

# A Workflow Based Online Software Review System

Hasan ÇİFCİ

June 2004

A WORKFLOW BASED ONLINE SOFTWARE REVIEW SYSTEM

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF INFORMATICS  
OF  
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

HASAN ÇİFCİ

IN PARTIAL FULFILLMENT FOR THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE  
IN  
THE DEPARTMENT OF INFORMATION SYSTEMS

JUNE 2004

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

---

Hasan ÇİFCİ

Approval of the Graduate School of Informatics

---

Prof. Dr. Neşe YALABIK  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Assoc. Prof. Dr. Onur DEMİRÖRS  
Head of Department

This is to certify that we have read this thesis and in our opinion, it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Assoc. Prof. Dr. Onur DEMİRÖRS  
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Onur DEMİRÖRS

---

Assoc. Prof. Dr. Nazife BAYKAL

---

Assist. Prof. Dr. Kürşat ÇAĞILTAY

---

Dr. Altan KOÇYİĞİT

---

Dr. Sadık EŞMELİOĞLU

---

# **ABSTRACT**

## **A WORKFLOW BASED ONLINE SOFTWARE REVIEW SYSTEM**

Çifci, Hasan

M.S., Department of Information Systems

Supervisor: Assoc. Prof. Dr. Onur Demirörs

June 2004, 123 pages

Software review is an effective and efficient way for detecting defects in software artifacts. To improve the effectiveness and the efficiency of software review, a variety of software review techniques have been developed. Furthermore, computer support has been used to facilitate the software review process. Accordingly, several software review tools have been developed. Although existing tools provide new solutions to support software review, in general they suffer from a number of shortcomings.

In this study, a brief description about major software review techniques is given along with the summary of the process. Additionally, existing tools and their features are briefly explained and a comparison between them is done. This study focuses on the development of a Web based review tool that has a workflow definition capability in order to support every kind of software review processes.

**Keywords:** Online Software Review, Workflow Based Software Review, Web Based Software Review.

# ÖZ

## İŞ AKIŞI TABANLI ÇEVİRİM İÇİ BİR YAZILIM GÖZDEN GEÇİRME SİSTEMİ

Çifci, Hasan

Yüksek Lisans, Bilişim Sistemleri

Tez Yöneticisi: Doç. Dr. Onur Demirörs

Haziran 2004, 123 sayfa

Yazılımın gözden geçirilmesi, yazılım ürünlerinde bulunan hata ve eksikliklerin etkili ve verimli bir şekilde ortaya çıkarılmasını sağlayan yöntemlerden biridir. Gözden geçirme işleminin etkinliğini ve verimliliğini artırmak için çeşitli yazılım gözden geçirme teknikleri geliştirilmiştir. Buna ek olarak, yazılım gözden geçirme sürecini kolaylaştırmak için bilgisayar desteği kullanılmaya başlanmıştır. Bu yüzden bir çok yazılım gözden geçirme aracı geliştirilmiştir. Mevcut araçlar, yazım gözden geçirme işlemini desteklemek için yeni çözümler ortaya koymalarına rağmen, genellikle bir takım yetersizlikle sahiptirler.

Bu çalışmada, önemli yazılım gözden geçirme teknikleri ve süreçleri hakkında kısa açıklamalara yer verilmiştir. Ayrıca, mevcut araçların özellikleri kısaca açıklanmış ve bu araçların kıyaslamaları yapılmıştır. Bu çalışma, her tür yazılım gözden geçirme sürecini destekleyebilmek için iş akışı tanımlama kabiliyetine sahip, Web tabanlı bir yazılım aracının geliştirilmesi üzerine odaklanmaktadır.

Anahtar Kelimeler: Çevrim İçi Yazılım Gözden Geçirme, İş Akışı Tabanlı Yazılım Gözden Geçirme, Web Tabanlı Yazılım Gözden Geçirme.

To my Wife Bahar  
and  
my Daughter Fatma Elifnur.

## **ACKNOWLEDGEMENTS**

I would like to express grateful acknowledgement for the valuable suggestions and help given by my advisor Assoc. Prof. Dr. Onur Demirörs.

I want to thank to Selahattin Ünal, Murat Günay, Tacettin Köprülü, Naciye Yağan and Halime Orhon for their assistance during the proofreading of the thesis document, and Captain Naci Kaygusuz and Lt. Colonel Turgay Aksaray for their helps.

I also would like to express my deepest appreciation to my wife Bahar who always gave me her love and emotional support and my dear daughter Fatma Elifnur for her loveliness.



# TABLE OF CONTENTS

ABSTRACT .....	IV
ÖZ.....	V
DEDICATION .....	VI
ACKNOWLEDGEMENTS .....	VII
TABLE OF CONTENTS .....	VIII
LIST OF TABLES .....	XII
LIST OF FIGURES.....	XIII
LIST OF ABBREVIATIONS .....	XV
CHAPTER	
1. INTRODUCTION.....	1
1.1    Problem Statement .....	2
1.2    Approach .....	3
1.3    Thesis Structure.....	5
2. MAJOR SOFTWARE REVIEW TECHNIQUES.....	7
2.1    Fagan’s Software Inspection.....	7

2.2	Structured Walkthroughs .....	9
2.3	Humphrey Inspection .....	10
2.4	FTArm.....	11
2.5	Gilb Inspection .....	13
2.6	Phased Inspection.....	14
2.7	N-fold Inspection .....	15
2.8	Active Design Review (ADR) .....	17
2.9	Code Reading .....	17
2.10	Two-person Inspection.....	18
2.11	Cleanroom Approach .....	18
3.	TOOL SUPPORT FOR SOFTWARE REVIEW .....	19
3.1	Tools for Paper Based Review .....	19
3.2	Tools for Online Review .....	20
3.3	Comparison of Tool-Based and Paper-Based Software Inspection .....	27
3.4	Comparison of Existing Tools .....	28
4.	SOFTWARE REQUIREMENTS AND DESIGN.....	31
4.1	Introduction .....	31
4.1.1	Scope .....	31
4.2	Product Perspective.....	32
4.3	Constraints, Assumptions and Dependencies .....	33
4.4	Specific Requirements .....	34
4.4.1	Log on to the System .....	34
4.4.2	Operations for Administering the System.....	34
4.4.3	Operations for Conducting Review Activities .....	48
4.5	Logical Database Requirements.....	56
4.6	System Architecture .....	59
4.6.1	First Tier (Client): .....	59
4.6.2	Middle-Tier: .....	60
4.6.3	Back-end Tier:.....	60
4.7	Decomposition Description.....	60
4.8	Dependency and Detailed Descriptions .....	61
4.8.1	GeneralProjectManager Class:.....	61

4.8.2	IssueTypeManager Class: .....	61
4.8.3	ProjectManager Class: .....	62
4.8.4	RoleManager Class: .....	62
4.8.5	SqlBean Class: .....	62
4.8.6	TaskProcessor Class:.....	62
4.8.7	UserManager Class: .....	63
4.8.8	WorkflowManager Class: .....	63
4.8.9	Data Classes: .....	63
4.9	User Interface Description .....	63
4.9.1	Main Page for Administration.....	65
4.9.2	Welcome Page of Review Participants .....	66
4.9.3	Main Page for Review Operations .....	66
5.	EXPERIMENTAL STUDY .....	68
5.1	Objectives.....	68
5.2	The Experiment.....	69
5.2.1	Participants.....	69
5.2.2	Location and Apparatus .....	69
5.2.3	Experiment Procedure.....	69
5.2.4	Limitations of the Experiment .....	71
5.3	Configuration of the Tool .....	71
5.4	Evaluation of the Tool.....	73
5.4.1	Fulfillment of Objectives .....	73
5.4.2	Usability of the Tool .....	76
5.4.3	Meeting the Specification of Review Procedure.....	78
6.	CONCLUSION AND FUTURE DIRECTIONS.....	82
6.1	Conclusion .....	82
6.2	Future Work .....	85
	REFERENCES.....	86
	APPENDICES	

APPENDIX A: ENTITY RELATIONSHIP DIAGRAM .....	91
APPENDIX B: COMBINED CLASS DIAGRAMS.....	92
APPENDIX C: SAMPLE USER INTERFACES.....	102
APPENDIX D: PARTICIPANT CONSENT FORM.....	105
APPENDIX E: PRE-TEST QUESTIONNAIRE.....	106
APPENDIX F: EASYREVIEW EVALUATION QUESTIONNAIRE	107
APPENDIX G: USABILITY EVALUATION RESULTS .....	109
APPENDIX H: REVIEW PROCESS OF THE ORGANIZATION ....	111
APPENDIX I: SPECIFICATIONS OF REVIEW PROCEDURE.....	116
APPENDIX J: INTENTIONAL MISTAKES IN THE REVIEW DOCUMENTS .....	118
APPENDIX K: COFIGURATION OF EASYREVIEW.....	121

## LIST OF TABLES

Table 1 : Comparison of Existing Tools .....	30
Table 2 : Software Products Required Implementing the System .....	33
Table 3 : Material Types that are Allowed For Online Annotation .....	53
Table 4 : Icons and Their Descriptions in EasyReview .....	64
Table 5 : Workflow of TAFF Organization's Review Procedure.....	72
Table 6 : Distribution of the Found Issues in the Experiment .....	75
Table 7 : Results of the Pre-Test Questionnaire .....	76
Table 8 : Results of the Questionnaire .....	77
Table 9 : Evaluations of the Organization's Review Procedure Specifications.....	79
Table 10 : Questionnaire for Practical Heuristics for Usability Evaluation.....	107
Table 11 : Usability Evaluation Results.....	109
Table 12 : Review Summary Report.....	114
Table 13 : Individual Review Record .....	115
Table 14 : Specifications of Review Procedure .....	116
Table 15 : Intentional Mistakes in the Review Documents .....	118

## LIST OF FIGURES

Figure 1: Fagan’s Inspection Process and Participants .....	8
Figure 2 : Structured Walkthrough Process and Participants .....	10
Figure 3 : Humphrey Inspection Process and Participants.....	11
Figure 4 : FTArm Process and Participants .....	12
Figure 5 : Gilb Inspection Process and Participants .....	14
Figure 6 : Phased Inspection Process .....	15
Figure 7 : N-Fold Inspection Process and Participants .....	16
Figure 8 : Active Design Review Process and Participants .....	17
Figure 9 : Operations for Administering the System .....	35
Figure 10 : Operations for Maintaining Roles .....	36
Figure 11 : Operations for Maintaining Issue Types .....	37
Figure 12 : Operations for Maintaining Users .....	38
Figure 13 : Operations for Maintaining Workflows .....	40
Figure 14 : Operations for Maintaining Projects .....	44
Figure 15 : Operations for Maintaining Reviews.....	46
Figure 16 : Operations for Conducting Review Activities .....	48
Figure 17 : Operations for Maintaining Issues.....	51
Figure 18 : System Architecture of EasyReview .....	60
Figure 19 : Decomposition Diagram of Package “tez” .....	61
Figure 20 : Decomposition Diagram of Package “tez.data” .....	62
Figure 21 : Main Page for Administration of EasyReview.....	65
Figure 22 : Welcome Page for Review Participants .....	66
Figure 23 : Main Page for Review Operations .....	67
Figure 24 : Workflow of TAFF Organization’s Review Procedure .....	73
Figure 25: A Seven-point Scale to Measure End-Users' Responses.....	78
Figure 26 : Class Diagram for SqlBean Class.....	92

Figure 27 : Class Diagram for GeneralProjectManager Class .....	93
Figure 28 : Class Diagram for IssueTypeManager Class .....	93
Figure 29 : Class Diagram for ProjectManager Class.....	94
Figure 30 : Class Diagram for RoleManager Class .....	95
Figure 31 : Class Diagram for UserManager Class .....	95
Figure 32 : Class Diagram for WorkflowManager Class .....	96
Figure 33 : Class Diagram for TaskProcessor Class.....	97
Figure 34 : Class Diagram for IssueResult Class.....	97
Figure 35 : Class Diagram for Issue Class.....	98
Figure 36 : Class Diagram for User Class.....	98
Figure 37 : Class Diagram for Material Class.....	99
Figure 38 : Class Diagram for Project Class.....	99
Figure 39 : Class Diagram for Task Class .....	100
Figure 40 : Class Diagram for Transition Class.....	100
Figure 41 : Class Diagram for Workflow Class.....	101
Figure 42 : Interface for Adding a New User .....	102
Figure 43 : Interface for Customizing a Review .....	102
Figure 44 : Interface for Modifying a Task within a Workflow .....	103
Figure 45 : Interface for Adding and Removing Issues (Microsoft Word Document) .....	103
Figure 46 : Interface for Adding and Removing Issues (Microsoft Excel Spreadsheet) .....	104
Figure 47 : Interface for Taking Report on Efforts .....	104

## **LIST OF ABBREVIATIONS**

ADR	Active Design Review
AISA	Asynchronous Inspector of Software Artifacts
ASSIST	Asynchronous/ Synchronous Software Inspection Support Tool
CAIS	Collaborative Asynchronous Inspection of Software
CASCADE	Computer Augmented Support for Collaborative Authoring and Document Editing
CSI	Collaborative Software Inspection
CSRS	Collaborative Software Review System
DBMS	Database Management System
DCI	Distributed Code Inspection
FTArm	Formal Technical Asynchronous review method
FTR	Formal Technical Review
GRCM	Goal Rule Checklist Metric
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IBIS	Internet Base Inspection System
ICICLE	Intelligent Code Inspection in a C Language Environment
IEEE	Institute of Electrical and Electronics Engineers
InspeQ	Inspecting software in phases to ensure Quality
IPA	Inspection Process Assistant
ISO	International Organization for Standardization
JDBC	Java Database Connectivity
JDK	Java Development Kit
JSP	Java Server Pages
JVM	Java Virtual Machine
NI	Notes Inspector



OMG	Object Management Group
QG4000	Quality Group 4000
SMTP	Simple Mail Transfer Protocol
SRS	Software Requirements Specifications
TAFF	Turkish Armed Forces Foundation
UML	Unified Modelling Language
URL	Unified Resource Locator
wbART	web-based Asynchronous Review Tool
WfMC	Workflow Management Coalition
WiP	Web inspection Prototype
WWW	World Wide Web
XATI	XML Annotation Tool for Inspection
Xfi	eXtensible framework for interoperability
XML	eXtensible Markup Language

## CHAPTER 1

# INTRODUCTION

Because of wide usage of computers, software products have become very popular today. Meyer states that, “As the use of computers pervades more and more of what society does, the effects of non-quality software just becomes unacceptable. Software is becoming more ambitious, and we rely on it more and more. Problems that could be dismissed quite easily before are now coming to the forefront” [Venners, 2003].

In the past two decades, software reviews have emerged as one of the most effective quality assurance techniques in software engineering. According to Johnson and Tjahjono, software review is a fundamental component of the software quality assurance process [Johnson, Tjahjono, 1997].

In the IEEE Standard Glossary of Software Engineering Terminology [IEEE, 1990] and IEEE Standard for Software Reviews [IEEE, 1997], review is defined as follows:

“A process or meeting during which a work product, or a set of work products, is presented to project personnel, managers, users, customers or other interested parties for comment and approval.”

According to IEEE Standard for Software Reviews [IEEE, 1997], software reviews can be used in support of the objectives of project management, system engineering, verification and validation, configuration management, and quality assurance. There are five different types of reviews and they reflect differences in the goals of each review type. Reviews are categorized into following types;

“Management Review”, “Technical Review”, “Inspection”, “Walkthrough” and “Audit”.

The fundamental goal of an inspection is to detect defects before the testing phase begins; and therefore strongly contribute to the improvement of the overall quality of the software [DeMarco, 1982], [Yourdon, 1997].

Software inspection was introduced by Michael Fagan in 1976 at IBM [Fagan, 1976] and since then it has become widely used [Wheeler et al., 1996]. After Fagan’s introduction of the general concept of inspection to the software domain at IBM in 1976, a large amount of contributions in the form of new methodologies and incremental improvements has been proposed promising to increase the inspection’s benefits within software development [Laitenberger, DeBaud, 1998].

Traditional software inspections are too exhausting and time consuming to arrange as development teams are geographically distributed, and team members are involved in many projects simultaneously [Hedberg, Harjumaa, 2001]. Without losing the advantages of traditional inspections, computer tool support for software inspections is a way of removing bottlenecks in the inspection process.

In this thesis, we developed a web based review system named EasyReview which supports different types of documents to be reviewed and has a capability to define workflows in order to customize review process according to an organization’s needs.

## **1.1 Problem Statement**

Traditional software inspection is poorly suited to modern distributed software engineering projects. Arranging inspections in a distributed and asynchronous manner is not a straightforward matter [Hedberg, Harjumaa, 2001]. Development teams are distributed geographically and it is a challenge to arrange a meeting because of scheduling and place concerns.

Perry et al. [Perry et al., 2002] suggest that the calendar time needed to complete an inspection increases when the activities of multiple developers must be

coordinated. They find that many large companies are developing software using multiple, geographically separated teams. In this situation, time-zone mismatches, travel, and long-distance mailings dramatically lengthen interval and are having considerable effects on total development time.

In a traditional review process, documents are created and then distributed to review participants. Participants read documents and guidelines to discover and record the defects. Issues are collected in the review meetings and so on. They are all error-prone routine tasks that can increase the chance of errors [Harjumaa et al., 2000].

According to the results of several experiments conducted in the University of Oulu; computerized inspection is feasible, and the concept of a distributed inspection tool with the ability to handle documents, checklists and annotations electronically is more valuable [Harjumaa et al., 2000].

Many tools have been developed to improve inspection efficiency, ranging from tools that support the workflow and bookkeeping of an inspection process, to integrated computer-aided verification techniques.

Murphy and Miller claim that existing systems have some limitations [Murphy, Miller, 1997]. Primarily, they support a single, usually proprietary, inspection process. They also support inspection of only plain text documents even though a number of different document types are produced.

## **1.2 Approach**

In this thesis, we developed a Web based review system named EasyReview that allows geographically distributed teams to conduct asynchronous software reviews. Review participants use only a Web browser at their computers to connect to the system.

Java Server Pages (JSP), JavaScript, Rational Rose Script, HyperText Markup Language (HTML) and ActiveX technologies were utilized to develop this system. Tomcat was used as a servlet container for JSP technology. The system connects the

MySQL database management system (DBMS) via Java Database Connectivity (JDBC).

Users are authenticated by their username and password pair while connecting to the system. EasyReview automatically authenticates the users and opens related screens according to the user's identity.

EasyReview supports two types of users; one of them is the administrator of the system and the other is an ordinary review participant. Administrators and ordinary users have different capabilities and screens. Administrators are able to customize the review process using workflow capability, define issue types, create users and roles, and create review projects. Ordinary users review the products, add issues and take reports about the review results.

EasyReview allows authorized users to put review materials and other supportive materials, such as standards and checklists in the server. Review participants can access and download all the review and supportive materials easily.

EasyReview supports defining new review processes according to the needs with workflow capability. The Workflow Management Coalition (WfMC) defines workflow as follows; "The computerized facilitation or automation of a business process, in whole or part" [WfMC, 1995]. WfMC points out that workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal.

In EasyReview, a workflow contains tasks, which are connected to each other as to the needed process. Every task is assigned one or more review roles, such as reviewer, moderator and author. Additional roles can also be created by the administrator whenever needed. The administrator determines what will be done in a specific task; documents will be uploaded, issues will be raised or raised issues will be criticized and eventually will be accepted or rejected.

An administrator selects a defined workflow in the system when he creates a review work. While creating reviews, users are assigned to roles, review materials

are determined, starting and ending dates of the workflow tasks are set. When an administrator starts a review work, the review process becomes active and review participants are informed via e-mail. All of the users of a review work can see the status of the review process; which task is active and which users finished their work on the active task. When a user finished his task, he selects the next task in accordance with the result of the task. When all of the users of a task have finished their tasks, the next task of the review workflow becomes active.

Although all of the document formats that are supported by Web browser are displayed; only four types of those document formats are supported to annotate in EasyReview. These are Microsoft Word document, Microsoft Excel spreadsheet, Microsoft PowerPoint presentation and Rational Rose model.

Users can add an issue by selecting the issue position in the reviewed document. They are not supposed to give any extra position information while adding an issue. The system automatically adds the issue to the selected position. According to the criticality level of the issue, color of the annotation is displayed in yellow, orange or red. To change or remove an issue, users only select the issue by a mouse click on the reviewed document without entering issue position information.

When an issue is entered in a review work, all of the users of that review work can see it. Issues are stored in the database and can be reported in terms of the specified criteria. Issues can be accepted or rejected in a task if that task is assigned for issue criticism. In addition, status of the issues -i.e. resolved or unresolved- can be updated in a task if that task is designed for updating issue status.

EasyReview allows users to take detailed reports about the review process. Users can select different options to take a report. These are; review documents, participants who have issues on the documents, issue type, criticality of the issue and status of the issue. In addition to the issue report, the report related to the estimated and real efforts for review tasks can also be taken.

### **1.3 Thesis Structure**

Chapter 2 provides a brief description about major software review techniques

along with summary of the process.

Chapter 3 presents existing tools ranging from managing and delivering inspection documentation systems to complex collaborative systems.

Chapter 4 includes the software requirement specifications of the tool and presents an overview of the design of the tool.

Chapter 5 explains how the tool overcame the shortcomings of existing tools and provides experimental study in order to examine the tool's capabilities.

Chapter 6 concludes the thesis and contains directions for future work about tool support for software reviews.

## CHAPTER 2

# MAJOR SOFTWARE REVIEW TECHNIQUES

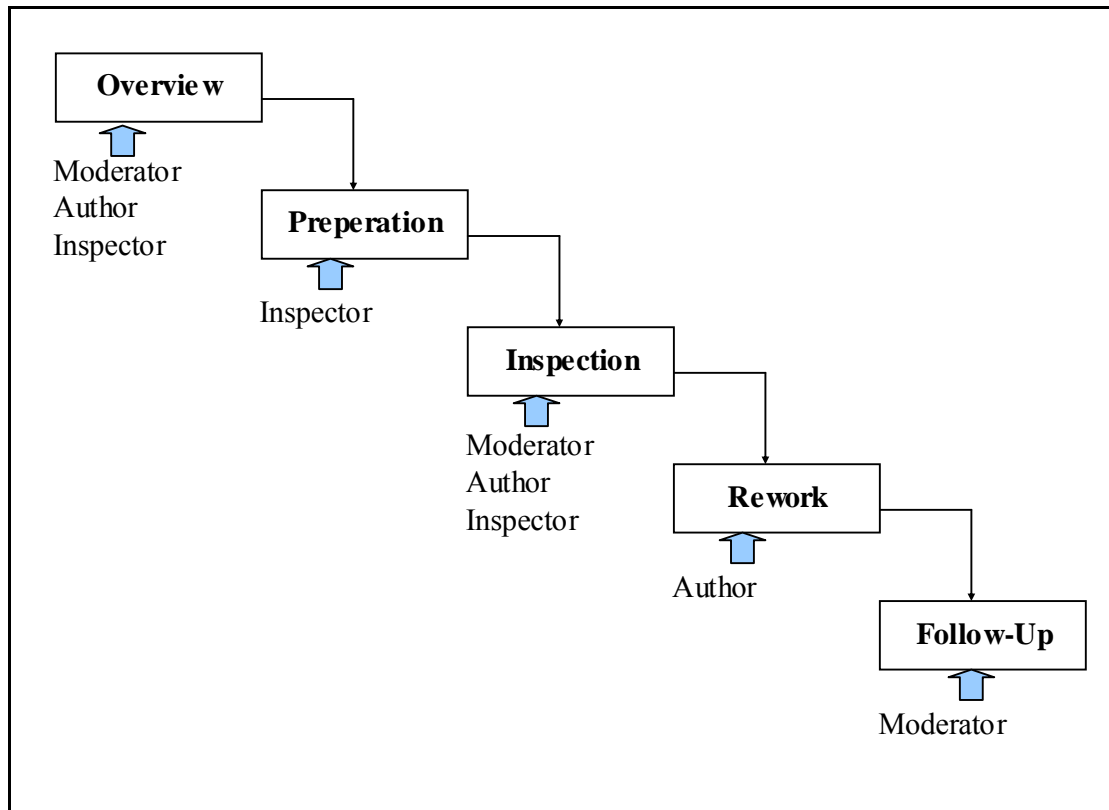
A number of new software review techniques have been developed after Fagan's Software Inspection technique. In this chapter, a brief description about major software review techniques is given along with a summary of the process.

### 2.1 Fagan's Software Inspection

Inspections are formal, efficient and economical method of finding errors in software artifacts. According to Fagan [Fagan, 1976], every program development process requires inputs and outputs that satisfy a specific set of exit criteria. He claims that the later the rework is performed the higher the cost. Consequently, detecting and fixing errors should be done as early as possible in order to reduce the overall cost of product development. Inspection insures that exit criteria are completely satisfied, minimizing the cost. Fagan's inspection emphasizes the use of inspected data to control the development process as well as the inspection process itself.

Fagan states that, software inspection is a review process for improving software development by focusing on team-based fault detection in software artifacts. There are three different roles in Fagan's process; moderator, inspector and author. Fagan advocated that meetings could increase the effectiveness of inspections. There are five operations in the inspection process as originally defined by Fagan (See Figure 1):





**Figure 1: Fagan's Inspection Process and Participants**

- Overview: The process begins with an overview. The author meets with the inspection team, describes the general area of work and gives a detailed presentation on the document produced.
- Preperation: The inspection team independently analyzes the artifact and any supporting documentation and record potential defects.
- Inspection: An inspection meeting involving all team members is held. The reader paraphrases the document. During this process, inspectors can stop the reader and raise any issues they have discovered, either in preparation or at the meeting. The issue is discussed until agreement is reached. If an issue is agreed to be a defect, it is classified as missing, wrong or extra. Its severity is also classified (major or minor). After the meeting, the moderator writes a report detailing the inspection and all defects found. This report is then passed to the author for rework.
- Rework: The author carries out modifications to correct the defects found in the document.

- Follow-up: The moderator ensures that all required alterations have been made. The moderator then decides whether to re-inspect the artifact depending on the quantity and quality of the rework.

According to Fagan, the results of inspections should never be used as criteria for assessment of programmers [Fagan, 1976]. Inspections are self-improvement processes because inspection results are used as feedback for both defects removal and professional development. Inspection statistics, such as inspection rates and error density, are collected and used for determining characteristics of inspections and for process improvement as well as defect hindrance.

## **2.2 Structured Walkthroughs**

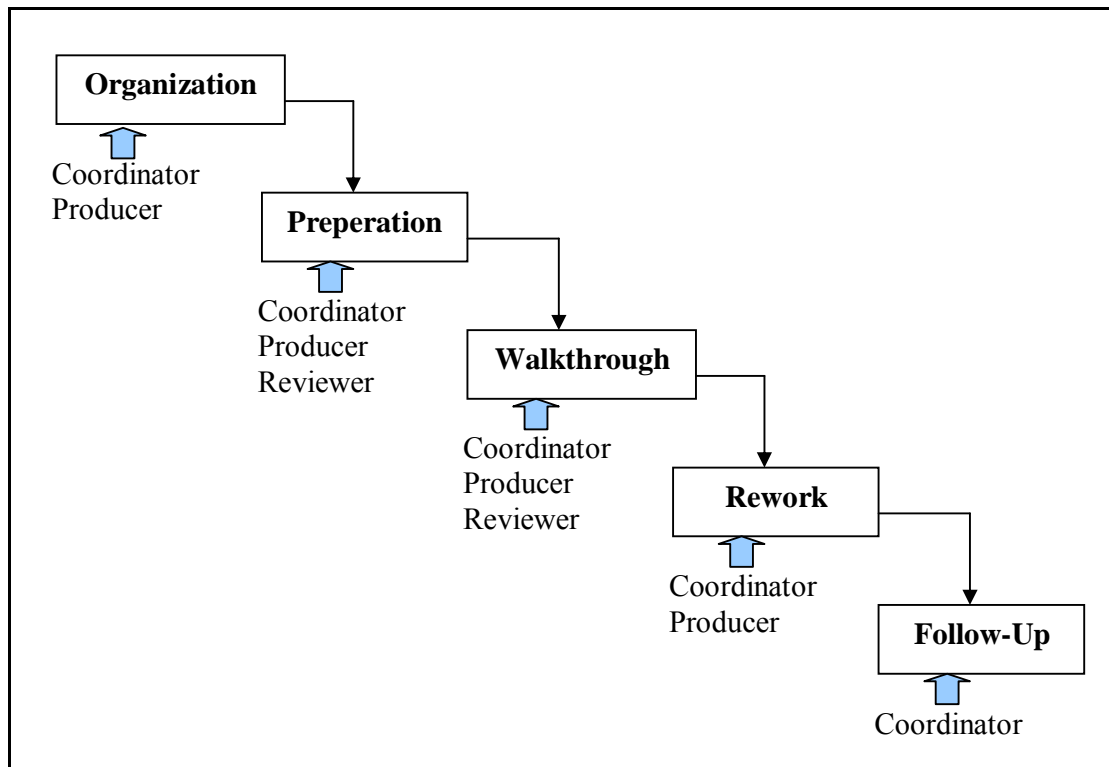
A structured walkthrough is an organized procedure for a group of people to review and discuss the technical aspects of software development and maintenance work products [Yourdon, 1989]. The major objectives of a structured walkthrough are to find errors and to improve the quality of the product. Errors typically take place as omissions, contradictions, flaws in logic, or inconsistencies in the work product style. Structured walkthroughs should not be used to discuss solutions for the errors that are found. The basic purpose of a walkthrough is error detection, not error correction. When the walkthrough is completed, the author of the work product is responsible for taking the necessary actions to correct the errors.

Structured walkthroughs should be conducted during all stages of the system lifecycle. Walkthroughs can be conducted in various formats, with various levels of formality, and with different types of participants. In some cases, end users can be included in walkthroughs. Management representatives do not participate in structured walkthroughs.

Each participant in the structured walkthrough process has a specific role. For a small size project, a person may have multiple roles. These roles are; coordinator, producer, reviewer and scribe.

The coordinator contacts participants, prepares and distributes documentation, and selects a schedule for the walkthrough. Participants spend time preparing for the

walkthrough by examining the product. The walkthrough, which is a group activity, is the central component of the review process where all participants meet and discuss or comment on the product. Then, a list of the comments is made and passed to the producer. This list is used to make changes in the product. A final management summary is also produced by the coordinator. Feedback is not used to control the development process. Walkthrough process and its participants are given in Figure 2.



