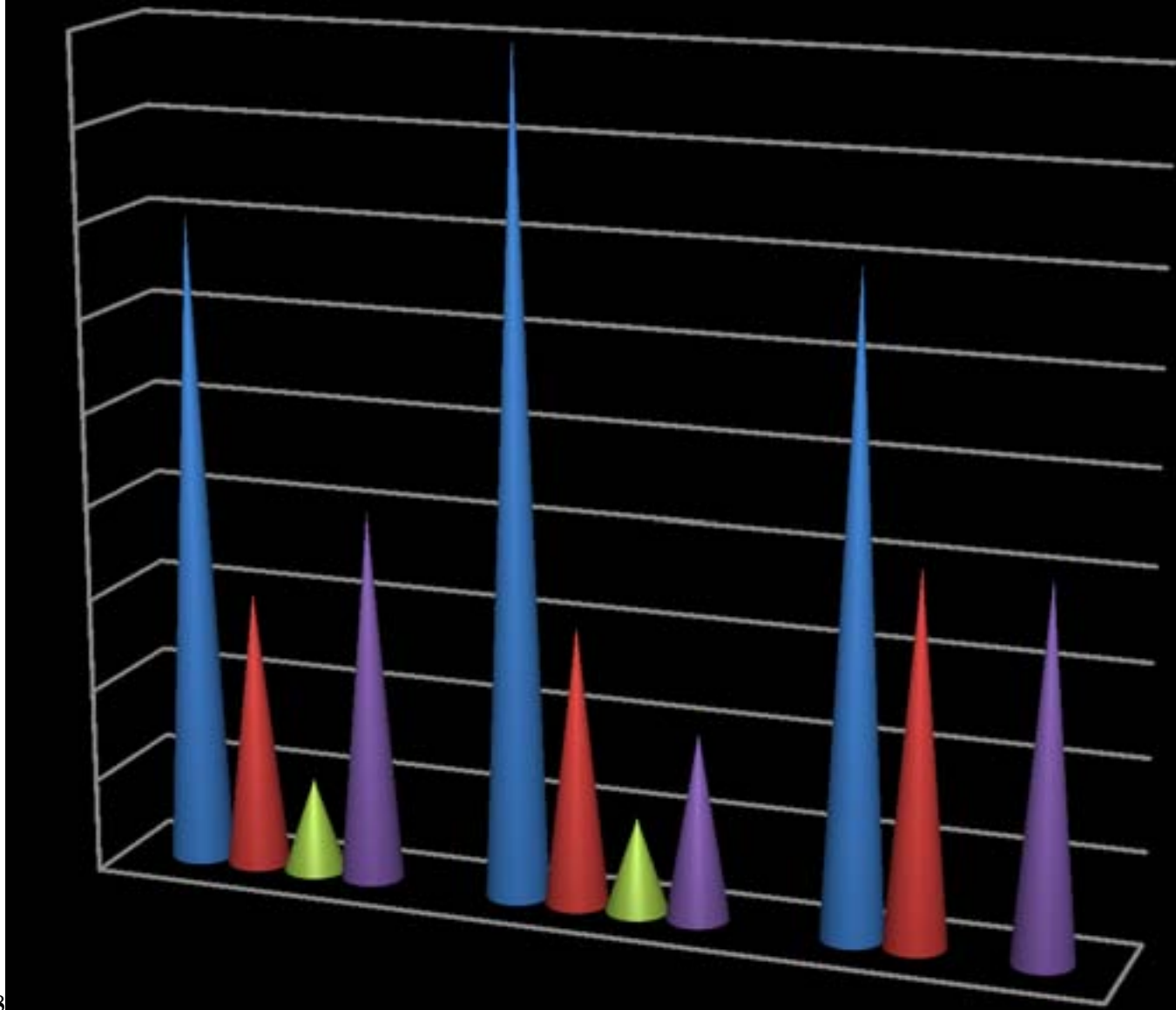
Figure 17: Figure 15 :

16 

Figure 18: Figure 16 :

17 

Figure 19: Figure 17 :



18

Figure 20: Figure 18 :

19 

Figure 21: Figure 19 :

20 

Figure 22: Figure 20 :

