2016

Duration Estimation of Template Based PHP Software Projects Based on Practical Exams Analysis: A Case Study

Ljubica Kazi, Zoltan Kazi, Radulović B., Duško Letić, Amar Kansara

University of Novi Sad, Technical Faculty “Mihajlo Pupin”, Zrenjanin

DURATION ESTIMATION OF TEMPLATE-BASED PHP SOFTWARE PROJECTS BASED ON PRACTICAL EXAMS ANALYSIS: A CASE STUDY

UDC 007:37-057.875
007:004.3
Original research

Ljubica KAZI[*], Zoltan KAZI[*], Biljana RADULOVIĆ[*], Duško LETIĆ[*], Amar KANSARA[**]
[*]University of Novi Sad, Technical faculty "Mihajlo Pupin", Zrenjanin, Republic of Serbia
[**]Parth Systems LTD, Navsari, Gujarat, India

Paper received: 25.12.2016.; Paper accepted: 5.11.2017.

Abstract: Within information system development, some of the most important issues in software project management are software complexity, software quality and performance, as well as estimation of software project duration and cost. It is of a great importance to introduce these important issues to students during their study time within teaching process. Moreover, in assessment and exams period, proper estimation of students' programming workload is another relevant issue. This paper deals with educational context of estimation of software projects duration, particularly for software developed for the organizational context, i.e. within information systems development. The main result of this paper is presenting teaching methodology within the case study in PHP programming as well as analysis of exams results. Teaching methodology is presented with practical lectures content (template-based programming, OOP, CRUD) and organization of classes, preparation for exams and exams conducting. Exams results analysis is the basis for a case study that demonstrates application of the method for estimation of students' PHP projects workload (duration) planning.

Keywords: students' software projects, estimation, duration, workload, CRUD, OOP, PHP.

I. INTRODUCTION

Common practice of students' theoretical knowledge assessment usually includes essays and multiple-choice exams [1] [2], while practical knowledge and skills assessment includes practical exams [3] and projects [4]. Many factors influence students' performance at knowledge and skills examination (such as student effort, student ability and teacher input [1]), but it is of a great importance to do proper design and evaluation of tests before their actual application with students [5]. Adequate workload estimation is one of the most important aspects in appropriate design of various examination forms in students' knowledge and skills assessment.

Students' projects could improve their project management skills [6], as well as programming skills, writing and culture [7]. According to Project Management Institute, "a project is a temporary endeavor undertaken to create a unique product, service, or result." [8] Project management, as important factor of any project success, expresses many potential paths in research and further development such as [9]: optimization in project scheduling, project selection, earned value analysis, task allocation and notification, cooperation of task operators, subcontractors and stakeholders, management of uncertainness and completion incentives, learning from and between projects, scalability of agile methodologies, sustainability of project management. Rational projects management is based on established system for key-performance criteria estimation and measurement, such as well-known "Iron Triangle": cost, time and quality [10]. It is of a great importance to define appropriate set of measures according to the target benefit scope.

Aim of this paper is to present research in duration estimation of students' software projects in PHP programming, based on analysis of results in PHP programming practical exams. Research is conducted as a case study with students' practical exams and students' projects in PHP programming at University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia. The proposed approach is based on idea of using analysis of success and duration of practical exams in PHP programming as foundation for

estimation to possible duration of students' projects in PHP programming. This way, real data is used for more precise estimation of students' projects duration, i.e. workload. Structure of this paper consists of: related work analysis, research methodology description, teaching methodology description, results analysis and conclusions.

