Sebastiano Panichella - Curriculum vitae

Biographical Sketch

Sebastiano Panichella was born in Isernia (Italy), he received (cum laude) the Laurea in Computer Science from the University of Salerno (Italy) in 2010 defending a thesis on IR-based Traceability Recovery, advised by Prof. Andrea De Lucia.

He received the PhD in Computer Science from the University of Sannio (Department of Engineering) in 2014 defending the thesis entitled *"Supporting Newcomers in Open Source Software Development Projects"*. During the PhD his work was supervised by Prof. Gerardo Canfora and Prof. Massimiliano Di Penta

Currently he is a Research Associate at University of Zurich working in the Software Evolution and Architecture Lab of Prof. Harald Gall. He is a member of IEEE. His research interests include Mining Software Repositories, Code Review, IR-based Traceability Recovery, Textual Analysis, Machine Learning (applied to SE problems), Software maintenance and evolution and Empirical Software Engineering.

Contact Information

Address: Department of Informatics
University of Zurich
Binzmuhlestrasse 14
CH-8050 Zurich, Switzerland.

Tel +41 44 63 545 78
E-mail: panichella@ifi.uzh.ch (or alternatively spanichella@gmail.com)
Office BIN 2.D.03
Home Page: www.ifi.uzh.ch/seal/people/panichella

Research Interests

Mining Software Repositories

Software repositories such as source control systems, archived communications between project personnel, and defect tracking systems are used to help manage the progress of software projects. Software practitioners and researchers are recognizing the benefits of mining this information to support the maintenance of software systems, improve software design/reuse, and empirically validate novel ideas and techniques. Research is now proceeding to uncover the ways in which mining these repositories can help to understand software development and software evolution, to support predictions about software development, and to exploit this knowledge concretely in planning future development. The Mining Software Repositories (MSR) field analyzes the rich data available in software repositories to uncover interesting and actionable information about software systems and projects.

Empirical Software Engineering

Empirical software engineering is a sub-domain of software engineering focusing on experiments on software systems (software products, processes, and resources). It is interested in devising experiments on software, in collecting data from these experiments, and in devising laws and theories from this data. Proponents of experimental software engineering advocate that the nature of software is such that we can advance the knowledge on software through experiments only. The scientific method suggests a cycle of observations, laws, and theories to advance science. Empirical software engineering applies this method to software.

Code Review

Peer code review, a manual inspection of source code by developers other than the author, is recognized as a valuable tool for reducing software defects and improving the quality of software projects. In 1976, Fagan formalized a highly structured process for code reviewing, based on line-by-line group reviews, done in extended meetings–code inspections. Over the years, researchers provided evidence on code inspection benefits, especially in terms of defect finding, but the cumbersome, time-consuming, and synchronous nature of this approach hinders its universal adoption in practice. Nowadays, many organizations are adopting more lightweight code review practices to limit the inefficiencies of inspections. In particular, there is a clear trend toward the usage of tools specifically developed to support code review. Modern code reviews are (1) informal (in contrast to Fagan-style), (2) tool-based, and (3) occurs regularly in practice nowadays, for example at companies such as Microsoft, Google, Facebook, and in other companies and OSS projects. The growth in usage of the modern code review process raises many questions. Recently, the research effort has as main focus to find approaches and tools to improve the code review process. Specifically, develop recommender systems able to (better) support developers during the code review process.

IR-based Traceability Recovery

Traceability has been defined as "the ability to describe and follow the life of an artefact (requirements, code, tests, models, reports, plans, etc.), in both a forwards and backwards direction". Thus, traceability links help software engineers to understand the relationships and dependencies among various software artefacts (requirements, code, tests, models, etc.) developed during the software lifecycle. The two main research topics related to the traceability management are event-based systems for traceability management and information retrieval based methods and tools supporting the software engineer in the traceability link recovery.

Textual analysis

Textual analysis can be described as the examination of a text in which an educated guess is formed as to the most likely interpretations that might be made of that text. It is where the researcher must decentre the text to reconstruct it, working back through the narratives mediations of form, appearance, rhetoric, and style to uncover the underlying social and historical processes, the metalanguage that guided the production. It is suggested that textual analysis can cover four main underlying constructs: language and meaning, ideology, ideology and myth, and historicity. In this sense, textual analysis is a methodology: a way of gathering and analysing information in academic research (Mckee, A 2001).

