

An Industrial Case Study on Requirements Volatility Measures

Annabella Loconsole, Jürgen Börstler

Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden

bella@cs.umu.se, jubo@cs.umu.se

Abstract

Requirements volatility is an important risk factor for software projects. Software measures can help in quantifying and predicting this risk. In this paper, we present an industrial case study that investigated measures of volatility for a medium size software project. The goal of the study was twofold: 1) to empirically validate a set of measures associated with the volatility of use case models (UCM); 2) to investigate the correlation between subjective and objective volatility. Measurement data was collected in retrospect for all use case models of the software project. In addition, we determined subjective volatility by interviewing stakeholders of the project. Our data analysis showed a high correlation between our measures of size of UCM and total number of changes, indicating that the measures of size of UCMs are good indicators of requirements volatility. No correlations was found between subjective and objective volatility. These results suggest that project managers at this company should measure their projects because of the risk to take wrong decisions based on their own and the developer's perceptions.

Keywords. Requirements, Volatility Measures, Empirical Validation, Case Study, Use Case Model.

1. Introduction

Requirements development is a learning, rather than a gathering, process. As a consequence, requirements change frequently, even during later stages of the development process. These changes have several impacts on the software development life cycle [44].

The concept of requirements volatility is not well defined. In the Oxford Dictionary, the term “volatile” is defined as “easily changing” [20]. We define requirements volatility as the amount of changes to a use case model over time, a definition similar to those proposed in [31, 37, 39].

Volatile requirements can cause cost and schedule overruns making the goals of the project hard to achieve. Studies show that requirements volatility has a high impact on project performance [40, 44]. Several aspects of requirements volatility have been studied, for example its impact on project or process performance [2, 14, 21, 30, 33, 39, 40, 44], assessment and prediction [1, 7, 8, 19], simulation models [14, 33], and sources and causes of volatility [17, 31, 39, 40]. Zowghi and Nurmuliani [44] perform an empirical study on requirements volatility and its impact on project performance. They measure the perceived volatility by the developers in different phases of the software development. Their results show that frequent communication between users and developers has a positive impact on the stability of the requirements. The aspect we are most interested in is measures predicting requirements volatility. Software measurement can help us in providing guidance to the requirements management activities by quantifying and predicting changes to requirements. Predicting volatility can help project managers to take appropriate actions in order to minimise project risks and to set up a more stable process.

Table 1 summarizes some measures related to requirements volatility proposed in the literature. These are measures that are applicable, if the specific requirements are well-documented. However, even in the case of well-documented requirements, measures have to be tailored towards the particular organisation because each company has its own way of documenting requirements. Among the measures in table 1, only Ambriola and Gervasi [1] describe an empirical validation of the measures, showing that the stability measure had a high predictive value of risky trends in a requirements analysis process. The reason for validating measures is to empirically demonstrate their practical utility [13, 23, 38]. A measure is empirically valid, if there is a consistent relationship between the measure and an external attribute [45]. To our knowledge, no empirical validation on requirements volatility measures has been performed in an industrial setting.

Industrial studies investigate real projects, with professional developers and real customers. In such stud-

ies, it is often difficult to get information because of confidentiality and because research is not a high priority. Therefore, it is hard to control variables.

Table 1. Measures used in requirements volatility literature

Reference	Measures
Ambriola and Gervasi [1]	Amount of information contained in requirements at a certain time.
Costello and Liu [10]	Number of changes (addition, deletions, modifications) classified by reasons for changes in a given time interval; cumulative number of changes; total number of requirements.
Harker and Eason [17]	Stable requirements; changing requirements classified in mutable, emergent, consequential, adaptive, migration.
Henderson - Sellers et al. [18]	Use case size measures; environmental factors; total number of atomic actions in all flows and alternative flow; number of atomic actions per goal and actor; number of goals per stakeholder.
Henry and Henry [19]	Number of specification change; for each specification change: average changed SLOC, average changed modules, average change SLOC per module, average SLOCs/personday.
Hammer et al. [16]	Total number of new requirements; modification to requirements; requirements traceability.
Javed et al. [21]	Pre/post functional specification changes; and post release changes.
Lam and Shankararaman [25]	Change effort; change volatility; change completeness; change error rate; requirements change density.
Malaiya and Denton [30]	Requirements changes in time; additions, deletions, and modifications to software.
Nurmuliani et al. [31]	Change types (addition deletion, modification); reason of change; origin.
Raynus [34]	Total number of system requirements; number of requirements added, modified, deleted; percentage of total requirements changes.
Rosenberg and Hyatt [36]	The percentage of requirements changed in a given time period
Stark et al. [39, 40]	Type of requirements; the planned and actual effort days for each requirement; the planned and actual number of calendar days for a version; requirements changes made to the version after plan approval (i.e. type of change, requesting group, and impact)

