

# Effort Estimation Menggunakan Metode Use Case Point untuk Pengembangan Perangkat Lunak

## (Studi Kasus Sistem Inventory Peminjaman Alat Laboratorium)

Rifki Adhitama <sup>#1</sup>, Condro Kartiko <sup>#2</sup>

# Fakultas Teknologi Industri dan Informatika, Institut Teknologi Telkom Purwokerto  
Jl. D.I Panjaitan 128 Purwokerto, Jawa Tengah, Indonesia

<sup>1</sup> rifki@ittelkom-pwt.ac.id

<sup>3</sup> condro.kartiko@ittelkom-pwt.ac.id

Accepted on September 28, 2018

### Abstract

Estimasi proyek perangkat lunak merupakan fase krusial yang dapat menentukan keberhasilan suatu proyek, dimana di dalamnya termasuk estimasi waktu pengerjaan, biaya dan sumberdaya manusia. Salah satu estimasi yang dilakukan dalam awal sebuah proyek perangkat lunak adalah *software effort estimation* atau estimasi resiko dan ukuran umum dari perangkat lunak yang akan dibuat pada sebuah proyek. Metode Use Case Points (UCP) merupakan metode *software effort estimation* yang menunjukkan performa yang lebih baik dibandingkan dengan metode lainnya. Penggunaan metode UCP dalam proyek perangkat lunak khususnya sistem inventory di indonesia dapat menjadi salah satu contoh dan acuan bagaimana estimasi perangkat lunak *inventory*. Hasil dari penelitian ini adalah resiko dari proyek sistem *inventory* peminjaman alat laboratorium memiliki resiko kecil, dapat dikerjakan dalam waktu yang relatif singkat dan tidak membutuhkan sumberdaya yang banyak.

**Keywords:** Use Case Points, Effort Estimation, inventory system

### I. INTRODUCTION

Software effort estimation merupakan salah satu aspek penting yang menentukan keberhasilan suatu proyek perangkat lunak. Banyak proyek perangkat lunak yang gagal dikarenakan manajemen proyek yang buruk [1], termasuk di dalamnya adalah perhitungan estimasi software yang buruk. Estimasi yang tepat sangat menentukan keberhasilan suatu proyek perangkat lunak, jika estimasi terlalu rendah dari seharusnya maka akan berakibat pada tidak terselesaikannya proyek perangkat lunak karena waktu maupun biaya. Tetapi jika estimasi terlalu tinggi, maka akan menyebabkan proyek perangkat lunak menjadi tidak realistis untuk dilaksanakan.

Meskipun project effort estimation memegang peran yang sangat krusial pada keberhasilan proyek, tetapi masih banyak permasalahan yang terdapat pada software effort estimation seperti kurangnya informasi pada suatu proyek perangkat lunak di awal proyek tersebut sehingga menyebabkan sulitnya melakukan estimasi dari proyek perangkat lunak tersebut. Permasalahan lain yaitu keberlanjutan proyek yang terkadang masih belum

dapat dipastikan, sehingga menuntut stakeholder untuk melakukan efisiensi sumberdaya, sehingga sumberdaya yang digunakan untuk estimasi menjadi tidak.

Estimasi proyek dapat dilakukan dengan beberapa cara, *cost model* seperti COCOMO, *Function Point Analysis* (FPA), Mark II Function Points [2] dan COSMIC [3] yang telah banyak digunakan pada industri, dimana kedua model ini menggunakan model estimasi parametrik dalam implementasinya. Meskipun telah banyak digunakan tetapi model ini masih memiliki kelemahan yaitu dibutuhkan orang yang telah memiliki pengalaman ataupun telah sangat paham dalam menghitung poin untuk tiap fungsi dari software yang di estimasi [1]. Untuk mengatasi kelemahan tersebut, dikembangkanlah sebuah metode untuk estimasi software yaitu "Use Case Point Estimation (UCP)". UCP dikembangkan oleh Gustav karner [4] untuk melakukan estimasi pada awal proyek perangkat lunak. Metode UCP merupakan sebuah estimasi awal berdasarkan dari use case diagram, dimana UCP dapat digunakan untuk melakukan memahami permasalahan yang mungkin terjadi pada perangkat lunak, estimasi ukuran proyek dan arsitektur umum sistem.