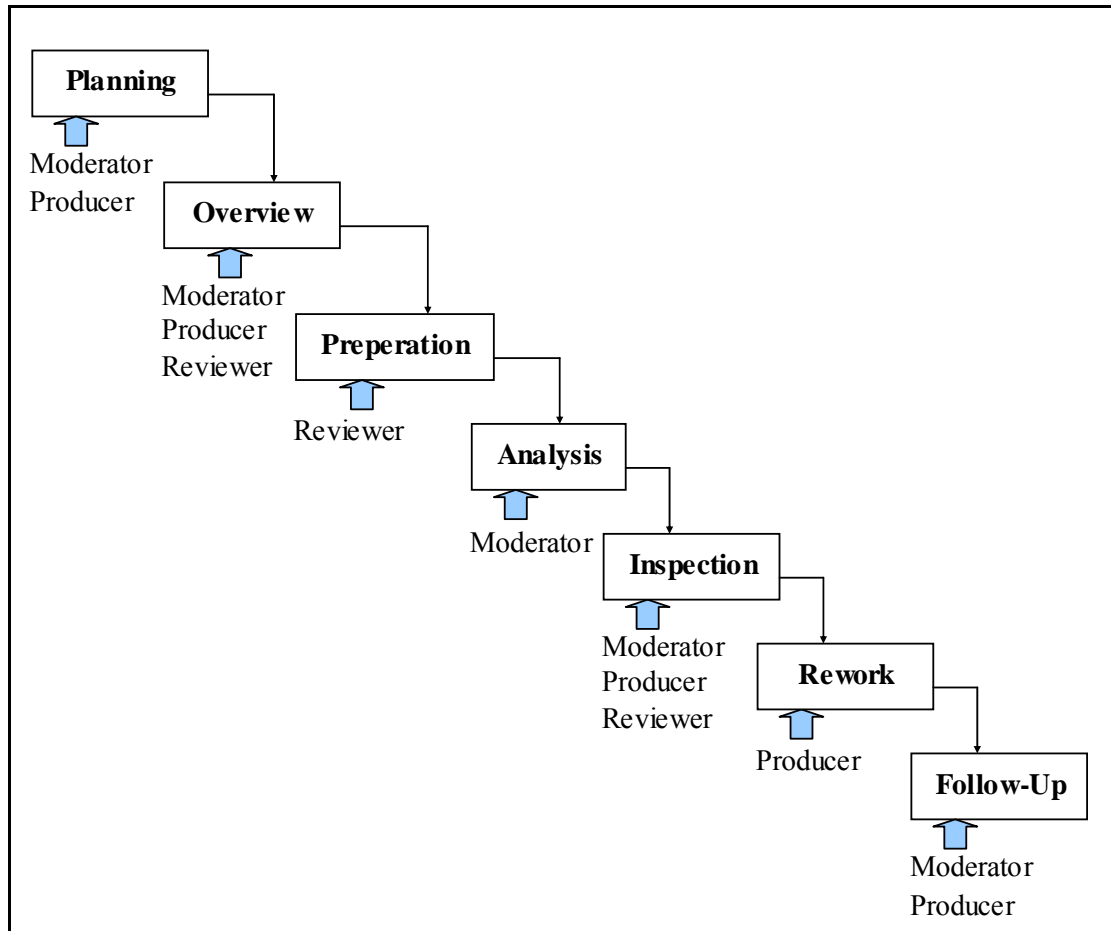
**Figure 2 : Structured Walkthrough Process and Participants**

## 2.3 Humphrey Inspection

Humphrey Inspection process [Humphrey, 1989] is very similar to Fagan Inspection process. There are differences in inspection roles, inspection stages and activities. It includes planning, overview, preparation, analysis, inspection, rework, and follow-up stages (See Figure 3).

In Humphrey Inspection, the inspection team consists of moderator, producer and reviewers. At first, the moderator selects participants and entry criteria for the inspection. In the overview stage, inspection objectives are described. Individual preparation starts after each participant receives inspection materials. Reviewers

individually inspect the material and create defect lists. All defect lists are passed to the moderator for preliminary analysis. During the inspection meeting, final defect list is produced. After inspection stage, rework and follow-up activities are performed.



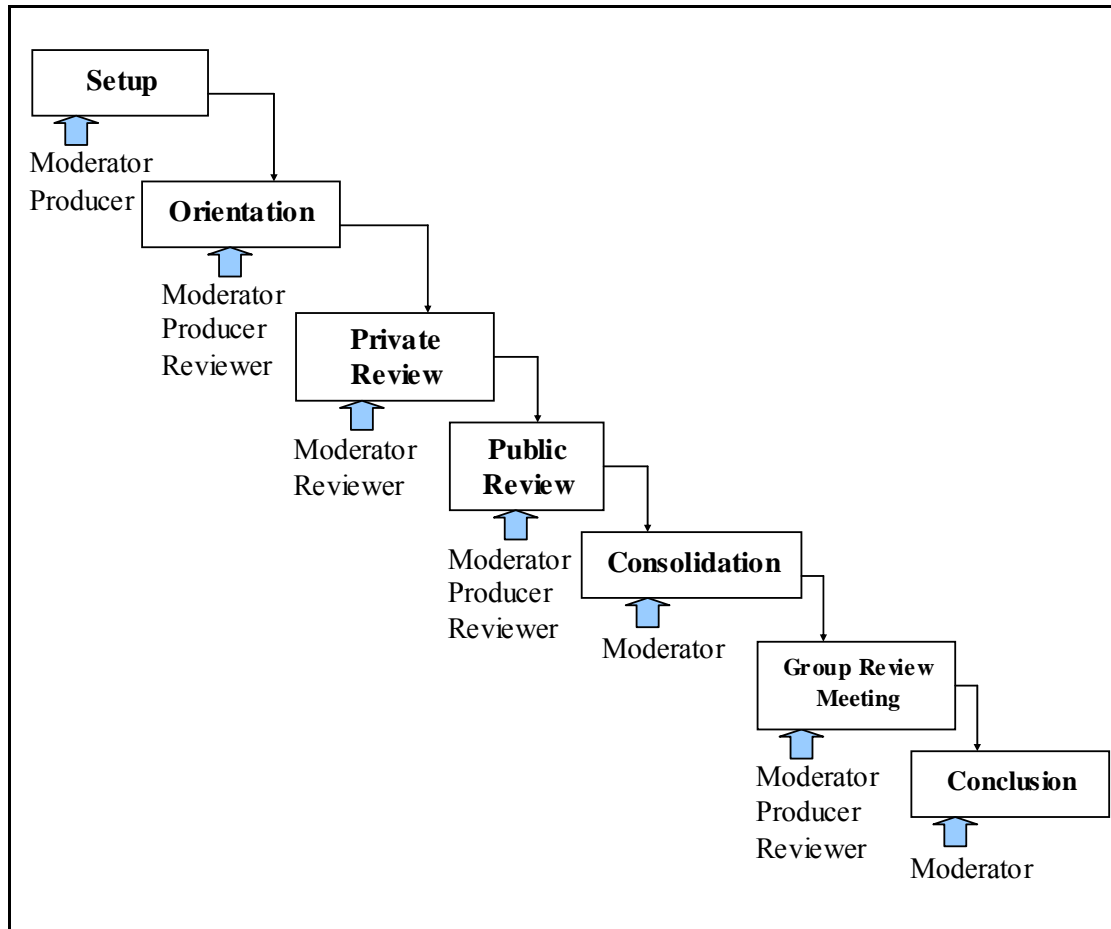
**Figure 3 : Humphrey Inspection Process and Participants**

## 2.4 FTArm

FTArm (Formal Technical Asynchronous review method) is a computer-mediated Formal Technical Review (FTR) method developed to address certain problems of manual FTR. The FTArm method is not specific to a review artifact type or development phase [Johnson, 1994]. The FTArm process consists of seven phases where participants interact within the roles of moderator, producer, reviewer, or administrator (See Figure 4).

- Setup: In this phase, the moderator and/or the producer determine the

review team and the documents to be reviewed. Documents to be reviewed are organized into a hypertext structure during this phase.



**Figure 4 : FTArm Process and Participants**

- **Orientation**: This phase prepares the participants for the private review phase through an overview of the review artifacts.
- **Private review**: In this phase, reviewers analyze the review artifacts privately and create issue, action and/or comment nodes. Issue and action nodes are not publicly available to other reviewers, though comment nodes are publicly available. The moderator can access all the nodes. This allows the moderator to monitor the progress of private review.
- **Public review**: In this phase, all nodes are now accessible to reviewers, and all other review participants. Participants can also create new issue, action or comment nodes based on the votes or nodes of others. Voting

is used to determine the degree of agreement within the group about the validity and implications of issues and actions.

- Consolidation: In this phase, the moderator analyzes the results of public and private review, and produces a detailed review report.
- Group review meeting: In this phase, the moderator presents only the unresolved issues or actions and summarizes the differences of opinion. After discussion, the group may vote to decide them, or the moderator may make the decision.
- Conclusion: The moderator produces final inspection report along with a metrics report.

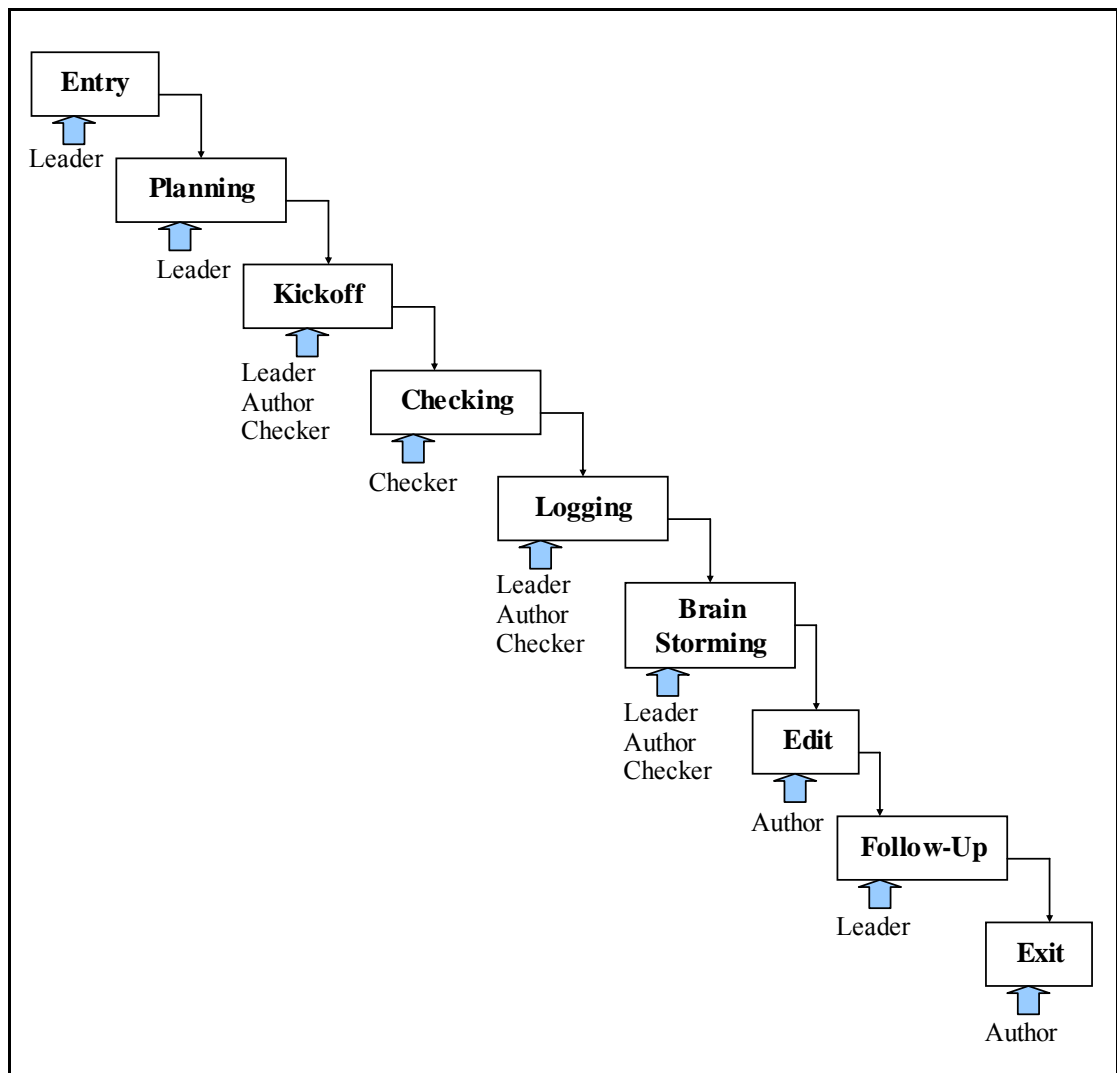
FTArm is designed to overcome weaknesses in the Fagan Inspection method, by leveraging parallel activities and employing computer control. In this inspection method, most of activities are carried out asynchronously.

## **2.5 Gilb Inspection**

This inspection method is based on Fagan Inspection; however, it has some differences [Gilb, Graham, 1993]. Inspection phases include entry, planning, kickoff, checking, logging, brainstorming, edit, follow-up and exit (See Figure 5). There are three defined roles in this type of inspection; leader, author and checker.

The inspection process begins when certain entry criteria are satisfied. In the inspection planning, the leader determines inspection participants and determines an inspection plan. In the kickoff phase, documents are distributed and the inspectors are briefed. In the checking phase, each checker works alone to find out defects in the document. The next phase, logging meeting, defects found by the checkers are collected. The leader moderates discussions during the logging meeting. In this phase, participants try to log as many issues as possible. After the logging meeting, brainstorming session may be held in order to collect process improvement suggestions. After all potential defects have been logged; the author takes the issue list and performs an edit on the product. When this phase is completed, the leader performs a follow-up and ensures that the edit phase has been properly executed.

Finally, some exit criteria must be satisfied to complete the inspection.



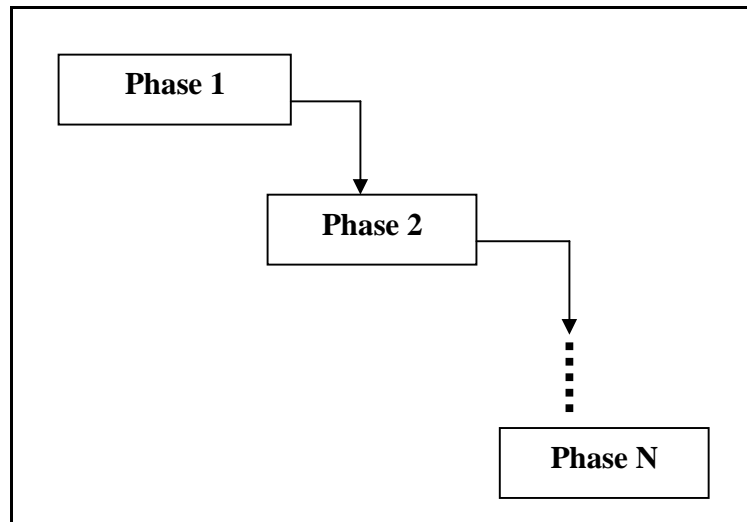
**Figure 5 : Gilb Inspection Process and Participants**

## 2.6 Phased Inspection

Phased Inspection technique is an enhanced inspection technique developed by Knight and Myers that tries to overcome weaknesses of other inspection techniques [Knight, Myers, 1993]. The goal of this technique is to produce defect-free, maintainable, portable and reusable software. This inspection aims to utilize computer support. Inspection process consists of an ordered set of phases (See Figure 6). Each phase is designed to ensure that the product hold some properties such as reusability, portability and compliance with standards.

There are two types of phase;

- Single-inspector phase: In this type of phase, a strict checklist is used. The phase cannot be completed until the product satisfies all checks.
- Multiple-inspector phase: This type of phase is designed for properties, which cannot easily be described by questions in single-inspector checklists.



**Figure 6 : Phased Inspection Process**

This inspection consists of three steps; examination, inspection and reconciliation. Documents are distributed to inspectors to examine. The product is then inspected individually by each inspector. This individual checking makes use of a checklist that is both application specific and domain specific. In the last step, reconciliation, inspectors meet to collate and compare their findings.

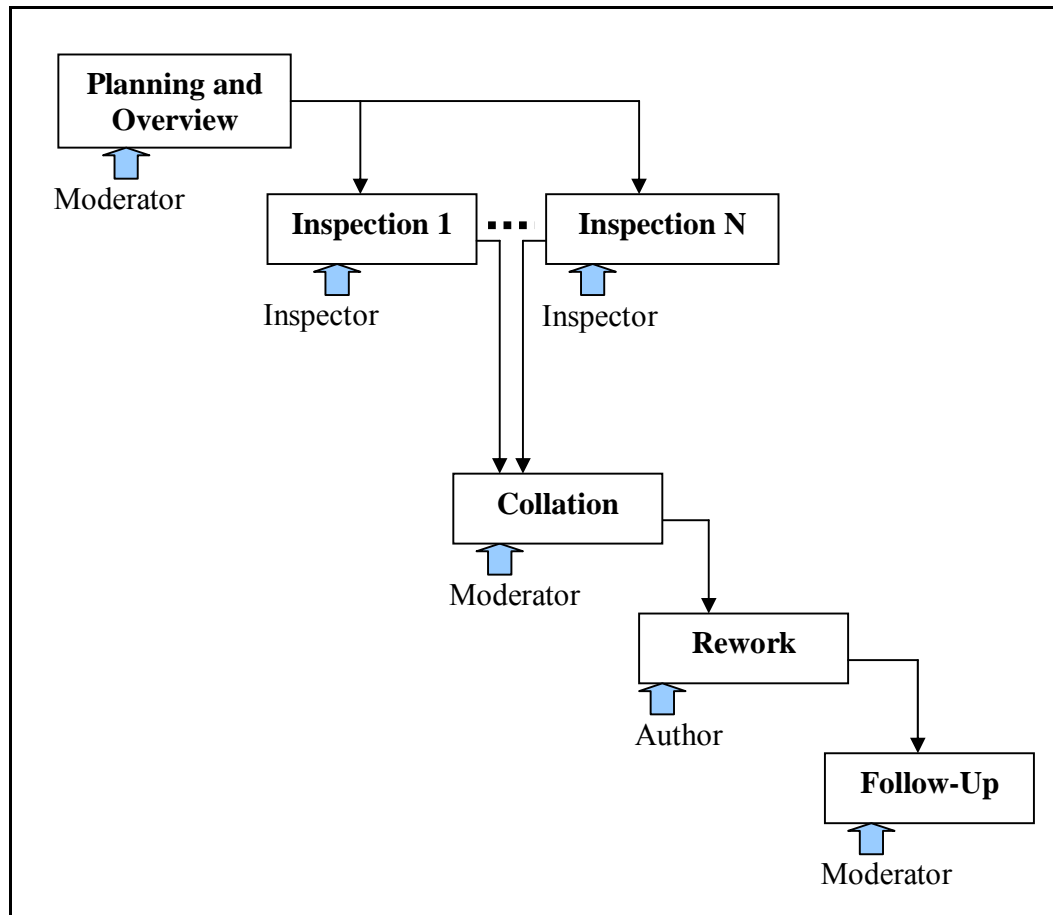
## **2.7 N-fold Inspection**

In the N-Fold inspection method, “N” numbers of parallel inspections are performed on the same artifact to improve effectiveness of the inspection by replicating it [Schneider, Martin, Tsai, 1992].

There may be some overlap between the defects found but the new defects can be found by independent teams. The technique was originally designed for user requirements documents, but it could be used for any other artifacts.



Inspection teams employ different techniques to improve overall inspection. For example, one team may use Fagan Inspection; another may use Gilb Inspection and so on.

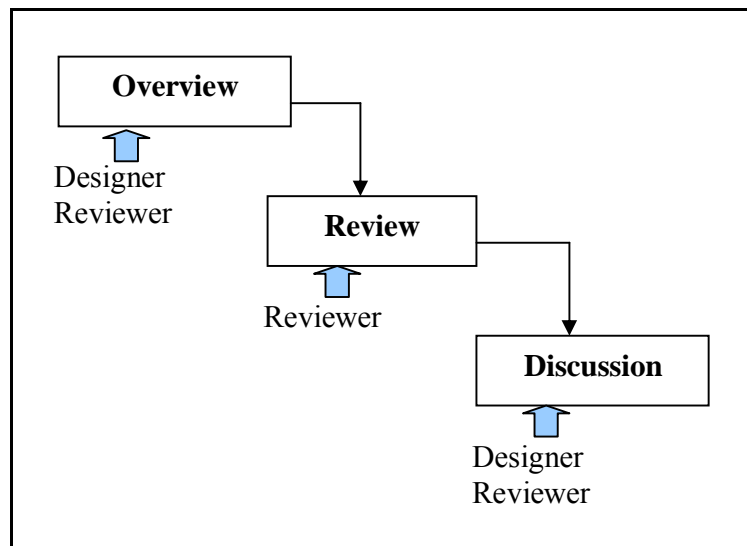


**Figure 7 : N-Fold Inspection Process and Participants**

This inspection process includes planning and overview, N-numbers of inspections, collation, rework and follow-up phases (See Figure 7). The inspection begins with planning and overview where the number of teams and other details relevant to inspection are decided. Then a number of inspection phases are performed by independent team following decided techniques. After each inspection, moderator collates the defects in the collation phase. A comprehensive defect list is produced and given to the author to rework on the artifact. In the follow-up phase, moderator ensures that all issues have been addressed.

## 2.8 Active Design Review (ADR)

The Active Design Review (ADR) process [Johnson, 1994b] was designed by Parnas and Weiss on the idea that instead of one review involving a large number of people, several smaller reviews are held, each one concentrating on different parts of the artifact for an effective review process.



**Figure 8 : Active Design Review Process and Participants**

There are two types of participants; reviewer and designer. Each reviewer has to be an expert on distinct area in order to take advantage of benefit from the review. The process includes overview, review and discussion phases [MacDonald, 1998] (See Figure 8). In the overview phase, designers present a design brief to reviewers, and tasks are assigned. Design of the artifact is reviewed in the review phase. Questionnaires are used in this phase in order to achieve a comprehensive review. In the discussion phase, designers read the completed questionnaires and meet with the reviewers to discuss raised issues in the review phase. When the reviewers and the designers agree on the issues, the review is complete and the designers make appropriate changes to the design document.

## 2.9 Code Reading

Code reading has been proposed as an alternative to formal code inspections [McConnel, 1993]. It is carried out informally by a small group of participants. In

code reading, the inspector simply focuses on reading source code and looking for defects, at an optimal rate, approximately 1,000 lines per day. The meeting is an optional activity. Code reading is in fact the individual defect detection activity in other software inspection techniques when the target materials are source code listings.

## **2.10 Two-person Inspection**

Bisant and Lyle [Bisant, Lyle, 1989] proposed that a two-person inspection is feasible. Its advantage is that it is less costly than the conventional Fagan Inspection. The inspection team consists of the author and one reviewer performing an individual review and a one-session inspection meeting.

## **2.11 Cleanroom Approach**

Cleanroom software engineering aims to produce a defect-free software product [Mills, Dyer, Linger, 1987]. It relies on formal software engineering methods and focuses on a team-oriented process. Team development process proceeds through formal specification, design and verification. In fact, the cleanroom approach is a software engineering approach where a major component is software review. The review is substantially subject to human activities. Authors of an artifact write code without compilers and they are not allowed to verify or execute the artifact. Thus, code is thoroughly inspected ensuring non-defective code.

## CHAPTER 3

# TOOL SUPPORT FOR SOFTWARE REVIEW

A number of computer tools have been developed to support software review to attempt to improve inspection efficiency and accuracy. Tools ranging from managing and delivering inspection documentation systems to complex collaborative systems have been developed.

According to Macdonald [Macdonald, 1998], there are two main types of tool; tools for paper-based reviews and tools for online review of documents.

### 3.1 Tools for Paper Based Review

#### **COMPAS:**

COMPAS is a multi-user document management system designed to support the software development process that appeared in 1984 [Barnard, Collicott, 1990]. The system allows documents related to planning, scheduling, monitoring, and testing to be stored and accessed in a timely manner.

COMPAS contains several major subsystems including an inspection and review subsystem. These subsystems are used to support inspection activities, including scheduling inspections, notifying participants, distributing materials, etc. Inspection statistics and metrics can be extracted from the system.

#### **Quality Group 4000:**

Actually, this nameless tool was used by Quality Group 4000 (QG4000) to support for software inspection. The tool is based on several simple UNIX tools.

Comments can be produced online via simple text files during reviews. A single command allows the moderator to collect all comments into a single file. The tool can then generate a paper report of these comments to be passed to the author. The tool can store data from multiple reviews.

#### **IPA:**

IPA (Inspection Process Assistant) mainly is used for online entering of defects in the products. IPA has a process model consisting of planning, individual preparation, meeting, rework, verification and finalization. IPA allows a database of available inspectors to be kept, and also allows information on documents to be stored. It supports production of defect lists and collection of some data related to inspection.

#### **Electronic Meeting Systems:**

Electronic meeting systems have been used as computer support for defect collection meetings in the software inspection process. They also provide real-time videoconferencing or audio conferencing for a team in a distributed environment. It is reported that these types of systems could significantly increase defect detection rate [van Genuchten et al., 1997].

### **3.2 Tools for Online Review**

#### **ICICLE:**

ICICLE (Intelligent Code Inspection in a C Language Environment) was designed to support activities in C and C++ code inspection, particularly the code inspection meeting [Brothers, et al., 1990]. The goal of the system is to reduce costs associated with inspection meeting. ICICLE employs a variety of techniques such as standardizing the interface among subtasks, providing statistics about defects discovered, and improving communication between subtasks. The source code is displayed in a large window with each line numbered. Inspectors can create annotations by using a Comment Proposal window. The presence of comments is indicated by letters on the left of line numbers. Users can click on the letters to view

the comments. ICICLE facilitates the code inspection meeting by synchronizing users' screens. During the meeting, the flow of activity is controlled and all participants are able to see new comments as soon as they are created. A teleconferencing facility is also available. When the inspection meeting is complete, ICICLE generates a list of all accepted defects to be given to the author of the product under inspection. A summary of the defects by type, class and severity is also generated.

### **InspeQ:**

InspeQ (Inspecting software in phases to ensure Quality) is a toolset developed by Knight and Myers to support their phased inspection technique. The technique was developed to allow the inspection process to be "rigorous, tailorable, efficient in its use of resources, and heavily computer supported" [Knight, Meyers, 1991]. The system provides five different displays. The inspectors use the "work product display" to view the document under inspection. "Checklist display" allows the status of each checklist to be displayed and modified. The inspection rationale and a detailed description is displayed in the "standards display". The "highlight display" helps in locating particular aspects of the document under inspection. InspeQ does not have support for group meetings.

### **CSI:**

CSI (Collaborative Software Inspection) is designed to support inspection of all software development products. [Mashayekhi et al., 1993]. It is based on Humphrey's inspection process. CSI supports both synchronous and asynchronous software inspection activities. Supported inspection activities include distribution of materials, individual inspection, defect reporting, inspection meetings, recording of action items, meeting reporting, etc. CSI provides a browser for viewing the document under inspection, which automatically numbers each line. In the inspection meeting, the document under inspection is made visible on a window on each inspector's screen. During inspection meeting, all participants can see the same artifacts simultaneously and communicate with others. The integrated defect list is available to the group, and new comments can be added during the synchronous

session. Participants access to information according to the permissions given them.

### **CSRS:**

Collaborative Software Review System (CSRS) is a customizable tool that supports different variants of the inspection process. Mainly it is developed to support FTArm (Formal Technical Asynchronous review method) inspection process [Johnson, 1994]. Customizability is accomplished using a process modeling language. This language has several facilities, including constructs for defining phases of the method, a construct for defining the role of each participant, and constructs to define the artifacts used during the inspection. The language can also be used to define the user interface, as well as to control the type of data analysis carried out by CSRS. E-mail notification feature is available for informing reviewers about newly created documents or comments. The tool does not support group meeting. The group review meeting must be held face-to-face in the traditional manner.

### **Scrutiny:**

Scrutiny is a collaborative inspection and review system that is configurable via ConversationBuilder protocols [Gintell et al., 1993]. It is a distributed system that can be used by geographically separated users. Inspection processes in Scrutiny try to model face-to-face inspections. This process consists of four stages; initiation, preparation, resolution, and completion. Support for various roles is provided. The inspection process is modified to maximize effectiveness of individual and team inspections. Inspectors can create annotations while reading documents. Inspection is completed when all annotations are resolved. Scrutiny automatically generates an inspection report containing all the relevant information about the inspection and its participants, details of the time spent by each participant in the inspection.

### **CAIS:**

Collaborative Asynchronous Inspection of Software (CAIS) is designed to be used for asynchronous inspection process [Mashayekhi et al., 1994]. This system uses CSI for displaying and annotating documents. The metrics such as number of comments per person, number of votes per person, time for individual defect

collection, total meeting time is collected.

### **Notes Inspector:**

Notes Inspector (NI) was built using Lotus Notes and implements an asynchronous model of inspection [Thompson, Riedl, 1995]. In this model, the inspection consists of an individual defect finding phase followed by an asynchronous meeting where participants discuss and vote on defects found. All participants vote whether to reject or accept an issue. This tool only permits the document to be read and annotated. Annotations can appear as additional lines either within the textual document, or as symbols next to the appropriate line. Data collection facilities are not available in the tool.

### **AISA:**

Asynchronous Inspector of Software Artifacts (AISA) is designed to allow asynchronous inspection of graphical documents, such as data flow diagrams and entity-relationship diagrams. The tool is based on the Mosaic Web client and supports a three stage inspection process; defect collection, defect correlation and inspection meeting. The tool is a Web version of CSI, CAIS, and Notes Inspector. Participants can vote to accept or reject defects proposed by others. A defect summary report is finally produced by the system.

### **TAMMi:**

TAMMi is a tool developed to support sharing of consistent information between designers and inspectors [Putala, Tervonen, 1997]. It supports a model that is based on goals (G), rules (R), checklists (C) and metrics (M); GRCM quality model. Each quality goal is broken down into a number of rules, which are in turn broken down to form checklists. Rules are used to guide software design, while checklists are used to help inspectors check for compliance with rules. TAMMi is designed to support both graphical and textual documents. Annotations can be marked directly on the document. It supports neither group meetings nor data collection.



### **HyperCode:**

HyperCode is a collaborative inspection tool that is essentially a set of “cgi” scripts used in conjunction with the Netscape Web browser plus an Application Programming Interface to two local configuration management systems [Perry et al., 1998]. The system supports; security, document preparation, process administration, document annotation, and archiving. E-mail notification is used to notify the users about their roles in an inspection. Users are provided with web ID's and passwords so that the system can control access to documents. Reviewers can check out the inspection package and make annotations to any portion of the artifact. Reviewers can also comment on the annotations made by other reviewers.

### **DCI:**

The Distributed Code Inspection (DCI) tool is designed to implement a distributed inspection process proposed by Doherty and Sahibuddin. It is a Web-based tool that supports both synchronous and asynchronous methods.

