Applying Software Development Risk Taxonomy in Use Case Points Complexity Factor

by Hafid Kholidi Hadi

Submission date: 30-Mar-2023 03:47AM (UTC+0700)
Submission ID: 2050280931
File name: opment_Risk_Taxonomy_in_Use_Case_Points_Complexity_Factor_3.pdf (276.58K)
Word count: 3168
Character count: 17091

2022 5th International Conference of Computer and Informatics Engineering (IC2IE)

Applying Software Development Risk Taxonomy in Use Case Points Complexity Factor

Hafid Kholidi Hadi, Renny Sari Dewi, Fresha Kharisma, Achmad Kautsar, Anita Safitri
Department of Digital Business
Universitas Negeri Surabaya
Surabaya, Indonesia
{hafidhadi, rennydewi, freshakharisma, achmadkautsar, anitasafitri}@unesa.ac.id

***Abstract*—The studies of software development risk continues to grow. One is the software development risk taxonomy released by Software Engineering Institute (SEI). Unfortunately, no risk management research has been explicitly combined with software size estimation. Therefore, this study aims to apply risk management over 13 technical and eight environmental factors in the Use Case Points (UCP) method. The dataset consists of 345 risk factors by any sources, then mapped and justified by three experts. This mapping generates the risk frequency, which is finally modified by adding it to the origin weight by UCP. The results of the application of risk to environmental factors dominate as much as 76.81 percent compared to technical factors. There are 2 objects that the most influencing risks, such skill and motivation for both the developer team and end-user. For further research, this study still challenges how risk management can be integrated to obtain better accuracy toward software effort estimation.**

Keywords*—software risk, risk management, software estimation, software effort, use case points

I. Introduction

Research on risk management in software development projects was started by Barry W. Boehm in 1991 [1]. In risk management, the risk is assessed and monitored so that the project's scope, time, and cost are under control. According to Boehm, assessing risk requires 3 stages: identification, analysis, and prioritization. Meanwhile, 3 phases are needed for risk control: planning, resolution, and monitoring. This study resulted in a top-10 risk which was then continued by Carr et al. [2] as a researcher at Software Engineering Institute. At that time, SEI launched a software development risk taxonomy which was expected to be a guide in determining risk categories.

According to a survey by the Project Management Institute (PMI), the potential failure of a software development project if the risk is not anticipated and controlled properly is 27% [3]. Given the importance of risk management in software development projects, the Project Management Body of Knowledge (PMBoK) guidance makes the risk management knowledge area an issue that deserves attention after project managers' scope, time, and cost management [4].

Potential risks always overshadow software development project activities. Therefore, a project manager must predict potential risks (known as a risk register) in his project planning document to control risk. The risk register is then used as the basis for project cost considerations. The calculation of the cost in the project planning document has been anticipated by several researchers, which is called software size and effort estimation. The magnitude of the potential risk should be directly proportional to the efforts of the software development team. However, this allegation has never been answered because no software measurement method has been integrated or combined with risk other than the Constructive Cost Model (COCOMO) method belonging to Boehm, the inventor of risk management [5], [6].

Some software measurement methods are commonly used by researchers and business people in software development, one of which is the Use Case Point (UCP) method discovered by Gustav Karner [7]. The UCP method comes from the story of the business process owner. However, project managers should be responsive to potential risks, even if only based on use case scenarios. From that narrative, it is then used to calculate the size of the software development team's effort.

After reviewing many related research articles, combining risk factors with the software size calculation method to determine the amount of software development effort is the importance of this risk mapping. Therefore, this study aims to remap the risk factors used as a reference by researchers in categorizing risks, especially SEI's risk taxonomy. From mapping the risk to the complexity factor in the UCP method, the weight of the percentage of technical and environmental complexity factors can be adjusted.

II. Literature Review

A. Use Case Point (UCP) Method

Function Points Analysis (FPA) is the method that inspired the birth of UCP [8]. The UCP method focuses on calculating actor weights and the complexity of the use case itself. Meanwhile, the formulation of technical and environmental complexity factors was developed from the main theory, namely the FPA method. The fact about UCP is the only method that is not recognized by international standards [9]. UCP is considered not qualified and illogical in its mathematical operations.

Several UCP studies have proposed multiple-sided adjustments. Adjustments to actor weight [10], use case transaction weight [11], [12] to technical and environmental factors [13]. However, there is no single study linking risk with the UCP method. Technical and environmental factors in the UCP method can be seen in Tables I and II.