Di sisi lain, histori dan kejelasan data keluar masuk barang adalah hal yang amat penting dalam hal inventarisasi. Masih banyak skema pencatatan inventaris di berbagai tempat masih menggunakan pencatatan manual, dimana hal ini rawan terhadap accidental disasater. Pembuatan sistem inventarisasi merupakan salah satu solusi dalam mengembangkan skema inventarisasi manual ayang telah ada, memberikan keamanan lebih, dan memberikan resistensi lebih terhadap accidental disaster yang mungkin terjadi.

Tujuan dari penelitian ini adalah membuat sebuah estimasi sistem inventarisasi barang dengan menggunakan metode estimasi UCP. Studi kasus yang digunakan adalah dalam proyek pembangunan sistem *inventory* peminjaman alat laboratorium. Dimana hasil dari penelitian ini diharapkan dapat menjadi salah satu contoh maupun acuan dalam pengembangan sistem inventarisasi barang lainnya. Sehingga masyarakat yang akan mengembangkan sebuah sistem inventarisasi memiliki gambaran umum yang jelas terhadap estimasi ukuran dan arsitektur umum dari sistem inventarisasi.

## II. LITERATURE REVIEW

Use Case Point (UCP) merupakan pengembangan dari function point, dimana kelebihan dari UCP adalah pada teknik analisisnya yang berdasarkan object oriented process. UCP dikembangkan dari function point karena kelemahan dari function point yang membutuhkan terlalu banyak penilaian subjektif dari manusia [5], sehingga hasilnya menjadi kurang objektif. Pentingnya penggunaan Use Case Point (UCP) dalam pelaksanaan sebuah proyek perangkat lunak telah dibuktikan dengan pengukuran yang telah dilakukan Kusumoto S, dkk. Penelitian tersebut mengukur lima proyek perangkat lunak dengan menggunakan beberapa teknik pengukuran. Hasil dari penelitian tersebut menunjukkan bahwa teknik UCP menghasilkan hasil yang cukup memuaskan [6]. Penelitian lain yang telah dilakukan oleh Scneider dan Winters, Use Case Point (UCP) menunjukkan performa yang lebih baik dibandingkan teknik estimasi yang lainnya [7].

Penelitian lain dilakukan oleh Anda, dkk pada tahun 2002 [8], dimana penelitian tersebut meneliti tentang bagaimana metode UCP beradaptasi pada lingkungan proyek untuk meningkatkan proses dari software effort estimation pada proyek perangkat lunak. Penelitian tersebut meneliti penggunaan UCP pada beberapa proyek perangkat lunak, melakukan wawancara terhadap seluruh stakeholder dari proyek perangkat lunak yang berjalan tentang bagaimana menggunakan UCP. Hasil penelitian menunjukkan bahwa keakuratan estimasi merupakan hal yang sangat krusial dalam keberhasilan proyek perangkat lunak.

Kelebihan UCP dibandingkan dengan teknik estimasi yang lain juga pernah diteliti oleh Damodaran dan Aqua Nette. Dimana penelitian dari mereka fokus pada membandingkan function point dan UCP. Hasil dari penelitian tersebut adalah UCP lebih relevan digunakan dalam system modern yang menggunakan object oriented system yang menggunakan Use Case [9]. Penggunaan UCP pada sebuah proyek perangkat lunak dapat memproyeksikan estimasi 20% lebih dini dibandingkan dengan estimasi dari ahli dan metode estimasi lainnya [10].

### III. RESEARCH METHOD

Estimasi dalam metode UCP dimulai dengan melakukan pengukuran kompleksitas dari actor (*Unadjusted Actor Weight*) dan use case (*Unadjusted Use Case Weight*) yang ada di dalam sistem. Selanjutnya dilakukan pengukuran terhadap kompleksitas teknis (*Technical Complexity Factor*) dan environment (*Environmental Factor*) sistem yang digunakan dalam pengembangan proyek perangkat lunak. Setelah itu baru dilakukan pengukuran terhadap Use Case Points, Productivity Factor dan Effort Estimation. Detail dari alur proses UCP adalah sebagai berikut.

#### 1) Kompleksitas actor

Kompleksitas actor dalam metode pengukuran UCP dibagi kedalam tiga kategori yaitu

- *Simple*: merepresentasikan sistem yang berkomunikasi dengan sistem lainnya menggunakan *Application Programming Interface (API)*.
- *Average*: merepresentasikan sistem yang berkomunikasi dengan actor lainnya menggunakan *protocol* khusus seperti HTTP dan FTP, ataupun manusia yang berkomunikasi dengan sistem menggunakan *terminal console*.
- *Complex*: Manusia yang menggunakan *Graphical User Interface (GUI)* untuk berkomunikasi dengan sistem.