Machine Learning

Machine learning deals with the issue of how to build computer programs that improve their performance at some tasks through experience. Machine learning algorithms have proven to be of great practical value in a variety of application domains. Not surprisingly, the field of software engineering turns out to be a fertile ground where many software development and maintenance tasks could be formulated as learning problems and approached in terms of learning algorithms. Examples of the successful application of machine learning algorithms to SE problems are Bug prediction, Code (and code change) prediction, Cost estimation, Prioritization or clustering of user reviews (in the context of mobile apps), etc..

Academic Appointments

Currently he is a Research Associate at University of Zurich working in the Software Evolution and Architecture Lab of Prof. Harald Gall. He is a member of IEEE. His research interests include Mining Software Repositories, Code Review, IR-based Traceability Recovery, Textual Analysis, Machine Learning (applied to SE problems), Software maintenance and evolution and Empirical Software Engineering. He is author of 29 papers appeared in International Conferences and Journals.

Academic Experience and History

University of Sannio, Italy

PhD., Computer Engineering, June/July 2014

- Thesis Title: *"Supporting Newcomers in Open Source Software Development Projects"*
- Advisers: Prof. Gerardo Canfora and Prof. Massimiliano Di Penta
- Thesis Topics: Supporting Developers, Mining of Software Repositories (Mailing lists, Issue trackers, Versioning Systems etc.)

University of Salerno, Italy

M.S., Computer Science, December 2010

- *Magna cum Laude*
- Thesis Title: *Improving IR-based Traceability Recovery Using Smoothing Filters*
- Adviser: Prof. Andrea De Lucia
- Thesis Topics: Software Engineering, Traceability Recovery, Textual Analysis

University of Molise, Italy

B.S., Computer Science, October 2008

- *Magna cum Laude*
- Thesis Title: *Improving IR-based traceability recovery via noun-based indexing of software artifacts*
- Advisers: Prof. Giovanni Capobianco, Dr Rocco Oliveto
- Thesis Topics: Software Engineering, Traceability Management, Natural Language Processing (NLP)

Refereed Journal Publications

In papers marked with (*) the authors are listed in alphabetic order. When such a rule is not followed, authors are listed by contribution.

[1] G. Canfora, A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Defect Prediction as a Multi-Objective Optimization Problem**. *Software Testing, Verification and Reliability (STVR)* 2015. doi:10.1002/stvr.1570

[2] G. Bavota, G. Canfora, M. Di Penta, R. Oliveto, S. Panichella. ***How the Apache Community Upgrades Dependencies**. *Empirical Software Engineering (EMSE)* 2014. doi:10.1007/s10664-014-9325-9

[3] A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Applying a Smoothing Filter to Improve IR-based Traceability Recovery Processes: An Empirical Investigation**. *Information and Software Technology (INFSOF)* 2012. doi:10.1016/j.infsof.2012.08.002

[4] G. Capobianco, A. De Lucia, R. Oliveto, A. Panichella, S. Panichella. ***Improving IR-based traceability recovery via noun-based indexing of software artifacts**. *Journal of Software: Evolution and Process (JSE)* 2012. doi:10.1002/smr.1564

[5] A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Labeling Source Code with Information Retrieval Methods: An Empirical Study**. *Empirical Software Engineering (EMSE)* 2013. doi:doi:10.1007/s10664-013-9285-5

Conference Publications

In papers marked with (*) the authors are listed in alphabetic order. When such a rule is not followed, authors are listed by contribution.

[6] S. Panichella, A. Di Sorbo, E. Guzman, C. Visaggio, G. Canfora, H. Gall. **ARdoc: App Reviews Development Oriented Classifier**. In: *24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering* will be held in Seattle, WA, USA

[7] A. Di Sorbo, S. Panichella, C. V. Alexandru, J. Shimagaki, C. A. Visaggio, G. Canfora, H. Gall. **What Would Users Change in My App? Summarizing App Reviews for Recommending Software Changes**. In: *24th ACM SIGSOFT International Symposium on the Foundations of Software Engineering* will be held in Seattle, WA, USA

[8] A. Panichella, C. Alexandru, S. Panichella, A. Bacchelli, H. Gall. **A Search-based Training Algorithm for Cost-aware Defect Prediction**. *25th International Conference on Genetic Algorithms (ICGA) and the 21st Annual Genetic Programming Conference (GP) (GECCO 2016).* Denver, Colorado, USA.