### **InspectA:**

InspectA is a tool that supports asynchronous inspection [Murphy, Miller, 1997]. It claims to overcome some weaknesses of AISA. After the individual inspection, the first phase, review is performed. The review phase is followed by another round of individual inspection, with all comment lists being available to each inspector. At the end of this phase, the moderator prepares a master list of comments to send to the author for repair. It supports only plain text documents. Meeting facilities does not exist in the tool; e-mail messaging is the only mechanism for supporting group processes.

### **WiP:**

Web inspection Prototype (WiP) is developed as a test-bed for examining the use of Web technology for a collaborative inspection environment. The inspection process model includes setup, individual inspection, peer inspection, public inspection, group inspection, consolidation and reporting. It is the first Web-based asynchronous software inspection tool that employs Java applets. Inspectors are

required to login before performing inspection. Each document to be inspected is presented as a dynamically created HTML page. It has an annotation tool allows users to make comments on inspected document.

### **ASSIST:**

ASSIST (Asynchronous/Synchronous Software Inspection Support Tool), was developed to be a flexible software inspection tool that both supports synchronous and asynchronous inspection activities [Macdonald, 1998]. Its key feature is the process modeling capability allowing different software inspection techniques to be used for controlling the inspection process. It has a flexible document type system, allowing new document types and their associated browsers to be added as required. The tool collects the time spent in inspection, the order in which lines of code are inspected, and the time each defect was found.

### **CASCADE:**

CASCADE (Computer Augmented Support for Collaborative Authoring and Document Editing) has been developed to augment commenting and review on electronic documents [Sapsomboon, 1999]. It has a number of similar features to those proposed in asynchronous software inspection tools, such as ICICLE, CSRS, WiP, AISA, and ASSIST. In addition, CASCADE offers a number of notable features including, color-coded labeling, flexible comment classifications, multiple windows, and navigational tools. With minor reorganization and a few enhancements, CASCADE can be modified to serve as an asynchronous software inspection tool.

### **IBIS:**

IBIS (Internet-Based Inspection System) is a Web-based inspection tool that support both synchronous and asynchronous inspection. System architecture is based on client and server sides. On the client side, the system uses common Internet-based application clients such as browsers, email readers, and -optionally- an audio-video plug-in. On the server side, the system uses only technologies based on Internet standards or web standards It does not require any Database Management System for

the data repository. All structured and persistent data are stored as XML files. It does not require any workflow management system or groupware platform. All groupware functionalities are developed from dynamic web pages on the basis of scripts and server-side components, when available; event notification is achieved by using the email service of a server side component, which transfers generated messages to an outgoing SMTP (Simple Mail Transport Protocol) server.

#### **XATI:**

XATI (XML Annotation Tool for Inspection) was developed to support “virtual software Inspections” [Hedberg, Harjumaa, 2001]. Emphasis in the development of this new tool has been placed on ease of use and deployment in addition to interoperability. XATI is based on experiences gathered during the development and use of our earlier tools, such as WiP. The XATI tool uses Mozilla as a generic application environment and as a viewer for a variety of document types. Integration with other software development tools and repositories is carried out with an interoperability framework called Xfi (eXtensible framework for interoperability).

#### **wbART:**

wbART (A Web-based Tool for Formal Technical Asynchronous Review) is a Web-based tool that provides automated support for formal technical asynchronous review of work products [Yanbaş, 2002]. It supports display and browse of multiple document formats. However, four widely used document formats are implemented to be displayed with review annotations attached on them; Microsoft Word document, Microsoft Excel spreadsheet, Microsoft PowerPoint presentation and Rational Rose models. Review roles defined within the tool are System Administrator, Moderator, Author and Reviewer. The tool does not support online annotation of issues, only supports displaying annotations in the documents. It collects different types of metrics such as number of materials reviewed, percentage of review materials accepted as is, and number of issues raised in all reviews.

#### **EasyReview:**

EasyReview is a workflow based asynchronous review system that allows

geographically distributed review teams to inspect software artifacts. It supports defining new review processes according to the needs. Four widely used document formats are supported for online issue annotation. Review participants are informed via automatic e-mails about the status of the review process. The system does not have a meeting support. Detailed specifications of EasyReview are given in the following chapter.

### **3.3 Comparison of Tool-Based and Paper-Based Software Inspection**

When moving from paper-based to even the simplest tool-based inspection, there are a number of advantages. Advantages of the tool-based inspections over paper-based inspections are as follows [Macdonald, Miller, 1997]:

- The tool can automatically ensure that the most up-to-date version of the document is used, especially if integrated with any version control system in use.
- It is possible to provide cross-referencing facilities for documents, from simple search facilities to sophisticated code browsers.
- Use of electronic documents avoids the cost of printing multiple copies of documents for each inspector.
- Online storage of defects allows defects to be linked to the position in which they occur in the document, allowing them to be accessed easily.
- Online storage of defects also removes any problems caused by unintelligible handwriting, which can be problematic in paper-based inspection.
- When a synchronous group meeting is held, defects can be easily shared between inspectors, since the tool simply shows everyone the same text. The defect can then be discussed, and any required changes made online by the scribe.
- Updated defects can be propagated to all participants easily.
- Finally, the online defect list allows a voting mechanism to be used to

determine the validity of a defect.

On the other hand, the move to a computer-based process does create several drawbacks. These are as follows:

- To train an inspector in the use of the tool requires time. If the tool is complex, this may reduce the effectiveness of the inspection.
- Many people are far slower at typing than handwriting, which may slow down the process of writing defects.
- Amount of screen space is limited in the computer tools.
- Reading text from a screen may be difficult and tiring.
- Finally, paper documents have been in use far longer than computers, and there is a natural tendency for users to prefer paper.

### **3.4 Comparison of Existing Tools**

Table 1 summarizes tools and their features in chronological order. Deficiencies in the existing tools are as follows:

- Some of the tools support only a single inspection process. They do not provide flexibility to modify the inspection process according to the needs. For example, Scrutiny only supports a single, hard-coded inspection process. InspeQ, developed by Knight and Myers supports their phased inspection process.
- Some of the tools only support inspection of plain text documents. ICICLE, Scrutiny, CSI, CAIS and HyperCode use the same technique of displaying the document with each line numbered. Annotations can then be made which are linked to an individual line. In InspeQ and InspectA comments are completely separate from the source document. AISA and TAMMi allow annotation of graphical documents. This lack of support for different document types is one of the major shortcomings of existing tools, and must be addressed if inspection support tools are to become a standard feature of software development

support environments [Macdonald, 1998].

- Classification of annotations is important to find out the most frequently occurring defect types. ICICLE, CSI, CAIS, AISA, Scrutiny, TAMMi, CSRS, InspectA, WiP and wbART allow classification of annotations. The other tools only allow their creation or deletion. This limits the scope for collection of defect type metrics.
- Support for distributed meetings is only relevant to the synchronous inspection tools. CSI utilizes an audio channel only. Scrutiny also supports the use of an audio channel, in addition to its discussion and messaging facilities. DCI provides textual discussion via chat and e-mail. ICICLE lacks these facilities and is designed to be used when the inspection meeting takes place in one room with all inspectors present. InspeQ and TAMMi lack such facilities. [Macdonald, 1998]
- Some of the tools are platform dependent. For example, ASSIST allows users to define and support their own inspection processes, but the system is built over its own document database. CSRS allows flexible process definition, but relies heavily on supporting technologies emacs and Unix.
- Some of the tools do not have a comprehensive metric collection capability. Gathering metrics is important to provide an overall perspective on review activities.

Tool/ Features	Year	Reference	Linked Annotation	Defect Classification	Meeting Support	Web- based	e-Mail Notification	Process Flexibility
ICICLE	1990	Brothers et al.	Yes	Yes	Yes			
InspeQ	1993	Knight, Myers						
CSI	1993	Mashayekhi et al.	Yes	Yes	Yes		Yes	
CSRS	1993	Mashayekhi et al.	Yes	Yes			Yes	Yes
Scrutiny	1993	Gintell et al.	Yes	Yes	Yes			
CAIS	1994	Mashayekhi et al.	Yes	Yes				
NI	1995	Thompson, Riedl	Yes			Yes	Yes	
AISA	1996	Stein et al.	Yes	Yes		Yes		
TAMMi	1996	Tervonen	Yes	Yes				
DCI	1997	Doherty, Sahibuddin			Yes			
HyperCode	1997	Perpich et al.	Yes			Yes		Yes
InspectA	1997	Murphy, Miller		Yes		Yes	Yes	
ASSIST	1997	MacDonald, Miller	Yes	Yes		Yes	Yes	Yes
Wip	1998	Harjumaa, Tervonen	Yes	Yes		Yes		
CASCADE	1998	Sapsomboon	Yes	Yes				
IBIS	2000	Caivano et al.	Yes	Yes		Yes	Yes	
XATI	2001	Hedberg, Harjumaa	Yes	Yes		Yes		
wbART	2002	Yanbaş	Yes	Yes		Yes	Yes	
EasyReview	2004	Çifci	Yes	Yes		Yes	Yes	Yes

Table 1 : Comparison of Existing Tools

## CHAPTER 4

# SOFTWARE REQUIREMENTS AND DESIGN

### 4.1 Introduction

This chapter describes the Software Requirements Specifications (SRS) and the design of EasyReview. Software requirements are prepared in accordance with IEEE STD. 830-1998, IEEE Recommended Practice for Software Requirements Specifications. Tailored version of the Standard is applied. An object-oriented approach is used in the design of the system. Object Management Group's (OMG) Unified Modeling Language (UML) is used as the design notation.

#### 4.1.1 Scope

EasyReview shall be a workflow based online review system that allows geographically distributed review teams to inspect software artifacts such as requirement specifications and design documents. Multiple users must be able to access the same artifact at the same time.

It shall have a workflow system to define and support different kinds of review processes. Workflow system allows authorized users to define new review process according to an organization's needs.

EasyReview shall support online annotation of issues or defects in the document formats of Microsoft Word document, Microsoft Excel Spreadsheet, Microsoft PowerPoint presentation and Rational Rose model. The concept of inspection roles must be supported such as moderator, inspector, author and other roles. The system shall allow defining additional roles when needed.



The tool should be able to provide metrics collection, such as comments, and defects.

Review participants must be able to make comments and classify them into categories, such as general comment, warning, etc. A default set for the classifications will be provided and will be user definable. Additionally, these comments should be able to be classified according to the criticality such as “High”, “Medium”, or “Low”.

Defects generated from multiple reviewers can be merged into a single report based on any of the defined defect categories. In addition, efforts of the review participants on the review tasks shall be merged into a report.

## **4.2 Product Perspective**

EasyReview will be a Web based tool that is accessible from a network such as an internal corporate network –intranet- or internet. The client can access the server, running Tomcat Application Server via a Web browser using HyperText Transfer Protocol (HTTP) version 1.1.

There will be two different graphical user interfaces (GUI) which support administrative operations and review related operations. Hence, the system will support two different types of user groups; “System Admin” and “Ordinary User”. “System Admin”’s can perform administrative operations such as defining users, roles, workflows and projects. “Ordinary User”’s can perform review related activities such as uploading or downloading review materials, adding issues about review materials and taking reports about defects and efforts.

The system will be developed in Java programming language. The Java Server Pages (JSP), JavaScript and Rational Rose Scripts, HTML (HyperText Markup Language) and XML (eXtensible Markup Language) will be used during development. To automate application specific functions, ActiveX Controls will be created. Software products required to implement the system are shown in Table 2.

**Table 2 : Software Products Required Implementing the System**

<b>Product Name</b>	<b>Version</b>	<b>Description</b>
JBuilder Enterprise	9.0.125.0	The integrated software development environment to develop the system.
Jakarta Tomcat	4.1	Application server container for JSP and servlets.
Rational Rose Enterprise Edition	2001.03.00	Modeling tool to be used for requirement and design models.
Microsoft Office	XP (2002)	Software used to work on Word, Excel and PowerPoint review materials.
Internet Explorer	5.0	The Web browser to connect to the system.
MySQL	3.0.8	The Database Management System (DBMS) to perform database transactions.
Microsoft FrontPage	2002	To prepare HTML Web pages.

In addition to the above software products, software libraries will be used to perform some activities required in the system. The libraries used are as follows; Java Development Kit (JDK) 1.4.1 for java runtime environment, Tomcat Servlet 4.1 for servlet container, Download4j 1.2 to be able to download review materials, UploadBean 1.4 to be able to upload review materials, and MySQL 3.0.8 Java Database Connectivity (JDBC) driver to connect to the database management system.

### **4.3 Constraints, Assumptions and Dependencies**

The following constraints, assumptions and dependencies have been made for the requirements specification.

It is assumed that:

- Users of EasyReview have basic the knowledge of using a Web browser.
- A Web browser, Microsoft Word, Microsoft Excel, Microsoft PowerPoint and Rational Rose Enterprise Edition is available on the client machine.
- The web browser on the client machine is Java Virtual Machine (JVM)

enhanced.

- The web browser on the client machine is JavaScript compliant in order to execute the scripts embedded in the HTML code.
- The Web browser on the client machine supports ActiveX control and security settings are set to allow ActiveX controls to run on the machine.
- The client computer has MySQL ODBC 3.51 driver installed on it.

## **4.4 Specific Requirements**

This section presents all of the specific requirements of EasyReview software. Emphasis is placed on the functional requirements, which are explained as groups of related use cases.

### **4.4.1 Log on to the System**

The user opens a web browser and enters the URL to connect the system. The system displays a Web page asking the user for his username and password pair. The user enters his username and password. System does not display the password typed, but displays “\*” for each character entered. Then, the system checks the username and password for authentication and authorization. The system gives an error message if authentication fails. If the user is authenticated correctly, the system opens a new page according to the user’s identity. If the user is an administrator of the system, the Web page for administering the system is opened. Likewise, if the user is an “ordinary user”, the Web page to conduct review operations is opened.

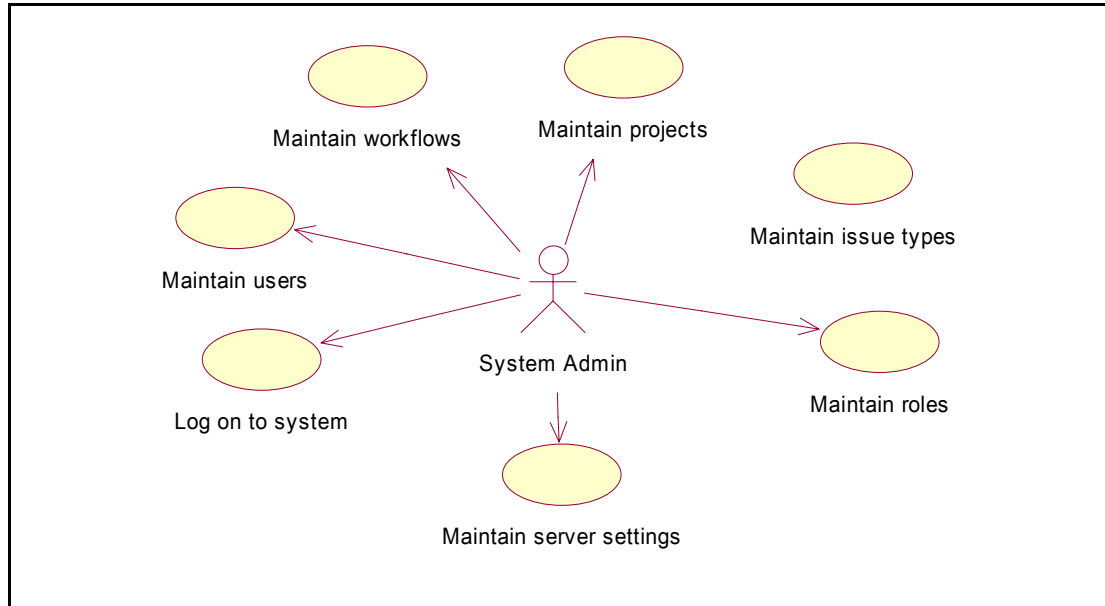
### **4.4.2 Operations for Administering the System**

Operations for administering the system are shown in Figure 9.

#### **4.4.2.1 Maintain server settings:**

The user can maintain “Web Server”, “Mail Server” and “Connection String” settings via this option. When the user selects this option, the system displays current settings. The user can change these values by entering the new values in the text

boxes next to the labels. By pressing the “Change” button, the user activates the new settings. Then the system displays a Web page containing the result whether the operation is successful.



**Figure 9 : Operations for Administering the System**

#### 4.4.2.2 Operations for maintaining roles:

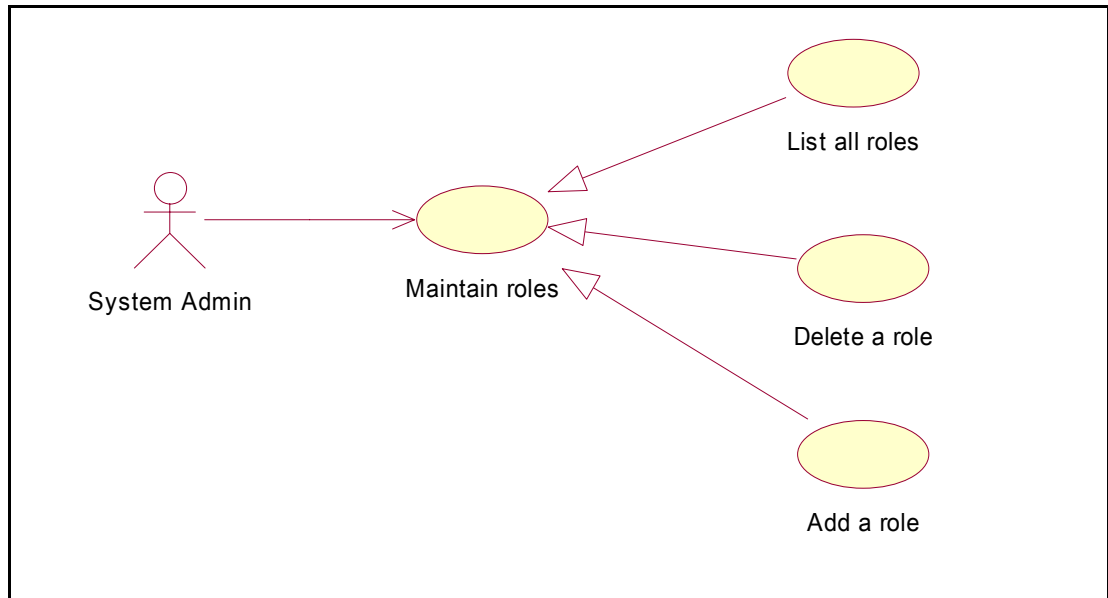
As stated in Chapter 2, there are different kinds of roles that are responsible for a phase of a review process. The system allows administrators to define new roles or delete existing roles according to the needs (See Figure 10). The system has built-in roles including “Moderator”, “Author”, “Reviewer”, “Inspector”, “Producer”, “Coordinator”, “Leader”, and “Checker”. The roles in the system do not have built-in capabilities. Capabilities of the roles are assigned in the workflows. The user can add a new role, delete an existing role or list existing roles by selecting the related option.

- Add a new role:

If a new role is needed for review activities, it can be defined by selecting this option in the system. The user enters the name of the new role and presses the “Add” button. The system checks if the role has been previously added to the database. Then a Web page that contains result of the operation is displayed by the system.

- Delete an existing role:

The system allows administrators to delete an existing role when that role is not needed. The user selects a role to be deleted within listed roles and then presses the “Delete” button. Finally, a Web page that contains the result of the operation is displayed by the system.



**Figure 10 : Operations for Maintaining Roles**

- List all existing roles:

When the user selects this option, the system lists the roles identified in the system. User can see the role names and can delete a role by pressing the icons next to each role.

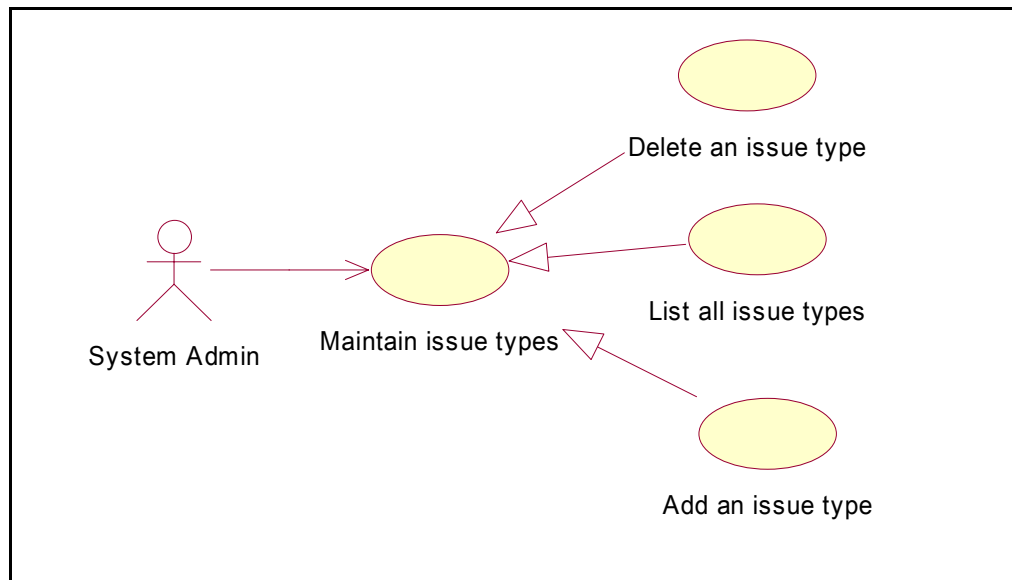
#### 4.4.2.3 Operations for maintaining issue types

The system allows administrators to define new issue types or delete existing issue types according to the needs (See Figure 11). The system has built-in issue types including “Clarity”, “Cosmetic”, “Extra”, “Inconsistent”, “Missing”, “Performance”, “Style”, “General”, “Wrong”, and “Other”. The user can add a new issue type, delete an existing issue type or list existing issue types by selecting related option.

- Add a new issue type

When a new issue type is needed for review activities, it can be defined by

selecting this option in the system. The user enters the name of the new issue type and presses the “Add” button. The system checks if the issue type has been previously added in the database. Next, a Web page that contains result of the operation is displayed by the system.



**Figure 11 : Operations for Maintaining Issue Types**

- Delete an existing issue type

The system allows administrators to delete an existing issue type when that issue type is not needed. The user selects an issue type to be deleted within listed issue types and then presses the “Delete” button. Finally, a Web page that contains result of the operation is displayed by the system.

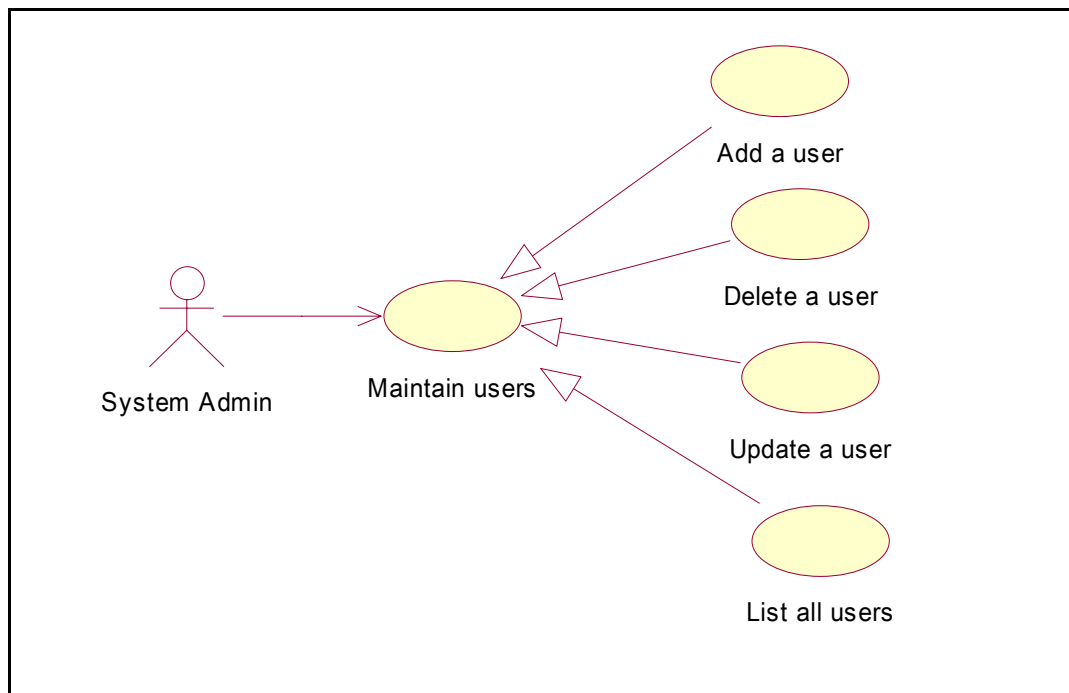
- List all existing issue types

When the user selects this option, the system lists the issue types identified in the system. User can see the issue type names and can delete an issue type by pressing the icons next to each issue type.

#### 4.4.2.4 Operations for maintaining users

The system allows administrators to add a new user, modify or delete an existing user, and list all the users that are already defined in the system (See Figure 12). There are two different types of the users in the system; “System Admin” and

“Ordinary User”. An administrator can define new administrators who are responsible for administering the system.



**Figure 12 : Operations for Maintaining Users**

- Add a new user

When the administrator wants to add a new user to participate in the review activities, he selects this option. The administrator enter username, name, surname, e-mail of new user, he gives a password and confirms it, and assigns it to a user group; “System Admin” or “Ordinary User”. After pressing the “Add” button, system checks if the password and confirmed password are identical and if the username already exists in the system. The system adds the user if all controls are satisfied. Finally a Web page that contains result of the operation is displayed by the system.

- Delete an existing user

The system allows administrators to delete an existing user when that user is not needed anymore. The user selects a user to be deleted within listed users and then presses the “Delete” button. Finally, a Web page that contains result of the operation is displayed by the system.

- Modify an existing user's information

The system allows administrators to modify information of an existing user when needed. The administrator selects a user within listed users and then presses the “Modify” button. The system fetches the selected user's information and displays in a Web page. The administrator updates information in displayed fields and presses “Modify” button. The system updates the user record and displays a Web page that contains result of the operation.

- List all existing users

When the user selects this option, the system lists the users defined in the system. User can see the usernames and can delete or modify a user by pressing the related icons next to each username.

#### 4.4.2.5 Operations for Maintaining Workflows

The system allows administrators to add a new workflow, customize or delete an existing workflow, and list all the workflows that are already defined in the system (See Figure 13). Review processes can be defined by workflows.

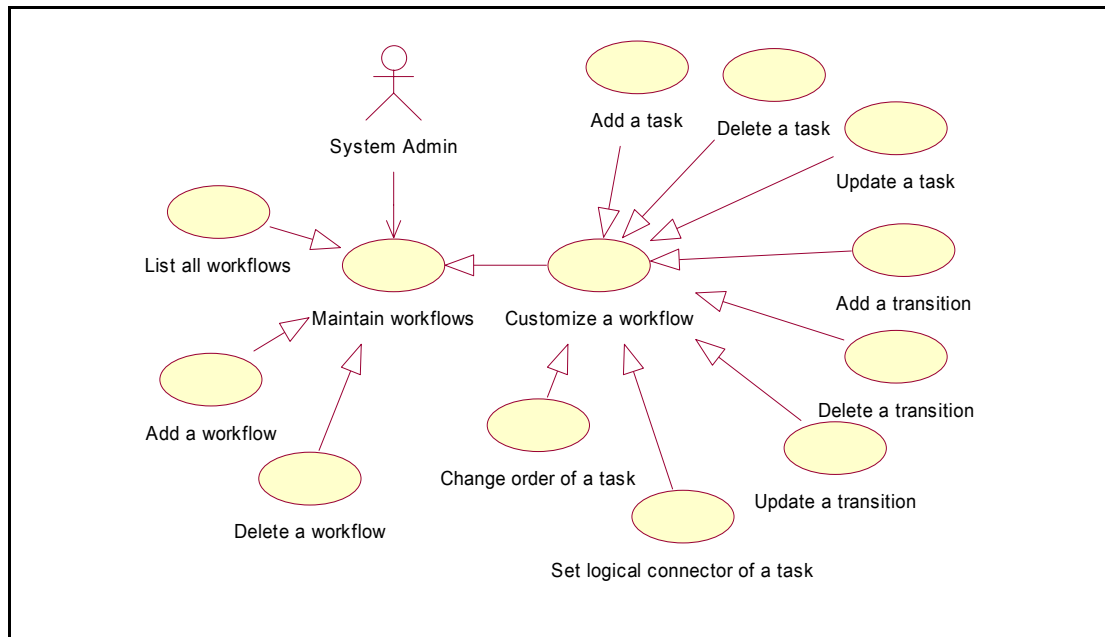
- Add a new workflow

When the administrator wants to add a new workflow, he selects this option. The administrator enter workflow name and presses “Add” button. Then the system checks if the workflow name already exists in the system. The system adds the workflow name if it does not exist. At the end, a Web page that contains result of the operation is displayed by the system.

- Delete an existing workflow

The system allows administrators to delete an existing workflow when that workflow is not needed anymore. The administrator selects a workflow to be deleted within listed workflows and then presses the “Delete” button. Finally, a Web page that contains result of the operation is displayed by the system.





**Figure 13 : Operations for Maintaining Workflows**

- **Operations for Customizing Workflows**

When the administrator selects customization of a workflow option, the system lists all defined workflows in the system. The administrator selects a workflow from the list and presses “Customize” button. The system displays the transitions, tasks and associated values about the transitions and tasks. In the opened Web page, administrator can do following operations.

- Add a task to the workflow
- Delete a task from the workflow
- Update an existing task in the workflow
- Add a transition to the workflow
- Delete a transition from the workflow
- Update an existing transition in the workflow
- Change order of the tasks in the workflow
- Set logical connectors of the tasks
- Add a task to the workflow

To be able to add a task to a workflow, administrator has to follow two steps, which are contained within two different Web pages. Administrator presses “Add a Task” icon and then a Web page containing “Step 1” appears.

Step 1: In this step, administrator enters a name for the task and determines a notification message sent to review participants when the task starts. Then he has to determine three attributes of the task.

1. “Are issues added/removed/updated in this task?”: If administrator selects “Yes” for this option, this means that, when this task is performed in a review, review participants who are responsible for this task can add issues to the reviewed artifacts, remove existing issues about those artifacts and update issues about review artifacts.
2. “Are the issues criticized in this task?”: When the administrator selects “Yes” for this option, this implies that when this task is performed in a review, review participants who is responsible for this task can accept or reject issues that are entered about review artifacts. This helps moderators to clean wrong, unintelligible, and identical issues.
3. “Are the statuses of the issues updated in this task?”: When the administrator selects “Yes” for this option, this implies that when this task is performed in a review, review participants who are responsible for this task can mark issues as “Resolved” or “Unresolved”. This helps authors of the review artifacts to indicate whether he resolved the issues entered.