In this paper, we describe an industrial case study that investigated the requirements volatility of a diagnostic software system. The measures defined in the study were associated to the internal attributes size of use case model (UCM) and size of change to UCM (see table 2). The measures are a subset of those defined in [27], obtained by applying the goal question metrics (GQM) [3, 41] to the requirements management key process area of the capability maturity model [32]. The

actual subset used for this case study has been tailored towards the specific company.

The study was performed at Land Systems Häglunds (BAE Systems), Sweden, with two goals: 1) to empirically validate a set of measures associated with the external process attribute requirements volatility; 2) to investigate the correlations between the subjective volatility and the objective volatility measured through size of change to UCMs (see table 2). We collected data in retrospect on fourteen UCMs (comprising 39 use cases) for a medium size software project and interviewed the stakeholders of the project (professional developers) about requirements volatility.

In the remainder of this paper, we present the case study in section 2 and the conclusions in section 3.

2. Case study description

We followed the guidelines of Wohlin et al. [42] and Kitchenham et al. [22]. All materials of the study are available on-line to enable replication [28].

Following the GQM template [3, 41] for the definition of the goals, we obtain the following definition for our case study: *Analyse* the use case models of a diagnostic software system. *For the purpose of* 1) empirically validating requirements measures and 2) investigating the correlation between the subjective and objective measures. *With respect to* volatility of requirements. *From the point of view of* two researchers at Umeå university. *In the context of* the company Land System Häglunds.

Table 2. Internal attributes and measures of volatility

Size of UCM	Size of change to UCM
number of lines	total number of changes
number of words	number of minor changes
number of actors	number of moderate changes
number of use cases	number of major changes
	number of revisions

The first goal of the study is to validate a set of requirements volatility measures empirically. A measure is valid, if it is internally valid [13] and part of a prediction system of the kind $Y = f(X)$, where Y is the external attribute, i.e. the dependent variable and X is the internal attribute i.e. the independent variable. According to Zuse [45], we have to find correlations and prove the causality between the dependent variable and the independent variables. In our case study the dependent variable is volatility measured through size

of change to UCM, and the independent variable is size of UCM (see table 2 for the measures). However, we can only test for correlation (and not causality) due to low control compared to formal experiments.

The second goal serves to evaluate the precision of estimations of requirements volatility by project members and managers (in past projects). This is important because project managers and developers often make predictions on future projects and eventually take decisions based on their estimations.

2.1. Planning

2.1.1. Context selection. The case study was performed in retrospect at Land Systems Hägglunds (BAE Systems) Sweden. The company produces automotive systems with embedded software and has ISO9001 and ISO14001 certifications. The project chosen for the study (host project), builds external diagnostics software that runs on personal computers. At first, this software was used in another project at the company. Later on, the software was used also by external customers. Ten people have worked on the host project: two project managers, and eight developers. The software development process used was Rational Unified Process (RUP) [24]. Two iterations were performed in the elaboration phase, four in the construction phase, and one each in the inception and transition phases. The system comprises fourteen UCMs created in the inception phase (see table 3). According to the project terminology, a file containing a UCM is made up of an introduction, a revision table, a use case diagram, and a description of all actors and use cases (see [28] for a template of a UCM). As can be observed in figure 1, use case modelling was the only technique used in the host project to describe requirements (besides the vision document, which contains only sketchy requirements). The abstraction level of the UCMs was high enough to be considered as requirements and not initial designs¹. Drafts of the UCMs were used to communicate requirements to the external customer. There were also eight non-functional requirements. Since we did not have historical data available about them, we did not consider them for the study. The project started in March 2001 and was completed in August 2003. The time delay between the project and the case study was about one year. The context of the case study can