II. RELATED WORK

During many years, research and practical efforts are made towards more precise estimation of software projects, particularly in the duration (time) aspect, which relates to costs and could affect quality of software product. Duration of project is closely related to workload. i.e. efforts that are usually expressed in man-hour, man-day or man-month [11] measurement units. Some of the estimation methods are based on experiences from previous projects or from estimation of experts (which have foundation of estimations in their own experiences) [12]. Some of these experience-based (empiric) estimation methods are Empirical parametric estimation models [12], where parameters are:

- lines of codes (LOC),
- functional points (FP),
- data flow graph and object-relation graph elements analysis,
- number of pages of system documentation,
- elements of object-oriented system analysis and design (number of objects and services of the system)
- COCOMO model that includes various parameters such as estimated full LOC of future software product, subjective assessment of hardware, personnel, project, and additional cost drivers.

Other estimation models include approaches such as [12]:

- Empiric non-parametric estimation models (use data on projects realized earlier, with similar characteristics in complexity etc.).
- Expert estimates (consulting other software experts and combining different opinions with methods such as Delphi).
- Analogue estimation models (analogies of new project with similar ones that are completed previously).
- Downward (component based) and upward (from implementation of parts to integration) estimates
- Percentage method [13], where the system development is divided into phases and estimated time is expressed with percentage (e.g. in Table 1):

Table 1: Percentage of efforts in system development, divided by phases (improved from [13])

Requirements analysis	11%
Logical system specification	5%
Physical design	10%
Coding and module test	52%
Integration	5%
System test	12%
Deployment	5%

In aim to gain simple useful practical directions, software estimates resulted in "rules of thumb", as a kind of "rules to apply in practice". These simplified rules are considered not exact, inaccurate and not liable standards, but could easily be applicable. Some of such rules are defined in year 1999, presented as conclusions from a workshop [13]:

- *The productivity of a programmer is approximately 5 FP per person month (PM).*
- *1 Function Point corresponds to about 100 LOC (Lines Of Code) of COBOL - 20 LOC for OO or generators and more than 300 LOC of Assembler.*

From this example, it could be concluded that as programming technology changes, definitions and pragmatic "rules of thumb" change. In contemporary programming environment, with object-oriented design patterns and frameworks, as well as highly automated software development tools, productivity of programmers changes. Estimation of software project duration should be related to particular

implementation technology. Therefore, in this paper, particular software development style (template-based) and programming language (PHP) is analyzed.

III. RESEARCH METHODOLOGY

Research objective of this paper is to present a case study on students' software project duration estimation. Estimation is based on analogy of programming effort made within:

- Practical exam (conducted in computer laboratory classroom), and
- Software project (implemented at home working environment).

Analogy is possible because both type of programming efforts use:

- The same programming language PHP,
- The same front-end design, available as template,
- The same back-end template, available with all coding samples/modules (CRUD – Create/Read/Update/Delete with Tabular presentation of data, Search and Print support – named "essential programming module CRUDTSP", implemented with OOP – object-oriented programming).
- The same development tools at classroom and at home: XAMPP (Apache web server, MYSQL database management system with visual tool PHPMYADMIN, programming tool Notepad ++).

Main hypothesis for this research is:

It is possible to have estimation of software project programming efforts based on analogy with programming efforts within practical laboratory exam.

Hypothesis will be examined within two research objectives:

1st research objective

The basic idea: Programming efforts within practical laboratory exam is measured in man-hour where "essential programming module" is set to be finalized within certain limited time period. In software related to business, i.e. information systems, is essential programming module consist of elements of user design and appropriate coding for enabling functions such as: data insert, update, delete, tabular presentation, search and print (CRUDTSP). By measuring time that is adequate to finalize completely functional CRUDTSP, we can define precise "essential programming module" duration.

Research method: Analysis of the practical exams (in CRUDTSP implementation with PHP, design template and OOP) results success and duration for each students' work. Maximum duration time for practical exam is 3 hours. It is to be determined if is it possible to finalize CRUDTSP PHP/template/OOP web application within 3 hours. Research population consists of practical exams from students in year 2016.

