CHAPTER 9

CONCLUSIONS AND FUTURE DIRECTIONS

9.1 Introduction .................................................................................................................. 147
9.2 Conclusions and Significance ...................................................................................... 147
9.3 Limitations and Future Directions ............................................................................... 150
9.1 INTRODUCTION

History shows that every time researchers and practitioners face a software crisis that needs a large number of human beings, technologies, methods and processes to solve it. There are not enough experts to produce all the software according to the economy and society need. To complete the need of society, this study has examined the Component-Based Software Engineering (CBSE) which is very useful for developing complex software because traditional software development process is not enough for complex software development. CBSE has received considerable attention among software developers, vendors and IT organizations. CBSE offers many potential benefits such as greater reuse and a commodity-oriented perspective of software. Researchers and practitioners found CBSE as the best way to develop complex software by using main concept of reusability. This study is a formidable task in the field of CBSE to complete so far and is an incipient effort towards the ultimate goal of giving full coverage of the field of Component-Based Software Reuse (CBSR) and Component-Based Software Testing (CBST). These approach advocates the acquisition, adaptation, integration and testing of reusable software component for rapid development and deployment of complex software systems with minimum engineering efforts and resource cost.

9.2 CONCLUSIONS AND SIGNIFICANCE

Before carrying out the present research work all the objectives were well thought and laid down. The primary objectives of the research were to study existing software models and testing processes, proposing a new component-based reusable model and testing processes. Extensive literature survey and analysis was carried out to understand the existing research work in the area of software models, Component-Based Development (CBD), software reuse and testing, component-based software, component reuse and testing. This study presents a X model for development of component-based software; modify development process of CBSE and two component-based testing processes, namely, test case
process documentation and process to construct a testable component. Six objectives were accomplished during the research work.

First objective was accomplished by performing critical analysis and survey of various challenges of CBSR. The CBSR orientation provides many advantages, but it also requires systematic approach for reuse of components. Some of the major challenges of CBSR have been concluded in the study such as component functionality, programming languages, component generality, encoding scheme and data format, time and efforts, algorithm and data structure, software system evolution, compatibility and maintenance, which are helpful for software engineers, researcher and practitioners for developing a reusable component through CBSR. These challenges are very helpful for software industry if software developer give a hard look at the allure of productivity enhancement of Component-Based Software Development (CBSD). If all these challenges are kept in mind during development of reusable component, the size, complexity, cost, and time will definitely be reduced. This study concludes few factors for success and failure of the CBS with non-technical aspects. So if software developers keep these factors in mind with the new directions proposed by this study, the quality of software can be easily improved.

Second objective was accomplished by developing a proposed X component-based model for development of component-bases software. This study concludes that X model supports reusability, testing and maintainability by two processes - development for reuse and development with reuse (development after modification and development without modification). Software engineer identifies potentially reusable components from existing reusable component repository of X model. A repository should address the problem of conceptual closeness to retrieve components that are similar to but not exactly the same as the desired one, therefore, the X model with two repositories, namely, Testable Component Repository (TCR) and Reusable Component Repository (RCR) solve this problem.
of retrieval. This study also concludes that X model can easily remove the problem of iteration and overlapping of phases.

Third objective was accomplished by proposing a modified development process of CBSE. Modified development process of CBSE helps in developing a reusable component. This reusable component helps in problem solving with different abstraction level in programming. This chapter concludes the benefits of reusable component in programming with the modified development process of CBSE.

Fourth objective was accomplished by concluding the factors which help in reducing the cost of software development and time in testing. However, the complexity of the software needed to support the computerized economy is increasing at an alarming rate. The size of software products is no longer measured in terms of thousands of lines of code, but millions of lines of code. This increasing complexity along with inadequacy of testing technique and testing infrastructure decreasing average market life expectancy for many software products. So this study concludes that if researchers and practitioners want to reduce the cost of software development and time in testing then they will try to give main emphasis to decrease the following four factors - failures due to poor quality, software development costs, time to market. These all factors arise due to inefficient testing and testing infrastructure. This study discusses and finds the ways, how to reduce the testability challenges in reusable component with the help of software components features, namely, test cases, testability and reliability, test drivers and stubs, and reusable test beds.

Fifth objective was accomplished by proposing component-based test process. This study concluded if software engineers and researchers seek to develop reusable component, they develop it with the two factors viz. CBST requirement and test case process documentation through modified component-based test process documentation. This process enhances the reusability of reusable components and due to good documentation, it is easy to reuse with another system.
Sixth objective was accomplished to propose a new test process to construct the
testable component with good interface. This study concludes that a better testable
component is developed with this process, which help in increasing component
testability with a well defined standard component interfaces.

9.3 LIMITATIONS AND FUTURE DIRECTIONS
Proper development and management of reusable component to develop software
can be serious advantage of software vendors over their competitors. Researchers
and practitioners can analyze how to successfully and quickly produce and deploy
such software component, without incurring additional overhead for development.
Although, as a form of advancement in CBSE, CBSR and CBST have number of
advantages, it also has a number of limitations. Simple and testable components
of component-based software during testing require proper environment when
software tester want to test these components in another application with different
platform and architecture. To enhance reusability, component standardization is
the most important obstacle to the full blossoming of the components-based
development. Modified development process helps in developing only software
component but it is not able to develop component-based software. Process to
develop testable component is only able to develop a component with better
interface but it does not able to develop a reliable component. X model helps in
removing iteration and overlapping to some extent.

The conclusions obtained from present study are quite useful for software
industry to realize the development and implementation in real life projects. Using
projects from industry provides better opportunity to examine the working of X
model, modified development process and component-based testing process. X
model may be further empirically validating using industrial projects. Both the
components-based test processes may also require further standardization and
validation with the real life projects from industry. Therefore, the future work in
this direction needs to be carried out on the related research objectives but it
requires the wide applicability on industry projects.