THE IMPLEMENTATION OF REFACTORING: AN EXPERIENCE REPORT

AMIR SHAZLAN BIN MOHD SHAH

UNIVERSITI TEKNOLOGI MALAYSIA

PSZ 19:6 (Pind. 1/97)

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN STATUS TESIS

JUDUL : THE IMPLEMENTATION OF REFACTORING – AN EXPERIENCE REPORT

SESI PENGAJIAN: 2004 / 2005

Saya AMIR SHAZLAN BIN MOHD SHAH
(HURUF BESAR)

mengaku membenarkan tesis (~~PSM~~/Sarjana~~/Doktor Falsafah~~)* ini di simpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut:

Tesis adalah hak milik Universiti Teknologi Malaysia.
Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan pengajian sahaja.
Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
**Sila tandakan (√)

□ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☑ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikkan dijalankan)

□ TIDAK TERHAD

Disahkan oleh

(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)

Alamat Tetap:
C001, KUARTERS GURU,
SEKOLAH SERI PUTERI,
63000 CYBERJAYA

EN. SAIFUL ADLI
(Nama Penyelia)

Tarikh: 27 SEPTEMBER 2004

Tarikh: 27 SEPTEMBER 2004

27th September 2004

Librarian
Perpustakaan Sultanah Zanariah
UTM, Skudai
Johor

Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

- THE IMPLEMENTATION OF REFACTORING: AN EXPERIENCE REPORT

-AMIR SHAZLAN

Please be informed that the above-mentioned thesis entitled **“THE IMPLEMENTATION OF REFACTORING: AN EXPERIENCE REPORT”** be classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are:

(i) This technical report consists proprietary information of the organization where the student was trained.

(ii) This technical report is related to a project, which involves Non-Disclosure Agreement (NDA) between the organization and its customer.

Thank you.
Sincerely yours,

Chong Sau Wei,
SCAN Associates Sdn. Bhd.,
6, Lrg Gurney Off Jalan Semarak,
54100 Kuala Lumpur,
+6019-3437539.

“~~I~~/We admit that ~~I~~/we have read this technical report and
in ~~my~~/our point of view this report is sufficient in scope and
quality for the award of the degree of Master of Science
(Computer Science - Real-Time Software Engineering) ”

Signature :

Academic Mentor : EN SAIFUL ADLI

Date 27TH SEPTEMBER 2004

Signature :

Industrial Mentor : EN. CHONG SAU WEI

Date 27TH SEPTEMBER 2004

THE IMPLEMENTATION OF REFACTORING:
AN EXPERIENCE REPORT

AMIR SHAZLAN BIN MOHD SHAH

This technical report is submitted to fulfill the
partial requirement for the award of the degree of
Masters of Science (Computer Science –
Real-Time Software Engineering)

Centre for Advanced Software Engineering
Universiti Teknologi Malaysia

SEPTEMBER 2004

I hereby declare that the technical report titled “The Implementation of Refactoring: An Experience Report” is the result of my own research except as cited in the references which henceforth I have defined the sources. This technical report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ------------------------------------

Name : AMIR SHAZLAN BIN MOHD SHAH

Date : 27th SEPTEMBER 2004

To all those important in my life.

Thank you for the unconditional love, courage and support.

ACKNOWLEDGEMENT

First of all, I would like to thank SCAN Associates for giving me the opportunity to participate in one of the company's project. Through my observation of their work culture I have gained a lot of valuable input and experience that will certainly help me guide my way in the future.

Utmost gratitude goes to my Industrial Mentor, En Chong Sau Wei and my Academic Mentor, En Saiful Adli for their supervision. I sincerely thank them for their willingness to give me guidance during the five months involvement in the project and the preparation of the thesis. Personal thank you to En Nazri Mahrin for his most valuable counsel in helping me get through the internship period.

I wish to personally thank the whole IDS Team for making me feel at ease in the workplace. Working there has enabled me to learn how to work together as a team, a chance I rarely get. This experience will come in handy when I enter the workforce soon, thanks to you guys.