1. We are aware of the fact that some researchers consider use cases as requirements under specified circumstances [5, 15, 18, 26, 35]. Others disagree with this statement [46].

briefly be characterized according to four different dimensions: 1) *online*, because it is performed in an industrial software development environment; 2) *professional*, i.e. non-student environment; 3) *real*, because it addresses a real problem; 4) *specific*, since it is focused on particular (volatility) measures.

In order to determine the subjective volatility, we

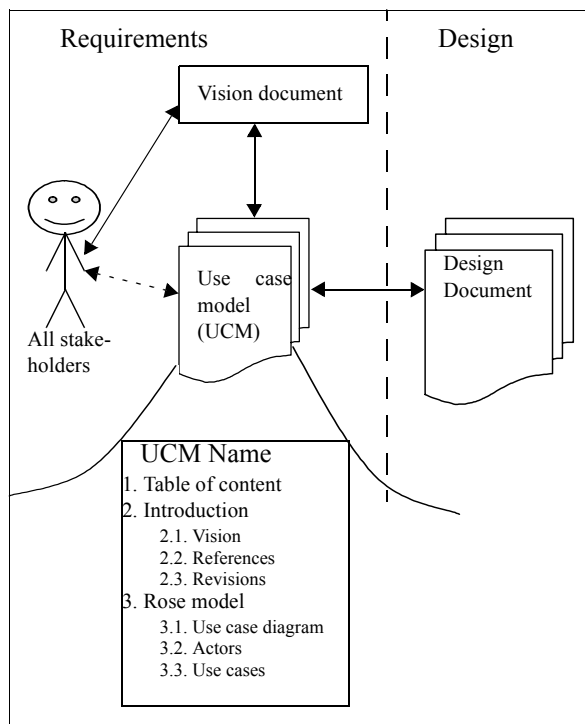


Figure 1. Requirements and design phases at the company

contacted ten stakeholders at the company. Among them were two project managers (of which one was also a developer), three developers and five internal users of the software system. The project manager in charge of the project helped us in identifying suitable subjects for our study. In the host project, the internal users were part of a “reference group” consisting of several stakeholders. The task of this group was to review the requirements specifications, and to act as the internal customer representative. All questions with impact on the requirements were managed by this group. External customers were not interviewed, since they were only familiar with initial versions of the UCMs. According to the project manager, they have never seen the complete use case models, only short descriptions of them.

The *objects* of the study were fourteen UCMs of the project at the company, comprising 39 use cases in total. Other documentation was used to gather data

Table 3. Data collected on UCMs

UC models	Number of revisions	Total number of changes	Number of minor changes	Number of moderate changes	Number of major changes	Number of actors	Number of use cases	Number of lines	Number of words	Subjective volatility
Ucm1	4	8	3	3	2	1	3	99	750	0.354
Ucm2	7	12	1	10	1	1	3	111	657	0.708
Ucm3	6	8	2	6	0	1	1	64	218	0.25
Ucm4	4	8	2	6	0	1	2	98	792	0.396
Ucm5	9	11	3	6	2	1	4	102	498	0.5
Ucm6	6	10	2	8	0	1	2	94	670	0.604
Ucm7	8	13	4	6	3	2	6	139	937	0.458
Ucm8	5	19	2	17	0	2	3	119	763	0.479
Ucm9	7	10	2	7	1	1	1	73	350	0.208
Ucm10	8	21	2	16	3	2	6	143	928	0.458
Ucm11	8	14	2	12	0	2	3	106	641	0.521
Ucm12	8	19	3	15	1	3	3	163	1068	0.271
Ucm13	4	3	2	1	0	1	1	58	216	0.25
Ucm14	6	5	2	3	0	2	1	67	283	0.292
Totals	90	161	32	116	13	5 (unique actors only)	39	1436	8771	not applicable

about the objects, in particular project plans, iteration plans, and test plans. The *guidelines* for the subjects were described in an email sent to all participants (see [28]). The *measurement instruments* used were Microsoft Excel forms and Minitab (a software package for statistical analysis).

