# OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES

A thesis submitted to the

*University of Petroleum and Energy Studies*

For the award of

***Doctor of Philosophy***

in

Computer Science

BY

Sunil Kaushik

September 2019

SUPERVISORS

Dr. Vinay Avasthi
Dr. Ashish Bharadwaj
Dr. Rajiv Sharma

School of Computer Science

University of Petroleum & Energy Studies
Dehradun-248007: Uttarakhand

**SEPTEMBER 2019**

## DECLARATION

I declare that the thesis entitled "OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES" has been prepared by me under the guidance of Dr Vinay Avasthi, Associate Professor of School of Computer Science, University of Petroleum & Energy Studies ; Dr. Ashish Bharadwaj, Director IT & Digital Innovation, BML Munjal University and Dr. Rajiv Sharma, Head, Technology Mission Division, Dept. of Science and Technology, Govt. of India .No part of this thesis has formed the basis for the award of any degree or fellowship previously.

Sunil Kaushik

Sunil Kaushik
School of Computer Science
University of Petroleum & Energy Studies,
Dehradun – 248007

Date : 30/9/2019

UPES
UNIVERSITY WITH A PURPOSE

# CERTIFICATE

I certify that Sunil Kaushik has prepared his thesis entitled "**OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES**", for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

30/9/2019

Internal Supervisor : Dr Vinay Avasthi ,
Department Name : School of Computer Science
University Name : University of Petroleum & Energy Studies
University Address : Dehradun
Date :

# CERTIFICATE

I certify that Sunil Kaushik has prepared his thesis entitled **"OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES"**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

External Supervisor : Dr Ashish Bharadwaj, Director – IT, Digital & Innovation
Department Name : IT and Digital Innovation
University Name : BML Munjal University
University Address : Gurgaon.
Date : 20-09-2019

सत्यमेव जयते

Telegram : Scienctech
दूरभाष/Tel. : 26962819, 26567373,
: 26562134, 26562122 (EPBAX)
फैक्स/Fax : 26569908, 26864570,
26863847, 26515637

भारत सरकार
विज्ञान और प्रौद्योगिकी मंत्रालय,
विज्ञान और प्रौद्योगिकी विभाग,
टेक्नोलॉजी भवन, महरौली मार्ग,
नई दिल्ली-110016

GOVERNMENT OF INDIA
MINISTRY OF SCIENCE AND TECHNOLOGY,
DEPARTMENT OF SCIENCE AND TECHNOLOGY,
TECHNOLOGY BHAVAN, NEW MEHRAULI ROAD,
NEW DELHI-110016

## CERTIFICATE

I certify that Mr. Sunil Kaushik has prepared his thesis entitled **"OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES"**, for the award of Ph.D degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

**(Rajiv Sharma)**

डॉ. राजीव शर्मा /Dr. Rajiv Sharma
वैज्ञानिक 'जी' तथा सलाहकार/Scientist 'G' & Advisor
विज्ञान और प्रौद्योगिकी विभाग
Deptt. of Science & Technology
विज्ञान और प्रौद्योगिकी मंत्रालय
Ministry of Science & Technology
भारत सरकार/Govt. of India
नई दिल्ली-110016/New Delhi-110016

# OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES

A thesis submitted to the
*University of Petroleum and Energy Studies*

For the award of
*Doctor of Philosophy*
in
Computer Science

BY
Sunil Kaushik
(SAP ID 500024916)

September 2019

Internal Supervisor
Dr. Vinay Avasthi
Associate Professor
School of Computer Science
University of Petroleum & Energy Studies

External Supervisor
Dr. Ashish Bharadwaj
Director - IT, Digital & Innovation
BML Munjal University

Dr. Rajiv Sharma
Head, Technology Mission Division,
Dept. Of Science & Technology, Govt. Of India

UPES
UNIVERSITY WITH A PURPOSE

School of Computer Science
University of Petroleum & Energy Studies
Dehradun-248007: Uttarakhand

**SEPTEMBER 2019**

# DECLARATION

I declare that the thesis entitled "OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES" has been prepared by me under the guidance of Dr Vinay Avasthi, Associate Professor of School of Computer Science, University of Petroleum & Energy Studies ; Dr. Ashish Bharadwaj, Director IT & Digital Innovation, BML Munjal University and Dr. Rajiv Sharma, Head, Technology Mission Division, Dept. of Science and Technology, Govt. of India .No part of this thesis has formed the basis for the award of any degree or fellowship previously.

Sunil Kaushik
School of Computer Science
University of Petroleum & Energy Studies,
Dehradun – 248007

Date :

# CERTIFICATE

I certify that Sunil Kaushik has prepared his thesis entitled "**OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES"**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

Internal Supervisor : Dr Vinay Avasthi ,
Department Name : School of Computer Science
University Name : University of Petroleum & Energy Studies
University Address : Dehradun
Date :

# CERTIFICATE

I certify that Sunil Kaushik has prepared his thesis entitled "**OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES"**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

External Supervisor : Dr Ashish Bharadwaj, Director – IT and Digital Innovation

Department Name : IT and Digital Innovation

University Name : BML Munjal University

University Address : Gurgaon.

Date :

# CERTIFICATE

I certify that Sunil Kaushik has prepared his thesis entitled "OPTIMIZING ERP IMPLEMENTATION AND CUSTOMIZATION WITH LEAN THINKING AND AGILE METHODOLOGIES**"**, for the award of PhD degree of the University of Petroleum & Energy Studies, under my guidance. He has carried out the work at the School of Computer Science, University of Petroleum & Energy Studies.

External Supervisor : Dr Rajiv Sharma, Scientist 'G' and Head
Department Name : Department of Science and Technology
Organization Name : Government of India
Address : New Delhi.
Date :

# ABSTRACT


## 1. INTRODUCTION

With the increase in global competition, enterprises are looking for a solution to have a seamless integration of all the departments providing information from one source of system. Enterprise Resource Planning (ERP) systems fit the context and help organizations to take informed decisions. ERP systems have ability to raise the productivity and profitability of business by leaps and bounds. ERP systems are complex in nature and it is time consuming and resource intensive to implement ERP. Unfortunately, the failure of ERP implementation is widespread across industries and geographies. The previous studies show that around 60% to 70% of ERP implementations end up in a failure or meet their end prematurely. Success and failure factors have been studied for successful ERP implementation. There are various implementation models to ensure success of ERP implementation. However, these models are inflicted with lack of feedback mechanism and 'release in one go approach'. This research presents a new model of ERP implementation (using Lean Thinking and Agile Methodologies) to ensure success of ERP implementation.

## 2. OBECTIVE OF THE STUDY

The objective of the research is divided into two categories

**Primary Objective**

- To create a framework which is a hybrid of Agile Methodologies and Lean Thinking which would provide an efficient way of ERP implementation and customizations by minimizing the total effort including the efforts elapsed in various testing, cost of quality, testing effort and duration and reducing defects and defect injection rate (DIR) thereby increasing quality.

**Secondary Objective**

- To explore the Lean and Agile Methodologies from software development perspective.
- To investigate the impact of Lean Thinking and Agile Methodologies on enhancing productivity and quality of ERP Implementation and Customization project.
- To prove the validity of the framework by applying it to an ERP Implementation and Customization project and comparing the following parameters with a project of similar complexity and size.
    - Effort
    - Effort elapsed in testing phases
    - Cost of Quality (COQ)
    - Total Defects
    - Defect Injection Rate (DIR)
- To create an algorithm to prioritize the requirements in an ERP implementation.

## 3. RESEARCH METHODOLOGY

The research has adopted the Hybrid Research methodology and was carried out in following phases.

- ***Phase 1a: Literature Research and Analysis***
  Several research papers, experience reports on ERP Implementation and Lean and Agile Methodology were studied. This helped in creating solid foundation of new framework.
- ***Phase 1b: Interaction with Industry Experts***
  Experts from industry were contacted to understand the issues and challenges in ERP implementations.

- ***Phase 2: Hybrid framework Development, Validation and Verification***

  A new framework, with roots of Agile Methodologies and Lean Thinking, was made. Framework was validated by applying in five internal projects.

- ***Phase 3: Research Synthesis***

  The results for parameters (Effort, Effort Deviation, Effort elapsed in various testing, Cost of Quality, Total Defects and Defect Injection rate) were compared and validated statistically using t-test. The results were found to be statistically significant.

In summary, this research incrementally adds to the body of knowledge regarding ERP implementations, provides framework and suggestions to practice on better managing ERP projects and throws light on the causal issues, their effect on the project outcomes and the solution to the issues.

# ACKNOWLEDGEMENT

I have read this section of all the PhD theses, that I have come across, to get a glimpse of life and the challenges faced by a research scholar. This section had always been motivating for me and I have always dreamt of writing one for my thesis. This section seems to be the hardest section to write – to acknowledge all those people who have contributed to my research by analyzing, augmenting, criticizing, filled in for me at my workplace and implemented it. Anyways, I am going to try my level best to show my gratitude to every person and if few names are missed, rest assured that my gratitude to you is not less than anyone listed below.

I owe a deep sense of gratitude, indebtedness and reverence for my supervisor Dr. Vinay Avasthi, Associate Professor, Dept. of Computer Science and Engineering, School of Computer Science, University of Petroleum and Energy Studies, Dehradun for his versatility, benevolence, splendid and stimulating guidance, patronage, keen interest and constructive suggestions in planning and execution of the research work and in preparation of this thesis. I have no words to express gratefulness for effort put by him; he made me work and smothered my path in difficulties

With a deep sense of veneration and obligation, I avail this opportunity to express thanks to my supervisor Dr. Ashish Bharadwaj, Director IT & Digital Innovation, BML Munjal University, Gurgaon for his invaluable inspirational mentorship, esteemed guidance, constant encouragement, untiring facilitating efforts, supportive attitude, critical comments, keen interest and constructive suggestions in the endeavor of present research. His expertise combined with his patience has helped and encouraged me to complete the research work in time.

I would like to express my deep and sincere gratitude for my external supervisor Dr. Rajiv Sharma, Head, Technology Mission Division, Dept. of Science and Technology, Govt. of India, New Delhi. His guidance and motivation helped me complete my studies despite the workload.

I am also grateful to Dr. Manish Prateek, Dean and Head, School of Computer Science, University of Petroleum and Energy Studies, Dehradun for providing necessary facilities to accomplish the present work.

The episode of acknowledgement would not be complete without the mention of my best friend and wife Dr. Irshita Kaushik, who did her best to improve my best. I thank her for her patience and loving, unstinted support during my late hours and her spoiled weekends. She was a source of constant encouragement, which boosted my morale in moments of despair and desperation. It would have been impossible for me to complete the research work without her support which kept me motivated and going right from the beginning till the final shape up of this research.

I feel short of words when I express my profound sense of gratitude for my parents and my parents-in-law who have very enthusiastically put their every effort to shape my life. I cannot record and measure their contribution in any conceivable words. I feel great pleasure in expressing my gratitude to my family members for their everlasting affection and constant encouragement. My loving thanks and apologies are due to my kids – Advait and Aryahi for not giving proper time to them.

I also express sincere thanks to my teammates and clients at Infosys, WIPRO for extending a helping hand, whenever I was in need. Thank you all for contributing to making all this possible.

The errors, idiocies, and inconsistencies remain my own, which I take absolute responsibility, as if customary, while hoping that the rest of material will be enough to stimulate insights and new trains of thought into the ever-inspiring field of software engineering.

I could complete my work within stipulated time by grace of God. So, I dedicate this work at his Lotus feet…

**Gurgaon** **Sunil Kaushik**

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF APPENDICES

# LIST OF ABBREVIATIONS

AIM – Oracle Application Implementation Methodology

ASAP – Accelerated SAP Implementation Methodology

BA – Business Analysis / Analyst

BI – Business Intelligence

CD – Construction and Configuration Phase

CDV – Validation and Verification of Construction and Configuration Phase

CM – Configuration Management

COQ – Cost of Quality

COTS - Commercial Off-The-Shelf

DCR – Defect Containment Ratio

DD – Design or Blueprint phase

DDV - Validation and Verification of Design or Blueprint phase

DIR – Defect Injection Rate

DIST – Distribution

ERP - Enterprise Resource Planning System

FPN - Feature Priority Number

IS - Information System

MRP - Material Requirements Planning

NFR – Non-Functional Requirements

OUM – Oracle Unified Methodology

PPN - Process Priority Number

QA – Quality Assurance

RS – Requirement Specification

RS - Validation and Verification of Requirement Specification Phase

SIT – System Integration Testing

SPN - Sub Process Priority Number

TCO – Total Cost of Operations

TPS – Toyota Production System

UAT – User Acceptance Testing

VSM - Value Stream Mapping

V&V – Validation and Verification

WIP – Work in Progress

# CHAPTER 1: INTRODUCTION

## 1.1 INTRODUCTION

Enterprise Resource Planning System (ERP) is an integral part of business and it plays a vital role in the functioning of various verticals of an organization [120]. Every function of a modern organization is heavily dependent on information and software such as ERP which provides cutting edge to the organization [132]. ERP is a single program that caters to all functions of an organization. ERP integrates the information of all the functions of organization and consolidates it to make informed decisions [40]. Central idea of ERP implementation is to improve efficiency, to enable organization to weed out duplicate and useless information, to take strategic decisions faster [48,26], to get tangible benefits and to achieve predefined business objectives. ERP software is a costly software system and a big chunk of corporate IT expenditure budget is spent towards it [30,128]. Costly software, huge effort in implementation and heavy change-management process make the ERP implementation critical. Thus, it becomes imperative to ERP Implementation to deliver value in a reasonably short time to justify the investment made [50]. Worldwide IT spending is expected to reach $3.7 trillion in 2018 and the enterprise application software market is expected to reach $391 billion and $424 billion in 2018 and 2019 respectively [157]. The huge amount at stake expects to innovate the ERP implementation strategy.

## 1.2 MOTIVATION

ERP failure report 2017 by Panorama Consulting suggests that ERP systems have low success rates and they fail to deliver up to the expected benchmark. The reason behind this failure can be attributed to technological failure, change-management failure of the organization or high expectations from ERP [160,141]. Studies have shown that organizations are finding it difficult to implement ERP and to reap the benefits. It is seen that less than 10% of ERP projects have spent within allocated budget and are completed within time schedule for ERP implementation. Also, close to one fifth of the ERP implementation projects are unsuccessful [10,37,69,85,93] and less than one third of the projects can be called successful ERP implementation [53,112]. ERP Failure Report of 2017 [160] hints that more than 37% of the companies failed to realize 50% benefit of ERP, 74% implementations have run over budget and 59% have exceeded the planned and scheduled duration.

It is the seen in figure 1.1 [160] that for the past seven years, approximately 60% of the projects have reported cost over, 62% projects have overshot the original schedule and approximately 50% even failed to realize 50% benefit of ERP.

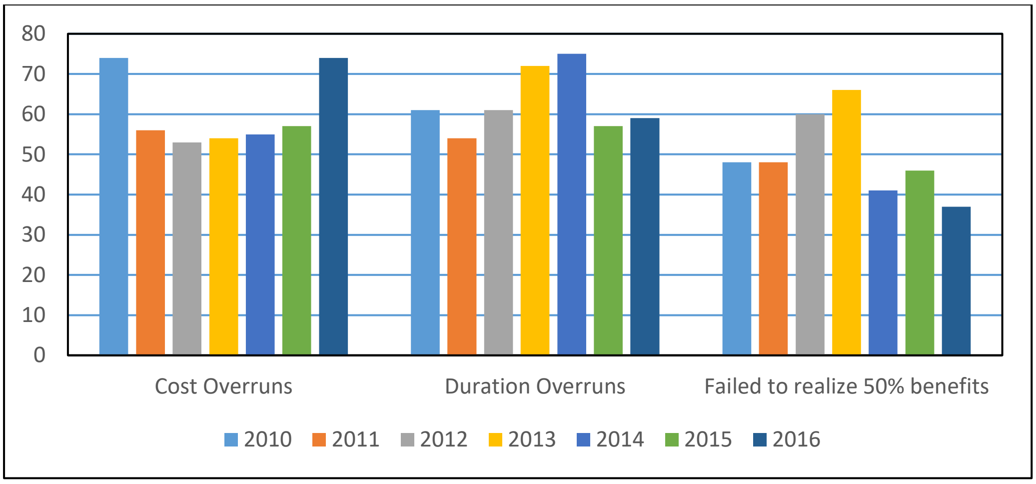


**Figure 1.1: ERP Failure Report 2017.**

There have been various studies to find out reasons for unsuccessful ERP implementation. Few significant reasons include cross functional coordination, ERP configuration and features, change management, human resources and capabilities, complexity of the package, organizational leadership etc. [37,38,56,71,11]. In technical-led approaches, project management and implementation strategies are often overlooked or even ignored but they are a vital component of ERP projects [36,13]. The current failure rate of ERP and the pressure on organizations to deliver the value at low cost have called for the need of optimizing the ERP implementation and customization process. Fact or Fiction Report [158] suggests that the challenges in ERP Implementation can be overcome by adopting Lean Thinking and Agile Methodologies.

Every organization strives to optimize its operations by improving quality and cutting costs by applying traditional or new management techniques. Recently, Agile methodologies are regarded as the solution for such concerns. The "Chaos Report" [135] in Figure 1.2 shows that approximately 10% of agile projects fail as compared to 30%, approximately, of traditional projects. Same suggestion is conveyed by survey conducted by IT Project Survey of 2013 [162] which is depicted in Figure 1.3.

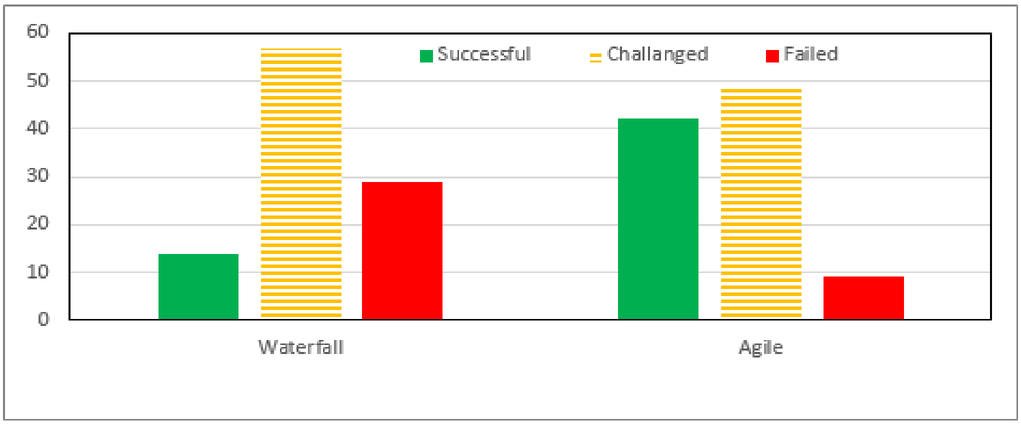


**Figure 1.2: Comparison of Agile and Traditional Methodologies - Chaos.**

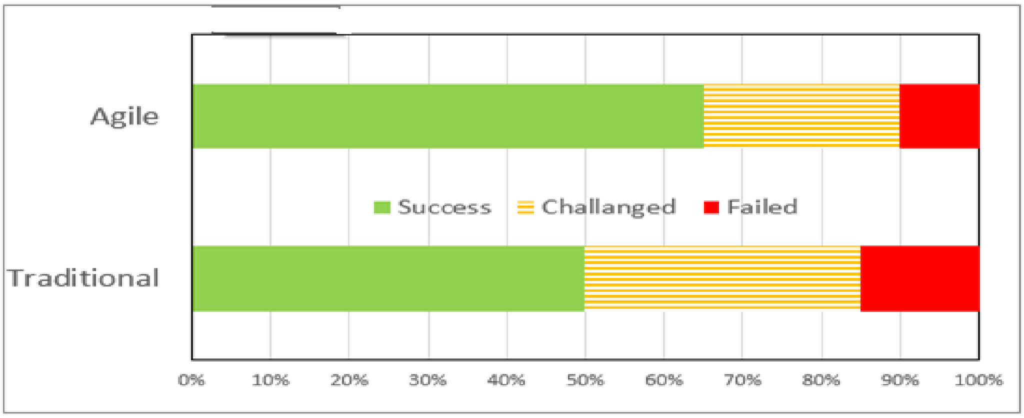


**Figure 1.3: Comparison of Agile and Traditional Methodologies - IT Project Survey**

Agile methodologies are more adaptive than predictive and embrace change as an opportunity for improvement [73,74]. It is seen that Agile Methodologies are well accepted by the Software Development Community because they are able to align the business needs and software development [46]. Few researchers

[125,44,22,69,93] suggest that a good extent of research has been done on using Agile for ERP but these researches lack practicality by assuming ERP deployment as ERP implementation. Also, ERP customizations are poorly subjected. It is noteworthy to point out that the customizations range from parameterization or configurations to code change which also includes updation or creation of a new module. Customizations by code change are very well accepted in ERP implementation consulting community and organizations that are implementing ERP [103]. Hence, ERP customizations are right candidates for application of Agile principles.

Lean thinking, often called as Lean, is a viewpoint of yielding maximum value and minimum waste [3,41,130]. IT and Manufacturing are analogous [63] to each other. *Sutton* [133] pointed out that Lean can be a good middle ground between craft development and "software factories", analyzing a real project in which use of lean techniques lead to both quality improvement and cost reduction. Later, *Raman* [113] argued that Lean Software Development is a feasible methodology for software development.

Lean and Agile are two entirely different paradigms. Lean emphasizes on waste reduction while Agile stresses to develop flexibility and adaptability to changing environment [70]. Despite the differences, Agile and Lean are mutually supportive [97,143]. Lean aligns with many Agile principles but considers a more holistic enterprise perspective [103,72]. Software industry has already been using Agile Methodologies and, recently, has shown an interest to apply Lean thinking to software development. Agility as a concept is closely related to Lean and a combination of Agile practices and Lean has proved to be efficacious [145,71,55]. However, any software development framework hybird of Lean Thinking and Agile Metholdologies is still an unexplored territory [146,154]. There is no literature available till date that talks about the application of any framework blending Lean Thinking coupled with Agile Software Development principle in ERP implementations and customizations, which always overrun schedule, budget and are unable to deliver the perceived value to customer. This study will help to understand the applicability of Lean Thinking and Agile Methodologies in ERP implementaton and customizations, impact of applying this combination to manage the total effort which is defined as amount of person hours required / spent to develop, review, test and deploy any software [62], effort elapsed in testing phases which also includes the effort elapsed in fixing the defects discovered in testing is also included in these phases. Cost of Quality (COQ) i.e. the total costs incurred on quality activities and issues are often split into prevention costs, appraisal costs, internal failure costs and external failure costs [1,12], total defects suggesting success and failure of ERP from various point of views and defect injection rate (DIR) defined as number of defects discovered for a phase of the software life cycle per 100 hours of effort [60,95]. Some past researches [57,142] suggest that Agile Methodologies should decrease software COQ but these researches don't produce any quantifiable result. *Noor and Khan* [98] reviewed various researches and concluded that Agile model is best suited for defect management. But, no past research has been done to study the effect of Lean thinking on software cost of quality. They also agree that more work is required to diminish defects at early stages. In addition, no past research has been done to study the effect of Agile or Lean thinking on Software defect injection.

Agile Methodologies and Lean Thinking are discussed in detail and compared in Chapter 2 – Review of Literature.

## 1.3 OBJECTIVE

The objective of the research is divided into two categories

**Primary Objective**

- To create a framework which is a hybrid of Agile Methodologies and Lean Thinking which would provide an efficient way of ERP implementation and customizations by minimizing the total effort including the efforts elapsed in various testing, cost of quality, testing effort and duration and reducing defects and defect injection rate (DIR) thereby increasing quality.

**Secondary Objective**

- To explore the Lean and Agile Methodologies from software development perspective.

- To investigate impact of Lean Thinking and Agile Methodologies on enhancing productivity and quality of ERP Implementation and Customization project.
- To prove the validity of the framework by applying it to an ERP Implementation and Customization project and comparing following parameters with a project of similar complexity and size. Also, the deviation from planned value was measured for each parameter.
    - Effort
    - Effort elapsed in testing phases
    - The total costs incurred on quality activities - cost of quality (COQ)
    - Total Defects
    - Number of defects discovered for a phase of the software life cycle per 100 hours of effort - defect injection rate (DIR)
- To create an algorithm to prioritize the requirements in an ERP implementation.

To achieve the above said objectives, a research methodology was framed, and hypotheses were made which are given in section 1.5 and 1.4 respectively.

## 1.4 HYPOTHESES

Following sets of hypotheses were made to validate the benefits of new hybrid framework (here after called as GenNext framework) on the parameters defined in the objective section of this thesis (section 1.3). Each set of hypotheses is given a unique ID and these ID are used to refer the set of hypotheses in the thesis. The projects executed with GenNext framework are referred as GenNext projects and projects executed with traditional methodology are referred as Traditional projects in the rest of the thesis.

### 1.4.1 TO VERIFY SAVING IN EFFORT

Set of hypotheses were made for each phase or activity and total effort for implementations. Set of hypotheses are given below in table 1.1 and are tested with data in Chapter 5 – Results and Discussions.

| ID | Activity | Hypotheses |
|---|---|---|
| HYP1 | Total Effort | *$H1_0$: GenNext projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1_a$: GenNext projects consume* ***less effort*** *than traditional projects.* |
| HYP1A | Requirement Specification | *$H1A_0$: Requirement Specification activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1A_a$: Requirement Specification activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |
| HYP1B | Design | *$H1B_0$: Design activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1B_a$: Design activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |
| HYP1C | Construction and Configuration | *$H1C_0$: Construction and Configuration activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1C_a$: Construction and Configuration activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |
| HYP1D | Validation and Verification of Requirement Specification | *$H1D_0$: Validation and Verification of RS activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1D_a$: Validation and Verification RS activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |
| HYP1E | Validation and Verification of Design | *$H1E_0$: Validation and Verification of design activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1E_a$: Validation and Verification of design activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |
| HYP1F | Validation & Verification of Construction and Configuration | *$H1F_0$: Validation and Verification of development activities in GenNext Projects consume* ***more or equal*** *effort than traditional projects.*<br>*$H1F_a$: Validation and Verification of development activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |
| HYP1G | SIT | *$H1G_0$: SIT activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*<br>*$H1G_a$: SIT activities in GenNext Projects consume* ***less effort*** *than traditional projects.* |

| ID | Activity | Hypotheses |
|---|---|---|
| HYP1H | UAT | *$H1H_0$: UAT activities in GenNext Projects consume **more or equal effort** than traditional projects.*<br>*$H1H_a$: UAT activities in GenNext Projects consume **less effort** than traditional projects.* |

**Table 1.1: Set of hypotheses to verify saving in effort by GenNext framework.**

### 1.4.2 TO VERIFY SAVING IN COQ AND INCREASE IN QUALITY

GenNext framework should be able to save the COQ of ERP implementation project. GenNext framework should be able to deliver high quality of ERP Implementations at a lower cost i.e. ERP Implementation projects executed using GenNext framework should have less number of defects and COQ. Statistical representation is given below in table 1.2.

| ID | Activity | Hypotheses |
|---|---|---|
| HYP2 | COQ | *$H2_0$: GenNext projects consume **more or equal COQ** than traditional projects.*<br>*$H2_a$: GenNext projects consume **less COQ** than traditional projects.* |
| HYP3 | Quality | *$H3_0$: GenNext projects show **lesser quality** than traditional projects.*<br>*H3a: GenNext projects show **better or equal quality** than traditional projects.* |

**Table 1.2: Set of hypotheses to verify saving in COQ and effect of quality.**

### 1.4.3 TO VERIFY EFFECT OF DIR

GenNext framework should not only be able to deliver high quality of ERP Implementations i.e. ERP Implementation projects executed using GenNext framework should have less number of defects but also these implementations should have less number of defects per 100 Person Hours. Statistical representation is given below in table 1.3.

| ID | Activity | Hypotheses |
|---|---|---|
| HYP4 | DIR | *$H4_0$: GenNext projects have* ***higher or equal DIR*** *than traditional projects.*<br>*$H4_a$: GenNext projects have* ***lesser DIR*** *than traditional projects.* |

**Table 1.3: Set of hypotheses to verify reduction in DIR.**

All these hypotheses are validated, and results are concluded in Chapter -5: Results and Discussions of the thesis.

## 1.5 RESEARCH METHODOLOGY

The research has adopted the Hybrid Research methodology. In Hybrid Research approach, knowledge claims are based on practical grounds. In this approach, the inquiry strategies help to better understand research problems by gathering both numeric and text information either concurrently or sequentially. Hence the collected data represents both quantitative and qualitative information.

- ***Phase 1a: Literature Research and Analysis***

  This phase includes extensive desk research of the existing ERP implementation methodology including change management technology and benefits of Lean, Agile Software Development methods. The literature research would help in creation of the hybrid framework which consists of Agile Methodologies and Lean Thinking encompassing merits of both methodologies.
- ***Phase 1b: Interaction with Industry Experts***

  This phase includes interaction with Industry experts to know the issues in ERP implementation which cause schedule overrun and cost overrun and increase in COQ. Interaction with Industry experts such as ERP Project Managers, ERP consultant and Client base of ERP users (at all levels), System Integration Consultants / Analysts, Client IT Leads / Managers and project sponsors was done to understand if Lean Thinking and Agile Methodologies can be applied to ERP implementation and customization.

This phase is elaborated in Chapter 2: Literature review of this thesis.

- ***Phase 2: Hybrid framework Development, Validation and Verification***

  This phase includes creation framework hybrid of Agile Methodologies and Lean Thinking and validation of the framework by applying it to an ERP project. Collect the Software Metrics Parameters such as Effort, Effort Deviation, Effort elapsed in various testing, Cost of Quality, Total Defects and Defect Injection rate for the project.

- ***Phase 3: Research Synthesis***

  Compare and interpret the results from the above project and a project of similar complexity to draw the conclusions and implications.

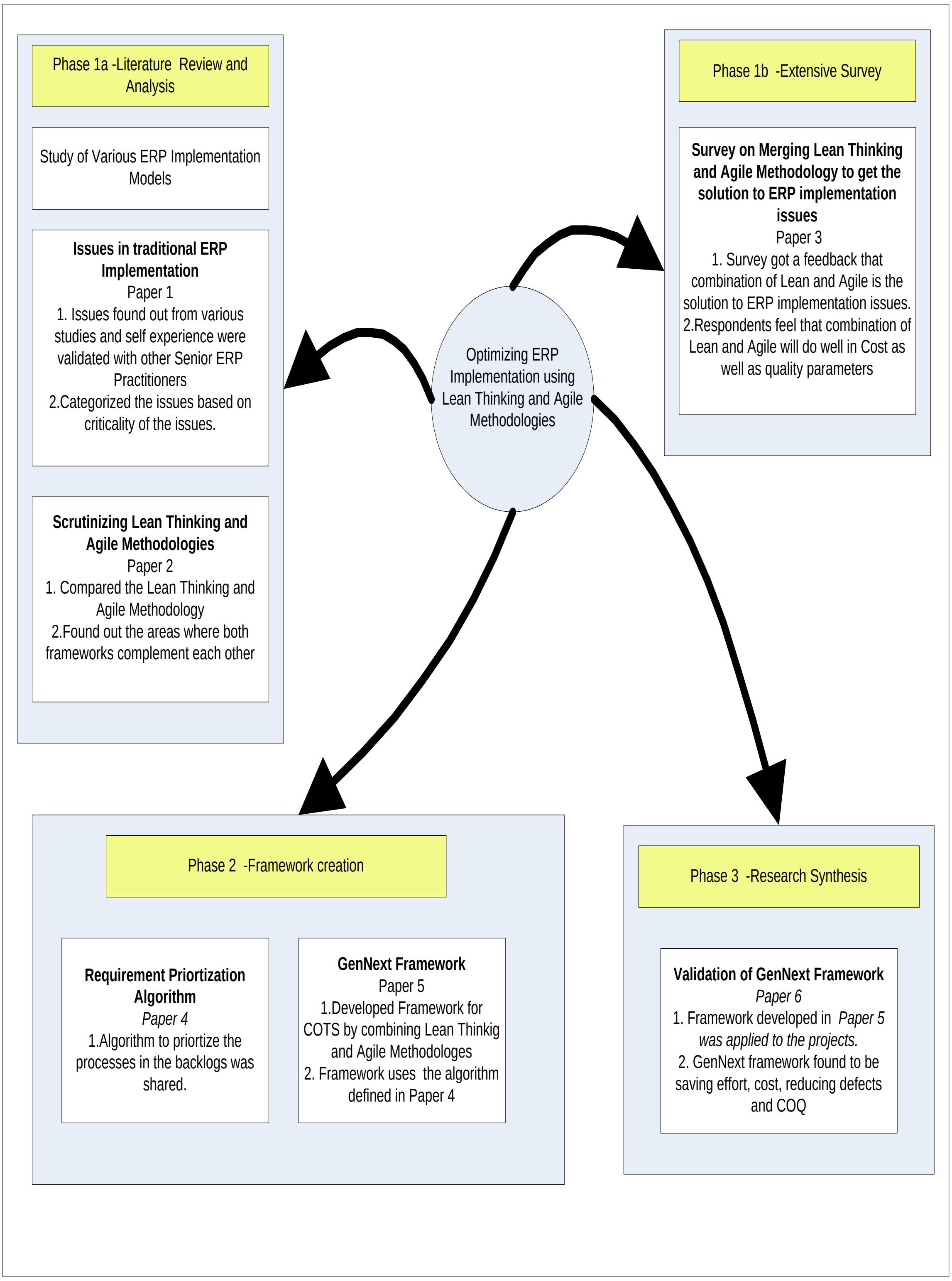


**Figure 1. 4: Research Methodology**

The research methodology is described in detail in Chapter 3- Research Methodology of the thesis.

## 1.6 ORGANIZATION OF THESIS

This thesis is organized into six chapters. A brief outline of the various chapters is given below:

### Chapter 1: Introduction

This chapter gives the background for this research. It explains the concept of enterprise integration through Information Systems (IS). A summarization of the research problem, motivation for research, scope and objectives and a chapter wise summary is provided in this chapter. The contributions of this research to the body of knowledge has been summarized in this chapter.

### Chapter 2: Literature Review

This chapter collates the information from the already published sources. This chapter talks about ERP, evolution of ERP, various models of ERP implementations and comparison of all these models, this chapter also talks about the survey to understand issues in ERP implementation and introduces Lean thinking and Agile methodology as potential solution for issues in ERP implementation and acceptance of Lean Agile. This chapter also highlights the research gap and need of new framework.

### Chapter 3: Research and Methodology

This chapter outlines the development of the conceptual framework leading to defining the research objectives and methodology. The purpose of this research is to develop a methodology to create a new framework and its verification and validations.

### Chapter 4: GenNext Framework

This chapter defines the framework hybrid of Lean Thinking and Agile Methodologies called as GenNext framework. It also defines the principles, values, various roles and practices in GenNext framework.

### Chapter 5: Results and Discussions

This chapter gathers data and compares the vital parameters such as effort, cost, defect and COQ for both kind of projects i.e. executed using GenNext framework and projects executed using traditional methodology. The hypotheses made in section 1.4 are validated and conclusions were drawn based on these hypotheses.

**Chapter 6: Conclusions and Future Scope**

This chapter talks about the benefits and the future scope of the research.

# CHAPTER 2: REVIEW OF LITERATURE

## 2.1 ENTERPRISE RESOURCE PLANNING

### 2.1.1 DEFINING ENTERPRISE RESOURCE PLANNING

Various researchers have tried to provide the definition of Enterprise Resource Planning (ERP). *Chung & Synder* [29] defined ERP as a category referring to alike products under one category and it is also defined as a tool which unifies the process and data to offer a business solution [66]. ERP system is the most innovative development done in any sector [6]. ERP can be thought of as a software that integrates the business functions of various departments and organizations [67]. This definition is an extension of the definition given by *Rooney & Bangert* [116] and is further supported by *Hoch & Dulebohn* [54]. *Marnewick and Labuschagne* [84] and *Aladwani* [4] referred ERP as business software package that integrates the majority of business processes of an organization and automates the system to make all departments, use common data and practice. Hence, ERP not only serves as a platform to enable several departments efficiently and effectively but also provides an environment where in all departments are connected to each other.

*Ross et al.*[119], conveniently using the above definition, defined ERP as a business management system integrating set of comprehensive business intensive software modules. ERP, COTS (Current Off the Shelf) systems are designed to integrate core functions of an enterprise on a unified database regardless of the type, size or nature of the business and ERP is an extension of the manufacturing resource planning (MRP). ERP is a configurable off the shelf software package that integrates systems and information for a range of operational or management activities [25,26,127,112].

For this study, ERP can be defined as application conforming to following points:

- Business management application.
- Consistent and integrated data
- Cost effective and efficient

- Integrates the business process and brings best in class processes.

The definition of ERP can be well understood by knowing the genesis of ERP and benefits provided by ERP. The history of ERP and benefits of ERP are discussed in the section 2.1.2 and 2.1.3 of this thesis.

### 2.1.2 HISTORY OF ENTERPRISE RESOURCE PLANNING

Current Enterprise Resource Planning (ERP) system is the outcome of continuous improvements done for five decades. The continuous improvement was due to management techniques and developments in the field of software and hardware technologies. In 1960's corporates created their in-house software based on the Classical / Scientific Inventory Control Techniques to automate their work and optimize the inventory [17]. Material Requirements Planning (MRP) systems were developed in the 1960s to calculate the material requirement according to master production schedule. MRP systems were further enhanced in 1980s by including the processes to optimize the inventory by simulation and forecasting [17]. 1980-90s global competition, shortened product cycle and customer focus led to the integration of various functional areas of an organization. The requirement for integration called for the requirements of Enterprise Resource Planning (ERP). The ERPs evolved containing all the back-office solutions such as financials, order management and distributions and human resources management system [17] and evolved to Decision Support Systems.

### 2.1.3 BENEFITS OF ENTERPRISE RESOURCE PLANNING

ERP systems are created to handle integrated multiple functions in a complex corporate environment in an efficient way by providing an information enabled environment [7]. ERP implementation has helped the organizations to be customer centric [30,114] and to reap the benefits of standardization, business process reengineering to take advantage of quality standards [151] . *Kennerley & Neely* [65] have listed the benefits of ERP implementation given in table 2.1.

| | |
|---|---|
| Increased Coordination and Control | The decision support systems have unburdened the administration departments and increased interaction between various parties have led to better control and coordination yielding high efficiency. |
| Inventory Optimization | Better control over stocks, assets and consumables. An improvement in the procurement strategy renders better inventory optimization. |
| Increased profitability | Informed decisions and better planning lead to increase in profitability. |
| Better insight to suppliers | Suppliers get the view of bigger picture and this allows them to negotiate better for contracts. |
| Capacity Optimization | Larger view of orders and inventory levels enables organizations to plan the capacity well in advance |

**Table 2.1: Benefits of ERP Implementation**

Further, *Turban et al.* [139] points out the reason of success of ERP and its benefit as ERP provides real time picture of all the transactions happening across organization. ERP systems integrate business functions to provide various tangible and intangible benefits [121]. The intangible benefits are standardization and improved business processes, informed decisions and better visibility of organization. ERP systems have flawlessly integrated the information from an entire enterprise [120]. Also, ERP applications could improve efficiency and effectiveness further by automation [111, 13].

Current ERP systems integrate the whole organization [93] and are a platform for forward integration with customers and backward integration with suppliers. ERP implementation methodology plays a vital role in realizing the perceived benefits of ERP [69,17]. Various ERP implementation methodology or models are discussed in detail in section 2.1.4 of the thesis.

### 2.1.4 ENTERPRISE RESOURCE PLANNING IMPLEMENTATION

ERPs are complex software systems or applications and they encompass various dimensions of the organization [116]. Benefitting every part of the organization is central idea of any ERP implementation. Thus, ERP implementation is a difficult task because of the constraints involved [84]. ERP Implementations are different from software development because both differ in the way they are created [69]. *Boubekri* [23] advises that implementation strategy should focus on strategic business objectives and should consider the complex business integration. He further advocates that implementation approach should consider business requirements from organizational, technical and human perspective. ERP Implementations require all the activities to be broken into parts called phase or stage so that it can be controlled easily [124]. The complexity of ERP Implementation is directly proportional to the number of modules to be implemented [54,117]. Various strategies for successful ERP implementation have been listed by *Aladwani* [4] for successful implementations. These strategies are categorized in table 2.2.

| | |
|---|---|
| Management Strategies | - Project Management<br>- Organizational Structure<br>- Change Management |
| Technical Strategies | - Understanding of system.<br>- Qualified technical and functional resources.<br>- Understanding of business and state of the art processes |
| Human Strategies | - Attitude Management<br>- Inclusion of people in ERP Implementations. |

**Table 2.2: Strategies of successful ERP Implementation.**

*Gefen and Ragowsky* [51] suggest that various business functions should be involved in ERP implementation and *Ara and Al-Mudimigh* [11] advocate to include various management functions and departments at various levels.

ERP implementation projects are large-scale software projects and are often incomplete without out of the box functionalities called customization [78,69]. The end-product is usable only after customizations [14,13,50] have been built and integrated. Customizations could be business process customizations which have low impact on the end-product or system customizations fulfilling the business requirements which have high impact on quality and cost because complete software development processes (requirements analysis, software design, coding and testing) are carried out [103,52]. ERP systems are complex commercial off-the-shelf (COTS) software packages and their fitment into organization's requirement is always a challenge [64,144]. The changes to fit into organization's working style is quite underestimated [141,154]. Organizations tend to fill the gap between ERP and their processes by either altering their business processes or customizing the ERP system by rewriting part of the delivered software or interfacing to an external system [49]. The various types of customizations for a 3 tier or layered architecture ERP are given [24] in Table 2.3.

| Tailoring Type | Description | Examples | Layer Involved |
|---|---|---|---|
| Configuration (customization, in SAP parlance) | Setting of parameters (or tables), in order to choose between different executions of processes and functions in the software package | Define organizational units; create standard reports; formulate available-to-promise logic; use of a standard interface to an archive system | All layers |
| Bolt-ons | Implementation of third-party package designed to work with ERP system and provide industry-specific functionality | Provide ability to track inventory by product dimensions (e.g., 2 500 m. lengths of cable do not equal 1 1000 m. length) | All layers |
| Screen masks | Creating of new screen masks for input and output (soft copy) of data | Integrate three screens into one | Communication layer |
| Extended reporting | Programming of extended data output and reporting options | Design new report with sales revenues for specific criteria | Application layer and/ or database layer |
| Workflow programming | Creating of non-standard workflows | Set up automated engineering change order approval process | Application layer and/ or database layer |
| User exits | Programming of additional software code in an open interface | Develop a statistical function for calculating particular metrics | Application layer and/ or database layer |
| ERP Programming | Programming of additional applications, without changing the source code (using the computer language of the vendor) | Create a program that calculates the phases of the moon for use in production scheduling | All layers |
| Interface development | Programming of interfaces to legacy systems or 3rd party products | Interface with custom-build shop-floor-system or with a CRM package | Application layer and/ or database layer |
| Package code modification | Changing the source-codes ranging from small change to change whole modules | Change error message in warning; modify production planning | Can involve all layers |

**Table 2.3: Typology of ERP Tailoring Type.**

### 2.1.5 ERP IMPLEMENTATION MODELS

Compared to number of researches on finding critical success factors, researches on ERP implementation models are very less [5,93]. Few of the researchers refer technique for ERP implementation as model and few researchers refer this as methodology and few researchers call this as framework. In this thesis, model and methodology are used interchangeably. The models can be categorized into two parts:

1. Methodologies or Models available in academic and scientific community.
2. Models provided by ERP vendors such as Oracle and SAP.