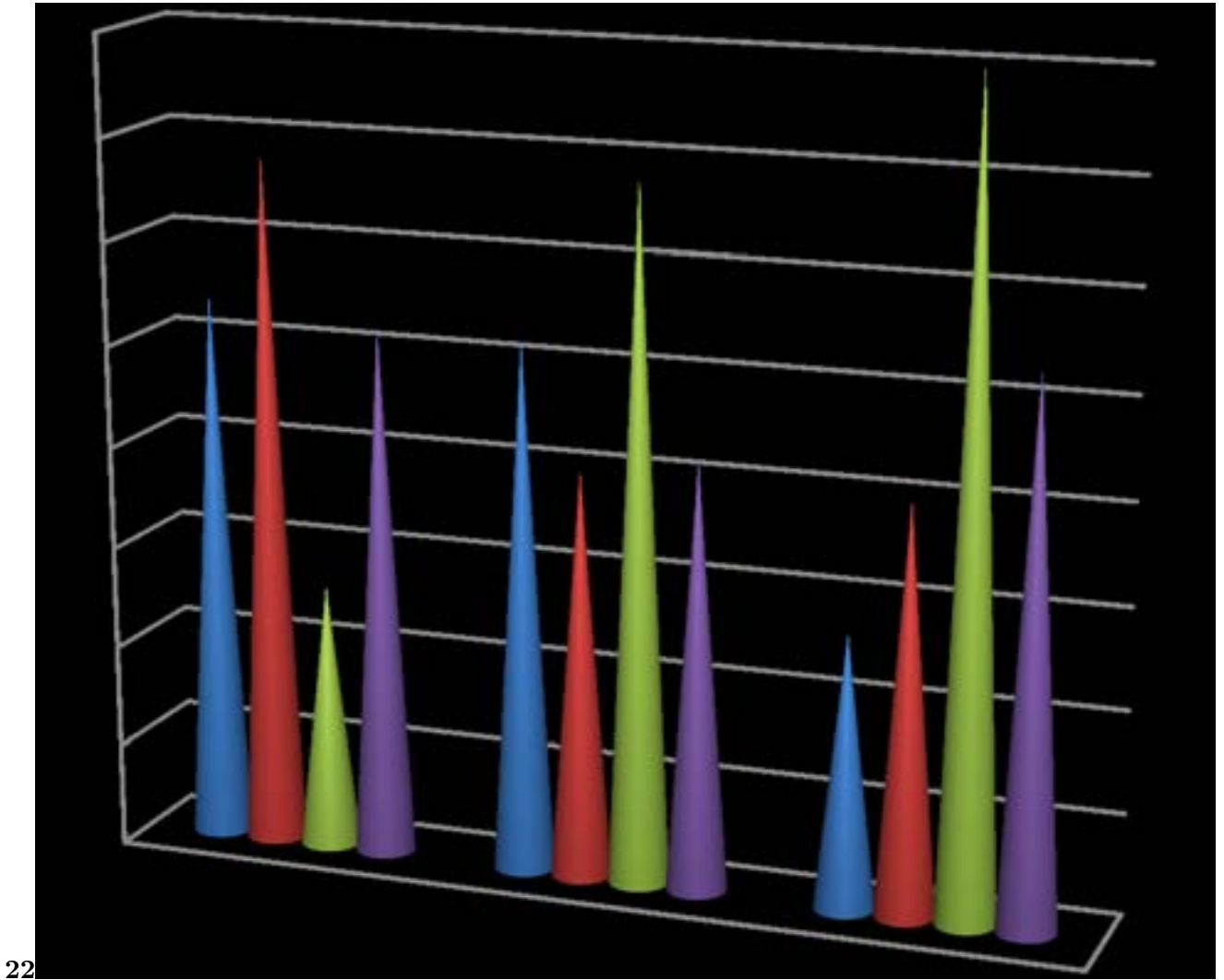
21 

Figure 23: Figure 21 :





Figure 24:



22

Figure 25: Figure 22 :



23

Figure 26: Figure 23 :



24

Figure 27: Figure 24 :



25

Figure 28: Figure 25 :



26

Figure 29: 26 :



26

Figure 30: Figure 26

Abstract-Agile and Traditional software development methodologies, both are being used in different projects of software development industry. Agile software development technology is an incremental software development process.