TABLE I. Technical Factors [7]

ID	Factors to Contribute to Technical Complexity	Weight
TF1	Distributed System	2.0
TF2	Portability	2.0
TF3	Throughput respond apps to user	1.0

ID	Factors to Contribute to Technical Complexity	Weight
TF4	End-user efficiency	1.0
TF5	Internal processing complexity	1.0
TF6	Code reusability	1.0
TF7	Able to modify	1.0
TF8	Concurrency	1.0
TF9	Security feature	1.0
TF10	Access availability to the third party	1.0
TF11	User training	1.0
TF12	Installation ease	0.5
TF13	Operational ease	0.5

TABLE II. ENVIRONMENTAL FACTORS [7]

ID	Factors to Contribute Efficiency	Weight
EF1	Requirement stability	2.0
EF2	Familiar with method	1.5
EF3	Object-oriented programming experience	1.0
EF4	Team motivation	1.0
EF5	Software development experience	0.5
EF6	Analytical skill	0.5
EF7	Part-timer dependency	–1.0
EF8	Difficulty of programming	–1.0

The weight for each technical and environmental factor (see Tables I and II) refers to the original method of UCP [7]. Karner pays great attention to the weights of EF7 and EF8 by giving a minus value. If the given scale (0 to 5) is getting bigger, the risk faced by the development team is getting higher too.

B. *Software Development Risk Taxonomy*

The Software Engineering Institute (SEI) released the software development risk taxonomy (SDRT) in 1993 so that it could be used as a reference for risk identification [2]. Broadly speaking, SEI groups risks into 3 classes and 13 elements. Details of risk classes and elements based on the SEI taxonomy can be seen in Fig. 1.

Fig. 1. Software development risk taxonomy by SEI

Fig. 1 shows SEI's software development risk taxonomy. However, several studies propose risk grouping based on academic and practical needs as risk develops. Some of them are as follows:

- According to [14], risks can be grouped into 5 phases in the software development lifecycle, namely planning, analysis, design, implementation and maintenance.
- Survey research conducted [15] stated that risk is divided into 6 dimensions, namely organizational environment, user orientation, requirements, team, and planning and control.
- Research [16] has a different opinion from others, namely that risk is categorized based on its class objects such as requirements, cost, scheduling, quality, and business.
- In line with research [14], a survey conducted by [17] also proposed risk categories based on SDLC, namely analysis and planning, design, coding, testing, and maintenance

III. RESEARCH METHOD

This study uses a quantitative and qualitative approach. The main purpose of this study is to readjust technical and environmental factors' weight after combining risk taxonomy. If the risk has been appropriately mapped, then the new weight of each technical and environmental factor is obtained. Based on our research roadmap, future research needs to calculate software effort prediction based on the risk factor.

A. *Data Collection*

This study uses datasets taken from published research articles [14], [15], [16], [17] and PhD thesis [18]. Details of each identified risk item from various sources can be seen in Fig. 2 and Table III.

Based on Fig. 2, risk data obtained from various references must go through a preprocessing stage, namely removing duplicate data or similar terms [19]. We have also ensured that PhD thesis dataset [18] did not refer to the other datasets. After removing duplicates and similarity terms contained in Table III, the total dataset processed was 345 risk factors.

TABLE III. RISK DATASET

Dataset	Publication Type	Risk Factor
[14]	Journal	50 records
[15]	Journal	27 records
[16]	Journal	90 records
[17]	Journal	64 records
[18]	PhD thesis	148 records

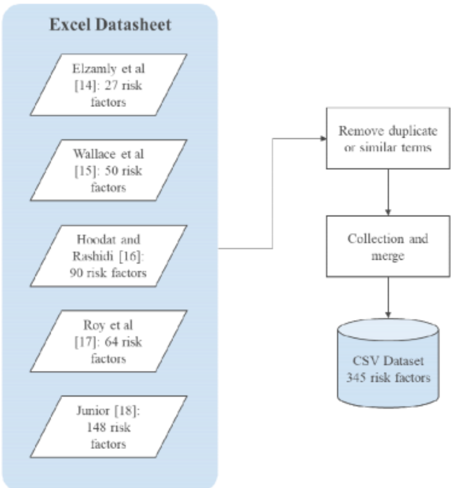

Fig. 2. Final risk dataset

Based on Table III, the dataset we have obtained represents software risk management with several approaches. The explanations are below:

- Study [14] describes the top-50 risk factors, which are classified based on the software development life cycle, which has an output of a software risk dataset. In line with this study, research by [17] also claimed that there are 64 risk factors that are categorized into 5 software development stages, such as planning and analysis, design, coding, testing, and application maintenance.
- Wallace et al. divided software risk items by 6 risk dimensions, namely organizational environment, user orientation, project complexity, requirements, planning and control, and team [15]. All risk dimensions defined 27 risk factors.
- A dataset by [16] had 90 risk factors categorized into 5 risk classes: requirements, cost, quality, scheduling, and business orientation.
- Risk factors summarized by [18] which is stated in PhD thesis as a whole dataset. A number of 148 software risk items were identified and analyzed by IT experts.

B. Research Step-by-Step

By using expert judgment, a total of 345 risk factors were mapped into 13 technical factors and 8 environmental factors. Mapping risk is not easy using machine learning techniques (especially the classification approach).

The expert qualification in question is experienced as a project manager of at least 7 years and well-educated on risk management and budgeting. There are 3 experts involved in this research. There are five steps to conduct this research as follows:

Step 1: Determine keywords (tokenization) that have a very close meaning (synonims) to technical or environmental factors in the UCP method.

Step 2: Count the number of frequency tokens that appear from the standardized dataset.

Step 3: Justify and map based on risk keywords against each technical and environmental factor.

Step 4: If there is a risk that is ambiguous or has the potential that claims two or more categories of similar factors, then experts need to be justified by an approach that is more technically or environmentally.

Step 5: After the risk mapping is completed, the adjustment weight calculation is obtained by averaging the origin weight and risk frequency.

IV. Result and Discussion

The research results obtained are described in detail as follows.

A. Mapping of Software Development Risk

A total of 345 risk factors were done and mapped by MS Excel. Chapter 3, the results of the execution of Step 1 to Step 4 can be seen in Table IV.

TABLE IV. Risk Mapping Towards Technical Factors

Technical Factor	Risk Selection
Distributed System	5
Portability	7
Throughput respond apps to user	2
End-user efficiency	6
Internal processing complexity	13
Code reusability	6
Able to modify	11
Concurrency	2
Security feature	3
Access availability to the third party	5
User training	14
Installation ease	2
Operational ease	4
Total Risk Factor	**80**

Table IV shows that only 80 of 345 risk items were mapped into 13 technical factors. This result is quite surprising that the risk to technical factors in the UCP method represents only 23.19 percent.

The mapping of risks to environmental factors is shown in Table V. There are 265 of 345 risk items that outperform non-technical factors.

TABLE V. Risk Mapping Towards Environmental Factors

Environmental Factor	Risk Selection
Requirement stability	27
Familiar with method	6
Object-oriented programming experience	9

Environmental Factor	Risk Selection
Team motivation	51
Software development experience	90
Analytical skill	69
Part-timer dependency	4
Difficulty of programming	9
Total Risk Factor	**265**

According to Table IV and V, there is an interesting fact that 76.81 percent of the risk becomes a burden that aggravating environmental factors. The top 3 environmental factors that represent the overall risk are software development experience, analytical skills, and team motivation.

Therefore, the experts conclude that the highest risk arises from 2 objects: the developer team and the end-user. The mapping results are an initial contribution to research on the integration between risk and software size prediction, especially the UCP method.

B. *Applying Risk Factors in Technical or Environmental Weight*

To apply risk factors to the UCP method, the authors follow Step 5 in Chapter 3. The Risk Weight column in Table VI represents the decimal of the frequency of occurrence of the risk divided by 80 (as total risk on the technical factor). While the Adjusted Technical Weight (ATW) is obtained from the original weight of the UCP method plus the risk weight according to the i-th technical factor. The main reason for adding risk weight ($Risk$) is that the amount of risk should be directly proportional to the increase in the software development effort. We propose (1) to get the ATW score.

$$ATW_i = Weight_i + Risk_i \tag{1}$$

TABLE VI. ADJUSTED WEIGHT OF TECHNICAL FACTORS

ID	Weight	Risk Weight	Adjusted Technical Weight
TF1	2.0	0.06	2.06
TF2	2.0	0.09	2.09
TF3	1.0	0.03	1.03
TF4	1.0	0.08	1.08
TF5	1.0	0.16	1.16
TF6	1.0	0.08	1.08
TF7	1.0	0.14	1.14
TF8	1.0	0.03	1.03
TF9	1.0	0.04	1.04
TF10	1.0	0.06	1.06
TF11	1.0	0.18	1.18
TF12	0.5	0.03	0.53
TF13	0.5	0.05	0.55