This section describes various models of both categories and compares the models in individual category.

#### 2.1.5.1 ERP IMPLEMENTATION MODEL – BANCROFT MODEL

*Bancroft et al.*,[15] carried out extensive studies and proposed a model for ERP implementation. In the model, they proposed five steps process for ERP implementation. Four of the steps were pre-implementation and one step is actual implementation. The steps are as follows –

1. **Focus Phase** – This phase is also referred as planning phase and all the initiation activities such as formation of various teams and plan is done in this phase.
2. **As-Is phase** – This phase is also referred as analysis phase. The existing processes and business process review is done during this phase.
3. **To-be phase** – This phase is also referred as design phase. Based on the outcome of analysis, new processes are designed at high level and detailed level.
4. **Configuration** – The main activities of this phase is configuring the ERP system and testing the system with near real time data.
5. **Go Live** – The actual roll out of the application happens in this phase only. The users are trained on the new system and other support is provided to the user.

Figure 2.1 depicts the schematic diagram of Bancroft Model.

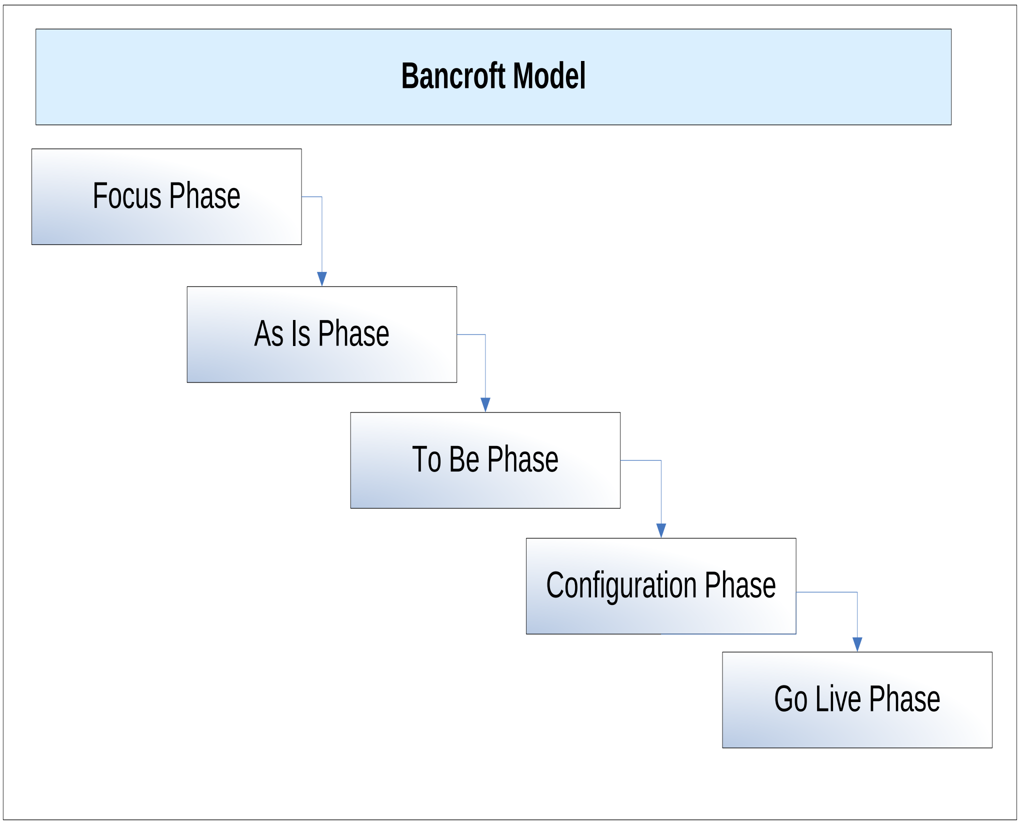


**Figure 2.1:Bancroft Model**

### 2.1.5.2 ERP IMPLEMENTATION MODEL –ROSS AND VITALE MODEL

*Ross and Vitale* [118], based on their experience, came up with a model of ERP Implementation. They proposed four phases in an ERP implementation project and are given below and are shown in figure 2.2.

1. **Design Phase** – This phase is similar to the planning phase in which critical decisions are taken for ERP implementation.
2. **Implementation Phase** – This phase contains all the exercises for readiness of final implementation.
3. **Cutover Phase** – This phase encompasses the activities for sign off and implementation phase.
4. **Continuous Improvement Phase** – All the necessary new requirements are added to the system and system is maintained up to date for requirements.

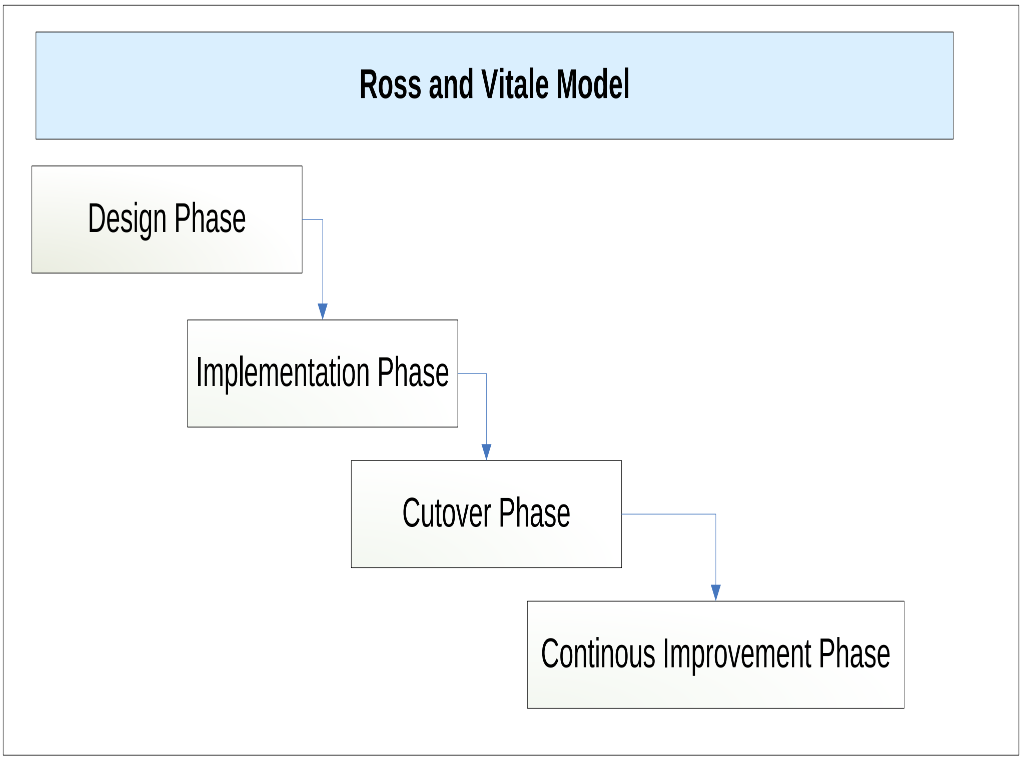


**Figure 2.2: Ross and Vitale Model**

#### 2.1.5.3 ERP LIFECYCLE MODEL

*Estevez and Pastor* [43] presented the ERP Life cycle model that comprised of 6 stages, which are as follows –

1. **Adoption decision phase** – This phase contains the requirement gathering phase and decision of one process over another is taken in this phase.
2. **Acquisition phase** - This phase consists of product selection based on various factors such as industry solution, implementation and training cost.
3. **Implementation phase** – This phase consists of procuring ERP, configuring and customizing ERP and user training activities.
4. **Use and Maintenance phase** – The implemented solution is used by the user and measures to remove disruptions and business changes are taken in this phase.
5. **Evolution phase** – The implemented system is integrated with other systems to ensure maximum benefits for the organization.

6. **Retirement phase** – The implemented solution is replaced by another MIS system for organizational needs.

ERP Lifecycle model is shown in figure 2.3.

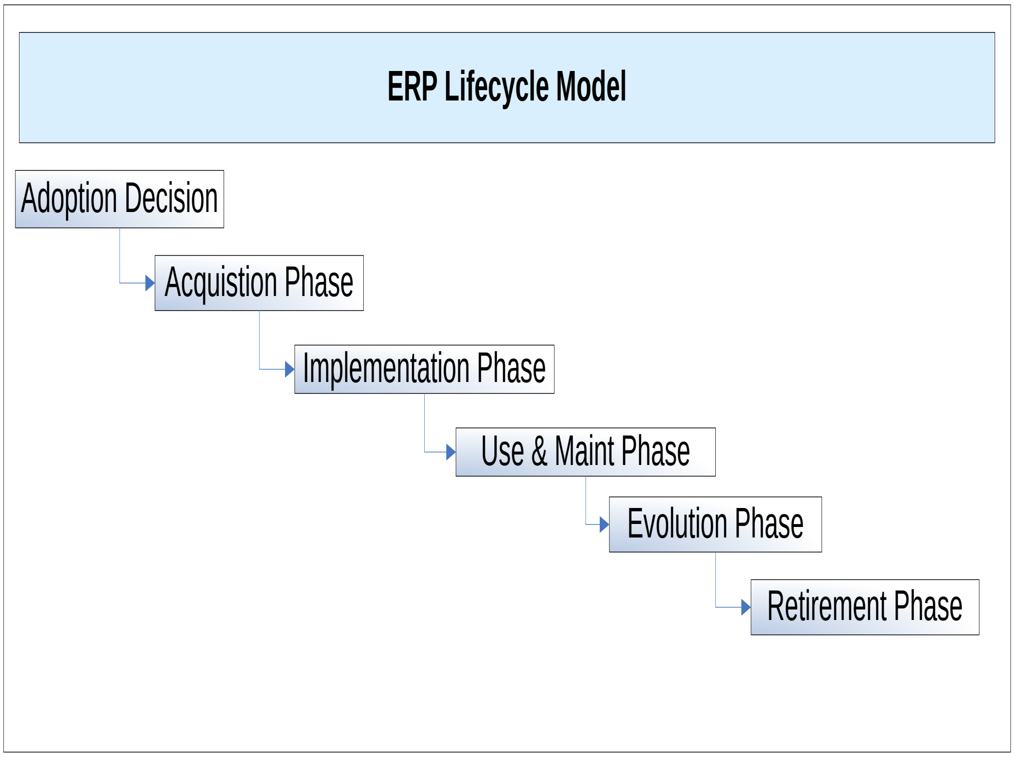


**Figure 2. 3: ERP Lifecycle Model**

### 2.1.5.4 PROCESS MODEL – MARKUS AND TANIS

*Markus and Tanis* [83] proposed four phases for successful ERP implementation. The four phases are related to key activities, key people and coordination between them. Nah et al. [94] suggest that Markus and Tanis' process model follows traditional system development methodology. The four phases are as given below and are shown in figure 2.4.

1. **Chartering Phase** – This phase includes business case preparation, financial approval, acquisition of team and preparation of schedules.
2. **Project Phase** – This phase includes ERP Configuration, integration with various systems, data migrations, user trainings, rollout and infant care.

3. **Shakedown Phase** – System begins to operate optimally in this phase.
4. **Onwards Phase** – New features are added to make system usable as per changing business.

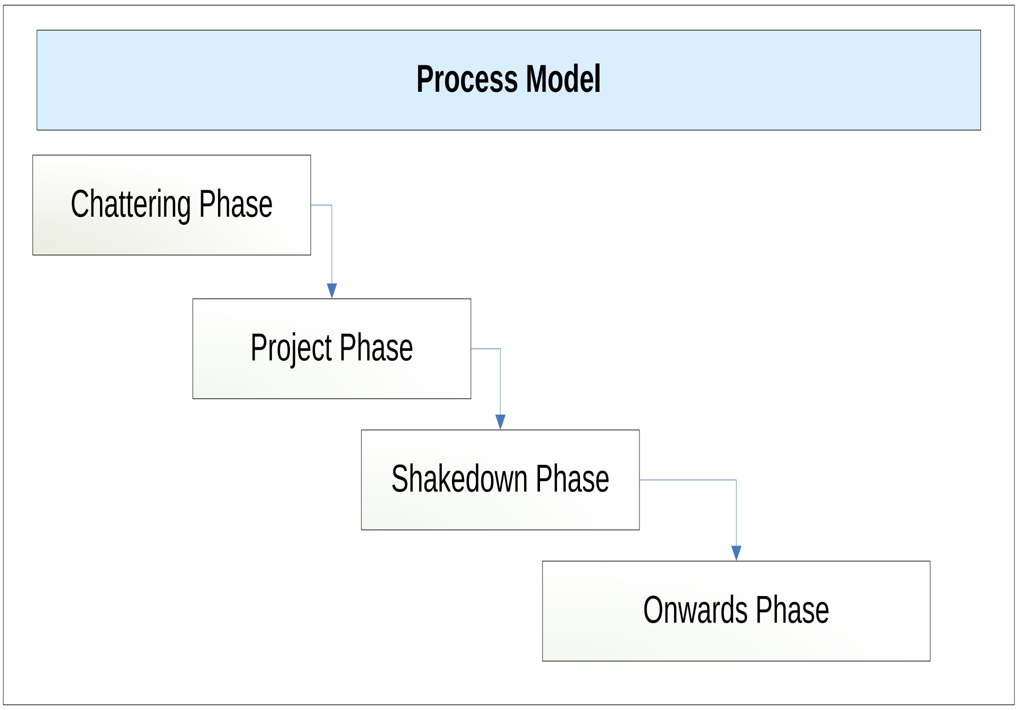


**Figure 2.4: Process Model**

### 2.1.5.5 PROJECT MODEL

*Parr and Shanks* [102] proposed a model which concentrates on project implementation and the critical success factors of each phase. They emphasize that an organization should learn from its past failed projects and should take necessary action from those. They suggest that a large project should be broken into multiple small projects. The model has following phases which are shown schematically in figure 2.5.

1. **Planning**- The planning phase of this model involves package selection, scope determination and implementation approach and selection of team. The phase also includes the formation of Steering Committee.
2. **Project phase** – This phase involves creation of functional and technical designs, reengineering of existing process and UATs.

3. **Cutover and Rollout** – Data from legacy system is migrated to ERP. The cutover strategy is made. Final system is configured and given to the user for usage.
4. **Enhancement** – The system is repaired, extended and transformed for new requirements of Government policies.

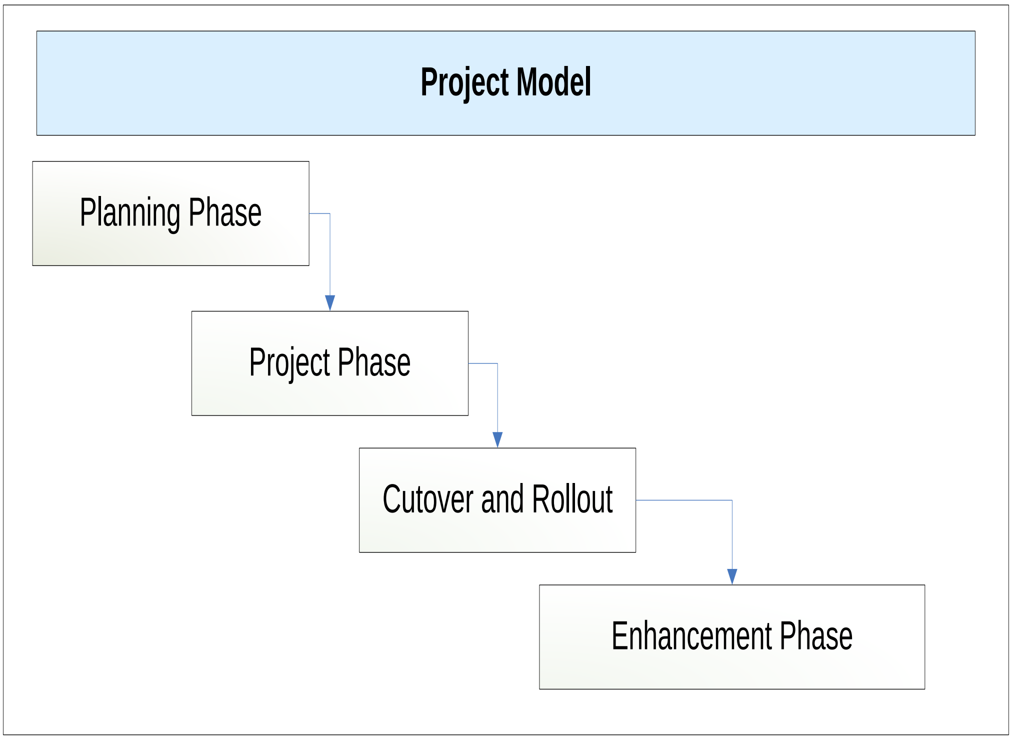


**Figure 2.5: Project Model**

### 2.1.5.6 ASAP 8 by SAP

The ASAP 8 methodology is provided by SAP and this can be categorized into Vendor provided methodology. ASAP consists of 6 phases. These six phases are shown in figure 2.6. The SAP ASAP 8 methodology provides pointers, accelerators, tools templates and empowers team to build solution faster [68].


**Figure 2. 6: ASAP 8 Methodology**

ASAP 8 methodology is a new methodology. The details of the phases in this methodology are given below –

1. **Discover –** This phase unearths the needs of ERP and includes activities like package selection, cost benefit analysis and implementation partner selection.
2. **Prepare –** This phase includes all the actual project initiation activities such as estimating hardware, server, infrastructure, preparation of project charter, assignment of a project manager and resources are done in this phase only. Agile team in ASAP 8 is given [68] below in figure 2.7.

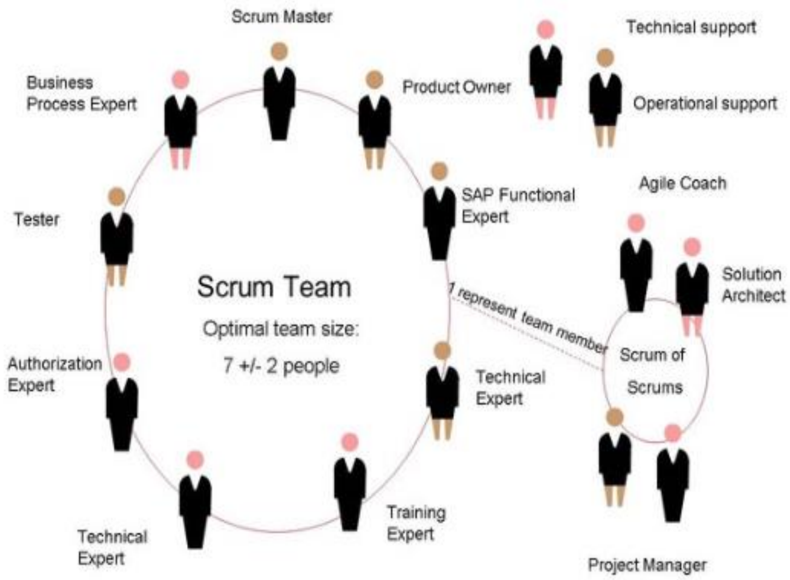


**Figure 2.7: SAP Agile Team**

3. **Explore –** This phase is a critical phase of the implementation. It includes the requirement gathering, fit-gap analysis and blueprinting. The output of this phase in list of customization and configurations required for the implementation.
4. **Realize –** Customizations are developed, and standard functionalities are configured in this phase. This phase also includes the SIT by project team and UAT by business users.

5. **Deploy –** After the UAT approval, cutover is planned, and developments are migrated to Production environment so that users can use it. Project team provides support, training and infant care for initial days and hands over the system to support team.
6. **Run** – System is run as a transaction system for capturing data and analysis. The new customizations are made and deployed in the Production as and when they are requested.

ASAP 8 is a relatively new methodology. At the time of the research for this thesis ASAP 7 was being predominantly used by consulting companies and implementation partners and it did not incorporate the Agile methodology.

#### 2.1.5.6 ORACLE UNIFIED METHOD (OUM)

Oracle, another ERP provider, has given a methodology called as Oracle Unified Method, previously called as AIM – Application Implementation Methodology. *Oracle* [100] has suggested that OUM extends the leading industry standard, Unified Software Development Process into scope of Oracle ERP projects. The five phases of OUM are given below and these are pictorial defined in figure 2.8 –

1. **Inception** – This phase captures the objective of ERP implementation from all stakeholder. The objectives are captured at high level and refined to ensure that all stakeholders are in sync on objectives. The SOW is written in this phase and risks are associated with each objective,
2. **Elaboration** – This phase includes the detailing of the requirements and identifying the required customizations and configurations for the solution. The Proof of Concepts (POC), if any, are done in this phase and design is confirmed in this phase.
3. **Construction -** This phase includes the development of customizations based on the design made in the last phase. Configurations are also done in this phase as per the given list of set ups. In other words, system is made ready for beta testing in this phase.

4. **Transition** – This phase includes the SIT and UAT by the user. Any defects found during the testing are also fixed in this phase. This phase readies the system for acceptance by the customer or the organization. The cutover and deployment are planned in this phase.
5. **Production** – System is deployed in this phase and full support to cater to the issues is provided. In addition, change management procedures to absorb new requirements are devised to ensure smooth functioning of the system.

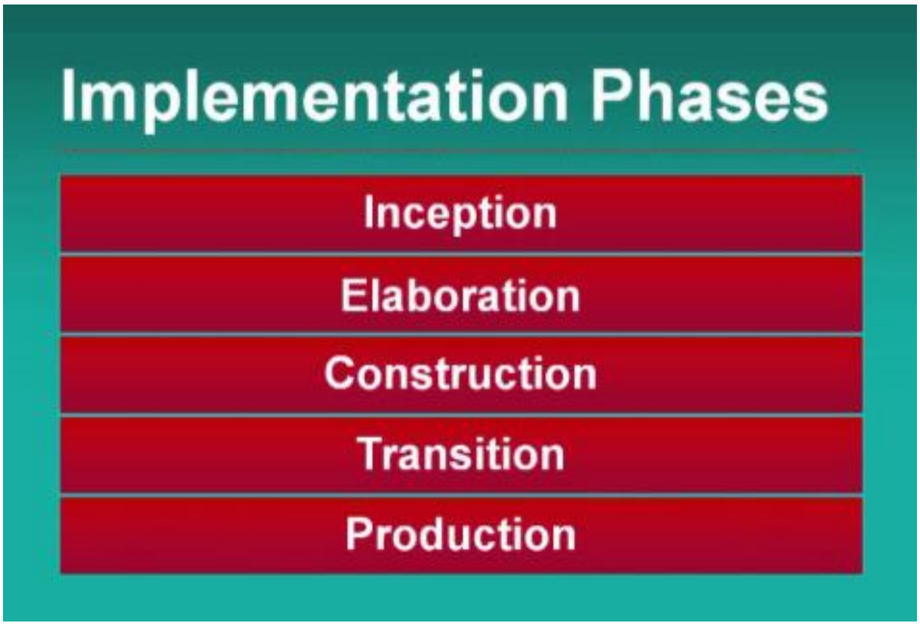


**Figure 2.8: Oracle Unified Method**

### 2.1.5.7 COMPARISON OF ERP IMPLEMENTATION MODELS

ERP Implementations involve changes in the business process or, sometimes, complete reengineering of business requirements. Hence, ERP implementation projects are very different from any software development project. All the frameworks / models provide an effective approach to successfully implement ERP. As discussed earlier, the available models can be categorized in two categories. Models in both categories are first compared with each other and then are compared inter-category i.e. models in scientific community and vendor specific models are compared first and then both categories are compared to each other.

All the given models in scientific community understand that maintenance and enhancement is an integral part of ERP implementation life cycle while *Bancroft model* [15] does not talk about any such phase. The model of Ross and Vitale [118] talks about transformation of ERP while *Esteve's model* [43] talks about the evolution of the ERP where in ERP is integrated with other business applications and helps in decision making in business process. *Esteve's model* [43] talks one step ahead and talks about the activities to retire the ERP system. Various models discussed in scientific community are compared in figure 2.9.

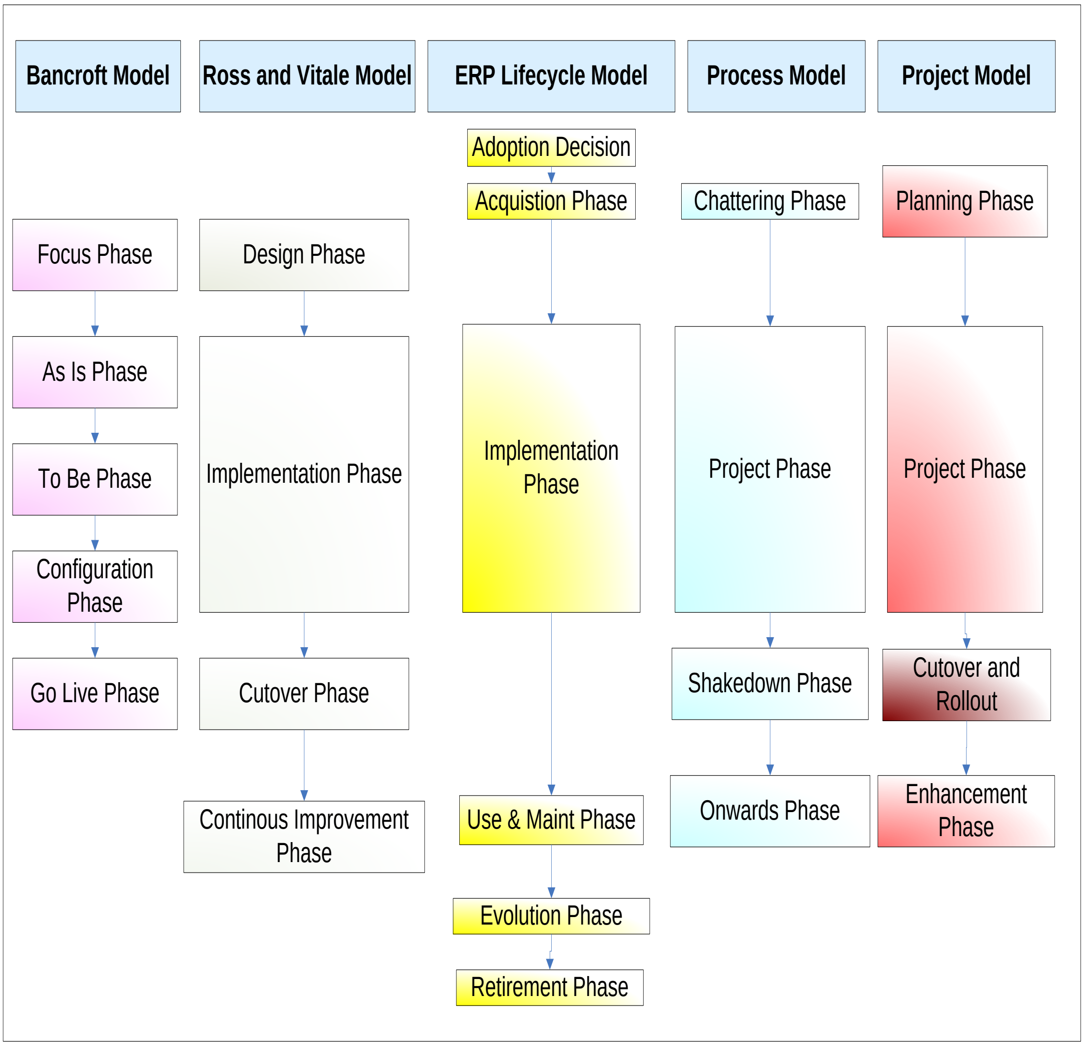


**Figure 2.9: Comparison of various ERP Implementation Models in Scientific community**

All the models proposed in scientific community till now have following issues –

1. High cost or effort involved in change.
2. Lack of feedback mechanism at any stage.
3. All the models follow big bang approach that may lead to issues in change management.
4. Lack of incremental build does not cope up with ever changing business needs.
5. Lack of mechanism to focus on business values and prioritize the work as per business needs.
6. The above implementation models don't talk about any mechanism to complete the implementations in a stipulated time i.e. time boxing of activities and predictable delivery.
7. No mechanism to predict cost and schedule.
8. No platform where various module owners, various function owners and people from various strata of levels talk to each other i.e. no platform to promote integrated working.

Models present in the scientific community do not seem to be up to date with the current trends in industry. Vendor specific or consultant specific methodologies seem to be more evolved than methodologies discussed in the scientific community [93]. As discussed earlier, this thesis discusses two vendor specific methodologies – ASAP 8 provided by SAP and OUM provided by Oracle. Prima facie, SAP markets ASAP 8 as Agile way of implementing SAP or any ERP and on similar lines Oracle also echoes that OUM can be used with Agile [68,100]. On deeper analysis, it is seen that ASAP 8 methodologies delivers the functionalities by dividing them in various phases, timeboxing each phase and reusing the accelerators [159] and OUM also delivers in an iterative fashion [153]. Iterative deliveries are phased deliveries in normal waterfall projects. The whole idea of Agile is to build incrementally absorbing the customer requirements. Both methodologies speak about similar things. However, both approaches lack the following:

1. Both models talk about iterative delivery, but incremental delivery is missing from both models. Both concentrates on fast delivery of ERP but do not talk about the constant pace.
2. Both models lack the idea of removing waste i.e. the Lean approach.
3. Both methodologies don't talk about the reflection of learning from past.
4. Both methodologies don't have any method to prioritize the requirements for each delivery nor they talk about incremental delivery.
5. Both methodologies are silent on handling requirements coming later in the lifecycle.
6. Like traditional methodologies, both methodologies lack feedback mechanism.

Various models – academic and vendor specific were discussed in this section. Next section discusses the issues in ERP implementation. These issues were identified in literature research and from my own experience in multiple ERP implementations in various roles.

### 2.1.6 ISSUES IN ERP IMPLEMENTATIONS

All the ERP models defined till now follow the Waterfall methodology for the configurations and customizations. A good number of scientific studies are present for finding critical success or failure factors for ERP Implementation but there are few studies depicting the issues in ERP implementations. *Chandra et al.* [27] have listed few of these issues. Based on my own experience as ERP Program Manager and consultant, I have validated these issues and have added few issues to the final list to understand the critical issues. The issues thus collated were further validated with the small cohort of ERP experts performing the role of Project managers – 41 Nos, Program managers -21 No, Architects-23 No, Quality assurance managers - 43, Process Owners -28, System Analysts – 62 No, IT Director – 14 No. The issues have been assigned criticalities of C1, C2, C3 and C4. The basis for the categorization is given in below:

1. C1-The problem / issue is mentioned by more than 50% of the respondents.
2. C2-The problem / issue is mentioned by more than 35% of the respondents.
3. C3-The problem / issue is mentioned by more than 25% of the respondents.
4. C4-The problem / issue is mentioned by more than 15% of the respondents.

The issues in the table below form the basis of this research and call for a new framework. The details of the survey are given in Appendix 1. The survey is based on the issues identified in various literature and my own experience.

| ID1 | Issue | Agreement | Source | Criticality |
|---|---|---|---|---|
| I1 | Developed system does not reflect the current requirement of system | >50 | Chandra [27] | C1 |
| I2 | Defect propagation into next phase | Between 35 and 50 | Chandra [27] | C2 |
| I3 | Missing Customer feedback mechanism | >50 | Anderson [8],Tipaldi et. al. [136] | C1 |
| I4 | Big-bang implementations push the test cycle at end that can lead to quality problems. No parallel testing leading to changes with high effort / cost | Between 35 and 50 | Boehm [20], Chandra [27] | C2 |
| I5 | Requirement Volatility | >50 | Cohen *et al.*[31] | C1 |
| I6 | Later phases squeezed after Initial phases consume more effort and time than allocated. | >50 | Sommerville [129] | C1 |
| I7 | Heavy documentation | >50 | Sommerville [129], McBreen [86] | C1 |
| I8 | Dropped requirements become business critical | >50 | Self-experience | C1 |
| I9 | High effort / Cost of Change. Tough to respond | >50 | Self-experience | C1 |
| I10 | Changes owing to Non-Functional Requirements | Between 35 and 50 | Self-experience | C2 |
| I11 | Multiple stakeholders and their influences | Between 35 and 50 | Self-experience | C2 |
| I12 | Integration issues within ERP | Between 25 and 35 | Self-experience | C3 |
| I13 | Integration issues with Third Party systems | Between 25 and 35 | Self-experience | C3 |
| I14 | Lack of knowledge sharing / transition | Between 25 and 35 | Self-experience | C3 |
| I15 | Gold plating - Defects due to value-adds provided to customer | Between 35 and 50 | Self-experience | C2 |
| I16 | Over processing such as extra validations, checks | Between 35 and 50 | Self-experience | C2 |
| I17 | Repeating defects in various modules. | Between 25 and 35 | Self-experience | C3 |
| I18 | Communication issues - Issues because of middlemen between developers and end users | Between 25 and 35 | Self-experience | C3 |
| I19 | Individual Competency issue | <25 | Self-experience | C4 |
| I20 | Wastage of time in waiting for approval and green signal from past phase. | Between 25 and 35 | Self-experience | C3 |

**Table 2.4: Issues in ERP Implementation**

The above issues are corroborated by research and consulting firms such as Panorama Consulting, Bista and FinanceOnline. Bista Solutions on their blog have clearly mentioned that improper requirement gathering leads to requirement volatility, requirements miss or drops. They also confirm that organizations don’t have clear destination which leads to all these issues and success of ERP becomes a moving target [152]. The issues in requirement may lead to work on wrong requirements and important requirements are often overlooked. The above points are also echoed by *Finance Online* [156] and the root cause of this could be identification of right people. In my experience, I have also seen that organizations have their own dynamics and these dynamics lead to choosing wrong people which is concurred by Bista Solutions [152] and they further mention that most of the issues arise because of insufficient testing because testing is the first area to be axed to meet the slipping deadline. The lack of testing often leads to integration issues within and boundaries of ERP. *Panaya Consulting* [161] in their blog have mentioned that untested and un-reviewed defects often move into next phases of life cycle which makes that it costlier to identify and fix. The defects may occur because of lack of change management [156] and change management becomes troublesome in case of big bang implementation and may lead to insufficient training to users and inadequate support to postproduction support team [160]. All the industry blogs and research reports [152,160,161] mention communication as key to success. The communication issue is attributed to multiple layers between the business user and developers or lack of business feedback on the system.

Agile has been proposed as solution to the issues discussed in this thesis [50,68,154,155,158] and few researchers have proposed Lean Thinking as solution to the current issues [112,138]. Software industry is looking for a solution to optimize the development and blend of Lean and Agile is presented as a solution to the problem [47,73,74,75]. Lean Thinking and Agile are discussed in detail in section 2.2 and 2.3, respectively, of the thesis.

## 2.2 LEAN THINKING

Lean Thinking was carved out of principles of Industrial Engineering to maximize the outcome or value and minimizing the waste. Japanese companies, accustomed to mass production, found themselves challenged with low volume production during and after the Second World War. This led them to devise new methods such as Lean. *Poppendieck and Poppendieck* [108] described 'Lean' as any efficient management practice that minimizes waste while *Shah and Ward* [123] defined Lean production as: 'an integrated socio-technical system whose main objective is to eliminate waste by concurrently reducing or minimizing supplier, customer and internal variability'. Hence, there is no concrete definition of Lean. During the study of Japanese production methods, MIT researchers found new methods in the framework called Toyota Production System (TPS). These new methods were christened as Lean Manufacturing [9,150]. The Lean manufacturing was centered on ideas of doing more with less and producing just the right quantity. Inspired by TPS, various researchers understood and summarized the Lean Thinking.

As per *Ohno* [99], the TPS revolves around the process of removing

- Muda (Waste)
- Mura (Variation)
- Muri (Overburden)

He defined waste as a non-value adding activity. He has further categorized waste in following categories:

1. Over-production – Producing more than the required or ordered quantity results in wastage of effort and energy to produce it and results in wastage of storage space, material handling capabilities. It may further lead to waste if the product perishes.
2. Waiting – Time subsequent operations remain idle to let complete predecessor complete the processing. This results in wastage of man-hours, machine hours and ultimately hits the profit.
3. Transportation – Unnecessary movement of product from one workstation to other workstation is a wasteful activity.

4. Over-processing – Commonly called as gold plating i.e. doing the activities multiple times to ensure absence of defects or increase customer delight. Time and effort used here could be used for right activities to increase profit.
5. Inventory – Inventory includes the finished goods, work in process goods and the raw material. Unnecessary inventory increases the cost and consumes the space.
6. Motion – Poorly arranged space and layouts result in unnecessary bends, reach or walks for doing a job. Productivity can be increased by removing these.
7. Defects – Defects occur due to non-standard operations and processes. Time and effort are used to correct the defects or defects may yields to wastage due to scrap.

An exhaustive listing of terms, tools and techniques used in Lean Thinking are given in The Lean Lexicon [82] and are presented here in Table 2.5. The Lean tools and techniques are given below in Table 2.7. TPS talks about 'Just-In-Time' (JIT) which later established the Lean Thinking or Lean Manufacturing [149].

| Practice / Tools / Techniques | Description |
|---|---|
| Automation (Jidoka) | Providing machines and operators the ability to stop the work immediately whenever a problem or defect is detected. Related to the concept of 'Stop-the line'. |
| A3 Report | Problem solving, analysis, corrective actions, and action plan on a single sheet of large paper. |
| Cell | Location of processing steps for a product immediately adjacent to each other so that parts can be processed in very nearly continuous flow. |
| Chief Engineer (Shusa) | Manager with total responsibility for the development of a product. |
| Just-in-Time (JIT | System for producing the right items at the right time in the right amounts. |
| Kaikaku | Process improvement through a radical change. |
| Kaizen | Continuous process improvement through small and frequent steps. |
| Kanban | Method based on signals for implementing the principle of pull by signaling upstream production and delivery. |
| Lead time | Metric defining the total time that the customer must wait in order to receive a product after placing the order. |
| Level schedule (Heijunka) | Levelling schedule by sequencing orders in a repetitive pattern and smoothing the day-to-day variations in total orders to correspond to longer-term demand. |
| Mistake-proofing (Pokayoke) | Method that helps operators to prevent quality errors in production by choosing the wrong part. |
| Big room (Obeya) | Project leaders room containing visual charts to enhance effective and timely communications, and to shorten the Plan-Do-Check-Act cycle. |
| Set-Based Decision (concurrent engineering) | Approach to design products and services by considering sets of ideas , defer the final design until a good decision is made |
| Self-reflection (Hansei) | Continuous improvement practice of looking back and thinking about how a process can be improved. |
| Six Sigma | Management system focused on improving quality by using mathematical and statistical tools to minimize variability. |
| Standardization | Precise procedures for each activity, including working on a sequence of tasks, minimum inventory, cycle time (the time required to complete an operation) and takt time (available production time divided by customer demand). |
| Usable knowledge | Capturing the knowledge to be applied in other projects. |
| Value Stream Mapping (VSM) | Tool for analyzing value and waste in the production process. VSM provides a standardized language for identifying all specific activities that occur along the value stream. |

| Practice / Tools / Techniques | Description |
|---|---|
| Work-In-Progress(WIP) | Items of work between processing steps. |
| 5-Whys | Method for analyzing/finding the root cause of the problems consisting in asking 'why' five times whenever a problem is discovered. |

**Table 2. 5: Elements of Lean Thinking**

*Womack and Jones* [148] studied the TPS system and Lean Manufacturing and provided roots by defining following terms to define Lean Thinking. These terms are given in table 2.6.

| Principle | Description |
|---|---|
| Value | Value is everything that a customer is willing to pay. Defining and understanding value from the perspective of the customer is the central focus of Lean thinking. The goal is to make organizations deliver as much customer value as possible. Its counterpart, waste, or 'muda' in Japanese, is everything that absorbs resources but outputs no value. |
| Value Stream | Value stream is the optimized end-to-end collection of actions required to bring a product/service from customer order to customer care, ensuring that each activity adds customer value. |
| Flow | Flow means that activities are organized as a continuous 'flow', eliminating discontinuities in the value stream and enabling smooth delivery. Flow requires that the unnecessary steps are eliminated (waste in Lean terminology). Opposite to mass production, the products are made using 'single piece flows' |
| Pull | Pull implies producing products only when they are really needed, by making customer needs and the market the primary decision-drivers. Accordingly, in a pull system, an upstream process only produces when a downstream process is ready and 'pulls' some more work from the upstream process. The goal is to minimize the inventories that do not produce customer value but are sources of waste. |
| Perfection | Perfection maintains the enterprise-level improvement and learning continuously ongoing. The aim of Lean is to achieve zero waste and defects based on the concept that there is no end to the strive for perfection. |

**Table 2. 6: The core principles of Lean thinking**.

The idea of Lean Thinking is to endure customer satisfaction, employee satisfaction and minimizing unnecessary cost to customer.

*Liker* [76], in his book the Toyota way, extending the TPS has provided a system – Lean Thinking which is designed to provide the tools for people to continually improve their work. This system has four pivotal points –

1. Long-Term Strategic philosophy
2. Right results are produced by right processes.
3. Value addition to the Organization by developing people
4. Solving root cause and learn from issues.

The 14 principles based on these 4 points are given below –

- Long-term and Strategic philosophy.
- Continuous process flow to make the problem visible.
- Use ‘pull’ systems to avoid overproduction.
- Level out the workload (heijunka).
- Do it right the first time.
- Standardization of tasks.
- Use tried and tested technology
- Grow leaders who thoroughly understand the work,
- Develop exceptional people and teams who follow your company’s philosophy.
- Help your partners to grow
- Go and see for yourself to understand the situation (genchi genbutsu).
- Make decisions slowly by considering all options and implement decisions rapidly (nemawashi).
- Learning organization (hansei) and continuous improvement (kaizen)

Morgan and Liker [91], in an attempt to define Lean, have provided 13 principles in Toyota Product Development Process. These 13 principles are classified into three categories:

**Process**

- Define value from customer perspective and isolate it from waste.
- Explore alternative solutions to maximize the design options.
- Balanced and level flow.
- Standardizations and flexibility to predict outcomes.

**People**

- Chief engineer owning the system.
- Good knowledge of current and cross functional systems.
- Technical competence.
- Integrated suppliers in product development.
- Learning.
- Excellence and continuous improvement.

