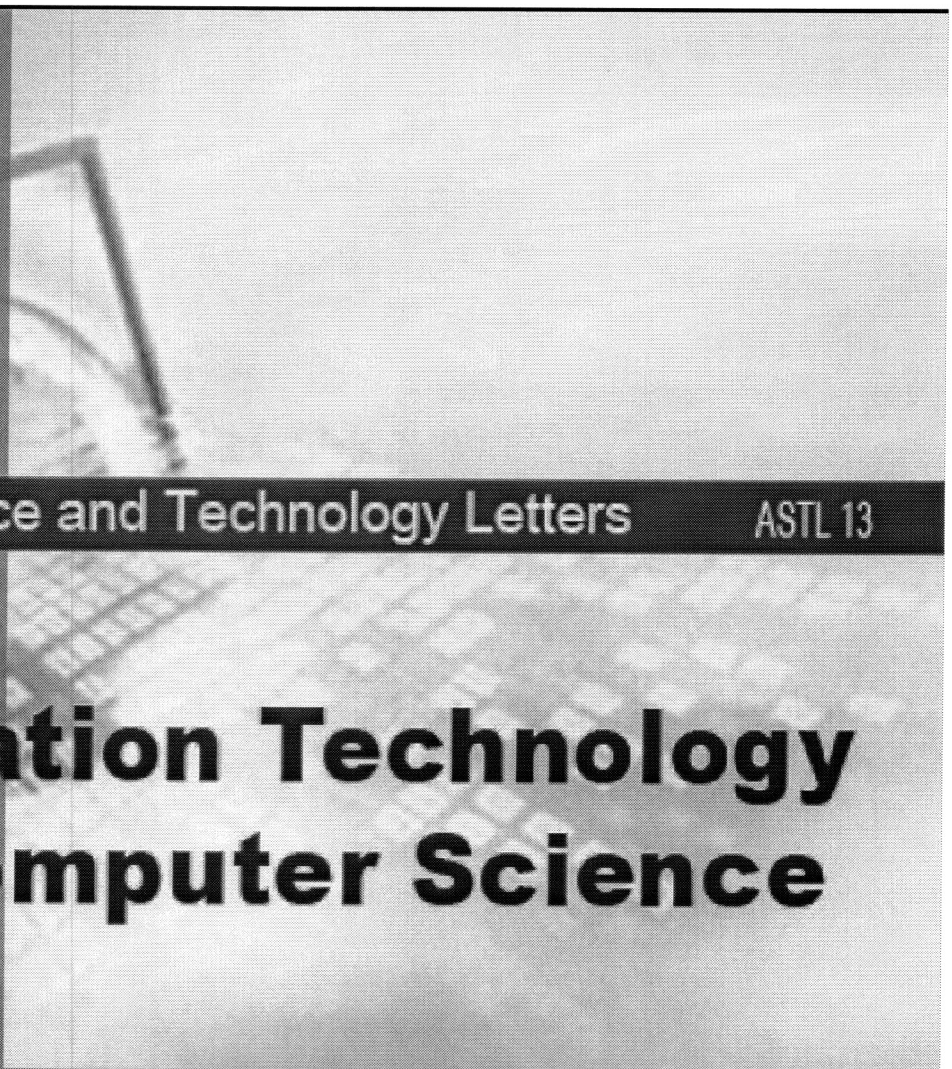
Jemal Abawajy
Dominik Slezak (Eds.)



Advanced Science and Technology Letters ASTL 13

Information Technology and Computer Science

Proceedings
International Conference, ITCS 2012
Porto, Portugal, July 2012

SERSC Science and Engineering
Research Support soCiety

Verification of Requirements Extraction and Prioritization using Use Case Points

So Young Moon[1], Bo Kyung Park[1], and Robert Young Chul Kim[1]

[1] Dept. of CIC(Computer and Information Communication), Hongik University,
Sejong Campus, 339-701, Korea
{symoon,bk,bob}@selab.hongik.ac.kr

Abstract. Calculating priority of requirements is required to make the maximum use of resources within a limited time. Previously, we proposed the calculating method of priority of all the requirements using the priority technique based on the goal-oriented Use Case method proposed by Cockburn. However, there is no verification method for this priority of the requirements. In this paper, we propose the verification method of our requirement priority technique using the Use Case point proposed by Karner.