[9] S. Panichella, A. Panichella, M. Bella, A. Zaidman, H. Gall. **The impact of test case summaries on bug fixing performance: An empirical investigation**. In: *Proceedings of the 38th International Conference on Software Engineering* (ICSE 2016), Austin, TX.

[10] A. Di Sorbo, S. Panichella, C. Visaggio, M. Di Penta, G. Canfora, H. Gall. . **DECA: Development Emails Content Analyzer**. In: *Proceedings of the 38th International Conference on Software Engineering* (ICSE 2016), Austin, TX.

[11] S. Panichella. **Supporting Newcomers in Software Development Projects**. In: *Proceedings of the 31st International Conference on Software Maintenance and Evolution* (ICSME 2015). Bremen, Germany.

[12] A. Di Sorbo, S. Panichella, C. Visaggio, M. Di Penta, G. Canfora, H. Gall. **Development Emails Content Analyzer: Intention Mining in Developer Discussions**. In: *30th international conference on Automated Software Engineering* (ASE 2015). Lincoln, Nebraska.

[13] S. Panichella, A. Di Sorbo, E. Guzman, C. Visaggio, G. Canfora, H. Gall. **How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution**. In: *Proceedings of the 31st International Conference on Software Maintenance and Evolution* (ICSME 2015). Bremen, Germany.

[14] G. Schermann, M. Brandtner, S. Panichella, P. Leitner, H. Gall. **Discovering Loners and Phantoms in Commit and Issue Data**. In: *Proceedings of the 37th International Conference on Program Comprehension* (ICPC 2015). Firenze, Italy.

[15] S. Panichella, V. Arnaoudova, M. Di Penta, G. Antoniol. **Would Static Analysis Tools Help Developers with Code Reviews?**. In: *Proceedings of the 22nd International Conference on Software Analysis, Evolution and Reengineering* (SANER 2015). Montreal, Canada.

[16] S. Panichella, G. Bavota, M. Di Penta, G. Canfora, G. Antoniol. **How Developers' Collaborations Identified from Different Sources Tell us About Code Changes**. In: *Proceedings of the 30th International Conference on Software Maintenance and Evolution* (ICSME 2014). Victoria, Canada.

[17] G. Bavota, S. Panichella, N. Tsantalis, M. Di Penta, R. Oliveto, G. Canfora. **Recommending Refactorings based on Team Co-Maintenance Patterns.**. In: *29th international conference on Automated Software Engineering* (ASE 2014). Vasteras, Sweden.

[18] C. Vassallo, S. Panichella, G. Canfora, M. Di Penta. **CODES: mining sourCe cOde Descriptions from developErs diScussions**. In: *Proceedings of the 36th International Conference on Program Comprehension* (ICPC 2014). Hyderabad, India.

[19] S. Panichella, G. Canfora, M. Di Penta, R. Oliveto. **How the Evolution of Emerging Collaborations Relates to Code Changes: an Empirical Study**. In: *Proceedings of the 36th International Conference on Program Comprehension* (ICPC 2014). Hyderabad, India.

[20] G. Bavota, G. Canfora, M. Di Penta, R. Oliveto, S. Panichella. ***The Evolution of Project Inter-Dependencies in a Software Ecosystem: the Case of Apache**. In: *Proceedings of the 29th International Conference on Software Maintenance* (ICSM 2013). Eindhoven, Netherlands.

[21] G. Bavota, G. Canfora, M. Di Penta, R. Oliveto, S. Panichella. ***An Empirical Investigation on Documentation Usage Patterns in Maintenance Tasks**. In: *Proceedings of the 29th International Conference on Software Maintenance* (ICSM 2013). Eindhoven, Netherlands.