Figure 31:

1

|  |  |
|--|--|
| <p>Privileged methodological text<br/>Information systems development is<br/>A managed controlled process<br/><br/>A linier sequential process<br/><br/>A replicable universal process<br/><br/>A rational, determined and goal<br/>driven process</p> | <p>Marginalized methodological text<br/><br/>Random, opportunistic process driven by ac-<br/>cident<br/><br/>Processes are simultaneous and overlapping<br/>and there are gaps in the between<br/><br/>Occurs in completely unique and idiographic<br/>forms<br/><br/>Negotiated, compromised and capricious</p> |
|--|--|

Figure 32: Table 1 :

*[Note: and gives feedback on how accurate they are to make better future estimates. Further tracker traces the progress of iterations and evaluate if the project goals are reachable within the allocated time with the current resources. The tracker is a programmer, not a manager or customer.? Programmer -Writes tests, design, and code and try to keep them simple and definite as possible. They refactor code identify and estimates tasks and stories. ? Tester -Helps customers write and develop functional tests. They run functional test often to broadcast results and they maintain the test tools. ? Customer -Writes stories and acceptance tests.]*

Figure 33:

- f) Traditional Software Development
  - i. Outline
    - ”By applying a methodology to the development of software

Figure 34: ?

2

Figure 35: Table 2 :

3

| Positivism   | Phenomenology  | Realism  |
|--|--|--|
| Objective truth analysis<br>Value-free data collection<br>Law-like generalization<br>Quantitative approach | Subjective truth analysis<br>People's account, motives and intentions<br>Complex and dynamic<br>Qualitative approach | Socially constructed environment analysis<br>Independent reality<br>Social influences recognized<br>Qualitative approach |

Figure 36: Table 3 :

4

| Exploratory  | Descriptive  | Explanatory  |
|--|--|--|
| A study to find new insights<br>Useful for clarifying the understanding of the problem<br>Qualitative approach | A study to describe an accurate profile of persons, events or situations<br>Useful for giving details of incidence or phenomena and for predictive findings<br>Quantitative approach | A study to find casual relationship between variables<br>Useful for explaining the relationship of two or more incidents in terms of cause and effect<br>Quantitative approach |

Figure 37: Table 4 :

5

|  |   |
|--|---|
| research apply individual in detail interviews, focus groups or questionnaires to gather | people do and say. It reports on the concepts, meanings, definitions,                 |
| Based on meaning derived from numbers  | Based on meanings expressed through words   |
| Collection results in numerical and standardized data                                    | Collection results in non-standardized data requiring classification into categories. |
| Analysis carry out throughout the use of statistics and diagram                          |   |

Figure 38: Table 5 :

6

|                               |                          |                    |              |
|-------------------------------|--------------------------|--------------------|--------------|
| Information technology<br>70% | Telecommunication<br>15% | Engineering<br>10% | Others<br>5% |
|-------------------------------|--------------------------|--------------------|--------------|

Figure 39: Table 6 :

7

5%  
10%

15%

Information  
technology  
Telecommunication  
Engineering

Developers 53% Figure 10 represent these results. iii. 70% Analyst Software engineer 13% 20% Respondents

Figure 40: Table 7 :

8

| Less than 10 | Between 10 and 20 | Between 21 and 50 | Between 51 and 100 | More than 100 |
|--------------|-------------------|-------------------|--------------------|---------------|
| 7%           | 20%               | 26%               | 27%                | 20%           |

Figure 11 represent these results

[Note: Figure 11: Employees' in software development in organizations iv. Agile and Traditional Software methodology Knowledge of the respondents]

Figure 41: Table 8 :

9

| Cost        | This surver | Ambler (2007) survey |
|-------------|-------------|----------------------|
| Very low    | 7%          | 2%                   |
| Slight low  | 22%         | 20%                  |
| No change   | 50%         | 54%                  |
| Slight High | 18%         | 21%                  |
| Very high   | 3%          | 3%                   |

Figure 42: Table 9 :

10

| Quality     | This Survey | Ambler (2007) survey |
|-------------|-------------|----------------------|
| Very low    | 5%          | 1%                   |
| Slight low  | 10%         | 2%                   |
| No change   | 39%         | 31%                  |
| Slight High | 33%         | 47%                  |
| Very High   | 13%         | 19%                  |

Figure 43: Table 10 :

---

Small scale Medium scale Large scale

Heavy Documentation  
 Comprehensive Upfront Planning  
 Autocratic management Style  
 Not able to change

17. How would you think the agile approaches affect cost of the three sizes of software projects than traditional methodologies?

