PhD in Information Technology and Electrical Engineering

Università degli Studi di Napoli Federico II

PhD Student: Vincenzo De Simone

XXX Cycle

Training and Research Activities Report – First Year

Tutor: Anna Rita Fasolino

Table of Contents

Information

I am Vincenzo De Simone and I obtained a Ms.Sc. Degree (*Laurea Magistrale*) in Computer Engineering (*Ingegneria Informatica*) at the University of Naples *Federico II* in December 2013.

I am attending the 30th cycle of the Information Technology and Electrical Engineering (ITEE) PhD Program with a **P.O.R. F.S.E.** grant. I am carrying out my research activity within the Research Group of Software Engineering under the supervision of the **Prof. Anna Rita Fasolino**.

Study and Training activities

Courses

During the first year of the PhD Program, I attended two Occasionally Provided Course, two ad-Hoc Courses and a Ms.Sc. Course. ***Table 1*** reports the information about the attended courses.

Table 1 – List of attended courses

Course type	Course Name	Provided by	Date	Number of Credits
Occasionally Provided	**Three Core Issues for the Internet: things, security and economics**	Prof. Henning Schulzrinne (Columbia University)	20/02/2015	2
	Designing and Writing scientific manuscripts	Prof. Barnet Parker (Pfeiffer University)	17/04/2015	3
ad-Hoc	**The Entrepreneurial Analysis of Engineering Research Projects**	Prof. Luca Iandoli	20/02/2015	3
	Project Management per la ricerca	Prof. Guido Capaldo	13/03/2015	3
Ms.Sc.	**Progettazione e Sviluppo di Sistemi Software**	Prof. Anna Rita Fasolino	28/07/2015	9

Seminars

During the first year, I attended the seminars reported in ***Table 2***. I also attended the Summer School on BI & Big Data Analytics.

Table 2- List of attended seminars

Title	Presenter	Date	Credits
Uml Specification of non functional Properties	Prof. Simona Bernardi	26/11/14	0,5
Site Reliability Engineering at Google	Dr. Marco Papa Manzillo	27/11/14	0,6
Verifica e Validazione di Sistemi Safety Critical	A. Della Corte, Nicola Giordano, Gabriele D'Avino	16/12/14	0,4
Applications for software development	Dr. Antonio Almazán	16/01/15	0,4

Efficient service distribution in next generation cloud	Proff. Llorca, Festa, Romano, Canonico, Avallone et al.	10/02/15	0,8
A new look at electro-magnetic induction	Prof. Giovanni Romano	19/03/15	0,5
Fundamentals of semiconductor power modules reliability	Dr. Alberto Castellazzi	24/03/15	0,4
Advanced power module thermal management and design for lifetime extension	Dr. Alberto Castellazzi	25/03/15	0,4
Power module on-board health monitoring	Dr. Alberto Castellazzi	26/03/15	0,4
Partial Possibilistic Regression Path Modeling	Prof. Rosaria Romano	20/04/15	0,4
Mathematical Modeling of Atomic Force Miscroscopes	Prof. Martin Homer	22/04/15	0,2
Agents with truly perfect recall	Dr. Nils Bulling	28/04/15	0,2
Summer School on BI and Big Data Analytics (SoData)	Proff. Picariello, Moscato, Mezzanzanica et al.	10-12/06/15	4,8

Research Activity

Support the definition and the enactment of software processes

The main objective of the proposed research activity is to define methods, techniques and tools aimed at supporting and improving the definition and the enactment of software processes.

Nowadays, software has acquired a great relevance in our society. Different domains relay in same ways on software systems. As an example, web applications, mobile application, software for embedded systems, software for safety-critical systems (power plants, railways, automotive, avionics, etc.), just to list a few, govern our everyday life. Since that, software processes that define how the software is developed, are gaining more and more relevance [1] [2].

In the last decades, Software Process research community focused on different aspects related to Software Process. Following the evolution of software systems, new approaches and development paradigms are being defined like the Agile, the model-based or the model-driven ones. Moreover, new methodologies and new enabling technologies are being proposed in order to improve their quality and productivity and to increment their efficiency and the effectiveness. In particular, the Software Process research community has been tackling different problems related to the description, the documentation, the monitoring, the automation and the enactment as well as the assessment of the activities related to the development of software systems. However, the need to produce software with strict time-to-market and in compliance with regulations and quality standards still makes the problem of supporting software processes of great interest for the software engineering community.