[22] G. Canfora, M. Di Penta, S. Giannantonio, R. Oliveto, S. Panichella. ***YODA: Young and newcOmer Developer Assistant**. In: *Proceedings of the 35th International Conference on Software Engineering* (ICSE 2013). San Francisco, CA, USA.

[23] G. Canfora, A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Multi-Objective Cross-Project Defect Prediction**. In: *Proceedings of the 7th International Conference on Software Testing, Verification and Validation* (ICST 2013). Luxembourg.

[24] G. Canfora, M. Di Penta, R. Oliveto, S. Panichella. ***Who is going to Mentor Newcomers in Open Source Projects?**. In: *Proceedings of the 29th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (FSE 2012). Cary, North Carolina, USA.

[25] A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Using IR Methods for Labeling Source Code Artifacts: Is It Worthwhile?**. In: *Proceedings of the 20th IEEE International Conference on Program Comprehension* (ICPC), 2012. Passau, Germany.

[26] S. Panichella, J. Aponte, M. Di Penta, A. Marcus, G. Canfora. **Mining source code descriptions from developer communications**. In: *Proceedings of the 20th IEEE International Conference on Program Comprehension* (ICPC), 2012. Passau, Germany.

[27] A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Improving IR-based Traceability Recovery Using Smoothing Filters**. In: *Proceedings of the 19th IEEE International Conference on Program Comprehension* (ICPC) 2011. Kingston, ON, Canada.

[28] G. Capobianco, A. De Lucia, R. Oliveto, A. Panichella, S. Panichella. ***On the role of the nouns in IR-based traceability recovery**. In: *Proceedings of the 19th IEEE International Conference on Program Comprehension* (ICPC) 2009. Vancouver, British Columbia, Canada.

[29] G. Capobianco, A. De Lucia, R. Oliveto, A. Panichella, S. Panichella. ***Traceability Recovery Using Numerical Analysis**. In: *Proceedings of the 16th IEEE Working Conference on Reverse Engineering* (WCRE) 2009. Lille, France.

Awards

In papers marked with (*) the authors are listed in alphabetic order. When such a rule is not followed, authors are listed by contribution.

1. **Best paper award**
   A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Improving IR-based Traceability Recovery Using Smoothing Filters**. In: *Proceedings of the 19th IEEE International Conference on Program Comprehension* (ICPC) 2011. Kingston, ON, Canada.
2. **Best tool award**
   C. Vassallo, S. Panichella, G. Canfora, M. Di Penta. **CODES: mining sourCe cOde Descriptions from developErs diScussions**. In: *Proceedings of the 36th International Conference on Program Comprehension* (ICPC 2014). Hyderabad, India.

Nominated as Best Paper

In papers marked with (*) the authors are listed in alphabetic order. When such a rule is not followed, authors are listed by contribution.

1. S. Panichella, G. Bavota, M. Di Penta, G. Canfora, G. Antoniol. **How Developers' Collaborations Identified from Different Sources Tell us About Code Changes**. In: *Proceedings of the 30th International Conference on Software Maintenance and Evolution* (ICSME 2014). Victoria, Canada.
2. S. Panichella, G. Canfora, M. Di Penta, R. Oliveto. **How the Evolution of Emerging Collaborations Relates to Code Changes: an Empirical Study**. In: *Proceedings of the 36th International Conference on Program Comprehension* (ICPC 2014). Hyderabad, India.
3. G. Bavota, G. Canfora, M. Di Penta, R. Oliveto, S. Panichella. ***The Evolution of Project Inter-Dependencies in a Software Ecosystem: the Case of Apache**. In: *Proceedings of the 29th International Conference on Software Maintenance* (ICSM 2013). Eindhoven, Netherlands.
4. G. Canfora, A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Multi-Objective Cross-Project Defect Prediction**. In: *Proceedings of the 7th International Conference on Software Testing, Verification and Validation* (ICST 2013). Luxembourg.
5. A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Using IR Methods for Labeling Source Code Artifacts: Is It Worthwhile?**. In: *Proceedings of the 20th IEEE International Conference on Program Comprehension* (ICPC), 2012. Passau, Germany.

