BORANG PENGESAHAN STATUS TESIS^

JUDUL: PERSONAL SOFTWARE PROCESS: ERROR RECORDER

SESI PENGAJIAN: SEMESTER 1 2006 / 2007

Saya FAKHRUL HAKIM BIN FAUZI

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Kolej Universiti Teknikal Kebangsaan Malaysia.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** 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 penyelidikan dijalankan)
___✓___ TIDAK TERHAD	

(FAKHRUL HAKIM FAUZI)

(PN. NOR HAFEIZAH HASSAN)

Alamat tetap : 419-B, JLN ALUR TEMBESU

23000 DUNGUN, TERENGGANU

Tarikh : 23/11/06

Tarikh : 23/11/2006

CATATAN: ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa. ^ Tesis dimaksudkan sebagai Laporan Projek Sarjana Muda (PSM)

PERSONAL SOFTWARE PROCESS: ERROR RECORDER

FAKHRUL HAKIM BIN FAUZI

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Software Development)

FACULTY OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

KOLEJ UNIVERSITI TEKNIKAL KEBANGSAAN MALAYSIA

2006

DECLARATION

I hereby declare that this project report entitled

Personal Software Process: Error Recorder

is written by me and is my effort and that no part has been plagiarized without citations

STUDENT : ______________________ Date : 23/11/06

(FAKHRUL HAKIM B. FAUZI)

SUPERVISOR : ______________________ Date : 23/11/2006

(NOR HAFEIZAH BT HASSAN)

DEDICATION

To the Al-Mighty Allah,

To my beloved parent, Fauzi Hamzah & Norhayati Adam

To my supervisor, Pn. Nor Hafeizah Hassan

ACKNOWLEDGEMENTS

First of all, praise to Allah s.w.t. the almighty for giving me chance and strengths to finish my final project. I'm also taking this opportunity to give my gratitude to a couple of parties for their help. I would like to thank my family especially my parent, Haji Fauzi bin Hamzah and Hajah Norhayati binti Adam for the support, love and encouragement that they provided to me through my entire life. I also would like to express my gratitude to my supervisor, Pn. Nor Hafeizah binti Hassan, whose give me ideas, guidance with understanding and patience to help me during completing this thesis and project. A very special thanks also goes to En.Sanusi bin Azmi, for the help and guidance, Chang Gee Guan, Mohd Kadri Mustafa and Ashabul Yamin Mustakim, for helping me during my tough time. Also thank you to all my friends in and outside KUTKM for their exchanging of knowledge, suggestions, ideas and morale support while completing this final project which give me a very memorable moments and experiences. Perhaps this little project could contribute something for this lovely KUTKM, lecturers and for the next generation of students. Wassalam.

ABSTRACT

One of important element in software and computer system development is programming. To be a good programmer, lot of things should be learnt. Also lot of experiences should be gained. An early step to learn about programming especially during campus time is very important for the students before they walk into working world. It can not be denying that during this learning process, student will do lot of mistakes and sometimes the way student think is not same with the way lecturers are thinking. This is a normal thing during the learning process. Unfortunately, not many students who want to give their ideas or meet the lecturers to confess their problems and way to solve errors which they did during programming session. In this situation, lecturers will face difficulties to trace student weakness and which aspect should be more concentrate on. So, this project will describe how this to-be software can help both side to detect student weakness, and try to overcome the weaknesses along the way to produce students who are expert in programming.

ABSTRAK

Pengaturcaraan adalah salah satu elemen penting dalam pembangunan sesebuah perisian dan sistem komputer. Untuk menjadi seorang pengaturcara yg baik dan berkebolehan, banyak perkara yang perlu dipelajari. Juga banyak pengalaman yang perlu ditimba. Langkah awal mendalami pengaturcaraan terutama ketika di kampus adalah penting untuk seseorang penuntut itu mengasah skill dan kebolehan mereka sebelum melangkah ke dunia pekerjaan. Namun, tidak dapat dinafikan semasa proses pembelajaran tersebut, akan berlaku kesilapan oleh penuntut dan percanggahan pendapat antara penuntut dan pensyarah. Ini adalah perkara biasa dalam proses pembelajaran. Walabagaimanapun, tidak ramai di kalangan penuntut yang mahu menyuarakan pendapat atau yang berkeinginan bertemu pensyarah menyatakan masalah dan cara untuk mengatasi kesilapan yang dilakukan semasa kelas pengaturcaraan. Dalam hal ini, pensyarah juga akan mengalami kesukaran untuk mengenal pasti kelemahan penuntut dan aspek mana yang perlu ditekankan lagi. Oleh itu, projek ini akan menerangkan bagaimana perisian yang bakal dibangunkan ini dapat membantu kedua-dua belah pihak untuk mengenalpasti kelemahan penuntut, seterusnya cuba mengatasi kelemahan tersebut dalam usaha untuk melahirkan penuntut yang mahir dalam pengaturcaraan.

TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DECLARATION	**i**
	DEDICATION	**ii**
	ACKNOWLEDGEMENTS	**iii**
	ABSTRACT	**iv**
	TABLE OF CONTENT	**v**
	LIST OF TABLES	**x**
	LIST OF FIGURES	**xi**
CHAPTER I	**INTRODUCTION**	**1**
	1.1 Project Background	1
	1.2 Problem Statement	2
	1.3 Objectives	3
	1.4 Scope	4
	1.5 Project Significance	5
	1.6 Conclusion	6

CHAPTER II LITERATURE REVIEW AND PROJECT METHODOLOGY 8

2.1 Introduction 8

2.2 Fact on Finding 9

2.2.1 Case Study on Using PSP in Programming Class 9

2.2.2 Open-Source Initiative to create PSP 13

2.3 Project Methodology 14

2.4 Project Requirement 16

2.4.1 Software Requirement 16

2.4.2 Hardware Requirement 16

2.5 Project Schedule/Milestone 17

2.5.1 Milestones 17

2.5.2 Gantt Chart 19

2.6 Conclusion 21

CHAPTER III ANALYSIS 22

3.1 Introduction 22

3.2 Problem Analysis 22

3.2.1 Background of Previous System 22

3.2.2 Problem Statement 24

3.3 Requirement Analysis 25

3.3.1 Functional Requirement 25

3.3.2 Software Requirement 31

3.3.3 Hardware Requirement 33

3.4 Conclusion 33

CHAPTER IV DESIGN 35

4.1 Introduction 35

4.2 High-Level Design 35

	4.2.1 Raw Data	36
	4.2.2 High-Level Logical View / Architecture	36
	4.2.3 User Interface Design	40
	4.2.4 Deployment View	43
	4.3 Detailed Design	43
	4.3.1 Software Specification	43
	4.3.2 Physical Database Design	46
	4.4 Conclusion	48
CHAPTER V	**IMPLEMENTATION**	**49**
	5.1 Introduction	49
	5.2 Software Development Environment Setup	49
	5.2.1 Software Architecture	50
	5.2.2 Hardware Architecture	50
	5.3 Software Configuration Management	50
	5.3.1 Configure Environment Setup	50
	5.3.2 Version Control Procedure	51
	5.4 Implementation Status	51
	5.5 Conclusion	52
CHAPTER VI	**TESTING**	**53**
	6.1 Introduction	53
	6.2 Test Plan	53
	6.2.1 Test Environment	53
	6.2.2 Test Schedule	54
	6.3 Test Strategy	55
	6.3.1 Classes of Tests	56
	6.4 Test Design	57
	6.4.1 Test Description	57

6.4.2 Test Data 59

6.5 Test Case Result 60

6.6 Conclusion 62

CHAPTER VII PROJECT CONCLUSION 63

7.1 Observation on Weakness and Strength 63

7.2 Propositions of Improvements 64

7.3 Conclusion 64

REFERENCES

APPENDIX

LIST OF TABLES

TABLES	TITLE	PAGE
2.1	List of software requirement	16
2.2	List of hardware requirement	16
2.3	Milestones Table	17
2.4	Gantt Chart Table	19
3.1	Software Requirement	31
3.2	Hardware requirement	33
4.1	Packages Description	38
4.2	Data Dictionary (Entity Description)	46
4.3	Data Dictionary for PSP: Error Recorder	47
5.1	Implementation Status	52
6.1	PSP: Error Recorder Test Environment	54
6.2	Test schedule according tasks, activities and duration to carry out testing activities.	54
6.3	Test design record according purpose, test description and expected result.	57
6.4	Test data record according test cases, test data and expected results	59
6.5	Test case record according test case ID, tester, test objective, test data, result.	60