Small scale Medium scale Large scale

Very high  
 Slightly high  
 No change  
 Slightly low  
 Very low

18. Small scale Medium scale Large scale

Very high  
 Slightly high  
 No change  
 Slightly low  
 Very low

19. To what extent do you follow agile techniques for the three sizes of projects?

Small scale Medium scale

100% follow all agile techniques  
 75% follow all agile techniques  
 50% follow all agile techniques  
 25% follow all agile techniques

20. Which methodology do you prefer for different software projects?

Small scale Medium scale Large scale

Agile methodologies  
 Traditional methodologies  
 Other (Please specify):

*[Note: Optional If you would like to have a summary of the survey results, please provide contact details Name: \_\_\_\_\_ Email: \_\_\_\_\_ Organisa- tion: \_\_\_\_\_ Thank you for all your valuable time in completing this questionnaire.]*

Figure 44:



## .1 Acknowledgement

I would like to acknowledge and state my total pleasure to those whose efforts and support in one way or the other contributed to the successful completion of this research study. I particularly wish to convey profound gratitude to my supervisor, Dr. Nigel Kermode, for his invaluable support, encouragement, useful suggestions and supervision throughout this research work. I also wish to thank my friend Mr. Khondker Razeeb-us-Saleheen, my class mates and my brother, Mr. Channa punchihewa for his support during my study. I am as ever, especially indebted to my parents, Mr. and Mrs. Rahman for their love and support throughout my life. Moreover, to my Almighty who made all the things possible.

Appendix I: Questionnaire The objective of this survey is to find out various methods been used in the software industry for software development. The data collected will be strictly confidential and will only be used for this academic research. Please share your views about your experiences and your personal opinions. If you require a summary of the findings please complete the optional section at the end of this questionnaire.

The questionnaire is divided in to three sections. The questions contained are all close end questions. But if you have any comments for any of the questions please include them with the questions.

For any questions or clarifications please contact me, A.K.M Zahidul Islam [akmzahidulislam102@gmail.com](mailto:akmzahidulislam102@gmail.com)

[Mcneil and Chapman ()] , P Mcneil , S Chapman . 1985. (Research Methods)

[Kerlinger and Lee ()] , F Kerlinger , H Lee . *Foundations of Behavioral Research* 2000. International Thomson Publishing.

[Lindvall et al. (ed.) ()] , M Lindvall , V Basili , B Boehm , P Costa , K Dangle , F Shull , R Tesoriero , L Williams , M Zelkowitz1 . *Agile Methods*, D. Wells and L. Williams (ed.) 2002. p. . XP/Agile Universe

[Saunders et al. ()] , M Saunders , P Lewis , A Thornhill . 2003. Prentice Hall. (3rd. Research Methods for Business Students)

[Dekkers ()] , R Dekkers . *International Journal of Networking and Virtual Organizations* 2005. 3 p. .

[Davis and Sitaram ()] ‘A concurrent process model of software development’. A Davis , P Sitaram . *ACM sigsoft. Software Engineering Notes* 1994. 19 (2) .

[Cockburn ()] *A Methodology Per Project*, A Cockburn . <http://alistair.cockburn.us/Methodology+per+project> 2008.

[Heckman and Williams] ‘A Model Building Process for Identifying Actionable Static Analysis Alerts’. S Heckman , L Williams . *2nd IEEE International Conference on Software Testing, Verification, and Validation*, (Denver, CO, USA) (to appear)

[Coffin and Lane ()] *A Practical Guide to Seven Agile Methodologies*, R Coffin , D Lane . <http://www.devx.com/architect/Article/32836/1954> 2007.