6. A. De Lucia, M. Di Penta, R. Oliveto, A. Panichella, S. Panichella. ***Improving IR-based Traceability Recovery Using Smoothing Filters**. In: *Proceedings of the 19th IEEE International Conference on Program Comprehension* (ICPC) 2011. Kingston, ON, Canada.

Professional Services and Experiences

Organising committee member of International Conferences:

- *24th IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2017)*, Klagenfurt, Austria.
- *25th International Conference on Program Comprehension (ICPC 2017)*, Buenos Aires, Argentina.
- *38th International Conference on Software Engineering (ICSE 2016)*, Austin, TX, May 14 - 22, 2016
- *13th International Conference on Mining Software Repositories (MSR 2016) - Mining Challenge*, Austin, TX, May 14 - 15, 2016
- *24th International Conference on Program Comprehension (ICPC 2016)*, Austin, TX, 2016.
- *42nd Euromicro Conference on Software Engineering and Advanced Applications (SEAA2016)*, Limasol, Cyprus, August 31 - September 2, 2016
- *41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA2015)*, Funchal, Madeira, Portugal.
- *23rd International Conference on Program Comprehension (ICPC 2015)*, Firenze, Italia.
- *22nd International Conference on Program Comprehension (ICPC 2014)*, Hyderabad, India.

Web Chair

- *21st International Conference on Program Comprehension (ICPC 2013)*, San Francisco, California, USA.

Reviewer for the following International Journals:

- *Empirical Software Engineering.*
- *Transactions on Software Engineering.*
- *Journal of Systems and Software.*
- *Information and Software Technology.*
- *Journal of Software: Evolution and Process.*
- *Science of Computer Programming.*
- *Journal of Computer Science and Technology.*

Additional reviewer of International Conferences:

- *31st IEEE/ACM International Conference on Automated Software Engineering (ASE 2016), Singapore, Singapore.*
- *30th IEEE/ACM International Conference on Automated Software Engineering (ASE 2015), Lincoln, Nebraska, USA.*
- *22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER 2015), Montreal, Canada.*

Internships

- *From 27 May 2013 to 27 July 2013 he has been a visiting researcher at the Ecole Polytechnique de Montrèal, Canada. Supervisor: Prof. Giuliano Antoniol*

External Reviewer of Grant Applications

- *External Reviewer of projects submitted in the Quebec-Flanders bilateral research cooperation program*

EU projects

- *Sebastiano Panichella partially funded with Gabriele Bavota, Gerardo Canfora, Massimiliano Di Penta, the EU FP7-ICT-2011-8 project Markos, contract no. 317743. Specifically, the MARKOS project is aimed to realize the prototype of a service and an interactive application providing an integrated view on the Open Source projects available the on web, focusing on functional, structural and licenses aspects of software code. My effort is focused on implementing relevant aspects of the Software System realized by Markos and and a generate new research results in the field of Software Engineering. Particular effort is spent on analysis of source code to study the evolution of software project to automatically extract reusable components from source code. From the other things I also extract licensing statements from the source code to monitor their evolution and avoid that changes in source code also generate the break of licenses.*

SNF projects

- *Sebastiano Panichella funded the SURF-MobileAppsData SNF project. The goal of the SURF-MobileAppsData project is mining mobile apps data available in app stores to support software engineers in better supporting maintenance and evolution activities for these apps.*

Research Meetings

- *Sebastiano Panichella was invited by the National Institute of Informatics (NII), Japan, to participate in NII Shonan Meeting entitled "Mobile App Store Analytics" (Japan).*
- *Sebastiano Panichella was invited by the Adesso company, Switzerland, to participate in "Adesso Quartalsmeeting" 24th feb 2016 (Zurich).*

Talks Given

International Summer School on Software Engineering 2011
How identify Mentors in software projects? Discussion and perspectives *July 2011.*

FSE 2012
Who is going to Mentor Newcomers in Open Source Projects?, *November 2012.*

ICPC 2012
Mining source code descriptions from developer communications, *June 2012.*

ICSE 2013
YODA: Young and newcOmer Developer Assistant, *May 2013.*

ICSM 2013
Empirical Investigation on Documentation Usage Patterns in Maintenance Tasks, *September.*

CSER 2013 - Concordia University downtown Montral (http://concordia.ca)
Supporting Developers, Mining of Software Repositories, *June.*