Last but not least, I extend my thanks to my fellow classmates who constantly lend me their shoulders to make this thesis possible. The hardships and the laughter we shared together will always remain forever etched in memory.

ABSTRACT

Many object-oriented software developers have embraced the technique of refactoring – restructuring the source-code of an object-oriented program without changing its external behavior – as an approach to modifying and maintaining existing software systems. This is because systems, and consequently their design, are prone to software degradation, which if left unattended will affect the maintainability of the system. Refactoring allows the developer to continually uphold the design of the system without spending as much time as other techniques such as reverse engineering. In this study the writer would discuss his experiences in applying refactoring on a system and explain how they are implemented so as to preserve the behavior of a system. A set of software metrics was utilized to measure the impact of the said refactoring activity to the system. The results show marked improvements on the maintainability, usability and complexity of the overall system. Thus, it proves that refactoring can slow down the software degradation rate of a system.

ABSTRAK

Teknik ‘refactoring’ ialah satu teknik kod sumber dibentuk semula tanpa mengubah sifat zahirnya. Ia kini telah menjadi satu pendekatan yang semakin mendapat sambutan dari pembangun perisian berasaskan teknologi objek dalam proses memelihara dan menyelenggara sistem sedia ada. Ini kerana sesuatu sistem perisian dan rekabentuknya akan mengalami kecenderungan untuk menjadi mundur sekiranya tiada langkah diambil untuk mencegah perkara tersebut. ‘Refactoring’ membolehkan pembangun perisian untuk memelihara rekabentuk. Tesis ini akan membincangkan pengalaman penulis mengaplikasikan teknik ini pada sebuah sistem dan menerangkan bagaimana ia harus dilaksanakan agak sifat zahir sistem terpelihara. Satu set metriks perisian telah dipilih untuk mengkaji kesan teknik ‘refactoring’ kepada sistem secara keseluruhannya. Keputusan akhir set metriks tersebut menunjukkan peningkatan kepada sifat–sifat kompleksiti, pemuliharaan dan penggunaan semula sistem. Ini membuktikan teknk ‘refactoring’ ini mampu memperlahankan kadar kemunduran sistem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AOP	-	Aspect Oriented Programming
CLOC	-	Comment Lines of Code
DIT	-	Depth In Tree
DoS	-	Denial of Service
FAMOOS	-	Framework-based Approach for Mastering Object-Oriented Software Evolution
GUI	-	Graphical User Interface
ICT	-	Information and Communications Technology
IDS	-	Intrusion Detection System
IPRS	-	Internet Pages Recovery System
IRPA	-	Intensified Research Project Area
LOC	-	Lines of Code
OID	-	Object Identifier
RFC	-	Response For Class
RUP	-	Rational Unified Process
SAD	-	Software Architecture Design
SCAN	-	SCAN Associates Sdn Bhd
SNMP		Simple Network Management Protocol
UML	-	Unified Modeling Language
WMC	-	Weighted Methods per Class
XP	-	Extreme Programming

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

This chapter describes the organization, which is SCAN Associates in brief. The introduction also describes the IDS project and its corresponding objectives. The final section is the description of the project detailing the objective, scope and project deliverables.

1.1 Company Background

This section describes the company background as an introduction to the working environment. The history and subsequent relevant company information are detailed. The IDS Project team information and organizational structure are also described in this section.

1.1.1 SCAN Associates

The idea of SCAN Associates started in 1996 when Prof. Dr. Norbik Bashah Idris from Universiti Teknologi Malaysia (UTM) established what was then called the "Security Control and Analysis System (SCAS)", an IRPA Research group funded by the Ministry of Science, Technology & the Environment under the 7th Malaysia Plan. For the research, he assembled a group of local researchers and talents to help realize the research. Many of these talents have qualifications ranging from Honors and Masters Degrees to PhDs in the area of information security. When the end of the research period came, he managed to go one step further and went on to form a partnership with two of his peers to capitalize on the human talent pooled and to further establish SCAS as a commercial entity. With this same core team still in existence until today, they form one of the best IT security company in the country.