[Boehm ()] ‘A Spiral Model of Software Development and Enhancement’. B W Boehm . *Computer* 1988. (IEEE software)

[Khan and Balbo] *A Tale of two Methodologies: Heavyweight versus Agile*, A Khan , S Balbo . <http://ausweb.scu.edu.au/aw04/papers/edited/balbo/>

[Ambler ()] *Agile Adoption Survey*, S Ambler . <http://www.amblysoft.com/surveys/agileFebruary2008.html> 2008.

[Larman ()] *Agile and iterative development: a manager’s guide*, C Larman . 2004. Addison-Wesley.

[Mccauley ()] ‘Agile Development Methods, Poised to Upset Status Quo’. R Mccauley . *SIGCSE Bullentin* 2001. 33 (4) p. .

[Langr ()] *Agile Java?: Crafting Code with Test-Driven Development*, J Langr . 2006.

[Agile Manifesto ()] *Agile Manifesto*, <http://www.agilealliance.org/> 2002. (Agile Alliance)

[Rakitin (2005)] *Agile Methods -Beyond the Hype. Food for Thought newsletter from Software Quality 22. Consulting*, S Rakitin . <http://www.swqual.com/newsletter/vol2/no7/vol2no7.html#article> 2005. July/August 2005. 04/05/2007. 2.

[Cockburn ()] *Agile software development*, A Cockburn . 2001. Boston, MA: Addison-Wesley.

[Highsmith ()] *Agile software development*, Cockburn Highsmith , A . 2001. Boston, MA: Addison-Wesley.

[Kalermo and Rissanen ()] *Agile software development in theory and practice*, J Kalermo , J Rissanen . 2002.

[Abrahamson (2001)] *Agile software development methods -review and analysis*, Abrahamson . <http://www.sdmagazine.com/documents/s=844/sdm0108a/0108a.htm> 2002. 2001. August. (The Agile Manifesto. Software Development Magazine)

[Schwaber and Beedle ()] *Agile Software Development with SCRUM*, K Schwaber , A Beedle . 2002. UpperSaddle River, NJ: Prentice-Hall.

[Highsmith and Cockburn ()] ‘Agile Software Development: the Business of Innovation’. J Highsmith , A Cockburn . *IEEEComputer* 2001. 34 (9) p. .