ICPC 2014
How the Evolution of Emerging Collaborations Relates to Code Changes: an Empirical Study, *June.*

ICPC 2014
CODES: mining sourCe cOde Descriptions from developErs diScussions, *June.*

ICMSE 2014
How Developers' Collaborations Identified from Different Sources Tell us About Code Changes, *September.*

ASE 2014
Recommending Refactorings based on Team Co-Maintenance Patterns, *September.*

SANER 2015
Would Static Analysis Tools Help Developers with Code Reviews? *March.*

ICSME 2015
How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution, *October.*

ICSME 2015
Supporting Newcomers in Software Development Projects, *October.*

ASE 2015
Development Emails Content Analyzer: Intention Mining in Developer Discussions, *November.*

EOSESE 2015
Textual Analysis or Natural Language Parsing? A Software Engineering Perspective, *December.*

"Adesso Quartalsmeeting" - 2016
Summarization Techniques for Code, Changes, and Testing, *February.*

Invited by Gran Sasso Science Institute, Center of Advanced Studies - 2016
Systematic Mining of Software Repositories, *July.*

ICSE 2016
The Impact of Test Case Summaries on Bug Fixing Performance: An Empirical Investigation, *May.*

STUDENTS SUPERVISED

Carol V. Alexandru, PhD student at University of Zurich, Switzerland, 2016.
- *A Search-based Training Algorithm for Cost-aware Defect Prediction* (GECCO 2016).
- *What Would Users Change in My App? Summarizing App Reviews for Recommending Software Changes* (FSE 2016).
- *Exploring Deep Learning Techniques for Supporting the Mining of information in Structured and Unstructured Data.*

Adelina Ciurumelea, PhD student at University of Zurich, Switzerland, 2016.
- *SURF-MobileAppsData SNF project.*

Gerald Schermann, PhD student at University of Zurich, Switzerland, 2015.
Discovering Loners and Phantoms in Commit and Issue Data (ICPC 2015).

Andrea Di Sorbo, PhD student at University of Sannio, Italy, 2016.
- *How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution* (ICSME 2015).
- *Development Emails Content Analyzer: Intention Mining in Developer Discussions* (ASE 2015).
- *DECA: Development Emails Content Analyzer* (ICSE 2016).
- *What Would Users Change in My App? Summarizing App Reviews for Recommending Software Changes* (FSE 2016).

Timofey Titov, Master student at University of Zurich, Switzerland, 2016.
Develop search-based approaches to better predict change and defect prone classes. Zurich, Switzerland. 2016.

Alessandro Rigamonti, Master student at University of Zurich, Switzerland, 2016.
Develop search-based approaches to better predict change and defect prone classes. Zurich, Switzerland. 2016.

Andreas Schaufelbhl, Bachelor student at University of Zurich, Switzerland, 2016.
Mining of mobile app data for supporting developers during software maintenance and evolution tasks. Zurich, Switzerland. 2016.

Carmine Vassallo, Master student at University of Sannio, Italy.
CODES: mining source code descriptions from developers discussions. (ICPC 2014)

Stefano Giannantonio, Bachelor student at University of Molise, Italy.
YODA: Young and newcOmer Developer Assistant. (ICSE 2013)

Skills, Competencies gained during the PhD

Statistics:
During the PhD experience, because of his works in "Empirical software engineering", he gained good experience in Statistics (the R environment was the main tool used for such purposes). He widely used several statistical tests (parametric and non) for formulating hypothesis and demonstrating the statistical significance (or superiority) of the proposed techniques.

Main Program Languages:
He currently use for his work program languages like Java (high level), Perl (base level). He is very skilled in scripting languages like R (high level), Matlab (medium level), Weka, RWeka.