Step 2: In this step, administrator can select roles that will be responsible for the task. The system displays available roles in a box. The administrator selects roles and presses “Add” button. Then the system adds selected roles to this task’s role list and removes those roles from available role list. Administrator can remove the roles from the task’s roles list by selecting those roles and pressing “Remove” button. The system allows administrator to navigate to Step 1 or add the task by pressing “Add” button. A message is displayed about the result of this operation whether it is successful or not.

- Delete a task from the workflow

The administrator can delete a task by pressing “Delete” icon next to each task. The system displays a message for confirmation. If the administrator confirms, task is deleted from the workflow. Transitions that have links to the deleted task are deleted automatically.

- Update an existing task in the workflow

The system allows administrators to modify information of an existing task when needed. The administrator selects a task within listed tasks and then presses the “Modify This Task” icon next to the task. The system fetches the selected task’s information and displays in a Web page. The administrator updates information in displayed fields in three steps and presses “Modify Task” button. The system updates the task information and displays a Web page that contains the result of the operation.

- Add a transition to the workflow

The administrator can add a transition to the workflow by pressing “Add a Transition” icon on the workflow customization page. If there are at least two tasks in the workflow a Web page is displayed for adding a transition. Administrator gives a name for the transition and selects “From task” and “To task” tasks from the tasks lists. Then he presses “Add” button. The system checks whether this operation is valid or not. If the following controls are satisfied, the system adds the transition to the workflow.

1. Transition name must be unique in a workflow.
2. Only one transition is allowed between two tasks in the same direction.
3. The transition directed TO the first task is not allowed.
4. The transition starts FROM the last task is not allowed.
5. The transition from a task cannot be directed to the same task.

- Delete a transition from the workflow

The administrator can delete a transition by pressing “Delete This Transition” icon next to each transition in the workflow customization page. The system displays a message for confirmation. If the administrator confirms deleting, transition is deleted from the workflow.

- Update an existing transition in the workflow

The administrator selects a transition within the listed transitions and then presses the “Modify This Transition” icon next to the transition. The system fetches the selected transition’s information and displays in a Web page. The administrator updates information in displayed fields “Modify” button. The system updates the transition information and displays a Web page that contains the result of the operation.

- Change order of the tasks in the workflow

The order of the tasks can be changed by the administrator by pressing “Change Order of the Tasks” icon in the workflow customization page. The system then displays the tasks in the current order. The administrator selects a task and presses “Up” or “Down” buttons to change the order of the selected task. Finally, the administrator presses to the “Done” button to make the new settings effective.

- Set logical connectors of the tasks

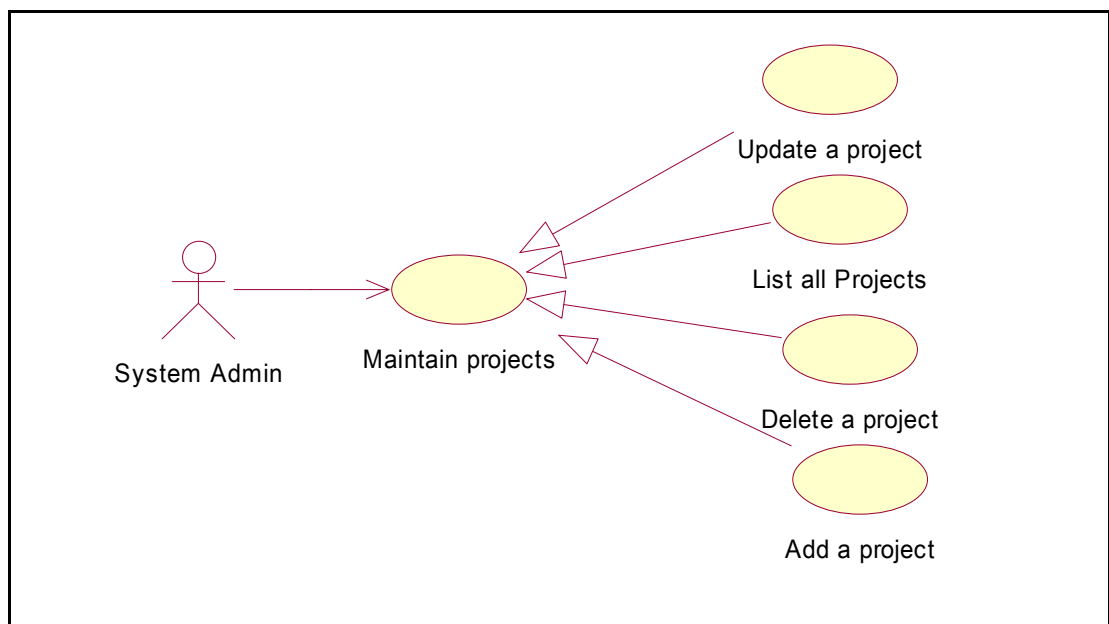
The system has two logical connectors; “AND” and “OR”. If a task has an “AND” logical connector, this means; when this task is finished, ALL of the connected tasks will be active after finishing the mentioned task. On the other hand, If a task has an “OR” logical connector, this means that; if this task is finished, ONE of the connected task will be active after finishing mentioned task (Active task is selected by review participant). The administrator can assign logical connectors to the tasks by pressing “Set Logical Connectors” icon in the workflow customization page. Then the system displays the tasks with their current logical connectors. Administrator selects a task and presses “AND” or “OR” button to set a new logical connector for the task. Finally, the administrator presses to the “Done” button to make the new settings effective.

- List all existing workflows

When the user selects this option, the system lists the workflows defined in the system. User can see the workflow names and can delete or customize a workflow by pressing the related icons next to each workflow.

#### 4.4.2.6 Operations for Maintaining Projects

The system allows administrators to add a new project, modify or delete an existing project, customize a review and list all the projects that are already defined in the system (See Figure 14). A project can contain one or more reviews.



**Figure 14 : Operations for Maintaining Projects**

- Add a new project

This option is selected by the administrator to add a new project. Firstly, the administrator enters a project name. The system lists all the users in the system and allows administrator to add any user to the project. Administrator selects users from the list and presses “Add” button to make them the project’s users. He presses “Remove” button after selecting users to be removed if needed. Then the administrator presses “Next” button to go to the next step of adding a new project. In step 2, he can select and upload supplementary materials in the project scope. All participants in all reviews in the project can see and download these supplementary

materials. Typical examples are; standards and checklists. The administrator presses the “Save” button finished all the works and then the system checks if the project name already exists in the system. The system adds the project name if it does not exist. At the end, a Web page that contains result of the operation is displayed by the system.

- Delete an existing project

The system allows administrators to delete an existing project when that project is not needed anymore. The administrator selects a project to be deleted within listed projects and then presses the “Delete” button. Finally, a Web page that contains result of the operation is displayed by the system.

- Update an existing project

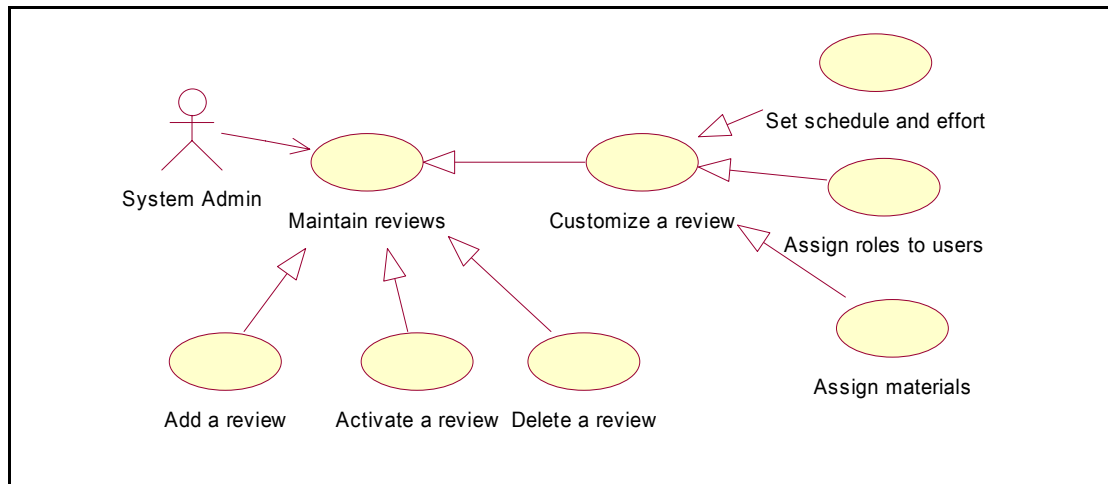
The system allows administrators to modify information of an existing project as needed. The administrator selects a project within listed projects and then presses the “Modify” button. The system fetches the selected project’s information and displays in a Web page. The administrator updates information in displayed fields in two steps and presses “Modify Project” button. The system updates the project information and displays a Web page that contains result of the operation.

- List all existing projects

When the user selects this option, the system lists the projects defined in the system. User can see the project names and can delete or modify a project, or customize the reviews of a project by pressing the related icons next to each project.

#### 4.4.2.7 Operations for Maintaining Reviews

The system allows administrators to add a new review to an existing project, customize or delete an existing review in a project, and activate reviews in a project (See Figure 15). If the administrator wants to administer reviews, selects “Review Administration” option and the system lists the projects in the system. Then he selects project to maintain its reviews. As stated before, a project can contain one or more reviews. “Start Date”, “And Date” and “Status” (Active, Inactive or Finished) of the reviews in selected project are displayed by the system.



**Figure 15 : Operations for Maintaining Reviews**

- Add a new review

When the administrator wants to add a new review, he presses “Add a Review” icon on the page. The administrator enter review name and selects a workflow to conduct this review. He presses “Add” button and then the system checks if the review name already exists in that project. The system adds the review name if it does not exist. At the end, a Web page that contains result of the operation is displayed by the system. The review is ready for assigning users, materials and time schedule for workflow tasks.

- Operations for Customizing Reviews

When the administrator selects customization of a review option, the system displays information about the review. At first, workflow tasks and associated roles of those tasks are displayed. In the opened Web page, administrator can assign users to the roles and set time schedule and effort for each task in the workflow.

- Assign roles to users

The system lists the project users and available roles in the workflow of the review. Initially, the administrator assigns users to the review from the project users. Next, he assigns roles to the selected review users. Finally he presses the “Save” button. If all the roles assigned to at least one user, the system saves the roles and associated users.

- Assign materials

Administrators can determine whether material uploading is allowed into the system during a review. There are three different options for material types. These are as follows:

1. Review Material: These types of materials are subject to review. In other words, review participants look into these materials and try to find out defects in them.
2. Supportive Material: These types of materials are supplementary ones that help review participants when they perform the review activities. Typical examples are coding standards, IEEE standards, articles about review activities, checklists and so on.
3. Work Result Material: In a review, it may be needed to upload a material about a review activity. Filled forms, filled checklists can be uploaded. Issues can be written in a Word document and that document can be uploaded and so on. By providing this option, the system provides much flexibility to the review participants to upload any type of material about the review activities.

The administrator gives a name for the review material and selects the type of the material.

- Set time schedule and effort

The administrator can set a schedule for the review and specify efforts for each task in the workflow of the review. The system displays the workflow tasks and allows the administrator to assign start and finish dates and effort values for each task. After pressing the “Save” button, the system checks whether starting and finishing dates are valid. Finally, a message is displayed by the system regarding the result of the operation.

- Delete an existing review

The system allows administrators to delete an existing review when that



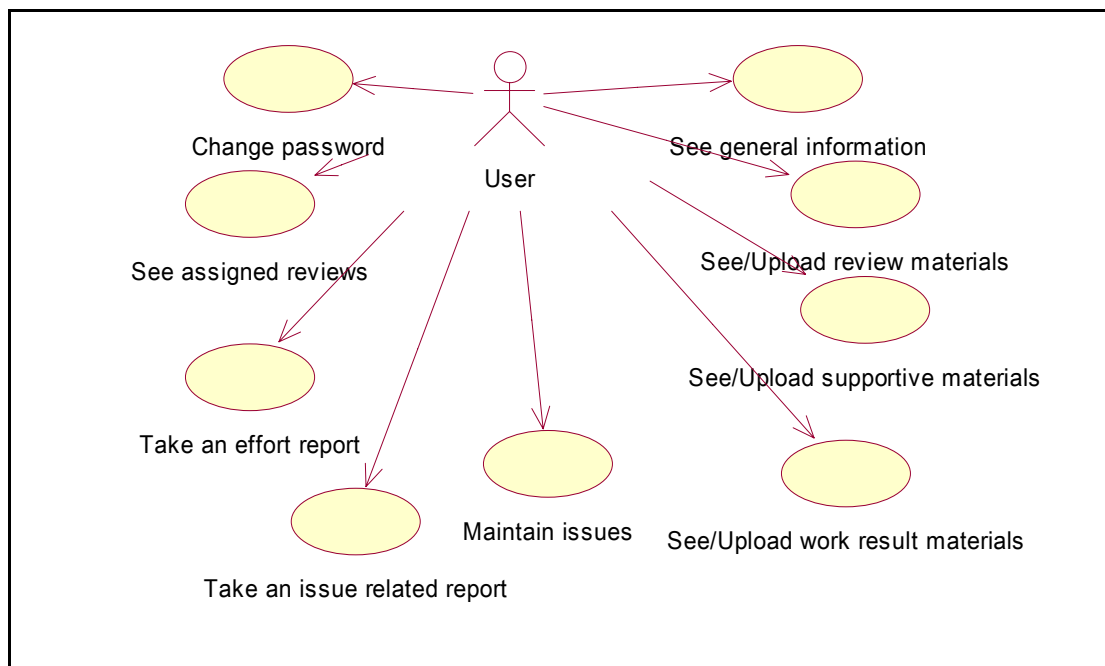
review is not needed anymore. The administrator selects a review to be deleted within listed reviews and then presses the “Delete This Review” icon next to each review. The system displays a message for confirmation. If the administrator confirms deleting, review is deleted from the project. Finally, a Web page that contains result of the operation is displayed by the system.

- Activate a review

After completing all the steps to define a review, administrator can activate the review. The system checks whether information of the review is complete. If all controls are satisfied, the review is activated; i.e. started. After activating a review, the system sends e-mail notification to all of the review participants.

#### 4.4.3 Operations for Conducting Review Activities

These operations are available to the “Ordinary Users” of the system (See Figure 16). The system allows these types of users to perform the following activities.



**Figure 16 : Operations for Conducting Review Activities**

#### 4.4.3.1 Change user password

When a user wants to change his password, he can select this option. The system asks for the old password, the new password and confirmation of the new password. After filling out all the fields, the user presses “Change” button. If old password is correct and new passwords are identical, the system changes the password of the user. At the end, a Web page containing the result of the operation is displayed by the system.

#### 4.4.3.2 See assigned reviews

A user selects this option to see the reviews that are assigned to him. The system displays project name, review name, and role of the user in the review. The user can select a review from the list to start reviewing.

#### 4.4.3.3 See general information about a review

When the user selects a review, general information about the review is displayed by the system. After entering this Web page, the user can see the general information whenever he wants by clicking the related option. The system displays the following information in this view:

- Active task (the task that is currently performed in the review), start and end date of the active task.
- Active task’s users and whether they finished their jobs.
- Tasks that are assigned to the user in the review and start and end dates of them.

#### 4.4.3.4 See and upload review materials

The user can see the uploaded review materials in the review. Material names and associated file name are displayed by the system. For example, material name may be “Software Requirements Specifications Document” and associated file name may be “SRS\_V1.doc”. User can see and download materials listed by the system. Thus, the system allows users to upload review materials if that review is designated to allow uploading review materials. In this situation, the system lists the material

names and whether uploading them is mandatory or not. The user uploads requested materials to the system. He can replace the uploaded materials until he finishes his job by selecting “Finish This Task” option. If uploading review materials is not allowed for the review, a message containing a warning is displayed to the user.

#### 4.4.3.5 See and upload supportive materials

The user can see the uploaded supportive materials in the review. Material names and associated files name are displayed by the system. For example, material name may be “Coding Standards” and associated file name may be “code\_std.pdf”. the user can see and download materials listed by the system. Thus, the system allows user to upload supportive materials if that review is designated to allow uploading supportive materials. In this situation, the system lists the material names and whether uploading them is mandatory or not. The user uploads requested materials to the system. He can replace the uploaded materials until he finishes his job by selecting “Finish This Task” option. If uploading supportive materials is not allowed for the review, a message containing a warning is displayed to the user.

#### 4.4.3.6 See and upload work result materials

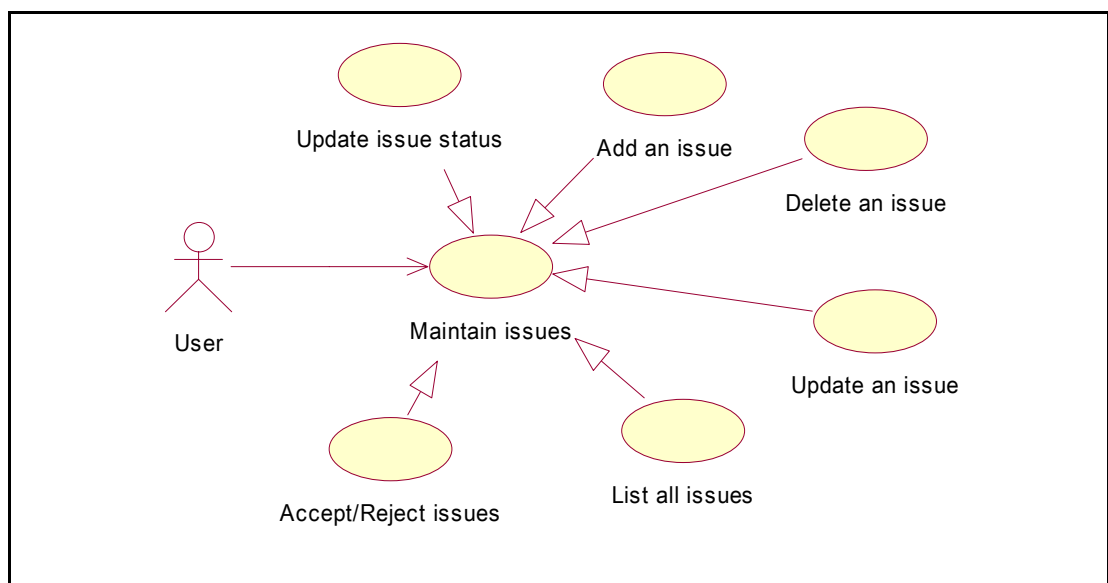
The user can see the uploaded work result materials in the review. Material names and associated files name are displayed by the system. For example, material name may be “Review Checklist Form” and associated file name may be “Chklst.xls”. User can see and download materials listed by the system. Thus, the system allow user to upload work result materials if that review is designated to allow uploading work result materials. In this situation, the system lists the material names and whether uploading them is mandatory or not. The user uploads requested materials to the system. He can replace the uploaded materials until he finishes his job by selecting “Finish This Task” option. If uploading work result materials is not allowed for the review, a message containing a warning is displayed to the user.

#### 4.4.3.7 Issue Related Operations in a Review

A user can add an issue to a review material and delete or change an existing issue in a review material. Furthermore, the system allows users to accept or reject

entered issues and update status of the issues whether they are resolved or not (See Figure 17). Issue related operations are displayed by the system according to the task's features. Following controls are performed by the system to display related options and allow users to perform those activities.

- “Are issues added/removed/updated in this task?”: If “Yes” is selected by the administrator for the task, users of this task are allowed to add, remove or change issues.
- “Are the issues criticized in this task?”: If “Yes” is selected by the administrator for the task, users of this task are allowed to accept or reject entered issues.
- “Are the statuses of the issues updated in this task?”: If “Yes” is selected by the administrator for the task, users of this task are allowed to mark issues as “resolved” or “unresolved”.



**Figure 17 : Operations for Maintaining Issues**

- Add, remove, and update issues

When the user selects this option, review materials are listed by the system. The user can select any of them and press “Open Material”. Then the system opens the selected review material. The system supports four types of the materials for online annotation of the issues (See Table 3):

1. Microsoft Word Document: Users can add an issue wherever he wants in the document. Users can select a letter, a word, several words, a line, a paragraph and even a whole page to add an issue to the selected area in the document. The system opens a Web page and asks users for issue type, criticality level of the issue, description of the issue and proposed action for the issue. After the user presses the “Add Issue” button, the issue is displayed in the document automatically. In addition to the information mentioned above, the username of the user who entered the issue is displayed in the issue cell. Color of the selected region for adding issue is changed according to the criticality level of the issue; red for high level of criticality, orange for medium level of criticality and yellow for low level of criticality.
2. Microsoft Excel Spreadsheet: Users can add an issue in any cell of a sheet. Users select a cell in a sheet and press “Add an Issue” button. The system opens a Web page and asks the user for issue type, criticality level of the issue, description of the issue and proposed action for the issue. After the user presses the “Add Issue” button, the issue is displayed in the sheet automatically. In addition to the information mentioned above, the username of the user who entered the issue is displayed in the issue cell. Color of the issue cell is changed according to the criticality level of the issue.
3. Microsoft PowerPoint Presentation: Users can add an issue to any of the slide in a presentation file. Users select a slide and press “Add an Issue” button. The system opens a Web page and asks the user for issue type, criticality level of the issue, description of the issue and proposed action for the issue. After the user presses the “Add Issue” button, the issue is displayed in the slide automatically. In addition to the information mentioned above, the username of the user who entered the issue is displayed in the issue cell. Color of the issue cell is changed according to the criticality level of the issue.
4. Rational Rose Model: There are four different views in the Rose

Models; “Use Case View”, “Logical View”, “Component View” and “Deployment View”. There are different types of diagrams in mentioned views including “Use Case Diagram”, “Class Diagram”, “Collaboration Diagram”, “Activity Diagram”, “Statechart Diagram”, “Sequence Diagram”, “Component Diagram” and “Deployment Diagram”. The user selects any object in any of the mentioned diagrams and presses “Add an Issue” button. The system opens a Web page and asks the user for issue type, criticality level of the issue, description of the issue and proposed action for the issue. After the user presses the “Add Issue” button, the issue is displayed in the sheet automatically. If the user does not select any object in the diagram, issue is inserted to the diagram as a general note. On the other hand, if the user selects an object, the issue is inserted to the diagram and attached to the selected object. In addition to the information mentioned above, the username of the user who entered the issue is displayed in the issue cell. Color of the issue cell is changed according to the criticality level of the issue.

**Table 3 : Material Types that are Allowed For Online Annotation**

<b>Material Type</b>	<b>Allowed Object to Attach an Issue</b>
Microsoft Word Document	A letter, a word, a group of words, a line, a paragraph, a group of paragraphs, a whole page etc.
Microsoft Excel Spreadsheet	A cell in any sheet.
Microsoft PowerPoint Presentation	A slide.
Rational Rose Model	A diagram (all types of diagrams are supported), an object in a diagram.

When a user wants to remove or change an issue, he selects the issue and presses “Change This Issue” or “Remove This Issue” button. The system automatically opens a Web page and displays information about the issue. The user can remove the issue or can make changes in the issue and update it.

- List issues

The system allows the users to list their issues on the review material. The user

presses the “List Issues” button then the system displays his own issues on the current review material in a different Web page. Position, type and criticality of the issues, issue description and proposed action to solve each issue are displayed in this page. The user can select an issue from the list by clicking the issue number on the left side of the page. After selecting an issue, he can remove or change the issue.

- Accept or reject issues

The system allows the users to accept or reject entered issues on the review materials (This user may be a moderator, review leader, etc.). At first, the user can see the entered issues on the review materials. When the user selects option for seeing the issues on the review materials, the system asks him for selecting a review material and the user or users (review participants) who have issues on that material. According to selected criteria, the system displays the issues on the material. Likewise, when the user wants to mark issues as “accepted” or “rejected”, the system asks him for review materials and users who have issues on the review materials for displaying the issues. Issues and detailed information about them are displayed according to selected criteria and the user can mark them as “accepted” or “rejected”.

- Update status of the issues

The user can update status of the entered issues on the review materials as “resolved” or “unresolved” (This user may be an author, producer, etc.). At first, the user can see the entered issues on the review materials. When the user selects option for seeing the issues on the review materials, the system asks him for selecting a review material and user or users (review participants) who have issues on that material. According to selected criteria, the system displays the issues on the material. Likewise, when the user wants to mark issues as “resolved” or “unresolved”, the system asks him for review materials and users who have issues on the review materials for displaying the issues. Issues and detailed information about them are displayed according to selected criteria and the user can mark them as “resolved” or “unresolved”.

#### 4.4.3.8 Reporting Operations

The system provides the users with two different kinds of report. One of them

is about issues and the other is about efforts.

- Take a report about issues

The users can take reports about issues based on the selected criteria. The system displays review materials and the users (review participants) who have issues on the review materials. The user can select review materials and users by pressing check boxes next to each material and user. Furthermore, he can select issue type, issue criticality and issue status (accepted/rejected, resolved/unresolved) as report criteria. After giving a report name and pressing the “Report” button, the system displays the result of the report according to selected criteria. Review material, the user who entered the issue, issue position, type, and criticality, description of the issue, proposed action to solve the issue and the issue status information (accepted/rejected, resolved/unresolved) is displayed in the report. The user can save or print the report. (Note: Save and Print operations will not be implemented in the current version.)

- Take a report about efforts

When the user selects this option, the system displays a report about the efforts of the users for the tasks. Effort information is expressed in “hour”. Tasks, estimated effort for each task, average effort for each task and individual users’ efforts for each task are displayed in the report. The users who have “0” (zero) efforts for a task (this implies that, they have not finished the task yet when the report is taken) will not be included in average effort calculation.

#### 4.4.3.9 Finish the task

The user selects this option when he finishes his task. The system asks the user for the total effort (in hours) and displays the target tasks that follow the current task. Automatic effort calculation is not handled by the system; the user has to give his total effort. If the current task has “AND” logical connector or there is only one task as a target task, the system does not ask the user for selecting a target task. Otherwise, the user is enforced by the system to select a target task that will be the next task after the finishing of the current task. When the user presses the “Done” button, the system assigns his status “Finished” for the current task. If all the users



that were assigned to the task finish their tasks, the system activates the next task(s). When a new task is activated, a notification message is sent to all review participants informing them about the situation.

## 4.5 Logical Database Requirements

In the system, all the data will be kept on a relational database management system. Entity relationship diagram is given in Appendix A. Tables and their fields with a brief explanation are as follows:

- Table “issue”: To keep information about an issue that is entered by the review participants.

prj: Name of the project.

review: Name of the review.

material: Name of the material in which issue is entered.

user: Name of the user who entered the issue.

position1: To keep position information of an issue.

position2: To keep position information of an issue.

position3: To keep position information of an issue.

issuetype: Type of the issue.

criticality: Criticality of the issue.

description: Description of the issue.

proposedaction: Proposed action to solve the entered issue.

accepted: To keep whether issue is accepted or not.

status: To keep whether issue is resolved or not.

- Table “issue\_criticality” : To keep issue criticality types such as “High”, “Medium”, and “Low”.

criticality: Criticality type of the issues.

- Table “issue\_result”: Temporary fields to exchange information between JSP and ActiveX codes.

result: To keep whether operation is successful or not (“OK” or “ERROR” is kept).

info: Detailed information about the issue operation.

- Table “issue\_type”: To keep types of the issues.

type: Name of the issue type such as “Wrong”, “Inconsistent” and “General”.

- Table “material”: To keep information about the materials.

wf: Name of the workflow.  
 task: Name of the task.  
 name: Name of the material.  
 type: Type of the material such as “Review Material” and “Supportive Material”.  
 uploadmandatory: To keep whether uploading the material is mandatory or not.  
 path: Path of the material in the file system.

- Table “material\_type”: To keep material types in the system.

type: Type of the material.

- Table “myuser”: To keep information about the users.

username: Username of the user.  
 password: Password of the user.  
 name: Name of the user.  
 surname: Surname of the user.  
 mail: Mail address of the user.  
 usergroup: To keep whether the user “System Admin” or “Ordinary User”.

- Table “project”: To keep project names in the system.

name: Name of the project.

- Table “projectusers”: To keep users of the projects.

prj: Name of the project.  
 user: Username of the project user.  
 usergroup: The group of the user.

- Table “review”: To keep information about a review.

name: Name of the review name.  
 wf: Name of the workflow.  
 prj: Name of the project.  
 status: Status of the review such as “Active”, “Inactive” and “Finished”.  
 role\_finished: To indicate whether assignment of the roles is finished or not.  
 schedule\_finished: To indicate whether assignment of the schedule is finished or not.

- Table “review\_schedule”: To keep schedule information of a review tasks.

prj: Name of the project.  
 review: Name of the review.  
 task: Name of the task.  
 startdate: Starting date of the review task.  
 enddate: Finishing date of the review task.  
 estimated\_effort: Estimated effort for a review task.

- Table “review\_user\_role”: To keep role of a user in a specific review.

prj: Name of the project.  
 review: Name of the review.  
 user: Username of the user.  
 role: Role of the user.

- Table “role”: To keep role names in the system.

name: Name of the role.

- Table “server\_settings”: To keep information about the servers.

webserver: Name or IP address of the Web server.  
 mailserver: Name or IP address of the mail server.

- Table “task”: To keep information about the workflow tasks.

name: Name of the task.  
 wf: Name of the workflow.  
 taskorder: Order of the task in the workflow.  
 splittype: Logical connector of the task (“OR” or “AND”).  
 startmsg: The message sent to all review participant when the task is started.  
 issueoperations: Whether adding/removing of the issues is performed in the task.  
 issuecriticism: Whether accepting/rejecting of the issues is performed in the task.  
 issuestatus: Whether issue status is updated in the task.

- Table “task\_role”: To keep the roles assigned to a workflow task.

wf: Name of the workflow.  
 task: Name of the task.  
 role: Name of the role.

- Table “tasks\_of\_users”: To keep information about the review tasks and users that are assigned to those tasks.

prj: Name of the project.  
 review: Name of the review  
 wf: Name of the workflow.  
 user: Username of the user.  
 task: Name of the task.  
 estimated\_effort: Effort of the user for the task.  
 status: Status of the task whether the user finished it or not.

- Table “temp\_issue”: To exchange information between JSP and ActiveX controls.

prj: Name of the project.  
 review: Name of the review.

material: Name of the material.  
user: Username of the user.  
position1: Position information of the issue.  
position2: Position information of the issue.  
position3: Position information of the issue.  
issuetype: Type of the issue.  
criticality: Criticality level of the issue.  
description: Description of the issue.  
proposedaction: Proposed action to solve the issue.