Ever since then SCAN Associates has since been involved in such research efforts as implementing strong cryptographic algorithms and security applications, development of digital notary systems, more and recently, the public key infrastructure. Throughout the years SCAN Associates has had collaborated extensively with other security research groups around the world to exchange ideas and opinions. From these activities and its own in-house research SCAN Associates has acquired the knowledge and developed the capability in the area of information security in subjects such as cryptography, authentication, authorization, access control and digital signature.

Unlocking their highly developed potential and capability to address the critical needs of organizations needing to secure their information assets, SCAN Associates sets out to provide consultation services on information and network security for the government, financial and commercial sector. Currently SCAN Associates is actively supporting various government agencies and organizations in the field of information security. Many organizations in the public and private sector have approached SCAN Associates to undertake various information security related activities. Among them are services such as resource persons on information and communication technology (ICT) security awareness, training & education, security technology advisor, intrusion incident response and handling, penetration testing, security auditing, designing secure network and security policy review to government agencies and institutions. SCAN Associates also participates as committee members in various advisory and consultative capacities.

Through these various experiences, SCAN Associates has matured and is currently expanding its consultancy capabilities and capacities by developing its own unique methodologies in the area of security assessment, penetration testing, security architecture and policy review. To highlight a few examples of innovations created, SCAN Associates has developed its own cryptographic engine, which is a set of programming libraries in Java and C, implementing well known strong encryption algorithm such as TripleDES, TwoFish, RC5 and RSA. Based on the cryptographic engine, a result of 4 years of extensive research, SCAN is also developing an indigenous Public Key Infrastructure (PKI) and its supporting tools.

1.1.2 SCAN Associates Vision

In the long run SCAN Associates aims to become the country's first truly indigenous group of expertise in information security implementing a world class standard of services. SCAN Associates also ultimately aims to be the trusted ICT Security Solutions Provider in Malaysia.

1.1.3 Awards Garnered

Products from SCAN Associates have won an IBM Java Home-brew Award in April 2000 and a Gold Invention Award in June 2000, in the category of "Idea, Concept and Algorithm" at the INATEX 2000 exhibition. Besides technical awards, consultants from SCAN Associates are actively involved in research capabilities, publishing papers and invited as speaker to conventions. Most recently, a senior security consultant was invited to present a technical paper at the highly regarded RUXCON hacking conference held in Sydney, Australia on July 2004.

1.1.4 IDS Project Team

The IDS project is to provide the customer with the capability to both proactively and reactively protect their vital information assets. It is a security solution designed to minimize threats by centrally managing intrusion detection and periodically performing vulnerability assessment. The solution will offer 24x7 active security monitoring and support, coupled together with analysis, reporting and response for security incidents. With the realization of the IDS, it will help the customer organization in establishing baseline security posture, elevating it to a better state and upholding it to thwart and minimize security threats.

1.1.4.1 IDS Objectives

Over the longer term IDS will help in securing vital personal and organization electronic data and thwart any security threats coming from inside or outside of the organization. The IDS project was initiated to achieve the following objectives:

- Provides the customer with the capability to proactively detect and identify information system weaknesses.
- Build a security forecasting center that is able to forecast cyber attacks likely to take place, therefore providing appropriate and early warnings to the organization allowing its workers and security personnel to better prepare and manage the security of their organization.
- Provides the customer with a 24x7 ICT security service that will reactively monitor and respond to ICT incidents throughout the organizational perimeter.
- Provides the customer with an advanced website monitoring and recovery system.
- To become a one-stop ICT Security reference center that provides first level assistance, advisories and alerts related to ICT security operations and management within the organization.