An effective management of software processes requires methods, techniques and tools enabling an easy definition of its static and dynamic aspects: the roles of software engineers involved in the process, the artifacts that should be produced, the flow of activities that should be carried out and the tools that should be adopted. Moreover, methods and guidelines should be defined describing the rules to be followed in the fulfillment of the development process as well as tools supporting its execution and its automation (where and when is possible). At the same time, there is the need to propose solutions apt at supporting the people involved in the execution of the software development activities. In particular, solutions aimed at supporting the execution of the process according to its definition, that are able to guide the users in the fulfillment of the different activities, to enforce the rules eventually defined and to give them an overview of the process in which they are involved should be provided. Moreover, there is the need to understand and assess the state and the behavior of the process during its execution. To this aim, methods to appraise and communicate the process and its state need to be proposed.

In this first year, I focused my attention on the aspects related to the definition and the documentation of software process. To this aim I preliminary analyzed the languages proposed in the literature to define and document software process [3], [4]. These languages allow to model various aspects of a software process in different ways adopting different formalisms. As an example languages proposed to model software processes are based on petri-net (SPADE), on programming languages (DPEL) and on UML (SPEM, UML4SPM, etc.). From the study of these languages emerged that SPEM Software & Systems Process Engineering Metamodel Specification [5] is the wider used language to define and document software

process that is accepted both from the academic and the practitioners' community based on its usability, modularity and tool support.

Moreover, I analyzed the possible solutions that can be used to support the execution of software processes like the Workflow Management systems and the Application Lifecycle Management systems [6]. ALM systems are proposed as a solution to handle different aspects of the software development process. The application's lifecycle includes the entire time during which an organization is spending money on this asset, from the initial idea to the end of the application's life. It provides features for the management of artefacts (e.g. requirements, source code, test cases) during the software product's lifecycle and offers mechanisms for automating some tasks during the software development process (e.g. build and test) [7]. Unfortunately, these systems do not start from a definition of a software process nor they provide a clear vision of the entire process they manage. As another aspect, they suffer from usability problems since the definition of the aspects of the process are usually done in a programmatic way. To overcome all these issues, I identified new research ideas to be carried out in the next years aimed at filling the gaps between the literature and the practices and allowing the definition of a system that supports the execution of a process defined through the approaches present in the literature. As an example, a model-driven approach to support the management of software process could be proposed. Starting from a definition of a software process through one of the software process modeling languages proposed by the literature it could be possible to automatically (or semi-automatically) generate and configure an ALM solution aimed at supporting its execution. Moreover, starting from an already defined ALM solution, in order to reuse it, it could be useful to derive a definition of the software process it supports through the exploitation of appositely defined reverse engineering techniques. This kind of approach could allow a clear comprehension of the software process being carried out and to easily modify it guaranteeing the alignment between its definition and the solution to support its execution.

During my first year, I was also involved in a research project (APPS4Safety) carried out by the Research Group on Software Engineering at DIETI in collaboration with Fiat Chrysler Automobile (FCA). This project is aimed at proposing new methodologies to support the development of safe vehicles. In this context, I had the opportunity to investigate the problems related to the introduction of the software processes recommended or prescribed by a quality standard in a company. In particular, the development of software of electronic control units (ECUs) should be done in accordance with the ISO 26262 standard [8] in order to guarantee its functional safety. In order to support the systematic introduction of the process prescribed by the standard, the capabilities of the company to execute what is requested by the standard should be preliminarily verified by a gap analysis. To this aim we defined a questionnaire-based and ALM-supported approach to support the execution of the gap analysis between the process carried out by the company and the one prescribed by the quality standard. The proposed approach allows to evaluate what are the elements prescribed/recommended by the standard the company is not capable to comply with and to define and evaluate the improvement actions needed to reduce these identified gaps.

Reverse Engineering of Spreadsheet Applications

Another research activity I carried out during this first year is the definition of techniques and a tool aimed at reverse engineering spreadsheet applications.

The high diffusion of software systems implies also the need to modify them in order to satisfy the changing users' requirements, to adapt them to new business models, to cope with new technology or to preserve them from quality decrease. In order to modify software, it needs to be comprehended. The software reverse engineering defines the methods and tools to derive information from existing software artifacts and to use them to carry out software engineering processes [9]. Since that the software engineering community is always looking for new methods to support the comprehension of existing software artifacts.