2nd research objective

The basic idea: If CRUDTSP PHP/template/OOP web application is possible to implement within 3 hours, then the whole software implementation duration, where implementation is based on exactly the same technology, could be estimated with simple analysis of USE CASE diagram for determination of all software functions, determination of number of CRUDTSP submodules within the use case diagram, multiplication of that CRUDTSP submodules number with time needed for single CRUDTSP submodule implementation (i.e. 3 hours in this analogy) using simple formula:

$$SPPD = N * D(CRUDTSPe); \quad (1)$$

Where:

SPPD – software project programming duration,

N – number of CRUDTSP submodules within the whole software project,

D – duration function,

CRUDTSPe – duration of a single CRUDTSP module implementation, within exam limited time period.

Research method: A case study with one student's project in analysis of USE CASE diagram, determination of sub-modules and application of formula that integrates single CRUDTSP duration with the whole planned software functions.

IV. THE TEACHING PROCESS

PHP teaching at University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia is conducted at several teaching subjects, such as: Elective subjects ("Internet tools and services", "Web design"), mandatory subject ("Information systems 2"). As per previous years statistic, these two elective subjects were chosen from approx.. 20% of the whole students population in higher education of information technologies at University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin. Since PHP programming knowledge and skills are very important (PHP programmers were second as required at job market in Novi Sad region in year 2013 analysis [14]), since February 2014. PHP was introduced within mandatory subject "Information systems 2" within final bachelor level study of Information Technologies Engineering at University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin.

Teaching process within subject "Information systems 2" is organized within three levels:

1. Basic level – implementation of database connectivity with standard mysql and mysqli programming commands, implementation of login, menus, sessions, data navigation between pages, basic CRUD implementation with standard PHP commands with one-table database, introducing OOP with extraction of database connectivity to separate class, creating simple entity/model class and including CRUD SQL queries within appropriate class methods. Implementation include inserting, update, delete, tabular presentation, filtering/search and printer friendly page.

2. Advanced basic level – working with two-tables database, combo box within user interface that uses data from other table, transactions, stored procedures and views, parametric printer friendly page, master-detail data input and print

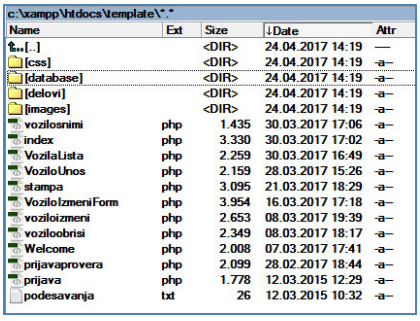
3. Advanced level – demonstration of Laravel and Symphony PHP frameworks, demonstration of Wordpress and Joomla.

Within each of first 2 blocks (levels) classes, there are demonstrations and implementation of examples. At the end of lectures, there is dedicated class for exam preparation, where students could implement example testing application, by using finalized template (created from previous classes) and transform it according to another semantic (given with textual description of needed database/table structure).

Knowledge and skills examination is organized alternatively (student can choose one of these two types of examination forms) as: Two practical laboratory exams (after each of first two levels, i.e. basic level exam and advanced basic exam), or student's project, which includes all elements from both practical laboratory exams.

V. TEACHING MATERIAL AND TEMPLATE

During implementation of practical exams or students' projects, all students are allowed to use previously created template, which included both front-end and back-end programming part. The template [15] could be downloaded and copied to "htdocs" folder within XAMPP (Figure 1). Basic elements of front-end part of template (baner, status strip, menu, main content block and footer) are presented at Figure 2.