Keywords: Use Case Point, Function Point, Goal Oriented Requirements Process, Requirements Prioritization

1 Introduction

The cost of error correction increases exponentially depending on the error detection time [1]. For example, if error is detected at the time of requirement collection, the cost of error correction would be 3, but at the time of design, the cost would be 5, at the time of coding, 9, at the time of testing, 17, and at the time of production, 160. Therefore, there would be more cost of error correction if detected at the time of latter project phase compared to the one immediately corrected after occurrence of error. As a result, requirements collection and analysis in the software development life-cycle is an important step for successful software development.

In the process of requirement definition and analysis, priority of requirements is vital since it can help to develop the high-quality product within a limited time with limited resource. That is, by determining priority of functions, we can plan a software development which provides the best value with minimum cost [2]. It is difficult to determine that which requirement has the highest priority, but using the Use Case is helpful to determine priority of requirements [3].

Previously in our research, we proposed the method of prioritizing requirements and Use Case by applying the goal-based Use Case method proposed by Cockburn [4,5,6]. Using this method, however, can determine priority but there is no method of verification. Therefore, in this paper, we verify the priority technique based on the goal-based Use Case using the Use Case Points proposed by Karner.

This paper is organized as follows: In chapter 2, related work is described such as Use Case Points proposed by Karner. In chapter 3, calculating Use Case priority using the Use Case Points is discussed. In chapter 4, the goal-based Use Case technique is verified by the Use Case Points. Finally in chapter 5, conclusion and future work are discussed.

2 Related Work

The Use Case Point has been developed by Gustav Karner based on the basic concept of Function Point [3]. The number, size and complexity of the Use Case are quantitatively measured by using actors and Use Cases in Use Case diagrams in order to measure the software size. The Use Case Point considers complexity of Use Case itself and actors which interacts with the Use Case. The Use Case Points are calculated by calculating Unadjusted Use Case Point through actors and Use Cases depicted in a diagram. Next, Use Case Points are calculated based on the Unadjusted Use Case Point using the Technical Complexity Factor and Environmental Factor. In the Technical Complexity Factor, points regarding factors which affect a system are calculated. In the Environmental Factor, factors which affect efficiency of a project development are reflected.

3 Calculating Use Case priority using the Use Case Points

We applied the actor weight, Use Case weight, weights of the technical complexity factor and environmental factor as proposed in the Use Case point calculation by Karner. In this chapter, we explain the calculating method of priority of requirements of TST Sejong multi-shop management program by using the calculation methods described in Step 1 to Step 6. Table 1 shows the results of priority of requirements of TST Sejong multi-shop management program by using the Use Case Points.