- Table “transition”: To keep information about the transitions in a workflow.

name: Name of the transition.  
fromtask: Starting task name of the transition.  
totask: Ending task name of the transition.  
wf: Name of the workflow.

- Table “workflow”: To keep information about the workflows in the system.

name: Name of the workflow.

## 4.6 System Architecture

EasyReview is designed as a three-tier architecture, which consists of presentation layer (also known as client layer), functional layer (middle tier) and data layer (back-end).

At the presentation layer, the client requests a service; the request is passed to the functional layer where it is processed while the data layer supplies the necessary stored data. After the processing is done, the functional layer passes the results to the presentation layer, which will display the result to the user. The architecture is depicted in Figure 18.

### 4.6.1 First Tier (Client):

The Web browser is the essential resource in the first-tier. A Java Virtual Machine (JVM) enhanced, JavaScript code enabled Web browser is used to connect to the system. HyperText Markup Language (HTML) is the main language in this tier. ActiveX controls and plug-ins must be enabled in the Web browser in order to control other applications such as Microsoft Word and Microsoft Excel within the browser.

#### 4.6.2 Middle-Tier:

Application logic resides in the middle-tier. Web server and application server are the containers for the services and resources. Java Server Pages (JSP) provides a framework to create dynamic content on the server using HTML, XML, and Java code. Through Java Database Connectivity (JDBC), this tier communicates with the database within the back-end tier. Besides, connection to the mail server is handled by this tier.

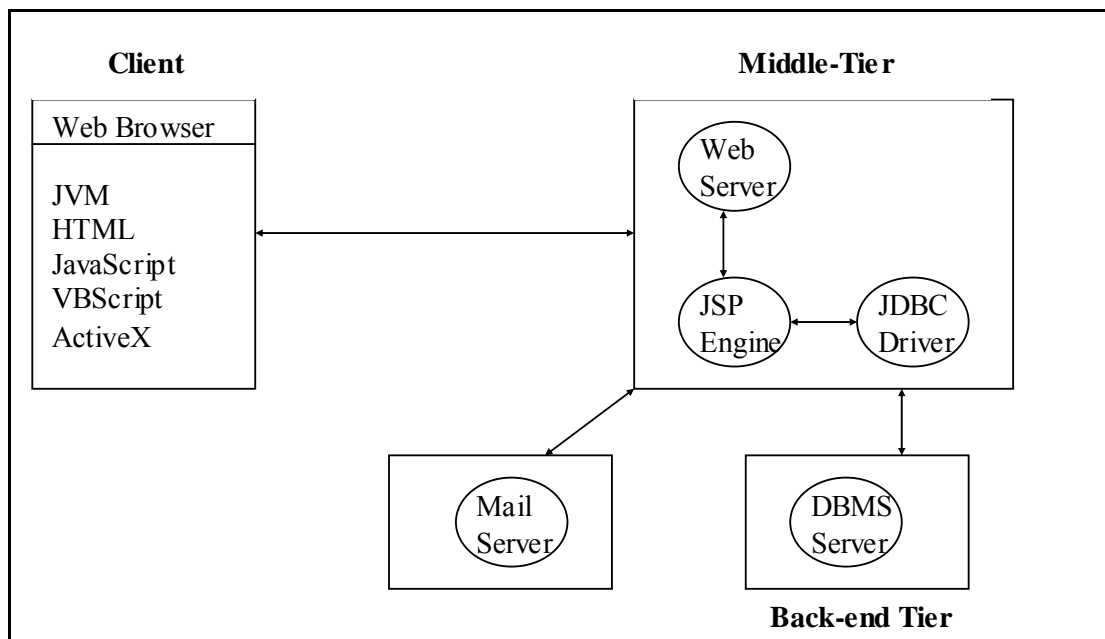


Figure 18 : System Architecture of EasyReview

#### 4.6.3 Back-end Tier:

Database of the system resides in the back-end tier, which is also known as the data-tier. MySQL relational database management system is used in EasyReview.

### 4.7 Decomposition Description

Decomposition description involves packages as design entities, and provides a hierarchical structure of the system as a whole under these packages. EasyReview software is developed under five packages; “tez”, “tez.data”, “java.lang”, “java.sql”, and “java.util”.

“tez” package contains the Java classes that realize the application logic of the

system. On the other hand, “tez.data” package contains Java classes that serve as a wrapper of the data entities in the database. Remaining packages are standard Java Development Kit (JDK) packages. The relationship between packages is shown in Figure 19 and Figure 20.

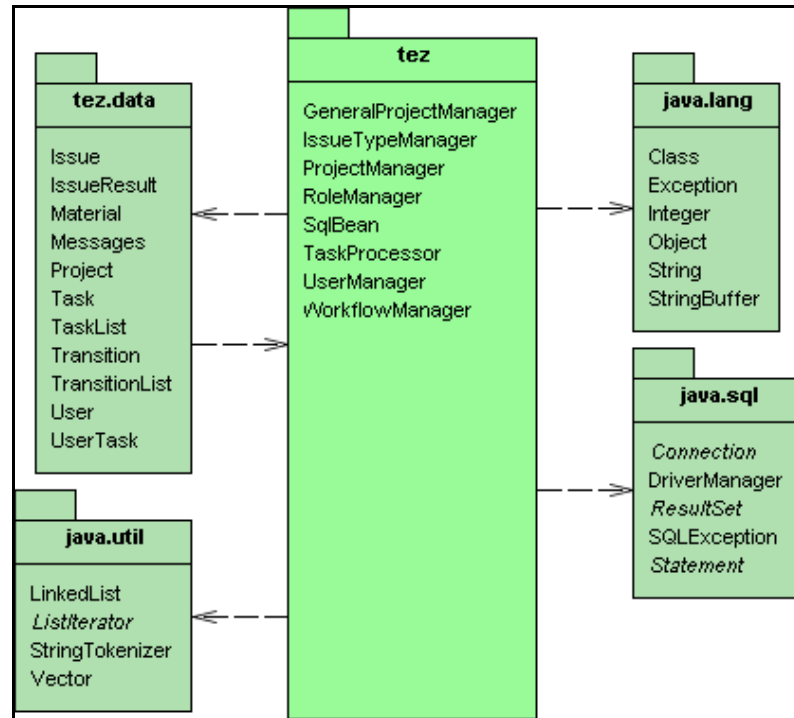


Figure 19 : Decomposition Diagram of Package “tez”

## 4.8 Dependency and Detailed Descriptions

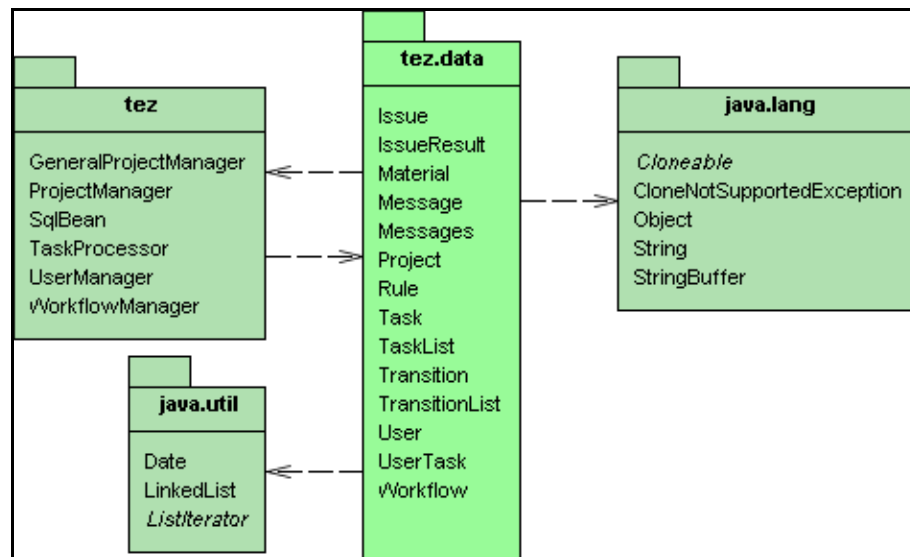
This section gives a brief overview of the classes. Combined class diagrams for each class in the system are placed in Appendix B. The combined class diagram for a class displays the class in the center of the diagram with the relationships surrounding it.

### 4.8.1 GeneralProjectManager Class:

This class is used for getting all the users of a project from the database.

### 4.8.2 IssueTypeManager Class:

This class is used for maintaining issue types in the database. There are methods for controlling, adding, deleting and listing issue types kept in the database.



**Figure 20 : Decomposition Diagram of Package “tez.data”**

### 4.8.3 ProjectManager Class:

This class is responsible for maintaining the projects and reviews in those projects. Adding, removing and updating review and project information are handled. All the controls and business logic about reviews, review schedules and review users are performed in this class.

### 4.8.4 RoleManager Class:

This class is used for maintaining roles in the database. There are methods for controlling, adding, deleting and listing roles kept in the database.

### 4.8.5 SqlBean Class:

The `SqlBean` class deals with database operations. Providing connection to database, getting, adding, removing and updating information in the database are performed via this class.

### 4.8.6 TaskProcessor Class:

All the issue related operations such as adding, removing and updating issues, and task related operations are handled by this class. Assigning tasks to users, finding active task in a review and detailed information about it, allowed uploading operations for a task, getting users of a task, getting information about uploaded

materials, and updating status of a task are performed by the TaskProcessor class.

#### **4.8.7 UserManager Class:**

User related operations are performed within UserManager class. Getting users from database, adding new users to database, deleting users from database, authenticating users when they attempt to login to the system and changing passwords of users are handled by this class.

#### **4.8.8 WorkflowManager Class:**

Operations related to maintaining workflows are achieved by this class. The WorkflowManager class is responsible for adding, removing and updating transitions, creating workflows, arranging tasks of workflows.

#### **4.8.9 Data Classes:**

The data classes reside in “tez.data” package and store data entities which are kept in the database. These classes are; Issue, IssueResult, Material, Message, Messages, Project, Task, TaskList, Transition, TransitionList, User, UserTask, and Workflow. Attributes of these classes are shown in class diagrams in Appendix B.

### **4.9 User Interface Description**
























This section describes the user interface layout of EasyReview. The main user interfaces are explained in detail. In order to provide effective user interfaces, following matters are considered during the design.

- The layout of the pages shall be uniform throughout the system.
- Main operations of the tool shall be visible and accessible from each page.
- Every page shall have an informative title.
- Navigation between pages shall be easy.






In EasyReview, every operation has a descriptive icon. These icons are placed in Table 4.

**Table 4 : Icons and Their Descriptions in EasyReview**

Icon	Description
	Assign Roles to Users
	Assign Dates
	Go Back
	Change Task Orders
	Maintain Server Settings
	Set Logical Connectors
	Determine Issue-Material-Owner
	Change Password
	Upload, See Materials
	User Administration, Add, Delete, Change, List Users
	Role Administration, Add, List, Remove Roles
	Workflow Administration, Add, Delete, List, Customize Workflows
	Add, Delete, Change Transitions
	Add, Delete, Change, Finish Task
	Starting Task, Ending Task
	Review Administration, Add, Delete, Add/Delete Operations, Customize, Activate Review
	Review status icons
	Add, Delete, Change, List, See Issues
	Issue Update Operations, Issue Update
	Issue Accept/Reject Operations, Issue Accept/Reject
	Issue Type Administration, Add, Delete, Change, List Issue Types
	Issue Report, Effort Report
	Project Management, Add, Delete, Change, List, Customize Projects

**Table 4 : Icons and Their Descriptions in EasyReview (cont.)**

Icon	Description
  	Informative icons

Main user interfaces in the system that have general operations are explained in the following sections. Sample user interfaces from the system are given in Appendix C.

#### 4.9.1 Main Page for Administration

After logging into the system, the users that have administrative privileges on the system (the users that are member of “System Admin” group) are redirected to this page. Server settings, role administration, issue type administration, user administration, workflow and project administration operations pages are accessed from this page (See Figure 21).



**Figure 21 : Main Page for Administration of EasyReview**

#### 4.9.2 Welcome Page for Review Participants

Ordinary users (review participants, the users that are member of the “Ordinary User” group) of the EasyReview are redirected to the welcome page. The users can change their passwords and see reviews that have been assigned for them. (See Figure 22)



Figure 22 : Welcome Page for Review Participants

#### 4.9.3 Main Page for Review Operations

After selecting a review from the welcome page, users are redirected to the main page in order to perform review related activities. Users can see the general information about the review, see, upload and download materials, take reports, perform issue operations and finish their tasks. There are three different issue operations in the system; “Add/Remove Issues”, “Accept/Reject Issues” and “Update Issue Status”. Issue operations are displayed on the page according to the current task’s properties. That is, if the task is assigned for adding or removing issues, “Add/Remove Issues” hyperlink is displayed. Likewise, if the task is assigned for accepting and rejecting issues, “Accept/Reject Issues” is displayed and so on. (See Figure 23)



Figure 23 : Main Page for Review Operations

## CHAPTER 5

# EXPERIMENTAL STUDY

The purpose of this chapter is to evaluate how EasyReview can overcome the shortcomings of traditional review process and inadequacies of the existing review tools.

### 5.1 Objectives

In the “Problem Statement” section of this thesis, a number of shortcomings of traditional review process and inadequacies of the existing tools were discussed.

In a traditional review process, there are a number of error-prone routine tasks. Documents are created and then distributed to all review participants. Review participants detect and record the defects on the review documents. Issues and defects are collected in the review meetings and so on.

Most of the existing review tools support only one review process and lack the capability to define a new review process according to the needs.

The following objectives have been extracted from “Problem Statement” section and efforts have been focused on fulfilling those objectives:

- **Objective 1:** To determine whether EasyReview provides support for review activities without face-to-face meeting.
- **Objective 2:** To determine whether EasyReview eliminates the distribution of review materials and other supportive materials to the review participants and facilitates the access to those materials.

- **Objective 3:** To determine whether EasyReview has the ability to define a new review process for an organization's needs.

## **5.2 The Experiment**

An experiment was conducted to determine whether EasyReview could fulfill the stated objectives. Moreover, usability of the tool was questioned in the experiment.

### **5.2.1 Participants**

Eight people were encouraged to participate in the experiment. Any special restriction was not imposed on the participants during the selection except all of them should have computer literacy. All participants were properly informed of the goals and procedures of the test and were asked to sign a consent form before the test.

### **5.2.2 Location and Apparatus**

The experiment was conducted in the facilities of an organization that is a Turkish Armed Forces Foundation (TAFF) investment. The organization was an ISO 9001, software quality assurance certified establishment.

A server was assigned to host the server side of EasyReview tool. Server and client computers were connected to the local area network of the organization. The Web address to which participants connected was "http://pchvkk6:8080/web/". Test participants connected to the tool by using their computers via Microsoft Internet Explorer Web browser. All the participants' computers had Microsoft Office XP and Rational Rose Enterprise Edition 2001 software.

### **5.2.3 Experiment Procedure**

The experiment was conducted from May 14 to May 21, 2004. Before the experiment, participants were informed about the study and they signed a consent form, which is given at Appendix D.

Properties and usage of the tool were explained in the briefing and the tool was configured for the experiment. The details of the configuration can be seen in the

following section, “Configuration of the Tool”. Then the eight participants were divided into two review teams, each consisting of four participants and individual roles in the review processes were assigned. After the briefing, the participants were given usernames and passwords and they filled out a pre-test questionnaire that is given in Appendix E.

Two different review processes were defined before the experiment. Each review process was carried out by one moderator, one author and two reviewers. Allocated time for both of the reviews were seven days. Participants did not need to spend time outside their jobs for this experiment. In fact, they integrated the requirements of the experiment into their daily workload. They used their own computers whenever they wanted to perform review activities. Participants contacted with me at any time they needed help. Review meetings were not held during the experiment.

In the experiment, two different review processes were conducted; “Fagan’s Inspection Process” and the organization’s own review process. In the Fagan’s Inspection process, a Microsoft PowerPoint presentation and a Microsoft Excel spreadsheet were reviewed by one of the review teams. Likewise, a Rational Rose model and a Microsoft Word document were reviewed using the organization’s own review process by the other review team.

Review materials were selected only to test the EasyReview tool in the experiment. In other words, review materials were only sample documents, not related to the participants’ current jobs in the organization. One of the four review documents was a Microsoft PowerPoint presentation was about specifications of programming languages with 31 slides. The second review document was about comparison of programming languages in Microsoft Excel format. In the third review document, well-known ATM (Automated Teller Machine) system was modeled with Rational Rose model. The last document had use-case definitions of ATM System model in the Microsoft Word document format with eight pages. Ten intentional mistakes and errors were inserted into each of these documents before the experiment.

There were “Author” roles in both of the conducted review processes. However, in the experiment, the participants were not the real authors of the review materials. In spite of this, the participants with author roles acted as the real author of the review materials during the experiment. However, they did not actually resolve the issues during the “Rework” activity of the review processes.

After the experiment, participants filled out an evaluation questionnaire (Questionnaire for Practical Heuristics for Usability Evaluation, [Perlman, 2002]) about the usability of the tool. This questionnaire is given in Appendix F. The participants were asked to choose the best suitable answer from bi-polar (i.e. good/bad) rating scales.

#### **5.2.4 Limitations of the Experiment**

There were three limitations in the experiment. The first limitation was that, the experiment could have been done with more participants. Unfortunately, only eight participants were available. The second limitation was that, content of the review materials was not related to some of the participants’ profession. Therefore, the participants could not detect content related defects, as they did not have the necessary background information. The last drawback of the experiment was that, there was not enough time to compare the result of this experiment with an experiment that should have been conducted using traditional review methods.

### **5.3 Configuration of the Tool**

The tool was configured to support the Fagan’s Inspection Process and the organization’s own review process, which is given in Appendix H. For simplicity, configuration steps are given only for the organization’s review process. The configuration steps are explained below:

1. Add new roles: “Moderator”, “Reviewer” and “Author” roles exist in the database. Therefore, there is no need to add new roles for this case.
2. Add new issue types: “Editorial”, “Not conforming to standards”, “Checklist item incomplete / missing / inadequate”, “Ambiguous”, and



“Incorrect” issue types were added. “Inconsistent” and “Other” issue types already exist in the database.

3. Add new users: New review participants (users) were added to the database of the EasyReview.
4. Define new workflow: A new workflow that handles the review procedure (process) of the organization was added to the system. Workflow information is depicted in both Table 5 and Figure 24.

**Table 5 : Workflow of TAFF Organization’s Review Procedure**

<b>Workflow Name:</b> TAFF Review Workflow		
<b>TRANSITIONS</b>		
NAME	FROM TASK	TO TASK
Planning Finished	Planning	Overview
Overview Finished	Overview	Preparation
Preparation Finished	Preparation	Review
Accepted As Is	Review	Review Reporting
Rework Required	Review	Rework
Rework Finished	Rework	Follow Up
Follow Up Finished	Follow Up	Review Reporting
<b>TASKS</b>		
NAME	LOGICAL CONNECTOR	
Planning	OR	
Overview	OR	
Preparation	OR	
Review	OR	
Rework	OR	
Follow Up	OR	
Review Reporting	OR	

5. Create a new project: A new project was created for this case. Users and supportive materials (Standards, checklists and guidelines) were assigned to the project.
6. Create new review in the project: A new review was created in the project,

which was created in the previous step. Review participants were assigned to the roles of each task in the workflow. Besides, starting times, ending times and estimated efforts for the tasks were assigned.

7. Activate the review: Review was activated by the administrator and the review process was started.

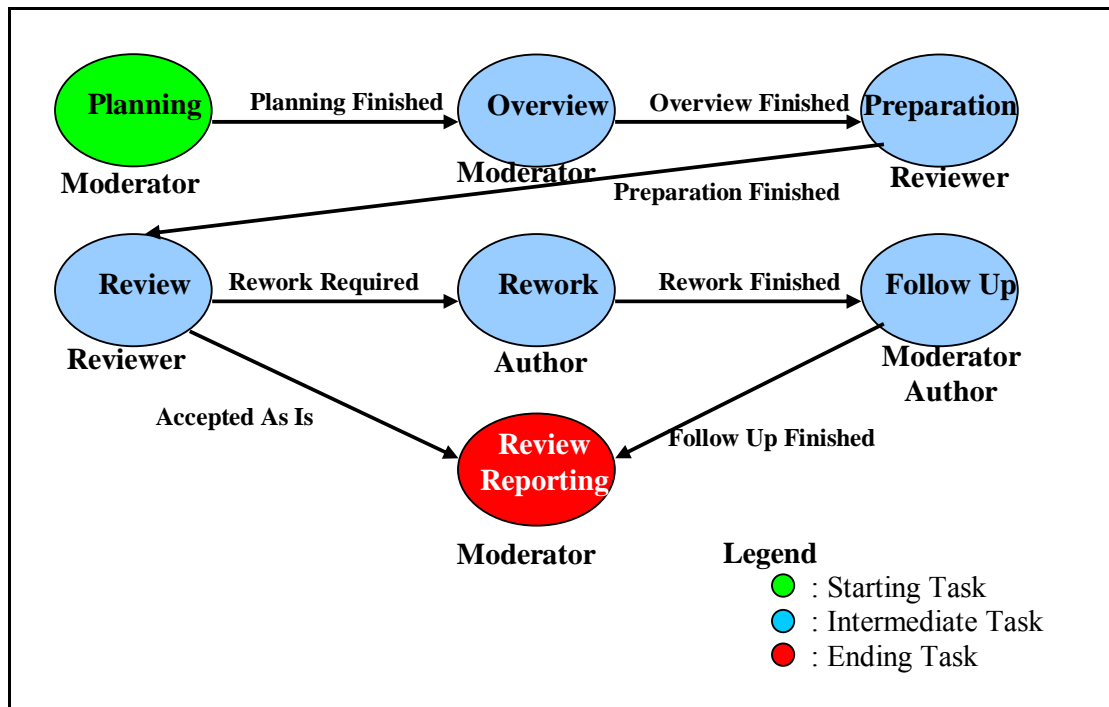


Figure 24 : Workflow of TAFF Organization's Review Procedure

## 5.4 Evaluation of the Tool

Evaluation of EasyReview is based on three criteria; the first one is the tool's fulfillment of the objectives stated in this chapter, the second one is the tool's usability, and the last one is the tool's capability to meet the specifications of a different ISO 9001 certified organization's review procedure, which is given in Appendix H. The review procedure of that organization is excerpted from Yanbaş's study [Yanbaş, 2002].

### 5.4.1 Fulfillment of Objectives

The experimental study has proved that EasyReview meets all the objectives stated in the beginning of this chapter. Fulfillment of each objective is explained in

the following paragraphs:

- **Fulfillment of Objective 1:** EasyReview is a Web based tool that runs on a Web Server and on an Application Server. Users of the tool can connect it using a Web browser only. In the experimental study, the review meeting was eliminated. Therefore, review participants performed their works without a face-to-face meeting. When a user connects to the system, he can see reviews and roles assigned to him. Users can also see detailed information of a review to which they are assigned such as their tasks in the review process, review and supportive materials, status of the review participants (whether they finished their tasks), efforts of the review participants for finished tasks. Menu of the tool is displayed according to the users' roles and tasks in the review process, so the users can easily perform their tasks.

In the experimental study, four of the eight participants (two “Inspector” in the Fagan’s process and two “Reviewer” in the organization’s process) reviewed the documents. The distribution of issues found by reviewers is shown in Table 6. Number of different issues is counted. The detailed information about the intentional mistakes is given in Appendix J.

Reviewers only select the position where the defect is found and enter the defect type, criticality of the issue, description of the issue and proposed action to correct the defect. When an issue is entered, all the mentioned information and the reviewer who entered the issue are displayed in a unified form. Issues are displayed on the review documents according to their criticality levels; red color for “High” critical issues, orange color for “Medium” critical issues and yellow color for “Low” critical issues. When an issue is entered, all the review participants can see it if they take a report about the issues. This capability gives chance to users to evaluate their issues before entering their own. Automatic data collection and elimination of the need for an issue recorder are the significant improvements in a review process.

**Table 6 : Distribution of the Found Issues in the Experiment**

<b>Document</b>	<b># of Different Issues Found in Intentional Mistakes</b>	<b># of Total Different Issues Found</b>
ATM System.mdl	6 of 10	6
Detailed Use Case Definitions.doc	8 of 10	11
PD_Karsilastirma_Tablosu.xls	7 of 10	8
PD_Sunum.ppt	6 of 10	6

When the reviewers finish their works, moderators can see the issues on the review materials or on a report. The tool allows moderators to choose some criteria to see or report the issues such as “issues with High criticality”, “issues raised by a reviewer”, and so on. The tool provides support for moderators to accept or reject the raised issues. Depending on the issues, moderators can request the author to rework on the review materials. When a rework is required, authors can see all of the issues on the review materials or on a report. Likewise, the tool allows authors to set some criteria to see or report the issues. The tool provides support for authors to indicate whether an issue is resolved during the rework.

All the mentioned activities were tested in the experimental study and it is proven that EasyReview tool successfully supports review activities without a review meeting.

- **Fulfillment of Objective 2:** In a review process, review materials and supportive materials such as standards and checklists must be delivered to the review participants before the review. EasyReview eliminates these error-prone tasks and allows moderators to put review materials and other supportive materials on the server. When a review participant connects to the tool, he can see and download all the materials. With the help of EasyReview, review participants can access all the review and supportive materials easily and moderators do not need to send materials to the review participants via e-mail or courier.

- **Fulfillment of Objective 3:** EasyReview allows authorized users to define and customize workflows. Activities (i.e. tasks) in a review process, relationships between those activities, order of the activities, responsible roles (e.g. Moderator, Author, Reviewer, etc.) for an activity can be defined in a workflow. In the experimental study, two different workflows were defined, one of them for “Fagan’s Inspection Process” and the other for organization’s own review process. Test participants performed their work using the tool. In addition to the workflow capability, the tool also allows defining new review roles and issue types that are essential for a review process.

#### 5.4.2 Usability of the Tool

At the beginning of the experimental study, participants filled out a pre-test questionnaire. With the help of the pre-test questionnaire, information about each participant was collected. In addition, their familiarization with the review literacy, operating systems and Web browsers they use were obtained. Results of the pre-test questionnaire are given in Table 7.

**Table 7 : Results of the Pre-Test Questionnaire**

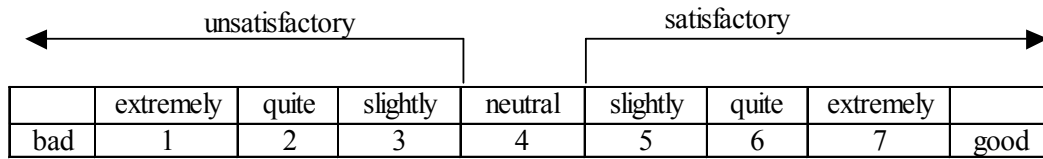
Features	Comments
Familiar With Review Literacy	4 of the 8 participants were not familiar with review literacy
Participating in a Formal Review	3 participants had participated more than five times in a formal review, 1 participant had participated less than five times and 4 participants had never participated in a formal review
Operating System	All of the participants use Windows operating systems most often
Web browser	All of the participants use Internet Explorer most often

At the end of the experimental study, participants filled out a questionnaire in order to evaluate the usability of the tool. Results of the questionnaire are given in Table 8. The leftmost cell of the table includes questions. The other cells of the table include minimum, maximum and median values of the grades. Fractional median values were rounded to lower bounds; for example, 2.5 was rounded to 2.

**Table 8 : Results of the Questionnaire**

<b>LEARNING</b>	<b>Min</b>	<b>Max</b>	<b>Median</b>
1. Help and Documentation Design for use without documentation. Provide easy-to-use task-oriented help.	2	4	2
2. Adopt the User's Viewpoint Speak the user's language (avoid jargon). Make use of existing knowledge (familiar mental models).	4	6	5
3. Simple and Natural Dialogue Avoid extraneous information, steps, actions. Information should be in a logical, natural order.	5	7	7
4. Design for Advancement Provide shortcuts (quick keys, customization).	4	7	5
<b>ADAPTING TO THE USER</b>	<b>Min</b>	<b>Max</b>	<b>Median</b>
5. Provide Maps and a Trail Give the user a way to preview where to go, what will happen. Give the user a way to review / return-to previous contexts.	6	7	7
6. Show the User What is (Not) Possible Provide affordances to indicate what can be done.	6	7	7
7. Intuitive Mappings Design good response compatibility between controls/actions.	5	7	6
8. Minimize Memory Load Remove the need to remember across dialogues. Provide multiple views for easy comparisons.	6	7	7
9. Consistency in the System and to Standards Make sure the same term/action has one meaning. When there is no better way, conform to a standard.	6	7	6
<b>FEEDBACK AND ERRORS</b>	<b>Min</b>	<b>Max</b>	<b>Median</b>
10. Feedback Provide timely feedback about all processes, system status	6	7	6
11. Prevent Errors Make it difficult to make errors.	6	7	7
12. Error Messages Diagnose the source and cause of a problem and suggest a solution	7	7	7
13. Clearly Marked Exits and Error Recovery Make sure the user can get out of an undesirable state easily. Design assuming that people will make errors and need to recover previous states	6	7	6

Seven-point scale was used in the questionnaire. In a seven-point scale, (See Figure 25) number 1 represents "extremely bad" and number 7 represents "extremely good". A value "5 and above" is considered "satisfactory", implying that end-users are generally pleased with the system and designers need not to make any changes. A value "3 and below" is deemed "unsatisfactory" indicating that end-users are dissatisfied with the system, and the designers should make necessary changes to correct the deficiency. A mid-value of 4 is taken to be "neutral", and probably designers should find out more from end-users and make changes if required.



**Figure 25: A Seven-point Scale to Measure End-Users' Responses**

According to the results, participants expect more online help with respect to the functions of the tool while performing the tasks. With the exception of this, participants are satisfied with EasyReview tool in terms of usability.

The last two questions are open-ended comment questions; therefore, they are not graded. Negative comments include the tool's lack of online help and GUI restriction during the definition of the workflows. On the other hand, by looking at the positive comments, the participants' high level of the satisfaction can be observed. The participants stressed that, with the workflow capability, the review process can be customized according to the needs. They also emphasize that, review participants can focus on only their tasks because of the automation, dynamic menu that is displayed according to the participants' tasks and automatic sending of e-mails. Evaluation results are given in Appendix G.

#### **5.4.3 Meeting the Specification of Review Procedure**

Table 9 summarizes the comparison of functional capabilities of EasyReview with respect to the specifications listed in the Appendix I. The specifications of review procedure of the ISO 9001 certified organization is excerpted from Yanbaş's study [Yanbaş, 2002]. In the "Comments" column of the table, detailed explanation

of the tool's functionality is provided.