Spreadsheet applications [10] are widely adopted by millions of end-users from several application domains and provide strategic support to many business, scientific, industrial, and organizational processes [11] [12]. They are developed and used from so called "end-user programmers" that are users with no or poor knowledge in programming or software engineering. Besides, this kind of applications are frequently used to support different phases in the software development process especially when model-driven approaches are adopted [13] or to support important decision-making processes. Since that, a great interest has been devoted by the Software engineering research community in order to apply well known software engineering techniques and approaches in the field of spreadsheets [14] [15].

These applications often acquire a crucial role in the processes they are adopted in and their poor quality often lead to huge cost losses [16]. To avoid that, they need to undergo maintenance or evolution interventions. Several factors complicate the analysis and comprehension [17] of these applications since they are usually developed and maintained by end-users without specific software engineering skills, grow over time, are not adequately documented, and do not present explicit separation between data, business logic and user interface layers. In order to support the execution of these interventions, techniques and tools aimed at Reverse Engineering them should be defined. In particular, during this year, I defined some reverse engineering techniques aimed at supporting the comprehension of spreadsheet applications. In particular, I focused on was the definition of techniques to automatically or semi-automatically reconstruct both the data and the features offered by this kind of applications. Regarding the reconstruction of data models from a spreadsheet application I proposed the definition of a process that relies on the application of different heuristic rules defined starting from the analysis of different types of spreadsheet applications. Besides, to support the comprehension of the features offered by this kind of applications, I analyzed and modeled them. From this analysis I identified that the use of VBA to extend the features offered by these application is one of their main issues that trouble their comprehension and I proposed the introduction of a reverse engineering tool, named EXACT to address the main comprehension issues that trouble VBA-based Excel applications. This tool extends the Excel IDE offering analysis and visualization features that can be used by end-users involved in the tasks of comprehending an existing application. The EXACT tool relies on a conceptual data model representing the main components of VBA-based spreadsheet applications and their relationships obtained from the analysis and modeling phase. The tool has been validated through a qualitative case study that was conducted in a real industrial context. This study showed the usefulness of the tool in supporting several comprehension tasks carried out in real maintenance and evolution projects. The features offered by the tool aided the end-users in the execution of both top-down and bottom-up comprehension processes [18]. The experiment also showed some limitations of the proposed tool that need to be improved along with the need for additional features.

Products

Accepted Papers

P1. Domenico Amalfitano, Nicola Amatucci, *Vincenzo De Simone*, Anna Rita Fasolino, Porfirio Tramontana: **"Toward Reverse Engineering of VBA Based Excel Spreadsheet Applications"**. Proceedings of the 2nd Workshop on Software Engineering Methods in Spreadsheets co-located with the 37th International Conference on Software Engineering (ICSE 2015)

P2. Domenico Amalfitano, *Vincenzo De Simone*, Anna Rita Fasolino and Vincenzo Riccio: "**Comparing model coverage and code coverage in Model Driven Testing: an exploratory study"**. [To appear] Proceedings of the 6th International Workshop on Testing Techniques for Event Based Software (TESTBEDS) co-located with 10th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015)

P3. Domenico Amalfitano, *Vincenzo De Simone*, Anna Rita Fasolino, Porfirio Tramontana, Giancarlo Di Mare, Stefano Scala: **"A Reverse Engineering Process for Inferring Data Models from Spreadsheet-based Information Systems: An Automotive Industrial Experience".** [To appear] Communications in Computer and Information Science, Volume 178 - chapter 9, Springer

Submitted Papers

P4. Domenico Amalfitano, *Vincenzo De Simone*, Anna Rita Fasolino, Porfirio Tramontana: "**EXACT: a Tool for Comprehending VBA based Excel Spreadsheet Applications".** Submitted to the Journal of Software: Evolution and Process, Wiley

In Preparation Papers

P5. Domenico Amalfitano, *Vincenzo De Simone*, Anna Rita Fasolino: **"A questionnaire-based and ALM-supported approach for the gap analysis of quality standards"**

Conferences and Seminars

I attended the **2nd Workshop on Software Engineering Methods in Spreadsheets** at the 37th International Conference of Software Engineering (18 May 2015, Florence, Italy) where I presented the paper: "**Toward Reverse Engineering of VBA Based Excel Spreadsheets Applications**".