2.1.2. Attributes and measures. The state and response variables for the first goal were size of UCM and size of change to UCM, respectively. The measures of the two internal attributes (described in table 2) are quite intuitive, except for *number of revisions*. A revision of a UCM is a version of a UCM with a unique identifier. Revisions to UCMs are done when changes are performed, but also to validate the UCMs. A revision can include several changes or no changes at all. Usually, a large amount of revisions corresponds to a large amount of changes. However, almost all UCMs have one revision (usually the last one) with no changes associated. Among the 90 revisions, we counted 11 revisions with no changes (besides the initial one for each UCMs which also have no changes associated). One possible measure of volatility could also be “number of associations between use cases”, connected to the internal attributes complexity and size of use case model. We discarded this measure because most UCMs in the host project did not have any associations.

In the second goal, the state variable was the internal attribute size of change to UCM, while the response variable was the external attribute subjective volatility.

Subjective volatility was measured by subject ratings, which were collected manually by an e-mailed form.

Although we are aware of the difficulty of determining the exact scale type [6], our measures seem to belong to the ratio scale. The rules defined for measuring the UCMs are described in [28].

2.2. Hypotheses formulation.

Our hypotheses were the following:

1. There is significant correlation between the size of change to UCM and the size of UCM.
2. There is significant correlation between the measures of size of change to UCM and size of UCM and the rating of volatility of the UCMs made by the subjects.

To test the first hypothesis we chose correlation [4]. For the second hypothesis we chose a within-subject design (i.e. all subjects filled in the same form).

2.3. Executing the study

The preparation of the subjects was made by explaining the definition of requirements volatility, describing the forms, and showing an example of how the form should be filled in [28]. The subjects were not aware of the hypotheses of the study. We handed in the material with the fourteen UCMs of the project, providing only their descriptive names. Each subject worked

alone and could use unlimited time. They rated the relative volatility defined as trends in changes to UCMs in the phases inception, elaboration, construction, and transition on a scale of three linguistic labels (low, medium, and high). Based on [11], we chose three linguistic labels because an odd number of possible outcome yields better results due to the fact that there is a single medial value. This allowed us to create a form that fitted on to a single page and was easy and quick to fill in, in order to encourage the subjects to participate in the study. We asked the subjects to not read the documentation of the UCMs in order to avoid letting the objective volatility affect the subjective volatility. This documentation contains some information (like the number of revisions, the description and the date of changes to the UCM) which may reveal the objective volatility of the UCMs.

The study was divided into two phases: the first part consisted of manual collection of data for the measures in table 2. We gathered data starting from the first available revision of the UCMs, which were dated July 2001 (three months after the beginning of the project). Data was collected by studying historical project documentation. In the second phase we distributed the forms. We contacted ten stakeholders at the company for the subjective data, and received answers from eight of them. Two of the forms were discarded because the data was considered unreliable (the same answer was given for all questions). The study did not affect the development project because it was done in retrospect. The data is described in table 3.

2.4. Analysis and interpretation

To verify the two hypotheses we first checked the distributions of data for normality. As the data distribution were not normal, we used non-parametric statistics and applied the Spearman correlation coefficient. We chose a level of significance $\alpha = 0.05$, i.e. the level of confidence is 95%. For a sample of size fourteen, the Spearman cutoff for accepting H_0 is 0.532 [43]. However, as the formality in our case study is low compared to formal experiments, we consider the Spearman cutoff only a reference point to judge the significance of our correlations.