- 968 [Boehm and Turner ( )] B Boehm , R Turner . *People Factors in Software Management: Lessons From Comparing*  
969 *Agile and Plan-Driven Methods. CROSSTALK; the Journal of Defense Software Engineering*, 2003.
- 970 [Cohn ( )] *Cited by Planning Poker in Detail*, M Cohn . <http://www.planningpoker.com/detail.html>  
971 2005.
- 972 [Robinson ( )] *Consider DSDM as an XP alternative*, S Robinson . [http://articles.techrepublic.com.com/5100-10878\\_11-1049982.html](http://articles.techrepublic.com.com/5100-10878_11-1049982.html) 2002.  
973
- 974 [Charette ( )] *Cutter consortium e-Project management Advisory service*, R Charette . 2001. 2 p. 19. (The decision  
975 is in: Agile versus heavy methodologies)
- 976 [Ashford and Ehn ( )] *Dynamic Systems Development Method, version 3*, Eng Ashford  
977 DSDM Consoritum. , P Ehn  
978 DSDM Consoritum. . 2008. 1989. Lawrence Erlbaum, Hillsdale, NJ. (Work-oriented design of computer  
979 artifacts)
- 980 [Beck ( )] 'Embracing change with Extreme Programming'. K Beck . *IEEE Computer* 1999. 13 p. .
- 981 [Beck ( )] *Extreme Programming Explained: Embrace Change*, K Beck . 2005. Addison-Wesley. (Second ed.  
982 Reading)
- 983 [Beck ( )] *Extreme Programming Explained: Embrace Change. Reading*, K Beck . 2000. Addison-Wesley.
- 984 [Paulk ( )] *Extreme programming from a CMM perspective. Software*, 18, 6, M C Paulk . 2001. p. .
- 985 [Ghosh and Chopra ( )] B N Ghosh , P K Chopra . *Dictionary of Research Method*, (Leeds, UK) 2003.
- 986 [Vaihansky et al. ( )] *Hyperproductivity In Large Projects Though Distributed Scrum. Agile journal*, P Vaihansky  
987 , J Sutherland , A Victorov . <http://www.offshoreagile.com/news/?doc=205> 2006.
- 988 [Avison and Fitzgerald ( )] *Information Systems Development: Methodologies, Techniques and Tools*, D E Avison  
989 , G Fitzgerald . 1995. McGraw-Hill.
- 990 [Royce] 'Managing the Development of Large Software Systems'. W W Royce . *IEEE Western Conference*  
991 *(Wescon). Pages*, p. .
- 992 [Rakitin ( )] 'Manifesto Elicits Cynicism'. S Rakitin . *IEEE Computer* 2001. 34 (12) p. 4.
- 993 [Renhui and Fengyong ( )] 'Multi-Attribute Evaluation for Software Project Risk'. L Renhui , Z Fengyong . *IEEE*  
994 *Xplore* 2007. p. .
- 995 [Booch ( )] *Object Oriented Analysis and Design with Applications*, G Booch . 1994. Addison Wesley Longman.  
996 (2nd edition)
- 997 [Parekh ( )] N Parekh . <http://www.buzzle.com/editorials/1-13-2005-64082.asp> *Spiral Model -A*  
998 *New Approach towards Software Development*, 2005b.
- 999 [Patzer ( )] Gordon L Patzer . *Experiment-Research Methodology in Marketing. Types and Applications, Quorum*  
1000 *Books*, (Westport) 1996.
- 1001 [Maylor and Blackmon ( )] *Researching Business and Management. Hampshire and 57*, H Maylor , K Blackmon  
1002 . 2005. New York: Palgrave MacMillan.
- 1003 [Peters and Pedrycz ( )] *Software Engineering: An Engineering Approach*, J F Peters , W Pedrycz . 2000. John  
1004 Wiley & Sons.
- 1005 [Lijek ( )] *Software life cycle models in the defense industry*, S Lijek . 2007.
- 1006 [Georgiadou ( )] 'Software Process and Product Improvement, A Historical Perspective'. E Georgiadou . *Volume1,*  
1007 *No1. Pages*, 2003. p. .
- 1008 [Miller ( )] 'The Characteristics of Agile Software process'. G G Miller . *The 39th International Conference of*  
1009 *Object Oriented Languages and Systems*, 2001.
- 1010 [Kim ; Nandhakumar and Avison ( )] 'The Fiction of Methodological Development: A Field Study of Information  
1011 Systems Development'. Moonzoo Kim ; Nandhakumar , J Avison , DE . *Information Technology & People*  
1012 2007. 1999. 12 (2) p. . (Agile development 28)
- 1013 [Takeuchi and Nonaka ( )] 'The New New Product Development Game'. H Takeuchi , I Nonaka . *Comptuer.*  
1014 *Harvard Business Review* 1986.
- 1015 [Devdaily] *The rational unified process*, Devdaily . [http://www.devdaily.com/java/java\\_oo/node19.shtml](http://www.devdaily.com/java/java_oo/node19.shtml)  
1016
- 1017 [Parekh ( )] *The waterfall model explained*, N Parekh . [http://www.buzzle.com/editorials/](http://www.buzzle.com/editorials/1-5-2005-63768.asp)  
1018 [1-5-2005-63768.asp](http://www.buzzle.com/editorials/1-5-2005-63768.asp) 2005a.
- 1019 [Williams ( )] 'The XP Programmer: The Few Minutes Programmer'. L Williams . *IEEE Software* 2003. 20 (3) .
- 1020 [Krasner ( )] 'Using the Cost of Quality Approach for Software'. H Krasner . *Computer. CROSSTALK; the Journal*  
1021 *of Defence Software Engineering* 1998.



- 1022 [Clifton and Dunlap ()] *What is DSDM*, M Clifton , J Dunlap . <http://www.codeproject.com> 2003a.
- 1023 [Clifton and Dunlap ()] *What is SCRUM*, M Clifton , J Dunlap . <http://www.codeproject.com> 2003b.
- 1024 [Schwaber ()] *What is Scrum*, K Schwaber . <http://www.scrumalliance.org/resources/227> 2007.
- 1025 [Sliwa ()] *XP, Scrum join forces*, C Sliwa . [http://www.computerworld.com/action/article.do?](http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=69183)  
1026 [command=viewArticleBasic&articleId=69183](http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=69183) 2002.