**Table 9 : Evaluations of the Organization's Review Procedure Specifications**

Item #	EasyReview	Comments
		<b>General</b>
1.1	N/A	This is a feature of review process itself.
1.2	N/A	This is a feature of review process itself.
1.3	Yes	Review materials, supportive materials are supported by EasyReview. Furthermore, work result materials such as filled forms are also supported.
1.4	Yes	Providing that a username and password, all personnel can be a review participant.
1.5	Yes	All outputs can be defined in the workflow of a review process.
1.6	Yes	Workflow can be customized according to this entry.
1.7	Yes	Workflow can be customized such that.
1.8	Yes	All types of the roles can be created in the tool.
1.9	Yes	Any role including "Author" and "Review Leader" can initiate a review in the tool.
1.10	Yes	Workflow can be customized such that.
1.11	Yes	Any role including "Recorder" can create "Individual Review Log".
1.12	Yes	Any role including "Review Leader" can be assigned to perform this task.
1.13	Yes	Any role including "Reviewer" can be assigned to perform this task.
1.14	Yes	There is no need for recorder; EasyReview automatically carries out his/her task.
1.15	Yes	Any role including "Author" can be assigned to perform this task.
		<b>Preparation</b>
2.1	Yes	Review process can be customized by workflow capability to perform such a process.
2.2	Yes	Supported.
2.3	Yes	Supported.
2.4	Yes	Supported.
2.5	Yes	Supported.
2.6	N/A	The review process in EasyReview is asynchronous. Therefore, identifying place for review is not applicable.
2.7	Yes	Supported.
2.8	Yes	Supported.
2.9	Yes	There is no need to ensure whether the materials are received by review participants. Because EasyReview does not distribute them. Review Participants can connect to the system and see and upload the review materials.



**Table 9 : Evaluations of the Organization’s Review Procedure Specifications (cont.)**

<b>Item #</b>	<b>EasyReview</b>	<b>Comments</b>
2.10	Yes	Review process can be designed such that in EasyReview.
		<b>Individual Checking</b>
3.1	Yes	Review process can be designed such that in EasyReview.
3.2	Yes	The anomalies detected during this checking are stored in the database.
3.3	Yes	Review process can be designed such that in EasyReview.
		<b>Internal Review Meeting</b>
4.1	Yes	Every review participants can see their roles and tasks in a review.
4.2	Yes	E-mail notification is sent to all review participants.
4.3	Yes	When a task is finished by a review participant, effort on that task is also asked.
4.4	Yes	It is always possible to change and arrange the schedule.
4.5	Yes	Review process can be designed such that.
4.6	Yes	Supported automatically. Anomalies found by each reviewer are accessible to the review team after submitting.
4.7	Yes	Automated support of EasyReview eliminates the need for this.
4.8	Yes	Different defect types can be defined before or during the reviews. Also, the defect type named “Other” exists in the database.
4.9	Yes	All the anomalies can be examined whether it is accepted or rejected a true anomaly.
4.10	Yes	Supported.
4.11	Yes	Review process can be designed such that.
4.12	Yes	The system does not allow reviewers to discuss.
4.13	Yes	Review process can be designed such that.
4.14	Yes	Review process can be designed such that.
		<b>Closure</b>
5.1	Yes	Supported.
		<b>Metrics</b>
6.1	Yes	Efforts on the review tasks can be tracked.
6.2	Yes	The review report generated by the EasyReview calculates the total number of issues identified in the review.
6.3	Yes	The review report generated by the EasyReview calculates the total number of issues identified for each anomaly class in the review.

According to this table, EasyReview meets 43 of the total 46 specifications of the organization’s review procedure. Three of the specifications are not applicable to an online review tool. These rates reveal the success of EasyReview tool.

The primary limitation of the tool is the poor GUI for workflow definition.

Tasks in a workflow and the transitions between the tasks are displayed in the list form. It would have been easy to define a workflow with a GUI that has drag and drop capability and visual objects.

The tool has one more limitation. It has a simple project administration capability with which, time schedule can be assigned to the activities of a review process. Start and end dates of the review activities can be set. However, the tool does not have the capability to control whether the review participants comply with the schedule.

## CHAPTER 6

# CONCLUSION AND FUTURE DIRECTIONS

This thesis focused on the development of a Web based review tool that aims to overcome the shortcomings of traditional review process and inadequacies of the existing review tools.

### 6.1 Conclusion

To improve the effectiveness and the efficiency of software review, a variety of software review techniques have been developed. Computer support has been used to facilitate the software review process. Accordingly, several tools ranging from managing and delivering review documentation systems to complex collaborative systems have been developed. The tool proposed in this thesis help to overcome shortcomings of traditional review process and some of the inadequacies of existing review tools.

According to the results of experimental study, it has been proven that the tool provides support for review activities without face-to-face meeting, it facilitates the access to review materials and it has ability to define a new review process according to an organization's needs.

EasyReview provides automated support for document handling, data collection, review process and review administration.

For document handling, computer support allows browsing of documents with easy cross-referencing with each other. In addition, it allows annotation of documents to be linked to related parts in the document. In document handling, significant benefits of the tool are as follows:

- Soft copies or hard copies of the materials are not need to be distributed to the review participants. Review participants can access review materials and other supportive materials easily via a Web browser.
- Review of multiple document formats is supported.
- All document formats are supported for display if the viewers of the documents exist.
- Annotations can be done online. Reviewers only select the location on the documents and enter issues in a unified interface.
- Concurrent reviews are allowed; multiple users can review a document at the same time.
- All issues entered by a review participant can be seen by any review participant.
- Issues are colored based on the criticality level of them.
- Only the original versions of the materials are stored on the server, issues are stored in a database.

For data collection, computer support automatically gathers review data for analysis and allowing reviewers to concentrate on reviewing. In data collection, significant benefits of the tool are as follows:

- The need for an issue recorder and logger is eliminated with the help of automatic data collection.
- Detailed reports about the issues can be taken according to the given criteria.
- Efforts of the review participants for each review activity can be reported.

For review process, computer support allows defining new review processes according to the needs. In addition, it allows defining additional issue types and review roles, automates information flow throughout the process and eliminates review meeting. Significant benefits of the tool for review processes are as follows:

- With workflow capability, it supports all kinds of review processes.
- Review meetings are eliminated.
- New issue types and review roles are allowed.
- Status of the review process and review participants (whether they finished their tasks or not) can be seen.
- E-mail notification is supported for informing review participants about the process.
- Different roles can be assigned to a user in different review processes and he or she can work on multiple reviews at the same time.

For review administration, computer support makes management and conduct of the reviews easily. In review administration, significant benefits of the tool are as follows.

- Administration of the tool can be done remotely with a Web browser.
- Users are provided with username and passwords so that the system can control access to materials and issues.
- The tool supports two kinds of user group, the one with administrative privileges and the other with review activity privileges.
- The tool allows project creation. Users and supportive materials can be assigned to any project. Multiple reviews are allowed within a project and a time schedule can be assigned to the reviews.
- Menu for the review operations are displayed based on the user's privileges and status.

According to the results of experimental study, beside the significant contributions and capabilities, the tool has some drawbacks such as insufficient online help and poor Graphical User Interface for workflow definitions. However, these drawbacks can be overcome in a future work.

## **6.2 Future Work**

The tool developed in this thesis can be extended in several ways. First, validation of the tool can be done by comparing traditional reviews with the cases in which the tool is used. In order to determine the tool's effectiveness, an experimental study can be conducted to compare the effectiveness of the review participants in traditional reviews with effectiveness of the review participants who perform review activities using EasyReview.

The tool can be integrated with other tools such as project management tools, distributed meeting tools, metrics processing tools, report generation tools, and other workflow management tools.

For workflows definition, a comprehensive Graphical User Interface with drag and drop capability can be used. Then, it would be easy to create and customize workflow tasks. Furthermore, workflows can be imported from an external file such as an XML file.

User management of the tool can be integrated with directory services such as Microsoft Windows Active Directory.

Instead of built-in metrics, user-defined metrics can be supported.

In the current version, when an issue is entered by a review participant, it can be seen by other participants. In addition to this capability, private issue entering can be supported in order to prevent other reviewers to see the raised issues.

Finally, online help can be provided and additional document formats can be supported such as Microsoft Visio and Microsoft Project.

## REFERENCES

- Barnard H.J., & Collicott R.B., (1990, March/April). COMPAS: A development process support system. AT&T Technical Journal.
- Bisant, D.B., & Lyle, J.R., (1989, October). A two-person Inspection method to improve programming productivity. IEEE Trans. on Software Engineering, 15(10), pp.1294-1304.
- Brothers, L.R., Sembugamoorthy, V., & Muller M., (1990, October). ICICLE: Groupware for Code Inspections. Proceedings of the 1990 ACM Conference on Computer Supported Cooperative Work, pp. 169–181.
- Caivano, D., Lanubile, F., & Visaggio, G., (1998). Scaling up Distributed Software Inspections. Italy: University of Bari.
- DeMarco, T., (1982). Controlling Software Projects. N.Y.: Yourdon Press.
- Fagan, M.E., (1976). Design and code inspections to reduce errors in program development. Reprinted from IBM Systems Journal, Vol. 15, No 3.
- Gilb, T., & Graham, D., (1993). Software Inspection. Addison-Wesley.
- Gintell, J.W., et.al., (1993, September). Scrutiny: A collaborative inspection and review system. Proceedings of the Fourth European Software Engineering Conference.
- Harjumaa, L., Hedberg, H., & Ternoven, I., (2000). A Path to Virtual Software Inspection. Finland: University of Oulu.
- Hedberg, H., & Harjumaa, L., (2001). Virtual Software Inspections for Distributed Software Engineering Projects. Finland: University of Oulu.

Humphrey, W., (1989). Managing the Software Process. N.Y.: Addison-Wesley.

IEEE Standard 1028, (1997). IEEE Standard for Software Reviews and Audits. The Institute of Electrical and Electronics Engineers, Inc. Piscataway, NJ.

IEEE Standard 610.12, (1990). IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Electronics Engineers, Inc. Piscataway, NJ.

Johnson, P.M., (1994a, May). An instrumented approach to improving software quality through formal technical review. Proceedings of the 16th International Conference on Software Engineering, pp. 113-122, Sorrento, Italy.

Johnson, P.M., (1994b). Supporting Technology Transfer of Formal Technical Review Through A Computer Supported Collaborative Review System. Honolulu: University of Hawaii.

Johnson, P.M., & Tjahjono, D. (1997, May). Assessing software review meetings: A controlled experimental study using CSRS. 19th International Conference on Software Engineering, Boston, Massachusetts.

Knight, J.C., & Meyers, E.A., (1991, July). Phased inspections and their implementation. Software Engineering Notes, 16(3), pp.29-35.

Laitenberger, O., & DeBaud, J., (1998). An Encompassing Life-Cycle Centric Survey of Software Inspection. International Software Engineering Research Network, 98-32.

Macdonald, F., (1998). Computer Supported Software Inspection. A thesis submitted to The University of Strathclyde, Glasgow, Scotland, UK.

Macdonald, F., & Miller, J., (1997) A Comparison of Tool-Based and Paper-Based Software Inspection. International Software Engineering Research Network, 98-17.

Mashayekhi, V., Feulner, C., & Riedl, J., (1994, December). CAIS: Collaborative Asynchronous Inspection of Software. SIGSOFT Software Engineering Notes 19(5), pp. 21-34.

McConnel, S., (1993). Code Complete. Chapter 24, Microsoft Press.



Mills, H.D., Dyer, M., & Linger, R.C., (1987). Cleanroom software engineering. IEEE Software, pp. 19-25.

Murphy, P., & Miller, J., (1997, July). A process for asynchronous software inspection. Proceedings of the 8th International Workshop on Software Technology and Engineering Practice, pp. 96–104.

Perlman, G. (2002). Web-Based User Interface Evaluation with Questionnaires. Retrieved December 12, 2002, from <http://www.acm.org/~perlman/question.html>.

Perry, D.E., Porter, A., Votta, L.G., & Wade, M.W., (1998). Evaluating Workflow and Process Automation in Wide-Area Software Development. Retrieved April 2, 2004, from <http://www.lsd.cs.uga.edu/activities/NSF-workflow/s.html>.

Perry, E.D., et.al., (2002). Reducing Inspection Interval in Large-Scale Software Development. IEEE Transactions on Software Engineering, Vol. 28, No. 7, pp. 695.

Putala, M., & Tervonen, I., (1997). Inspecting Postscript documents in an object-oriented environment. 5th European Conference on Software Quality.

Sapsomboon, B., (1999). Software Inspection and Computer Support. A thesis submitted to University of Pittsburg, Pennsylvania, USA.

Schneider, G. M., Martin, J., & Tsai, W. T., (1992). An experimental study of fault detection in user requirements documents. ACM Transactions on Software Engineering and Methodology, pp. 188–204.

Stein, M., Riedl, J., Harner, S.J., & Mashayekhi, V., (1997). A case study of distributed, asynchronous software inspection. Proceedings of the 19th International Conference on Software Engineering, pp. 107–117.

Thompson, C., & Riedl, J., (1995). Collaborative asynchronous inspection of software using Lotus Notes. Technical Report TR 95-047, Computer Science Department, University of Minnesota.

van Genuchten, M., Cornelissen, W., & van Dijk, G. (1997). Supporting inspections with an electronic meeting system. Proceeding of the Thirtieth Hawaii International Conference on System Sciences (HICSS'97), Vol. 2.

Venners, B., (2003). The Demand for Software Quality, A Conversation with Bertrand Meyer. Part I, Retrieved April 2, 2004, from

<http://www.artima.com/intv/serious.html>.

WfMC Specification, (1995). The Workflow Reference Model. Workflow Management Coalition, Hampshire, UK.

Wheeler, D.A., Brykczynski, B., & Meeson, R.N., (1996). Software Inspection - An Industrial Best Practice. IEEE Computer Society Pres.

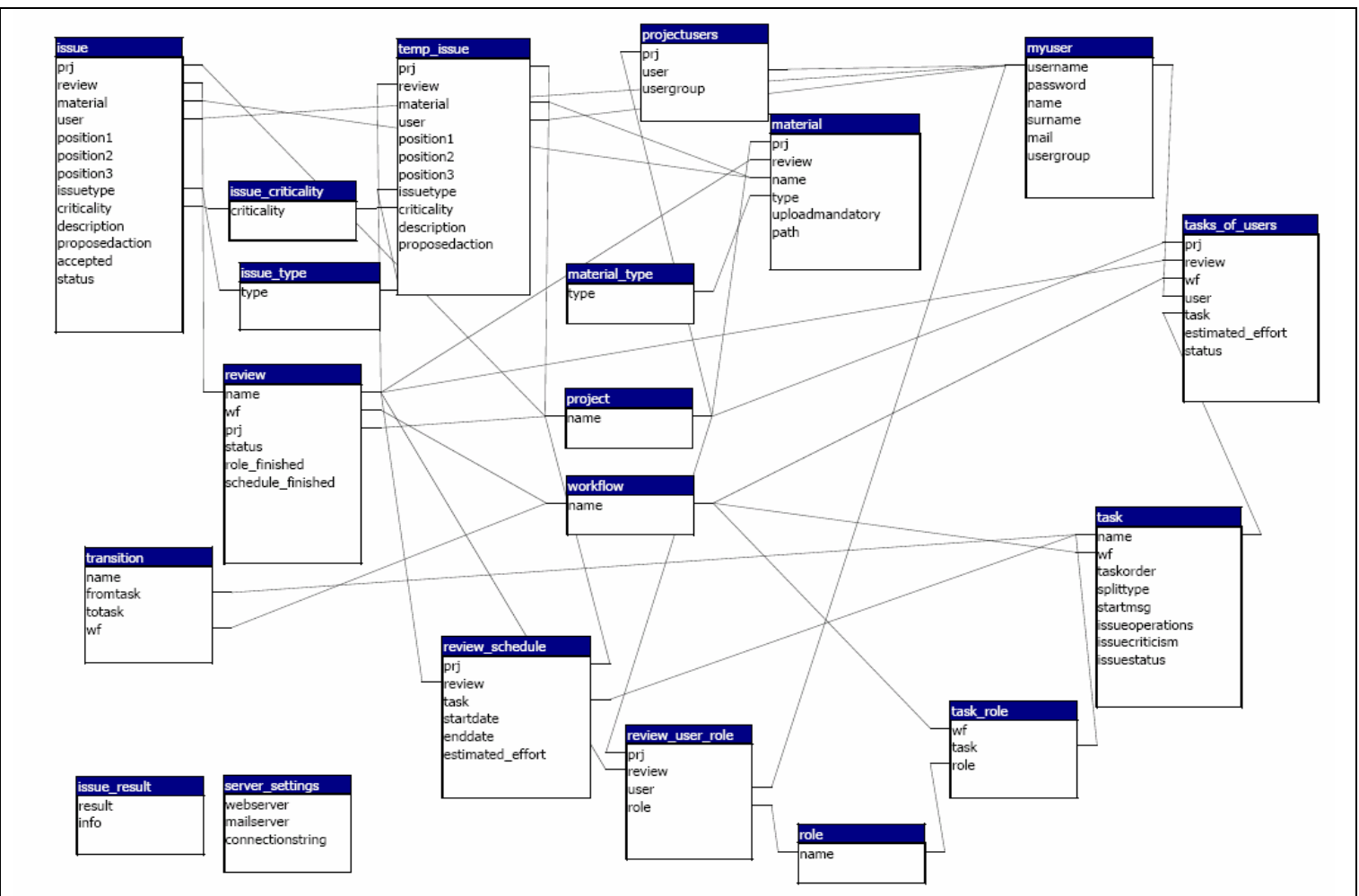
Yanbaş, E., (2002). A Web-based Tool for Formal Technical Asynchronous Review. A thesis submitted to Middle East Technical University, Ankara, Turkey.

Yourdon, E., (1989). Structured Walkthroughs. N.Y.: Yourdon Press.

Yourdon, E., (1997). Death March. Prentice Hall.

## **APPENDICES**

# ENTITY RELATIONSHIP DIAGRAM



# COMBINED CLASS DIAGRAMS

## 1. Combined Class Diagrams

The combined class diagram for a Java source file or class file displays the class in the center of the diagram with the relationships surrounding it.

Relationships are displayed in four areas: left, right, above, and below. Relationships to the left of the central class are associations and relationships to the right are dependencies. Associated extended classes and implemented interfaces are displayed above the central class. Extending and implementing classes are displayed below the central class.

Associations and dependencies are further organized into three areas: upper, middle, and lower. The upper area shows reverse relationships, such as classes that are associated or dependent on the central class. The middle area contains mixed relations. The lower area contains classes that the central class has an association to or dependency on.