**Tools**

- Choose technology to fit the process and people. Adapt technology to fit your people and process.
- Visual Communication.
- Tools for standardization and organizational learning.

In all the above discussions, it is seen that there is no concrete definition of Lean Thinking. However. *Womack and Jones* [148] have tried defining Lean Thinking formally as doing more in less i.e. increasing the value and reducing the non-value. Lean Thinking has seven principles which can be used for development of software [107,108,109,110] and are given below in table 2.7. Each of these principles are assigned an identification id, referred as ID, and these principles will be referred by these IDs further in the thesis.

| ID | Lean Thinking Principle |
|---|---|
| LT01 | **Eliminate waste:** Anything not adding any value to customer or product should be removed. |
| LT02 | **Amplify Learning**: Learning can be termed as getting better understanding of customer needs, solutions and testing strategies. Processes and practice should support and provide environment for learning |
| LT03 | **Defer commitment**: Delay the commitment of irreversible decision till last moment and option for change should remain open till last moment |
| LT04 | **Deliver as fast as possible**: Minimize delivery time and turnaround time from requirement gathering to final delivery of software. |
| LT05 | **Respect people**: Create a team of technically excellent people, provide them reasonable and realistic goals and trust them on getting the job done |
| LT06 | **Build quality in**: Ensure that defects are not injected in first place and they should be eliminated as soon as they are discovered. |
| LT07 | **Optimize the whole**: Optimize the whole value stream i.e. from receipt of a requirement to software deployment and avoid sub-optimization |

**Table 2.7: Lean Thinking Principles**

All the above discussions talk heavily about banishing the waste. In manufacturing industry, waste can be identified as the physical material, motion or worker activities but the same is tough to be identified in case of software development [130]. *Maglyas et al.* [80] suggests that Lean Thinking helps in shortening release cycles and reducing time-to-market by creating a pull environment for the flow. *Poppendieck and Cusumano* [106] suggest using Lean Thinking principles to software engineering to have process and quality improvements. Lean Thinking has drastically improved traditional ways of production by improving efficiency, quality, faster delivery and further empowering the team members [98,104]. *Liker* [76] has provided an idea of wastes in any business which is shown in figure 2.10


**Figure 2.10: Non-value-adding waste in business.**

### 2.2.1 APPLICATION OF LEAN IN SOFTWARE APPLICATION DEVELOPMENT

*Poppendieck and Poppendieck* [107] conceptualized the application manufacturing principle of Lean Thinking in the software development [81]. Later, Lean architecture [34] and Kanban [10] was introduced to make software using Lean Thinking. Various researchers have spoken about application of Lean Thinking in software development. Table 2.8 summarizes the interpretation of Lean Thinking by few researchers in application development area.

| Author | Interpretation |
|---|---|
| Poppendieck and Poppendieck (2003, 2006, 2009) | Seven Principles:<br>1. Eliminate waste<br>2. Build quality<br>3. Create knowledge<br>4. Defer commitment<br>5. Deliver fast<br>6. Respect people<br>7. Optimize the whole |
| Middleton and Sutton (2005) | Five Lean principles: value, value stream, flow, pull and perfection. |
| Larman and Vodde (2008) | **Pillars**:<br>1. Respect people<br>2. Continuous improvement<br>**Principles**:<br>1. Long-term philosophy,<br>2. Flow<br>3. Pull<br>4. Less variability and overburden<br>5. Stop and fix<br>6. Master norms<br>7. Visual management<br>8. Good technology<br>9. Grow Leaders within<br>10. Develop people<br>11. Help partners<br>12. Go-See<br>13. Consensus<br>14. Reflection and kaizen. |
| Anderson (2010) | **5 Properties**<br>1. Visualize workflow<br>2. Limit WIP<br>3. Make policies<br>4. Measure and manage flow<br>5. Models to recognize opportunities. |
| Reinertsen (2009) | Product development flow, managing queues, reducing batch size, WIP constraints. |

**Table 2.8: Various definitions of Lean Thinking in application development.**

Manufacturing companies have also used Agile to handle ever changing demand volume using production system changing quickly and swiftly. *Christopher and Towill* [28] defined agility as a concept of Flexible Manufacturing System. *Sharifi and Zhang* [124] and *Sherehiy et al.* [126] stress that qualities such as responsiveness, competency, flexibility and quickness help in meeting the changing requirements [87,92].

## 2.3 AGILE SOFTWARE DEVELOPMENT

Agile officially came into existence post the formulation of Agile Manifesto. The Agile manifesto defines the four values which are shown below in the figure 2.11.

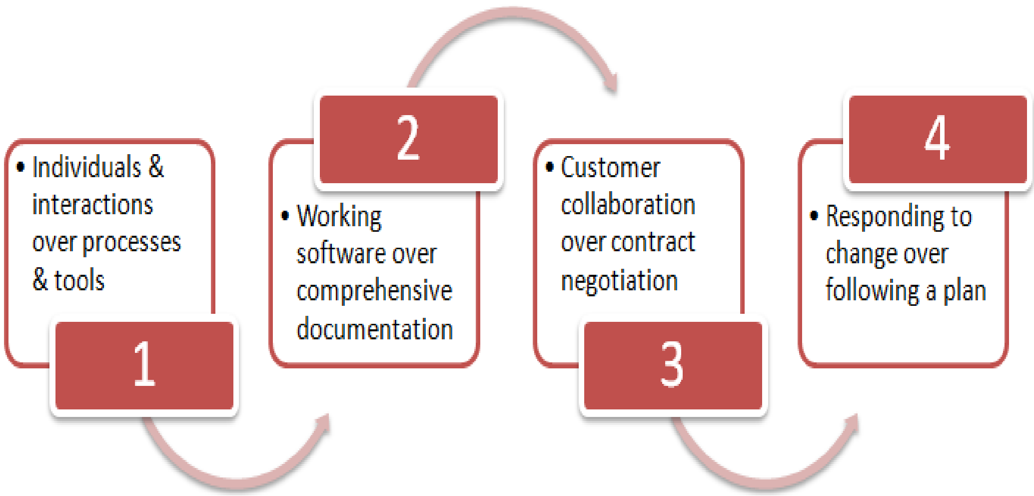


**Figure 2.11: Agile Values.**

Agile methodologies were seen as an alternative to heavy weight document driven software development methodologies [32]. Agile Manifesto is based on principles given below in table 2.9. Each principle is assigned an ID and these principles will be referred with these ID in subsequent chapters.

| ID | Principle |
| --- | --- |
| AM1 | Satisfy Customer. |
| AM2 | Welcome changing requirements, even in late development. |
| AM3 | Deliver working software frequently |
| AM4 | Cooperation between business people and developers throughout the project. |
| AM5 | Trust motivated individuals and build projects around them. |
| AM6 | Enable Face-to-face conversation. |
| AM7 | Measure progress by delivered working software. |
| AM8 | Maintain a constant pace of delivery. |
| AM9 | Technical excellence and good design to high quality product for easy scalability and maintenance. |
| AM10 | Simplicity |
| AM11 | Self-organizing teams |
| AM12 | Continuous reflection |

**Table 2.9: Agile Principles**

Agile Manifesto made the foundation of ASD and several practices such as eXtreme Programming (XP), Scrum, Dynamic Systems Development Method and Feature-Driven Development came into light [16,122,131,101].

Agile Methodologies have been seen to be efficient in satisfying customer, speeding up the delivery with increase in quality, visibility to stakeholders and collaboration in decisions [18,88,47]. Customer collaboration enables to provide faster resolution of issues, choosing right functionalities for successful implementations [21,90] and it also enables to deliver quality in lesser period. Agile Methodologies believe in transparency among stakeholders and have confidence in working prototypes, codes, iterative development and delivery with minimum documentation. Figure 2.12 represents the factors affecting Agile Implementation methodology by *Livermore* [77].

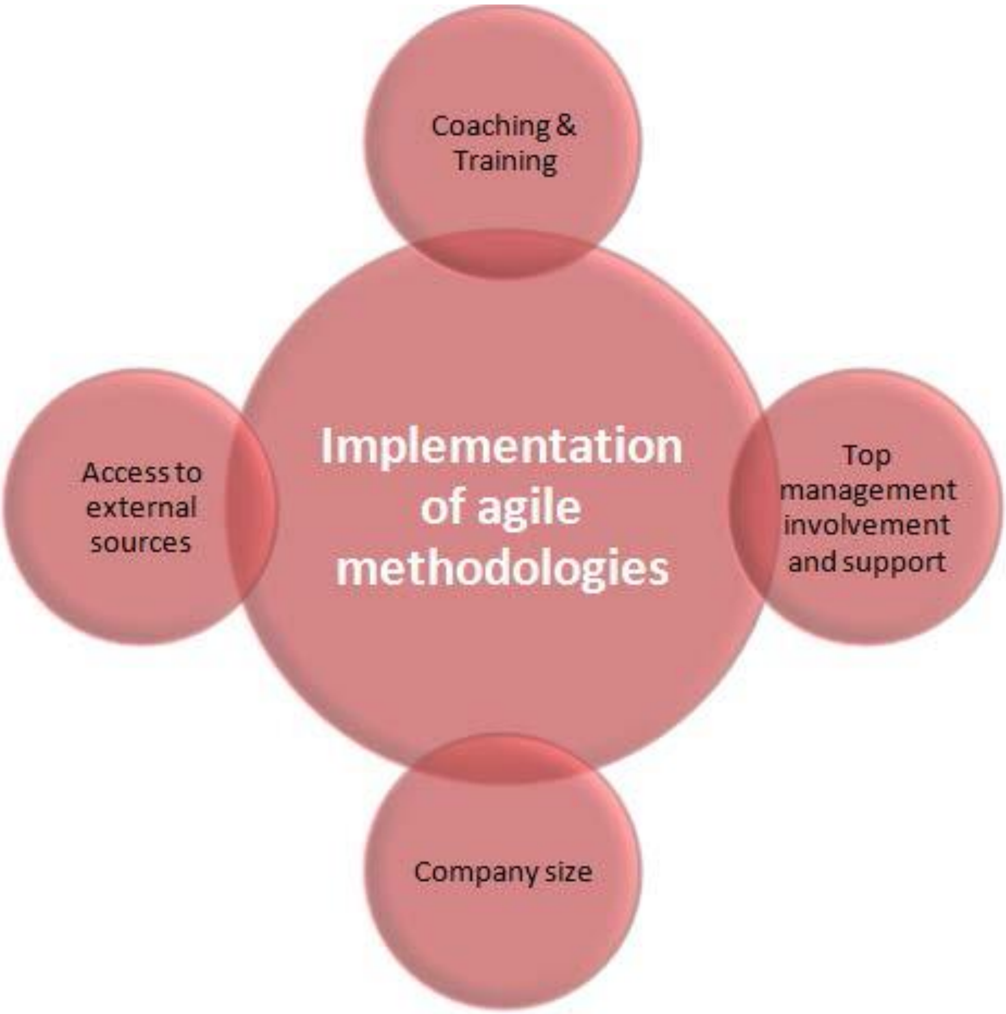


**Figure 2.12: Factors affecting the implementation of Agile Methodology.**

Most commonly used Agile practice in software development is Scrum [122], which requires team to have flexibility to respond to customer changes. The requirements are defined in the backlogs, stories and features which are developed in time boxed or constrained manner called Sprints. The sprint backlog is subset of the product backlog which is identified by some criteria and it never changes during the sprint. Sprints have major roles as Product Owner – Voice of Customer, Scrum Master – The servant master and Team – self organized and empowered developers who can take their own decisions. The scrum process (adapted from [89]) is shown in figure 2.13.

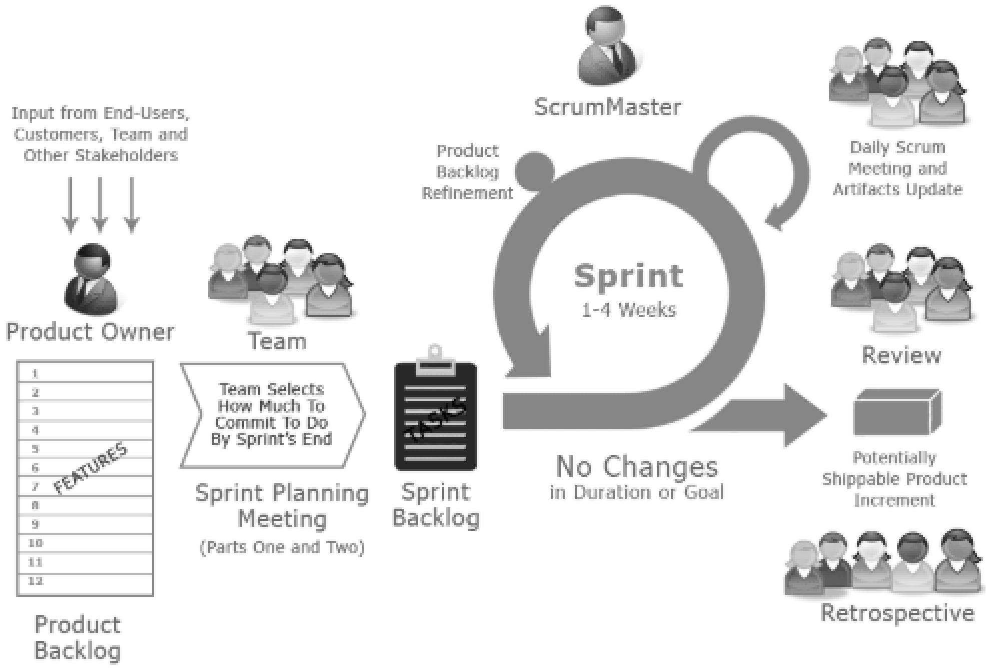


**Figure 2.13: The Scrum process**

*Dyba and Dingsoyr* [42] have claimed exciting benefits of Agile but on the other hand practitioners have raised few challenges on it. They have found that Agile has limitations on scaling for large and complex developments [140,20]. Also, the teams working in distributed environment have cited communication issues [105]. Agile methodologies also fail to yield result in the area where dependence is upon the other parts of enterprise which don't come around scrum [39,137] and require more research in the area of Agile Software Development [2,58]. The benefits and challenges of Agile [89] are given below in table 2.10.

| Benefits | Challenges |
|---|---|
| (1) requirements are more precise due to reduced scope and thus easier to estimate,<br>(2) direct communication in teams reduces the need for documentation,<br>(3) early feedback due to frequent deliveries,<br>(4) rework reduction,<br>(5) testing resources are used more efficiently<br>(6) higher transparency of who is responsible for what creates incentives to deliver higher quality<br>(7) low requirements volatility in projects<br>(8) reduction of waste (discarded requirements) in the requirements engineering process. | (1) challenges in regard to realize continuous testing,<br>(2) increased maintenance effort with an increase in the number of releases,<br>(3) management overhead due to the need for coordination between teams,<br>(4) detailed dependencies are not discovered on a detailed level due to lack of focus on design,<br>(5) long requirements engineering duration, due to complex decision processes in requirements engineering,<br>(6) requirements priority lists are hard to create and maintain,<br>(7) waiting times in the process, specifically in design waiting for requirements,<br>(8) reduction of test coverage due to shortage of projects and lack of independent testing,<br>(9) increased configuration management effort. |

**Table 2.10: Benefits and Challenges of implementing Agile**

In the next section, Lean and Agile are compared based on various aspects.

## 2.4 COMPARING LEAN THINKING AND AGILE METHODOLOGIES

Table 2.11 categorizes the activities in an IT project.

| Planning | Execution | Deliver |
|---|---|---|
| • Specify Value by Customer Focus<br>• Quality of the Product<br>• Managing the flow | • Pace of Development<br>• Requirement Changes and Flexibility<br>• Motivation and People Management<br>• Co-operation in different teams involved | • Delivery and releases<br>• Learning |

**Table 2.11: Project Planning, Execution and Delivery Matrix**

### 2.4.1 PLANNING PHASE

#### 2.4.1.1 DEFINE VALUE USING CUSTOMER FOCUS

Different situations and contexts define value in their own way but for this research value is confined to value-based software engineering. Lean Thinking and Agile Methodologies focus on customer satisfaction. Lean Thinking achieves customer satisfaction by removing the wasteful or non-value adding task and performing only tasks which add some value to process, product or customer [110]. On the other hand, Agile Methodologies satisfy customer by frequent deliveries of working software. Prima facie, both Lean Thinking and Agile Methodologies look to be different in approach but on digging down one can understand that frequent deliveries can only be achieved by removing waste. Thus, Lean Thinking and Agile Methodologies agree on customer focus.

#### 2.4.1.2 QUALITY

Quality is often described as conformance to requirement and absence of anomalies or defects [1,149]. Lean Thinking (LT06) maintains quality by removing the defects as soon as it is discovered. Lean Thinking follows the idea of 'Product Excellence' i.e. delivering the product which is free from any kind of defect and conforms to requirements [110]. However, Agile Methodologies (AM09) focusses on technical excellence to deliver quality [88]. In simpler words, we can say that Lean Thinking and Agile Methodologies differ from each other on aspect of quality. Lean Thinking talks about quality control and Agile Methodologies talk about quality assurance.

#### 2.4.1.3 OPTIMIZE THE FLOW

Lean Thinking emphasizes on looking at the total flow and optimizing it (LT07) over fixing the small stages or phases. The flow should be optimized considering the customer needs from inception to delivery. No such equivalent is found in the Agile Methodologies.

### 2.4.2 EXECUTION PHASE

#### 2.4.2.1. VELOCITY OF DEVELOPMENT

Agile Methodology (AM08) talks about delivering the product or service at a constant, consistent and sustainable pace. However, Lean Thinking does not talk about the pace, speed or velocity of delivery. Nevertheless, Lean Thinking can help in increasing the pace by using Kanban.

#### 2.4.2.2 REQUIREMENT FLUCTUATIONS AND FLEXIBILITY

Lean Thinking advocates that any promise technical or functional leading to any irreversible action should be delayed as late as possible to save the system (LT03). Agile Methodology suggests accepting change, even if it comes in later phases of implementation or construction (AM02). In first look, Lean and Agile Methodologies appear to be same as both accept changes. But both are entirely different. Agile Methodologies try to appease customer by accepting change, but Lean Thinking ensures to have correct requirements before it is late to reduce the requirement volatility.

#### 2.4.2.3 ENTHUSIASM AND TEAM MANAGEMENT

Agile (AM05) talks about building a team with self-motivated and highly enthusiastic people. The team should get proper conducive environment to work and deliver successfully. Lean Thinking (LT05) talks about having a good leader and strong force under him [79]. The team will have mutual respect and will pick their own task [47]. Both Lean Thinking and Agile Methodologies concur each other on this aspect. Both talk about talented team and no hierarchies and team picking up their own tasks, but Lean Thinking includes all stakeholders and promotes communication.

#### 2.4.2.4 TEAMWORK IN DIFFERENT TEAMS

Agile (AM04) talks about various teams such as Technical and Business to work together and cohesively. It also stresses on cooperation and colocation.

But Lean Thinking does not talk about colocation. However, it talks about the removing of waste from whole system and that is possible only if there exists a cooperation between all affected teams.

### 2.4.3 DELIVERY PHASE

#### 2.4.3.1 DELIVERY AND RELEASES

Agile Methodologies (AM03) talk about the frequent delivery of working software and progress should be measured by working deliveries (AM07). Lean Thinking (LT04) talks about the delivery of working product in minimum time. Diving down, both talk about the same. As per Lean Thinking, incomplete product does not add any value to customer, process or product. Hence, as per Lean Thinking, reducing the delivery time is as good as frequent delivery by continuously learning and reflecting the learning (AM12, LT02) in upcoming deliveries.

#### 2.4.3.2 LEARNING AND IMPROVEMENT

Agile Methodology (AM12) conduces the system where in team delivers the current iteration by correcting and learning from mistakes done in past iterations. This stresses on constant thinking and efficiency and builds teams confidence by reflecting the efficiency. Lean Thinking (LT02) evolves the idea of learning the process, defects and customer needs during the development process to get a better understanding of customer wants, potential changes in architecture, development and quality strategy and risks involved. Lean Thinking (LT02) goes to the extent of achieving better results by implementing the knowledge and lessons learnt.

Both paradigms essentially stress on following –

- Learning from past mistakes or right deeds to plan new actions. Setting the benchmark and defining the process.
- Feedback mechanism and augmenting the actions by implementing the feedback.
- Creating and transferring knowledge to team members

Lean and Agile differ on various aspects and seem to be complementing each other. In the next section possibility of combining Lean and Agile for ERP implementation is evaluated.

## 2.5 COMBINING LEAN AND AGILE FOR ERP IMPLEMENTATIONS

With the success of Lean Thinking, which was predominantly used in manufacturing organizations, and Agile in software development, it is assumed that Lean Thinking and Agile is the solution to the problems troubling the ERP Implementations. These papers describe in detail about the theoretical aspect of the problem and fail to talk about any practical case [154]. *Rajakaruna and Wijayanayake* [112] conclude that there is no study that talks about the complete usage of Agile or any related methodology in ERP implementation. They also conclude that Agile alone cannot be the solution to the problem that ERP implementation projects face. *Isetta and Sampietro* [59] also concur the same and suggest that a hybrid process which takes the best of the waterfall and Agile could be the solution to the problem. *Goedhard* [50] suggest that Agile methodologies have been able to show an effect on small implementations, but no statistical conclusion could be drawn between success and Agile Methodologies for ERP Project. He even contributed the reason of failure was Agile' s inability to scale. The reason of this inability is attributed to failure to recognize the bigger picture or look things at whole. *Jayawickrama et al.* [61] suggest that Agile can be used for projects but for organization wide projects such as ERP implementation could not be benefitted by Agile. However, they also maintain that for such projects, organizations should use Agile with nonconventional methods – which have proven beneficial in taming other problems of similar magnitude. As per SAFe, Lean Thinking could be one such method to enable Agile to scale up to Organizational problems.

Lean Thinking and Agile Methodologies are two entirely different paradigms, Lean emphasizes on increasing value by removing waste and Agile aims at plasticity and adaptability to changes [70]. Notwithstanding that both are different, Lean Thinking and Agile Methodologies support each other in principle [97,72]. Lean carries all Agile principles but reflects a more inclusive enterprise [107,71,143]. On the other hand, Agility as a concept is closely related to leanness [145]. Customer requirements require cost efficiency and faster turnaround from software industry. As already discussed, agile methodologies provide flexibility to embrace rapid changes. On the contrary, lean focusses on providing economically feasible solution and value-add for customer. The same is depicted in Figure 2.14 - Market winner and qualifier matrix [88].

| | Market Qualifiers | Market Winners |
|---|---|---|
| Agile | 1. Quality<br>2. Cost<br>3. Lead Time | 1. Service Level |
| Lean | 1. Quality<br>2. Lead Time<br>3. Service Level | 1. Cost |

**Figure 2.14: Winner/ Qualifier Matrix for Lean and Agile**

Furthermore, Agility in Software Engineering can be presented as "the continual readiness of an Information System Development to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" [33]. This also indicates concepts of Leanness which proves beneficial while conceiving the notion of Agile [103]. *Fagerholm et al.,* [45] suggest that merging Lean and Agile also affects the team's productivity and over all development process. However, combination of Lean and Agile for ERP Implementation is still an unexplored territory [147].

## 2.6 NEED FOR A NEW MODEL FOR ERP IMPLEMENTATION

After critical review of the existing ERP Implementation Models, Lean and Agile Methodologies, following gaps are identified which point to new model of ERP implementation

- A good extent of research has been done on using Agile for ERP [44,22,154,155,68,69,112]. These researches lack practicality by assuming ERP implementation as ERP Deployment and does not consider ERP Customization which is vital in any ERP implementation. However, no past study indicating application of Lean Thinking in ERP implementation has been done.
- Non-availability of literature that talks about the application of Agile Software Development coupled with Lean Thinking principle in ERP implementations and customizations, which always overrun schedule, budget and are unable to deliver the perceived value to customer.
- Some past researches suggest that Agile Methodologies will decrease Software Cost of Quality [57,142,69,13] but these researches don't produce any quantifiable result. Also, no past research to study the effect of Lean thinking on Software Cost of Quality has been done.
- *Noor and Khan* [98] and *Jayawickrama et al.,* [67] reviewed various past researches and concluded that Agile model is best suited for defect management. They also agree that more work is required to diminish defects at early stages. However, no past research to study the effect of Agile and Lean thinking on Software Defect Injection in normal as well as in ERP ambit has been done.

There are very few studies [142,144] which suggests that ERP Implementation and Customization should be considered as software development to ensure the successful implementation. Jalote [60] and Sommerville [129] have argued that measuring few parameters and taking measures to control these parameters during software development ensures the success of software project and software product. These parameters are given below –

1. Effort spent and deviation from planned effort at all stages of development and validation or verification of deliverable. The various stages are –
    a. Requirement Specification
    b. Design
    c. Construction and Configuration
    d. Integration testing ( SIT)
    e. User Acceptance Testing ( UAT)
2. Total defects in the application and defect distribution in various stages.
3. Defect Injection Rate i.e. number of defects injected per 100 person hours of effort.
4. Cost of Quality i.e. ratio of effort spent on finding and fixing the defect to the total effort spent in complete lifecycle or a stage of the project.

There had been many studies [4,5,6,10,17,26,36,40,56] in the past which evaluate the ERP Implementation and identify the success factors from management perspective. Few of the studies cite the ERP implementation using Agile [59,68,69,112,126,154,155] and few studies consider quality i.e. number of defects for successful ERP implementation [13,14,142]. However, there is absence of study that assures success of ERP implementation project.

A new framework called GenNext framework that combines the Lean Thinking and Agile Methodology is proposed in the chapter 4 of the thesis. GenNext framework includes feedback mechanism, ensures latest requirements are developed, reduces requirement volatility , reduces effort on nonvalue-adding activities , emphasizes on communication and reflects the learning of initial phases in to later phases. Thus, GenNext frameworks overcomes all the issues found in the models discussed in this chapter.

To ensure the efficacy of the GenNext framework hypotheses ( defined in section 1.4 ) were mapped to parameters defined by Jalote [40] and Sommerville [129].

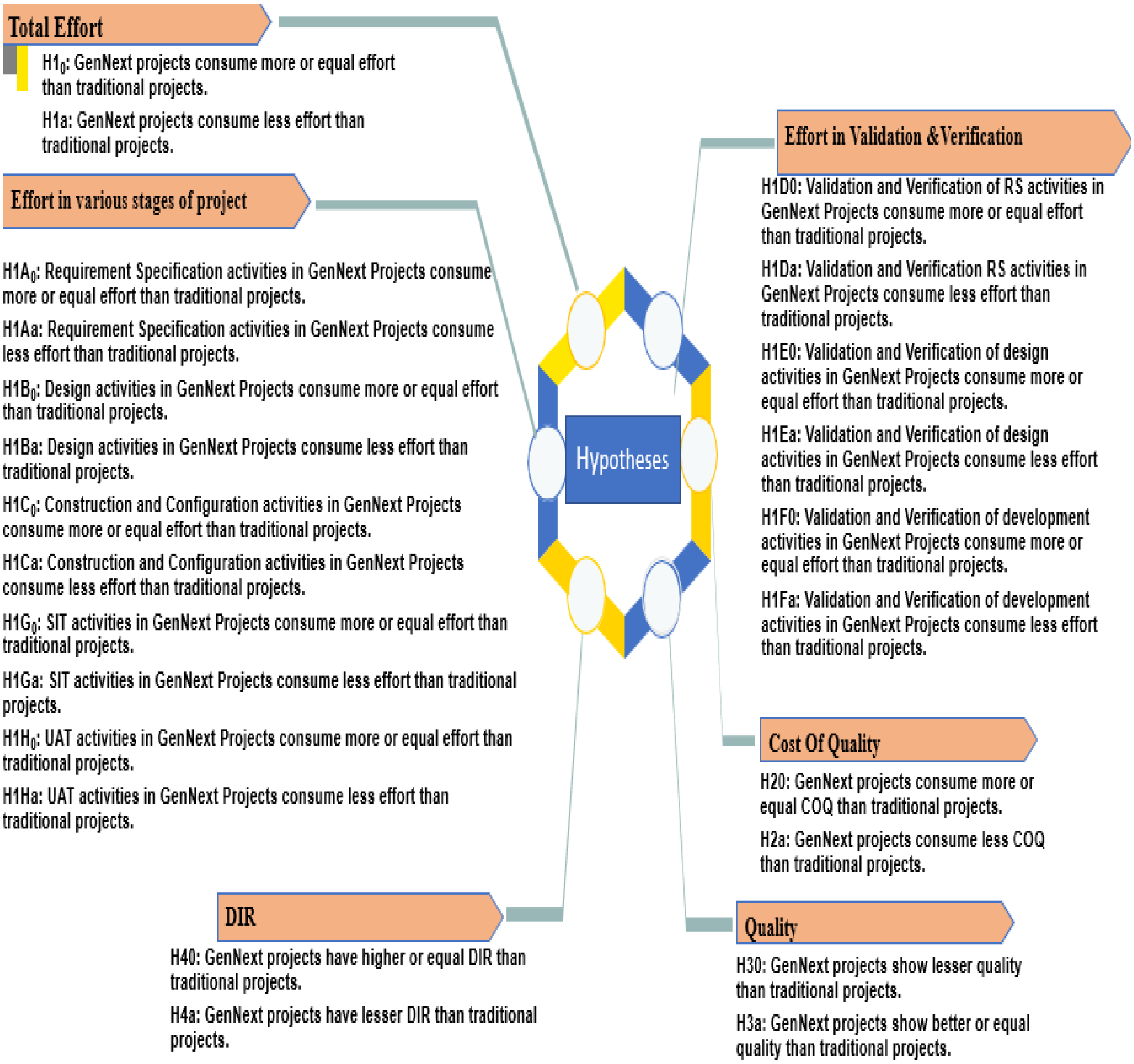


**Figure 2.15: Mapping of hypotheses to project parameters**

After applying GenNext framework, the results obtained for parameters discussed for software development are evaluated and statistically validated in chapter 5 of the thesis. The difference between GenNext and Traditional methodologies are discussed in detail in section 4.8.

Next chapter discusses about the roadmap and scheme of the research presented in the thesis.

# CHAPTER 3: RESEARCH METHODOLOGY

## 3.1 INTRODUCTION

The research was carried out in three phases using hybrid method of research. Hybrid method combines the strengths of qualitative research and quantitative research which develops a sturdies understanding of the study [35]. This section provides the high level and schematic structure of the research. The schematic framework of the research is given in Figure 3.1 and details are given below.

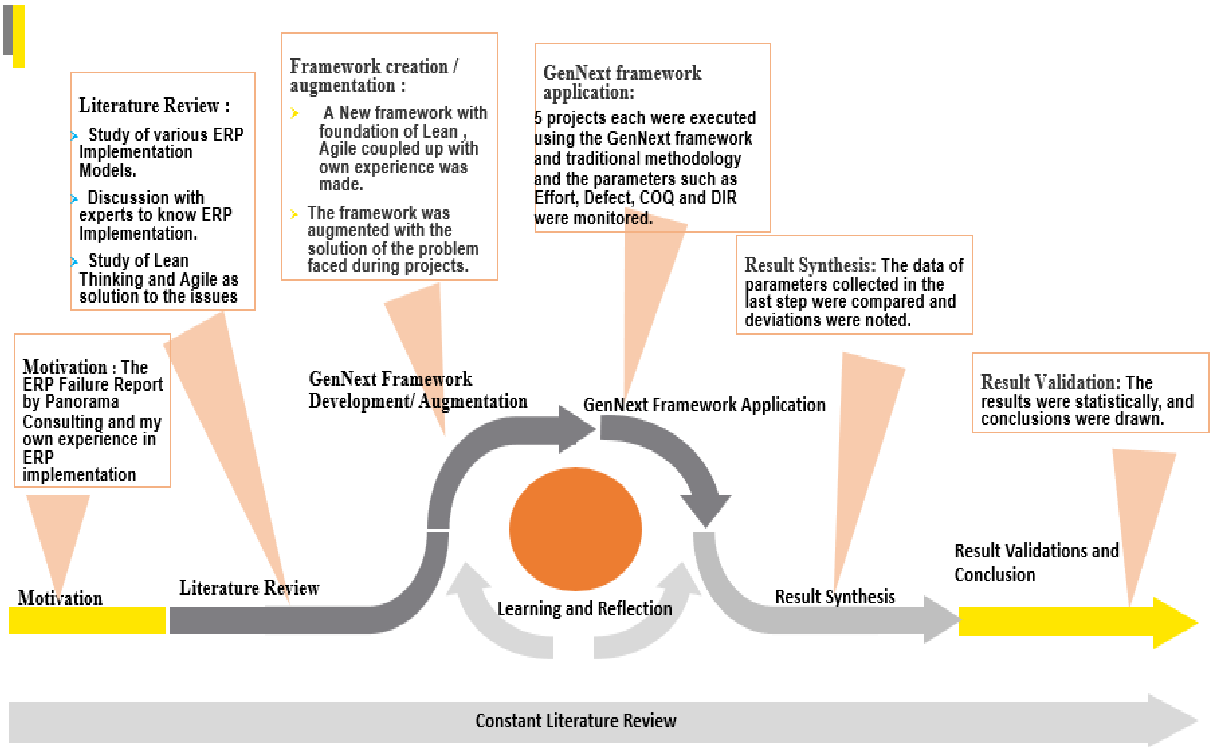


**Figure 3.1: Schematic flow of research activities**

The research was carried out with the motivation to solve an industrial problem of failing ERP implementations. The reasons of these failure were studied and analyzed through own experience; the literature reviewed for this study is presented in Chapter 2 of the thesis. A new framework , described in detail in Chapter 4 ,was conceived and applied to ERP projects to see if the new framework solves the identified issues. The results were analyzed and concluded (described in detail in Chapter 5) that new framework is able to solve the issues which were identified in chapter 2.

The research methodology and design are described in detail in further sections of this chapter.

## 3.2 RESEARCH DESIGN

The work on this thesis started with the aim of delivering ERP implementation project within defined cost, schedule, effort and quality parameters. The research process adopted is pictorially represented in Figure 3.2 and is explained below:

**Literature Research, Analysis Interaction with Industry experts –**

**a)** The reasons of ERP Implementation failure were studied through available literature and an exhaustive list of the issues was prepared.

**b)** The list of issues was further augmented with my own experience in ERP implementation.

**c)** Industry experts were contacted to understand the issues in ERP implementation that causes schedule overrun and cost overrun and increase in COQ.

**d)** Collected data was analyzed and issues were categorized in four categories viz.

  i C1-The problem / issue is mentioned by more than 50% of the respondents.

  ii C2-The problem / issue is mentioned by more than 35% of the respondents.

  iii C3-The problem / issue is mentioned by more than 25% of the respondents.

  iv C4-The problem / issue is mentioned by more than 15% of the respondents.

e) Existing ERP Implementation Models / framework were studied. The benefits and short comings of each model were compared, and a common gap was found out.

Lean Thinking and Agile Methodologies were analyzed for finding suitability to ERP Implementation. Both paradigms were compared to each other to find out the benefits and shortcomings of one over the other.


**Figure 3.2: Flow of research activities**

### 3.3 CREATION OF GENNEXT FRAMEWORK

After reviewing the available literature and assessing the need of a new framework – GenNext framework was created. The various inputs and steps involved in creation of GenNext framework is given below in Figure 3.3.

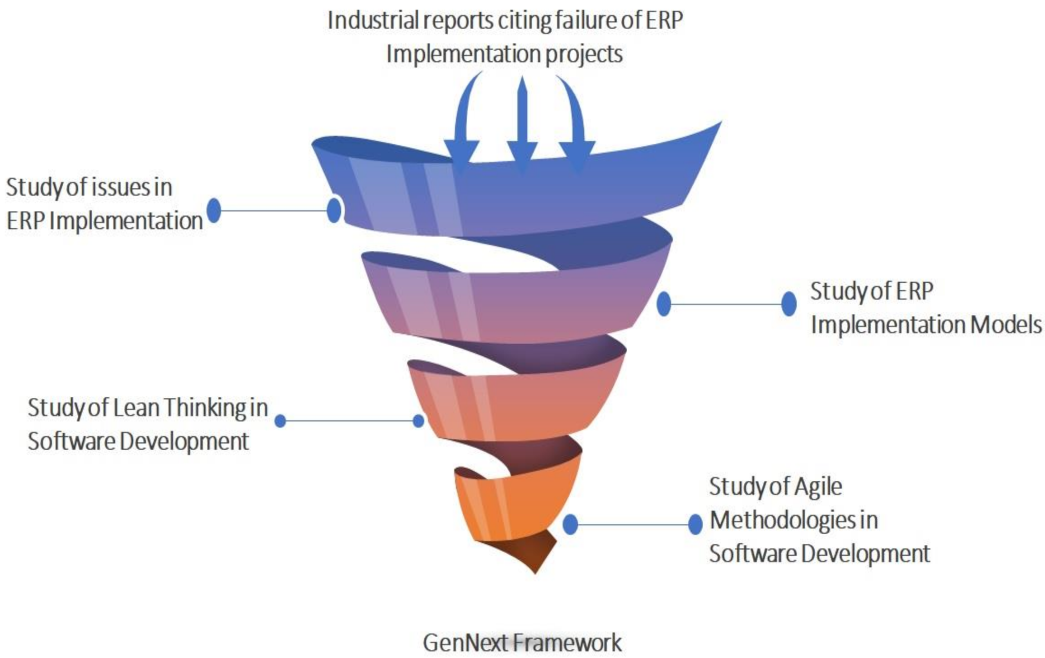


**Figure 3.3: Genesis of GenNext Framework**

The GenNext framework is discussed in detail in chapter 4 of the thesis. GenNext framework has the foundation of principles of Lean Thinking and Agile Methodologies. GenNext framework describes various roles and activities of each role and best practices. GenNext framework will prove to be beneficial for Project Sponsors, Program Managers, Project Managers and System Owners. GenNext will prove to lower TCO in long run with the given immediate benefits. The major benefits of the GenNext framework over traditional methodologies are -

1. Less development effort
2. Less testing effort
3. Less defects
4. Low cost of quality (COQ)

5. Low DIR or low defect density.

The GenNext framework has been defined in detail in Chapter 4: GenNext Framework of the thesis. This chapter defines various roles, activities, phases with input and output of each phase.

## 3.4 EXECUTION OF PROJECTS WITH TRADITIONAL METHODOLOGY

Five ERP Implementation projects were executed using Oracle AIM /OUM Methodology which is based on ERP Lifecycle Model hence referred to as the Traditional Methodology. I have played multiple roles ranging from technical consultant, functional consultant to Program Manager in multiple Oracle Apps ERP implementation projects and well versed with Oracle OUM/ AIM methodology. Hence Oracle AIM/OUM is used as traditional methodology and there is no other criteria or logic used in selection of the traditional methodology. Also, at the time of execution of projects and data collection for the research, ASAP 7 was predominantly used in the consulting community. These five projects were given the unique IDs as TP1, TP2, TP3, TP4, TP5 for uniquely identifying each project. These 5 projects are referred to as Traditional projects for referring in the rest of the thesis.

**<u>Scope</u>** – All the five projects had same requirement of implementing the following Oracle Apps Modules – Accounts Receivable, Accounts Payables, Cash Management, Fixed Assets, General Ledger, Inventory Project Costing, Purchasing. To ensure the similarity in solution, the customizations were also defined in the implementation and were same across the projects. The scope is defined in detail in appendix and brief overview of complexity is given below in table 3.1. The detailed scope of the projects is given in Appendix 2.

| Modules | Configuration | Customization | Grand Total |
|---|---|---|---|
| Accounts Receivable | 4 | 9 | 13 |
| Accounts Payables | 14 | 34 | 48 |
| AP and PO Integration | | 2 | 2 |
| Cash Management | 1 | 5 | 6 |
| Fixed Assets | 5 | 19 | 24 |
| General Ledger | 5 | 1 | 6 |
| Inventory | 8 | 23 | 31 |
| Project Costing | 7 | 14 | 21 |
| Purchasing | 6 | 31 | 37 |
| **Grand Total** | **50** | **138** | **188** |

**Table 3.1: Configurations and Customizations identified in traditional projects.**

**<u>Participants</u> -**These five projects were used in training of the inhouse consultants, cross training the senior members such as Senior Consultants, Project Managers, Program Managers in Oracle Apps ERP and existing Oracle Apps competency members who feel to refresh the knowledge.

The parameters such as effort in various phases and activities, defects, DIR, COQ and variances were collected for the projects. Also, the deviation was calculated. The deviation was calculated as ratio of difference between planned value and actual value to planned value.

### 3.5 APPLICATION OF GENNEXT FRAMEWORK

GenNext framework was applied to five projects. These five projects were used for training new implementation consultants and cross training existing ERP implementation consultants. These five projects were given unique IDs as GP1, GP2, GP3, GP4, GP5 for uniquely identifying each project. These 5 projects are referred to as GenNext projects for referring in the rest of the thesis.

**<u>Scope</u>** – All the five projects had same requirement of implementing the following Oracle Apps Modules – Accounts Receivable, Accounts Payables, Cash Management, Fixed Assets, General Ledger, Inventory, Project Costing, Purchasing. To ensure the similarity in solution, the customizations were also defined in the implementation and were same across the projects. The scope is defined in detail in the appendix and brief overview of complexity is given below in table 3.2. The detailed scope of the projects is given in Appendix 2.

| **Modules** | **Configuration** | **Customization** | **Grand Total** |
|---|---|---|---|
| Accounts Receivable | 4 | 9 | 13 |
| Accounts Payables | 14 | 34 | 48 |
| AP and PO Integration | | 2 | 2 |
| Cash Management | 1 | 5 | 6 |
| Fixed Assets | 5 | 19 | 24 |
| General Ledger | 5 | 1 | 6 |
| Inventory | 8 | 23 | 31 |
| Project Costing | 7 | 14 | 21 |
| Purchasing | 6 | 31 | 37 |
| **Grand Total** | **50** | **138** | **188** |

**Table 3.2: Configurations and Customizations identified in GenNext projects.**

**Participants** **-**These five projects were used in training of the inhouse consultants, cross training the senior members such as Senior Consultants, Project Managers, Program Managers in Oracle Apps ERP and existing Oracle Apps competency members who feel to refresh the knowledge.

The parameters such as effort in various phases and activities, defects, DIR, COQ and variances were collected for the projects using GenNext framework. Also, the deviation was calculated. The deviation was calculated as ratio of difference between planned value and actual value to planned value.

### 3.6 VALIDATION OF FRAMEWORK AND RESEARCH SYNTHESIS

- The parameters such as effort in various phases and activities, defects, COQ and variances were collected for the projects executed using Traditional framework and executed using GenNext framework.
- Results of projects using GenNext Framework and traditional framework were compared.

- Hypotheses, defined in section 1.4, were verified statistically using ‘t-test’ and conclusions were interpreted.