1.1.4.2 IDS Project Organizational Structure

The importance of the IDS security solutions is reflected in the creation of a dedicated team solely focused on building this security solution with a different site office. This enables the team to work together in a common environment. The organizational structure of the IDS Project team is formalized in figure 1.1.

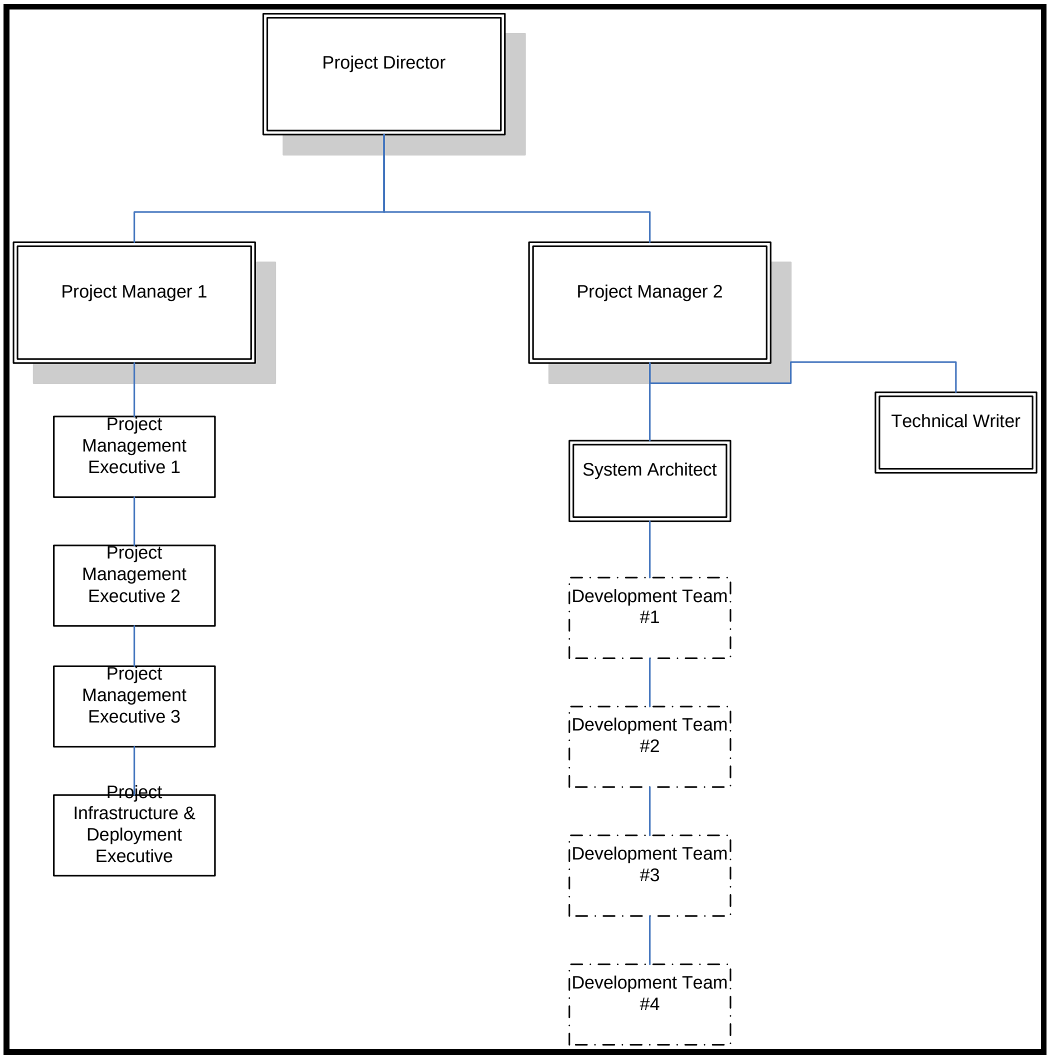


Figure 1.1 IDS Project Organizational Structure

1.2 Project Brief

This sections spells out the project in terms of its objective, scope, planning and deliverables.