## 2. Class Diagrams for “tez” Package’s Classes

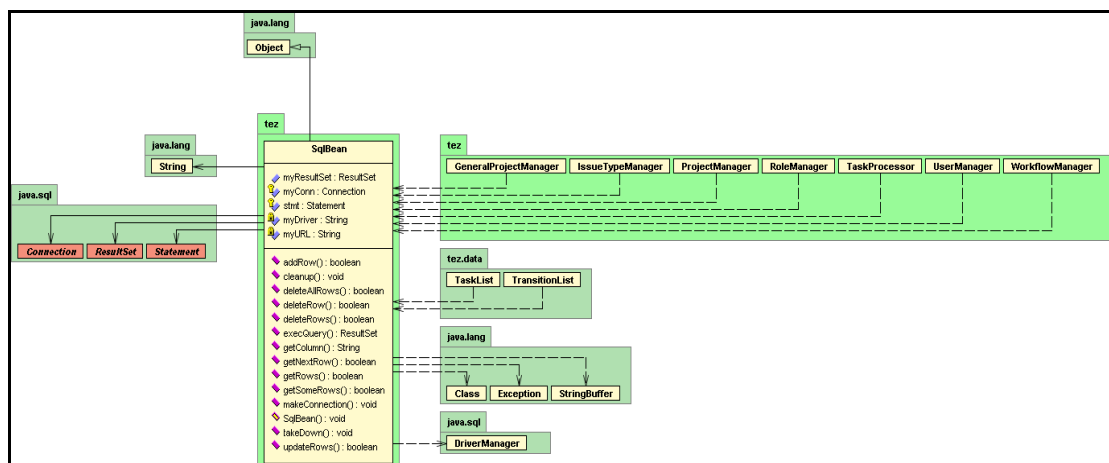


Figure 26 : Class Diagram for SqlBean Class

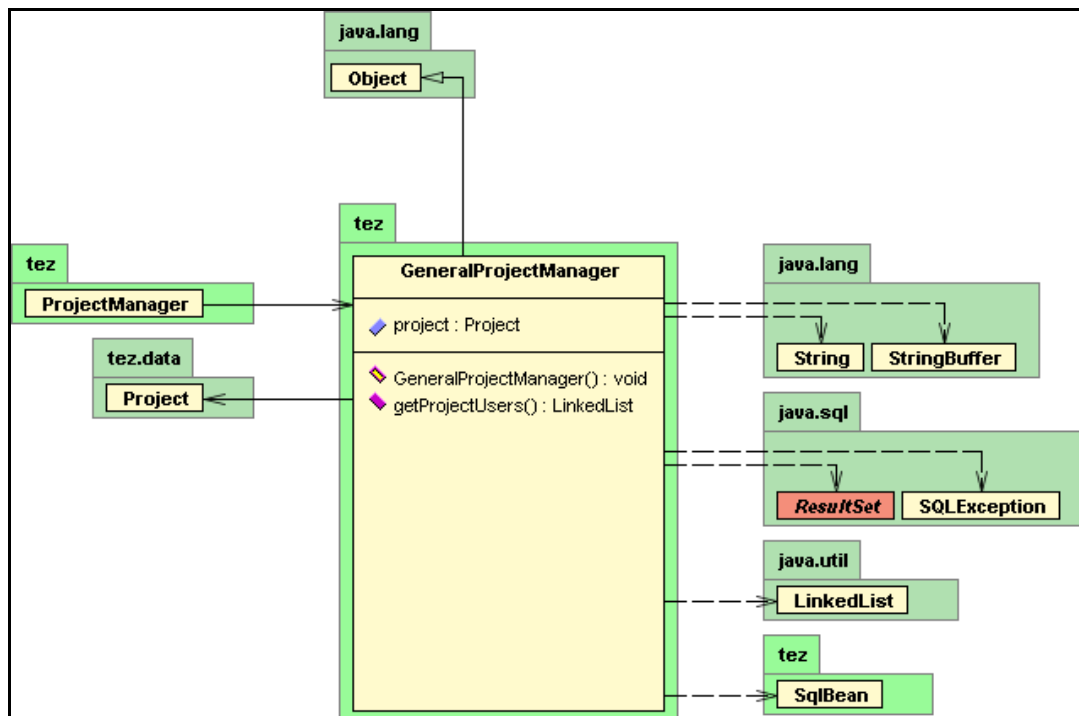


Figure 27 : Class Diagram for GeneralProjectManager Class

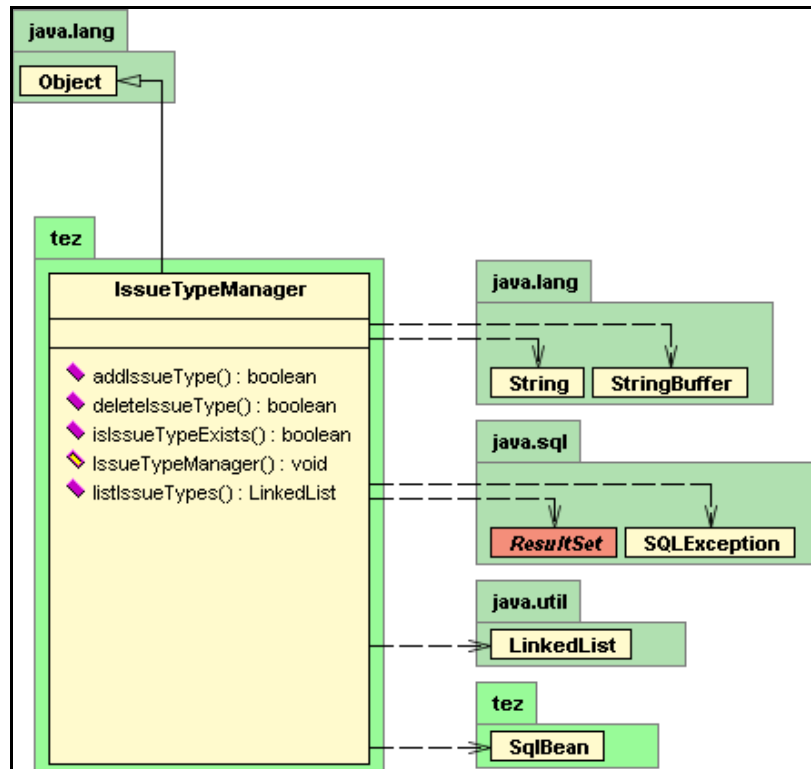


Figure 28 : Class Diagram for IssueTypeManager Class

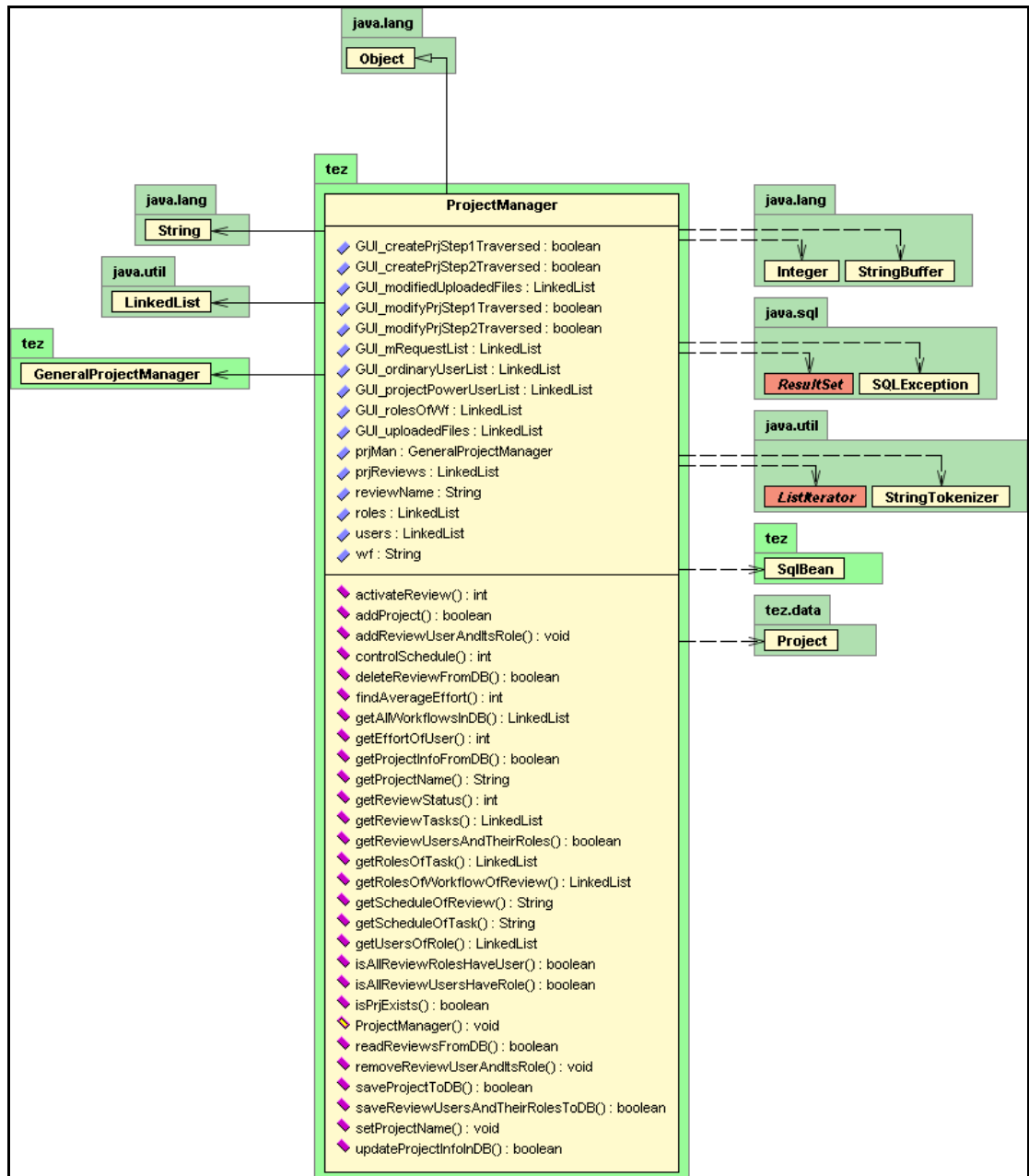


Figure 29 : Class Diagram for ProjectManager Class

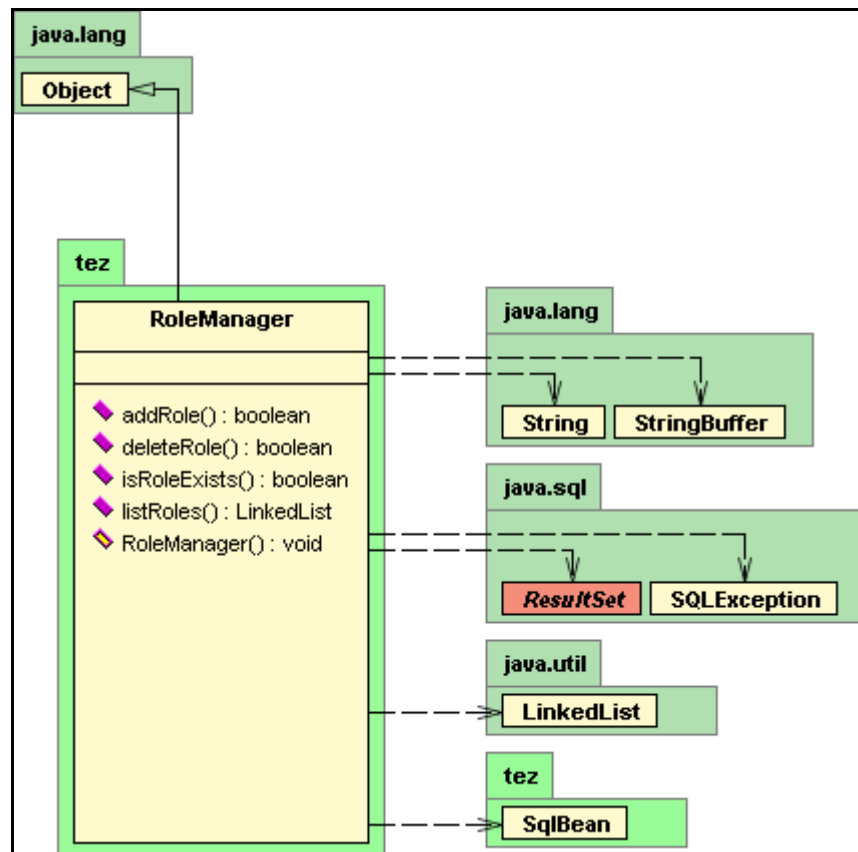


Figure 30 : Class Diagram for RoleManager Class

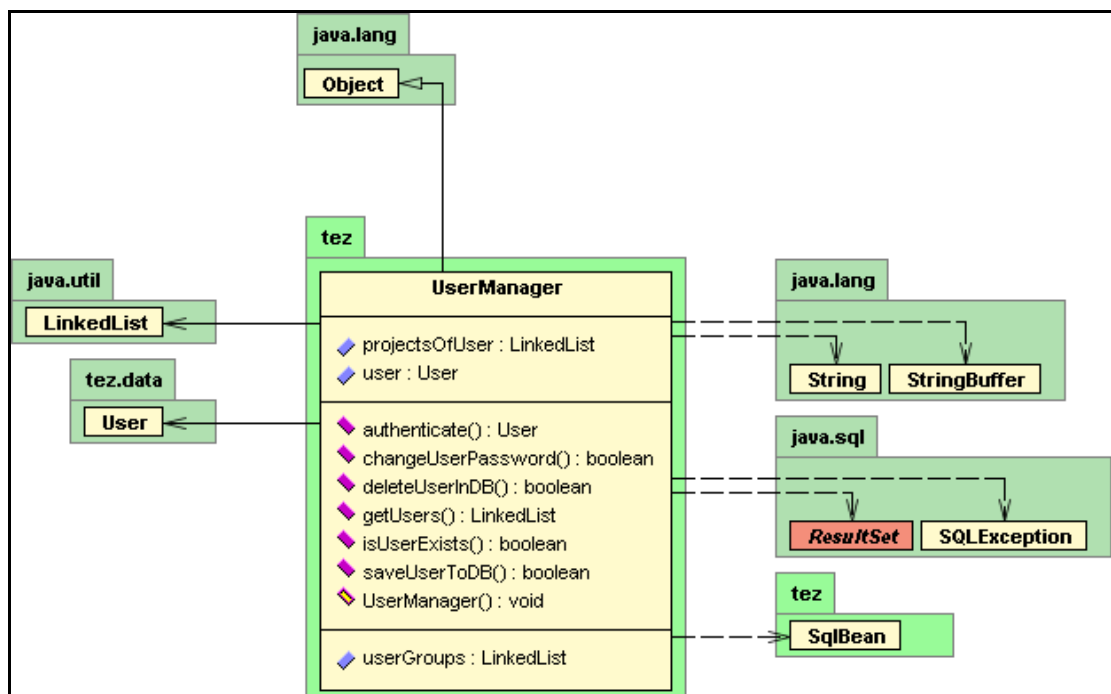


Figure 31 : Class Diagram for UserManager Class



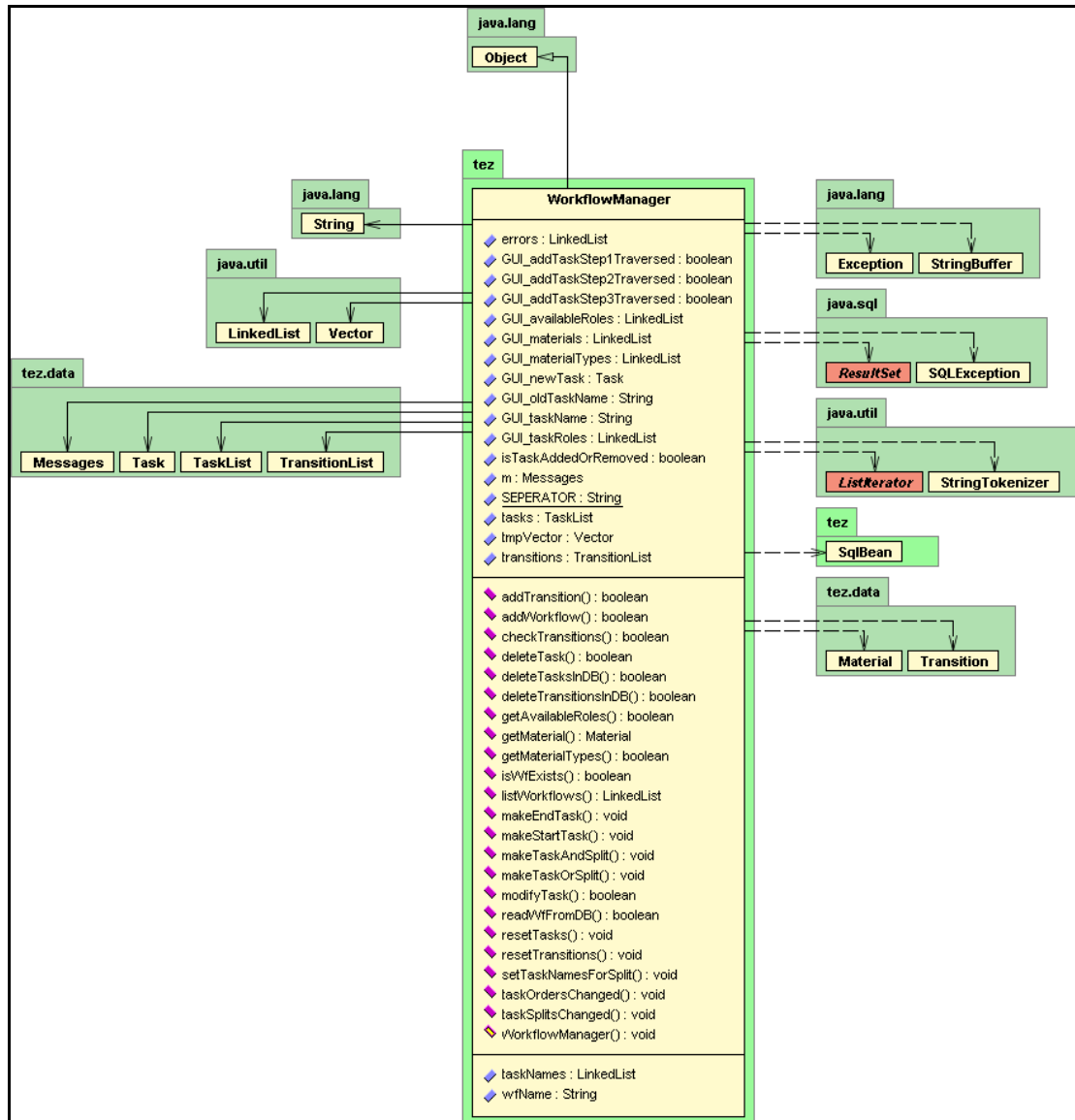


Figure 32 : Class Diagram for WorkflowManager Class

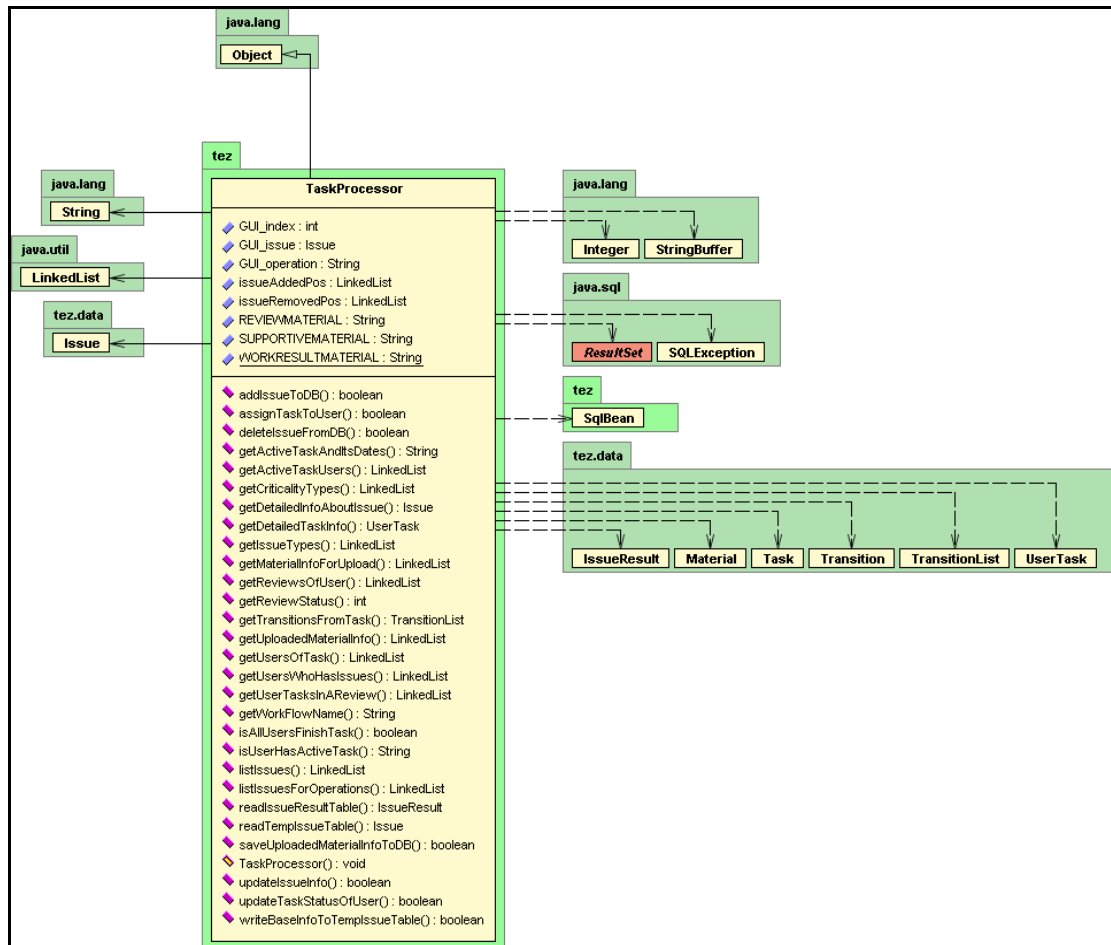


Figure 33 : Class Diagram for TaskProcessor Class

### 3. Class Diagrams for “tez.data” Package’s Classes

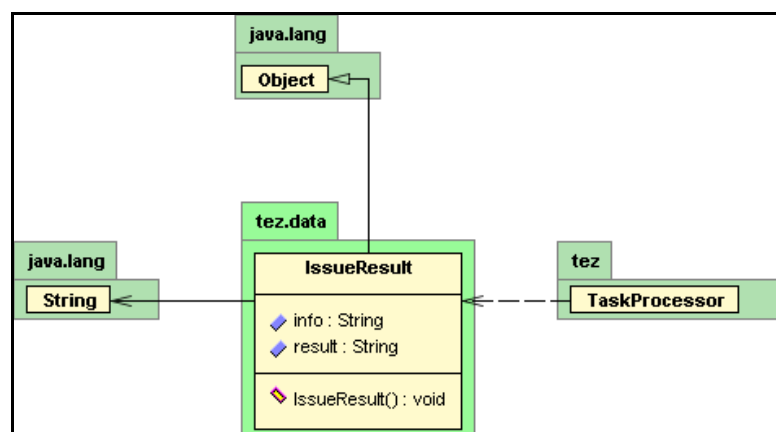


Figure 34 : Class Diagram for IssueResult Class

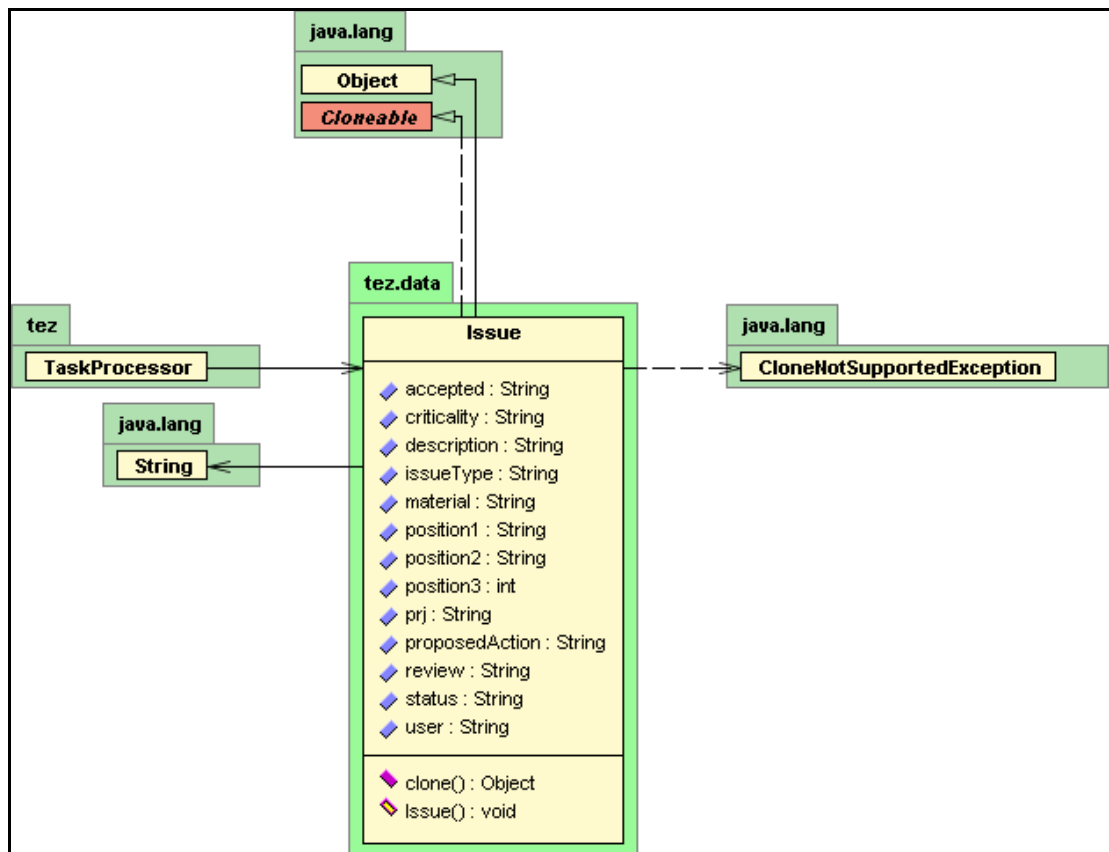


Figure 35 : Class Diagram for Issue Class

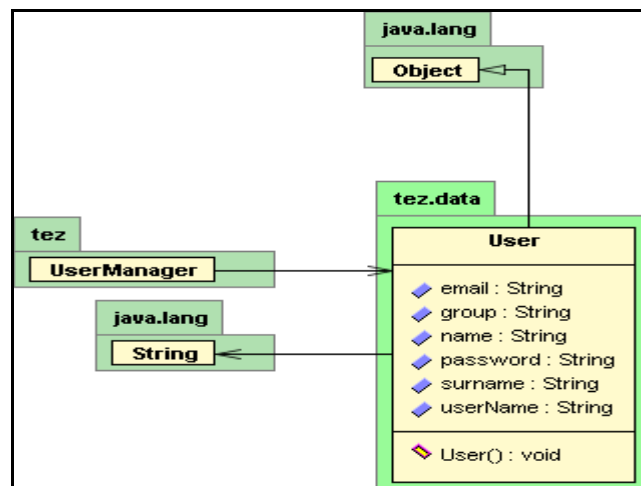


Figure 36 : Class Diagram for User Class

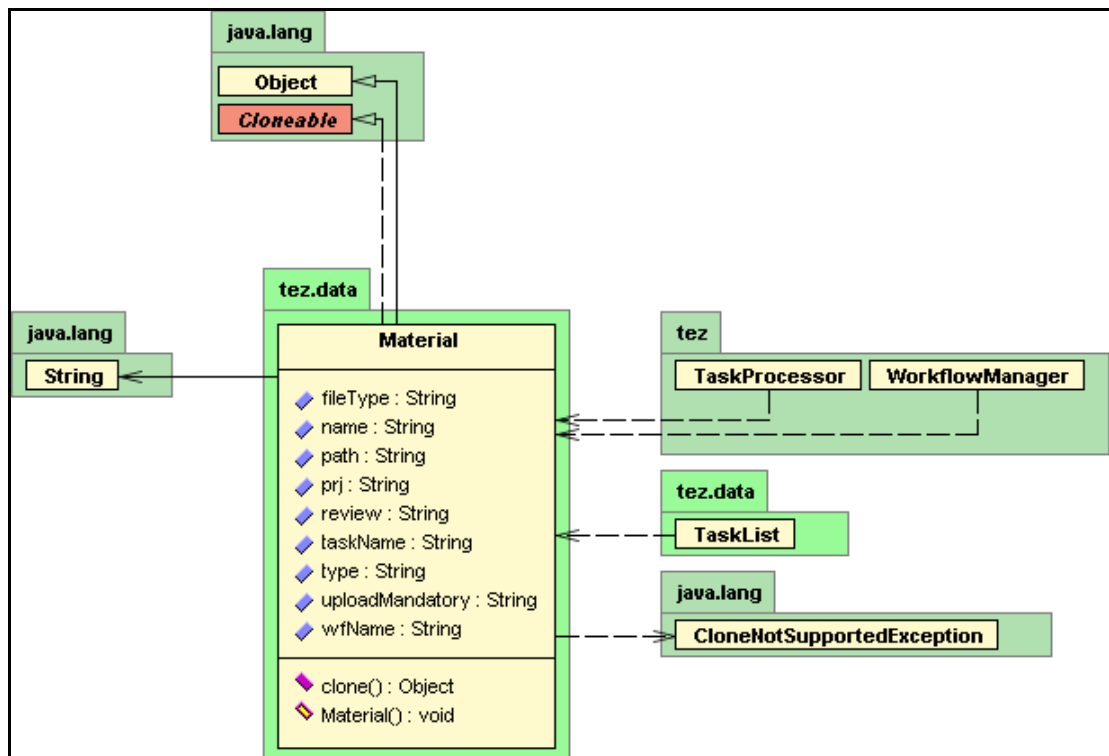


Figure 37 : Class Diagram for Material Class

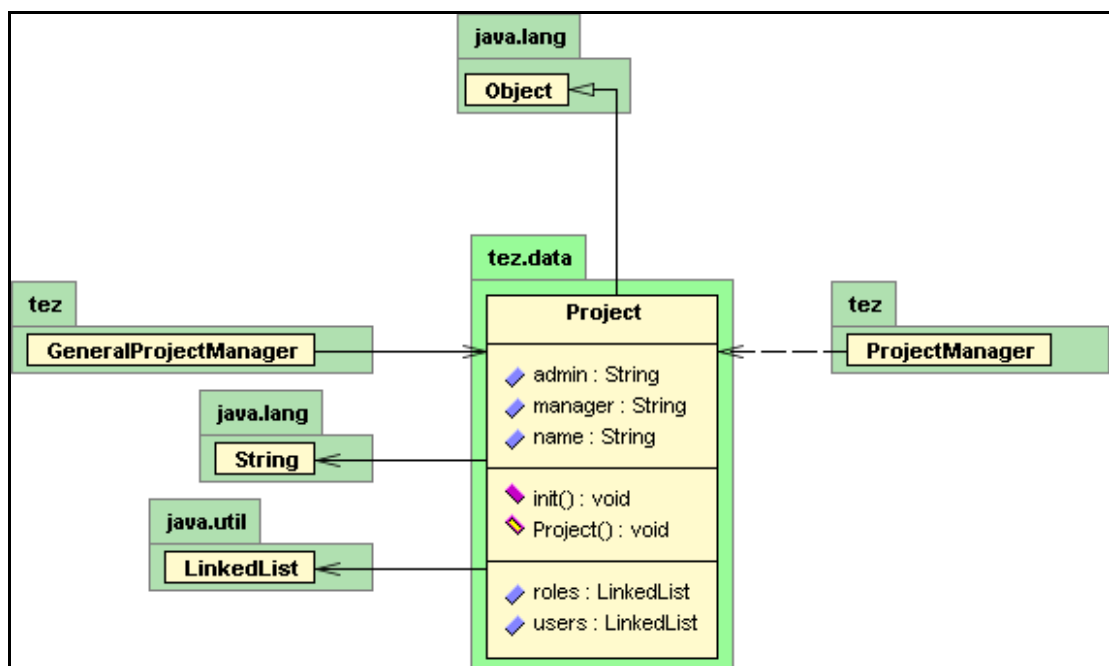


Figure 38 : Class Diagram for Project Class

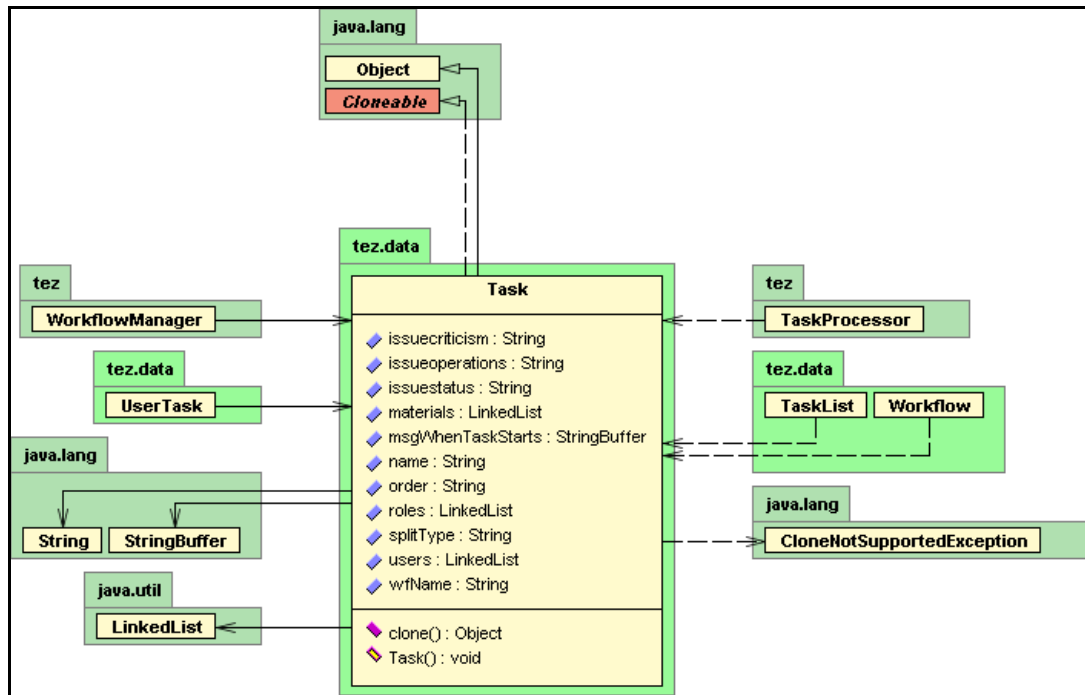


Figure 39 : Class Diagram for Task Class

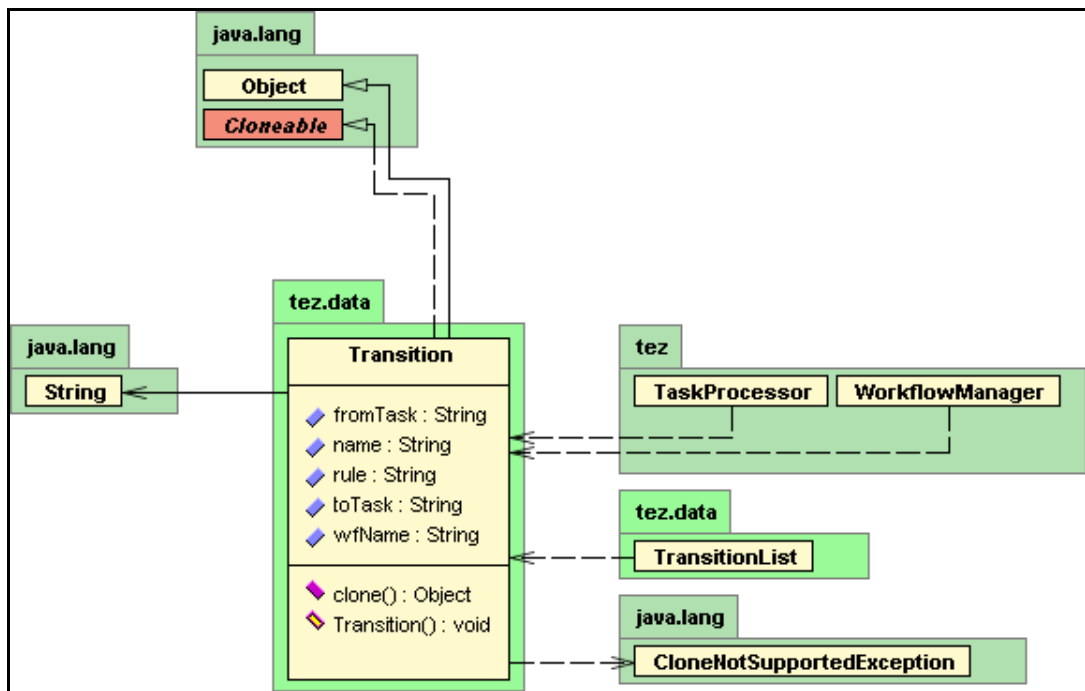


Figure 40 : Class Diagram for Transition Class

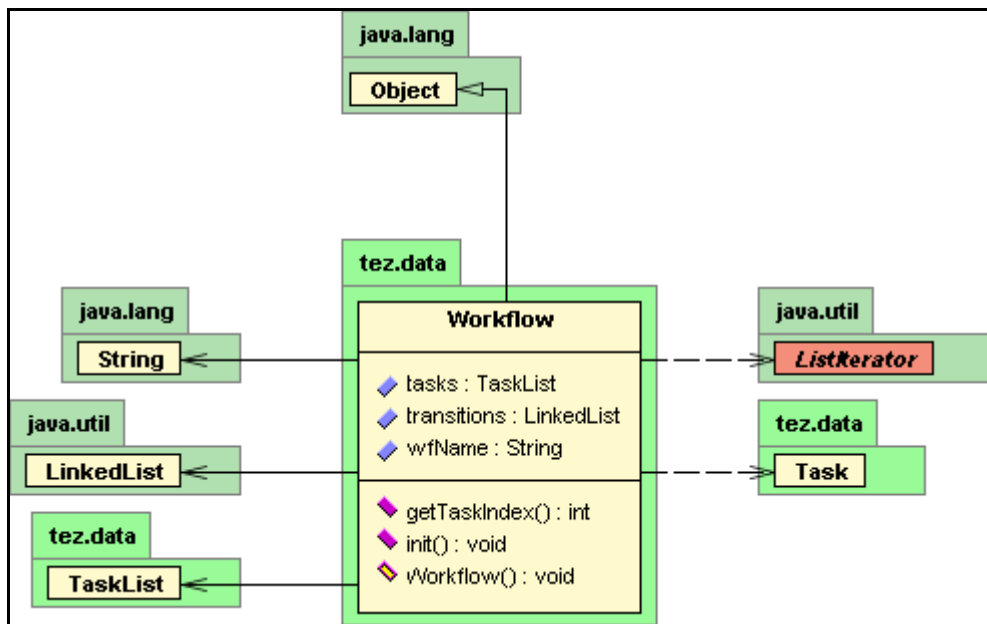


Figure 41 : Class Diagram for Workflow Class

## APPENDIX – C

# SAMPLE USER INTERFACES

**Figure 42 : Interface for Adding a New User**

**Figure 43 : Interface for Customizing a Review**

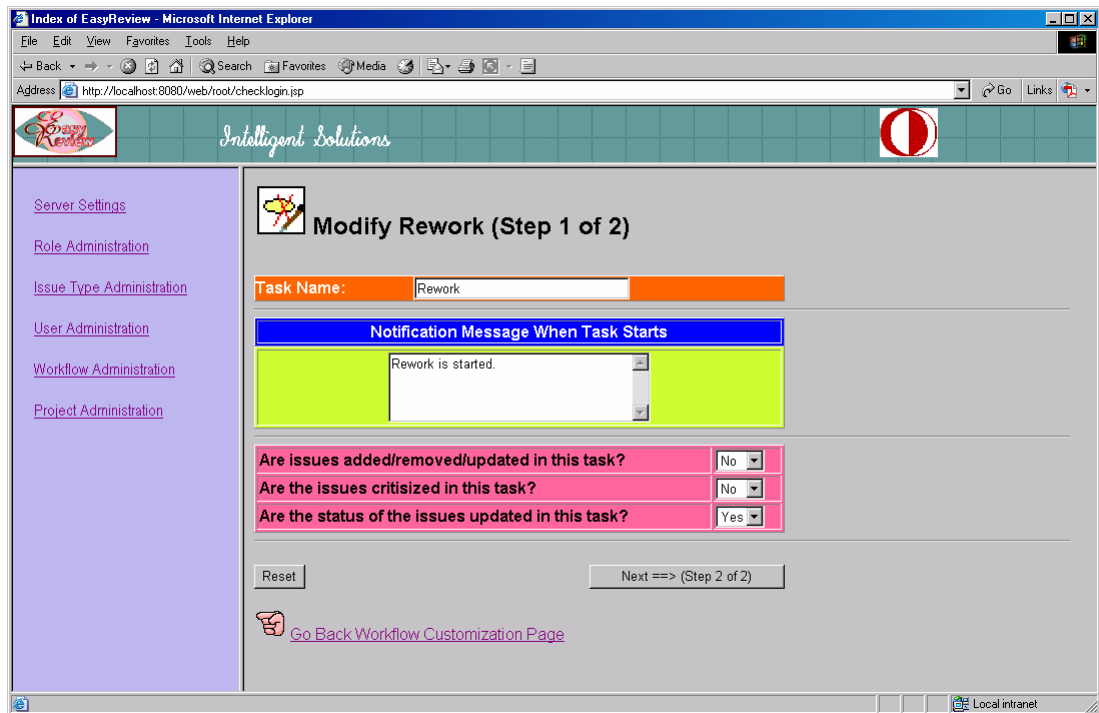


Figure 44 : Interface for Modifying a Task within a Workflow

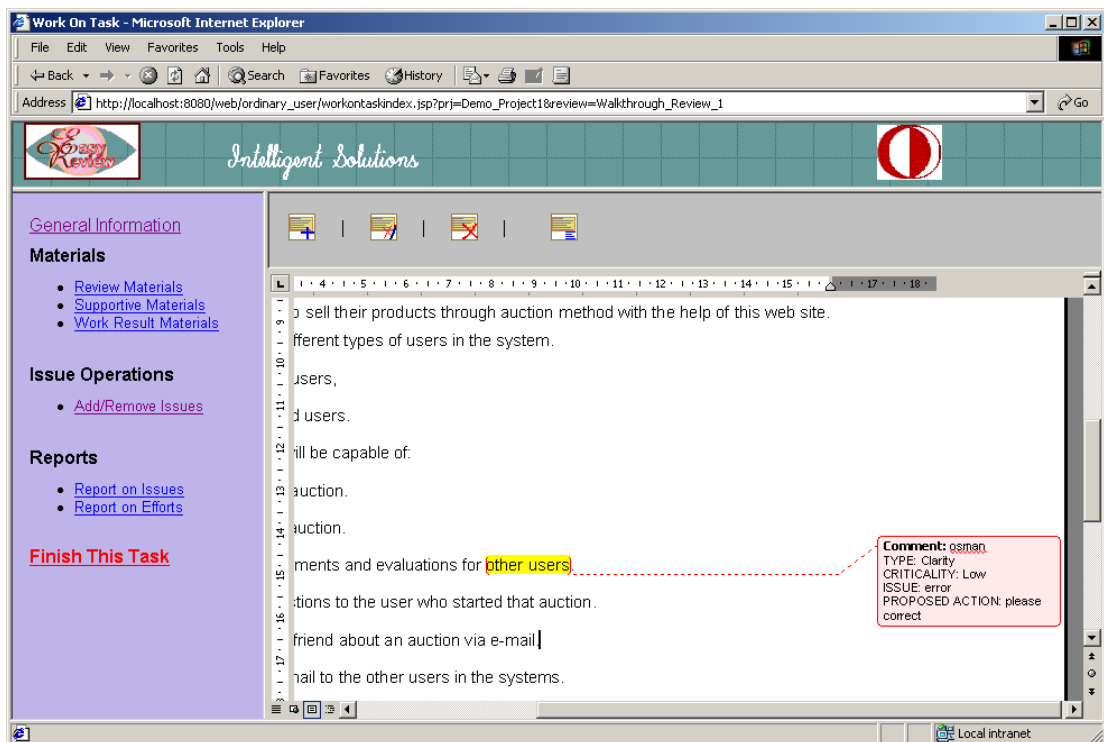


Figure 45 : Interface for Adding and Removing Issues (Microsoft Word Document)