Untuk masing masing kelas yang dideskripsikan sebelumnya dilakukan perhitungan *Unadjusted Actor Weight (UAW)* atau penghitungan bobot masing-masing actor, dimana aturan pembobotan adalah tiga (3) untuk actor *complex*, dua (2) untuk actor *medium* dan satu (1) untuk actor *simple*. Untuk perhitungan rumus UAW adalah sebagai berikut

$$UAW: \sum \text{bobot } (c) \times \text{jumlah actor } (c) \quad (1)$$

dimana:

*c*: kelas dari aktor yang bersangkutan {*simple, medium, complex*}

#### 2) Kompleksitas Use Case

Langkah selanjutnya adalah mengukur kompleksitas *Use Case*. Berbeda dengan kompleksitas actor, kompleksitas use case diukur berdasarkan kompleksitas transaksi dari masing-masing use case. Transaksi di dalam use case merupakan aktivitas /skenario yang terdapat dalam masing-masing use case. Kompleksitas transaksi dari use case dibagi ke dalam tiga (3) kategori yaitu.

$$\text{kompleksitas use case} \begin{cases} \text{simple} & \text{jika transaksi} < 4 \\ \text{average} & \text{jika } \leq 4 \text{ transaksi} \leq 7 \\ \text{complex} & \text{jika transaksi} > 7 \end{cases}$$

Untuk masing-masing kelas yang dideskripsikan dilakukan perhitungan kompleksitas use case dengan menghitung *Unadjusted Use Case Weight (UUCW)*, dimana aturan pembobotan adalah lima (5) untuk *simple use case*, sepuluh (10) untuk *average use casae* dan lima belas (15) untuk *complex use case*. Untuk perhitungan rumus UUCW adalah sebagai berikut

$$UUCW: \sum \text{bobot } (c) \times \text{jumlah use case } (c) \quad (2)$$

dimana:

*c*: kelas dari use case bersangkutan {*simple, average, complex*}

#### 3) Faktor teknis dan environment pengembangan perangkat lunak

Metode UCP menggunakan dua puluh satu (21) pengukuran untuk mengukur factor teknis dan environment pengembangan sistem yang dibagi menjadi tiga belas (13) parameter untuk mengukur teknis sistem (*Technical Complexity Factors*) dan delapan (8) parameter untuk mengukur environment pengembangan sistem (*Environmental Factors*). Detail dari *Technical Complexity Factor* ditunjukkan pada Tabel I dan *Environmental Factors* ditunjukkan pada Tabel II

TABEL I  
 TECHNICAL COMPLEXITY FACTORS

Technical Complexity factors		
Factor	Description	Weight
T1	Distributed System	2
T2	Performance	1
T3	End-User Efficiency	1
T4	Complex processing	1
T5	Reusable code	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Security features	1
T12	Access for third parties	1
T13	Special training required	1

TABEL II  
 ENVIRONMENTAL COMPLEXITY FACTORS

Environmental Complexity factors		
Factor	Description	Weight
F1	Familiarity with standart process	1.5
F2	Application experience	0.5
F3	Object-oriented experience	1
F4	Lead analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Difficult programming language	-1

Pengaruh *Technical Complexity Factor (TCF)* pada metode pengukuran UCP berada di rentang antara nol (0) sampai lima (5), dimana makin besar pengaruh dari faktor yang bersangkutan dalam pengerjaan proyek, maka makin besar pula nilai estimasi pengaruhnya. Perhitungan TCF adalah mengalikan bobot penilaian pengaruh, dengan bobot yang telah ditentukan pada Tabel I. Formula untuk perhitungan UCF adalah sebagai berikut.

$$TCF: 0.6 + (0.01 * \sum_{i=1}^{13} \text{bobot faktor}_i * \text{nilai pengaruh}_i) \quad (3)$$

Sedangkan pengaruh *Environment Factor (EF)* dilakukan perhitungan yang sama dengan TCF, yaitu memberikan nilai estimasi pada setiap faktor sesuai dengan pengaruhnya pada rentang nol (0) sampai lima (5) kemudian dikalikan dengan bobot masing-masing factor sesuai Tabel II. Formula untuk perhitungan EF adalah sebagai berikut.