LIST OF FIGURES

FIGURES	TITLE	PAGE
2.1	PSP Process Evolution	11
2.2	PSP tool architecture	12
2.3	Interface for defect form	13
2.4	Milestones	18
2.5	Gantt Chart	20
3.1	Business Flow for Previous System	24
3.2	Overview of PSP: Error Recorder system	27
3.3	Business Flow of PSP: Error Recorder	27
3.4	Use Case Diagram	28
3.5	Interaction Diagram for Login Process	30
3.6	Interaction Diagram for Basic Flow	31
4.1	System software architecture overview based on 3-tier architecture	37
4.2	PSP: Error Recorder system packages	38
4.3	Class Diagram for PSP: Error Recorder	39
4.4	Sample screen for login	40
4.5	Sample screen to view student's errors.	40
4.6	Sample screen to view student report / performance	41
4.7	Overview of Navigation Design for PSP: Error Recorder	42

4.8	Deployment View	43
4.9	Entity Relation Diagram for PSP: Error Recorder	46
5.1	Software Architecture	50
5.2	Personal computer	50
5.3	Version control procedure	51

CHAPTER I

INTRODUCTION

1.1 Project Background

Personal Software Process (PSP) is a guide to use disciplined personal practices to do superior software engineering. The PSP will show you how to plan and track your work and how to consistently produce high-quality software. Using PSP will also give you data that show the effectiveness of your work and identify your strengths and weaknesses. Beside been used by the software engineers, PSP also can or should be used in campus for students who is taking programming subject.

We can see today, majority of universities and colleges offer programming subject to their students. So, it is very suitable to adapt the PSP in universities and colleges in our country to make the students become familiar with PSP. The PSP was designed to help software engineers do good work. It shows them how to apply advanced engineering methods to their daily tasks. It provides detailed estimating and planning methods, show engineers how to track their performance against these plans, and explains how defined processes can guide their work (Watts S. Humphrey, 1997). Thus, it is very suitable now to introduce the PSP to the students before they become real software engineers soon.

There were various types of assessments in PSP. The main focus of this system is to help lecturers to trace all the errors that have been done by students in programming subject, specifically Microsoft Visual C++. So, the lecturers will know where the student's weakness and can more concentrate on them. The students will also get benefit from this system. They can view the mistakes that have been done during the semester.

1.2 Problem Statement(s)

During programming lab session, students usually will face errors while writing or debugging line. The errors will appear during debugging process. Students have to fix them before execute the coding. There were some students who can fix the problems by themselves, but at the other part some student can not settle the problems by their own. They usually depend on friends around them. They did not refer to the lecturer. By depending to their friends, of course they manage to settle down the problems and execute the coding, but they will have difficulties when it comes to individual projects, lab test and during real working environment.

When this situation happens, many people will put their blame onto lecturers and faculty. Some said lecturers did not know how to teach students or more concentrate on other business rather than teaching. While the faculty were blamed because did not prepare the silibus which are relevant to present industrial requirement. Many articles in newspapers and magazines nowadays said that ICT graduates from our local university did not have skills and knowledge needed in industrial today. Many academician and thinkers now discuss why these phenomenons happen. But to put all the blames on lecturers and faculty was an unfair decision.

From experience and observation, usually students have lack of self confidence. They were afraid or feel shy to raise questions when they do not have same thought with lecturers or when they did not understand in certain parts. From the situation, lecturer can not get input from their students and lecturers will face difficulties to trace or detect student's weakness. Students were still feeling like there was a gap between lecturers and them and more believe to peer around them. This system will try to overcome this situation. It will record all the errors done by each student, so lecturers can view their performance and trace their weakness.

1.3 Objective

Every project must have its own objectives, to ensure the project can be developed and run smoothly. Objectives must be stated clearly so it will not only ease for the system developer, but also others who involve in this project. Below are the objectives of this project.

a) Help lecturers to detect student weakness.