c:\xampp\htdocs\template*.*

Name	Ext	Size	↓Date	Attr
[..]		<DIR>	24.04.2017 14:19	——
[css]		<DIR>	24.04.2017 14:19	-a--
[database]		<DIR>	24.04.2017 14:19	-a--
[delovi]		<DIR>	24.04.2017 14:19	-a--
[images]		<DIR>	24.04.2017 14:19	-a--
vozilosnimi	php	1.435	30.03.2017 17:06	-a--
index	php	3.330	30.03.2017 17:02	-a--
VozilaLista	php	2.259	30.03.2017 16:49	-a--
VoziloUnos	php	2.159	28.03.2017 15:26	-a--
stampa	php	3.095	21.03.2017 18:29	-a--
VoziloIzmeniForm	php	3.954	16.03.2017 17:18	-a--
voziloizmeni	php	2.653	08.03.2017 19:39	-a--
voziloobrisi	php	2.349	08.03.2017 18:17	-a--
Welcome	php	2.008	07.03.2017 17:41	-a--
prijavaprovera	php	2.099	28.02.2017 18:44	-a--
prijava	php	1.778	12.03.2015 12:29	-a--
podesavanja	txt	26	12.03.2015 10:32	-a--

Figure 1. Files and folders of template for front-end and back-end PHP programming

Figure 2. Structure of front-end part of template

Template (implemented as example of Tourism Agency) includes implementation for all CRUD operations, as presented at next figures. Students were given this template (with both front-end design and back-end programming) to use it and transform to another semantic. The semantic of the template is tourism agency, while students were given (within classes and exams time) to transform the template web application to adjust to another semantics, as per given task description (such as information systems support to production industry, healthcare, education and many other business processes support).