2.4.1. Analysis of the first hypothesis. All size of change to UCM measures were correlated separately with the size of UCM measures. As we can observe in table 4, the values in bold show high correlation. The measure `total number of changes` is highly correlated with the measures of size of UCM. The `number of`

`minor changes` is not correlated at all, but this is not surprising, because we have defined minor changes as changes in the style or structure of the file. These changes do not affect the size of UCM. The other measures seems to be somewhat correlated. Concluding, there is a high correlation between the size of UCM and the `total number of changes` to it. For the moment, we do not claim any causality (i.e. that larger use cases cause a higher number of changes). Further studies are needed to check the causality and to identify guidelines for optimal size of UCMs from a volatility perspective.

2.4.2. Analysis of the second hypothesis. Since the rating of volatility made by the subjects yielded ordinal data we applied a simple transformation (weighted average) to obtain ranked data¹. For the current analysis, we averaged the answers for the phases to one value for each UCM. Each of the measures was correlated separately with the transformed rating of volatility made by the subjects (last row in table 4). Because all coefficients are below the cutoff, we can conclude that there is no significant correlation between our measures and the subjective rating of volatility. It is interesting to note that measures like `total number of changes`, which is largely utilised to measure volatility [10, 21, 30], did not show very high correlation. This may be due to the small size of the sample. Another reason can be that the subjective volatility by the stakeholders could be affected by other parameters. The stakeholders could perceive as highly volatile those UCMs affected by frequent changes very late in the development process, since the late changes are closest to the time of the case study. This could be tested by checking the transition phase separately. The subjects might recall that there was some problem with a specific UCM but in reality the problems were in other phases of the software development and did not affect the functionality agreed upon with the customer, i.e. without affecting the specific UCM.

2.5. Validity evaluation

2.5.1. Threats to conclusion validity. One issue that could affect conclusion validity is the size of the sample data (fourteen UCMs and six subjects). If the sample size is small, non-parametric tests can lead to accept the

1. We counted the number of occurrences of each answer and multiplied the number obtained by 0 (low), 0.5 (medium), or 1 (high). Finally we summed the result and divided by 6 (the max number of subjects).

Table 4. Spearman correlation coefficient obtained analysing the two hypotheses

	Total number of changes	Number of minor changes	Number of moderate changes	Number of major changes	Number of revisions
Number of lines	0.909	0.322	0.675	0.579	0.573
Number of words	0.675	0.410	0.523	0.473	0.240
Number of actors	0.632	0.270	0.497	0.129	0.427
Number of use cases	0.757	0.457	0.412	0.731	0.600
Subjective volatility	0.433	-0.255	0.383	0.123	0.267

null hypothesis [6]. Therefore, we consider our results as preliminary.

The objective data have been calculated using a computerized tool and are therefore reliable, except the categorization of changes as minor, moderate, and major respectively, because human judgement was involved. However, we have defined measurement rules to keep the judgement as objective as possible [28].

The participants of the study formed an heterogeneous group. We could have chosen a homogenous group with the disadvantages of decreasing the number of subjects and affecting external validity. However, if the group is very heterogeneous the variations due to the differences among the subjects is bigger than the variations due to the treatments, which actually is a threat in our case.

2.5.2. Threats to construct validity. The subjective volatility is based on judgement of the subjects. The subjects chosen were all involved in the host project as developers or internal users. These users also had the task of reviewing requirements. Therefore, we believe that they were capable of accurately and reliably completing the form. To measure more precisely the reliability of the subjects, we evaluated the interrater agreement by applying the Cohen's Kappa method [9]. The Kappa value ranges from 0 (no agreement) to 1 (full agreement). We obtained 0.54 which is considered a moderate strength of agreement on the four benchmarks mentioned in [12]. Details of how we obtained this value are available on-line (see [28]).

We consider construct validity of the measures in table 2 somewhat ensured, since we used the GQM [3, 41] to define the measures and we theoretically validated them [29].

2.5.3. Threats to internal validity. *Differences among subjects.* Because the group of subjects was heterogeneous, error variance due to differences among subjects is reduced. Our subjects had different background, but it

was not necessary to have previous knowledge about requirements engineering to be able to fill in the forms distributed. Therefore this threat was considered small.

Knowledge of the domain. All UCMS belonged to the same universe of discourse, and the knowledge of the domain of the project did not affect internal validity.

