CHAPTER 9

SUMMARY AND CONCLUSIONS

9.1 SUMMARY OF RESEARCH PROBLEM

9.1.1 Background of Study

Formulating reliable and fault tolerant software is difficult and requires discipline both in specifying system functionality and in implementing systems correctly. Approaches for developing highly reliable software include the use of formal methods, and rigorous testing methods.

Testing cannot assure whether commercial and business software are correct, and verification requires enormous human effort and is subject to errors. Automated support is necessary to help ensure software correctness and fault tolerance.

Fault injection modelling involves the deliberate insertion of faults or errors into a computer system in order to determine its response. It has proven to be an effective method for measuring and studying response of defects, validating fault-tolerant systems, and observing how systems behave in the presence of faults. In this study, faults are injected in key phases of software development of business application following a typical water fall software life cycle viz., SRS, Design and Source code.

Current fault injection tools have several shortcomings that limit their usage studies without customizing the fault injector in accordance to the target system. One reason for this is that many of the tools were built to demonstrate a novel fault injection method and were then optimised to inject the specific fault into a selected target system with a low intrusion level in hardware and mission.life critical software applications.
In the background described above, the researcher identified a need to understand the defect prediction, leakage and amplification patterns in business applications which would benefit the Indian IT industry moving towards quantitative process management.

9.1.2 Need for the Research Study

In commercial and business software testing, fault injection is a technique for improving the coverage of a test by introducing faults in order to test code paths, in particular error handling code paths that might otherwise rarely be followed. It is often used with stress testing and is widely considered to be an important part of developing robust software. The Carnegie Mellon Software Engineering Institute and The Defense Acquisition University Program Manager Magazine reported that at least 42-50 percent of the software defects are originated at the requirements phase.

The work undertaken and data reported are from large projects involving life critical and mission critical applications. There are numerous limitations that prevail in the present business context and methods used for software reliability.

The reasons listed above and demand from High Mature IT organisations was the driving force for experimenting and studying fault injection patterns on business applications to predict defect amplification patterns across life cycle phases. This is validated in Cambridge Solutions experiment results.

The advantages of understanding and validating defect amplification index across software lifecycle phases make organisations suitable for understanding, designing and testing fault tolerant business applications.
9.1.3 Statement of the Problem

The problem of the present research has emerged as “The Study Of Fault Injection Patterns In Software Development To Analyse Defect Leakage And Amplification”.

9.2 OBJECTIVES OF THE RESEARCH STUDY

The key objectives of the research are listed below:

a) To study and statistically analyse the behavior of faults and defects pattern by injecting known defects in business software applications artifacts (Software Requirements Specification, Design, Source Code)

b) To study the “Amplification Index”, Domino Effect and defect leakage in key Software Development Life Cycle (SDLC) phases (Requirements, Design, Coding, Testing) of business application development, with domain and technology variants.

9.2.1 Scope and Delimitations of the Study

The scope of business system applications selected for introduction of defects is classified under following:

a) Domain Variant Systems

b) Technology Variant Systems

The main delimitations of the research are listed below:
The study is limited to Software Development projects following a defined contemporary Waterfall software life cycle development Model Methodology. The study is selected and limited to four similitude business applications available in domain verticals in the IT industry in India which could provide statistically valid data and information.

9.3 SUMMARY OF RESEARCH METHODOLOGY

9.3.1 Research Design

Keeping in view the two key objectives of the research, the study was designed consisting of the following five phases:

- **Phase I** Understanding Business Domain and Technology
- **Phase II** Project Planning and Application Selection
- **Phase III** Development of Application and Project Work Products
- **Phase IV** Injection of Defects
- **Phase V** Study and Analysis Defects

9.3.2 Domain and Technology Assessment

The selection of domain and technology to conduct the research study was carried out through industry professional interaction and brainstorming. For this purpose the researcher had discussions with 10 industry quality assurance professional experts. Stratified purposive sampling technique was used. The sample was selected from three strata viz. QA professionals working in Indian IT Organisations inside and outside Tamil Nadu

a) CMMI High Maturity Organisations
b) CMMI Low Maturity Organisations
c) ISO 9001 Certified Organisations.