Figure 3. Reading and "tabular" presentation of data

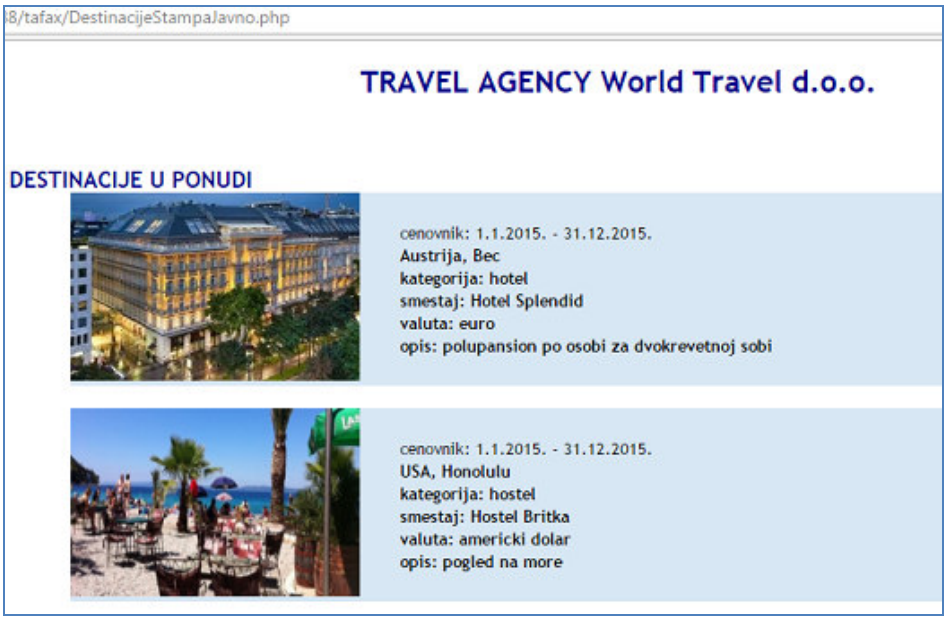


Figure 4. Printer friendly page

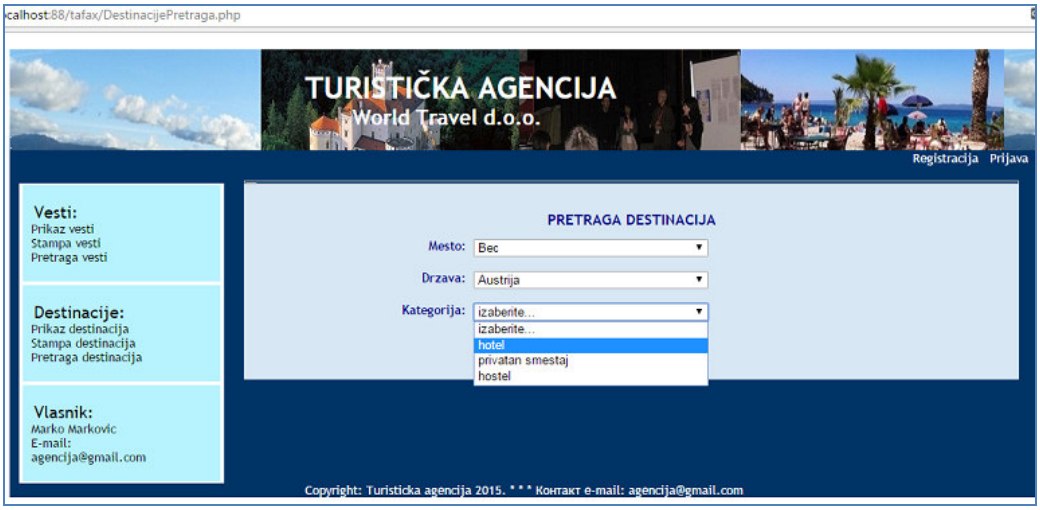


Figure 5. Filtering page

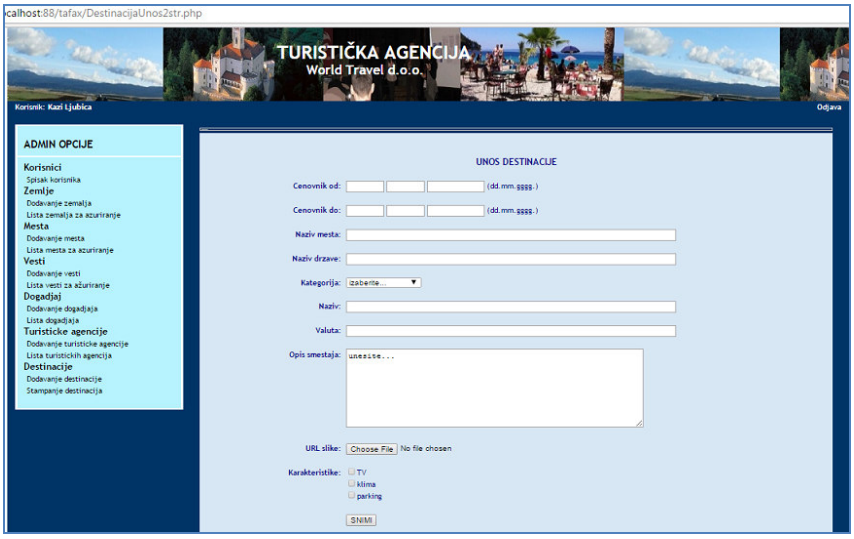


Figure 6. Data input page

Figure 7. Update and delete enabling list with buttons for "update", "delete"

VI. RESULTS

1st research objective results

Total population of students' results for analysis within first research objective (attended teaching process and exams within subject "Information systems 2" during year 2016) is presented with Table 2.

Table 2. Exam results in Information systems 2 with population of students, in year 2016.

	%	max=5	max=5	max=30	max=30
Student No	**Attendance**	**Class work**	**Class activity**	**Exam 1**	**Exam 2**
1	100	5	5	30	30
2	100	5	5	30	29
3	100	5	5	30	30
4	100	5	5	30	30
5	100	5	2	28	14
6	91	5	3	28	29
7	91	5	2	25	30
8	82	4	5	28	30
9	82	4	5	29	30
10	82	4	3	28	30
11	73	4	5	30	27
12	73	4	1	29	29
13	64	3	1	28	30
14	64	3	0	28	30
15	64	3	0	29	29
16	64	3	0	29	14
17	55	3	5	30	30
18	55	3	5	28	30
19	55	3	1	28	30
20	55	2	1	25	26
21	*55*	*2*	*0*	*7*	*14*
22	*36*	*5*	*5*	*30*	*30*