Likewise, to calculate the weight of the adjustment of environmental factors that have taken into account the risk. Adjusted Environmental Weight (AEW) is obtained from the addition of its origin weight by Karner and risk weight. The Risk Weight column in Table VII represents the decimal of the frequency of occurrence of the risk divided by 265 (as total risk on the environmental factor). After operating (2), the AEW is obtained.

$$AEW_i = Weight_i + Risk_i \tag{2}$$

TABLE VII. ENVIRONMENTAL FACTORS

ID	Weight	Risk Weight	Adjusted Environmental Weight
EF1	2.0	0.10	2.10
EF2	1.5	0.02	1.52
EF3	1.0	0.03	1.03
EF4	1.0	0.19	1.19
EF5	0.5	0.34	0.84
EF6	0.5	0.26	0.76
EF7	–1.0	0.02	–0.98
EF8	–1.0	0.03	–0.97

The implication of this research can use to readjust technical and environmental complexity factors in UCP method to estimate software development effort.

V. CONCLUSION

From the description above, we can conclude several points as follows:

- A dataset consisting of 345 risk factors is obtained from 5 sources.
- From the results of risk mapping on technical and environmental factors in the UCP method, about 23.19 percent represents the technical one (80 risks). Then, the environmental factor is dominant to 76.81 percent (265 risks).
- After being mapped into environmental factors, the most significant risks are software development experience, analytical skills, and team motivation.

This research is a continuation of recommendations from previous studies, namely risk grouping into software development activities [19]. Furthermore, the results of this study become the basis for integrating risk into the UCP method in order to obtain a more comprehensive software effort prediction.

ACKNOWLEDGEMENT

We would like to deliver our gratitude to Universitas Negeri Surabaya for facilitating and granting this research as stated in contract number 663/UN38/HK/PP/2022.

REFERENCES

[1] B. W. Boehm, “Software risk management: Principles and Practices,” *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, 1991.

[2] M. J. Carr, S. L. Konda, I. Monarch, F. C. Ulrich, and C. F. Walker, “Taxonomy-Based Risk Identification Taxonomy-Based

Risk Identification," Pittsburgh, Pennsylvania, 1993.

[3] PMI, "Pulse of the Profession: Success in Disruptive Times," 2018.

[4] K. Känsälä, "Integrating risk assessment with cost estimation," *IEEE Softw.*, vol. 14, no. 3, pp. 61–68, 1997.

[5] B. Boehm *et al.*, "COCOMO II Model Definition Manual," *Univ. South. Calif.*, vol. 4, no. 1, pp. 6–6, 2000.

[6] B. W. Boehm, "Software Engineering Economics," *IEEE Trans. Softw. Eng.*, vol. SE-10, no. 1, pp. 4–21, 1984.

[7] G. Karner, "Resource Estimation for Objectory Projects," 1993.

[8] A. J. Albrecht, "Measuring application development productivity," in *IBO Conference on Application Development*, 1979, pp. 83–92.

[9] A. Abran, *Software Metrics and Software Metrology*. Canada, US: John Wiley & Sons, Inc., 2010.

[10] M. Ochodek, J. Nawrocki, and K. Kwarciak, "Simplifying effort estimation based on Use Case Points," *Inf. Softw. Technol.*, vol. 53, no. 3, pp. 200–213, 2011.

[11] R. S. Dewi and R. Sarno, "Software effort estimation using early COSMIC to substitute use case weight," in *Proceedings - 2020 International Seminar on Application for Technology of Information and Communication: IT Challenges for Sustainability, Scalability, and Security in the Age of Digital Disruption, iSemantic 2020*, 2020.

[12] A. B. Nassif, L. F. Capretz, and D. Ho, "Enhancing Use Case Points Estimation Method Using Soft Computing Techniques," *J. Glob. Res. Comput. Sci.*, vol. 1, no. 4, pp. 12–21, 2010.

[13] M. M Kirmani, "Re-UCP Effort Estimation for Web Application Development," *Orient. J. Comput. Sci. Technol.*, 2018.