## 3.7 CONCLUSION

The research methodology presented above helped and provided directions in targeting the specific problems in ERP implementation, finding out the solution of these issues , deducing a new framework using Lean and Agile and critically evaluating the results of application of framework in ERP implementation projects to draw a conclusion that Lean and Agile can be used to ensure success in ERP implementations.

The next chapter discusses the proposed framework, referred to as GenNext framework here after, in detail. The results obtained for projects using traditional methodologies and GenNext framework are discussed and assessed in Chapter 5 of this thesis.

# CHAPTER 4: PROPOSED FRAMEWORK

## 4.1 GENNEXT FRAMEWORK

To meet the objectives and solve the identified issues a new framework called as GenNext was created using Agile Methodologies and Lean Thinking. The new framework considers important and practical aspects of Lean and Agile to solve the problems, issues and increase the efficiency of ERP Implementations. Though, framework has been created primarily for ERP Implementations, but it can be used for other Enterprise Implementations such as BI Reporting or any other COTS implementations.

## 4.2 VALUES / PILLARS OF GENNEXT FRAMEWORK

GenNext Framework, like any other framework is based on few foundation pillars. These foundation pillars are given below in table 4.1 –

| PILLAR ID | DESCRIPTION |
|---|---|
| P1 | Continuously optimize the flow and improve value of the system. |
| P2 | Manage change and respond to pulls. |
| P3 | Relationship by co-development and delivery of working software. |
| P4 | Product success over functional success. |
| P5 | Trustworthy, empowered and self-managing team. |

**Table 4.1: Pillars of GenNext Framework.**

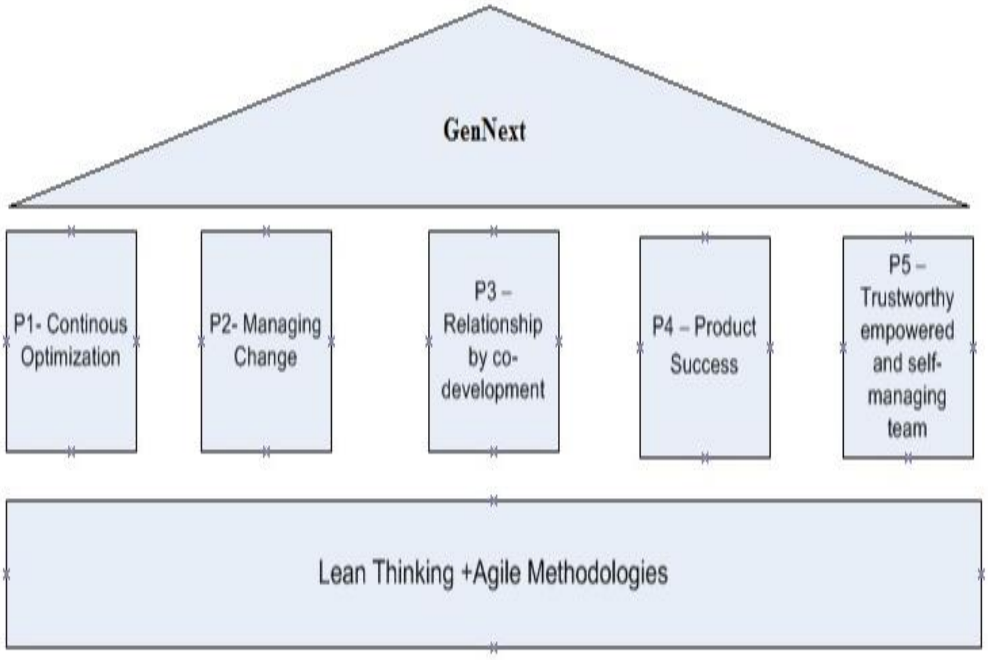


**Figure 4.1: House of GenNext Framework.**

**P1 - Continuously improve and optimize the flow and value of the system**

The core idea behind the Pillar 1 of the house of GenNext framework is to optimize the whole system and to reduce non value adding activies. The whole sysem should be considered for optimization (LT07). The opimization can be achieved by eliminating the waste (LT01) and maximizing the value with reduction in non value adding activties (LT01) by innovation and continuous improvement. Innovation comes from better understanding of the customer needs, wants and existing processes (LT02). Understanding thus recevied helps in continuous improvement and adjusting the behavior to a request or situation. The learning should be reflected continuously through out the project (AM12). Improvements cannot be proposed unless the flow of whole system ,value of the system and processes is known. Thus, whole system should be considered to identify bigger chunks of waste (LT07) to optimize the whole value chain than suboptimizing the subprocesses.

**P2 – Manage change and respond to pulls.**

Most often, the duration between requirement gathering and the delivery is huge which makes the requirement stale or not holding good by the time system is delivered. Way of doing business changes frequently and business users want those changes to be pulled in the system. This principle emphasizes on accepting the customer pull which is inline with AM02 of Agile methodologies. Developers want to include the changes in the system but, most of the time, are unable to accept such requests owing to structural design changes. Developers are suggested to keep the design simple (AM10) and options or have work around for requirements. Simplicity not only removes the non-value adding activities/ functions, thus waste, but also provides the efficiency of getting maximum output and at the same time it provides flexibility to change as per the new requirement. Provisions should be made to accommodate such pulls and variability should be managed internally by team (LT03). Customer pull can range from requirement changes to defect fixes of early deliveries. Pulls should be responded in a way that these are managed and all deliveries happen at a constant pace and velocity (AM08).

**P3 – Relationship by co-development and delivery of working software**

ERP or any COTS implementations cannot be done in silos. Requirements finalized in preparation and blueprinting phase are usually changed/dropped during the realization or UAT phase. System, processes usually have gaps and system integrators, sometimes, have gap in understanding the system. These gaps are filled by making few assumptions. These assumptions need to be validated and verified at each stage for each communicated requirement and changes. All stakeholders should have responsibility to co-create the system by providing their feedbacks and inputs. These feedbacks or inputs should be free from bias or fear. This bias or fear can be eliminated if all the members involved in development of system share a good relationship and share sense of ownership. The understanding becomes better when people talk to each other in meetings and convince each other about the importance of change (AM06). The relationship between all stakeholders increases when people understand mutual pain and value (AM03).The relationship should work toward releasing delivering the new version of software and feedback received should be included in the next deliveries (AM03).

**P4 – Product success over functional success**

An ERP implementation should be considered as a product roll out than a program involving various teams. Various teams such as Functional Analysts, users, developers and program owners should look ERP implementation from their perspective and other members' perspective to impart their level of excellence to the system with an integrating mindest. All the teams should have one target to achieve product success, product excellence (AM09) with the desire of making product right at first time (LT06).

**P5 –Trustworthy empowered and self-managing team**

Development team, business users and stakeholders should be able to deliver the solution in least directions, guidance and should be able to decide directions to provide solutions to achieve the final success. Teams should be entrusted to decide the priority, design and schedule of delivery. Entrusted teams consisting of motivated individuals, usually, come up with best solution and alternatives (LT05,AM11). They pull the requirements, deliver them and set the rhythm to complete on time. Entrusted and self-organizing team create a conducive environment of learning from mistakes and deliver at high efficiency and least defects.

## 4.3 GUIDING PILLARS OF GENNEXT FRAMEWORK

Following section defines the guiding pillars of GenNext framework. Each guiding pillar is given a unique ID to be referred.

| ID | PRINCIPLE |
|---|---|
| GP-1 | Satisfied Customer |
| GP-2 | Lever to change requirement even in later stages of implementation. |
| GP-3 | Frequent and constant paced deliveries. |
| GP-4 | Eliminate non-value adding activities and wastes |
| GP-5 | Cohesive cooperation between users and developers during the complete implementation cycle. |
| GP-6 | Self-organizing team of motivated individuals and respect for everyone |
| GP-7 | Seamless communication |
| GP-8 | Measurement by deliveries made. |
| GP-9 | Technical and qualitative excellence. |
| GP-10 | Keep everything simple. |
| GP-11 | Contious refelection of Amplified learning in the work and delivery. |
| GP-12 | Optimized flow. |
| GP-13 | Create a Pull Environment. |

**Table 4.2: Guiding Pillars of GenNext framework.**

## 4.4 BUILDING BLOCKS OF THE GENNEXT PROJECT

The scope of an ERP Implementation Project is defined by high level requirements, initially,and are blown up further to get the detailed requirements. The requirements thus collected are compiled in **Requirement Register**. The requirement are divided into Processes and Features. The processes and features are delivered incrementally and iteratively in a Lap. Thus, at the end of the Lap features and processes are delivered to ensure the incremental value addition instead of piling of components. This ensures a frequent feedback and a glimpse of customer's vision which helps in maximising the possibility that product / ERP system meets customer requirements. Following are the terms, given in table 4.3, used as building blocks in GenNext framework.

| TERMS | DESCRIPTION |
|---|---|
| Requirement Register | The list of all product features with appropriate prioritization and complexity or effort rating. This list includes the epics contained therein where already defined. |
| Feature | A group of requirements, from a business perspective, whereby each feature is on a very high level of abstraction and are described concisely in a short paragraph. The description is sufficient for experts to assess the complexity and thus also the scope of work. Example of features–<br>1.Ability to reserve material from available stock.<br>2.Ability to check material in stock.<br>3.Ability to calculate the time of making an item from raw stock<br>4.Ability to calculate the Lead time and Lag time for items to be bought or to be moved from one inventory location to another inventory location.<br>5. Ability to move the material from one sub inventory to another sub inventory. |
| Sub Process | Sub process can be defined as a significant portion or stage of a process that accomplishes a goal. A Sub process is usually a group of features. Example –<br>1. Ability to provide a date based on the item type using the feature to calculate the manufacturing time and reserve the material.<br>2.Ability to provide a date based on the item type using the feature to check the item in its stock, in near-by stocks and calculating the time of availability using feature to calculate the lead/ lag time of movement. Also, reserving the material. |

| TERMS | DESCRIPTION |
|---|---|
| Process | A specific functionally independent requirement. It includes an acceptance criteria or condition of satisfaction to check the correct functioning of this requirement. A Process is an Independent, Negotiable, Valuable, Estimable, Small, Testable requirement. Example –<br>1. Order Sceduling Process which provides a scheduled ship date for all item types. It internall calles sub-processes that decides the 'pick','make' or 'buy' based on the date retreived from stock avaialbility in local inventory or time elapsed in movement from near-by inventory loaction, time elapsed in manufacturing and reserves the material.<br>Above example of process has included all the subprocess and features defined in above definitions. |
| Epic | Group of related processes. Example –<br>Order Management Epic that includes the Order booking process, Order scheduling process, Pick Release process and Shipping the material process. |
| Lap | Basic unit of development including configurations. It is a time-boxed activity and results in single or multiple useable piece of product / system, added incrementally to system. |
| Galop | Atomic division of an activity in a Lap or an activity that produces atomic subpart of piece to be delivered in a Lap. The atomic parts add some value to the whole system and integrated to test environment on regular basis. |

**Table 4.3: Terms used in GenNext framework**.

## 4.5 ROLES AND RESPONSIBILITIES

GenNext project team includes all those people who are necessary to ensure that the backlog items are transformed into delivearable software. GenNext team like any other software development team consists of the crossfunctional team members,Architects, Business Analysts, Developers, Tester ,Performance Analyst and PMO / Compliance team. The depiction of GenNext team is given in Figure 4.2.The team members estimate the extent of each product requirement register item to be delivered and communicate to the product owner.

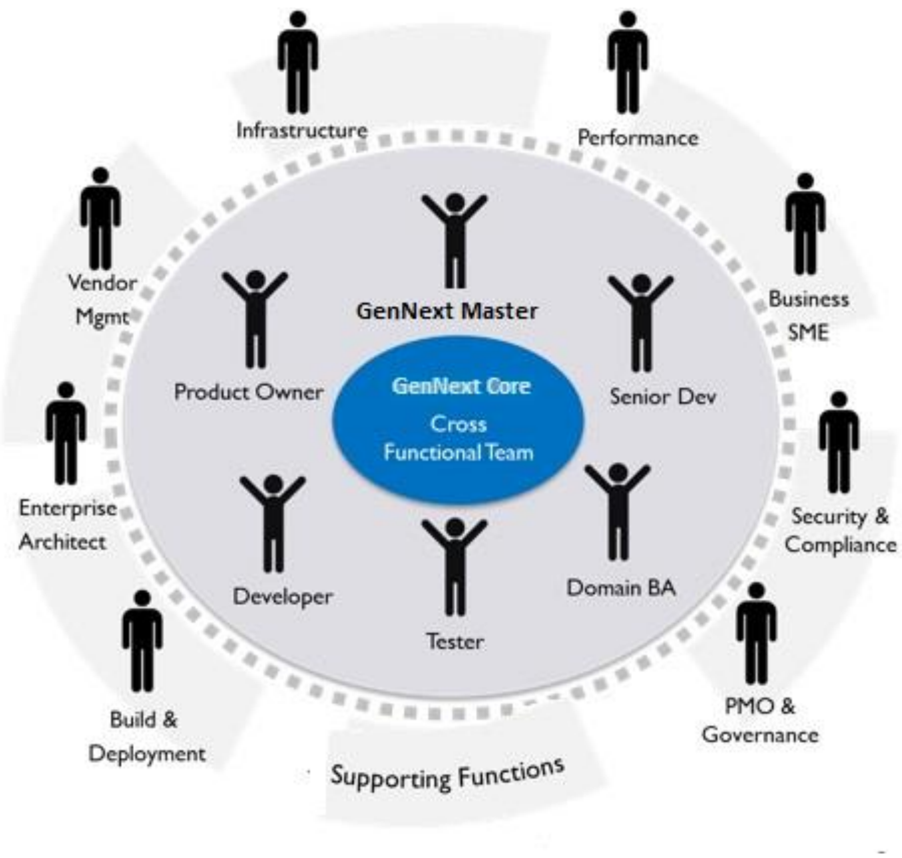


**Figure 4.2: GenNext Team.**

**Product Owner**

- Leads the vision to the team
- Drives and negotiates the Business Value

- Builds and refines the product requirement register.
- Works continuously on the Epic and the Lap plan.
- Steers the product development and is responsible for ensuring that the team develops the desired functionalities in the correct order
- Works daily with the team and takes all the necessary decisions in a timely manner
- Ensures that the project results justify the financial investment for the project
- Manage stakeholders and interests ‘Voice of the Customers’.
- Optimizes the total cost of operation.
- Defines the Done Criteria
- Authorizes to accept the deliverables

**GenNext Master**

- Process Coach and Facilitator
- Driving /Allowing the team to self-organize to get the work done
- Encourages collaboration across teams
- Radiates information and reduces risks
- Supports the team by removing impediments
- A change agent
- Helps the team achieve its goals
- Educates all persons involved in the project so that they can understand and carry out their roles
- Guides continuous improvement of team performance
- Protects the team from outside interferences to ensure that their productivity is not affected
- Creates transparency

- Improves stakeholder relations

**GenNext Team**

- Team Size in the range of 10 +/-2
- Mix of skills representing multiple disciplines
- All roles (Designers, Technical Consultants, Functional Consultants, BA, DBAs, Developers, Testers, UX, System Engineers etc.)
- Takes up only processes and features which meets the Definition of Ready
- Participates in refinement sessions
- Self-organization & Cross functional
- Shares updates and raising impediments
- Updates desired tool/Visual Board daily to reflect progress
    - Ensures Delivery compliance to Definition of Done
- The team owns delivery, controls the amount of work that it can handle and therefore accepts responsibility for the quality of the delivery.

**Users & Stakeholders**

- Derives value out of system being developed but not responsible for deliverables
- Participates in Release Review

## 4.6 PHASES OF GENNEXT FRAMEWORK

GenNext Framework works by dividing the whole implementation process in three major phases which are as follows and depicted in figure 4.3–

1. Propose
2. Construct and Configure
3. Delivery and Maintenance

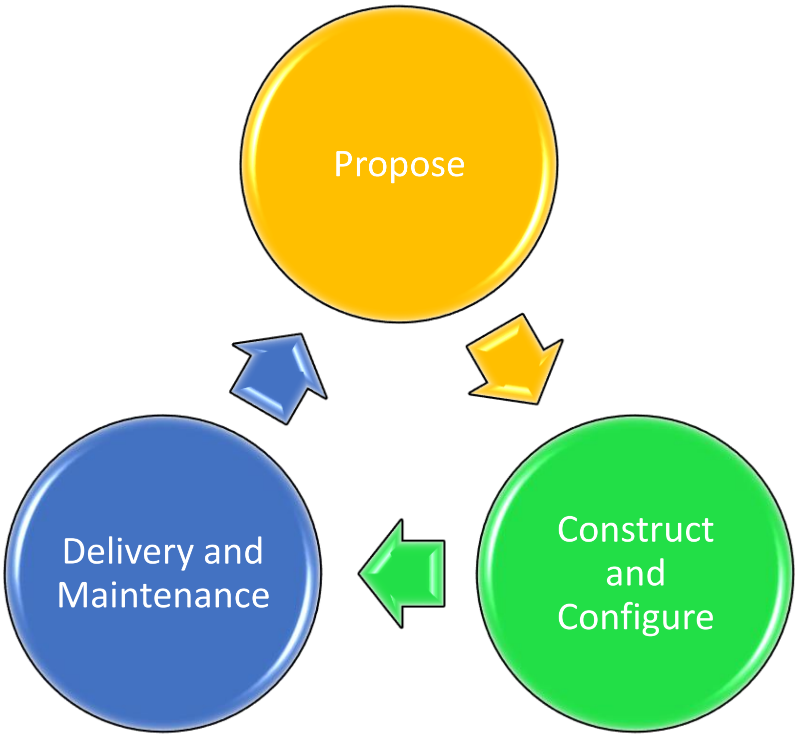


**Figure 4.3: Phases in GenNext Framework.**

In an ERP implementation, these three phases may be executed sequentially or in parallel. Usually, requirements are identified, developed and delivered. However, any new process or requirement identified during the construction phase will add to propose phase and triggers re-prioritization. Figure 4.3 depicts that any learning or any feedback from user or experience is considered while working on new requirements or developments.

### 4.6.1 PROPOSE

Proposing or Planning is an essential activity usually done in start of any project. Planning shows the intention that how a project would be handled and steered to ensure the deliveries. Planning sets the priority and pitch of the project. This stage can be defined as collating the ideas from conception to materialization. Planning activity can be divided in following sub phases –

- **Prepare –** Objective of this sub phase is to identify the scope of the project and listing all the processes including the As-Is / To Be Analysis. All the requirements , at high level ,are stored in requirement register. The interdependancy between the process and integration with external interface is also noted during this subphase.
    - **Tools –** 5S, Brain Storming , Gap Analysis and Expert Judegement,Gemba
    - **Roles –** Product Owner, Process Owner,GenNext Master and GenNext Team ( Functional Consultants).
    - **Input –**None.
    - **Output –** Requirment register with standard configurations and process flow based on the project charter.
    - **Activities**
        - Identification of the Product Owner and various process owners.
        - Product owner calls up for a workshop with process owners and GenNext Master to describe and understand various processes and features.
        - Identification of interdependancies and alternatives of processes and features.
        - GenNext Master should enable Process owner to use 5S to analyze the Process and features to weed out any wasteful activity.
    - **Best Practices**
        - GenNext Master and Product Owner should crate a requirement register before meeting.
        - Set a template and circulate the template to all Process Owners. All Process Owners should fill and send this back to Product Owner and GenNext Master atleast 2 days prior to meeting.
        - A review meeting should be done after capturing all the informations to understand and get concensus by all the Process Owners.

- GenNext Master and Product owner should use visual digram defining the process to be created to identify any missing process.
- Proess owner and GenNext Master should visit the site following the Gemba tool and should see the process by themselves. This would reduce the chance of missing any wasteful activity.
- Process owner and GenNext master should do a Gap analysis and use expert judgement in case of any major gap in standard ERP process and actual process.
- GenNext master should record the requirements in requirement register and should capture business value, atlernatives of the process and features.
- GenNext master should ensure that all process and features to be configured or customized are captured in the requirement register.
- GenNext master and Product Owner should keep on updating the requirement register with new or changed requirements. In case of changed requirements, development should consider the latest requirements to start with. Any changes to the requirements being developed should be tried to include in current Galop / Lap or should be added to requirement register.
- GenNext Master should capture the Non Functional Requirements (NFR) and these NFR should be treated as feedback or learning and should be included in all future deliveries.

- **Estimate –** Objective of this subphase is to get the approximate effort for carrying out the activities. Different organizations and researchers have devised many number of estimation techniques. Estimation technique, as agreed with customer, should be used and should be provided the results in Person Hours. The effort estimate can be drawn from velocity of similar projects. The velocity may vary from team and enterprise level.

- **Tools –** Expert Judgement or Organization estimation template.
- **Roles** – Product Owner, Process Owner,GenNext Master, GenNext technical team members.
- **Input –** Requirement Register without estimates.
- **Output –** Requirement Register with high level estimates.
- **Activities**
    - GenNext Master and GenNext team should ensure that all the processes, subprocesses and feature have high level estimated.
    - Use commonly accepted estimated guidelines and provide high level estimates for a sub process or a feature.
    - Roll up the efforts of all subprocess and features to the process level and get effort estimate for the process.
- **Best Practices**
    - GenNext Master and Product Owner should leverage the consultants' previous experience or internal IT team's experience of estimation.
    - GenNext Master and team should use only one methodology or template to estimate all the efforts. This reduces chance of error within the requirement register.3 point estimation method should be used to rule out aggressive or passive estimates.
    - GenNext Master and team should keep on estimating the the effort of new requirements and should keep on repriortizing the requirements.
    - GenNext master should tune the estimation technique based on the deviation from planned effort.
    - GenNext master should consider first 3-4 galops and laps to get the idea of average velocity.

- **Prioritize-** Objective of the subphase is to rank the process or requirements in an order based on the Business value, effort , alternatives and reusability. Based on self experience, I have proposed to divide the process into sub process, features and get the Process Priority Number or Feature Priority Number by applying the priortization algorithm which is defined in this section.

    - **Tools –** GenNext Priortization Algorithm ( described in detail in this section).

      ERP consists of processes. Each process is made of sub-process which in turn is made up of features or functionalities. The purpose of the algorithm is to categorize the requirements into a prioritized sequence based on business value, effort and alternative available. Following terms and considerations should be understood before understanding the algorithm.

        - **Business value** is usually perceived as user value or ROI (return of investment). In the start of the project, the requirements are known at higher level and the business value is always a perceived or notional value and may not be the exact dollar value of the ROI of the process.
        - **Effort** to configure or code the process in the application plays a vital criterion. Processes of high business values and low effort are delightful and should be given higher priority over the others.
        - **Reusability** of the features and functionalities or even the process cannot be denied in application. Usually, applications have the processes which are reusable in the other processes too. The Reusability is calculated as R = n+1, where n is the number of times feature is being reused in the application. For example, if feature F1 is reusable in 2 process than the reusability R will be 2+1 = 3

- **Alternatives –** Sometimes processes or functionalities can be achieved through other means without development of current functionality. For example, an order booking button and Order Booking Batch does the same work. However, Order booking button does the operations real time and Batch does it on frequency of 5 minutes, but the process does not stop, and Order Booking Button has alternative. The Alternative value is calculated as A= n+1, where n is the number of alternatives available.

The prioritization algorithm is based on the philosophy that processes with features which have high business value and low customization / configuration effort should be given the priority over others. It is based on the user experience. Hence, Feature Priority Number (FPN) for a feature or functionality in a process can be –

$$FPN \propto \text{Business Value} / \text{Effort} \qquad \text{(Eq 1)}$$

Based on my own practical experience in IT industry, a feature which can be reused in multiple processes or sub-processes should be given high priority over others. Also, if a feature can be available alternatively, and can be achieved without development of the functionality in other process, should be given low priority over others with no alternative available. Thus,

$$FPN \propto \text{Reusability} / \text{Alternatives} \qquad \text{(Eq 2)}$$

With Eq 1 and Eq 2, it can be easily deduced as –

$$FPN = k \frac{(\text{Business Value} * \text{Reusability})}{(\text{Effort} * \text{Alternatives})} \qquad \text{(Eq 3)}$$

Where k is coefficient of proportion. For the simplicity, k is assumed 1. Thus,

$$FPN = \frac{(\text{Business Value} * \text{Reusability})}{(\text{Effort} * \text{Alternatives})} \qquad \text{(Eq 4)}$$

As discussed earlier, a sub-process is made of feature and functionalities. Hence the Sub Process Priority Number (SPN) can be obtained by adding the values of FPN for all features and functionalities in a sub process. Assume, there are *l* number of features in a subprocess and each feature will have an FPN. The *l* is a variable and can have any positive whole number.

$$SPN = \sum_{n=1}^{l} \text{FPN} \qquad \text{(Eq 5)}$$

Thus, Process Priority Number (PPN) for a process can be obtained by adding the values of SPN for all sub process in a process. Assume, there are $m$ number of subprocess in a process and each feature will have an SPN. The $m$ is a variable and can have any positive whole number.

$$PPN = \sum_{n=1}^{m} \text{SPN} \qquad \text{(Eq 6).}$$

Hence, the proposed prioritization rule is "To prioritize the processes with the highest ratio of importance to actual effort will be prioritized first and skipping processes or feature that are "too big" for current release".

Let

Set of processes -$\Psi = \{\Psi_1, \Psi_2, \Psi_3, \ldots\ldots\ldots, \Psi_n\}$

For each process $\Psi i$, there are following sub processes

$\sigma_i = \{\sigma_{i1}, \sigma_{i2}, \sigma_{i3}, \ldots\ldots\ldots, \sigma_{in}\}$

For each subprocess $\sigma_{ij}$, there are following features

$\Phi_j = \{\Phi_{j1}, \Phi_{j2}, \Phi_{j3}, \ldots\ldots\ldots, \Phi_{jn}\}$

$\beta$ → Business Value

$\rho$ → Reusability

$\varepsilon$ → Effort

$\alpha$ → No of Alternative

$\Pi$ → Process Priority Number for a process.

$\delta$ → Sub-process Priority Number for a feature in sub process.

$\tau$ → Feature Priority Number for a feature in sub process.

l → Number of features in a subprocess

m → Number of subprocesses in process

n → Number of process to be priortized

**Input:** Unprioritized requirements of ERP Processes to be implemented.

$\Psi = \{\Psi_1, \Psi_2, \Psi_3, \ldots\ldots\ldots, \Psi_n\}$

**Output:** Prioritized requirements of ERP Processes to be implemented.

$\Psi' = \{\Psi'_1, \Psi'_2, \Psi'_3, \ldots\ldots\ldots, \Psi'_n\}$

**Start Prioritization of the requirements**

**Step 1: Identify the Processes to be prioritized in order.**

List out all the processes, sub processes in a process and features in the sub process.

**Step 2: Calculate the FPN, SPN and PPN.**

```
for (i=1; i <= n; i++)
{
    for (j=1; j <= m; j++)
    {
        for (k=1; k<= l; j++)
        {
            //Calculate FPN
            τijk = (βijk * ρijk) / (εijk *αijk)
    }
        // Calculate SPN
        δij = τij1 + τij 2 + τij3 ........................ + τijm
    }
    // Calculate PPN
    Πi = δi1 + δi2 + δi3 +................................. + δij
}
```

$$\tau_{ijk} = (\beta_{ijk} * \rho_{ijk}) / (\varepsilon_{ijk} * \alpha_{ijk})$$

$$\delta_{ij} = \tau_{ij1} + \tau_{ij2} + \tau_{ij3} \ldots + \tau_{ijm}$$

$$\Pi_i = \delta_{i1} + \delta_{i2} + \delta_{i3} + \ldots + \delta_{ij}$$

**Step 3: Order the process based on PPN.**

```
for (i=1; i <= n; i++)
{
    Order the Ψ = {Ψ1, Ψ2, Ψ3,…, Ψn} based
    on   the Πi
}

Result: Ψ′ = {Ψ′1, Ψ′2, Ψ′3,……, Ψ′n}
```

- **Roles –** Product Owner, Process Owner,GenNext Master
- **Input –** Unpriortized Requirement Register.
- **Output** – Priortized Requirement Register.
- **Activities**
  - Priortization and repriortization of the Requirement Register by GenNext Master and Product Owner.
  - Product Owner and GenNext Master should verify that one process maps to only one business process. Incase one process is mapped to two business processes , break the process to have one to one mapping and repriortize the Requirement Register.
- **Best Practices**
  - Product Owner and GenNext Master should ensure that a process or a sub process be considered as alternative of any process or sub process if it does all the intended functionality of the process.
  - Incase no process is found to be alternative of a process, part of processes which can be reused for the process should be searched.
  - Incase of a tie in PPN, the PPN of up stream and down stream should be considered. Process with higher sum of PPN ( Upstream,Down stream and the process) should be given precedence over the other.

Once the requirement register has initial priortized requirements, GenNext master and team can start working on construction and configuration activities.

### 4.6.2 CONSTRUCTION AND CONFIGURATION

This phase can be summarized as continuous construction or configuration of the defined requirements. This phase encompasses analyze, design, development, integration and testing and execute these as a continuous activity, not as a sequential process as done in traditional models.

Purpose of this stage is not only deploying the release but to ascertain that release is deployable. This stage is all about preparing for release, including final documentation; pre-release staged testing and releasing the product to end users.

The implement phase consists of three sub phases viz.

- **Release Planning and Development-** The main purpose of this sub phase is to plan the release, design the architecture or choose a way by minimally disturbing the architecture and developing the application. The phase also serve as input to the requirements in case a new requirement is discovered during the progressive elaboration phase. This subphase helps us not only include development but also a feedback mechanism.

    - **Tools** – Expert Judgement, Bottleneck Analysis, Heijunka, Jidoka
    - **Roles** – Product Owner, Process Owner,GenNext Master, GenNext technical team members ,business users
    - **Input –** Requirement Register.
    - **Output –** Lap Plan and Galop plans,Configured and customized system, CM Plan, Test plan and stratgey , Master Component List, Dependancy Structure Matrix, Resource Loading Sheet
    - **Activities**
        - GenNext master to do high level planning for Laps and Galops in a Lap. Plan conveys what all activities to be in stipulated time.
        - GenNext Master to create Lap Plan.
        - GenNext Master and Product owner to do instance planning and set up of various environments and servers for various activities in a Lap.
        - GenNext Master and Product owner to discuss with DBAs and set up the cloning frequency and sequence of instance clonning.

- GenNext Master to set up the defect and bug tracking mechanism.
- GenNext Master and Product owner to set up the code storage and configuration management planning for various configurable items.
- GenNext Master and Product owner to identify and understand the architecture and efforts should be put to make sure that existing architecture is not disturbed.
- GenNext Master and Product owner to set up the test stratgey.
- GenNext Master to refine and deep dive into process to be implemented in a Lap.
- GenNext Master to create and update the dependancy struture matrix.
- GenNext team to prepare acceptance test cases and integration test cases.
- Configure the standard processes and and construct the customization in a Lap or Galop.
- Continuously deploy and integrate the new build and test the build.
- Get the required number of resources for various tracks and activities.

- **Best Practices**
  - GenNext Master should consider first 3-4 galops and laps to get the idea of average velocity to ensure Heijunka.
  - GenNext Master should explore the accelerators , automated deployment of code review, deployment , testing of document creation.
  - GenNext Master should create Lap plans rolling into Epic implementation plan.

- GenNext Master should plan atleast following environment and instances – Development, SIT, UAT and Prod instances.
- GenNext Master should  ensure that UAT instance for one lap should be used as the SIT instance of the other lap for continous integration and testing.
- GenNext Master along with team should vertically slice the process and features to identify the work to be done in all layers. Incase of any isssue arising owing to this, GenNext Master should repriortize the process and update the DSM.
- Processes should be refined during the Lap planning and Epic back log should be updated back and repriortized. In addition, dependancy structure matrix should be updated back.
- Deployment, testing and acceptance should be shared responsibility of team.
- Code deployment, integration stratgey is to be to built initially.
- Ensure that developers pick task at their own.
- Automate the code deployment and integration.
- Developers should test their code proactively before flagging the code for testing.
- Templates should be used to make sure that previous defects don't occur.Templates should be updated to reflect the learning.
- GenNext team and Master should learn from mistake and learning should be reflected in all future Laps and Galops.

- **Monitoring and Control –** The Monitoring and control subphase is to capture and control the vital statistics such as cost, quality and effort. In the GenNext Framework monitoring is done from the customer / end user perspective and effective measure are taken to control deviations. This sub phase includes the usage of visual controlboards and daily clarification meeting with time boxing of the development or configuration. At the end of each Lap a review and demo is conducted. The main purpose of the review is to retrospect and identify the points which could have been better and reflect the knowledge thus gained in next iterations.

- **Tools –** Expert Judegement, Bottleneck Analysis, Heijunka, Jidoka
- **Roles –** Process Owner,GenNext Master, GenNext technical team members.
- **Input –** Requirement Register.
- **Output –** Lap Plan,Galop plan, Defect Tracker , CM Plan,
- **Activities**
  - GenNext Master, team, Product Owner, Process Owners should have Galop Meeting / Lap meeting in start of Lap or Galop **to** discuss process and feature to be delivered are discussed and finalized in the meeting. Acceptance criteria is also defined in the initial Galop or Lap meeting.
  - GenNext Master and team **to** decompose all the process and feature into tasks. Assign responsibility and accounatbility to different member.
  - GenNext Master and team to have daily stand up meeting to understand how much has been done.
  - GenNext Master and team to find out the impedents of daily tasks and get support to remove them. Business users should be involved to get clairification on processes and feed back.
  - Have a mid Lap meeting with QA team to understand how much and what quality has been delivered as of now.

        - Measure the quantity as how much has been delivered actually vs how much was planned.
        - Measure quality of Lap by passed test cases by delivery.
        - Use task boards or similar tool such as JIRA, which is accessible to all, to track and update task progress state i.e. Not Started, In Progress and Done.
        - Review Lap deliverables and find out the Acceptance rate and Spillover.
        - Have a review meeting on last day of Galop / Lap to find out feedback. Update the user feedback and team feedback in the system. Look for candid voice and act accordingly to bring in involvement and effectiveness in Lap.Try to find out following
            - What should be started?
            - What should be stopped ?
            - What should be continued?
        - Set up various automated reports for daily or weekly basis to understand the various aspects. Take preventive action incase of any deviation.

- **Quality Assurance –** Quality Control and assurance is the part of any project to check the conformance to the requirements. In GenNext framework, this role is not limited to testing and ensuring the processes that defects are unearthed or curtailed. In GenNext framework QA is adaptive and ensures that processes are tested as soon as they are developed and configured. In GenNext framework, process test cases are written first with the category of simple, medium and complex. The processes are built to ensure that they initally meet the simple test cases then medium and complex. It ensures that no gold plating or extra / non value adding work is done.
    - **Tools** – Expert Judegement, Bottleneck Analysis, Standardization , PokaYoke and 5Why.

- **Roles** – Process Owner,GenNext Master, GenNext QA team, Business Userss.
- **Input** –Requirement Register.
- **Output** – Test plans, Test cases, Acceptance Test cases,
- **Activities**
  - GenNext Master to ensure that test cases are written at the time of requirement and before the implementations.
  - GenNext Master to let team do just enough configuration or customization to be done to pass the test.
  - Developers are to ensure that code passes all the test defined and a round of testing is done by developers themselves to ensure minimum hand offs.
  - GenNext Master and Product Owner to ensure test cases are written from business users point of view.
  - Team to execute test cases as soon as process is configured / customized and integrated with testing environment.
  - GenNext Master to ensure that test case execution is automated and should have minimum human intervention.
  - Generate testing report as soon as testing is over. Share testing report with all the stake holders. Share detailed defect report with technical team.
  - GenNext Master to perform periodic defect analysis to find the root cause of defects. Fix the root cause of the defects and see if the defects occur again.Any learning through the analysis should be updated in templates or technique used.
- **Best Practices**
  - Encourage technical team to keep the design simple i.e no over processing / gold plating should be done. Team should write just enough code or set up just enough

configuration that passes the acceptance test – nothing more than or less than acceptance criteria.

- GenNext Master to ensure to use tools and techniques such as templates, automated code review or usage of IDE to ensure mistake proofing i.e. Pokayoke.
- GenNext Master and QA team should do a root cause analysis using 5Why tool to get the real root cause of the issue.
- Technical team should perform a round of testing by themselves before integrating it with Integration testing environment.
- Coding standard and templates to be followed to ensure standardization and mistake proofing.
- QA team should perform a proactive causal analyis or defect prevention analysis to forecast defects based on the current trend and take necessary actions alongwith other stakeholders.
- Test case / plan design should take care of Dependancy structure matrix and perform a regression testing where ever needed.
- Technical team should be informed immediately incase of major defects.

The configuration and customization thus built should be deployed to Production environment as soon as it is developed and new requirements on priortization list should be picked to develop.

### 4.6.3 DELIVERY AND MAINTENANCE

This phase encompasses the activities related to incremental and final delivery. The deployable releases are integrated incrementally with a view of final delivery and product success instead of functional successes of each function. This can be divided in two sub phases -

- **Deployment and Stabilization –** Each release is continuously delivered and integrated with the final product. A new release may cause few issues in the product and needs to be closely monitored and product needs to be supported at the highest priority.
- **Documentation and Training -** Each release has few activities such as documentation and training is required. Documentation and training are required for both the end users and team members who are going to support the application.

## 4.7 COMPARISION OF GENNEXT, LEAN THINKING AND AGILE METHODOLOGY

The three frameworks have been compared on the various parameters

**Satisfied Customer –** GenNext framework gives top importance to customer satisfaction. This is in line with the Lean Thinking and Agile Methodologies.

**Lever to change requirement in later stages of implementation –** GenNext frameworks gives opportunity to user to change in the requirements during the later stages of implementation. Lean Thinking also provides similar flexibility to users. However, Agile Methodologies don't provide such option to users.

**Frequent and constant paced deliveries** – GenNext framework suggests that project should be delivered in multiple releases. The releases should be frequent and should be at a constant pace. Agile Methodologies and Lean Thinking also talk about the frequent deliveries but don't talk about pace of deliveries.

**Elimination of non-value adding activities and wastes –** GenNext framework like Lean Thinking promotes that elimination of low or no output effort consuming activities. Similar principle does not exist in Agile Methodologies.

**Seemless communication and cooperation** – GenNext framework and Agile Methodologies agree on the importance of communication between developers and users for complete lifecycle. However, Lean Thinking does not suggest on communication.

**Self-organizing team of motivated individuals and respect for everyone** – Lean Thinking , Agile Methodolgies and GenNext framework coherently talk about the people. The people should be self motivated and should be respected.

**Measurement by Deliveries –** GenNext and Agile Methodology propose to measure the progress of project by measuring the delivery of working application. Lean Thinking is silent on measuring the progress of project.

**Excellence** – GenNext echoes the Lean Thinking and Agile Methodology on technical excellence and excellence in quality.

**Simplicity –** GenNext framework follows Agile Methodologies to keep everything simple.

**Contious refelection of Amplified learning in the work and delivery –** Agile and Lean share a common understanding on amplifying the learning in each iteration and learning of previous iteratins should be visible in next iterations. GenNext also concurs with Agile and Lean Thinking on this.

**Optimized flow –** GenNext framework infers the idea of optimizing the whole value chainto increase the efficiency and deliver the application fast. Agile Methodologies does not talk about the optimization of flow or value stream.

**Pull Environment –** Neither Agile Methodologis nor Lean Thinking talks about the Pull system. GenNext framework introduces a requirement priortization algorithm that enables it to pull the requirements from backlog.

The above discussions are summarized below in table 4.3.

| PRINCIPLE | LEAN THINKING | AGILE METHODOLOGY | GENNEXT |
|---|---|---|---|
| Satisfied Customer | Y | Y | Y |
| Lever to change requirement in later stages of implementation. | Y | N | Y |
| Frequent and constant paced deliveries. | Y | Y | Y |
| Eliminate non-value adding activities and wastes | Y | N | Y |
| Cohesive cooperation between users and developers during the complete implementation cycle. | N | Y | Y |
| Self-organizing team of motivated individuals and respect for everyone | Y | Y | Y |
| Seamless communication | N | Y | Y |
| Measurement by deliveries made. | N | Y | Y |
| Technical and qualitative excellence. | Y | Y | Y |
| Keep everything simple. | N | Y | Y |
| Contious refelection of Amplified learning in the work and delivery. | Y | Y | Y |
| Optimized flow. | Y | N | Y |
| Create a Pull Environment. | N | N | Y |

**Table 4.4: Difference in Agile Methodologies, Lean Thinking and GenNext framework.**

## 4.8 COMPARISION OF GENNEXT AND TRADITIONAL METHODOLOGY

GenNext framework differs to traditional methodology in the following manner :-

**Customer Feedback Mechanism –** Traditional methodology lacks the customer feedback mechanism [8,136]. However, GenNext framework heavily depends on the feedback provided.

**Developed system does not reflect the current requirement of system –** *Chandra* [27] has pointed that duration between the requirements gathering phase and go live is huge so most of the given requirements get changed. In GenNext framework the implementation is done in small iterations or releases so current requirements are incorporated in the system.

**Requirement Volatility -**Business users tend to change the requirement every now and then [31]. *Cohen et al.,* [31] also mentioned that this change could be because of knowledge limitation or the limited visibility they have. The GenNext framework believes in looking at the system as whole so that anomalies can be put on surface and can be eliminated. Also, GenNext involves the various layers of stakeholders who can reduce the variability in requirement. This helps in removing the requirement volatility,

**Later phases squeezed to absorb delays of Initial -** In traditional methodology where everything is sequential, the delay in one activity must be absorbed by the later activities. The testing activities, usually done in last leg, must absorb the delays of all preceding activities hence affecting quality [129]. However, GenNext framework proposes to deliver in the releases so delay in one activity does not impact the complete implementation and quality.

**High effort or Cost of Change. –** The GenNext framework believes in keeping everything simple and the changes are handled, responded well in time due to its structure of releases.

**Communication –** GenNext framework believes in seamless communication between developers and business users at all levels while it is found to be missing in traditional methodology.

**Reflection of learning –** GenNext framework enables the assimilation and reflection of learning from previous releases into next releases which is not possible in traditional methodology.

**Wastage of time and resource –** GenNext believes in reducing the non-value adding activities to minimum. It relies on minimum documentation and working application. The approvals are received within the verification and validation meeting which reduces time in waiting or approvals.

**Acceptance oriented** – GenNext believes in configuration and customization the system to a minimum extent which satisfies user requirements. It does not believe in over -processing or gold plating as, usually, done in traditional methodology.

The effectiveness of the GenNext framework w.r.t. traditional methodologies is discussed in next chapter – Chapter 5. The effectiveness has been measured on the parameters defined in section 1.3 and have been statistically verified the hypothesis defined in section 1.4.