23	*36*	*2*	*0*	*21*	*30*
24	*36*	*1*	*0*	*30*	*27*
25	36	2	0		
26	27	1	0	15	
27	18	1	0	29	
28	18	1	0	27	6
29	18	1	0		
30	18	1	0		
31	18	1	0	29	0
32	9	0	0		
33	9	1	0		

Table 2 presents list of students' work in PHP programming at regular classes and at exams. Column 2 presents percentage of students' attendance at practical laboratory classes, where students were practicing PHP programming. Success of students' work within classes exercises work is expressed in points (third column – quality of work, fourth columns – activity (agility, preciseness, discussions, creativity)). Columns 5 and 6 present results of evaluation of students' work within practical laboratory exams in PHP programming. Maximum number of points for each exam is 30 and each of the two exams were organized with duration limitation of 3 hours. Each exam was organized that each students had one computer and his personal programming task to finalize. So duration could be expressed as 3 hours per student.

Analysis of previous results:

- Population for research on 1st research objective consist of 33 students' work within Information systems 2 teaching subject at University of Novi Sad, Technical faculty "Mihajlo Pupin" Zrenjanin, Serbia
- Within the research population, 63.64% were regularly-attending students, while 36.36 % were non-regularly attending students
- Within regularly-attending population, 95.24% of students passed practical exams, i.e. only one student did not pass practical exams (achieved 7/30 and 14/30 within two exams).
- Within non-regularly attending population, 25% of students passed practical exams
- Within complete population, (20 + 3) / 33, i.e. 69.69% of students passed practical exams.

Conclusion from the 1st research objective is that regularly-attending students could successfully finalize each practical exam within 3 hours duration limit. This leads to conclusion that 3 hours is sufficient time to implement PHP application with basic CRUDTP functionality.

2nd research objective result

Having results from 1st research objective, it is possible to start activities in 2nd research objective. A case study of estimation of complete software programming effort is based on analysis of a use-case diagram (Figure 8.) combined with results of 1st research objective.

Figure 8. Use case diagram for example of tourism agency web application software project

Figure 8. presents a use case diagram for software functions in software project for tourism agency web application development. There are two actors (software user profiles): TA officer and TA client. TA officer adds destinations and TA client adds reservations on destinations. This is a simplified example of use-case diagram with only essential software functions, that is created in aim to illustrate the idea of the proposed methodology and to enable application of the formula to calculate estimated duration for software project duration.

Since within the first research objective was proved that the “essential programming module” (consisting of software functions on data input, update, delete, tabular presentation with filter and printing = “CRUDTPe”, i.e. CRUDTP essential module) could be finalized within 3 hours limit, in this section this fact will be used for determination of the whole software programming effort calculation. Use-case diagram in this case study is analyzed to determine number of CRUDTP modules and upon that number, total programming effort is calculated to express duration of programming for the whole software project with multiple functions.