Accuracy of collected data. The changes to requirements were not well documented. They were determined by manually comparing several versions of files. Rules of measurements were defined. However, there is a risk that the way we defined changes to requirements can be different from the subjects' view of change.

Accuracy of responses. One factor affecting the reliability of the answers can be the time that had passed since the end of the project. However, the best time to collect reliable data and reduce the recall bias is debatable. Considering that the average length of a typical software development project at the company is about three years, one year of delay in a project context is not a long time. Furthermore, the subjects work on very few projects in parallel. Other factors affecting the subjective measures can be personal problems and mood. The developers might not have read the definition of volatility carefully and answered the form randomly. The developers might even have read the documentation related to the UCMS, affecting the perceived volatility by studying the objective data. However, this threat is small because it takes time to read the documentation of all the UCMS and we believe that the developers were not willing to spend that time. This would have been a threat in case of correlation.

Motivation of subjects. Due to the small size of the form we believed that it was not necessary to motivate the subjects. Only 75% of the answers were valid, but the subjects most heavily involved in the project did answer.

Other factors. To fill in the form required less than 30 minutes, therefore fatigue effects were not a relevant factor. Plagiarism could be checked easily, while influence among the subjects could not be controlled for and we could only trust the answers received.

2.5.4. Threats to external validity. Our threats to external validity are minimal because the study was performed in an industrial environment. Therefore, the materials used and the project were real, and the subjects were professionals. The only threat could be the relatively small size of the project.

3. Conclusions

In this paper, we have described a retrospective case study on requirements volatility performed at Land Systems Hägglunds (BAE Systems), Sweden. We collected different measures of size of UCM and size of change to UCMs, associated to the external process attribute requirements volatility for a medium size software project. We furthermore interviewed project stakeholders about their perception of requirements volatility.

Analysing the results for the **first hypothesis** (empirical validation), we found that the measure total number of changes to UCMs is highly correlated with the size of UCM. The measures of size of UCM are validated in the specific environment and can be considered good indicators of volatility. This result supports the intuitively viable notion that larger UCMs are more volatile than smaller ones and should encourage developers to modularize UCMs. This serves to re-emphasise some fundamental software engineering principles for the need of modularity and cohesion in order to manage complexity and localize change. However, it is not clear yet, whether there is a linear relationship between the size and the number of changes to UCMs. Further studies are needed to identify guidelines for optimal size of UCMs from a volatility perspective. The results are limited to one project and therefore cannot be generalised.

Analysing the results for the **second hypothesis**, we could not find significant correlations between any of our measures and the rating of volatility by the subjects. The important result is that perception did not match our measures. This is somewhat surprising, since measures of changes to requirements are suggested as reliable indicators of requirements volatility in the literature. There are many possible explanations for our results and further investigations are needed. It might for example be possible that the subjects did not reliably recall the evolution of all UCMs. Further factors might have contributed to subjective volatility, like the actual impacts of changes, priorities of use cases, or functional details in general. Even things that have nothing to do with requirements per se as for example

changing design decisions. This implies that decision makers may take a high risk when basing decisions solely on subjective requirements volatility. Therefore, we suggest that project managers at this company measure their projects in order to minimize this risk.