# CHAPTER 5: RESULTS AND DISCUSSIONS

## 5.1 INTRODUCTION

A study was conducted to verify the efficacy of the GenNext framework. The GenNext was applied to five projects. These five projects given the unique IDs as GP1, GP2, GP3, GP4, GP5 for uniquely identifying each project. These projects are referred as GenNext projects in the thesis. The results obtained (total effort, effort in testing, total number of defects, COQ and DIR) were compared with 5 projects (projects given the unique IDs as TP1, TP2, TP3, TP4, TP5 for uniquely identifying each project) of similar complexity. These projects are referred as traditional projects in the thesis.

## 5.2 ANALYSIS OF EFFORT SPENT IN VARIOUS PHASES OF IMPLEMENTATION

### 5.2.1. COMPARISON OF THE EFFORTS IN REQUIREMENT SPECIFICATION

*Anderson* [8] and *Boehm* [19] suggest that initial requirement gathering phase consumes more effort than the scheduled effort. The schedule deviation in the requirement specification phase for the projects executed using traditional methodologies is given below in the table (Table 5.1). Effort for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent (PHrs) | Planned Effort (PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 1052 | 792 | 32.83 |
| TP2 | 1052 | 792 | 32.83 |
| TP3 | 1026 | 792 | 29.55 |
| TP4 | 1072 | 792 | 35.35 |
| TP5 | 1054 | 792 | 33.08 |

**Table 5.1: Effort consumed by traditional projects in requirement specification.**

GenNext framework consists of Laps and Gallops. Effort spent in various Laps is combined and enumerated below in the table (Table 5.2).

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 904 | 792 | 14.14 |
| GP2 | 894 | 792 | 12.88 |
| GP3 | 910 | 792 | 14.9 |
| GP4 | 916 | 792 | 15.66 |
| GP5 | 916 | 792 | 15.66 |

**Table 5.2: Effort consumed by GenNext projects in requirement specification.**

Looking at the Table 5.1, it is found that the average effort deviation in case of projects executed using the traditional methodology is 33% approximately ranging from 29.5% to 35.4%. In case of projects executed using GenNext methodology, the average effort deviation was found to be 14.6% with a spread of 12.8% to 15.6%. Application of GenNext framework on similar projects helped to save effort by approximately 140 PHrs on an average in the project. In other words, projects executed with GenNext framework completed the requirement specification activities in approximately 20% less time. This is evident in figure 5.2.

The comparison of the effort spent in the requirement specification activities for the projects executed using GenNext and Traditional methodologies with planned effort is shown in the figure 5.1. The lesser effort deviation in the requirement specification activities ensures that subsequent phases / activities would get the required time and effort. This also suggests that GenNext takes care of issue I6 which has a criticality C1.

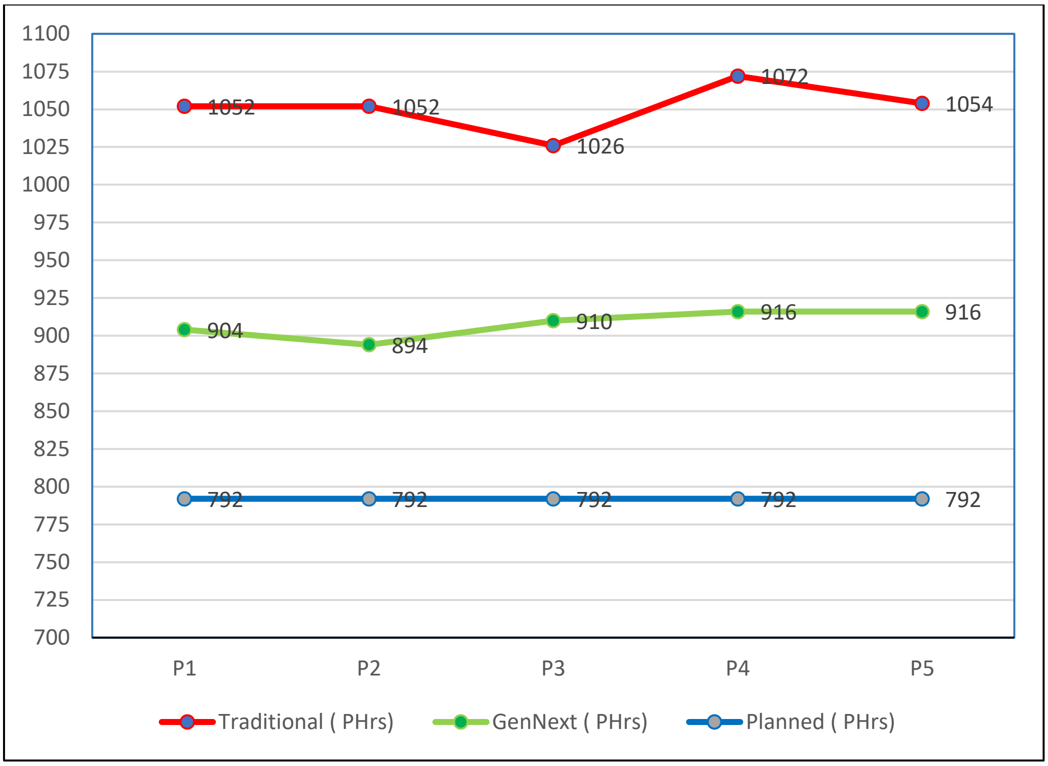


**Figure 5.1: Comparison of the effort consumed in requirement specification GenNext Projects, traditional projects with planned effort.**

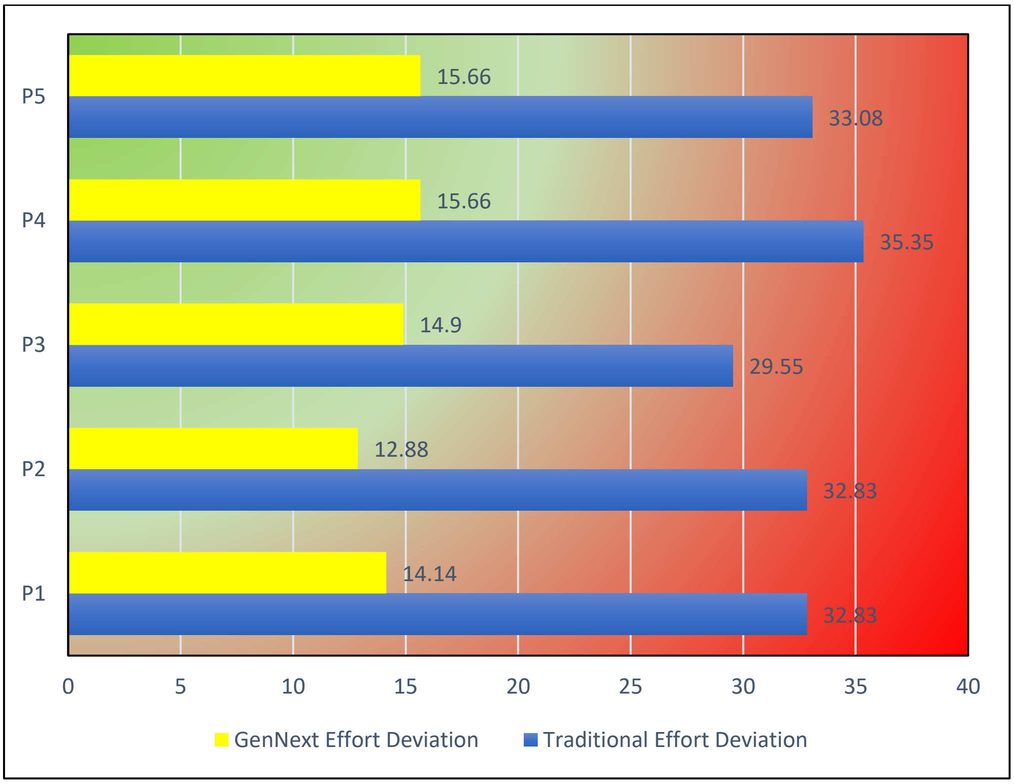


**Figure 5.2: Comparison of the effort deviation from planned effort in requirement specification for GenNext projects and Traditional projects.**

The results were evaluated statistically using the hypothesis HYP1A (defined in section 1.4). The hypothesis is given below -

*$H_{1A0}$: Requirement Specification activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*

*$H_{1Aa}$: Requirement Specification activities in GenNext Projects consume* ***less effort*** *than traditional projects.*

Statistical analysis of the results is given below in Table 5.3. It is found that critical value is 2.306 which is less than the t value 16.9899 and the p value is less than 0.00001 which is less than the α value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{1A0}$ should be rejected. The risk to reject $H_{1A0}$ while it is true is less than 0.01%. This signifies that *ERP Implementation Projects executed with GenNext framework consume lesser effort in Requirement Specification activities than that of traditional methodology. than those executed with traditional methodology.*

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 1051.2 | 908 |
| Variance | 269.2 | 86 |
| Stand. Dev. | 16.4073 | 9.2736 |
| n | 5 | 5 |
| t (observed value) | 16.9899 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | < 0.00001 | |
| α | .05 | |

**Table 5.3: Statistical analysis of the effort consumed in the requirement specification activity for traditional projects and GenNext projects.**

### 5.2.2 COMPARISON OF THE EFFORTS IN DESIGN ACTIVITIES

In case of traditional methodologies, design or blueprinting is done before the configuration or coding for customization. In case of GenNext framework, the development is done in an iterative format and hence the blueprinting of configurations and design of customization is done iteratively. The design issues in the traditional methodology are often found during the configuration / construction phase or testing phase and are found to be effort and cost intensive [8,19]. In case of incremental design and development, the need of design refactoring is also identified [27]. The effort spent in the design / blueprint activities in the projects executed using traditional methodology is given below in Table 5.4. Effort for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 1028 | 792 | 29.8 |
| TP2 | 1040 | 792 | 31.31 |
| TP3 | 1038 | 792 | 31.06 |
| TP4 | 1052 | 792 | 32.83 |
| TP5 | 1044 | 792 | 31.82 |

**Table 5.4: Effort consumed in blueprint activities for traditional projects**

The GenNext framework emphasizes on having an initial blueprint and then layered blueprinting with refactoring wherever and whenever required. The effort spent in the design activities for projects executed using the GenNext framework is given below in Table 5.5.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 934 | 792 | 17.93 |
| GP2 | 930 | 792 | 17.42 |
| GP3 | 956 | 792 | 20.71 |
| GP4 | 916 | 792 | 15.66 |
| GP5 | 934 | 792 | 17.93 |

**Table 5.5: Effort consumed in blueprint activities for GenNext projects.**

The figures 5.3 and 5.4 suggest that the projects executed using the traditional methodology showed an average effort deviation of 32% approximately, whereas the projects executed using the GenNext Framework showed an average effort deviation of 18% approximately. The spike in the effort deviation in case of projects executed using GenNext framework is because of the design refactoring activities, which were not planned initially.

To evaluate further, the results were evaluated statistically using the hypothesis HYP1B (defined in section 1.4). The hypotheses are given below -

> $H_{1B0}$*: Design activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*
>
> $H_{1Ba}$*: Design activities in GenNext Projects consume* ***less effort*** *than traditional projects*

Statistical analysis of the results is given below in Table 5.6. It is found that critical value is 2.306 which is less than the t value 14.14774 and the p value is < 0.00001 which is less than the α value of 0.05. The t-value analysis and p-value suggests that the results are significant and Null hypothesis $H_{1B0}$ should be rejected. The risk to reject $H_{1B0}$ while it is true is less than 0.01%. This signifies that *ERP Implementation Projects executed with GenNext framework consume lesser effort in in Design / blueprint activities than those executed with traditional methodology.*

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 1040.4 | 934 |
| Variance | 76.8 | 206 |
| Stand. Dev. | 8.7636 | 14.3527 |
| n | 5 | 5 |
| t (observed value) | 14.1477 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | <0.00001 | |
| α | 0.05 | |

**Table 5.6: Statistical analysis of the effort consumed in the design activity for traditional projects and GenNext projects.**

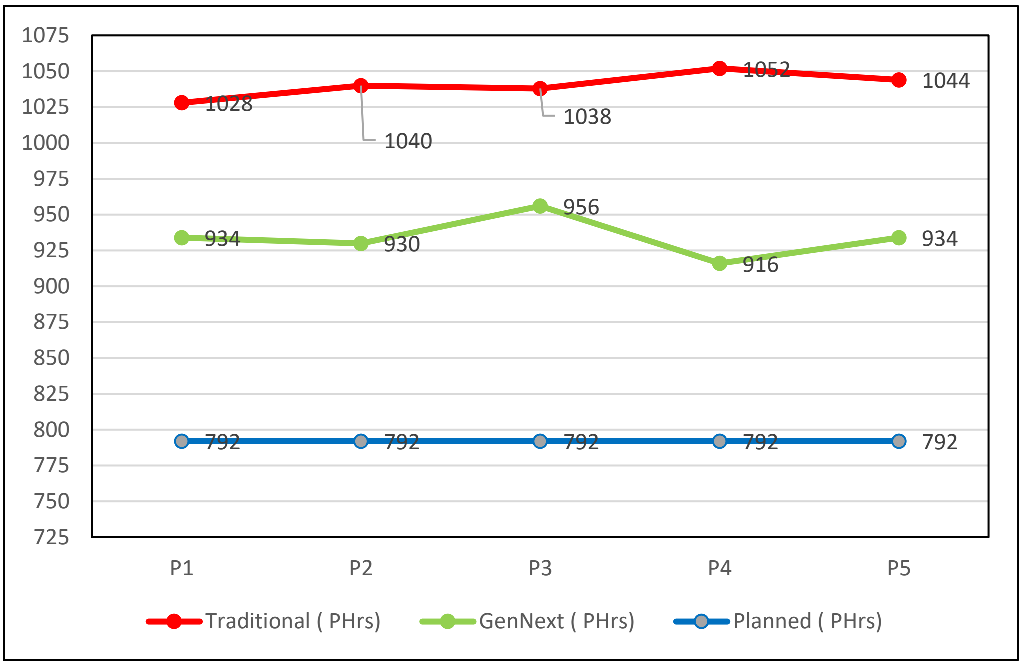


**Figure 5.3: Comparison of the effort consumed in design of GenNext projects, traditional projects with planned effort.**

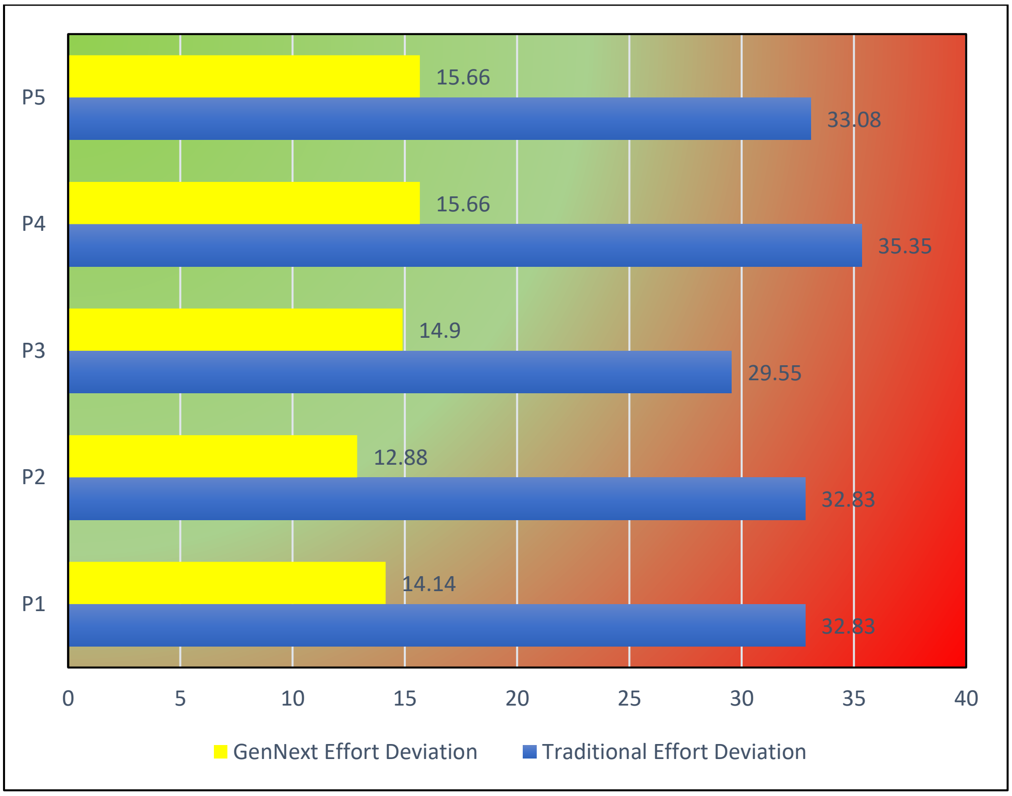


**Figure 5.4: Comparison of the effort deviation from planned effort in blueprint / design for GenNext projects and Traditional projects.**

### 5.2.3 COMPARISON OF THE EFFORTS IN CONSTRUCTION AND CONFIGURATION ACTIVITIES

In case of traditional methodologies, design is followed by development, which includes the configuration, and customization development. In case of GenNext framework, all the activities are iterative, and scope based. Any new scope identified is taken in the new Lap. The newly configured / built system or customizations are continuously integrated with the system in iterative format and hence the blueprinting of configurations and design of customization is done iteratively. The effort in construction and customization also includes the effort spent in code refactoring, though no special refactoring cycle was planned. In case of GenNext framework, functional consultant provided the requirement in form of test cases to developers and developers were asked to write the code only to pass the test. This ensured that no requirement is missed and there is no unnecessary gold plating and unnecessary validation or code routes. Table 5.7 and Table 5.8 present the effort spent in the customization and configuration of the requirements for projects executed using the traditional methodology and GenNext Framework respectively. Effort for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 2380 | 1540 | 54.55 |
| TP2 | 2369 | 1540 | 53.83 |
| TP3 | 2492 | 1540 | 61.82 |
| TP4 | 2464 | 1540 | 60 |
| TP5 | 2404 | 1540 | 56.1 |

**Table 5.7: Effort consumed in construction and configuration for Traditional projects.**

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 1916 | 1540 | 24.42 |
| GP2 | 1928 | 1540 | 25.19 |
| GP3 | 1978 | 1540 | 28.44 |
| GP4 | 1888 | 1540 | 22.6 |
| GP5 | 1978 | 1540 | 28.44 |

**Table 5.8: Effort consumed in construction and configuration for GenNext projects.**

The figures below (Figure 5.5 and Figure 5.6) and the tables (Table 5.7 and Table 5.8) point that effort deviation was seen in both the methodologies. Projects executed using the GenNext framework overshoot the effort by approximately 25%; on the other hand, projects executed using the traditional methodology overshoot the effort approximately by 58% with a range of 55 to 62 approximately. GenNext framework could curtail the effort deviation by 57% from the traditional methodology.

The results were evaluated statistically using the hypothesis HYP1C (defined in section 1.4). The hypotheses are given below -

> *$H_{1C0}$: Construction and Configuration activities in GenNext Projects consume **more or equal effort** than traditional projects.*
>
> *$H_{1Ca}$: Construction and Configuration activities in GenNext Projects consume **less effort** than traditional projects.*

Statistical analysis of the results is given below in Table 5.9. It is found that critical value is 2.306 which is less than the t value 16.2104 and the p value is $< 0.00001$ which is less than the $\alpha$ value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{1C0}$ should be rejected. The risk to reject $H_{1C0}$ while it is true is less than 0.01%.

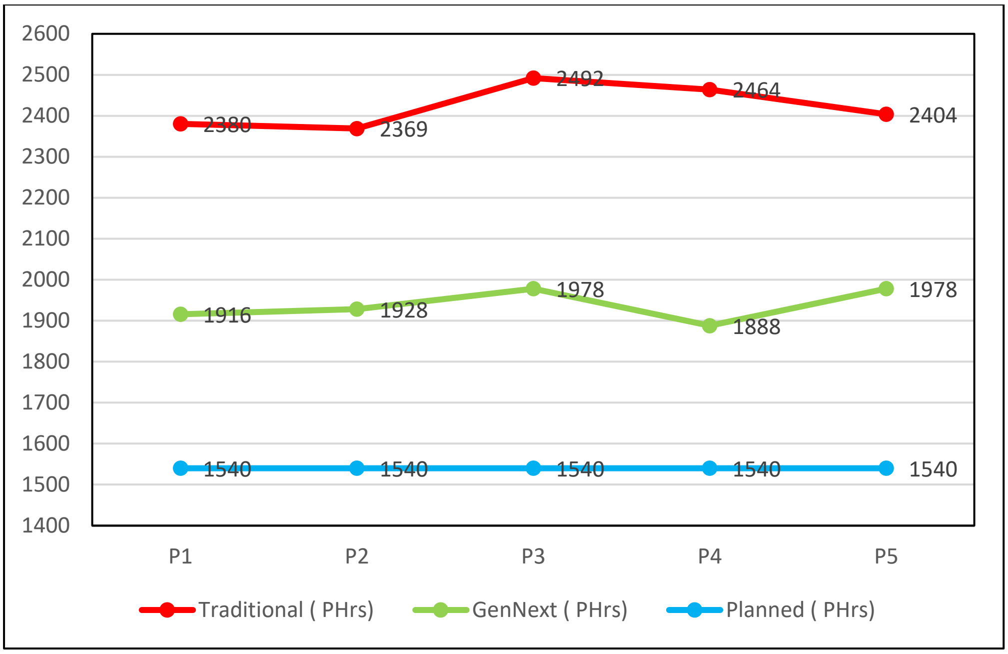


**Figure 5.5: Comparison of the effort consumed in construction and configuration of GenNext projects, Traditional projects with planned effort.**

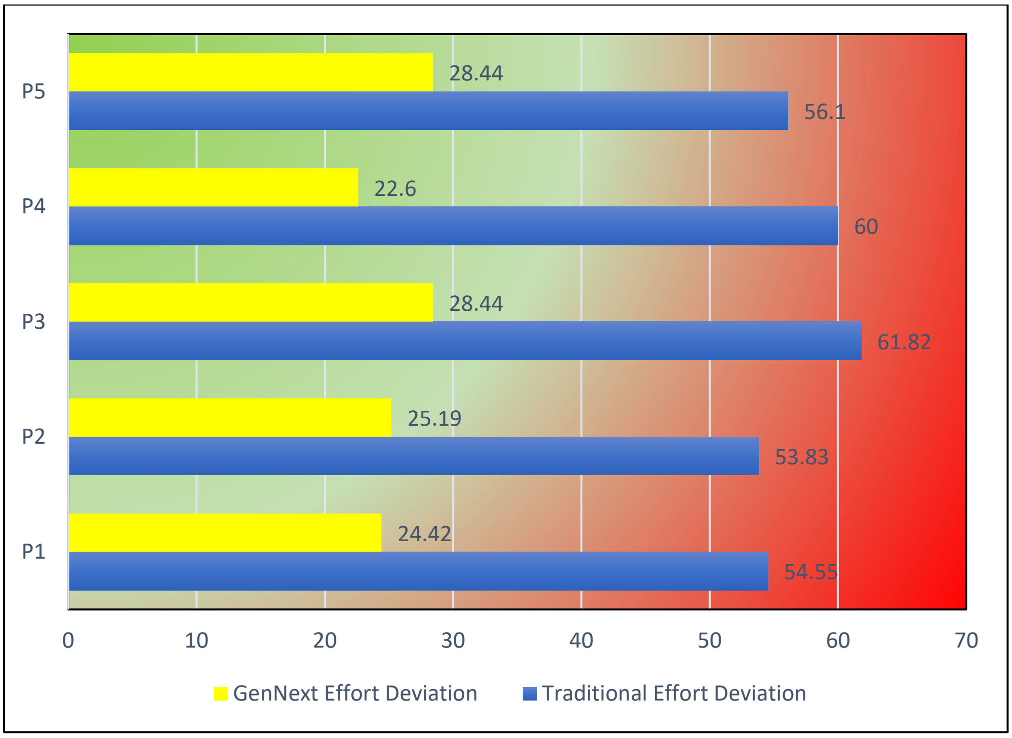


**Figure 5.6: Comparison of the effort deviation from planned effort in construction and configuration for GenNext projects and Traditional projects.**

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 2421.8 | 1937.6 |
| Variance | 2890.2 | 1570.8 |
| Stand. Dev. | 53.7606 | 39.6333 |
| N | 5 | 5 |
| t (observed value) | 16.2104 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | <0.00001 | |
| α | 0.05 | |

**Table 5.9: Statistical analysis of the effort consumed in the construction and configuration activity for Traditional projects and GenNext projects.**

## 5.3 ANALYSIS OF EFFORT SPENT IN VERIFICATION AND VALIDATION PHASES OF IMPLEMENTATION

The verification and validation activities are performed to ensure that deliverables conform to the requirements and satisfy the defined quality standards. Effort spent in verification and validation activity not only includes the effort spent in inspection, but it also includes the effort spent in fixing the observation shared.

Validation and verification activities appear to be waste activities. Nevertheless, it helps to ensure that deliverables are correct and, thus, its importance cannot be neglected. It can be categorized into Type 1 Muda in terms of Lean. GenNext framework attempts to reduce the effort in this activity rather than eliminating it.

### 5.3.1 COMPARISON OF THE EFFORTS IN VERIFICATION AND VALIDATION OF REQUIREMENT SPECIFICATION

The verification and validation of the requirement specifications is usually done after the requirement gathering or project preparation phase and business blueprint phase. The project preparation and business blueprint phase help to identify and clarify the scope of implementation, various processes and their mapping into ERP system. The traditional methodology suggests having verification and validation activities to check the understanding and do it once the requirements are gathered. However, in case of GenNext framework, it is recommended to involve various stakeholders and have reviews of the requirements or integrations as soon as any integration or requirement is found. Table 5.10 illustrates the effort spent in validation and verification of requirement specification for projects executed using Traditional Methodology. Efforts for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 260 | 132 | 96.97 |
| TP2 | 256 | 132 | 93.94 |
| TP3 | 272 | 132 | 106.06 |
| TP4 | 288 | 132 | 118.18 |
| TP5 | 288 | 132 | 118.18 |

**Table 5.10: Effort consumed in verification and validation of requirements for Traditional projects**

Table 5.11 illustrates the effort spent in validation and verification of the requirement specification for projects executed using GenNext framework.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 176 | 132 | 33.33 |
| GP2 | 172 | 132 | 30.3 |
| GP3 | 172 | 132 | 30.3 |
| GP4 | 184 | 132 | 39.39 |
| GP5 | 184 | 132 | 39.39 |

**Table 5.11: Effort consumed in verification and validation of requirements for GenNext projects**

Analysis of the effort spent in validation and verification of requirements suggests that projects executed using the traditional methodology over utilized the effort by more than 100%. This suggests that requirements were either not captured or were changed during the initial phase. This is in line with the issues identified in interviews of industry professionals. On the other hand, projects executed using GenNext framework involved customer, stakeholders at all discussions and reviews. This ensured that requirements are non-ambiguous, unique and feasible. Further analysis of the reason of the effort deviation was due to delay in feasibility analysis where in technical and functional consultants were not sure about the do-ability of requirements or customization. Projects executed using GenNext framework showed average effort deviation of 34% approximately.

Figure 5.7 and figure 5.8 show the difference in effort deviations with respect to planned effort for projects executed using traditional as well as GenNext framework.

To evaluate further, the results were evaluated statistically using the hypothesis HYP1D (defined in section 1.4). The hypotheses are given below -

> $H_{1D0}$*: Validation and Verification of RS activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*
>
> $H_{1Da}$*: Validation and Verification RS activities in GenNext Projects consume* ***less*** *effort than traditional projects*

Statistical analysis of the results is given below in Table 5.12. It is found that critical value is 2.306 which is less than the t value 13.1015 and the p value is less than 0.00001 which is less than the α value of 0.05. The t-value analysis and p-value suggests that the results are significant and Null hypothesis $H_{1D0}$ should be rejected. The risk to reject $H_{1D0}$ while it is true is less than 0.01%.

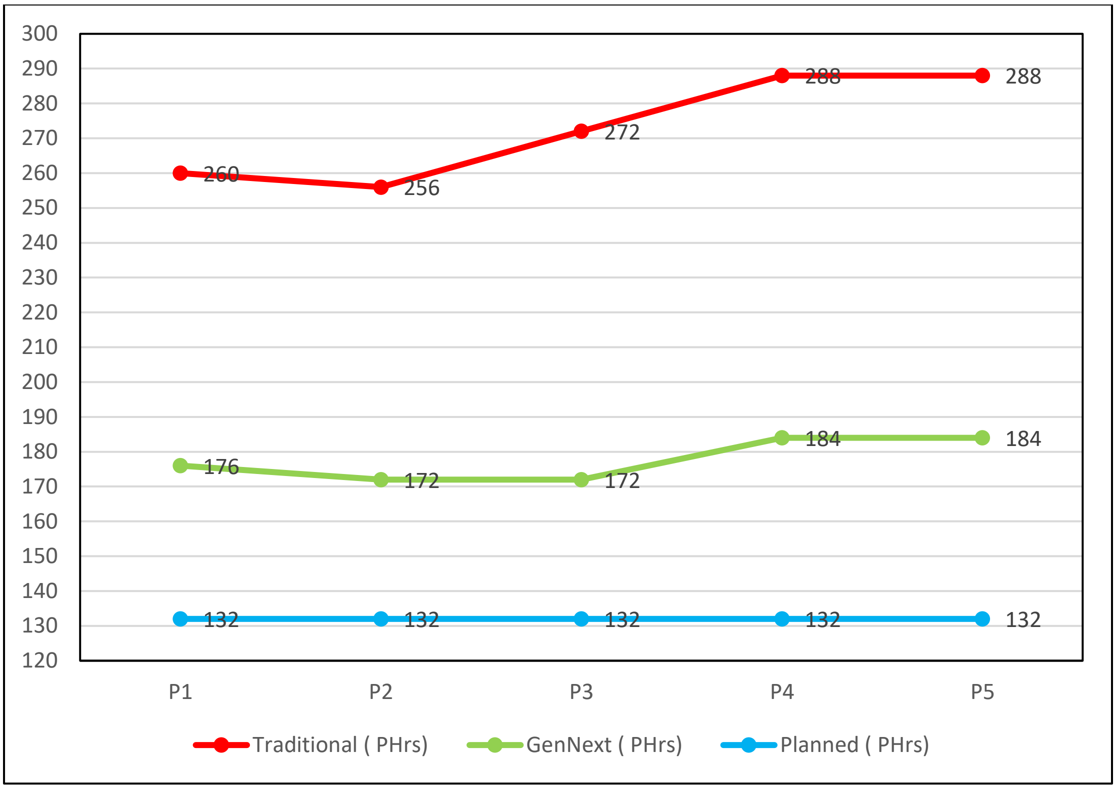


**Figure 5.7: Comparison of the effort consumed in verification and validation of GenNext projects, Traditional projects with planned effort.**

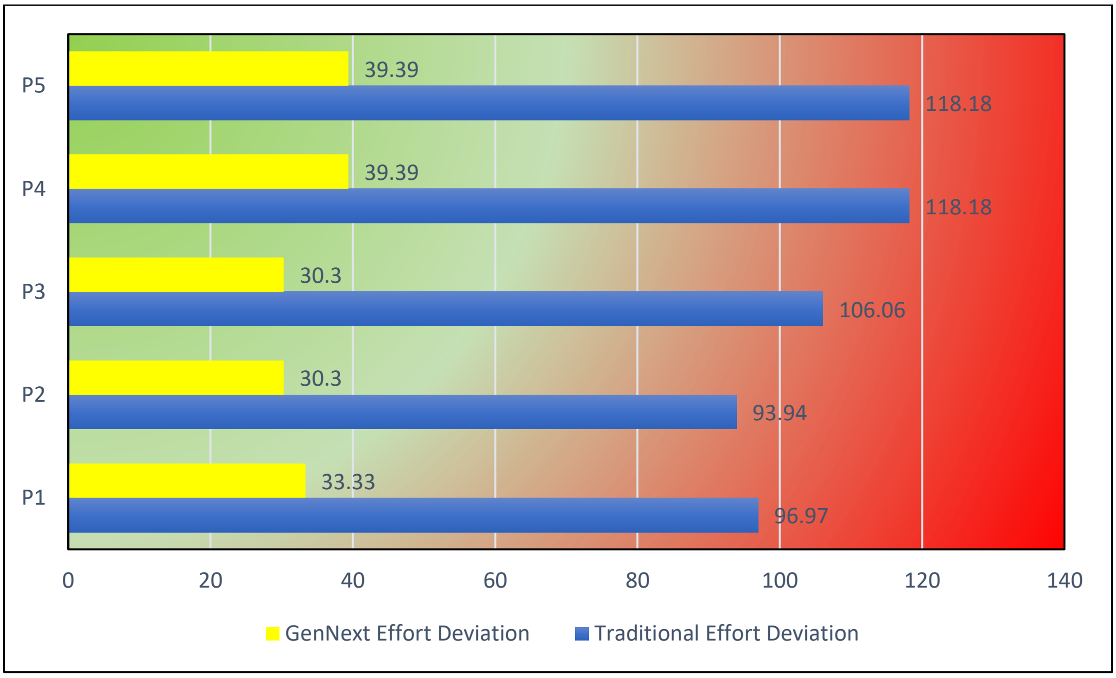


**Figure 5.8:Comparison of the effort deviation from planned effort in verification and validation of requirements for GenNext projects and Traditional projects.**

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 272.8 | 177.6 |
| Variance | 227.2 | 36.8 |
| Stand. Dev. | 15.0732 | 6.0663 |
| n | 5 | 5 |
| t (observed value) | 13.1015 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | <0.00001 | |
| α | 0.05 | |

**Table 5.12: Statistical analysis of the effort consumed in the verification and validation of requirements for Traditional projects and GenNext projects.**

### 5.3.2 COMPARISON OF THE EFFORTS IN VERIFICATION AND VALIDATION OF DESIGN

The design of customization is a part of the business blueprinting. However, the stage produces its own set of deliverables called as functional design documents and technical design documents (Often called as MD 50 and MD 70 respectively in Oracle AIM methodology). Review of design serves as a mechanism to catch the issues before they occur and multiply the impact. In case of traditional methodologies, verification of 'To Be' process and design of customization is done at series of individual levels and multiple group levels. However, in case of GenNext framework, team does not consist of freshers and hence only group reviews (including architects of various tracks, stakeholders and process owners) were carried out. Group reviews helped in optimizing the whole value stream and finding probability and impact of the issue on complete value stream. Group reviews helped drastically to crash the schedule, but the efforts were shot up. The effort rise owing to group reviews in the GenNext framework (Table 5.14) did not cause as much effort deviation as the verification and validation of design in projects (Table 5.13) using traditional methodology. Efforts for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort (PHrs) | Effort Deviation (%) |
|---|---|---|---|
| TP1 | 272 | 132 | 106 |
| TP2 | 256 | 132 | 94 |
| TP3 | 288 | 132 | 118 |
| TP4 | 256 | 132 | 94 |
| TP5 | 256 | 132 | 94 |

**Table 5.13: Effort consumed in verification and validation of design for Traditional projects.**

| Project ID | Effort Spent (PHrs) | Planned Effort (PHrs) | Effort Deviation (%) |
|---|---|---|---|
| GP1 | 152 | 132 | 15 |
| GP2 | 156 | 132 | 18 |
| GP3 | 148 | 132 | 12 |
| GP4 | 165 | 132 | 25 |
| GP5 | 158 | 132 | 20 |

**Table 5.14: Effort consumed in verification and validation of design for GenNext projects.**

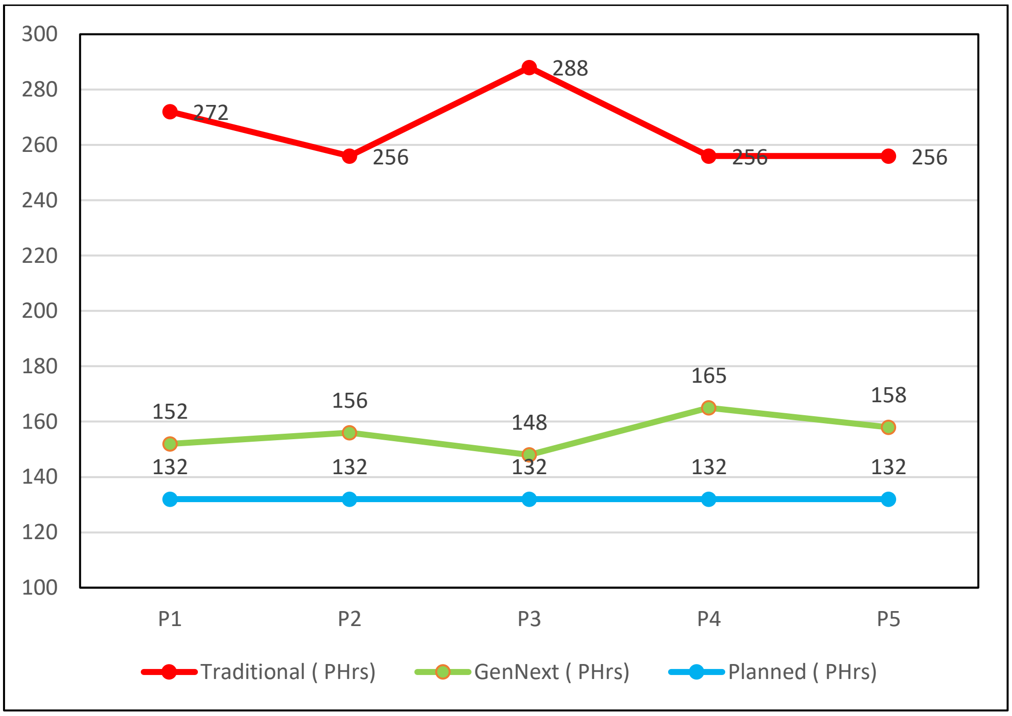


**Figure 5.9: Comparison of the effort consumed in verification and validation of design of GenNext projects, Traditional projects with planned effort.**

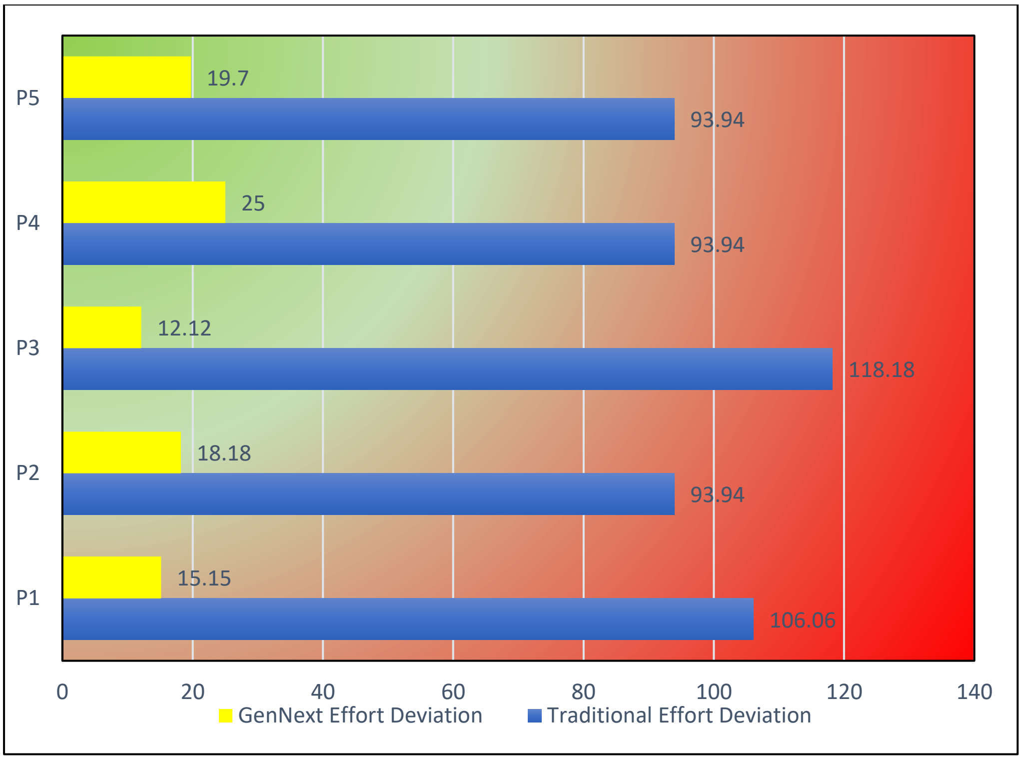


**Figure 5.10: Comparison of the effort deviation from planned effort in blueprint / design for GenNext projects and Traditional projects.**

Analysis of the Figure 5.9 and Figure 5.10 suggests that the projects executed using the traditional methodology noted the average effort deviation of more than 100% while the projects executed using the GenNext framework noted an average effort deviation of 18% with a range of 12% to 25%. GenNext framework is seen to be effectively optimizing the effort spent. The reasons for such low deviation in case of GenNext framework are already described.

To evaluate further, the results were evaluated statistically using the hypothesis HYP1E (defined in section 1.4). The hypotheses are given below -

> $H_{1E0}$: *Validation and Verification of design activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*
>
> $H_{1Ea}$: *Validation and Verification of design activities in GenNext Projects consume* ***less effort*** *than traditional projects.*

Statistical analysis of the results is given below in Table 5.15. It is found that critical value is 2.306 which is less than the t value 15.6539 and the p value is < 0.00001 which is less than the α value of 0.05. The t-value analysis and p-value suggests that the results are significant and Null hypothesis $H_{1E0}$ should be rejected. The risk to reject $H_{1E0}$ while it is true is less than 0.01%.

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 265.6 | 155.8 |
| Variance | 204.8 | 41.2 |
| Stand. Dev. | 14.3108 | 6.4187 |
| N | 5 | 5 |
| t (observed value) | 15.6539 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | <0.00001 | |
| α | 0.05 | |

**Table 5.15: Statistical analysis of the effort consumed in Validation and Verification of design activity for Traditional projects and GenNext projects.**

### 5.3.3 COMPARISON OF THE EFFORTS IN VERIFICATION AND VALIDATION OF CONFIGURATION AND CONSTRUCTION ACTIVITIES

Customizations and configurations, often, are reviewed to find and eliminate the trivial issues such as infinite loops, memory leaks to ensure that all the requirements and features have been taken care of and a maintainable system is produced. In case of traditional methodology, the verification and validations of configurations are done using various peer reviews and final approval from program manager. Program Manager may be from different functional area and hence does not add any value to the process. In addition, for customizations, code must undergo a series of reviews such as peer review, module lead review and architect review. These activities add a little value to the implementation, are effort intensive and dampen the velocity of implementations. In case of

GenNext framework, team was empowered and was encouraged to go through the checklist while releasing the code for review. The checklist of the developer kept on growing based on the feedback received from reviews. The code was reviewed in-group reviews jointly by Technical Lead, Architect and functional consultant to ensure that all technical aspects are met. As stated earlier, Functional consultants provided the requirements in case of test cases so that developer could write code only to pass the test.