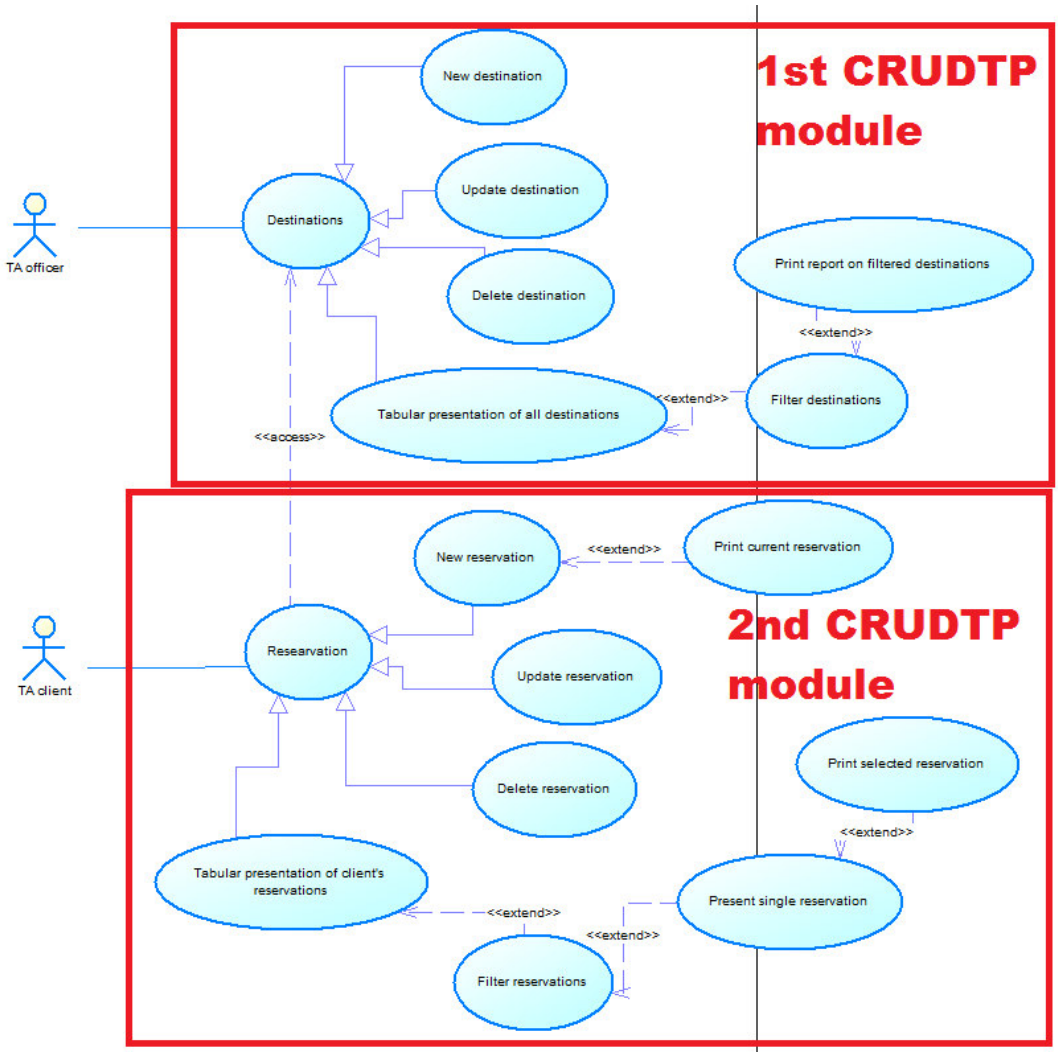


Figure 9. Analysis of use case diagram for determination of number of CRUDTP modules

As per analysis of the example use case diagram, there are 2 CRUDTPe modules within the specified set of software functions. Therefore, the formula could be calculated from formula: SPPD = N * D(CRUDTSPe);

N = 2 (i.e. 2 CRUDTP modules), D(CRUDTSPe) = 3 hours/student

SPPD = 2 * 3 hours = 6 hours per student.

Complete programming effort for the whole software project is estimated and calculated as 6 hours per student.

VII. CONCLUSION

The research presented in this paper is related to method for estimation of software project duration within educational context. It is of a great importance to estimate students' workload in aim to enable good teaching and knowledge evaluation conditions for students.

In introductory part of this paper, it has been shown that related work in this field consists of many methods for software project duration estimation. Some of these methods are based on experts' opinions which are consulted, while others are, similarly, based on analogy with previous projects of the same complexity. Some related work in software project duration estimation emphasizes the importance of implementation technology which could correlate with results of estimation.

This paper had two research objectives – first to determine the essential programming module duration and second – to analyze use case diagram to determine the number of essential programming modules

within the whole software project. This way, it is possible to calculate whole software project programming effort by simple multiplication of number of essential programming modules (CRUDTPe) with duration for one CRUDTPe module implementation.