Credits Summary

Student: Vincenzo De Simone vincenzo.desimone2@unina.it

Tutor: Anna Rita Fasolino fasolino@unina.it

Cycle XXX

	Credits year 1								Credits year 2								Credits year 3								
		1	2	3	4	5	6			1	2	3	4	5	6			1	2	3	4	5	6		
	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Estimated	bimonth	bimonth	bimonth	bimonth	bimonth	bimonth	Summary	Total
Modules	20		5	3	3	9		20	10							0								0	20
Seminars	10	1,5	1,2	2,5	4,8			10	5							0								0	10
Research	30	8,5	3,8	4,5	2,2	1	10	30	45							0								0	30
	60	10	10	10	10	10	10	60	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60

Bibliography

[1] A. Fuggetta, «Software Process: a roadmap,» in *Proceedings of the Conference on The Future of Software Engineering*, Limerick, Ireland, 2000.

[2] A. Fuggetta e E. Di Nitto, «Software Process,» in *Future of Software Engineering, {FOSE} 2014*, Hyderabad, India, 2014.

[3] L. Garcìa-Borgonon, M. A. Barcelona, J. A. Garcìa, M. Alba e M. Escalona, «Software process modeling languages: A systematic literature review,» *Information & Software Technology,* vol. 56, n. 2, pp. 103--116, 2014.

[4] R. Bendraou, J. Jezequel, M.-P. Gervais e X. Blanc, «A Comparison of Six UML-Based Languages for Software Process Modeling,» *IEEE Transactions on Software Engineering, 36-5,* pp. 662-675, 2010.

[5] Object Management Group (OMG), «SPEM 2.0 Formal Specification,» 01 04 2008. [Online]. Available: http://www.omg.org/spec/SPEM/2.0/. [Consultato il giorno 31 10 2015].

[6] H. Lacheiner e R. Ramler, «Application Lifecycle Management as Infrastructure for Software Process Improvement and Evolution: Experience and Insights from Industry,» in *Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on*, Oulu, Finland, 2011.

[7] D. Chappell, «What is application lifecycle management?,» Microsoft Press, 2008.

[8] ISO, «ISO 26262:2011 Road vehicles - Functional safety,» ISO, 2011.

[9] G. Canfora, M. Di Penta e L. Cerulo, «Achievements and Challenges in Software Reverse Engineering,» *Commun. ACM,* vol. 54, n. 4, 2011.

[10] B. Nardi e J. Miller, «The spreadsheet interface: A basis for end user programming,» in *Human-Computer Interaction, INTERACT '90*, 1990.

[11] C. Scaffidi, M. Shaw e B. Myers, «Estimating the numbers of end users and end user programmers,» in *isual Languages and Human-Centric Computing, 2005 IEEE Symposium on*, 2005.

[12] R. Panko e D. Port, «End user computing: The dark matter (and dark energy) of corporate it,» in *System Science (HICSS), 2012 45th Hawaii International Conference on*, 2012.

[13] M. Francis, D. S. Kolovos, N. D. Matragkas e R. F. Paige, «Adding Spreadsheets to the MDE Toolkit,» in *Model-Driven Engineering Languages and Systems - 16th International Conference, MODELS 2013*, Miami, 2013.

[14] F. Hermans, M. Pinzger e A. van Deursen, «Supporting professional spreadsheet users by generating leveled dataflow,» in *Software Engineering (ICSE), 2011 33rd International Conference on,*, 2011.

[15] J. Cunha, J. Fernandez, J. Mendez e J. Saraiva, «Mdsheet: A framework for model-driven spreadsheet engineering,» in *Software Engineering (ICSE), 2012 34th International Conference on*, 2012.

[16] Eusprig, «Eusprig Horror Stories,» [Online]. Available: http://www.eusprig.org/horror-stories.htm.

[17] M. Storey, «Theories, methods and tools in program comprehension: past, present and future,» in *Program Comprehension, 13th International Workshop on*, 2005.

[18] S. Wiedenbeck e A. Engebretson, «Comprehension strategies of end-user programmers in event-driven application,» in *Visual Languages and Human Centric Computing, 2004 IEEE Symposium on*, 2004.