1.2.1 Project Overview

The author's arrival in the company for the second part of industrial attachment has coincided with the IDS project management's desire to better strengthen its software engineering activities. Part of its focus regarding to the said activities is in streamlining the design of its components and making sure its internal documentation are accurate and in tandem with the source code of its components. Since the IDS solution is currently in its Final Acceptance Test phase, this task is urgently needed for the eventual change that is going to happen after the testing is done. The task makes sure the design documents produced follows the source code produced and modifying the design done earlier in accordance with the source code should there be any difference. Along the way the source code would be checked for compliance with the coding guidelines subscribed. Besides doing that, another major task is that the author would also be applying refactoring techniques to a selected component of the IDS in preparation for its reengineering. The refactoring technique will be further described in coming chapters.

1.2.2 Objective

The project objectives are as follows:

- To implement the refactoring process as applied to the source code
- To revise the system architecture document as to reflect the current source code and also current software model.

- To study the refactoring process and its impact to the component in question.
- To impart the knowledge of refactoring among the development teams so that they will be inspired to use it in their daily work.

1.2.3 Scope

The project scope is as follows:

- The refactoring for both source code and its related internal documentation is to be done for one component in the IDS project. The component is the Internet Page Recovery System (IPRS).
- Some of the documentations are confidential in nature. For this prior approval will be sought from the Industrial Mentor to document any design-related subjects in the report.
- The internal documentation to be updated is the Software Architecture Design (SAD) document for the specified components only.
- The coding guideline to be used as reference is the IDS projects own Java Programming Guidelines.
- The refactoring to be applied is the refactoring techniques that are only applicable within classes only.

1.2.4 Project Planning

The project schedule is planned in the project Gantt chart that is documented in Appendix A. As a summary, the project is to be implemented over a span of 20 weeks beginning 23rd March 2004 until 20th August 2004 in 6 phases, as stated in the Gantt chart.

1.2.5 Project Deliverables

At the end of the industrial training period, the expected deliverables are as follows:

- An up to date Software Architecture Document for the IPRS.
- A refactored IPRS source code that complies with the IDS Java Programming Guidelines.
- A training session to the developers for implementing refactoring.
- A technical presentation to CASE, including a technical report. The technical presentation includes the training brochure, documented in Appendix B.

1.2.6 Technical Report Outlines

The technical report has six chapters in all and is organized as follows: Section 2 is the literature study of the issues involved. Section 3 presents the methodology used in the duration of the project. Section 4 tabulates the data collected in the project and discusses the results. Lastly, Section 5 draws conclusions and recommendations for future works. Some extra-related information is reported in the appendices.

REFERENCES

Axelssion, S. (2000). *Intrusion Detection System : A Survey and Taxonomy*. Technical Report No99-15. Department of Computer Engineering, Chalmers University.

Axelsson, S (1999). *Research in Intrusion Detection systems: A Survey*. Technical Report No98-17. Department of Computer Engineering, Chalmers University.

Beck, K. (2000). *Extreme Programming Explained: Embrace Change.* US: Addison Wesley Professional.

Benlarbi, S., El-Emam, K., Goel, N., and Rai, S. (2000). *Thresholds for Object-Oriented Measures.* NRC/ERB 1073. (National Research Council of Canada).

Bennett, K. and Rajlich, V. (2000) A Staged Model for the Software Lifecycle, *IEEE Computer*. 33(7): 66-71

Bradly, D. (1999). Practical Analysis for Refactoring. University of Illinois: PhD thesis.

Burke, B. (2003). Aspects: A New Paradigm. *Java Developers Journal*. 8(12):10-11.

Demarco, T. (1986) *Controlling Software Projects: Management, Measurement and Estimates.* US: Prentice-Hall.

Demeyer S., Ducasse S., and Nierstrasz O. (2003). *Object-Oriented Reengineering Patterns*. US: Morgan Kaufmann Publishers.