The result of the proactive steps taken in GenNext framework are seen to be yielding result if we compare the tables 5.16 and 5.17. It is seen in Figure 5.11 and Figure 5.12 that the projects executed with the traditional methodology consumed 46% more effort than the projects executed using the GenNext framework and the projects executed using traditional methodology showed an average effort deviation of 70%. Projects executed using the GenNext framework also showed an average effort deviation of 16%. Further analysis of the effort deviation of projects executed using the GenNext framework is approximately 15% in three projects whereas effort deviation of the projects executed using traditional methodology was more than 50% for all the projects. GenNext framework is seen to be effectively optimizing the effort spent. The effort for various activities for both traditional and GenNext projects is given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 532 | 340 | 56 |
| TP2 | 584 | 340 | 72 |
| TP3 | 532 | 340 | 56 |
| TP4 | 608 | 340 | 79 |
| TP5 | 648 | 340 | 91 |

**Table 5.16: Effort consumed in validation and verification of construction and configuration for Traditional projects**

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 396 | 340 | 16 |
| GP2 | 384 | 340 | 13 |
| GP3 | 402 | 340 | 18 |
| GP4 | 384 | 340 | 13 |
| GP5 | 418 | 340 | 23 |

**Table 5.17: Effort consumed in validation and verification of construction and configuration for GenNext projects.**

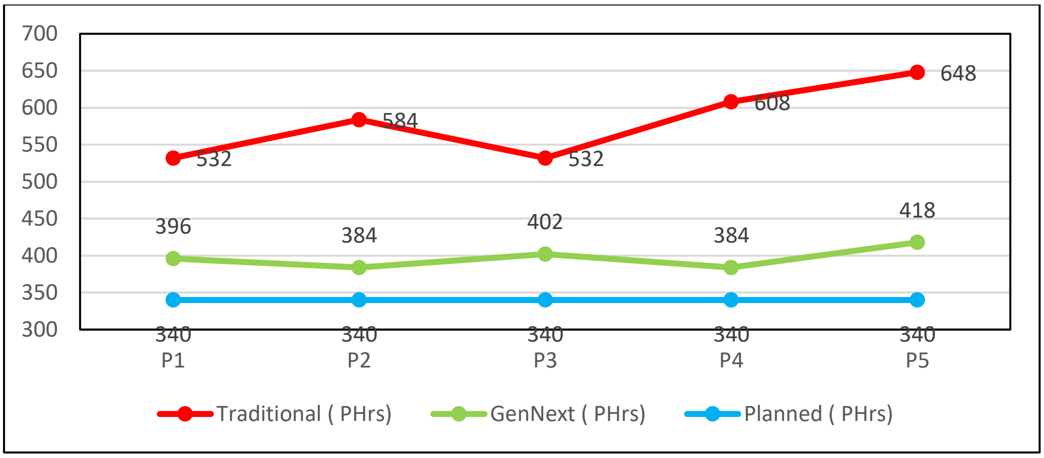


**Figure 5.11: Comparison of the effort consumed in validation and verification of construction and configuration of GenNext projects, Traditional projects with planned effort.**

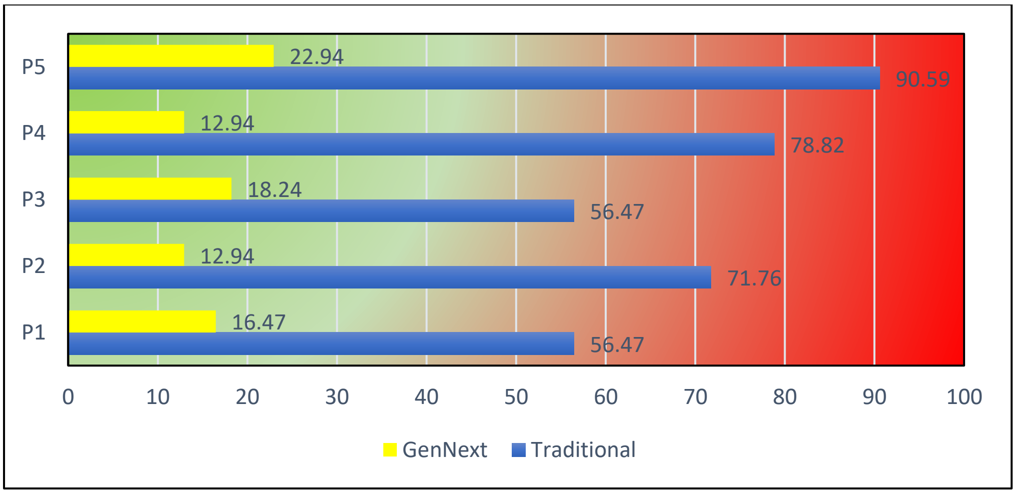


**Figure 5.12: Comparison of the effort deviation from planned effort in validation and verification of construction and configuration for GenNext projects and Traditional projects.**

To evaluate further, the results were evaluated statistically using the hypothesis HYP1F (defined in section 1.4). The hypotheses are given below -

*$H_{1F0}$: Validation and Verification of development activities in GenNext Projects consume **more or equal effort** than traditional projects.*

*$H_{1Fa}$: Validation and Verification of development activities in GenNext Projects consume **less effort** than traditional projects.*

Statistical analysis of the results is given below in Table 5.18. It is found that critical value is 2.306 which is less than the t value 7.9058 and the p value is 0.000048 which is less than the α value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{1F0}$ should be rejected. The risk to reject $H_{1F0}$ while it is true is less than 0.01%.

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Variance | 2507.2 | 201.2 |
| Stand. Dev. | 50.0719 | 14.1845 |
| n | 5 | 5 |
| t (observed value) | 7.9058 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | 0.000048 | |
| α | 0.05 | |

**Table 5.18: Statistical analysis of the effort consumed in the validation and verification of construction and configuration activity for Traditional projects and GenNext projects.**

## 5.4 ANALYSIS OF THE EFFORT SPENT IN VARIOUS TYPES OF TESTING

Testing of the ERP system provides information about quality of ERP system implemented, conformance to the requirements and performance of the stakeholders and enables them to take the decisions. Usually, the objectives for the testing are:

- Validate system functionality and conformance to the requirements.
- Verification of configurations in the various flows.

In case of traditional methodology, both configuration and customization, undergoes following types of testing –

1. **Self-Testing** – Done by Technical / functional component himself before releasing it for testing.
2. **Unit Testing** – Done by Technical lead or Functional Lead of the track. In case of customizations both Technical and Functional Lead do a round of testing before the component is integrated with the system.
3. **System Integrated Testing (SIT)** – Done after customizations and configurations are integrated with system by Functional team. The idea is to ensure that configurations and customizations behave in the right way and the system does not break.
4. **User Acceptance Testing (UAT)** – Performed by a set of business users who have fair idea about the ERP system. Hand off from IT to business is started post UAT approval. UAT is performed, post smoke testing, to test the basic functionalities of the process.

In case of GenNext framework, functional team provides requirements in case of test cases and technical team or configuration team builds up only the system that passes the test case. This methodology helps in eliminating the need of unit testing and helps further in reducing the effort and time in performing self-testing, integration testing and UAT.

### 5.4.1 COMPARISON OF THE EFFORTS IN INTEGRATION TESTING

Integration testing enables us to test the system end to end. Modules to be tested usually take the input from the data or from user, manipulates and performs intended operations and provides the processed data to another module or to user. The modules to be tested are approved and unit tested. Integration testing may follow top-down or bottom up approach, but the idea remains the same i.e. to weed out non-conformance and ensuring end to end processing.

Table 5.19 shows the effort spent in SIT in the projects executed using Traditional methodology. It is clearly seen that the projects executed using traditional methodology overshoot the effort by approximately 24% average with minimum deviation in the range of 19% to 29%. Efforts for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 472 | 396 | 19.19 |
| TP2 | 486 | 396 | 22.73 |
| TP3 | 496 | 396 | 25.25 |
| TP4 | 512 | 396 | 29.29 |
| TP5 | 492 | 396 | 24.24 |

**Table 5.19: Effort consumed in SIT for Traditional projects**

Table 5.20 shows the effort spent in SIT in the projects executed using GenNext framework. It is clearly seen that the projects executed using traditional methodology overshoot the effort by approximately 3% in average with maximum deviation as 7% approximately and in one of the projects, effort spent in SIT was lesser by 1% than planned effort.

Figure 5.13 shows the comparison of the efforts spent by various projects executed using traditional methodology and GenNext framework. Figure 5.14 shows the comparison of the deviation in the efforts spent by various projects executed using traditional methodology and GenNext framework. Projects executed using GenNext methodology are seen to be completed within or minimum deviation from planned effort.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 408 | 396 | 3.03 |
| GP2 | 400 | 396 | 1.01 |
| GP3 | 416 | 396 | 5.05 |
| GP4 | 392 | 396 | -1.01 |
| GP5 | 424 | 396 | 7.07 |

**Table 5.20: Effort consumed in SIT for GenNext projects.**

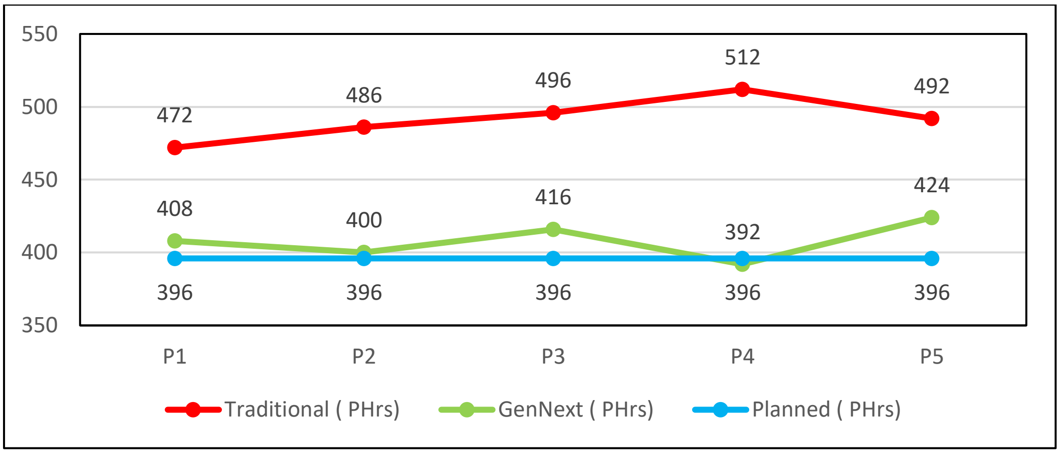


**Figure 5.13: Comparison of the effort consumed in SIT of GenNext projects, Traditional projects with planned effort.**

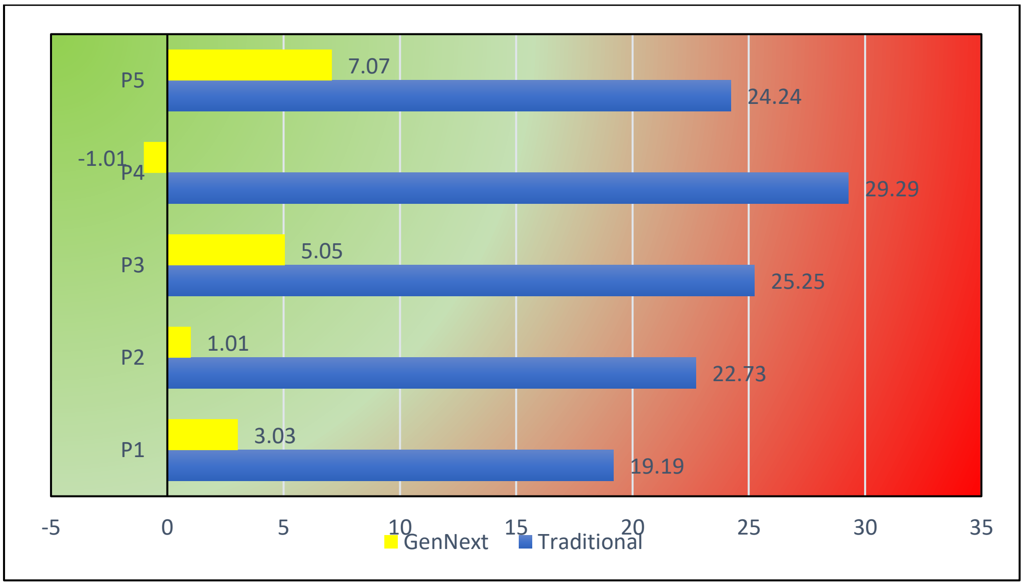


**Figure 5.14: Comparison of the effort deviation from planned effort in SIT for GenNext projects and Traditional projects.**

To evaluate further, the results were evaluated statistically using the hypothesis HYP1G (defined in section 1.4). The hypotheses are given below -

> *$H_{1G0}$: SIT activities in GenNext Projects consume **more or equal effort** than traditional projects.*
>
> *$H_{1Ga}$: SIT activities in GenNext Projects consume **less effort** than traditional projects.*

Statistical analysis of the results is given below in Table 5.21. It is found that critical value is 2.306 which is less than the t value 9.6818 and the p value is 0.000011 which is less than the α value of 0.05. The t-value analysis and p-value suggests that the results are significant and the Null hypothesis $H_{1G0}$ should be rejected. The risk to reject $H_{1G0}$ while it is true is less than 0.01%. Hence, *ERP Implementation Project executed with GenNext framework consume lesser effort in* SIT *that of executed with traditional methodology*.

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 491.6 | 408 |
| Variance | 212.8 | 160 |
| Stand. Dev. | 14.5877 | 12.6491 |
| N | 5 | 5 |
| t (observed value) | 9.6818 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | 0.000011 | |
| α | 0.05 | |

**Table 5.21: Statistical analysis of the effort consumed in the SIT for Traditional projects and GenNext projects.**

### 5.4.2 COMPARISON OF THE EFFORTS IN USER ACCEPTANCE TESTING

Acceptance testing is usually done by a group of business users or their representatives, to ensure conformance to user’s functional and nonfunctional requirements. Acceptance testing is important because it directly affects decision and nonconformance may even lead to scrapping of the project. Acceptance testing ranges from smoke test to complete testing of the system. It is final step before ERP, or any system goes for maintenance. Table 5.22 shows the effort spent in UAT in the projects executed using Traditional methodology. It is clearly seen that projects executed using traditional methodology overshoot the effort by approximately 40% in average. Efforts for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 198 | 132 | 50 |
| TP2 | 164 | 132 | 24.24 |
| TP3 | 184 | 132 | 39.39 |
| TP4 | 184 | 132 | 39.39 |
| TP5 | 192 | 132 | 45.45 |

**Table 5.22: Effort consumed in UAT for Traditional projects.**

Table 5.23 shows the effort spent in UAT in the projects executed using GenNext framework. It is clearly seen that projects executed using traditional methodology overshoot the effort by approximately 8% in average with most of the project having effort deviation of less than 9%.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 136 | 132 | 3.03 |
| GP2 | 128 | 132 | -3.03 |
| GP3 | 144 | 132 | 9.09 |
| GP4 | 164 | 132 | 24.24 |
| GP5 | 144 | 132 | 9.09 |

**Table 5.23: Effort consumed in UAT for GenNext projects.**

Analysis of the Figure 5.8 postulates that most of the projects executed using the GenNext framework completed the UAT within least effort deviation from planned effort. Only one project overshot the effort unexpectedly. The effort in the project was overshot because of a defect found during the UAT of the project. Effort elapsed in the UAT was acceptance testing effort, defect fixing effort, integration testing effort and re-acceptance testing effort. Hence, GenNext framework assures reduction in effort elapsed in acceptance testing.

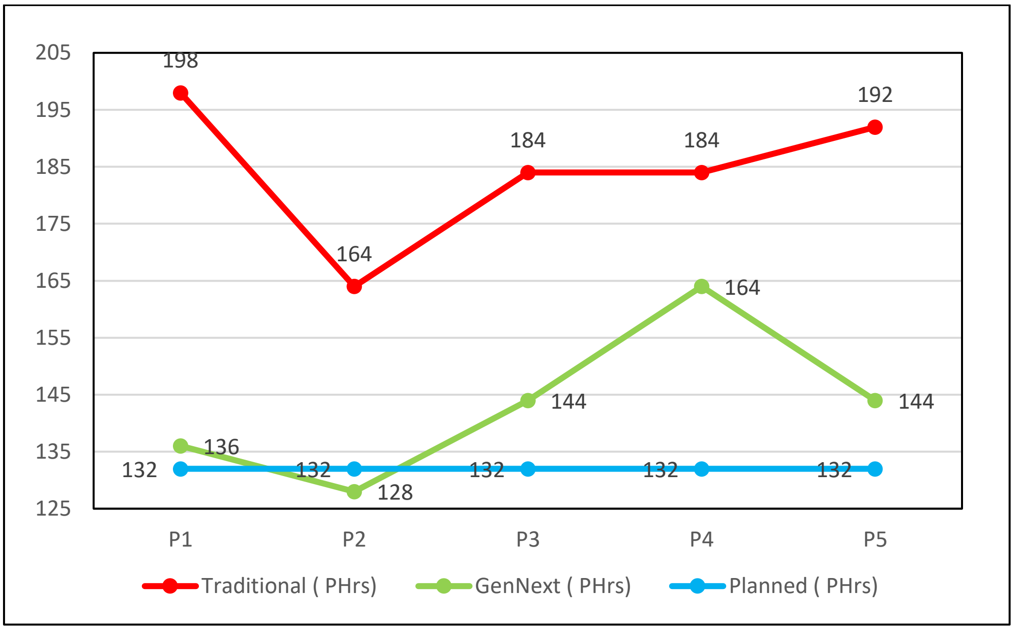


**Figure 5.15: Comparison of the effort consumed in UAT of GenNext projects, Traditional projects with planned effort.**

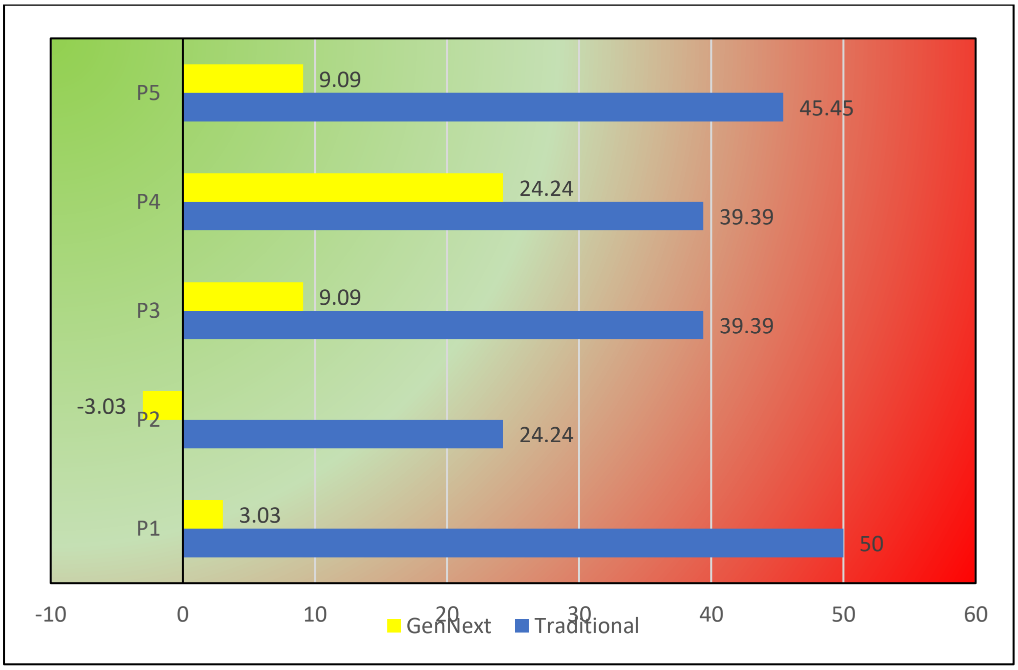


**Figure 5.16: Comparison of the effort deviation from planned effort in UAT for GenNext projects and Traditional projects.**

To evaluate further, the results were evaluated statistically using the hypothesis HYP1H (defined in section 1.4). The hypotheses are given below -

> *$H_{1H0}$: UAT activities in GenNext Projects consume* ***more or equal effort*** *than traditional projects.*
>
> *$H_{1Ha}$: UAT activities in GenNext Projects consume* ***less effort*** *than traditional projects.*

Statistical analysis of the results is given below in Table 5.24. It is found that critical value is 2.306 which is less than the t value 4.9671 and the p value is 0.001097 which is less than the α value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{1H0}$ should be rejected. The risk to reject $H_{1H0}$ while it is true is less than 0.01%. This signifies *ERP Implementation Project executed with GenNext framework consume lesser effort in* UAT *that of executed with traditional methodology.*

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 184.4 | 143.2 |
| Variance | 164.8 | 179.2 |
| Stand. Dev. | 12.8374 | 13.3866 |
| N | 5 | 5 |
| t (observed value) | 4.961 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | 0.001097 | |
| α | 0.05 | |

**Table 5.24: Statistical analysis of the effort consumed in the UAT for Traditional projects and GenNext projects.**

## 5.5 ANALYSIS OF THE TOTAL EFFORT

Total effort elapsed in the ERP implementation is the sum of efforts elapsed in all activities of an ERP implementation. It is one of the major indices indicating overall health of the project. The Table 5.25 shows the total effort elapsed in the projects executed using the traditional methodology and the Table 5.26 shows the shows the total effort elapsed in the projects executed using the GenNext framework. Efforts for various activities for both traditional and GenNext projects are given Appendix 3.

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| TP1 | 3740 | 5238 | 40.05 |
| TP2 | 3740 | 5175 | 38.37 |
| TP3 | 3740 | 5332 | 42.57 |
| TP4 | 3740 | 5384 | 43.96 |
| TP5 | 3740 | 5274 | 41.02 |

**Table 5.25: Total Effort consumed in Traditional projects.**

| Project ID | Effort Spent(PHrs) | Planned Effort(PHrs) | Effort Deviation(%) |
|---|---|---|---|
| GP1 | 3740 | 4386 | 17.27 |
| GP2 | 3740 | 4364 | 16.68 |
| GP3 | 3740 | 4486 | 19.95 |
| GP4 | 3740 | 4348 | 16.26 |
| GP5 | 3740 | 4476 | 19.68 |

**Table 5.26: Total Effort consumed in GenNext projects.**

Table 5.25 and Table 5.26 show that projects executed using the traditional methodology observed an average effort deviation of 41% and projects executed using the traditional methodology observed an average effort deviation of 18%, which was 56% less than the deviation observed in traditional methodology. Traditional projects consumed approximately 20% more effort than the projects executed using GenNext framework. Figure 5.17 shows the comparison of the total effort elapsed in traditional methodology, GenNext framework and planned effort. Figure 5.18 shows the deviation of total effort elapsed in traditional methodology, GenNext framework from planned effort.

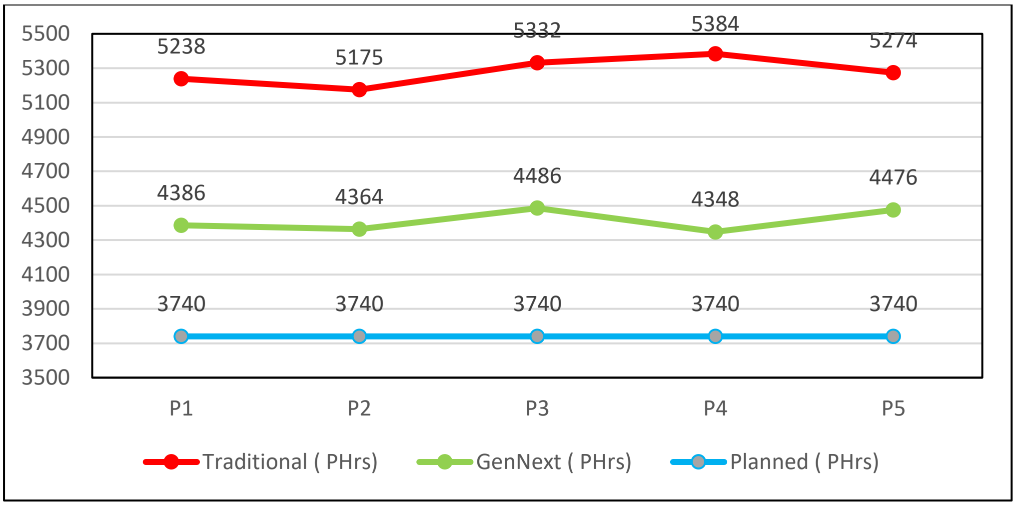


**Figure 5.17: Comparison of the total effort consumed in GenNext projects, Traditional projects with planned effort.**

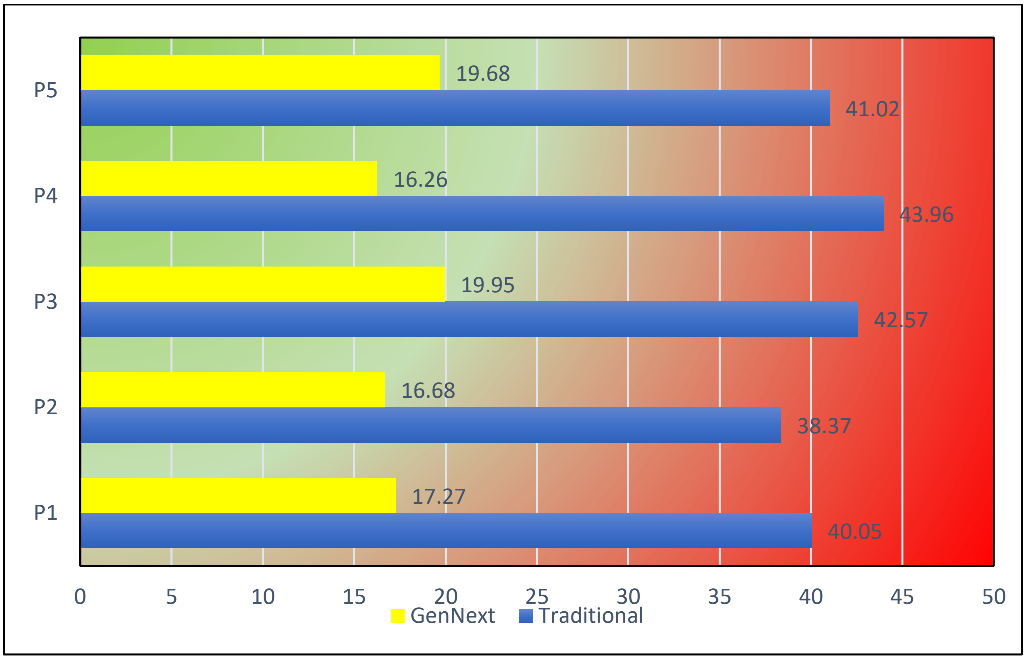


**Figure 5.18: Comparison of the total effort deviation from planned effort in GenNext projects and Traditional projects.**

To evaluate further, the results were evaluated statistically using the hypothesis HYP1 (defined in section 1.4). The hypotheses are given below -

> $H_{10}$: *GenNext projects consume* ***more or equal effort*** *than traditional projects.*

> $H_{1a}$: *GenNext projects consume* ***less effort*** *than traditional projects.*

Statistical analysis of the results is given below in Table 5.27. It is found that critical value is 2.306 which is less than the t value 18.7346 and the p value is 0.00001 which is less than the α value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{10}$ should be rejected. The risk to reject $H_{10}$ while it is true is less than 0.01%.

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 5280.6 | 4412 |
| Variance | 6585.8 | 4162 |
| Stand. Dev. | 81.1529 | 64.5136 |
| N | 5 | 5 |
| t (observed value) | 18.7346 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | 0.00001 | |
| α | 0.05 | |

**Table 5.27: Statistical analysis of the total effort consumed in Traditional projects and GenNext projects.**

## 5.6 ANALYSIS OF COST OF QUALITY (COQ) IN ERP IMPLEMENTATION

In case of ERP or any COTS implementations, COQ is measured at every phase and total COQ is calculated to describe the overall health of the project and product. The study concentrates on the following COQs

- $COQ_{Requirements}$ – COQ for the verification, reviews, validation and rework of requirements i.e. Gap Analysis
- $COQ_{Design}$ – COQ for the verification, reviews, validation and rework of design of configurations and customizations.
- $COQ_{Coding}$ – COQ for the verification, reviews, validation and rework of configurations and development of customizations.

- COQ $_{Total}$ – COQ for the verification, reviews, validation and rework of complete rework

Table 5.28 and Table 5.29 illustrates the COQ $_{Requirement}$ for the projects executed using traditional methodology and GenNext framework. Comparison of both tables suggests that projects executed using the traditional methodology overshot the planned COQ$_{Requirement}$ and have an average deviation of approximately 28 % while the projects executed using the GenNext framework undershot the planned COQ $_{Requirement}$ and have an average deviation of approximately -3 %.

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 24.71 | 20 | 23.55 |
| TP2 | 24.33 | 20 | 21.65 |
| TP3 | 28.07 | 20 | 40.35 |
| TP4 | 25.37 | 20 | 26.85 |
| TP5 | 25.81 | 20 | 29.05 |

**Table 5.28: COQ$_{Requirement}$ for projects executed using traditional methodologies.**

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 19.47 | 20 | -3.63 |
| GP2 | 19.24 | 20 | -5.3 |
| GP3 | 18.9 | 20 | -3.26 |
| GP4 | 20.09 | 20 | -2.5 |
| GP5 | 20.09 | 20 | -1.1 |

**Table 5.29: COQ$_{Requirement}$ for projects executed using GenNext framework.**

Table 5.30 and Table 5.31 illustrates the COQ $_{Design}$ for the projects executed using traditional methodology and GenNext framework. Comparison of both the tables suggests that projects executed using the traditional methodology overshot the planned COQ $_{Design}$ and have an average deviation of approximately 28% while the projects executed using the GenNext framework undershot the planned $COQ_{Design}$ and have an average deviation of approximately -16%.

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 26.46 | 20 | 32.3 |
| TP2 | 24.62 | 20 | 23.1 |
| TP3 | 24.66 | 20 | 23.3 |
| TP4 | 27.38 | 20 | 36.9 |
| TP5 | 24.52 | 20 | 22.6 |

**Table 5.30: $COQ_{Design}$ for projects executed using traditional methodology.**

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 16.27 | 20 | -18.65 |
| GP2 | 16.77 | 20 | -16.15 |
| GP3 | 17.57 | 20 | -12.15 |
| GP4 | 16.16 | 20 | -19.2 |
| GP5 | 16.92 | 20 | -15.4 |

**Table 5. 31: $COQ_{Design}$ for projects executed using GenNext framework.**

Table 5.32 and Table 5.33 illustrates the $COQ_{Coding}$ for the projects executed using traditional methodology and GenNext framework. Coding as an activity is always known for high COQ because of high cost of review processes, high cost of fixing the defect and re-review price. Comparison of both the tables suggests that the projects executed using the traditional methodology overshot the planned $COQ_{Coding}$ and have an average deviation of approximately 20 % and the projects executed using the GenNext framework undershot the planned $COQ_{Coding}$ and have an average deviation of approximately 2.6 %.

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 26.46 | 20 | 32.3 |
| TP2 | 24.62 | 20 | 23.1 |
| TP3 | 24.66 | 20 | 23.3 |
| TP4 | 27.38 | 20 | 36.9 |
| TP5 | 24.52 | 20 | 22.6 |

**Table 5.32: $COQ_{Coding}$ for projects executed using traditional methodology.**

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 20.67 | 20 | 3.35 |
| GP2 | 19.92 | 20 | -0.4 |
| GP3 | 20.32 | 20 | 1.6 |
| GP4 | 20.34 | 20 | 1.7 |
| GP5 | 21.13 | 20 | 5.65 |

**Table 5.33: $COQ_{Coding}$ for projects executed using GenNext framework.**

$COQ_{Total}$ is the ratio of the total effort on quality to the total effort elapsed in the implementation. $COQ_{Total}$ is an indicator of the quality efforts in any process. Table 5.34 shows the $COQ_{Total}$ of the projects implemented using the traditional methodology. Careful observation of the Table 5.34 shows that all projects executed using traditional methodology overshot the planned COQ with average deviation of 13% and have minimum deviation of approximately 8% and maximum deviation of approximately 18%.

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 33.33 | 30 | 11.1 |
| TP2 | 32.46 | 30 | 8.2 |
| TP3 | 33.91 | 30 | 13.03 |
| TP4 | 34.62 | 30 | 15.4 |
| TP5 | 35.27 | 30 | 17.56 |

**Table 5. 34: $COQ_{Total}$ for projects executed using traditional methodology.**

Table 5.35 shows the $COQ_{Total}$ of the projects implemented using the GenNext framework. Table 5.35 shows that all projects executed using GenNext framework undershot the planned COQ with average deviation of -3% and have minimum deviation of approximately -1% and maximum deviation of approximately -5%.

| Project ID | COQ (Actual) | COQ (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 28.91 | 30 | -3.63 |
| GP2 | 28.41 | 30 | -5.3 |
| GP3 | 29.02 | 30 | -3.26 |
| GP4 | 29.25 | 30 | -2.5 |
| GP5 | 29.67 | 30 | -1.1 |

**Table 5. 35: $COQ_{Total}$ for projects executed using GenNext framework.**

Figure 5.19 illustrates that all the projects executed using traditional methodology always incurred more COQ but on the other hand, projects executed using GenNext framework always incurred low COQ than planned. Figure 5.20 compares the deviation of actual COQ from planned COQ for projects executed using traditional methodology and GenNext framework.

To evaluate further, the results were evaluated statistically using the hypothesis HYP2 (defined in section 1.4). The hypotheses are given below -

> *$H_{20}$: GenNext projects consume **more or equal COQ** than traditional projects.*
>
> *$H_{2a}$: GenNext projects consume **less COQ** than traditional projects.*

Statistical analysis of the results is given below in Table 5.36. It is found that critical value is 2.306 which is less than the t value 9.5182 and the p value is 0.00001 which is less than the $\alpha$ value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{20}$ should be rejected. The risk to reject $H_{20}$ while it is true is less than 0.01%.

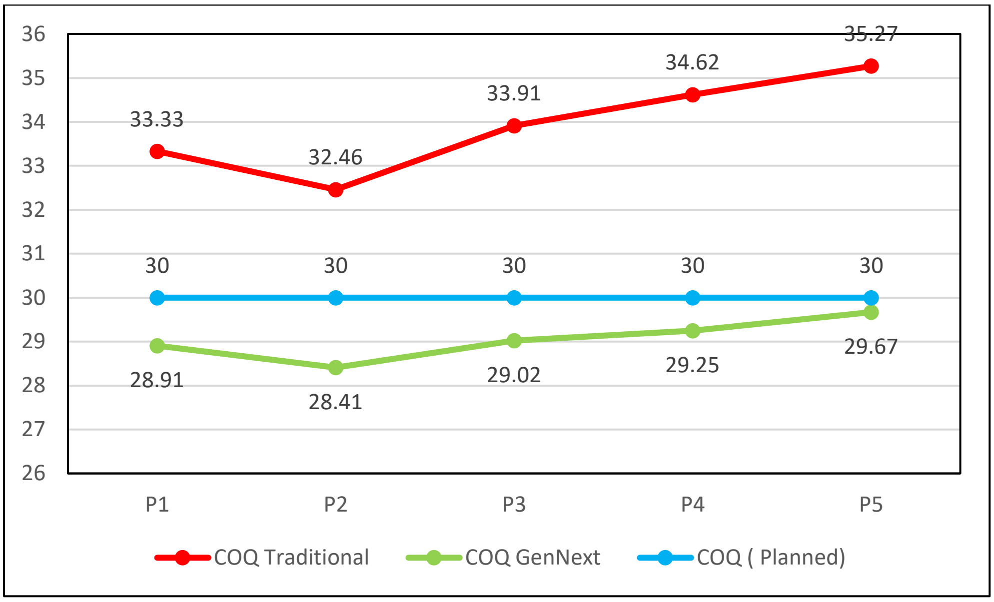


**Figure 5.19: Comparison of the $COQ_{Total}$ for GenNext Projects, Traditional Projects with planned $COQ_{Total}$.**

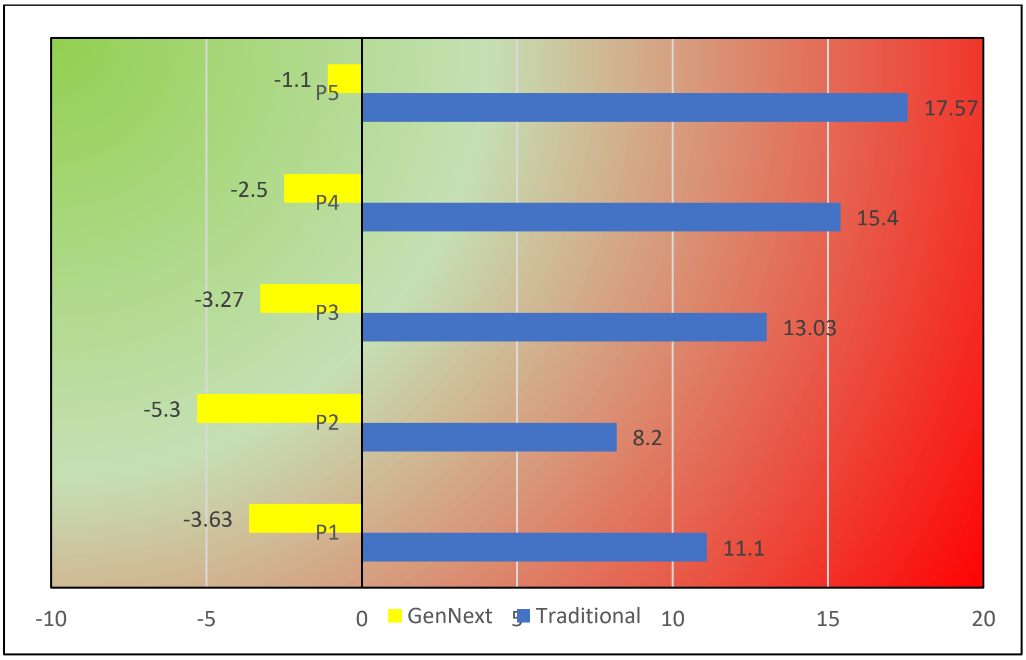


**Figure 5.20: Comparison of the deviation from planned $COQ_{Total}$ with $COQ_{Total}$ for GenNext projects and Traditional projects.**

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 33.918 | 29.052 |
| Variance | 1.1981 | 0.2136 |
| Stand. Dev. | 1.0946 | 0.4622 |
| N | 5 | 5 |
| t (observed value) | 9.1582 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | <0.00001 | |
| α | 0.05 | |

**Table 5.36: Statistical analysis of the $COQ_{Total}$ for Traditional projects and GenNext projects.**

### 5.7 ANALYSIS OF DEFECTS IN ERP IMPLEMENTATION

Table 5.37 below shows the defects captured in the various phase of implementation and the corresponding source of defect for the projects implemented using traditional methodology. Defects found in the SIT and UAT with source as SIT and UAT are not considered because of very low values. Analysis of the table suggests that defect containment ratio (i.e. Ratio of Defect captured in same phase that of origin to the total number of defects.) is highest for the coding phase for most of the projects. On the other hand, it is least for the defects occurred in the requirements specification phases. Coding phase has maximum number of defects introduced and highest distribution of the defects. Coding phase, alone, is responsible for approximately 60% of defects while the Requirement Specification and Design phase each roughly contributes to 20% of total defects. In table 5.37 Requirement specification, Distribution, Defect Containment Ratio is RS, DIST, DCR respectively.

| P1 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Phase captured/ Source | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 44 | 27 | 13 | 12 | 3 | 4 | 103 | 17.76 | 42.72 |
| Design | 0 | 62 | 18 | 14 | 7 | 3 | 104 | 17.93 | 59.62 |
| Coding | 0 | 0 | 193 | 119 | 34 | 17 | 363 | 62.59 | 53.17 |
| SIT | 0 | 0 | 0 | 7 | 0 | 0 | 7 | 1.21 | |
| UAT | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 0.52 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 44 | 89 | 224 | 152 | 47 | 24 | 580 | 100 | |
| P2 | | | | | | | | | |
| Phase captured/ Source | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 28 | 19 | 12 | 3 | 0 | 0 | 62 | 13.14 | 45.16 |
| Design | 0 | 68 | 40 | 19 | 0 | 3 | 130 | 27.54 | 52.31 |
| Coding | 0 | 0 | 130 | 122 | 19 | 2 | 273 | 57.84 | 47.62 |
| SIT\ | 0 | 0 | 0 | 7 | 0 | 0 | 7 | 1.48 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 28 | 87 | 182 | 151 | 19 | 5 | 472 | 100 | |
| P3 | | | | | | | | | |
| Phase captured/ Source | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 34 | 23 | 28 | 9 | 0 | 0 | 94 | 18.04 | 36.17 |
| Design | 0 | 31 | 44 | 19 | 13 | 0 | 107 | 20.54 | 28.97 |
| Coding | 0 | 0 | 153 | 132 | 27 | 8 | 320 | 61.42 | 47.81 |
| SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 34 | 54 | 225 | 160 | 40 | 8 | 521 | 100 | |

| P4 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Phase captured/ Source | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 33 | 23 | 18 | 19 | 4 | 0 | 97 | 17.54 | 34.02 |
| Design | 0 | 24 | 48 | 24 | 3 | 0 | 99 | 17.9 | 24.24 |
| Coding | 0 | 0 | 168 | 156 | 24 | 9 | 357 | 64.56 | 47.06 |
| SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 33 | 47 | 234 | 199 | 31 | 9 | 553 | 100 | |
| P5 | | | | | | | | | |
| Phase captured/ Source | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 24 | 45 | 23 | 13 | 2 | 0 | 107 | 17.46 | 22.43 |
| Design | 0 | 36 | 13 | 48 | 19 | 0 | 116 | 18.92 | 31.03 |
| Coding | 0 | 0 | 140 | 209 | 39 | 2 | 390 | 63.62 | 35.9 |
| SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 24 | 81 | 176 | 270 | 60 | 2 | 613 | 100 | |

**Table 5.37: Phase-wise distribution defects captured for each source phase for Traditional projects.**

The total number of defects in each phase and complete project is alarming. Defects are wasteful activities as the time spent on finding and fixing the defects does not add any value to the system. Fixing the defect in subsequent phase is always costlier than fixing the defect in the phase of origin. Ideally, no defect should escape and move to subsequent phases but owing to limited exposure to each type of defect, situation and nature of defects, it is humanly impossible to find out defects at the origin itself. This implies that defect containment ratios should be high. Quality is measured by the number of defects and shares inversely proportional relationship with number of defects. GenNext framework has prime liability to reduce waste and increase quality i.e. reduce number of defects and reduce the number of escaped defects i.e. increasing the defect containment ratio.