One of the purposes of this project is to help lecturers to detect student's weakness. Errors from each lab session will be recorded in certain files and stored in student database. Same errors that been done repeatedly by a student will give lecturers some clue or scope on which part of student weakness.

b) Students can view their performances during the semester.

Beside lecturers who get benefit from this project, another side who also get the benefit is student itself. Student can view mistakes that have they done in each lab session during the semester for guidance about their performance. They will know which part of their weakness and can improve it in future.

c) Reduce fear among students to C++ subject

After view all the errors, lecturers can give guidance to students how to solve or overcome the mistakes done by them. So, student will know that there were always a solution for the mistakes and problems done by them. Hopefully in future, they will not feel afraid to do the C++ programming.

1.4 Scopes

Scopes of the project is very important as to make sure the project development will achieve its goal. It give some overview on what the project goal is all about, the content of this project, limitation and functionalities of the project that want to be develop. Each project has their on scope based on requirements and facilities that they have. Below are the scopes defined for this project.

a) Create unique ID and login for each students and administrator

Each student will have their own ID and password for login. Each time before they start lab session, they must login first. Administrator also will have its own ID to prevent from something unnecessary happen.

b) Record errors for lab session

All errors that have been done by the students during lab session will be recorded and stored in their database. That is the reason why student will have their own unique ID and password. So, lecturers will know errors and weakness of each student based on their ID

c) Develop database for programming students

Each programming student will have their own database. The purpose for developing this database is to store all the errors have been done by student during every lab session.

d) Create a medium to view all the errors done by students

Every error that has been recorded can be view by the lecturers. From this, lecturer will know what kind of errors that always been done by the students. Main purpose of this function is to show the performance of the student. It will help both lecturer and student to define the weakness and which part that should be more concentrates on.

1.5 Project Significant

Information technology (IT) now was rapidly developed and become one of industrial necessary. Most universities and colleges now offer courses in IT field. The question now is how to produce a good graduate in IT especially in software engineering department. Software programming students are generally not aware of practicing good programming habits. As a result, the student is unable to check where their best is and where they need to improve (NurHafeizah Hassan, 2003).

In way to produce a good software engineers, students should be introduce with PSP. The Personal Process Software (PSP) is a framework designed to teach software engineers and also students to do better work. It shows how to estimate and plan work, how to track our performance against these plans and how to improve the quality of the programs we produce.

This system gives benefit to both student and lecturer. It is develop to help students to improve their programming skills by know their weakness based from all the errors recorded from each lab session. At the same time, lecturers can make analysis on which part they should more focus on. Each student comes with their own attitude, habits and abilities. Sometimes students may not have same thoughts with lecturers. Lecturers design some task with thought that students will use the concepts discussed in the earlier task. But sometimes different situation happened.

Even tough the scope of this project is only record and stored errors done by student during lab session, but it can play important roles to improve student skills in programming and become a good software engineer soon. It is an early step to teach new generation of software engineers to apply PSP methods especially when they enter real working environment. Course data on thousands of PSP class programs show that the PSP is effective in improving engineer's planning performance and the quality of the products they produce.

1.6 Conclusion

The Personal Software Process (PSP) was developed at Software Engineering Institute (SEI) by Watts Humphrey. It is designed to bring discipline to the practices of individual software engineers, by providing a framework for measuring and analyzing their development work so that they can produce programs of higher quality and in more predictable manner. More specifically the objectives of PSP are as follow:

- To introduce students and engineers to a process-based approach to developing software.
- To show students and engineers how to measure, estimate, schedule and track their work.
- To show students and engineers how to improve the quality of the programs.

As if we look in our local scene, PSP is not very familiar with our pre-graduate software engineering students in universities and colleges. Supposedly, they should be introduced with PSP methods from campus, so they can apply during real working environment. The PSP can help students and engineers to improve the quality of programs they produced, saving time and cost.

By developing this system, it is hoped that the quality of our software engineering students can be improve. It was design to help students to know their weakness spot, for improvement and to help them master in programming, especially Microsoft Visual C++. It also another helping tool for lecturers to know their students well and which part to be more concentrate on.

The next chapter, Literature Review and Project Methodology will be discussing about fact and finding, study case, project requirement and also project schedule and milestones.