Demeyer S., Ducasse S. and Nierstrasz O. (2000). Finding Refactorings via Change Metrics. *Proc. OOPSLA 2000 (Object-Oriented Programming, Systems, Languages, and Applications)*. October 15-19. Minneapolis, Minessota.

Eeles, P., Houston, K. and Kozaczynski, W. (2002*). Building J2EE Applications with the Rational Unified Process.* US: Addison-Wesley Professional.

Fenton, E., F. and Pfleeger, S., L., (1997) *Software Metrics: A Rigorous and Practical Approach.* UK: PWS Publishing Co.

Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code.* US: Addison-Wesley Professional.

Gorp, P.V., Stenten, H., Mens, T., and Demeyer, S., (2003). Towards automating source-consistent UML Refactorings. *Proc. of UML 2003 – The Unified Modeling Language*. October 20-24 2003. CA, USA.

Hightower R., Onstine W., Visan P., Payne, D. and Gradecki, J.D. (2004). *Professional Java Tools for Extreme Programming: Ant, XDoclet, JUnit, Cactus, and Maven (Programmer to Programmer)*. US:Wrox.

Insitute of Electrical and Electronic Engineers (1998). *Standard For Software Maintenance*. US. Std 1219-1998.

International Organization for Standards (1995). *Software Life Cycle Processes*. Geneva. ISO 12207.

Jacobson, I., Booch, G. and Rambaugh, J. (1999). *The Unified Software Development Process*. US : Addison-Wesley Professional.

Jonathan Rasmusson (2003) Introducing XP into Greenfield Projects: Lessons Learned. *IEEE Software*. 20(3):21-28.

Krutchen, P (2003). *The Rational Unified Process: An Introduction (2nd Edition)*. US: Addison-Wesley Pub. Co.

Marko, B., Thorsten, S. and Fragemann, P. (2002) Refactoring Browser for UML. *Proc. 3rd Int'l Conf. on eXtreme Programming and Flexible Processes in Software Engineering.* 2002. US. 77-81

McHugh, J., Christie, A. and Allen, J., (2000). Defending Yourself: The Role of Intrusion Detection Systems. *IEEE Software*. 17(5): 43-51

Mens, T. and Tourwe,T (2004). A Survey of Software Refactoring. *IEEE Transactions on Software Engineering*. 30(2) :126-139.

Mens, T. and Deursen, A. W. (2003). Refactoring: Emerging Trends and Open Problems. Proc. *First International Workshop on REFactoring: Achievements, Challenges, Facts (REFACE)*. Nov 13 2003, University of Waterloo.

M. Lehman (1980). Program, Life-Cycle, and the Law of Program Evolution. *Proc. of the IEEE*. 1060-1076.

Opdyke, W.F. (1995). Refactoring Object-Oriented Software to Support Evolution and Reuse. *WISR 7: Seventh Annual Workshop on Software Reuse.* April 1995. St. Charles, Illinois.

Opdyke, W. F. (1992). *Refactoring Object-Oriented Frameworks*. University of Illinois at Urbana-Champaign:PhD thesis.

Pressman, R. S. (2001). Software *Engineering: A Practitioner's Approach*. 5th Edition. US: McGraw-Hill.