9.3.3 Development of Applications

The activities performed in each step of application development for research study are listed below:

Step 1: Formulation of software project plan
Step 2: Developing the Software Requirements
Step 3: Developing the Software Design
Step 4: Developing the Application Source Code
Step 5: Testing of the Applications
Step 6: Consolidation of results for analysis

9.3.4 Introduction to Applications

The full lifecycle applications developed for the research study are listed below:

a) Library Management System (LMS)
b) Post Office Management System (POMS)
c) Point Of Sales (POS)
d) Audit Tracking System (ATS)

9.3.5 Tools Developed for Research and Data Collection

Four categories of tools were developed and used for collecting data, information gathering, work product preparation and review of project artefacts for the study are listed as follows:

a) Templates
i. Project Plan

ii. Test Plan

iii. Software Requirement Specification (SRS)

iv. Software Design Document

b) Forms

i. Review form

ii. Test Record

iii. Results consolidation Form

c) Checklists

i. Checklist for collection of domain and technology information

ii. Checklist for review of Software Requirements

iii. Checklist for review of Software Design

iv. Checklist for review of Software Source Code

v. Checklist for review of Software Test Case

d) Application Development Standards

i. Coding standards for C#.Net

ii. Coding standards for VB

iii. Coding standards for Java

MiniTab Version 12.0 statistical tool package and MS Office 2003 Suite was used for data tabulation, consolidation, analysis of defect data and validation.
9.3.6 Methods of Data Collection

The researcher himself visited organisations for collecting the data and analysis. Causal analysis and meetings were arranged with each of the subject matter expert individually as per their convenience. These meetings were conducted according to a plan based on the completion of the application development phases. A few interactions were exchanged through emails and other modes of communications with the subject matter experts. The defect data was reported by the respective stakeholders in the appropriate research tools.

The steps followed for collecting the data are represented below:

Step 1: Briefing about the research project and steps involved in data collection
Step 2: Interview with industry experts for domain and technology assessment
Step 3: Selection of applications developments and phases for data collections
Step 4: Development of Application and associated project/engineering artifacts
Step 5: Review/Testing of engineering artifacts to identify and document defect
Step 6: Collection of review and test defects for engineering artifacts and recording defect data
Step 7: Independent injection selected defects in baselined engineering artifacts
Step 8: Application wise consolidation of defect data
9.3.7 Data Analysis and Control

Evidently data analysis was designed relating to the objectives of the study.

The techniques used for analyzing the data are listed below:

a. The responses to the questionnaire on domain and technology survey will be analysed using appropriate statistical techniques.

b. The percentage of defect leakage will be calculated for each technologies and domain.

c. Domino effect of injected defects across phases of the development lifecycle.

d. Amplification Index of defects across each work products as the defects moves from one phase (Parent) to another phases (Child).

e. Hypothesis on statistical significance to amplification index across technologies and domains will be tested by applying ‘T’ test.

Restricted access rights were provided to the development team to access the project artifacts and data provided. Data sanity and integrity was ensured at each stage of data capturing (input), processing and output. Tools for data capturing ensured data input and transcription errors were prevented. Further, the data was validated with relevant stakeholders on a sample basis and discussed with subject matter experts during causal analysis.

The data collected for research was verified and validated by independent subject matter experts from Cambridge Solutions Limited., Chennai. Their evaluation of the research data from the applications confirmed
validation of data collection and resulting models. A report of their findings was submitted to the researcher.

9.4 MAJOR FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

9.4.1 Major Findings and Conclusions

The key findings and conclusions from the research study and experimentation which were interpreted and statistically validated are listed below:

a) Technology variant results:
   i. Requirement defects amplification index across on identified technologies remains are same
   ii. Design and Code defects amplification index across on technologies vary based on technologies for the common application developed in the same domain

b) Domain variant results:
   i. Within the experimentation limits, it was concluded and validated statistically that there is no statistically significant relationship, and correlation between defects amplification index on requirements vs design and design vs code on applications developed under different domains under a common technology

c) “Domino” effects chart was developed based on the above variants.
d) The defect leakage and analysis study concluded that the number of defects increases exponentially as a direct result of defects leaked from previous stages.

9.4.1.1 Amplification Index Trend Results

The summary of Amplification Index (AI) in the study of technology and domain variant applications are summarised below:

a) Amplification Index Trend for Technology Variant Applications

The Amplification Index indicates the extent of damage caused by a defect in various phases of the project. The index increases with every step in the life cycle of the project. This is evident in the case of Microsoft technologies (VB and C#.Net) but AI in the case of open source technologies such Java, the AI increases in requirements and design but in Code, it is found to have marginal decrease compared to other technologies. It is also seen that defects amplification in the Microsoft (VB) Technology show substantial increase in the amplification index across phases compared to other selected technologies.