Table 1. Calculated UCP and TCF

		Unadjusted Actor Weight (UAW)			Unadjusted Use Case Weight (UUCW)					UUCP	TCF 2	TCF 3	TCF 4	TCF 7	TCF 9	TCF 11	TCF Value
											0~5						
No	Use Case	(User) Actor Weight	(Manager) Actor Weight	Actor Weight	Basic Flow	Alternative Flow	Exceptional Flow	Total Transaction	Use Case Weight		1	1	1	0.5	1	1	
UC1	Login	3	3	6	1	1	1	3	5	11	1	3	0	2	3	0	21
UC2	Custom_Register	3	3	6	1	1	0	2	5	11	1.5	0.5	0	1	3	0	7.5
UC3	Custom_Update	3	3	6	1	0	0	1	5	11	2	4	0	3	3	0	30
UC4	Custom_Retrieve	3	3	6	1	1	0	2	5	11	3	1.5	0	2	3	0	18
UC5	Custom_Delete	3	3	6	1	0	0	1	5	11	1.5	1.5	0	0.5	3	0	13.5
UC6	Stock_Register	3	3	6	1	0	1	2	5	11	2	2	0	2.5	3	0	18
UC7	Stock_Retrieve	3	3	6	1	1	0	2	5	11	3	3	0	3.5	3	0	27
UC8	Stock_Delete	3	3	6	1	0	0	1	5	11	1	1.5	0	1	3	0	12
UC9	Sale_Register	3	3	6	1	0	0	1	5	11	1	2	0	1	3	0	15
UC10	Sale_Retrieve	3	3	6	1	1	0	2	5	11	3	3	0	4	3	0	27
UC11	Sale_Update	3	3	6	1	1	0	2	5	11	3	4	2	5	3	0	33
UC12	Sale_Delete	No Use	3	3	1	1	0	2	5	8	3	4	0	4	3	4	33
UC13	Product_Register	3	3	6	1	0	0	1	5	11	1	0.5	0	1	3	0	6
UC14	Product_Retrieve	3	3	6	1	1	0	2	5	11	3	2	0	1.5	3	0	21
UC15	Product_Delete	3	3	6	1	0	0	1	5	11	2	1	0	2	3	0	12
UC16	Inventory_Retriev	3	3	6	1	1	0	2	5	11	2	2	2	2	3	0	18
UC17	Income_Retrieve	No Use	3	3	1	3	0	4	10	13	4	5	5	5	3	4	42
UC18	Expense_Create	3	3	6	1	0	1	2	5	11	2	3	0	3	3	0	24
UC19	Expense_Update	3	3	6	1	0	0	1	5	11	3	4	2	4	3	0	33
UC20	Expense_Retrieve	3	3	6	1	3	0	4	10	16	4	4	4	3	3	0	36
UC21	Expense_Delete	No Use	3	3	1	0	0	1	5	8	3	3	0	3	3	4	27
UC22	Print	3	3	6	1	1	1	3	5	11	0	1	0	1	3	0	6

- Step 1: Actor weight calculation

Actor can be categorized into two: user and supervisor. How to calculate an actor weight in the Use Case points is to calculate an actor as simple, average and complex actor. However, in this paper, an actor's weight is calculated in each Use Case for priority of requirements.

In case of the income query (UC17) which can be accessed only by a supervisor, the weight of a general user is none, and there is actor weight only for the supervisor. Since an actor and system interact with each other through the GUI, the weight of an actor on the income query is 3 which is complex. In case of the customer register (UC2), since a general user and supervisor can access and interact through the GUI, the weight for a general user is 3, and for a supervisor is 3; total weight of the actor is 6.

- Step 2: Use Case weight calculation

The Use Case weight is simple if the number of transactions is less than 3, average if the number of transactions is between 4 and 7, and complex if the number of transactions is over 8. Total sum of all calculate the Use Cases weights becomes the final Use Case weight. However, in this paper, total sum is not calculated but each Use Case weight is calculated for calculating priority of the Use Cases.

In Table 1, the Use Case weight is assigned 10 since basic flow 1, alternative flow 3, and exception flow 0 which makes the total transaction 4. Finally unadjusted Use Case point is calculated as 13 by summing the actor weight and Use Case weight calculated in step 1.

- Step 3: unadjusted Use Case point calculation

In Table 1, an unadjusted Use Case point is calculated by summing actor weight and Use Case weight by Use Cases. UC2 customer register is 11 by calculating (Actor Weight:6)+(Use Case Weight:5), and UC17 income query is 13 by (Actor Weight:3)+(Use Case Weight:10). Table 1 shows all the calculated results which sum all the weights of actor and Use Case in all 22 Use Cases.

- Step 4: Technical complexity factor calculation

To calculate the technical complexity factor in the Use Case points, the weights are given between 0 (no effect) and 5 (great effect) to each factor in terms of overall system effect. In this paper, weights are given by 0.5 unit for detailed priority of requirements.

- Step 5: Environmental factor calculation

For the environmental factor, it is calculated by applying weights between 0 to 5depending on its category such as familiarity of the life-cycle model during the project (1.5), experience on the area (0.5), experience on development methodology used (1), ability of analyzer (0.5), motivation of the team (1), stabilization of the requirements (2), part-time team member use (-1) and difficult programming language use (-1). In this paper, weight for the environmental factor is given 3 to all the Use Cases and the calculated value is 13.5. However, this value is discarded since all have the same value.

- Step 6: Priority based on the Use Case points

After completing all the calculations between Step 1 to Step 5, priority of UCP can be determined by calculating overall priority.

4 Verification of goal-oriented requirements priority using Use Case priority with Use Case point

Through customer's requirements, Use Case priority is calculated using the Use Case points. In this chapter, two methods are compared and evaluated: one result from the priority technique using previously proposed goal-oriented requirements, and the other result from the Use Case priority using the Use Case points.