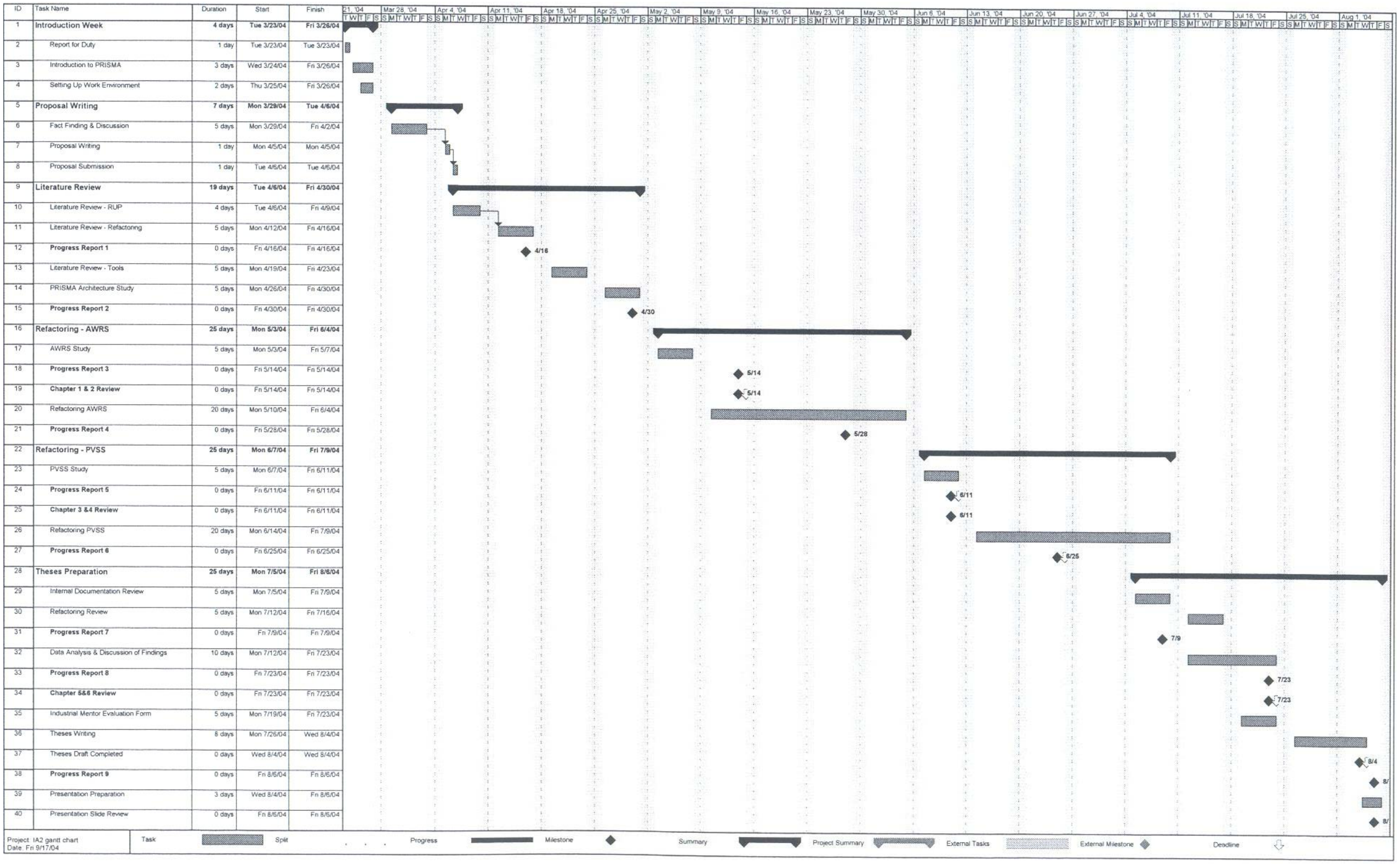
Rosenberg, L (1998). Applying and Interpreting Object-Oriented Metrics. *Software Technology Conference*. April 13-15 1998. Larry Hyatt, Utah.

Sellappan, P. (1999) *Software Engineering: Management and Methods*. KL: Saujana Publishing.

Thompson S. and Reinke C., (2003) A Case Study in Refactoring Functional Programs. *VII Brazilian Symposium on Programming Languages*. May 2003. Brazil: Sociedade Brasileira de Computacao, 1-16.

Wake, W. C. (2004). *Refactoring Workbook*. US: Addison-Wesley Professional.

Wang, W. (2003). *Steal This Computer Book 3*. US: No Starch Press Inc.