The relative growth of AI across phases in Java technology is less compared to Microsoft technology. This indicates a better fault tolerance for applications developed in Java technology.

b) Amplification Index Trend for Domain Variant Applications

The Amplification Index indicates the extent of damage caused by a defect in various phases of the project. The index increases with every step in the life cycle of the project in all domains. Initial analysis of these defects in the education domain application shows substantial increase in the amplification index across phases compared to other application domains.
It was concluded and validated statistically that there is no significant relationship and correlation between AI on requirements vs design and design vs code.

9.4.1.2 Defect Leakage Results

The summary of Defect Leakage analysis in the study of technology and domain variant applications are summarised below:

a) Defect Leakage for Technology Variant Applications

The defect leakage analysis emphasises the importance of thorough and systematic reviews in the early stages of a software project with an emphasis on defect prevention. The analysis indicates a high increase of cost and effort to remove the defects at later stages. The cost considered here is effort, which has a direct relationship. The number of defects increases exponentially as a direct result of defects leaked from previous stages.

b) Defect Leakage for Domain Variant Applications

The defect leakage analysis emphasises the importance of thorough and systematic reviews in the early stages of a software project with an emphasis on defect prevention. The analysis indicates a high increase of cost and effort to remove the defects at later stages. The cost considered here is effort, which has a direct relationship. The number of defects increases exponentially as a direct result of defects leaked from previous stages.

9.4.1.3 Domino Charts

The domino charts for the applications developed in the variants of technologies and domains were prepared. The domino charts help the application development team in understanding the defects patterns during the
development phases and prepare prediction model based on the emerging patterns. These charts help in development of Process Performance Model (PPM) used in a quantitative process and project management.

### 9.4.2 Recommendations

The following recommendations are made based on the findings of the study. These recommendations are applicable for all types of IT organisations involved in the study and particularly focussed on the process development group. This research study by the researcher has put forward information and enough material for preparing and defining guidelines for the following:

#### 9.4.2.1 Fault Tolerant System

The study of the defect types caused due to other defects enable organisations to establish guidelines in building business applications considering technologies and domains used in the scope of the study that can withstand potential failures, improve fault tolerance and hence lend continuity to business applications. The down time of business software systems can thus be reduced improving reliability, availability and customer satisfaction meeting the agreed and defined Service Level Agreements (SLAs).

#### 9.4.2.2 Defect Prevention and Prediction Guidelines

This research study of defect types has also enabled steps that can be taken by software development teams to prevent defects and to predict the type of defects that can potentially occur in a software system. The researcher has show ways to establish Process Performance Models (PPM) to understand sub-processes variations in software development lifecycles. This information
would facilitate organisation targeting CMMI high maturity and quantitatively managed processes.

9.4.2.3 Impact on Organisation Cost of Software Quality (CoSQ) and Cost of Rework with Defect Prevention and Prediction Guidelines

a) Cost of Software Quality: Impact on organisational Cost of Software Quality baselines as result of adopting the guidelines prescribed for Defect Prevention and early defect prediction is expected to yield in reduced cost of software quality.

b) Cost of Rework: The direct impact on the organisational defect prevention and prediction guidelines is expected to reduce the Cost of Rework in software development.

9.4.3 Conclusion

In the outcome of the research study the Amplification Index patterns across software development lifecycle in different applications developed in variant technologies are published. Further, fault tolerance patterns in technologies and domains (applications) quantitatively exhibited facilitating professionals in identifying technologies and domains for fault tolerance business application development. This is a significant contribution done by the researcher for developing defect amplification patterns in business applications. It fulfils a much needed direction for the Indian IT organisations targeting High Maturity and quantitative process improvements for developing Defect Prevention and process predication models which was piloted and validated by an IT organisation in Tamil Nadu, India. Thus, the research study has contributed a significant milestone in the journey of quality improvement of business software application development.
9.5 DIRECTIONS FOR FURTHER RESEARCH

Considering the limitation of scope, time, resources and funding for the current research study, the researcher has enlisted the scope of future progressive research study and extension of this study in this chapter.

The following are the likely list of scope for further extension of this research study program:

a) This research study could be extended to other business domains such as Telecom, Insurance and Financial services.

b) This research study could further analyse the effect of the defects in the Maintenance phase of the project.

c) Test Stop Criteria for application in different domains and technologies through error seeding during application testing could be established and studied for the patterns.

d) Guidelines for software artefacts review time and effort estimation derived by analysing and definition of the review and test stop criteria could be studied and profiled.

The scope of extension for this research study is more prominent for a High Mature organisations involved in the establishment and implementation of quantitative process performance models to understand as well as predict statistical sub-process performance variations in the project’s critical sub-processes using organisational process baselines. These studies would help high mature organisations to establish appropriate process performance models based on the different buckets of organizational baselines.

The research could be extended to understand product reliability and capability index.