Main Competencies Gained:

1) Machine Learning, Text Analysis and Natural Language Processing

He his expert in Mining of Software repositories and successfully adopted/conceived tools based on Machine Learning (ADTree, Logistic Regression etc.) methods, Natural Language Processing (Stanford NLP parser, Stanford NLP POS Tagger etc.) techniques and Text Analysis (e.g. Vector Space Model, Latent Dirichlet Allocation, Latent Semantic Indexing Jensen and Shannon Model etc.) techniques. For example, an specific example of application of such competencies is represented by the implementation of the tool ARdoc (App Reviews Development Oriented Classifier) which is a Java tool that automatically recognizes natural language fragments in user reviews that are relevant for developers to evolve their applications. Specifically, natural language fragments are extracted according to a taxonomy of app reviews categories that are relevant to software maintenance and evolution. The categories were defined in our previous paper entitled *"How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution"* and are: (i) Information Giving, (ii) Information Seeking, (iii) Feature Request and (iv) Problem Discovery. ARdoc implements an approach that merges three techniques: (1) Natural Language Processing, (2) Text Analysis and (3) Sentiment Analysis to automatically classify app reviews into the proposed categories. The purpose of ARdoc is to capture informative user reviews (requesting a new feature, description of a problem, or proposing a solution) and consequently to allow developers to better manage the information contained in user reviews.

2) Social Network Analysis

He is also expert of Social Network Analysis (SNA) and successfully used such information for profiling developers/expert in developers' SNA. Read for example the papers *How the Evolution of Emerging Collaborations Relates to Code Changes: an Empirical Study* and *Who is going to Mentor Newcomers in Open Source Projects?* and download the related tool Yoda (Young and newcOmer Developer Assistant) which is an Eclipse plugin (available in http://www.ifi.uzh.ch/seal/people/panichella/tools/YODA-tool.html) able to profile expert in developers' SNA.

3) Other technologies

Other languages that he used during his academic experience are C, C++, Perl, Scilab, Pascal, Visual basic, Prolog, Lisp, PHP, JSP and Servlet. I also have strong experience with scientific software and tools, such as Matlab, R, Weka, that are widely used to build mathematical models through machine learning techniques (including defect prediction models). Other technologies and tools that he used during the academic years include SVN/GIT and DBMS, PostgreSQL, Gerrit code review Tool.

I work currently without problem with different Operating Systems, like Windows, Mac OS, and Linux (I know very well the Ubuntu distribution).

He is also very familiar with SQL (He currently use for his research work PostgreSQL). He proficiently use GIT/SVN as versioning systems. He also wrote a series of research paper using Latex tool as main reference.

Research Tools Implemented

YODA:
Yoda (Young and newcOmer Developer Assistant) is an Eclipse plugin (available in http://www.ifi.uzh.ch/seal/people/panichella/tools/YODA-tool.html) that identifies and recommends mentors for newcomers joining a software project. Yoda mines developers' communication (e.g., mailing lists) and project versioning systems to identify mentors using an approach inspired to what ArnetMiner does when mining advisor/student relations. Then, it recommends appropriate mentors based on the specific expertise required by the newcomer.

CODES:
CODES (mining sourCe cOde Descriptions from developErs diScussions) is an Eclipse plugin (available in http://www.ifi.uzh.ch/seal/people/panichella/tools/CODES-tool.html) to automatically extract method descriptions of Java Systems from discussions in Stack-Overflow. Actually, CODES implements an approach defined in our previous work [2], that automatically extracts method descriptions from developers' communication. CODES considers as good descriptions paragraphs describing methods that obtained the higher score and allows developers to put the chosen description into the code as a Javadoc comment also becoming de facto an API description.

DECA:
DECA (Development Emails Content Analyzer) is a java tool (available in http://www.ifi.uzh.ch/seal/people/panichella/tools/DECA.html) to automatically recognize natural language fragments in emails that are relevant in the software engineering domain. Actually, DECA implements an approach which allows to recognize most informative sentences for development purposes by exploiting a set of recurrent natural language patterns that developers often use in such communication channel. DECA purpose is to capture the intent of each informative sentence (requesting a new feature,

description of a problem, or proposing a solution) and consequently to allow developers to better manage the information contained in emails.