$$EF: 1.4 + (-0.03 * \sum_{i=1}^8 \text{bobot faktor}_i * \text{nilai pengaruh}_i) \quad (4)$$

#### 4) Use Case Points

Untuk menghitung *Use Case Point*, harus dilakukan perhitungan *Unadjusted Use Case Point (UUCP)* dengan menambahkan *Unadjusted Actor Weight (UAW)* dengan *Unadjusted Use Case Weight (UUCW)*. Formula untuk menghitung UUCP ditunjukkan pada formula (5).

$$UUCP: UAW + UUCW \quad (5)$$

Setelah didapatkan UUCW maka dilakukan perhitungan *Use Case Point* dengan formula yang ditunjukkan pada formula (6).

$$UCP: UUCP * TCF * EF \quad (2)$$

dimana:

*UUCP* : *Unadjusted Use Case Points*

*TCF* : *Technical Complexity Factors*

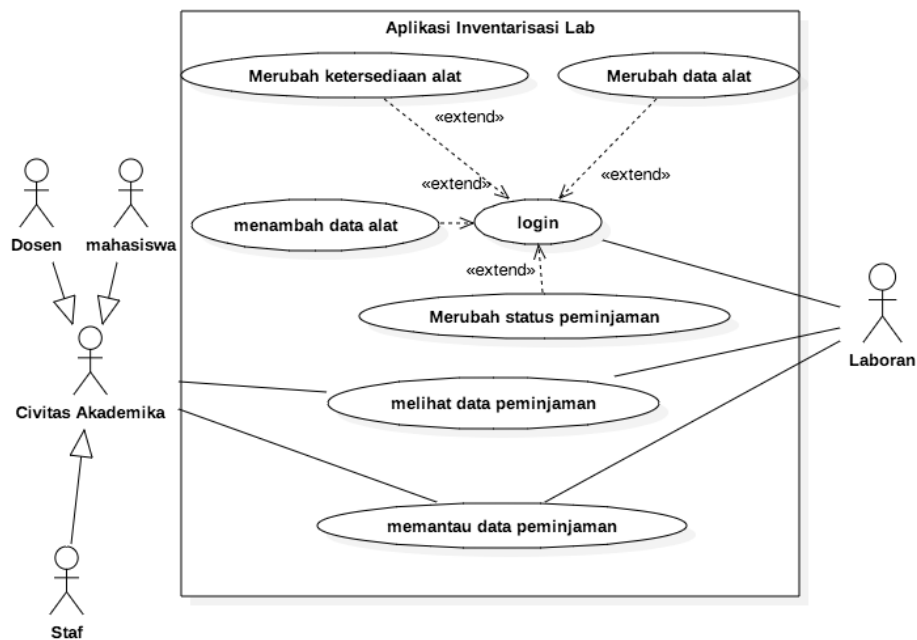
*EF* : *Environment Factor*

5) *Productivity Factor* dan *Effort Estimation*

Nilai *effort estimation* didapatkan dari perkalian antara *Productivity Factor* dengan nilai UCP. Untuk mendapatkan nilai *productivity factor* menurut perhitungan *Sneider* dan *Winters* adalah jika Nilai EF  $\leq$  dua (2) maka nilai *productivity factor* adalah 20 untuk setiap UCP, jika nilai EF berada pada rentang tiga (3) dan empat (4) maka nilai *productivity factor* adalah 28 untuk setiap UCP, jika nilai EF lebih dari 4 maka nilai *productivity factor* yang digunakan adalah 36 untuk setiap UCP [7].

#### IV. RESULTS AND DISCUSSION

Pada penelitian ini teknik UCP diimplementasikan pada sistem *inventory* peminjaman alat laboratorium komputer untuk mengetahui secara umum, effort dan resiko pada proyek perangkat lunak *inventory* peminjaman alat laboratorium. Use Case sistem *inventory* laboratorium pada studi kasus penelitian ini ditunjukkan pada Gambar I.