Table 5.38 illustrates the phase of origin of defects and corresponding phases in which the defects were discovered for the projects executed using GenNext framework Analysis of this table points that combined defects in design phase and requirement phase are approximately 30% of the total and highest number of defects are found in the coding phase. Analysis also suggests that defect containment of the requirement phases, design phases and coding phases have increased drastically in case of projects executed using GenNext framework. In addition, the number of defects has reduced drastically for the projects executed using GenNext framework. This reduction in number of defects and increase in the efficiency of validation and verification has increased owing to continuous involvement of users, continuous integrations and deployment, group reviews with an eye to the complete business process and understanding of upstream and downstream systems, which helps in optimizing the whole value chain. The high defect containment and reduction in number of defects is in line with reduction in the time or effort elapsed in SIT, UAT, and reduction in total defects in SIT and UAT phases of each project. In table 5.37 Requirement specification, Distribution, Defect Containment Ratio is RS, DIST, DCR respectively

| P1 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 24 | 17 | 4 | 2 | 0 | 0 | 47 | 16.32 | 51.06 |
| Design | 0 | 28 | 12 | 9 | 0 | 0 | 49 | 17.01 | 57.14 |
| Coding | 0 | 0 | 103 | 70 | 12 | 0 | 185 | 64.24 | 55.68 |
| SIT | 0 | 0 | 0 | 4 | 0 | 0 | 4 | 1.39 | |
| UAT | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 1.04 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 24 | 45 | 119 | 85 | 15 | 0 | 288 | 100 | |

| P2 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 24 | 18 | 8 | 3 | 0 | 0 | 53 | 17.49 | 45.28 |
| Design | 0 | 22 | 15 | 8 | 0 | 0 | 45 | 14.85 | 48.89 |
| Coding | 0 | 0 | 103 | 84 | 13 | 2 | 202 | 66.67 | 50.99 |
| SIT | 0 | 0 | 0 | 3 | 0 | 0 | 3 | 0.99 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 24 | 40 | 126 | 98 | 13 | 2 | 303 | 100 | |

| P3 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 20 | 14 | 5 | 2 | 0 | 0 | 41 | 13.49 | 48.78 |
| Design | 0 | 31 | 16 | 9 | 0 | 0 | 56 | 18.42 | 55.36 |
| Coding | 0 | 0 | 124 | 70 | 13 | 0 | 207 | 68.09 | 59.9 |
| SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 20 | 45 | 145 | 81 | 13 | 0 | 304 | 100 | |

| P4 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 23 | 15 | 4 | 3 | 0 | 0 | 45 | 16.92 | 51.11 |
| Design | 0 | 24 | 19 | 2 | 0 | 0 | 45 | 16.92 | 53.33 |
| Coding | 0 | 0 | 92 | 76 | 8 | 0 | 176 | 66.17 | 52.27 |
| SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 23 | 39 | 115 | 81 | 8 | 0 | 266 | 100 | |

| P5 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RS | Design | Coding | SIT | UAT | Go Live | Sum | DIST | DCR |
| RS | 24 | 18 | 0 | 2 | 1 | 0 | 45 | 13.47 | 53.33 |
| Design | 0 | 32 | 13 | 0 | 2 | 0 | 47 | 14.07 | 68.09 |
| Coding | 0 | 0 | 140 | 85 | 15 | 2 | 242 | 72.46 | 57.85 |
| SIT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| UAT | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Go Live | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 24 | 50 | 153 | 87 | 18 | 2 | 334 | 100 | |

**Table 5.38: Phase wise distribution defects captured for each source phase for GenNext projects.**

It is clearly visible from the Table 5.39 that the projects executed using the GenNext framework reported approximately 50% less defects on average for each project.  Figure 5.21 and Figure 5.22 presents the number of defects in each project executed using traditional methodology and executed using GenNext framework w.r.t planned defects.

| Project ID | No of defects (Traditional) | No of defects (GenNext) |
|---|---|---|
| P1 | 580 | 288 |
| P2 | 472 | 303 |
| P3 | 521 | 304 |
| P4 | 553 | 266 |
| P5 | 613 | 334 |

**Table 5.39: Comparison of the defects discovered in Traditional projects and GenNext projects.**

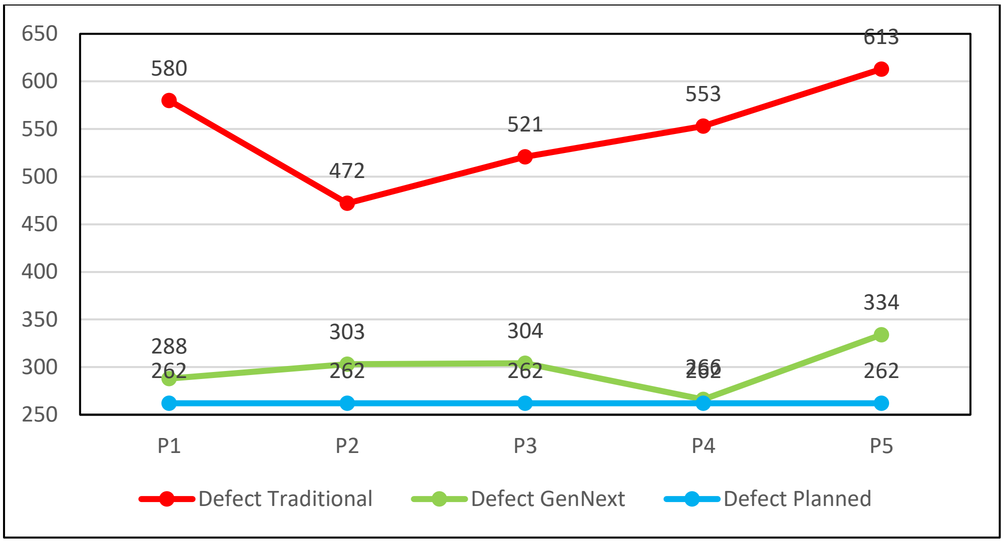


**Figure 5.21: Comparison of the Total Defects in the GenNext projects and Traditional projects with planned defects.**

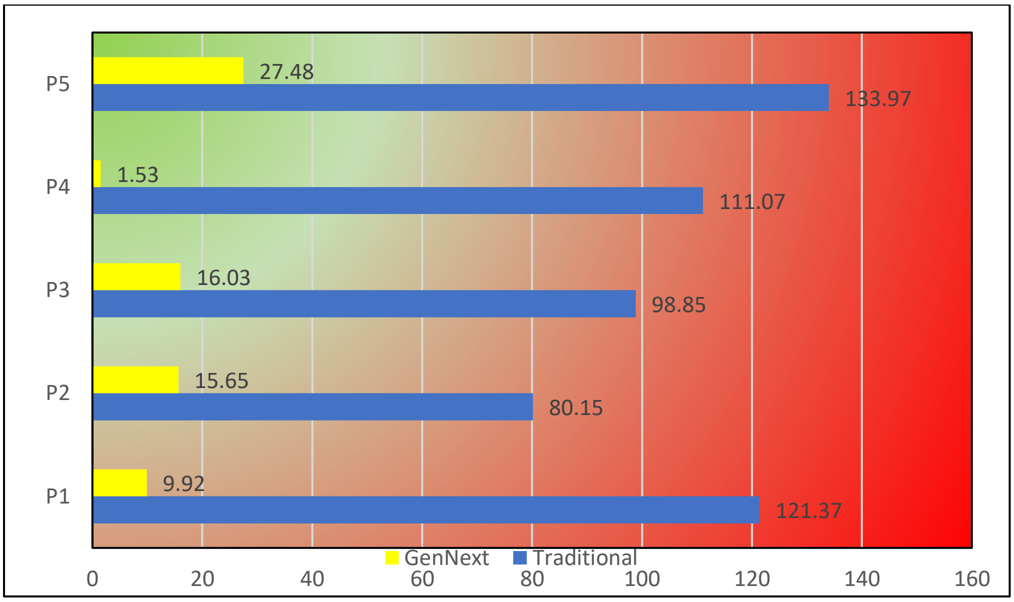


**Figure 5.22: Comparison of the deviation of total defects from planned total defects in GenNext Projects and Traditional projects.**

Table 5.40 shows the statistical analysis of the defect. It is found out that critical value is 2.306 which is less than the t value 9.3193 and the p value is 0.00001 which is less than the α value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{30}$, taken from HYP3 (defined in section 1.4), should be rejected. The risk to reject $H_{30}$ while it is true is less than 0.01%.

*$H_{30}$: GenNext projects show* ***lesser quality*** *than traditional projects.*

*$H_{3a}$: GenNext projects show* ***better or equal quality*** *than traditional projects.*

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 547.8 | 299 |
| Variance | 2944.7 | 619 |
| Stand. Dev. | 54.2651 | 24.8797 |
| N | 5 | 5 |
| t (observed value) | 9.3193 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | <0.00001 | |
| α | 0.05 | |

**Table 5.40: Statistical analysis of the no of defects in Traditional projects and GenNext projects.**

## 5.8 ANALYSIS OF DEFECT INJECTION RATE (DIR) IN ERP IMPLEMENTATION

From the Table 5.1 & 5.2 and Table 5.37 & 5.38, DIR for requirement specification activity is calculated as given below in Table 5.41 for projects executed using traditional methodology and Table 5.42 for projects executed using GenNext framework.

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 9.79 | 5 | 95.8 |
| TP2 | 5.89 | 5 | 17.8 |
| TP3 | 9.16 | 5 | 83.2 |
| TP4 | 9.05 | 5 | 81 |
| TP5 | 10.15 | 5 | 103 |

**Table 5.41: DIR for requirement specification phase in traditional projects.**

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 5.2 | 5 | 4 |
| GP2 | 5.93 | 5 | 18.6 |
| GP3 | 4.51 | 5 | -9.8 |
| GP4 | 4.91 | 5 | -1.8 |
| GP5 | 4.91 | 5 | -1.8 |

**Table 5.42: DIR for requirement specification phase GenNext projects.**

Analysis of the Table 5.41 suggests that the projects executed using the traditional methodologies have an average DIR of 8 with most of the projects having DIR above 9 and one of the projects has DIR of 5.89. Average deviation from planned DIR is 76% for these projects. In case of projects, executed using GenNext projects (Table 5.42), majority of the projects had the DIR less than 5 (planned DIR) and 2 projects had DIR more than 5. The highest DIR reported in case of GenNext projects was in range of the lowest DIR reported for projects executed using projects with traditional methodology.

From the Tables 5.4 & 5.5 and Tables 5.37 & 5.38, DIR for design activity is calculated as given below in Table 5.43 for projects executed using traditional methodology and Table 5.44 for projects executed using GenNext framework.

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 10.12 | 5 | 102.4 |
| TP2 | 12.5 | 5 | 150 |
| TP3 | 10.31 | 5 | 106.2 |
| TP4 | 9.41 | 5 | 88.2 |
| TP5 | 11.11 | 5 | 122.2 |

**Table 5.43: DIR for design phase for the traditional projects.**

| Project ID | DIR (Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 5.25 | 5 | 5 |
| GP2 | 4.84 | 5 | -3.2 |
| GP3 | 5.86 | 5 | 17.2 |
| GP4 | 4.91 | 5 | -1.8 |
| GP5 | 5.03 | 5 | 0.6 |

**Table 5.44: DIR for design phase for the GenNext projects.**

Analysis of the Table 5.43 suggests that projects executed using the traditional methodologies have an average DIR of 10.7 and all projects have DIR approximately 10 or above. Average deviation from planned DIR is 113% for these projects. In case of projects executed using GenNext projects, Table 5.44 shows that majority of the projects have DIR more than 5 (planned DIR) but less than 6 and 2 projects had DIR less than 5. The highest DIR reported in case of GenNext projects was 60% less. The lowest DIR was reported for projects executed with traditional methodology.

From the Tables 5.7 & 5.8 and Table 5.34 & 5.35, DIR for customization and configuration is calculated as given below in Table 5.45 for projects executed using traditional methodology and Table 5.46 for projects executed using GenNext framework.

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 15.25 | 10 | 52.5 |
| TP2 | 11.52 | 10 | 15.2 |
| TP3 | 12.84 | 10 | 28.4 |
| TP4 | 14.49 | 10 | 44.9 |
| TP5 | 16.22 | 10 | 62.2 |

**Table 5.45: DIR for configuration and customization phase for the traditional projects**

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 9.66 | 10 | -3.4 |
| GP2 | 10.48 | 10 | 4.8 |
| GP3 | 10.47 | 10 | 4.7 |
| GP4 | 9.32 | 10 | -6.8 |
| GP5 | 12.23 | 10 | 22.3 |

**Table 5.46: DIR for configuration and customization phase for GenNext projects.**

Analysis of the Table 5.45 suggests that projects executed using the traditional methodologies have an average DIR of 14.6 and all projects have DIR above 10. Average deviation from planned DIR is 40% for these projects. In case of projects executed using GenNext projects, Table 5.46 shows that most projects have DIR more than 10 (planned DIR) and 2 projects had DIR less than 10. The highest DIR reported in case of GenNext projects was in range of lowest DIR reported for projects executed with traditional methodology.

Table 5.47 shows the DIR for the complete lifecycle of the projects executed using traditional methodology. It is visible that all projects overshot the DIR by average of 40%.

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| TP1 | 11.07 | 7 | 52.5 |
| TP2 | 9.12 | 7 | 15.2 |
| TP3 | 9.77 | 7 | 28.4 |
| TP4 | 10.27 | 7 | 44.9 |
| TP5 | 11.62 | 7 | 62.2 |

**Table 5.47: DIR for the complete lifecycle of the traditional projects.**

Table 5.48 shows the DIR for the complete lifecycle of the projects executed using GenNext framework. These projects showed an average deviation of -3% with lowest deviation being -12.57% and highest being 6.57%. The low over all DIR can be attributed to less number of defects in SIT, UAT phases that is possible because of inclusion of stakeholders during each phase of life cycle, considering the whole life cycle at each points and continuous integration of build into the system.

| Project ID | DIR(Actual) | DIR (Planned) | Deviation (%) |
|---|---|---|---|
| GP1 | 6.57 | 7 | -6.14 |
| GP2 | 6.94 | 7 | -0.86 |
| GP3 | 6.78 | 7 | -3.14 |
| GP4 | 6.12 | 7 | -12.57 |
| GP5 | 7.46 | 7 | 6.57 |

**Table 5.48: DIR for the complete lifecycle of the GenNext projects.**

The comparison of the DIR of complete life cycle of the projects executed using the GenNext framework and traditional model with that of the planned DIR is shown in the Figure 5.23. It is evident from the figure that the projects executed using the GenNext have a DIR less than planned DIR i.e. introduced less defect per 100 hrs. while the projects executed using, traditional methodology introduced more defects per 100 hrs. and overshot the planned DIR. Figure 5.24 shows the deviation of observed DIR from planned DIR incase of traditional and GenNext projects.

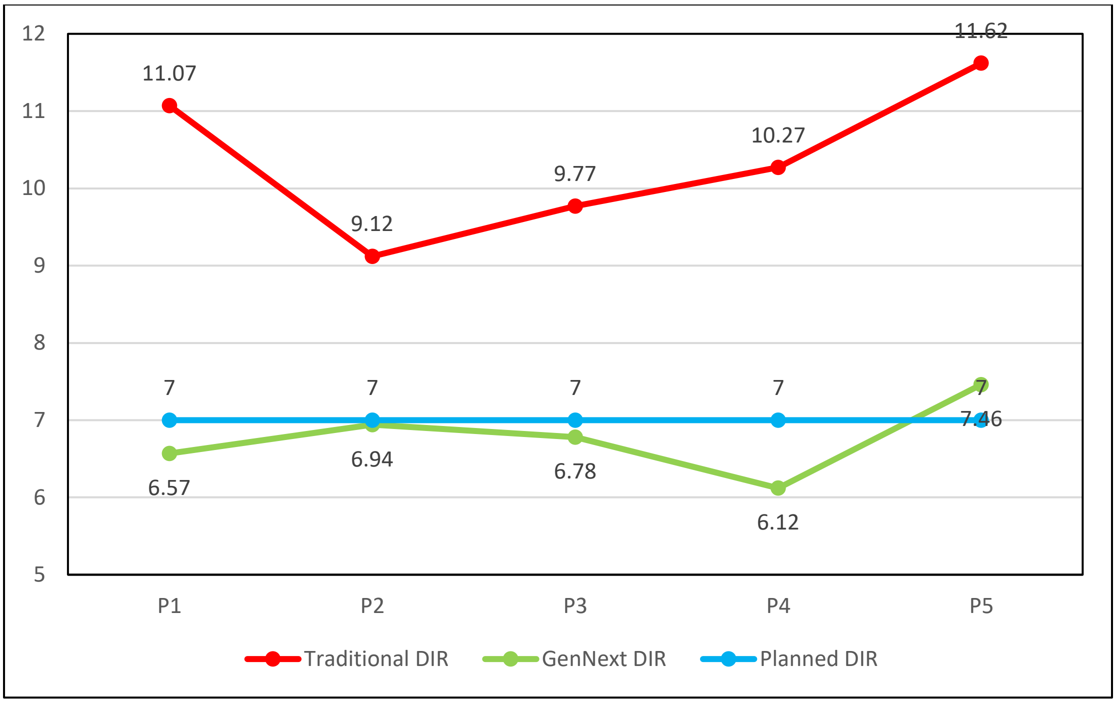


**Figure 5.23: Comparison of the DIR of complete life cycle of the GenNext projects and Traditional projects with planned DIR.**

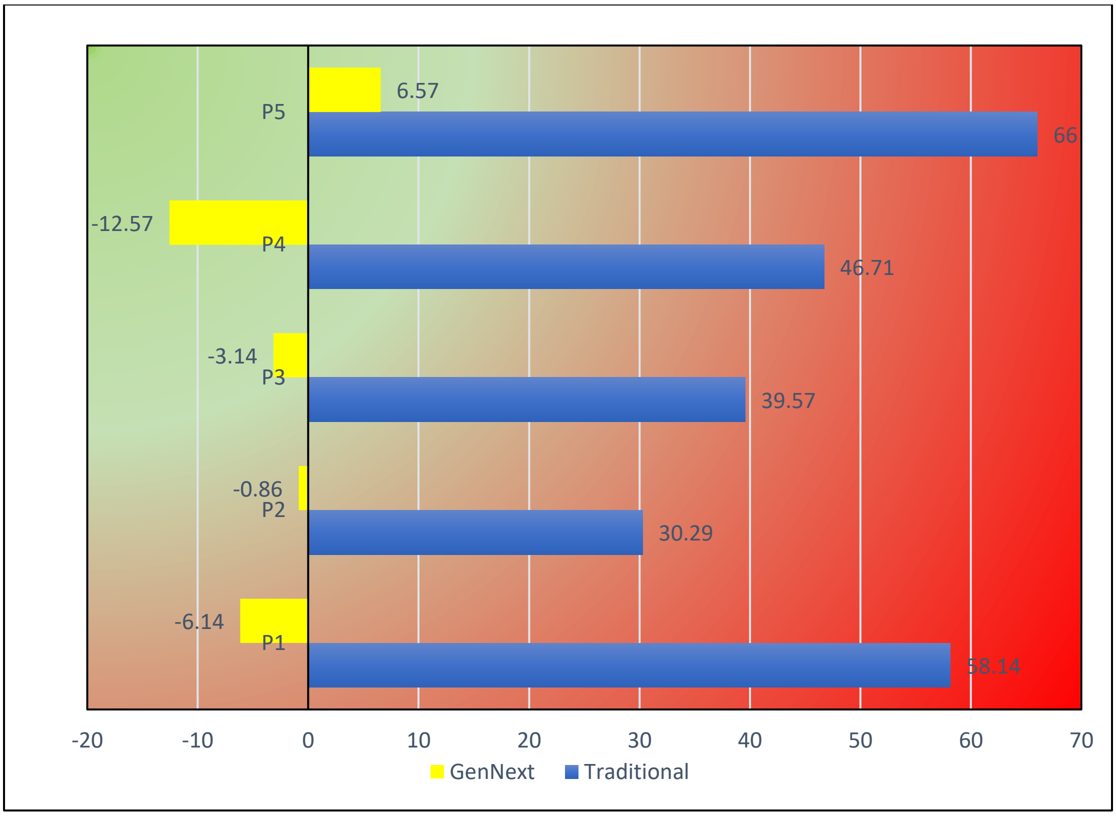


**Figure 5.24: Comparison of the deviation of DIR of complete life cycle of the traditional projects and GenNext projects.**

To evaluate further, the results were evaluated statistically using the hypothesis HYP4 (defined in section 1.4). The hypotheses are given below -

*$H_{40}$: GenNext projects have **higher or equal DIR** than traditional projects.*

*$H_{4a}$: GenNext projects have **lesser DIR** than traditional projects.*

Statistical analysis of the results is given below in Table 5.49. It is found that critical value is 2.306 which is less than the t value 7.8021 and the p value is 0.00026 which is less than the α value of 0.05. The t-value analysis and p-value suggests that results are significant and the Null hypothesis $H_{40}$ should be rejected. The risk to reject $H_{40}$ while it is true is less than 0.01%. Thus, it can be concluded that *ERP Implementation Projects executed with framework show lesser DIR than that of executed with traditional methodology.*

| | Traditional Methodology | GenNext Framework |
|---|---|---|
| Mean | 10.37 | 6.678 |
| Variance | 0.9963 | 0.1233 |
| Stand. Dev. | 0.9981 | 0.3511 |
| n | 5 | 5 |
| t (observed value) | 7.8021 | |
| Degrees of freedom | 8 | |
| t (Critical value) | 2.306 | |
| p-value | 0.00026 | |
| α | 0.05 | |

**Table 5.49: Statistical analysis of the DIR for traditional projects and GenNext projects.**

Next chapter discusses the effectiveness of the GenNext framework and concludes if it works better than traditional methodology. Also, the future scope and benefits of GenNext are discussed in the next chapter.

# CHAPTER 6: CONCLUSION AND FUTURE SCOPE

## 6.1 CONCLUSION

The objective of the new framework was to minimize the total effort elapsed in implementation, minimize the defect injection rate (DIR), minimize the cost incurred in reviews, fixing the review comments and issues i.e. cost of quality (COQ), maximize quality i.e. minimize the number of defects and minimize the effort elapsed in testing. It can be easily concluded by figure 6.1 that GenNext Framework is able to show more value-add than traditional framework. The aim of minimizing is to perform at or under the planned values or have minimum deviation from planned values. The mean of the results obtained from the projects executed using the GenNext framework and traditional methodologies are compared together on the parameters selected in figure 6.2. It is evident from figure 6.2 that GenNext framework has the least deviation from the curve plotted with planned values and the traditional methodologies have the maximum deviation from the planned values. Projects executed using the GenNext framework produces the average DIR and $COQ_{Total}$ which is well within the limits of planned $DIR_{Total}$ and $COQ_{Total.}$ Figure 6.2 helps to conclude that GenNext framework can help to optimize the projects performance. Earlier section of this chapter helps to conclude that results obtained in the projects executed using the GenNext framework are statistically significant. Figure 6.1 compares the value addition and reduction in wasteful activities in traditional project and GenNext projects. It can be easily concluded that GenNext framework increases the value add and reduces the waste. Also, on comparing the overall performance of GenNext projects with Traditional project, it is seen that performance curve plotted GenNext almost superimposes to that of planned values while the performance curve plotted by Traditional projects is very much offset to planned curve. The comparison is shown in figure 6.2. The offset in effort is not widely seen in the figure because of the scale used. The mean values efforts of each phase are individually compared in figure 6.3. It can be easily concluded that while GenNext projects consumed the effort with least deviation to planned effort in each phase and Traditional projects showed the maximum deviation. It will be worthy to note that GenNext projects consumed lesser than planned effort in requirement specification and UAT activities.

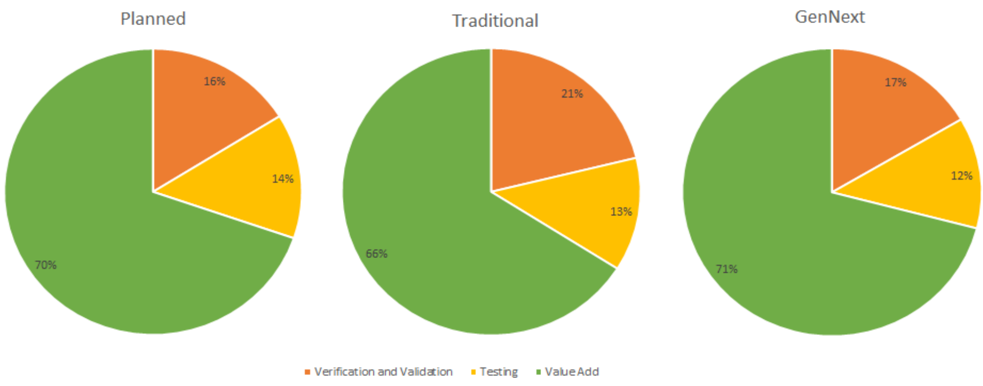


**Figure 6.1: Comparison of value adding activities in GenNext and Traditional projects.**

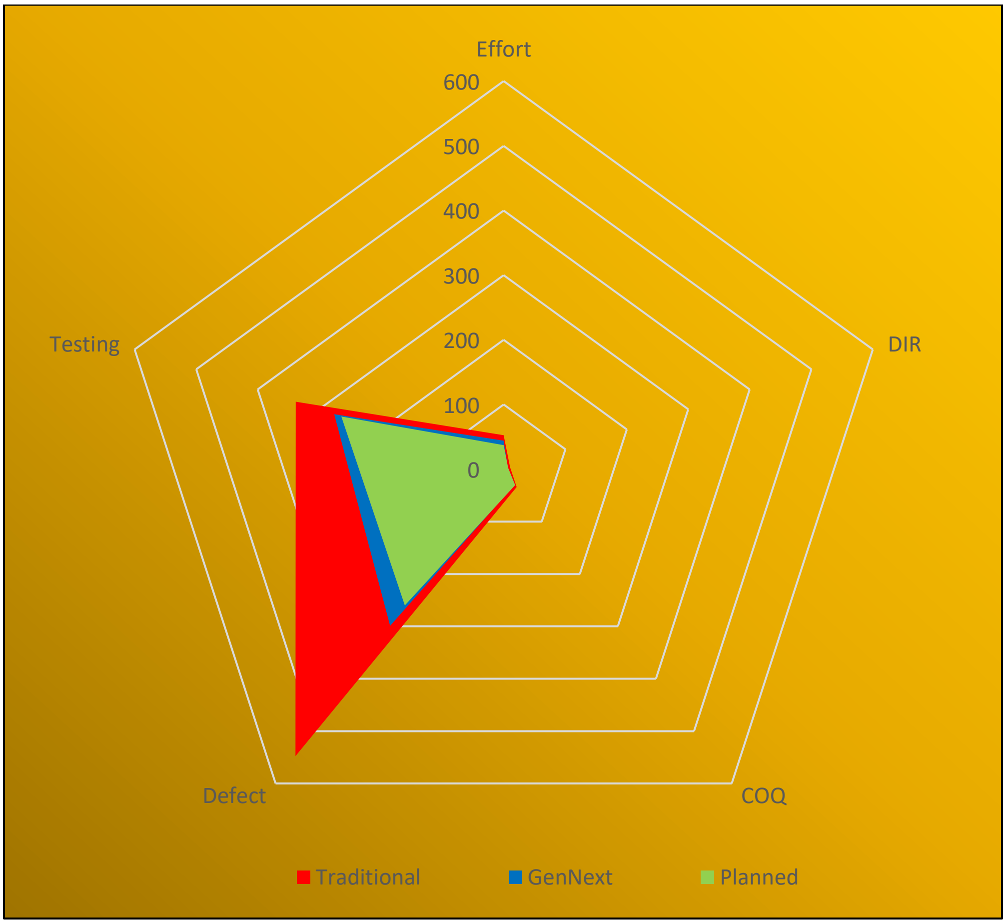


**Figure 6.2: Comparison of the mean results of selected parameters for projects executed using GenNext framework and traditional methodology with planned values.**

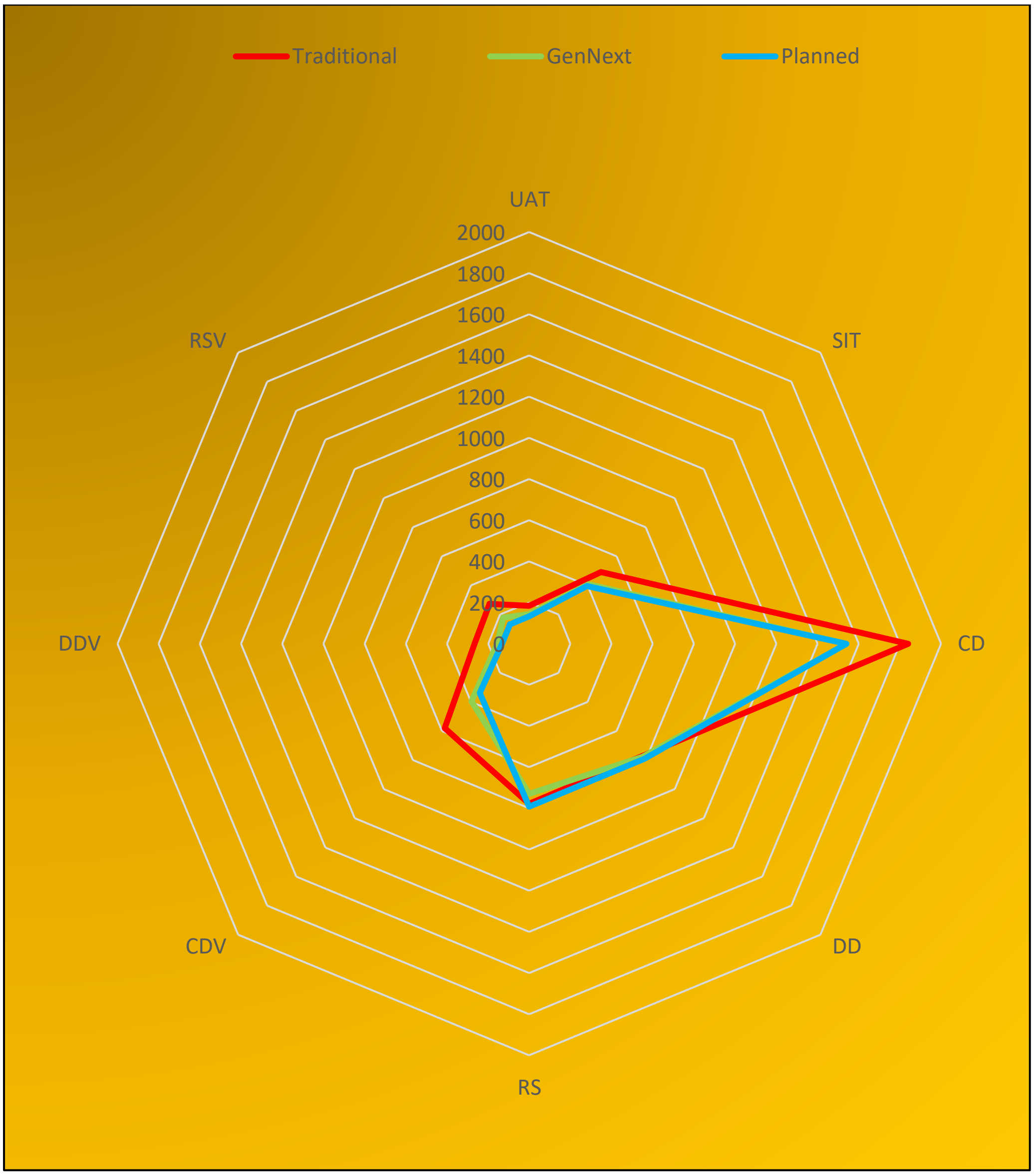


**Figure 6.3: Comparison of the mean effort elapsed in various activities for GenNext Projects, Traditional projects with planned values**

### 6.2 BENEFITS OF PROPOSED FRAMEWORK

The significant achievements of the proposed framework are given below –

1. **A Model for implementing ERP within defined parameters –** GenNext framework provides the step by step approach to implement the ERP system in an organization. The framework presents the methodology to complete ERP implementation with effort, cost and schedule while maintaining the quality.

2. **Prioritization of Requirements –** ERP system consists of the processes which are in turn consists of the sub-processes and features. Proposed algorithm considers the impact of processes, effort and reusability to rank the priority of the requirements.

3. **Managing uncertainties –** Proposed framework considers waste elimination, user involvements and incremental delivery results in responding and managing uncertainties.

4. **Evolution –** Proposed framework provides a platform to apply the lessons learnt and reflect the learning in further activities.

5. **Optimized Operations –** Proposed framework considers the stabilization of system and further evolution of system as part of implementation. This results in stabilized operations**.**

## 6.3 FUTURE SCOPE

The research done as part of the thesis can be extended in the following dimension in the future –

1. **Application in Large projects-** The proposed framework can be applied to the large ERP implantation projects which involve implementation of multiple modules across multiple geographies.

2. **Application in other COTS implementation-** The proposed framework can be applied to the implementations of COTS systems other than ERP system. The other COTS systems can be Enterprise BI implementation or Data warehousing implementations.

3. **Addition of factors in Process Prioritization algorithm –** Currently proposed algorithm considers only limited factors. More factors such as effort required in regression or performance testing can be found out to augment the algorithm.

# CHAPTER 7: REFERENCES


[1]. A Guide to the Project Management Body of Knowledge (PMBOK Guide). Newtown Square, Pa: Project Management Institute, Inc; 2004.

[2]. Abrahamsson, P., Conboy, K. and Wang, X. (2009), Lots done, more to do: the current state of agile systems development research. European Journal of Information Systems, Vol. 18 No.4, pp. 281–28.

[3]. Agarwal, A., Shankar, R. and Tiwari, M.K. (2006), Modeling the metrics of lean, agile and leagile supply chain: An ANP-based approach. European Journal of Operational Research, Vol. 173 No.1, pp. 211–225.

[4]. Aladwani, A. M. (2001), Change management strategies for successful ERP implementation, Business Process Management Journal, Vol. 7 No. 3, pp.266-275.

[5]. Alizai, F. and Burgess, S. (2009), Comparative Analysis of ERP Implementation in Large and Midsize Businesses. Emergent Challenges in IS/IT. Information Institute.

[6]. Al-Mashari, M. (2002), Enterprise resource planning (ERP) systems: a research agenda. Industrial Management & Data Systems, Vol 102 No. 3, pp. 165-170.

[7]. Amoako-Gyampah, K. (2007), Perceived usefulness, user involvement and behavioral intention: an empirical study of ERP implementation. Computers in Human Behavior, Vol. 23 No. 3, pp.1232-1248.

[8]. Anderson, D.J. (2003), Agile management for software engineering: Applying the theory of constraints for business results. Prentice Hall Professional.

[9]. Anderson, D. J. (2010). Kanban – successful evolutionary change for your technology business. Sequim, WA: Blue Hole Press..

[10]. Appleton, E. (1997), How to survive ERP. Datamation, Vol. 43 No.3, pp. 50-53.

[11]. Ara, A. and Al-Mudimigh, A. S. (2011), The Role and Impact of Project Management in ERP project implementation life cycle. Global Journal of Computer Science and Technology, Vol 11 No.5.

[12]. Armand, F.V. (1956), Total Quality Control. Harvard Business Review, Vol 34 No. 6.

[13]. Aversano, L. and Tortorella, M. (2013), Quality evaluation of floss projects: application to ERP systems. Information and Software Technology.

[14]. Aydin, A.O. (2012), A New Way to Determine External Quality of ERP Software. Computational Science and Its Applications – ICCSA 2012.

[15]. Bancroft, N., Seip, H. and Sprengel, A. (1998), Implementing SAP R/3. 2nd ed, Manning Publications, Greenwich.

[16]. Beck, K. and Andres, C. (2004), Extreme programming explained: embrace change. Addison-Wesley Professional.

[17]. Bishnoi V. (2011), Critical Success Factors for Implementing ERP software in Multinational Companies Operating in India, Yojna, Vol 2 No.1, pp. 9- 12.

[18]. Bjørnson, F. O. and Vestues, K. (2016), Knowledge sharing and process improvement in large-scale agile development. In Proceedings of the Scientific Workshop Proceedings of XP2016, pp. 7. ACM.

[19]. Boehm, B. (2001), Get ready for agile methods, with care. Computer Vol. 35 No. 1, pp.64-69.

[20]. Boehm, B. (2011), Towards richer process principles. Proceedings of the 2011 International Conference on Software and Systems Process, ACM.

[21]. Bosch, J. and Bosch-Sijtsema, P.M. (2011), Introducing agile customer-centered development in a legacy software product line. Special issue of Software: Practice and Experience, Vol.41 No. 8, pp. 871-882.

[22]. Botta-Genoulaz, V., Millet, P.A. and Grabot, B. (2005), A survey on the recent research literature on ERP systems. Computers in Industry, Vol.56 No.6, pp. 510-522.

[23]. Boubekri, N. (2001), Technology Enablers for Supply Chain Management. Integrated Manufacturing Systems, Vol. 12 No. 6, pp,394-399.

[24]. Brehm, L., Heinzl, A. and Markus M. (2000), Tailoring ERP systems: A spectrum of choices and their implications, proceeding of 34th Hawaii IEEE International conference on System sciences.

[25]. Bueno, S. and Salmeron, J. (2008), TAM-based success modelling in ERP. Interacting with Computers, Vol. 20, pp. 515–523

[26]. Campbell, D. (2000), Successful ERP implementation Strategies, Integrated Solutions.

[27]. Chandra V. (2015), Comparison between various software development methodologies. International Journal of Computer Applications. Vol.131 No.9, pp.7-10.

[28]. Christopher, M. and Towill, D. (2000), Supply chain migration from lean and functional to agile and customized. Supply Chain Management: An International Journal Vol. 5 No.4, pp. 206–213.

[29]. Chung, S., and Synder, C. (1999), ERP initiation-a historical perspective. Proceedings of America Conference on Information Systems.

[30]. Cissna, T. (1998), ERP Software Implementation brings pain with gains. Electric and Power.

[31]. Cohen, D., Larson G. and Ware B. (2001), Improving software investments through requirements validation. NASA Goddard software engineering workshop, pp. 106-14.

[32]. Cohen, D., Lindvall, M. and Costa, P. (2004), An introduction to agile methods. Advances in Computers Vol 62, pp. 1–66.

[33]. Conboy, K. (2009), Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. Infor. Systems Research Vol. 20 No. 3, pp. 329–354.

[34]. Coplien, J.O. and Bjørnvig, G. (2011), Lean architecture: for agile software development. Wiley.

[35]. Creswell, J. W. (2009), Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.Thousand Oaks, CA: Sage Publications.

[36]. Daneva, M. (2004), ERP Requirements Engineering Practice: Lessons Learned, IEEE Software, pp: 26-33.

[37]. Davenport, T. (2000), Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press.

[38]. Davenport, T.D. (1996), Holistic Management of Mega Package Change: The Case of SAP. Association for Information Systems Annual conference, Phoenix, Arizona.

[39]. Dikert, K., Paasivaara, M. and Lassenius, C. (2016), Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software.

[40]. Dmaithan, A. (2016), Antecedents of ERP systems implementation success: a study on Jordanian healthcare sector. Journal of Enterprise Information Management. Emerald. Vol. 29 No. 4, pp. 549-53.

[41]. Dong, W. (1995), Lean production and industrial engineering applied in China. Computers & Industrial Engineering, Vol. 20 No. 1, pp. 233- 237.

[42]. Dybå, T. and Dingsøyr, T. (2008), Empirical studies of agile software development: A systematic review. Information and software technology. Vol.50 No. 9, pp. 833–859.

[43]. Esteves, J. M. and Pastor, J. A. (1999), An ERP Lifecycle-based Research Agenda. First International Workshop in Enterprise Management Resource and Planning: Methods, Tools and Architecture EMRPS, 359-371.

[44]. Esteves, J., Bohorquez, V. (2007), An updated ERP systems annotated bibliography: 2001-2005. Communications of AIS, pp. 386-446.

[45]. Fagerholm, F., Ikonen, M., Kettunen, P., Münch, P., Roto, V., Abrahamsson, P. (2015), Performance Alignment Work: How software developers experience the continuous adaptation of team performance., Inf. Softw. Technol., Elsevier, Vol. 64, pp. 132-147.

[46]. Fitzgerald, B., Stol, K. (2017), Continuous software engineering: a roadmap and agenda. J Syst Softw, Vol. 123 pp.176–189

[47]. Fogelstrom N, Gorschek T, Svahnberg M and Olsson P. (2010), The impact of agile principles on market-driven software product development. Journal of Software Maintenance and Evolution: Research and Practice, Vol. 22 No. 1, pp.53–80.

[48]. Forger G. (2000), ERP Goes Mid-Market. Modern Materials Handling, Vol. 55, pp. 65-68

[49]. Fryling, M. (2010), Estimating the impact of enterprise resource planning project management decisions on post-implementation maintenance costs: a case study using simulation modelling. Enterprise Information Systems Vol. 4 No. 4, pp. 391–421.

[50]. Goedhard, B. (2016), On the determination of the optimal methodology for ERP projects. Radboud University Press.

[51]. Gefen, D. and Ragowsky, A. (2005), A multi-level approach to measuring the benefits of an ERP System in manufacturing firms. Information Systems Management Journal, Vol. 22 No.1, pp. 18-25.

[52]. Heikkila, M., Paasivaara, C., Lasssenius, D. and Damian. (2017), Managing the requirements flow from strategy to release in largescale agile development: a case study at ericsson. Empirical Software Engineering, pp 1–45.

[53]. Ho, C.W., Raha, S., Gehringer, E. and Williams, L. (2004), Sangam: A distributed pair programming plug-in for Eclipse.

Proceedings of eclipse ‘04 Proceedings of the 2004 OOPSLA workshop on eclipse technology exchange. Pp. 73-77

[54]. Hoch, J. E., and Dulebohn, J. H. (2012), Shared leadership in enterprise resource planning and human resource management system implementation. Human Resource Management review.

[55]. Hoda, R., Norsaremah, S., John G., and Tee, H. (2017), Systematic Literature Reviews in Agile Software Development: A Tertiary Study. Information and Software Technology, Vol 85.

[56]. Holland, C. and Light, B. (1999), Success Factors model for ERP Implementation. IEEE Software, Vol. 16 No. 3, pp. 30-36.

[57]. Hossain, A., Kashem, M.A. and Sultana S. (2014), Enhancing Software Quality Using Agile Techniques, IOSR Journal of Computer Engineering, Vol.10 No.2, pp.87-93.

[58]. Iivari, J. and Iivari, N. (2011), The relationship between organizational culture and the deployment of agile methods. Information and Software Technology, Vol.53 No.5, pp. 509–520.

[59]. Isetta, S. and Sam Pietro, M. (2018), Agile in ERP Projects. PM World Journal, Vol. 8 No. 9.

[60]. Jalote, P. (2000), CMM in Practice: Processes for Executing Software Projects at Infosys. Addison-Wesley Professional,

[61]. Jayawickrama, U., Liu, S. and Hudson Smith, M. (2017), Knowledge prioritization for ERP implementation success. Industrial Management & Data Systems, Vol. 117 No. 7, pp. 1521–1546.