ID	Task Name	Duration	Start	Finish
1	**Introduction Week**	**4 days**	**Tue 3/23/04**	**Fri 3/26/04**
2	Report for Duty	1 day	Tue 3/23/04	Tue 3/23/04
3	Introduction to PRISMA	3 days	Wed 3/24/04	Fri 3/26/04
4	Setting Up Work Environment	2 days	Thu 3/25/04	Fri 3/26/04
5	**Proposal Writing**	**7 days**	**Mon 3/29/04**	**Tue 4/6/04**
6	Fact Finding & Discussion	5 days	Mon 3/29/04	Fri 4/2/04
7	Proposal Writing	1 day	Mon 4/5/04	Mon 4/5/04
8	Proposal Submission	1 day	Tue 4/6/04	Tue 4/6/04
9	**Literature Review**	**19 days**	**Tue 4/6/04**	**Fri 4/30/04**
10	Literature Review - RUP	4 days	Tue 4/6/04	Fri 4/9/04
11	Literature Review - Refactoring	5 days	Mon 4/12/04	Fri 4/16/04
12	**Progress Report 1**	0 days	Fri 4/16/04	Fri 4/16/04
13	Literature Review - Tools	5 days	Mon 4/19/04	Fri 4/23/04
14	PRISMA Architecture Study	5 days	Mon 4/26/04	Fri 4/30/04
15	**Progress Report 2**	0 days	Fri 4/30/04	Fri 4/30/04
16	**Refactoring - AWRS**	**25 days**	**Mon 5/3/04**	**Fri 6/4/04**
17	AWRS Study	5 days	Mon 5/3/04	Fri 5/7/04
18	**Progress Report 3**	0 days	Fri 5/14/04	Fri 5/14/04
19	**Chapter 1 & 2 Review**	0 days	Fri 5/14/04	Fri 5/14/04
20	Refactoring AWRS	20 days	Mon 5/10/04	Fri 6/4/04
21	**Progress Report 4**	0 days	Fri 5/28/04	Fri 5/28/04
22	**Refactoring - PVSS**	**25 days**	**Mon 6/7/04**	**Fri 7/9/04**
23	PVSS Study	5 days	Mon 6/7/04	Fri 6/11/04
24	**Progress Report 5**	0 days	Fri 6/11/04	Fri 6/11/04
25	**Chapter 3 &4 Review**	0 days	Fri 6/11/04	Fri 6/11/04
26	Refactoring PVSS	20 days	Mon 6/14/04	Fri 7/9/04
27	**Progress Report 6**	0 days	Fri 6/25/04	Fri 6/25/04
28	**Theses Preparation**	**25 days**	**Mon 7/5/04**	**Fri 8/6/04**
29	Internal Documentation Review	5 days	Mon 7/5/04	Fri 7/9/04
30	Refactoring Review	5 days	Mon 7/12/04	Fri 7/16/04
31	**Progress Report 7**	0 days	Fri 7/9/04	Fri 7/9/04
32	Data Analysis & Discussion of Findings	10 days	Mon 7/12/04	Fri 7/23/04
33	**Progress Report 8**	0 days	Fri 7/23/04	Fri 7/23/04
34	**Chapter 5&6 Review**	0 days	Fri 7/23/04	Fri 7/23/04
35	Industrial Mentor Evaluation Form	5 days	Mon 7/19/04	Fri 7/23/04
36	Theses Writing	8 days	Mon 7/26/04	Wed 8/4/04
37	Theses Draft Completed	0 days	Wed 8/4/04	Wed 8/4/04
38	**Progress Report 9**	0 days	Fri 8/6/04	Fri 8/6/04
39	Presentation Preparation	3 days	Wed 8/4/04	Fri 8/6/04
40	Presentation Slide Review	0 days	Fri 8/6/04	Fri 8/6/04

Project: IA2 gantt chart
Date: Fri 9/17/04