First research objective is finalized with the conclusion that 3 hours is sufficient time to implement CRUDPe module in PHP programming, based on previously given software template (PHP example application with both front-end and back-end implementation). This result is used within second objective. An example use-case diagram is analyzed to determine number of CRUDTPe modules and since it was a simplified use-case diagram (prepared for this paper) it consisted of 2 CRUDTPe modules. Since duration of one CRUDTPe module implementation is 3 hours, for the whole software project that consist of 2 CRUDTPe modules the duration for the whole software project programming is 6 hours per student.

The proposed method in this paper could be applied for estimation of software projects duration in professional environment, as well as within educational context. Future research could include empirical evaluation of the proposed method with other technologies. The formula that is used within the proposed method could consider diversity of CRUDTPe modules, in aim to have more precise estimations.

REFERENCES

[1] Williams R, Clark L: “College students’ ratings of student effort, student ability and teacher input as correlates of student performance on multiple-choice exams”, Journal Educational Research, Vol 46, 2004, Issue 3, pp. 229-239. (Online December 2010)

[2] Zeider M: “Essay versus Multiple-Choice Type Classroom Exams: The Student’s Perspective”, The Journal of Educational Research, Volume 80, 1987, Issue 6, pp 352-358 (Online January 2015)

[3] Newhouse C. P: “Computer-based Practical Exams in an Applied Information Technology Course”, Journal of Research on Technology in Education, Volume 45, 2013, Issue 3, pp. 263-286.

[4] Sokolik, M., Tillyer A. "Beyond Portfolios: Looking at Students' Projects as Teaching and Evaluation Devices." College ESL 2.2 (1992): 47-51.

[5] Smith, J. E., Merchant S. "Using competency exams for evaluating training." Training & Development Journal, Aug. 1990, p. 65+. Academic OneFile, Accessed 24 Apr. 2017.

[6] Sariwati M S, Zaimy J. J.. Norina A. J.: "Assessment of Project Management Skills and Learning Outcomes in Students’ Projects", 6th International Conference on University Learning and Teaching (InCULT 2012), Procedia - Social and Behavioral Sciences 90 (2013) 745 – 754.

[7] Kitao K: "Web-based students' projects: Teaching Awareness of culture and writing", Japan Society for the Promotion of Science, Doshisha University, Kyoto, Japan, 2001.

[8] Project Management Institute: “A guide to the project management body of knowledge”, Fifth Edition, 2013.

[9] Hall N, “Project Management: Recent developments and research opportunities”, J Syst Sci Eng (Jun 2012), 129-143. ISSN 1004-3756.

[10] Atkinson R: “Project management: cost, time and quality, two best guesses and a phenomenon, it’s time to accept other success criteria”, International Journal of Project Management, Vol. 17, No 6, pp 337/342, 1999.

[11] Brooks, F. P.: “The Mythical Man Month”, Addison-Wesley, 1975.

[12] Zivadinovic J, Medic Z, Maksimovic D, Damnjanovic A, Vujcic S: “Methods of effort estimation in software engineering”, Proceedings of I International Symposium Engineering Management and Competitiveness 2011 (EMC 2011), Zrenjanin, Serbia, pp. 417-422.

[13] Bundschuh M: “Estimation of IT-projects, Highlights of a Workshop”, Metrics News, Vol 4, No 2, December 1999, pp. 29-37.

[14] Kazi Lj, Radulović B, Ivković M, Bhatt M, Stanciu O: ”IT jobs market in Serbia – a preliminary analysis”, Proceedings International conference “Applied Internet and Information Technologies”, 2003, pp. 300-304.

[15] PHP programming template (front-end and back-end) for exams and student projects at University of Novi Sad, Technical faculty “Mihajlo Pupin” Zrenjanin, Serbia, http://www.tfzr.rs/Predmet/informacioni-sistemi-2/download