[62]. Jørgensen, M., Teigen, K.H. and Ribu, K. (2004), Better sure than safe? Over-confidence in judgement based software development effort prediction intervals. Journal of Systems and Software, Vol. 70 No. 1, pp. 79-93.

[63]. Julia, H. (2007), Bringing ‘Lean’ Principles to Service Industries, HBS Working Knowledge.

[64]. Juntao, G. and Zhang, L. (2008), Detecting Gaps between ERP Software and Organizational Needs: A Semantic Similarity

Based Approach. Proceedings of IEEE International Workshop on Semantic Computing and Systems, pp. 21 – 26.

[65]. Kennerley, M., and Neely, A. (2001), Enterprise resource planning: analyzing the impact. Integrated Manufacturing Systems, Vol.12 No.2, pp. 103–113.

[66]. Klaus, H., Rosemann, M. and Gable, G. (2000), What is ERP? Information Systems Frontiers, Vol, 2 No 2, pp.141-162.

[67]. Koch, C. (2003), The ABC of ERP. Enterprise Resource Planning Research Centre. CIO.

[68]. Kraljic A. and Kraljic, T. (2017), Agile Software Engineering Practices and ERP: Is a sprint too fast for ERP Implementation? Joint Proceedings of the BIR 2017 pre-BIR Forum, Workshops and Doctoral Consortium, Copenhagen, Denmark.

[69]. Kraljic, A., Kraljic, T., Poels, G. and Devos, J. (2014), ERP Implementation Methodologies and Frameworks: A Literature Review. Proceedings of the 8th European Conference on IS Management and Evaluation ECIME2014, University of Ghent, Belgium, pp 309-316.

[70]. Krishnamurthy, R., Yauch, C. (2007), Leagile manufacturing: A proposed corporate infrastructure. International Journal of Operations and Production Management.

[71]. Laanti, M. (2012), Agile methods in large-scale software development organisations. Applicability and model for adoption. Doctoral thesis. University of Oulu.

[72]. Larman, C. and Vodde, B. (2008), Scaling lean & agile development: Thinking and organisational tools for large-scale Scrum. Addison-Wesley Professional.

[73]. Larman, C. and Vodde, B. (2017), Large-Scale Scrum: More with LeSS: Addison-Wesley professional.

[74]. Leffingwell, D., Yakyama, A., Knaster, R., Jemilo, D., and Oren, I. (2017), SAFe reference guide: Scaled Agile Framework for Lean Software and Systems Engineering: Addison Wesley.

[75]. Lei, H., Ganjeizadeh, F., Jayachandran, P. K., and Ozcan, P. (2017), A statistical analysis of the effects of Scrum and Kanban on software development projects. Robotics and Computer-Integrated Manufacturing, Vol. 43, pp. 59-67

[76]. Liker J. (2004), The Toyota way. Esensi.

[77]. Livermore, J. (2011), Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. Journal of Software, Vol. 3.

[78]. Luo, W. and Strong. D. (2004), A framework for evaluating ERP implementation choices. IEEE Transactions on Engineering Management , Vol. 51 No. 3, pp. 322-333.

[79]. Lycett, M., Macredie, R., Patel, C. and Paul, R. (2003), Migrating Agile methods to standardized development practice. IEEE Computer Magazine.

[80]. Maglyas, A., Nikula, U. and Smolanderm K. (2012), Lean Solutions to Software Product Management Problems. IEEE Software, Vol. 29 No. 1, pp. 40-46.

[81]. Mandic, V., Oivo, M., Rodriguez, P., Kuvaja, P., Kaikkonen, H. and Turhan, B. (2010), What is flowing in Lean Software Development? Proceedings of the 1st International Conference on Lean Enterprise Software and Systems (LESS), pp.72–84.

[82]. Marchwinski C. and Shook J. (2008), Lean lexicon: a graphical glossary for lean thinkers. 4th edition, Lean Enterprise Institute.

[83]. Markus, M. L. and Tanis, C. (2000), The Enterprise Systems Experience: From Adoption to Success. In Framing the Domains of IT Management: Projecting the Future Through the Past. Zmud, R.W. edition, 173 – 207.

[84]. Marnewick, C. and Labuschagne, L. (2005), A conceptual model for Enterprise Resource Planning (ERP). Information Management & Computer Security, Vol. 13 No.2, pp. 144–155.

[85]. Martin, M. (1998), Smart Managing, Fortune Magazine, Vol. 2, pp. 149-151.

[86]. McBreen, P. (2012), Software craftsmanship: the new imperative. Addison-Wesley Professional.

[87]. Middleton, P. and Sutton, J. (2005), Lean software strategies: proven techniques for managers and developers. Productivity Press.

[88]. Mishra, A. and Mishra, D. (2011), A curriculum for agile software development methodologies. Software Engineering Notes, Vol.36 No. 3, pp. 1-2.

[89]. Mishra, D. and Mishra, A. (2011), Complex software project development: agile methods adoption. Journal of Software Maintenance and Evolution: Research and Practice.

[90]. Moe, N. B., Olsson, H. and Dingsøyr, T. (2016). Trends in large-scale agile development: A summary of the 4th workshop at xp2016. In Proceedings of the Scientific Workshop Proceedings of XP2016, pp. 1. ACM.

[91]. Morgan, J.M. and Liker, J.K. (2006), The Toyota product development system. Productivity press.

[92]. Nagel, R. and Dove, R. (1991), 21st century manufacturing enterprise strategy: An industry led view. DIANE Publishing.

[93]. Nagpal, S., Khatri, S. and Kumar, S. (2015), Comparative Study of ERP Implementation Strategies. Systems, Application and Technology Conference's – 2015. IEEE Xplore.

[94]. Nah, F., Lau, J. and Kuang, J. (2001), Critical factors for successful implementation of enterprise systems. Business Process Management Journal, Vol. 7 No. 3, pp.285-296.

[95]. Nandyal, R. (2011), CMMI: A Framework of Constellations for Building World-class IT organizations, 2nd Edition. Tata Mc-Graw Hill 2011

[96]. Narang, R. (2008), Some issues to consider in lean production. First International Conference on Emerging Trends in Engineering and Technology, pp.749–753.

[97]. Naylor, J., Naim, M. and Berry, D. (1999), Leagility: integrating the lean and agile manufacturing paradigms in the total supply

chain. International Journal of production economics, Vol. 62 No. 1, pp.107-118.

[98]. Noor, R. and Khan, M.F. (2014), Defect Management in Agile Software Development. Int. Jour. Modern Education and Computer Science, Vol.3, pp.55-60.

[99]. Ohno, T. (1988). Toyota production system: beyond large-scale production. Productivity press.

[100]. Oracle White Paper. (2014), Oracle Unified Method (OUM) Oracle's Full Lifecycle Method for Deploying Oracle-Based Business Solutions. Oracle Press.

[101]. Palmer, S. and Felsing, M. (2001), A practical guide to feature-driven development. Pearson Education.

[102]. Parr, A. and Shanks, G. (2000), A model of ERP project implementation. Journal of Information Technology. Vol. 15 No. 4, pp. 289–304.

[103]. Parthasarathy, S. and Daneva, M. (2014), Customer requirements based ERP customization using AHP technique. Business Process Management Journal, Vol. 20 No. 5, pp.-730-751.

[104]. Peter, K. and Lanza, G. (2011), Company-specific quantitative evaluation of lean production methods. Prod Eng. WGP 2011, Vol. 5, pp.81-87.

[105]. Pikkarainen, M., Haikara, J., Salo, O. and Abrahamsson, P., Still. J. (2008) The impact of agile practices on communication in software development. Empirical Software Engineering Vol. 13 No. 3, pp.303–337.

[106]. Poppendieck, M. and Cusumano, M. (2012), Lean Software Development: A Tutorial. IEEE Software, Vol. 29 No. 5, pp. 26–32.

[107]. Poppendieck, M. and Poppendieck. T. (2003), Lean software development: An agile toolkit. Addison-Wesley Professional.

[108]. Poppendieck, M. and Poppendieck. T. (2006), Implementing lean software development: From concept to cash. Addison-Wesley Professional.

[109]. Poppendieck, M. and Poppendieck. T. (2009), Leading lean software development: Results are not the point. Pearson Education.
[110]. Poppendieck, M. and Poppendieck. T. (2010), Leading lean software development: Results are not the point. Upper Saddle River, NJ: Addison-Wesley.
[111]. Ragowsky, A. and Gefen, D. (2008), What Makes the Competitive Contribution of ERP Strategic. The DATA BASE for Advances in Information Systems, Vol. 39 No. 2, pp. 33-49.
[112]. Rajakaruna, R.J.P.K. and Wijayanayake, J. (2018), An approach to coexistence analysis between agility and ERP implementation. International Research Conference on Smart Computing and Systems Engineering - SCSE 2018, Department of Industrial Management, Faculty of Science, University of Kelaniya, Sri Lanka. p.221.
[113]. Raman, S. (1998), Lean Software Development: Is It Feasible? in 17th Digital Avionics Systems Conference., IEEE: New York., pp.13-18.
[114]. Rao, S. S. (2000). Enterprise resource planning: business needs and technologies. Industrial Management & Data Systems, Vol. 100 No. 2, pp. 81–88.
[115]. Reinertsen, D.G. (2009), The principles of product development flow: second generation lean product development. Celeritas Redondo Beach, Canada.
[116]. Rooney, C. and Bangert, C., (2000). Is an ERP System Right for You? Adhesives Age, Vol. 43 No. 9, pp. 30-33.
[117]. Rosa, W., Packard, T., Krupanand, A., Bilbro, J. W. and Hodal, M. M. (2012), COTS integration and estimation for ERP. Journal of Systems and Software, Vol. 86, pp.538– 550
[118]. Ross, J. W. and Vitale, M.R. (2000), The ERP revolution: surviving vs. thriving. Information Systems Frontiers, Vol. 2 No. 2, pp. 233–241.

[119]. Ross, J. W., Weill, P. and Robertson, D. (2006), Enterprise Architecture as Strategy. Harvard Business School Press, Boston, MA.

[120]. Ruhi, U. (2016), An experiential learning pedagogical framework for enterprise systems education in business schools. The International Journal of Management Education, Vol. 14 No. 2, pp.198–211

[121]. Sandoe, K. Corbitt, G. and Boykin, R. (2001), Enterprise Integration. Wiley: NY.

[122]. Schwaber, K. and Beedle, M. (2002), Agile software development with Scrum. Prentice Hall.

[123]. Shah R. and Ward P. (2007), Defining and developing measures of lean production. Journal of operations Management, Vol. 25 No. 4, pp. 785–805.

[124]. Sharifi, H. and Zhang, Z. (1999), A methodology for achieving agility in manufacturing organizations: An introduction. International Journal of Production Economics Vol. 62 No. 1, pp.7–22.

[125]. Shehab, E.M. and Sharp, M.W., Supramaniam, L., Spedding, T.A. (2004), Enterprise resource planning: An integrative review. Business Process Management Journal, Vol. 10, No. 4, pp. 359-386.

[126]. Sherehiy, B. and Karwowski. W., Layer, J.K. (2007), A review of enterprise agility: concepts, frameworks, and attributes. International Journal of Industrial Ergonomics, Vol. 37 No. 5, pp. 445–460.

[127]. Silva, L. and Fulk, H. K. (2012), From disruptions to struggles: Theorizing power in ERP implementation projects. Information and Organization, Vol. 22 No. 4, pp. 227-251.

[128]. Somers, T., Nelson, K. and Ragowsky, A. (2004), Enterprise Resource Planning ERP for the Next Millennium: Development of an Integrative Framework and Implications for

Research. In: Proceedings of Americas Conference on Information Systems AMCIS.

[129]. Sommerville, I. (2011), Software Engineering. Pearson Education Ltd.

[130]. Staats, B.R., Brunner, D.J. and Upton, D.M. (2011), Lean principles, learning, and knowledge work: Evidence from a software services provider. Journal of Operations Management, Vol. 29 No. 5, pp. 376–390.

[131]. Stapleton, J. (2003), DSDM: Business focused development. Addison-Wesley Professional.

[132]. Sumner, M. (2006), Enterprise Resource Planning. Pearson Education, New Delhi,

[133]. Sutton, J.M., (1996), Lean Software for the Lean Aircraft. 15th Digital Avionics Systems Conference.1996, IEEE: New York. Pp. 49-54.

[134]. Somers, T. and Nelson, K. (2001), The Impact of Critical Success Factors across the Stages of Enterprise Resource Planning Implementations. Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS-34), Vol. 8, pp.8.-16.

[135]. The Standish Group, (2012), The Standish Group report: Chaos, The Standish Group, Boston, Massachusetts.

[136]. Tipaldi, M., Götz, C., Ferraguto, M., Troiano, L. and Bruenjes, B. (2013), The robust software feedback model: an effective waterfall model tailoring for space SW. DASIAdata systems in aerospace conference 2013.

[137]. Tolfo, C. and Wazlawick, R.S. (2008), The influence of organizational culture on the adoption of extreme programming. Journal of Systems and Software, Vol. 81 No. 11, pp.1955–1967.

[138]. Tripp, J., Saltz, J. and Turk, D. (2018), Introduction to Agile and Lean: Organizations, Products, and Development Minitrack.

Proceedings of the 51st Hawaii International Conference on System Sciences

[139]. Turban, E., Leidner, D., Mclean, E. and Wetherbe, J. (2006), Information Technology Management: Transforming Organisations in the Digital Economy. New York: John Wiley & Sons.

[140]. Turk, D., France, R. and Rumpe, B. (2002), Limitations of agile software processes. Third International Conference on eXtreme Programming and Agile Processes in Software Engineering (XP 2002), pp. 43–46.

[141]. Van Stijn, E. and Wensley, A. (2001), Organizational memory and the completeness of process modeling in ERP systems: some concerns, methods and directions for future research. Emerald Business Process Management Journal, Vol. 7(3). 181–194.

[142]. Vijay, D. and Gopinath, G. (2014), Guidelines to minimize the cost of software quality in Agile scrum process, International Journal of Software Engineering & Applications, Vol.5, No.3.

[143]. Vilkki, K. (2010), When agile is not enough. In: Lean Enterprise Software and Systems. Springer, pp. 44–47.

[144]. Vilpola, I., Mattila, K. and Salmimaa T. (2006), Applying Contextual Design to ERP System Implementation", CHI, Experience Report, Montreal, Quebec, Canada, ACM, pp.147-152.

[145]. Vinodh, S. and Aravindraj, S. (2013), Evaluation of leagility in supply chains using fuzzy logic approach. International Journal of Production Research, Vol. 51 No. 4, pp. 1186-1195 Taylor Francis Online.

[146]. Wang, X. and Conboy, K. (2011), Comparing Apples with Oranges? Perspectives of a Lean Online Community on the Differences between Agile and Lean. Thirty Second International Conference on Information Systems, Shanghai.

[147]. Wang, X., Conboy, K. and Cawley, O. (2012), 'Leagile' software development: An experience report analysis of the application

of lean approaches in agile software development. Journal of Systems Software, Vol. 85 No 6, pp.1287–1299.

[148]. Womack, J.P. and Jones, D.T. (1996), Lean thinking: Banish waste and create wealth in your organisation. New York, Rawson Associates.

[149]. Womack, J.P. and Jones, D.T. (2015), The machine that changed the world. London: Simon and Schuster.

[150]. Womack, J.P., Jones, D.T. and Roos, D. (1990), The Machine That Changed the World: The Story of Lean Production: How Japan's Secret Weapon in the Global Auto Wars Will Revolutionize Western Industry. New York, Rawson Associates.

[151]. Wortmann, J.C. (1998), Evolution of ERP systems. Strategic Management of the Manufacturing Value Chain, Kluwer Academic, Boston, MA. pp 11-23.

[152]. Bista Solutions. (2013). 10 reasons for ERP Implementation failures. Available at https://www.bistasolutions.com/resources/blogs/erp-implementation-failures/ accessed on 18 Oct 2018.

[153]. Clark, L. (2015), Can OUM Support Agile Development? Oracle Blogs available at https://blogs.oracle.com/oum/can-oum-support-agile-development accessed on 13 Oct 2018.

[154]. Coe J. (2017). Operating Agile with an ERP Implementation. Available at https://www.projectmanagement.com/contentPages/article.cfm?forcemobile=on&ID=379457&thisPageURL=/articles/379457/Operating-Agile-with-an-ERP-Implementation accessed on 18 Oct 2018.

[155]. Doig, C. (2015), How Agile techniques can improve enterprise software implementation. Available at https://www.cio.com/article/2978861/enterprise-resource-planning/how-agile-techniques-can-improve-enterprise-software-implementation.html accessed on 6 Dec 2018.

[156]. Finance Online (2017). Why ERP Implementation Is a Challenging Task? 9 Common Mistakes Companies Make. Available at https://financesonline.com/why-erp-implementation-is-a-challenging-task-9-common-mistakes-companies-make accessed on 13 Oct 2018.

[157]. Gartner. (2018). Gartner Says Global IT Spending to Grow 6.2 Percent in 2018. Available at https://www.gartner.com/newsroom/id/3871063 accessed on 5 Oct 2018.

[158]. Genesis Consulting (2011). Fact or Fiction: ERP Projects Can Be Delivered Using Agile. Available at https://pdfs.semanticscholar.org/presentation/c89c/363f181263c9bee4de87a4ea96e48e7ca7ce.pdf accessed on 18 Oct 2018

[159]. Musil, J. (2017). ASAP 8 Agile – Deep Dive. SAP Webinar, available at https://pdfs.semanticscholar.org/presentation/1e3d/6febb0fe95dddebc85333a6dffc01ed8522a.pdf accessed on 13 Oct 2018.

[160]. Panaroma Consulting ERP Report 2017 (2017) available at http://go.panorama-consulting.com/rs/603-UJX-107/images/2017-ERP-Report.pdf accessed on 16 Nov 2018.

[161]. Panaya (2017). Tired of Defects Inflation? Available at https://www.panaya.com/blog/testing/system-wide-defect-management/ accessed on 24 Nov 2018.

[162]. Scott Amblers + Asscoiates (2013), IT Project Success Rates Survey Results. Available at Http://www.ambysoft.com/surveys/success2013.html . accessed on Jan 15, 2015.

## APPENDIX 1

## QUESTIONAIRE FOR ISSUES IN ERP IMPLEMENTATION

Name:____________________Designation______________________________

Organization:______________________________________________________

| Question | Yes | No |
|---|---|---|
| Do you feel that ERP Big-bang implementations push the test cycle at end that leads to quality problems? | | |
| Do you feel that Changes owing to Non-Functional Requirements impact quality of implementations? | | |
| Do you feel that ERP fails because of communication issues - Issues because of middlemen between developers and end users | | |
| Do you feel that ERP fails because of defect propagation into next phase? | | |
| Do you think that developed system does not reflect the current requirement of ERP system? | | |
| Do you feel that Dropped requirements during implementations become business critical most of times? | | |
| Do you think that most of issues in customization happen because of gold plating - due to unnecessary value-adds provided to customer? | | |
| Do you think that your time line of implementation was missed because of huge documentation | | |
| Do you think that you cannot include new requirement in implementation because of high effort / Cost of Change? | | |
| Do you think that your ERP imp project failed because Individual Competency issue of team member/s? | | |
| Do you think that your ERP Implementation failed because of Integration issues with Third Party systems? | | |
| Do you think that your ERP Implementation failed because of Integration issues within ERP? | | |
| Do you think that team in ERP lacks knowledge sharing / transition? | | |
| Do you squeeze later phases of implementation to make out delays of earlier phases? | | |
| Do you miss the Customer feedback mechanism in current ERP implementation model that you use? | | |
| Do you think that ERP implementations are delayed because of multiple stakeholders and their influences? | | |
| Do you think that your ERP over processes the data such as extra validations, checks? | | |

| Question | Yes | No |
| --- | --- | --- |
| Do you feel that most of the defects in all the modules are repeating defects? | | |
| Do you think that requirement volatility is biggest enemy of ERP Implementation | | |
| Do you think in mostly you are waiting for approval and green signal from past phase to start new phase of implementation? | | |

# APPENDIX 2

## 1.INTRODUCTION

### 1.1 GENERAL INFORMATION

#### 1.1.1 BASIC BUSINESS NEED

Client uses Oracle E-Business Suite Version R.12.2.4 as ERP for their business transactions and reporting for various functions. On the other hand, new entity uses OFA ERP 6.0 EHP 12 and some legacy applications for similar needs. This request for proposal is for inviting proposals for implementation and migration of new entity Operations into existing Oracle ERP system of client and migration of existing client Legal Entities into new entity upon completion of merger. This would include Solution, Design & Development, migrations, conversions, integrations, testing, implementation and post production maintenance of comprehensive, integrated and centralized solution for client and new entity.

#### 1.1.2 PROJECT OBJECTIVES

The objective of the project to migrate all Client's entities (including new entity) to current Oracle Application instance of client.

- Migration of new entity – OFA to Oracle Global instance
- Process harmonization and improved automation by leveraging best practices
- Merger of legal Entities (One company code in Oracle)

#### 1.1.3 PROJECT BENEFITS

Client envisages following key benefits from this project:

- Same ERP platform for operations of Merged entity
- Ensuring Customer delight by brining Operational Excellence
- Bringing operational alignment across circles
- Improving Co-ordination between Circles and Corporate
- Aligning Internal & External Processes

- Business process Improvement

**1.2 HIGH LEVEL SCOPE**

The Implementation partner is expected to own following activities as part of this project:

- New entity and client Business Process harmonization by leveraging their best practices. If there are some gaps in the processes which cannot be mapped to existing system solution, then implementation partners would provide the system solution which would be applicable for client.
- Solution designing to meet the business requirements and gaps based on process workshops to be conducted as part of the project.
- Design, Development and Testing of all technical components required for mapping business process requirements, Integration between Oracle and legacy systems, data migration for Masters / open transactions, Reporting requirements for the existing & changed / new processes and Business Process Improvements as part of this project.
- Integrated Testing i.e. System Integration Testing and support for User Acceptance Testing.
- Deployment of Oracle Applications in IT framework.
- Training and Change Management for core team & end users to support and manage the overall project.
- Merger of existing client's legal entities in Oracle with newly created entity post legal approval of merger.
- Post Production Support and Maintenance for the period mentioned in the RFP
- Recommendation for suitable hardware sizing for proposed solution landscape in IT framework.
- Achievement of Performance benchmark for ERP performance as per RFP scope.
- Recommending appropriate changes to the existing IT infrastructure if required for the performance enhancement of application.

- Provide all relevant services to successfully implement and support the proposed system at all organizational entities.

Detailed functional, technical, integration, data migration and reporting requirements are defined in subsequent sections.

## 1.3 EXISTING ERP FOOTPRINT

### 1.3.1 OVERVIEW OF CURRENT ERP APPLICATIONS

**Oracle R12 Application Overview**

Client is using Oracle e-Business Suit R12.2.4 for addressing various business requirements in Financials, Supply Chain Management, Projects, Billing and HR. Following is high level summary of the same.

- Financial Management
    - General Ledger
    - Accounts Payables
    - Cash Management
    - Accounts Receivables
    - Fixed Assets
- Project Management
    - Project Costing
- Procurement and Inventory
    - Purchasing
    - Inventory Management
- Employees Expense Management
    - expense (including custom Travel Request module)

New Entity is using OFA 6.0 for addressing various business requirements in Financials, Supply Chain Management, Projects, Billing and HR. Following is high level summary of the same:

- Finance
- General Ledger

- Fund Management
- CO (Controlling)
- Accounts Payables
- Asset Accounting
- Accounts Receivables
- Project & Asset Management
- Projects
- Fixed Assets
- Procurement and Inventory
- Material Management
- Sales & Distribution
- SD

**Oracle EBS:**

- Application Version: Oracle e-Business Suite R12.2.4
- Platform and Version: Red hat RHEL 6.7
- Data Base Version: Oracle 11g 11.2.0.4

**ERP:**

- Application Version: EHP6 FOR OFA ERP 6.0
- Platform and Version: Linux GNU SLES-9 x86_64 cc4.1.2
- Data Base Version: Oracle 11g 11.2.0.3.0

**Data Mapping and Extraction from OFA**: It will be responsibility of Vendor to prepare Data Conversion templates after mapping of source data with target database design. Data will be extracted by vendor from OFA in these Data conversion templates for all data elements.

**Data Reconciliation & Cleansing**: Extracted data will be shared with Business Owners and other designated members of user community. Business community will validate the extracted data by running relevant reports in OFA. If required they will cleanse the data and send back to project team for uploading in Oracle.

**Data Upload**: Once data is received from Business owners in the specified templates, Project team will upload the data into Oracle. Data uploading can be done by using Manual upload, Data Loader or Automated conversion scripts as specified in the Data Migration / Conversion approach.

**Data Validation and Testing**: Once data is uploaded in Oracle, project team would provide the uploaded data extracts with the help of database queries or reports available in system to ensure that the conversion process has processed all records present in the file. Business team and project team should validate the data files for all data elements. They should validate the data files for number of records, overall file totals, and totals by grouping of key data elements. Reconciliation reports should be generated as part of the data validation. Project team, who is primarily responsible for importing the data into the Oracle instance would have appropriate reconciliation mechanisms in place to ensure that data has been correctly loaded in the Oracle Instance. Some of the reconciliation procedure may include:

- Validating the number of records imported into Oracle based on data files sent by business
- Validating the overall totals of the data elements that were imported based on data files
- Build reports that were documented in the To-Be process reports to help with the data reconciliation efforts
- Business owners will be responsible for validating the data that is uploaded in the system.
- Reconciling and validating the data after it is loaded on to various instance as per data conversion approach.

The Implementation Partner shall after development and customization/configuration of the ERP solution, conduct tests to demonstrate that the system meets all the requirements (functional and technical) specifications as brought out in this DOCUMENT and would be in accordance with the procedures detailed in approved To-Be process document. Based on these tests, a report would be submitted by the Implementation Partner for review. The test results and response times should be demonstrated by the Implementation Partner during the testing. The Testing approach should include below testing phases during the project lifecycle.

**Unit Testing**

Unit testing shall be carried out by the Implementation Partner to ensure unit functionality meets the designed requirements and baseline configuration is valid.

**Functional and Process Testing**

Implementation Partner shall document the functional and process test cases and the test cases shall be tested with business process procedures.

**Integration Testing**

Integration testing shall be carried out by the implementation partner to ensure cross function modules are integrated and transaction data is flowing across the modules accurately. Implementation Partner is required to plan and execute integration and ensure satisfactory results are obtained.

**SYSTEM ACCEPTANCE**

Implementation Partner shall document the user acceptance test cases and test cases shall be tested with the designed business process procedures. The purpose of this acceptance is to ensure conformance to the required process operations response time, the integrity of the software after installation, and to eliminate any operational bugs. This will include Fine tuning of the software, ensuring all required related component software are installed and any debugging required. The acceptance tests will be carried out before go-live at site.

Users shall be testing the test cases and would be tested with the designed business process procedures developed by Implementation Partner. At the satisfactory conclusion of these Acceptance Tests to the satisfaction of users, the commissioning/ implementation of the project shall be complete for the respective location.

# APPENDIX 3

## Weekly effort for Projects

Project ID: TP1

| | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 320 | | | | | | | |
| W2 | 320 | | | | | | | |
| W3 | 320 | | | | | | | |
| W4 | 92 | 20 | | | | | | |
| W5 | | 120 | | | | | | |
| W6 | | 120 | | | | | | |
| W7 | | | 320 | | | | | |
| W8 | | | 320 | | | | | |
| W9 | | | 320 | | | | | |
| W10 | | | 68 | 32 | | | | |
| W11 | | | | 120 | | | | |
| W12 | | | | 120 | | | | |
| W13 | | | | | 400 | | | |
| W14 | | | | | 400 | | | |
| W15 | | | | | 400 | | | |
| W16 | | | | | 400 | | | |
| W17 | | | | | 400 | | | |
| W18 | | | | | 380 | | | |
| W19 | | | | | | 80 | | |
| W20 | | | | | | 80 | | |
| W21 | | | | | | 80 | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | 80 | | |
| W24 | | | | | | 80 | | |
| W25 | | | | | | 52 | | |
| W26 | | | | | | | 160 | |
| W27 | | | | | | | 160 | |
| W28 | | | | | | | 152 | |
| W29 | | | | | | | | 160 |
| W30 | | | | | | | | 38 |

Project ID: TP2

| Week | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 320 | | | | | | | |
| W2 | 320 | | | | | | | |
| W3 | 320 | | | | | | | |
| W4 | 92 | 16 | | | | | | |
| W5 | | 120 | | | | | | |
| W6 | | 120 | | | | | | |
| W7 | | | 320 | | | | | |
| W8 | | | 320 | | | | | |
| W9 | | | 320 | | | | | |
| W10 | | | 68 | 16 | | | | |
| W11 | | | | 120 | | | | |
| W12 | | | | 120 | | | | |
| W13 | | | | | 400 | | | |
| W14 | | | | | 400 | | | |
| W15 | | | | | 400 | | | |
| W16 | | | | | 400 | | | |
| W17 | | | | | 400 | | | |
| W18 | | | | | 369 | | | |
| W19 | | | | | | | | |
| W20 | | | | | | 80 | | |
| W21 | | | | | | 80 | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | 80 | | |
| W24 | | | | | | 80 | | |
| W25 | | | | | | 80 | | |
| W26 | | | | | | 80 | | |
| W27 | | | | | | 24 | 160 | |
| W28 | | | | | | | 160 | |
| W29 | | | | | | | 166 | |
| W30 | | | | | | | | 164 |

Project ID: TP3

| Week | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 320 | | | | | | | |
| W2 | 320 | | | | | | | |
| W3 | 320 | | | | | | | |
| W4 | 66 | 32 | | | | | | |
| W5 | | 120 | | | | | | |
| W6 | | 120 | | | | | | |
| W7 | | | 320 | | | | | |
| W8 | | | 320 | | | | | |
| W9 | | | 320 | | | | | |
| W10 | | | 78 | | | | | |
| W11 | | | | 120 | | | | |
| W12 | | | | 120 | | | | |
| W13 | | | | 48 | 400 | | | |
| W14 | | | | | 400 | | | |
| W15 | | | | | 400 | | | |
| W16 | | | | | 400 | | | |
| W17 | | | | | 400 | | | |
| W18 | | | | | 400 | | | |
| W19 | | | | | 92 | 52 | | |
| W20 | | | | | | 80 | | |
| W21 | | | | | | 80 | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | 80 | | |
| W24 | | | | | | 80 | | |
| W25 | | | | | | 80 | | |
| W26 | | | | | | | 160 | |
| W27 | | | | | | | 160 | |
| W28 | | | | | | | 160 | |
| W29 | | | | | | | 16 | 24 |
| W30 | | | | | | | | 160 |

Project ID: TP4

| Week | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 320 | | | | | | | |
| W2 | 320 | | | | | | | |
| W3 | 320 | | | | | | | |
| W4 | 112 | 48 | | | | | | |
| W5 | | 120 | | | | | | |
| W6 | | 120 | | | | | | |
| W7 | | | 320 | | | | | |
| W8 | | | 320 | | | | | |
| W9 | | | 320 | | | | | |
| W10 | | | 92 | 16 | | | | |
| W11 | | | | 120 | | | | |
| W12 | | | | 120 | | | | |
| W13 | | | | | 400 | | | |
| W14 | | | | | 400 | | | |
| W15 | | | | | 400 | | | |
| W16 | | | | | 400 | | | |
| W17 | | | | | 400 | | | |
| W18 | | | | | 400 | | | |
| W19 | | | | | 64 | 48 | | |
| W20 | | | | | | 80 | | |
| W21 | | | | | | 80 | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | 80 | | |
| W24 | | | | | | 80 | | |
| W25 | | | | | | 80 | | |
| W26 | | | | | | 80 | | |
| W27 | | | | | | | 160 | |
| W28 | | | | | | | 160 | |
| W29 | | | | | | | 160 | |
| W30 | | | | | | | 16 | 24 |

Project ID: TP5

| | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 320 | | | | | | | |
| W2 | 320 | | | | | | | |
| W3 | 320 | | | | | | | |
| W4 | 94 | 48 | | | | | | |
| W5 | | 120 | | | | | | |
| W6 | | 120 | | | | | | |
| W7 | | | 320 | | | | | |
| W8 | | | 320 | | | | | |
| W9 | | | 320 | | | | | |
| W10 | | | 84 | 16 | | | | |
| W11 | | | | 120 | | | | |
| W12 | | | | 120 | | | | |
| W13 | | | | | 400 | | | |
| W14 | | | | | 400 | | | |
| W15 | | | | | 400 | | | |
| W16 | | | | | 400 | | | |
| W17 | | | | | 400 | | | |
| W18 | | | | | 404 | | | |
| W19 | | | | | | 80 | | |
| W20 | | | | | | 80 | | |
| W21 | | | | | | 80 | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | 80 | | |
| W24 | | | | | | 80 | | |
| W25 | | | | | | 80 | | |
| W26 | | | | | | 88 | | |
| W27 | | | | | | | 160 | |
| W28 | | | | | | | 160 | |
| W29 | | | | | | | 160 | |
| W30 | | | | | | | 12 | 144 |
| W31 | | | | | | | | 48 |

Project ID: GP1

| | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 160 | | | | | | | |
| W2 | 66 | 44 | 74 | | | | | |
| W3 | | | 160 | | | | | |
| W4 | | | | 38 | | | | |
| W5 | | | | | 240 | | | |
| W6 | | | | | 239 | | | |
| W7 | | | | | | 80 | | |
| W8 | | | | | | 19 | 80 | |
| W9 | 66 | | | | | | 22 | 34 |
| W10 | 156 | 40 | 80 | | | | | |
| W11 | | | 160 | | | | | |
| W12 | | | | 38 | 144 | | | |
| W13 | | | | | 240 | | | |
| W14 | | | | | 88 | 24 | | |
| W15 | | | | | | 80 | | |
| W16 | | | | | | | 80 | |
| W17 | 136 | | | | | | 24 | 32 |
| W18 | 96 | 40 | 80 | | | | | |
| W19 | | | 160 | 32 | 152 | | | |
| W20 | | | | | 232 | | | |
| W21 | | | | | 94 | 16 | | |
| W22 | | | | | | 80 | 8 | |
| W23 | | | | | | | 72 | |
| W24 | 128 | | | | | | 16 | 30 |
| W25 | 96 | 52 | 68 | | | | | |
| W26 | | | 152 | | | | | |
| W27 | | | | 44 | 201 | | | |
| W28 | | | | | 240 | | | |
| W29 | | | | | 46 | 97 | 16 | |
| W30 | | | | | | | 90 | 40 |

Project ID: GP2

| | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 160 | | | | | | | |
| W2 | 64 | 43 | 72 | | | | | |
| W3 | | | 160 | | | | | |
| W4 | | | | 40 | | | | |
| W5 | | | | | 244 | | | |
| W6 | | | | | 240 | | | |
| W7 | | | | | | 80 | | |
| W8 | | | | | | 16 | 80 | |
| W9 | 63 | | | | | | 42 | 40 |
| W10 | 160 | 44 | 73 | | | | | |
| W11 | | | 160 | | | | | |
| W12 | | | | 36 | 152 | | | |
| W13 | | | | | 240 | | | |
| W14 | | | | | 88 | 16 | | |
| W15 | | | | | | 80 | | |
| W16 | | | | | | | 80 | |
| W17 | 128 | | | | | | 44 | 44 |
| W18 | 100 | 48 | 96 | | | | | |
| W19 | | | 136 | 44 | | | | |
| W20 | | | | | 240 | | | |
| W21 | | | | | 244 | | | |
| W22 | | | | | | 104 | | |
| W23 | | | | | | | 80 | |
| W24 | 120 | | | | | | 40 | 48 |
| W25 | 99 | 37 | | | | | | |
| W26 | | | 160 | | | | | |
| W27 | | | 73 | 36 | | | | |
| W28 | | | | | 240 | | | |
| W29 | | | | | 240 | 88 | | |
| W30 | | | | | | | 80 | |
| W31 | | | | | | | 40 | 32 |

Project ID: GP3

| | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 160 | | | | | | | |
| W2 | 64 | 40 | 80 | | | | | |
| W3 | | | 160 | | | | | |
| W4 | | | | 40 | | | | |
| W5 | | | | | 244 | | | |
| W6 | | | | | 240 | | | |
| W7 | | | | | | 80 | | |
| W8 | | | | | | 32 | 80 | |
| W9 | 64 | | | | | | 32 | 40 |
| W10 | 160 | 40 | 80 | | | | | |
| W11 | | | 160 | | | | | |
| W12 | | | | 32 | 168 | | | |
| W13 | | | | | 240 | | | |
| W14 | | | | | 118 | 24 | | |
| W15 | | | | | | 80 | | |
| W16 | | | | | | | 80 | |
| W17 | 142 | | | | | | 24 | 40 |
| W18 | 96 | 48 | 96 | | | | | |
| W19 | | | 136 | 40 | | | | |
| W20 | | | | | 240 | | | |
| W21 | | | | | 248 | | | |
| W22 | | | | | | 102 | | |
| W23 | | | | | | | 80 | |
| W24 | 128 | | | | | | 24 | 40 |
| W25 | 96 | 44 | | | | | | |
| W26 | | | 160 | | | | | |
| W27 | | | 73 | 36 | | | | |
| W28 | | | | | 240 | | | |
| W29 | | | | | 240 | 84 | | |
| W30 | | | | | | | 80 | |
| W31 | | | | | | | 16 | 24 |

Project ID: GP4

| | **PRS** | **PRSV** | **PDD** | **PDDV** | **PCD** | **PCDV** | **PSIT** | **PUAT** |
|---|---|---|---|---|---|---|---|---|
| W1 | 160 | | | | | | | |
| W2 | 64 | 40 | 72 | | | | | |
| W3 | | | 160 | | | | | |
| W4 | | | | 40 | | | | |
| W5 | | | | | 232 | | | |
| W6 | | | | | 240 | | | |
| W7 | | | | | | 80 | | |
| W8 | | | | | | 32 | 80 | |
| W9 | 64 | | | | | | 16 | 40 |
| W10 | 160 | 44 | 80 | | | | | |
| W11 | | | 152 | | | | | |
| W12 | | | | 40 | 168 | | | |
| W13 | | | | | 240 | | | |
| W14 | | | | | 96 | 24 | | |
| W15 | | | | | | 80 | | |
| W16 | | | | | | | 80 | |
| W17 | 144 | | | | | | 24 | 44 |
| W18 | 92 | 52 | 88 | | | | | |
| W19 | | | 144 | 48 | | | | |
| W20 | | | | | 224 | | | |
| W21 | | | | | 232 | | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | | 80 | |
| W24 | 136 | | | | | | 24 | 40 |
| W25 | 96 | 48 | | | | | | |
| W26 | | | 160 | | | | | |
| W27 | | | 60 | 37 | | | | |
| W28 | | | | | 240 | | | |
| W29 | | | | | 216 | 88 | | |
| W30 | | | | | | | 80 | |
| W31 | | | | | | | 8 | 40 |

Project ID: GP5

| | PRS | PRSV | PDD | PDDV | PCD | PCDV | PSIT | PUAT |
|---|---|---|---|---|---|---|---|---|
| W1 | 160 | | | | | | | |
| W2 | 68 | 44 | 80 | | | | | |
| W3 | | | 160 | | | | | |
| W4 | | | | 40 | | | | |
| W5 | | | | | 242 | | | |
| W6 | | | | | 240 | | | |
| W7 | | | | | | 80 | | |
| W8 | | | | | | 32 | 80 | |
| W9 | 64 | | | | | | 24 | 40 |
| W10 | 160 | 44 | 88 | | | | | |
| W11 | | | 152 | | | | | |
| W12 | | | | 40 | 168 | | | |
| W13 | | | | | 240 | | | |
| W14 | | | | | 120 | 32 | | |
| W15 | | | | | | 80 | | |
| W16 | | | | | | | 80 | |
| W17 | 136 | | | | | | 24 | 32 |
| W18 | 96 | 48 | 90 | | | | | |
| W19 | | | 144 | 48 | | | | |
| W20 | | | | | 240 | | | |
| W21 | | | | | 248 | | | |
| W22 | | | | | | 80 | | |
| W23 | | | | | | 24 | 80 | |
| W24 | 144 | | | | | | 32 | 32 |
| W25 | 88 | 48 | | | | | | |
| W26 | | | 160 | | | | | |
| W27 | | | 60 | 44 | | | | |
| W28 | | | | | 240 | | | |
| W29 | | | | | 240 | 90 | | |
| W30 | | | | | | | 88 | |
| W31 | | | | | | | 16 | 40 |

## LIST OF PUBLICATIONS

| Title | Journal Name | Link | Date | Index |
|---|---|---|---|---|
| Need for blending Agile Methodologies and Lean thinking for ERP implementation: An industry point of view | IEEE Explore (Scopus) | http://ieeexplore.ieee.org/document/7375221/ | Sep -15 | Scopus |
| Scrutinizing Lean Thinking and Agile Methodologies from Practitioner's Point of View. | Indian Journal of Science and Technology | http://indjst.org/index.php/indjst/article/view/89068 | May-16 | Scopus |
| A Novel Framework for Requirement Prioritization for ERP Implementation | Indian Journal of Science and Technology | http://indjst.org/index.php/indjst/article/view/104021 | Dec-16 | Scopus |
| Applicability and issues in traditional model of ERP implementations: an industry perspective | International Journal of Advanced Computer Research | http://accentsjournals.org/paperInfo.php?journalPaperId=944&countPaper=102 | Apr-17 | Scopus |
| GenNext: Framework for Optimizing ERP Implementations | Indian Journal of Science and Technology | http://indjst.org/index.php/indjst/article/view/110913 | Jun-17 | Scopus |
| Effectiveness of GenNext Framework on critical parameters of ERP Implementation: A statistical comparison of Traditional Methodology and GenNext Framework. | International Journal of Advanced Computer Research | http://accentsjournals.org/PaperDirectory/Journal/IJACR/2017/9/3.pdf | Aug-17 | Scopus |
| Analysis And Comparision Of Traditional And Industrial Erp Implementation Models: A Call For New Model | International Journal of Engineering and Science Invention | http://www.ijesi.org/current-issue.html | Feb-19 (Accepted) | EBSCO |

# CURRICLULUM VITAE

Sunil Kaushik holds a B.E. (Hons.) and MBA. He has approximately 2 decades of experience in implementing , managing Oracle Apps ERP applications. He has performed various roles ranging from Technical Consultant, Functional Consultant, Technical Architect , Project Manager and Senior Program Manager at various consulting firms and have worked across various geographies of the world. He is a regular speaker at the CIO forms, Oracle Open Worlds. He has intertest in applying various ideas to ERP implementation programs to increase success and predictability of the program. His research interest includes enterprise systems, change management and research methodology.