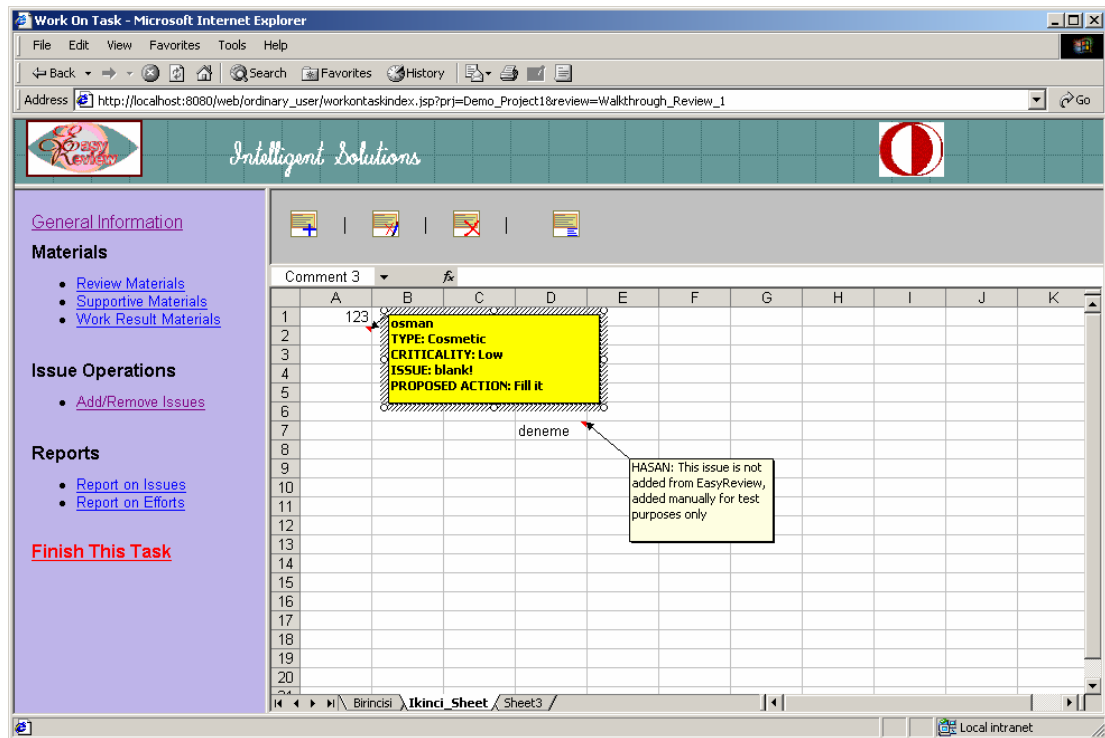


Figure 46 : Interface for Adding and Removing Issues (Microsoft Excel Spreadsheet)

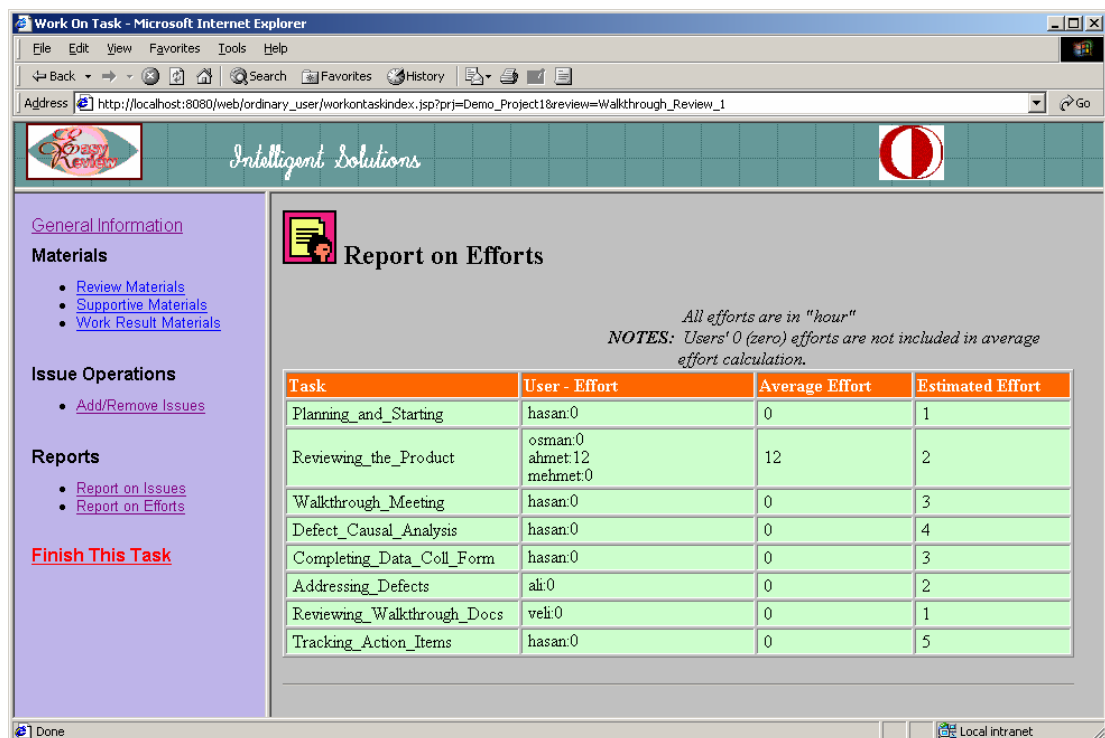


Figure 47 : Interface for Taking Report on Efforts

## APPENDIX D

# PARTICIPANT CONSENT FORM

### Participant Consent Form

Thank you for your participation in the experimental study of EasyReview tool.  
This study will test how usable and functional the tool is.

You will be asked to complete a pre-test questionnaire and an evaluation questionnaire. During the experiment, you will participate in a review using the tool.

All your personal information will be kept highly confidential, and be used for research purpose only.

After you have read the above description of my experiment, if you still agree to participate in the test, please sign below:

Participant: \_\_\_\_\_

Date: \_\_\_\_/\_\_\_\_/\_\_\_\_

Signature: \_\_\_\_\_

## APPENDIX E

# PRE-TEST QUESTIONNAIRE

### Pre-Test Questionnaire

**Name** : \_\_\_\_\_

**Surname** : \_\_\_\_\_

**Role in the Experimental Review** : \_\_\_\_\_

1. Are you familiarized with review, workflow, review process, and formal review terms? (Do not explain; answer “yes/no” for each term)

\_\_\_\_\_

2. Have you ever participated in a formal review? (Once, less than five, more than five, etc)

\_\_\_\_\_

3. What operating system do you use most often? (Windows, Mac, Linux, etc)

\_\_\_\_\_

4. Which Web browser do you prefer to use? (Microsoft Internet Explorer, Netscape Navigator, etc)

\_\_\_\_\_

Thank you for your participation

## APPENDIX F

# EASYREVIEW EVALUATION QUESTIONNAIRE

(This questionnaire was retrieved and adapted from Perlman's Web site [Perlman, 2002])

**Table 10 : Questionnaire for Practical Heuristics for Usability Evaluation**

<b>LEARNING</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>NA</b>
1. Help and Documentation Design for use without documentation. Provide easy-to-use task-oriented help.	bad								good
2. Adopt the User's Viewpoint Speak the user's language (avoid jargon). Make use of existing knowledge (familiar mental models).	bad								good
3. Simple and Natural Dialogue Avoid extraneous information, steps, actions. Information should be in a logical, natural order.	bad								good
4. Design for Advancement Provide shortcuts (quick keys, customization).	bad								good
<b>ADAPTING TO THE USER</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>NA</b>
5. Provide Maps and a Trail Give the user a way to preview where to go, what will happen. Give the user a way to review / return-to previous contexts.	bad								good
6. Show the User What is (Not) Possible Provide affordances to indicate what can be done.	bad								good
7. Intuitive Mappings Design good response compatibility between controls/actions.	bad								good
8. Minimize Memory Load Remove the need to remember across dialogues. Provide multiple views for easy comparisons.	bad								good
9. Consistency in the System and to Standards Make sure the same term/action has one meaning. When there is no better way, conform to a standard.	bad								good

Table 10 : Questionnaire for Practical Heuristics for Usability Evaluation (cont.)

FEEDBACK AND ERRORS		1	2	3	4	5	6	7	NA	
10. Feedback Provide timely feedback about all processes, system status.	bad								good	
11. Prevent Errors Make it difficult to make errors.	bad								good	
12. Error Messages Diagnose the source, cause of a problem and suggest a solution.	bad								good	
13. Clearly Marked Exits and Error Recovery Make sure the user can get out of an undesirable state easily. Design assuming that people will make errors and need to recover previous states	bad								good	
14. Negative Comments About the Tool Write comments regarding the aspects of the tool that must be improved										
15. Positive Comments About the Tool Write positive aspects of the tool										

## APPENDIX G

# USABILITY EVALUATION RESULTS

**Table 11 : Usability Evaluation Results**

	PARTICIPANTS											
		1	2	3	4	5	6	7	8	min	max	Median
QUESTIONS	1	2	2	2	2	3	3	4	4	2	4	2
	2	4	4	5	6	4	5	6	6	4	6	5
	3	5	6	7	7	7	7	7	6	5	7	7
	4	5	6	6	7	6	5	4	5	4	7	5
	5	6	6	7	7	7	7	7	6	6	7	7
	6	6	7	7	6	7	7	7	6	6	7	7
	7	5	6	5	6	7	7	5	6	5	7	6
	8	6	7	7	7	6	7	7	7	6	7	7
	9	6	7	7	7	7	7	7	7	6	7	7
	10	6	7	6	7	7	7	6	6	6	7	6
	11	7	7	7	7	7	7	6	7	6	7	7
	12	7	7	7	7	7	7	7	7	7	7	7
	13	6	7	6	7	7	7	6	6	6	7	6

## 1. Negative Comments

- There is not any online help in the Web pages related to the operations.
- It is hard to define and verify a workflow because of the GUI restriction.
- It does not warn the review participants when due date of a task is over.

## 2. Positive Comments

- Workflow capability enables an organization to customize a new review process.

- With this tool, there is no need to perform review meetings; this saves time.
- The documents can be reviewed in an interactive way.
- This tool can be used for different purposes such as document review and task assigning.
- It is user-friendly software.
- Easy to use, learn.
- It is an easy specification of sending e-mail after finishing a task.
- I recommend this tool to my friends.
- Icons on the Web pages are explanatory.
- It is easy to annotate a review document.
- Software is scalable from small projects to very large enterprise level projects.
- Reviewers can only focus on reviewing the documents; the tool eliminates overheads such as distributing documents, sending mail, etc.

## **REVIEW PROCESS OF THE ORGANIZATION**

### **1. SCOPE**

This document defines the minimum procedure for performing a review.

Reviews provide a timely and objective means of ensuring quality by assessing the software products during preparation for compliance with acquirer requirements and project standards.

Reviews provide a timely and objective means of ensuring compliance by assessing the products of each development phase for compliance with customer requirements and the project software development standards.

Reviews are used to inspect the software documents during the development phases of the software products.

The review process is used to ensure that the work of the previous phase has been satisfactorily completed by ensuring that the product is technically correct, it meets the requirements of the previous phase, it forms a suitable baseline for the subsequent phase and that the plans for the next phase will lead to a satisfactory implementation of that phase.

### **2. GENERAL REQUIREMENTS**

Reviews shall be used to inspect software products prepared during the software development life cycle. Record of these reviews will be documented on Individual Review Records and Review Summary Reports.

The review revolves around an inspection team, which consists normally of between three and five members.

A Moderator shall be appointed and responsible for organizing the review. Two



to four other members of the team shall be selected to perform the inspection.

### **3. REVIEW PROCESS AND MEMBERS**

A review is a formal method for finding errors in products during their preparation. The application of the following procedure shall be adopted unless an acceptable alternative procedure exists.

#### **3.1. Review Team**

The review revolves around an inspection team, which consists normally of between three and five members.

- The Moderator: The Moderator shall be responsible for organizing the review and ensuring each member has a copy of or has access to the necessary document(s) to be reviewed. The Moderator shall act as the chairperson in the review. He or she shall ensure that the review maintains its flow. The Moderator shall also ensure the discussion does not stray from its purpose and that the members maintain their assigned roles.
- The Members: Two to five other members of the team shall be selected to perform the inspection. One of them is the author of the document or product and the others are reviewers.

#### **3.2. Review Process**

- Planning: The Moderator, on receiving the document for inspection organizes the location and time for the review and distributes the document to the members for inspection.
- Overview: The Moderator shall prepare a short summary describing where the software product fits into the project. This shall be provided to the members with the software product to be inspected.
- Preparation: The members should work alone to familiarize themselves with the material and discover possible defects to present to the review.

- **Review:** The review shall be attended by the members and the Moderator. The review involves a read through of the document or product individually. All presumed defects are noted by the reviewers on “Individual Review Record” documents. No discussion about whether it is really a defect is allowed, or any discussions about possible solutions. The object of the review is to identify defects only. The Moderator cannot raise defects although he or she may make observations at the end of the review.
- **Rework:** After the review, the defects list shall be assigned to author for "repair".
- **Follow Up:** The Moderator is responsible for ensuring that defects have been corrected, generally by a further iteration of the “Rework” but involving only the moderator and the author.
- **Review Reporting:** All Individual Review Records are collected and The Review Summary Report will be completed at the conclusion of the review. The Moderator is responsible for reporting activities.

**Table 12 : Review Summary Report**

REVIEW SUMMARY REPORT			
Initiation Date:		Ending Date:	
Project Name:		Product Name:	
Review Participants	Moderator:		
	Author:		
	Reviewers:	1.	3.
		2.	
Review Material:			
Review Meeting Time:		Review Meeting Location:	
Review Meeting Duration:		Total Individual Prep.Time:	
Appraisal Status:	ACCEPTED ___ as is ___ with minor rework	NOT ACCEPTED ___ with rework verification ___ review again	
Actual Rework: (if applicable)	Rework Effort:	Rework Completion Date:	
ISSUES			
Issue Type Number	Issue Type Description	Number of Issues Identified	
1	Editorial (grammar, spelling, punctuation, etc)		
2	Not conforming to standards		
3	Checklist item incomplete/missing/inadequate		
4	Ambiguous		
5	Incorrect		
6	Inconsistent		
7	Other (please explain each)		
TOTAL:			
SIGNATURES			
Moderator:		Author:	
Reviewers: 1:			
2:			
3:			

[illegible]

## APPENDIX – I

### SPECIFICATIONS OF REVIEW PROCEDURE

(These specifications were excerpted from Yanbaş's study [Yanbaş, 2002].)

**Table 14 : Specifications of Review Procedure**

Item #	Specifications
	<b>General</b>
1.1	Review process examines whether the product satisfies its specifications and conforms to applicable standards, guidelines, and procedures.
1.2	Review process identifies deviations from standards and specifications and collects data such as anomaly and effort data.
1.3	Inputs are a document or a product to be reviewed and materials such as standards, guidelines, and procedures.
1.4	Document or product to be reviewed is generated internally by personnel or externally by a subcontractor.
1.5	Outputs are Review Record, Individual Review Log and Review Meeting Log.
1.6	Entry condition is the initiation of the Review Record.
1.7	Exit condition is the close of the Review Record.
1.8	The roles in the review process are Review Leader, Reviewer, Recorder, and Author.
1.9	Review Record is created by Author/ Review Leader and delivered to reviewers.
1.10	Individual Review Log is created by reviewers and delivered to Review Leader.
1.11	Review Meeting Log is created by Recorder and delivered to project file.
1.12	Review Leader is responsible for administrative tasks pertaining to the review.
1.13	Reviewer is responsible for identifying and describing anomalies in the product.
1.14	Recorder is responsible for documenting anomalies and recommendations found by the review team. One of the reviewers can act as the recorder.
1.15	Author is responsible for making the product ready for the review and for performing any rework to resolve issues identified.
	<b>Preparation</b>
2.1	When a document or product is completed, its author initiates a review process using a Review Record form and forwards the record to the Review Leader identified in the project plan.
2.2	The Review Leader identifies the review team.
2.3	The Review Leader assigns roles and responsibilities to review team members.
2.4	The Review Leader identifies materials (standards, guidelines, etc.).
2.5	The Review Leader identifies review time.
2.6	The Review Leader identifies review place.
2.7	The Review Leader fills out initiation date, initiated by, project name, product name, product description, number of pages/ FPs/ LOCs, role assignments, review materials fields of the Review Record.
2.8	The Review Leader distributes the Review Record, the Individual Review Log, the product to be reviewed and all supportive materials to the participants.
2.9	The Review Leader ensures that all participants have received all available documentation and information related to the review.
2.10	In case the review team is composed of a single reviewer who acts as the Review Leader, s/he receives documentation from the author and performs individual checking without any preparation other than obtaining the review materials.

**Table 14 : Specifications of Review Procedure (cont.)**

Item #	Specifications
	<b>Individual Checking</b>
3.1	Each review team member examines the product against the appropriate checklist and any applicable standards and procedures, prior to review meeting.
3.2	Anomalies detected during this checking are recorded in Individual Review Log.
3.3	If the Review Leader is the only reviewer, s/he uses Review Record instead of Individual Review Log to document the review findings.
	<b>Internal Review Meeting</b>
4.1	The Review Leader introduces the participants and describes their roles.
4.2	The Review Leader states the purpose of the review and reminds the reviewers to focus their efforts to anomaly detection, not resolution.
4.3	The Review Leader gathers individual preparation times from the reviewers, and the recorder writes the total in the Review Meeting Log.
4.4	The Review Leader may reschedule the meeting if s/he concludes that the reviewers are not adequately prepared.
4.5	The Review Leader presents the product to the review team.
4.6	The review team members scan their Individual Review Logs and report the anomalies they found to the review team.
4.7	The Review Leader focuses on creating a unified anomaly list and the recorder enters each anomaly, location, description and classification on Review Meeting Log.
4.8	If a reviewer reports an anomaly not defined in the checklist, then the anomaly is listed at the end of the checklist as a different type of issue.
4.9	If there is a disagreement about an anomaly, the potential anomaly is logged and marked by the recorder for resolution at the end of the meeting.
4.10	At the end of the review meeting, the Review Leader has the anomaly list reviewed with the team to ensure its completeness and accuracy.
4.11	The Review Leader allows time to discuss every anomaly where disagreement occurred.
4.12	The Review Leader does not allow the discussion to focus on resolving the anomaly but on clarifying what constitutes the anomaly.
4.13	To close the review meeting, an exit decision is taken to determine if the product meets the review criteria.
4.14	The review team identifies the product disposition as one of the following: <i>Accept with no or minor rework/ Accept with rework verification/ Review</i>
	<b>Closure</b>
5.1	The Review Leader is responsible for gathering documentation related to the review, completing the Review Record, and handing it to the Team Leader who will save the record into the project file.
	<b>Metrics</b>
6.1	Total preparation time for the review is collected and tracked.
6.2	Total number of issues identified in the review is collected and tracked.
6.3	Total number of issues identified for each anomaly class in the review is collected and tracked.

## APPENDIX – J

# INTENTIONAL MISTAKES IN THE REVIEW DOCUMENTS

**Table 15 : Intentional Mistakes in the Review Documents**

Found	Document	Position of the Mistake	Explanation of Mistake
	ATM System.mdl	Use Case View, Main Diagram	“Validate User” use case’i olduğu halde, diagramda görülmemektedir.
Yes	ATM System.mdl	Use Case View, Main Diagram	“Query the Balance” ile “Check Account Balance” arasındaki ok işaretinin üstünde “uses” olmalı.
	ATM System.mdl	Use Case View, Main Diagram	“Transfer Funds”, “Transfer Funds” olarak değiştirilecek.
	ATM System.mdl	Use Case View, Main Diagram	“Deposit Funds” ile “Bank System” arasında association olmalı.
Yes	ATM System.mdl	Use Case View, Main Diagram	“New Use Case” adlı alakasız bir use case var.
	ATM System.mdl	Logical View, Main Diagram	“Customer” dan “Account”a giden association üzerinde “own” olmalı.
	ATM System.mdl	Logical View, Main Diagram	“ATM Machine” ile “Bank Card” arasında 1-1 değil, “1..n”-“1..n” multiplicity ilişkisi olmalı.
Yes	ATM System.mdl	Logical View, Main Diagram	“New Class” adlı alakasız bir class var
Yes	ATM System.mdl	Logical View, Business Package, Main Diagram	“Screen” classı diagramda çizilmemiş.
Yes	ATM System.mdl	Logical View, GUI Package, Main Diagram	Diagram yok.

**Table 15 : Intentional Mistakes in the Review Documents (cont.)**

Found	Document	Position of the Mistake	Explanation of Mistake
	Detailed Use Case Definitions.doc	General	Yazı rengi mavi değil siyah olmalı
Yes	Detailed Use Case Definitions.doc	Page 1	“Table of Cotents” yazılmış “Table of Contents” olacak.
Yes	Detailed Use Case Definitions.doc	Page 1	“3. Usecases” olmuş, “2. Usecases” olacak.
Yes	Detailed Use Case Definitions.doc	Page 1	“Check Account Balance”ın sayfa numarası 17 değil, 7 olacak.
Yes	Detailed Use Case Definitions.doc	Page 2	Costomer, Customer olacak.
Yes	Detailed Use Case Definitions.doc	General	Notes ve Sources, Alternate Flow bölümleri genelde boş bırakılmış.
Yes	Detailed Use Case Definitions.doc	Page 3	“Validate User” da “Postcondition” yok
Yes	Detailed Use Case Definitions.doc	Page 6	“Transfer Funds” da “Definition” yok
	Detailed Use Case Definitions.doc	Page 7	“Check Account Balance” adlı usecase hakkında hiçbir bilgi yok. (TOC’e göre olması gerekirdi)
Yes	Detailed Use Case Definitions.doc	Page 8	Yazı içinde alakasız bir şekilde “HATA BURADA GÖRDÜNMÜ” şeklinde bir yazı var.
Yes	PD_Karsilastirma_Tablosu.xls	Karşılaştırma Tablosu, E-2	Asic yazılmış, Basic olacak
	PD_Karsilastirma_Tablosu.xls	Karşılaştırma Tablosu, G-6	Bu cell neden boş?
Yes	PD_Karsilastirma_Tablosu.xls	Karşılaştırma Tablosu, G-21	“Ne bileyim” yazıyor. Evet/Hayır olmalı

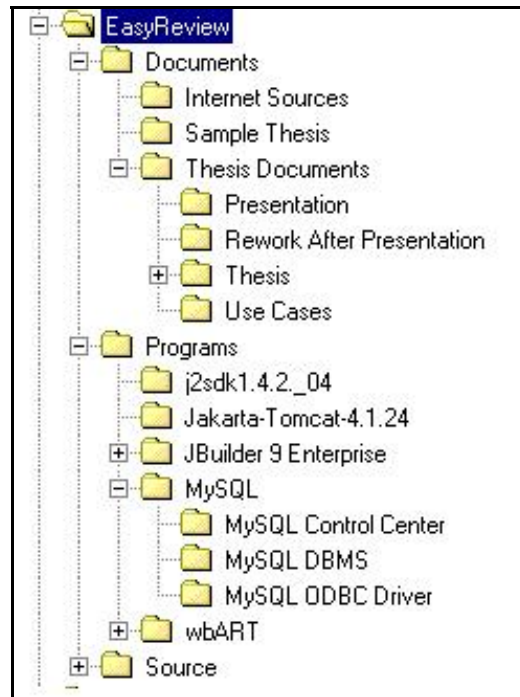


**Table 15 : Intentional Mistakes in the Review Documents (cont.)**

Found	Document	Position of the Mistake	Explanation of Mistake
	PD_Karsilastirma_Tablosu.xls	Karşılaştırma Tablosu	Geçen kısaltmalar dipnot şeklinde yazılsa iyi olur.
Yes	PD_Karsilastirma_Tablosu.xls	Karşılaştırma Tablosu, E-12	Bu alan neden sarı renkli?
Yes	PD_Karsilastirma_Tablosu.xls	Tanımlar	Bazı kavramların tanımları verilmemiş. (Assertion gibi)
	PD_Karsilastirma_Tablosu.xls	Tanımlar	Tanımlamalar alfabetik sıralamaya dizilse daha iyi olur.
Yes	PD_Karsilastirma_Tablosu.xls	Tanımlar, A-10	Tanım adı var, detayı yok
Yes	PD_Karsilastirma_Tablosu.xls	Tanımlar, A-11	Tanım adı var, detayı yok
Yes	PD_Karsilastirma_Tablosu.xls	Tanımlar A-3	“Meta Kalas” yazılmış, “Meta Class” olacak
Yes	PD_Sunum.ppt	Slide 1	“eçimi” yerine “Seçimi” olacak
Yes	PD_Sunum.ppt	Slide 6	1’den sonra madde numarası 2 yerine 3 yazılmış.
Yes	PD_Sunum.ppt	Slide 2	Temel Kavramların sadece adı var, tanımları konulmalıdır.
	PD_Sunum.ppt	Slide 7	6. slide’da verilen maddelerden daha farklı bir başlık verilmiş.
	PD_Sunum.ppt	Slide 8	Hangi dilin kaç slide’dan oluştuğunu anlayabilmek için 1/6 gibi başlığın yanına slide sayısı yazılabilir
	PD_Sunum.ppt	Slide 13	C++ da pointer aritmetiği vardır.
	PD_Sunum.ppt	Slide 20	“Strongly typed” (Static typing) olmalı, “Dynamic” değil.
Yes	PD_Sunum.ppt	Slide 27	C# anlatılırken başlığa C++ yazılmış
Yes	PD_Sunum.ppt	Slide 29	“InternationalConsortium” bitişik yazılmış
Yes	PD_Sunum.ppt	Slide 31	Başlık var devamı yok.

## CONFIGURATION OF EASYREVIEW

### 1. Contents of EasyReview CD-ROM



### 2. Configuring EasyReview Server

- a) Insert EasyReview CD-ROM into the CD drive. Copy all files and folders from “Source” to “C:\EasyReview”.
- b) Install Jakarta-Tomcat-4.1.24 to C drive.
- c) Install Java 2 Platform Standard Edition (j2sdk1.4.2\_04) to C drive.
- d) Install MySQL DBMS.
  - a. After installing with default settings; run  
“C:\mysql\bin\winmysqladmin.exe”.
- e) Install MySQL Control Center.

- f) Create EasyReview database.
  - a. Open MySQL Control Center.
  - b. Create MySQL server with the following options
    - i. Name: root@localhost:3306
    - ii. Hostname: localhost
    - iii. User: root
    - iv. Password: (Default blank, but you can change)
    - v. Port: 3306
  - c. Right click on “Databases” folder of “MySQL Servers” screen and choose “New Database”. Enter “EasyReviewDB” for database name.
  - d. Press “EasyReviewDB” and click “SQL” on the upper menu.
  - e. Open “Source\database\EasyReviewDB.sql” from CD and execute the code (Now EasyReview database is ready with default data).
- g) NOTE: If EasyReview server and clients are connected to different local area networks, port number 3306 must be opened in the router configuration.

### **3. Starting and Stopping EasyReview Server**

- a) Use “C:\EasyReview\Start EasyReview.bat” to start the server.
- b) Use “C:\EasyReview\Stop EasyReview.bat” to stop the server.

### **4. Configuring Clients**

- a) Install MySQL ODBC Driver with default settings.
- b) Enable ActiveX Controls.

- a. Open Internet Explorer (tested on 5.x and 6.x versions).
- b. Open Tools → Internet Options → Security → Trusted Sites.
- c. Add EasyReview server as a trusted site.
- d. Open “Custom Level” security and set “Enable” for all settings.

## 5. Using EasyReview

- a) Start EasyReview server.
- b) Connect to server (from server or client) by entering  
“[http://server\\_name/web/root/login.html](http://server_name/web/root/login.html)” in the Web browser’s address field  
(“server\_name” is the name or IP address of the EasyReview server).
- c) Enter “**admin**” for “Username” and “**a**” for “Password” (Default settings, you can then change password, add new users, etc.).
- d) Maintaining Server Settings
  - a. Change “Web Server”, “Mail Server” and “Connection String” settings according to your configuration.
  - b. Default “Connection String” is “DRIVER={MySQL ODBC 3.51 Driver};SERVER=pchvkk6;Port=3306;Database=EasyReviewDB;UID=root;PWD=admin;OPTION=3;”. You HAVE TO change “SERVER” parameter, which is the name or IP address of the EasyReview server. You MAY change other settings according to your configuration (You can get help from <http://www.connectionstrings.com> internet address).
- e) To be able to perform other admin and review participant operations, you can refer to Chapter 4 (especially section 4.4).