CHAPTER II

LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

The literature study in the previous system and papers that have been discussed and presented before could give more references in system development process. All the advantages from the previous system can be implemented in the system development while we can learn from their weakness to make sure it can be improve.

There were many papers and research that have been done before about PSP and teaching software engineering students. An essay written by Guy Keren, titled "*Why Do Universities Fail Teaching Software Engineering?*" discusses reasons why some software engineering students were hard to survive in real-working environment after finish up their studies in universities. One of his reasons was the university try to teach software engineering as a set of rules of thumb, hoping that the students will follow them. even if only because they are being enforced somehow when grading exercises and exams.

Even if some of the teachers, or teaching assistants, do have experience with large and lasting software projects, often their students don't know about that, and hence don't trust them. When you don't trust someone, you don't listen to advise they give you if you cannot see an immediate benefit (and in software engineering, there is no benefit for very small exercises). Thus, the students actually tend to ignore their teachers' rules of thumb, seeing them as a burden, and this causes more damage than if those rules were never given in the first place (Guy Keren. 2001).

Meanwhile, a book written by Watts S. Humphrey, *"Introduction to the Personal Software Process"* introduced the PSP as a solution to teach engineers and students how to become far more effective, efficient and productive by allowing them to manage their work habits and personal software management techniques.

In order to come out with a good system, we should not ignore the important of case study. Literature review in methodologies, strategies and project analysis should be done to gain more understanding on the project implementation study. It is important to establish solid principles first before design and constructing of the system. The other important things of case study is, we can use the last previous project document as the main reference to describe the main user requirement.

2.2 Fact and Finding

2.2.1 Case Study on Using PSP in Programming Class

Over the years, the evolving of technology, models, approaches as well as issues in system development forced the industries to put as many requirements as possible for an IT position.

Teaching the first year programming students is challenging to both students and lecturers or tutors. The lecturers are very enthusiastic in designing the task for the course completion and the students are busy with adapting process to new learning environment. The tasks are important because they require students to use their skills to new situations. However, do we really care that the tasks are actually served as vehicle to the learning behaviors in the students? For example, often we overlook the idea that the students may not have the same thought of us, the lecturer. The lecturers design task B with the thoughts that the students will use the concepts discussed in task A. But, the fact is, for the students they do not have the thoughts that they are suppose to memorize the yesterday's concepts and implement it today. This is one of the habits as described by Baird and Northfield called poor learning tendency. He points out that poor learning tendency(s) were all characterized by a passive, dependent and uniform approach to learning. The learners may want to succeed but he sees his roles as passive recipients of information and instruction, the lecturer will tell the students what to do, how to do, when to stop, what types of examples worth considering and even whether or not they understand the work. These poor learning tendency, adopted in programming class may lead to the poor programming practice that later, undoubtedly, create the individual programming habits (Nor Hafeizah Hassan, 2003a).

A software engineer candidate who has more skills for a position is most likely to be selected than those with few. Therefore it is a challenge for the academia to provide a course structure that fit the situations. Most of the software engineers have difficulty adopting new methods. This is because they first learned to develop software during their formal educations and have since followed the same practices with a few adjustments and refinements. Since they are comfortable with these methods and have not seen compelling evidence that other methods work better, they are reluctant to try anything new. This problem is compounded by the fact that software engineering students are rarely able to experiment. They were bounded for delivery on short and demanding schedule. An experiment would thus entail considerable risk (Nor Hafeizah Hassan, 2003b).

From the situation occurred by the author, she suggested students should determined to posses multiple skills by using self-checking approach. The reason for this idea is the approach could enable them to established practices through their plans and procedures of tasks being carried out. In most professions, competent work requires the disciplined use of established practices. It is not a matter of creativity versus discipline, but of bringing discipline to the work so that creativity can happen.

The author stated that the closest alternative for this approach is the Personal Software Process (PSP). This is based on what Humphrey had stated before about professional software engineer. A professional software engineer should produce a good disciplined practice in his work. The practice shall be weighted in quantitative measurement in order to determine their level of achievement. Over the years, the computer scientists have come out with quantitative measurements for assessing the software quality produced. The goal of software engineering itself is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software. Thus, the Personal Software Process (PSP) approach that