References

- [1] V. Ambriola, and V. Gervasi, "Process Metrics for Requirements Analysis", 7th European Workshop on Software Process Technology, Kaprun, Austria, 21-25 Feb. 2000, pp. 90-95.
- [2] F. Armour, and B. Catherwood, "A Framework for Managing Requirements Volatility Using Function Points as Currency", 12th European Software Control and Metrics Conference, London, England, Shaker Publishing, 2001, pp. 295-304.
- [3] V.R. Basili and H.D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments", *IEEE Transactions on Software Engineering*, 14(6), 1988, pp. 758-773.
- [4] V.R. Basili, R.W. Selby, and D.H. Hutchens, "Experimentation in Software Engineering", *IEEE Transactions on Software Engineering*, 12(7), July 1986, pp. 733-743.
- [5] Bray, I., *Introduction to Requirements Engineering*, Addison-Wesley, 2003.
- [6] L.C. Briand, K. El Eman, and S. Morasca, "On the Application of Measurement Theory in Software Engineering", *Empirical Software Engineering: An International Journal*, 1(1), Kluwer Academic Publishers, 1996, pp. 61-88.
- [7] D. Bush, and A. Finkelstein, "Requirements Stability Assessment Using Scenarios", 11th IEEE International Requirements Engineering Conference (RE'03), Monterey Bay, California, Usa, 8-12 Sept. 2003.
- [8] D. Bush, and A. Finkelstein, "Environmental Scenarios and Requirements Stability", Proceedings of the International Workshop on Principles of Software Evolution, Orlando, Florida, 2002, pp. 133-137.
- [9] J. Cohen, "A Coefficient of Agreement for Nominal Scales", *Educational and Psychological Measurement*, 20(1), 1960, pp. 37-46.
- [10] R.J. Costello, and D. Liu, "Metrics for Requirements Engineering," *Journal of Systems and Software*, 29(1), April 1995, pp. 39-63.
- [11] E.P. Cox, "The optimal Number of Response Alternatives for a Scale: A Review", *Journal of Marketing Research*, 17 Nov. 1980, pp. 407-422.
- [12] K. El Emam, "Benchmarking Kappa: Interrater Agreement in Software Process Assessments" *Empirical Software Engineering: An International Journal*, 4(2), 1999, pp. 113-133.
- [13] Fenton, N.E., and S.L. Pfleeger, *Software Metrics - A Rigorous & Practical Approach*, 2nd Edition, International Thomson Publishing, Boston, MA, 1996.
- [14] S. Ferreira, J. Collofello, D. Shunk, G. Mackulak, and P. Wolfe, "Utilization of Process Modelling and Simulation in Understanding the Effects of Requirements Volatility in Software