Gambar. 1. Use Case sistem *inventory* peminjaman alat laboratorium komputer

Dalam *use case* sistem *inventory* peminjaman alat laboratorium terdapat tujuh (7) *use case* dengan dua (2) aktor utama yaitu *laboran* dan *civitas akademika* yang terdiri dari staf, dosen dan mahasiswa. *Behavior* sistem terhadap staf, dosen dan mahasiswa adalah sama, maka dari itu ketiga sub-aktor tersebut digeneralisasi menjadi *civitas akademika*. Langkah-langkah proses estimasi UCP pada sistem *inventory* peminjaman alat laboratorium dijabarkan pada poin-poin berikut:

1) Kompleksitas *actor*

*Actor* pada sistem ini dibagi menjadi dua (2), dimana kedua *actor* dalam sistem *inventory* peminjaman alat laboratorium merupakan manusia sehingga digolongkan ke dalam *complex actor* yang memiliki bobot tiga (3). Detail perhitungan *Unadjusted Actor Weight (UAW)* pada sistem *inventory* peminjaman alat laboratorium ditunjukkan pada Tabel III.

TABEL III  
 UNADJUSTED ACTOR WEIGHT

UAW			
kelas	Jumlah actor	bobot	Jumlah actor * bobot
<i>Simple</i>	0	1	0
<i>Medium</i>	0	2	0
<i>Complex</i>	2	3	6
<b>Total UAW</b>			<b>6</b>

2) Kompleksitas *Use Case*

Sistem *inventory* peminjaman alat laboratorium pada penelitian ini memiliki tujuh (7) *use case* yaitu login, merubah data alat, merubah ketersediaan alat, menambah data alat, merubah status peminjaman, melihat data peminjaman dan memantau data peminjaman. Tujuh (7) *use case* tersebut akan dibagi ke dalam tiga (3) kelas yaitu *simple*, *average* dan *complex* berdasarkan jumlah transaksi yang ada didalamnya untuk mendapatkan *Unadjusted Use Case Weight (UUCW)*. Detail dari pembagian *use case* berdasarkan kompleksitasnya ditunjukkan pada Tabel IV dan hasil UUCW ditunjukkan pada Tabel V.

TABEL IV  
 PEMBAGIAN KOMPLEKSITAS USE CASE

Use Case	transaksi	kelas
Login	2	<i>Simple</i>
Merubah data alat	4	<i>Average</i>
Merubah ketersediaan alat	4	<i>Average</i>
Menambah data alat	5	<i>Average</i>
Merubah status peminjaman	3	<i>Average</i>
Melihat data peminjaman	2	<i>Simple</i>
Memantau data peminjaman	2	<i>Simple</i>

TABEL V  
 UNADJUSTED USE CASE WEIGHT

UUCW			
kelas	Jumlah use case	bobot	Jumlah use case * bobot
<i>Simple</i>	3	5	15
<i>Average</i>	4	10	40
<i>Complex</i>	0	15	0
<b>Total UUCW</b>			<b>55</b>

3) Faktor teknis dan *environment* pengembangan perangkat lunak

Proses selanjutnya adalah menghitung *Technical Complexity Factor (TCF)* dan *Environment Factor (EF)*. Pembobotan estimasi pengaruh *factor* dari TCF dan EF dilakukan pada rentang nol (0) sampai lima (5), dimana makin tinggi pengaruh /*impact* dari *factor* tersebut maka semakin tinggi pula bobot estimasinya. Detail dari pembobotan TCF dapat dilihat pada Tabel VI.

TABEL VI  
 TECHNICAL COMPLEXITY FACTORS

Technical Complexity factors				
Factor	Description	Bobot	impact	Bobot * impact
T1	Distributed System	2	2	4
T2	Performance	1	5	5
T3	End-User Efficiency	1	3	3
T4	Complex processing	1	1	1
T5	Reusable code	1	4	4
T6	Easy to install	0.5	4	2
T7	Easy to use	0.5	4	2
T8	Portable	2	3	6
T9	Easy to change	1	2	2
T10	Concurrent	1	2	2
T11	Security features	1	3	3
T12	Access for third parties	1	2	2
T13	Special training required	1	2	2
<b>Total Factor</b>				<b>38</b>

Dari Tabel VI, maka didapatkan nilai TCF adalah.

$$TCF: 0.6 + (0.01 * 38)$$

$$: 0.98$$

Sedangkan untuk detail pembobotan EF ditunjukkan pada Tabel VII.