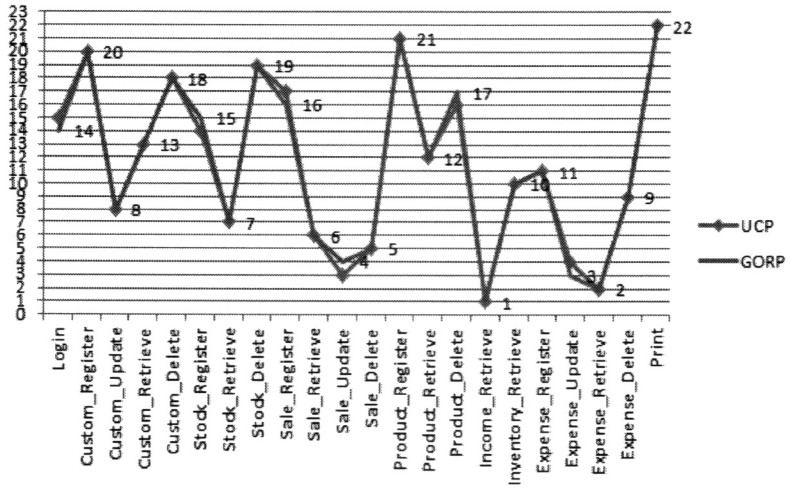


Fig. 1. Chart for UCP and GORP

In Figure 1, priority of login is 15 in UCP, 14 in GORP. Priority of Stock_Register is 14 in UCP, and 15 in GORP. Also priority of Sale_Register is calculated as 17 in UCP, 16 in GORP. Priority of Sale_Update is measured as 3 in UCP, 4 in GORP. Priority of Product_Delete is calculated as 16 in UCP, 17 in GORP, and priority of Expense_Update is measured as 4 in UCP, 3 in GORP. As shown in a chart of Figure 1, the verification of Goal oriented Use Case Requirement Priority has been done by using Use Case Points. The results are shown as almost in consistent although there is a little difference in ranking caused by the subjective difference of two methods.

5 Conclusion

In this paper, priority is calculated by applying transactions and technical complexity factor using the Use Case points. We verify the priority of requirements based on the goal-based Use Case by comparing the Use Case priority calculated by the goal-oriented requirements process method and the Use Case priority using the Use Case points. As a result, though there is a slight difference between two priorities due to the subjective judgment of the evaluator, we conclude that the results are consistent with each other.

However, when we see the result of the Use Case priority using the Use Case points, it has a difficulty to calculate the Use Case priority due to the wide range of the technical complexity factors and environmental factors. Therefore, as our future work, it will be studied that research on finding the new technical factors and environmental factors which will be applied to the Use Case priority.

Acknowledgments. This work was supported by the IT R&D Program of MKE/KEIT [10035708, "The Development of CPS(Cyber - Physical Systems) Core Technologies for High Confidential Autonomic Control Software"] and the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation.

References

1. Grady, Robert, B.:An Economic Release Decision Model: Insights into Software Project Management. In Proceedings of the Applications of Software Measurement Conference, Orange Park, FL, Software Quality Engineering, pp. 227-239 (1999)
2. Karl E. Wiegers: Software Requirements. MicrosoftPress (2003)
3. Karner, G.: Resource Estimation for Objectory Projects. Objective System SF AB(copyright owned by Rational Software) (1993)
4. Bokyung Park, Soyoung Moon, Dongho Kim, Chaeyeon Seo, R. Youngchul Kim, "A Study on Extraction of Goal Oriented Use Case Based Requirements". Proceedings of 2012 Korea Conference on Software Engineering (2012)
5. Bokyung Park, Soyoung Moon, Chaeyeon Seo, R. Youngchul Kim, "Extraction & Prioritization of Goal Oriented Use Case Based Requirements". Proceedings of KISM Spring Conference 2012, vol. 1, no. 1, pp. 62-65 (2012)
6. Bokyung Park, R. Youngchul, Kim, "Extraction & Prioritization of User Preference Requirements through User Needs". The 37th KIPS Spring Conference (2012)