- Development”, International Workshop on Software Process Simulation and Modelling (ProSim), Portland, Oregon, Usa, 3-4 May 2003.
- [15] Futrell, R.T., D.F. Shafer, and L.I. Shafer, *Quality Software Project Management*, Prentice Hall, 2002.
- [16] T.F. Hammer, L.L. Huffman, and L.H. Rosenberg, “Doing requirements right the first time”, *CROSSTALK The Journal of Defence Software Engineering*, Dec. 1998, pp. 20-25.
- [17] S.D.P. Harker and K.D. Eason, “The Change and Evolution of Requirements as a Challenge to the Practice of Software Engineering,” Proceeding of IEEE International Symposium on Requirements Engineering, 1993.
- [18] B. Henderson-Sellers, D. Zowghi, T. Klemola, and S. Parasuram, “Sizing Use Cases: How to Create a Standard Metrical Approach”, Proceedings of the 8th International Conference on Object-Oriented Information Systems, Montpellier, France, 2-5 Sept. 2002.
- [19] J. Henry, and S. Henry, “Quantitative Assessment of the Software Maintenance Process and Requirements Volatility”, Proceedings of the 1993 ACM conference on Computer Science, Indianapolis, Indiana, USA, 1993, pp. 346-351.
- [20] Hornby A.S., *The Oxford Advanced Learner's Dictionary of Current English*, Oxford University Press, 1980.
- [21] T. Javed, M. Maqsood, and Q.S. Durrani, “A Study to Investigate the Impact of Requirements Instability on Software Defects”, *ACM SIGSOFT Software Engineering Notes*, 29(3), May 2004, pp. 1-7.
- [22] B. Kitchenham, and L. Pickard, “Evaluating Software Engineering Methods and Tools. Part 9: Quantitative Case Study Methodology.” *ACM SIGSOFT - Software Engineering Notes*, 23(1), Jan. 1998, pp. 24-26.
- [23] B. Kitchenham, S.L. Pfleeger, and N. Fenton, “Towards a Framework for Software Measurement Validation”, *IEEE Transactions on Software Engineering*, 21(12), Dec. 1995, pp. 929-943.
- [24] Kruchten, P., *The Rational Unified Process- An Introduction*, Addison Wesley, Reading, Massachusetts, 1999.
- [25] W. Lam, and V. Shankaraman, “Managing Change in Software Development Using a Process Improvement Approach”, Proceedings of the 24th Euromicro Conference, Västerås, Sweden, IEEE Computer Society, 25-27 Aug. 1998, pp. 779-786.
- [26] Larman C., *Applying UML and Patterns- An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Prentice Hall, 1998.
- [27] A. Loconsole, “Measuring the Requirements Management Key Process Area”, Proceedings of ESCOM - European Software Control and Metrics Conference, London, UK, April 2001, pp. 67-76.
- [28] A. Loconsole, “An Industrial Case Study on Requirements Volatility Measures: Appendices”, <http://www.cs.umu.se/research/software/apsec05>, Aug. 2005.
- [29] A. Loconsole, and J. Börstler, “Theoretical Validation and Case Study of Requirements Management Measures”, Umeå University, Internal Report, UMINF 03.02, July 2003.
- [30] Y.K. Malaiya, and J. Denton, “Requirements Volatility and Defect Density”, 10th IEEE International Symposium on Software Reliability Engineering, Boca Raton, Florida, 01-04 Nov. 1999, pp. 285-294.
- [31] N. Nurmuliani, D. Zowghi, and S. Fowell, “Analysis of Requirements Volatility during Software Development Life Cycle”, Proceedings of the Australian Software Engineering Conference, Melbourne, Australia, 13-16 April 2004.
- [32] M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, *Capability Maturity Model for Software, Version 1.1*, Software Engineering Institute, CMU/SEI-93-TR-24, ESC-TR-93-177, Pittsburgh, PA, USA, 1993.
- [33] D. Pfahl and K. Lebsanft, “Using Simulation to Analyse the Impact of Software Requirements Volatility on Project Performance”, *Information and Software Technology*, vol. 42, 2000, pp. 1001-1008.
- [34] Raynus J., *Software Process Improvement with CMM*, Artech House, Boston, 1999.
- [35] Robertson S., and J. Robertson, *Mastering the Requirements Process*, Addison Wesley, 1999.
- [36] L.H. Rosenberg, and L. Hyatt, “Developing an Effective Metrics Program”, European Space Agency Software Assurance Symposium, the Netherlands, March 1996.
- [37] L. Rosenberg, and L. Hyatt, “Developing a Successful Metrics Program”, The Second Annual Conference on Software Metrics, Washington, DC, June 1996.
- [38] N.F., Shneidewind, “Methodology for Validating Software Metrics”, *IEEE Transactions on Software Engineering*, 18(5), May 1992, pp. 410-422.
- [39] G. Stark., P. Oman., A. Skillicorn, and R. Ameen, “An Examination of the Effects of Requirements Changes on Software Maintenance Releases”, *Journal of Software Maintenance Research and Practice*, Vol. 11, 1999, pp. 293-309.
- [40] G. Stark., A. Skillicorn, and R. Ameen, “An Examination of the Effects of Requirements Changes on Software Releases”, *CROSSTALK The Journal of Defence Software Engineering*, Dec. 1998, pp. 11-16.
- [41] Van Solingen, R., and E. Berghout, *The Goal/Question/Metric Method - A practical Guide for Quality Improvement of Software Development*, McGraw-Hill, London, 1999.
- [42] Wohlin, C., P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in Software Engineering An Introduction*, Kluwer Academic Publishers, Boston, 2000.
- [43] Zar J.H., *Biostatistical Analysis*, Prentice-Hall, New Jersey, 1999.
- [44] D. Zowghi, and N. Nurmuliani, “A Study of the Impact of Requirements Volatility on Software Project Performance”, Proceedings of Ninth Asia Pacific Software Engineering Conference, Gold Coast, Australia, 4-6 Dec. 2002, pp. 3-11.
- [45] Zuse, H., *A Framework of Software Measurement*, Walter de Gruyter, Berlin, 1998.
- [46] Young, R.R., *The Requirements Engineering Handbook*, Artech House, 2004.