TABEL VII  
 ENVIRONMENTAL COMPLEXITY FACTORS

Environmental Complexity factors				
Factor	Description	Weight	impact	Bobot * impact
F1	Familiarity with standart process	1.5	3	4.5
F2	Application experience	0.5	2	1
F3	Object-oriented experience	1	3	3
F4	Lead analyst capability	0.5	3	1.5
F5	Motivation	1	1	1
F6	Stable requirements	2	3	6
F7	Part-time workers	-1	4	-4
F8	Difficult programming language	-1	4	-4
<b>Total Factor</b>				<b>9</b>

Dari Tabel VII, maka didapatkan nilai EF adalah.

$$EF: 1.4 + (-0.03 * 9)$$

$$: 1.13$$

4) *Use Case Points*

Tahapan selanjutnya adalah menghitung *use case points* dengan mengalikan UUCW dengan TCF dan EF, hasil dari perhitungan UUCP adalah sebagai berikut.

$$UUCP: UAW + UUCW$$

$$: 6 + 55$$

$$: 61$$

Sehingga nilai UCP adalah sebagai berikut

$$UCP: UUCP * TCF * EF$$

$$: 61 * 0.98 * 1.13$$

$$: 67$$

5) *Productivity Factor dan Effort Estimation*

Berdasarkan perhitungan nilai EF maka nilai *productivity* yang digunakan adalah 20 jam untuk setiap UCP dimana resiko proyek pengerjaan tergolong rendah dan *effort estimation* dari pengerjaan adalah 1340 jam pengerjaan.

## V. CONCLUSION

Pembangunan sistem *inventory* peminjaman alat laboratorium pada studi kasus yang digunakan dalam penelitian ini membutuhkan kurang lebih 1340 jam pengerjaan, dengan resiko pengerjaan proyek yang tergolong rendah. Jika diasumsikan pengerjaan dengan tim yang berisi tiga (3) orang dengan jam pengerjaan tiga puluh (30) jam per minggu maka proyek ini seharusnya selesai dalam empat belas (14) minggu pengerjaan.

Secara umum resiko pembangunan sistem *inventory* tidak tinggi, waktu pengerjaan yang relative singkat dan tim yang dibutuhkan tidak banyak. Penelitian ini dapat dikembangkan dengan mengukur hasil actual dari pembuatan keseluruhan sistem, dan menggunakan metode UCP untuk estimasi model sistem *inventory* yang lain di Indonesia, sehingga dapat diperkirakan rerata *effort* yang diperlukan dalam pembuatan *inventory system*.

## ACKNOWLEDGMENT

Penulis berterimakasih kepada Institut Teknologi Telkom Purwokerto yang telah memberikan dukungan pada penelitian ini di bawah hibah untuk penelitian internal – Tahun 2018 (Nomor: IT Tel 1016/LPPM-000/Ka. LPM/IV/2018)

## REFERENCES

- [1] P. P. Jena and S. Mishra, "Survey Report on Software Cost Estimation using Use Case Point Method," *International Journal of Computer Science & Engineering Technology (IJCSET)*, vol. 5, no. 4, pp. 280-287, 2014.
- [2] S. C.R, *Software Sizing and Estimating: Mk II FPA (Function Point Analysis)*, New York, NY: John Wiley & Sons, Inc, 1991.
- [3] A. Abran, S. Oligny, C. Symsons and J. M. Desharnais, "Functional Size Measurements Method - COSMIC - FFP; Design and Field Trials," in *FESMA-AEMES Software Measurements Conference 2000*, 2000.
- [4] M. M. Kirmani and A. Wahid, "Use Case Point Method of Software Effort Estimation: A Review," *International Journal of Computer Applications*, vol. 116, no. 15, pp. 43-47, 2015.
- [5] K. G, "Metrics for Objectory," University of Linkoping, Sweden, 1993.
- [6] S. Kusumoto, M. Tsuda and K. Inoue, "Use Case Points Measurement Tool for Effective Project Management," in *14th Asia-Pacific Software Engineering Conference (APSEC'07)*, Aichi, Japan, 2007.



- [7] G. Schneider and J. P. Winters, *Applying Use Cases: A Practical Guide* (2nd Edition), Boston, Massachusetts: Addison-Wesley Professional, 2001.
- [8] B. Anda, E. Angelvik and K. Ribu, "Improving Estimation Practices by Applying Use Case Models," in *International Conference on Product Focused Software Process Improvement*, Rovaniemi, Finland, 2002.
- [9] M. Damodaran and A. N. E. Washington, *Estimation Using Use Case Points*, Texas, Victoria: University of Houston, 2002.
- [10] R. K. Clemmons, "Project estimation with use case points," *The Journal of Defence Software Engineering*, pp. 18-22, 2006.