[14] A. Elzamly, B. Hussin, and N. M. Salleh, "Top Fifty Software Risk Factors and the Best Thirty Risk Management Techniques in Software Development Lifecycle for Successful Software Projects," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 6, pp. 11–32, 2016.

[15] L. Wallace, M. Keil, and A. Rai., "How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. Decision Sci. 35(2) 289–321," *Decis. Sci.*, vol. 35, no. 2, pp. 289–321, 2004.

[16] H. Hoodat and H. Rashidi, "Classification and analysis of risks in software engineering," *World Acad. Sci. Eng. Technol.*, vol. 56, pp. 446–452, 2009.

[17] B. Roy, R. Dasgupta, and N. Chaki, "A study on software risk management strategies and mapping with SDLC," *Adv. Intell. Syst. Comput.*, vol. 396, pp. 121–138, 2016.

[18] J. V. Menezes Júnior, "Measuring risks in software development projects," 2019.

[19] R. S. Dewi, "Reclassify and Readjust Software Risk Taxonomy in Software Development Activities Context," in *2022 5th International Conference on Information and Communications Technology (5th ICOIACT 2022)*, 2022.

Applying Software Development Risk Taxonomy in Use Case Points Complexity Factor

ORIGINALITY REPORT

12%	7%	12%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.tudelft.nl Internet Source	2%
2	"Copyright Page", 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), 2022 Publication	1%
3	publikace.k.utb.cz Internet Source	1%
4	"Table of Content", 2022 5th International Conference of Computer and Informatics Engineering (IC2IE), 2022 Publication	1%
5	Renny Sari Dewi, Yogantara Setya Dharmawan, Siti Nur Aisah. "Measuring Software Size and Effort Estimation on Islamic Banking Application", 2020 3rd International Conference on Information and Communications Technology (ICOIACT), 2020 Publication	1%

6	Ardiansyah Ardiansyah, Ridi Ferdiana, Adhistya Erna Permanasari. "Optimizing complexity weight parameter of use case points estimation using particle swarm optimization", International Journal of Advances in Intelligent Informatics, 2022 Publication	1 %
7	Janice Hill. "A Software Safety Risk Taxonomy for Use in Retrospective Safety Cases", 31st IEEE Software Engineering Workshop (SEW 2007), 03/2007 Publication	1 %
8	vufind.katalog.k.utb.cz Internet Source	1 %
9	Ika Diyah Candra Arifah, Anita Safitri, Hujjatullah Fazlurrahman, Goh Kai Chen, Asrul Nasid Masrom, Fresha Kharisma. "Smart Economy Implementation in Supporting SMEs Growth: Case Study in Indonesia & Malaysia Smart Cities", 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2022 Publication	1 %
10	bura.brunel.ac.uk Internet Source	<1 %
11	web.archive.org Internet Source	<1 %

12	www.kosmix.com Internet Source	<1 %
13	Mourad Badri, Linda Badri, William Flageol, Fadel Toure. "Source code size prediction using use case metrics: an empirical comparison with use case points", Innovations in Systems and Software Engineering, 2016 Publication	<1 %
14	Sripada Rama Sree, Chatla Prasada Rao. "Chapter 8 A Study on Application of Soft Computing Techniques for Software Effort Estimation", Springer Science and Business Media LLC, 2020 Publication	<1 %
15	"Proceedings of the Third International Afro-European Conference for Industrial Advancement — AECIA 2016", Springer Science and Business Media LLC, 2018 Publication	<1 %
16	Ali Bou Nassif, Luiz Fernando Capretz, Danny Ho. "Estimating Software Effort Based on Use Case Point Model Using Sugeno Fuzzy Inference System", 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence, 2011 Publication	<1 %

17	Marzuki Pilliang, Munawar Munawar. "Risk Management in Software Development Projects: A Systematic Literature Review", Khazanah Informatika : Jurnal Ilmu Komputer dan Informatika, 2022 Publication	<1 %
18	Yuming Zhu, Keith W. Hipel. "Life span risk management in brownfield redevelopment", 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007 Publication	<1 %
19	ijain.org Internet Source	<1 %
20	ir.lib.uwo.ca Internet Source	<1 %
21	Ali Bou Nassif, Luiz Fernando Capretz, Danny Ho. "Calibrating use case points", Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014, 2014 Publication	<1 %
22	Nassif, Ali Bou, Danny Ho, and Luiz Fernando Capretz. "Towards an early software estimation using log-linear regression and a multilayer perceptron model", Journal of Systems and Software, 2013. Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		