ARdoc:
ARdoc (App Reviews Development Oriented Classifier) is a Java tool that automatically recognizes natural language fragments in user reviews that are relevant for developers to evolve their applications. Specifically, natural language fragments are extracted according to a taxonomy of app reviews categories that are relevant to software maintenance and evolution. The categories were defined in our previous paper entitled *"How Can I Improve My App? Classifying User Reviews for Software Maintenance and Evolution"* and are: (i) Information Giving, (ii) Information Seeking, (iii) Feature Request and (iv) Problem Discovery. ARdoc implements an approach that merges three techniques: (1) Natural Language Processing, (2) Text Analysis and (3) Sentiment Analysis to automatically classify app reviews into the proposed categories. The purpose of ARdoc is to capture informative user reviews (requesting a new feature, description of a problem, or proposing a solution) and consequently to allow developers to better manage the information contained in user reviews.

Languages

Sebastiano Panichella currently speak three languages: Italian (mather tongue), English (*B2*) and German (*A1.2*).

Teaching Experience

University of Zurich, Switzerland

Lecturer **March-June 2016**

- Lecturer for the *Software Maintenance and Evolution* course.

Lecturer **March-June 2015**

- Lecturer for the *Software Maintenance and Evolution* course.
- Co-Lecturer (with Prof. Harald Gall and Dr. Philipp Leitner) for the *Advanced Software Engineering* course.

University of Sannio, Italy

Lab Instructor **December 2013**

- Lab instructor for the Programming Techniques course of Professor Gerardo Canfora:
  - The Languages and Grammars
  - JavaCC parser

University of Sannio, Italy

Teaching Assistant

- Teaching Assistant for the Software Engineering course of Prof. Massimiliano Di Penta:
  - Recovering Traceability Links via Information Retrieval Methods
- Seminaries at the *Ecole Polytechnique de Montreal*:
  - Who is going to Mentor Newcomers in Open Source Projects?

- Mining Source Code Descriptions from Developers Communications

University of Molise, Italy

Seminary

- Seminary in the Software Engineering course of Dott. Rocco Oliveto:
  - Improving IR-based Traceability Recovery Using Smoothing Filters.

Professional Memberships

- IEEE Membership (2011–present)

References

Harald C. Gall, Ph.D (e-mail: gall@ifi.uzh.ch)
- Full Professor, University of Zurich
- ◇ Binzmuhlestrasse 14, CH-8050 Zurich (Switzerland).
- ⋆ *Dr. Gall, co-author (and advisor) of some publications at the University of Zurich.*

Giuliano Antoniol, Ph.D (e-mail: antoniol@ieee.org)
- Professor, Ecole Polytechnique de Montreal
- ⋆ *Dr. Antoniol, co-author of some publications.*

Andrian Marcus, Ph.D (e-mail: amarcus@utdallas.edu; phone: (972-883-4246)
- Associate Professor, University of Texas at Dallas
- ◇ 800 W. Campbell Road; MS EC31 Richardson, TX 75080 U.S.A.
- ⋆ *Dr. Marcus, member of dissertation committee and co-author of some publications.*

Andy Zaidman, Ph.D (e-mail: A.E.Zaidman@tudelft.nl)
- Professor, University of Delft
- ⋆ *Dr. Zaidman, co-author of some publications.*

(Advisor During the PhD) Massimiliano Di Penta, Ph.D (e-mail: dipenta@unisannio.it)
- Associate Professor, University of Sannio
- ◇ Palazzo ex Poste, Via Traiano, I-82100 Benevento (Italy).
- ⋆ *Dr. Di Penta, PhD adviser.*

(Advisor During the PhD) Gerardo Canfora, Ph.D (e-mail: canfora@unisannio.it)
- Full Professor, University of Sannio
- ◇ Palazzo ex Poste, Via Traiano, I-82100 Benevento (Italy).
- ⋆ *Dr. Canfora, PhD adviser.*

Andrea De Lucia, Ph.D (e-mail: adelucia@unisa.it; phone: +39 089963376)
- Professor, University of Salerno
- ◇ Via Giovanni Paolo II, 132 - 84084 Fisciano (SA), Italy.
- ⋆ *Prof. De Lucia, co-author of some publications.*

Rocco Oliveto, Ph.D (e-mail: rocco.oliveto@unimol.it; phone: +39 0874 404123)
- Assistant Professor, University of Molise
- ◇ C.da Fonte Lappone - 86090 Pesche (IS), Italy.
- ⋆ *Dr. Oliveto, co-author of some publications.*