SEGAL RESEARCH PROGRAMME

Our research builds on three central elements:

1. The ***study of communication*** is pervasive in most of my research endeavors, whether we studied requirements engineering processes and practice, stakeholders' negotiation or discovery of requirements, or socio-technical coordination in large software projects. Communication emerged as an essential ingredient of software engineering practice – but one extremely difficult to achieve effectively, especially in distributed software development.

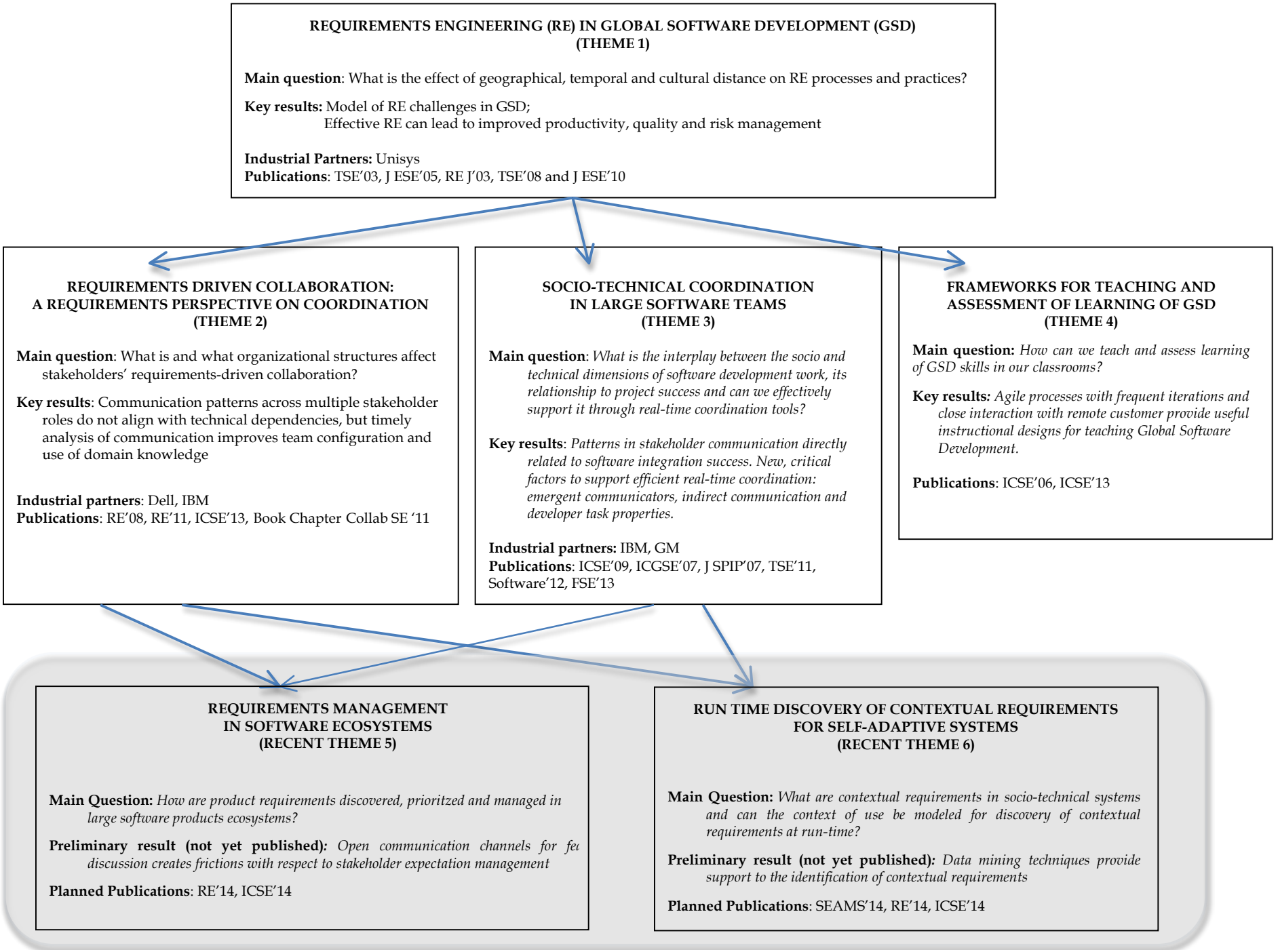
2. Work on ***industrially-relevant problems***: We try to keep my research very close to the practice of requirements engineering and software engineering in industry, particularly in commercial software development organizations. We aim to deliver research that addresses industrially-informed needs, and to report it in a way that facilitates its application in industry.

3. *Empirically-validated research*: Whether we pursue theory or tool development, we choose research methodologies that deliver empirical evidence for well-founded conclusions, and we seek industrial validation whenever possible.

The following pages describe six themes driving my research programme. For each theme, we describe our guiding research questions, as well as how these questions or themes evolved through emerging research results. The last two themes (5 and 6) represent very recent research developments and part of our vision for future work. Diagram 1 summarizes this information and intends to show the relationships, conceptual and temporal, among these themes. For each of the six themes, we make references to our research contributions and selected publications (numbers as in the Publication List at the end of the document), and we list research or industrial collaborators, and graduate students without whom this research would not have been possible.

SEGAL (Software Engineering Global interAction Lab, http://thesegalgroup.org) provides a physical and intellectual space for my research team. The lab was equipped with state-of-the-art collaboration technologies (e.g. large interactive displays and videoconferencing systems), through a large Canada Foundation for Innovation (CFI) Grant ($560,000) in 2006. SEGAL serves as a highly collaborative and creative working space for my UVic students and our international collaborators. Not only do our students collaborate with several international graduate student visitors to our lab yearly, but also get first-hand experience with the collaboration technologies we study through using them in our knowledge transfer activities with industrial partners and adjoining labs at other universities such as Aalto University (Finland), Open University (UK), Univ. of Bari (Italy), PUCRS (Brazil) and Utrecht Univ. (Netherlands). As a result, our research has yielded higher quality work, richer research perspectives, increased student publications and recognized international publications and recognized international collaborations.

DIAGRAM 1. THEMES, RESEARCH QUESTIONS AND KEY RESULTS IN SEGAL RESEARCH



Theme 1: Requirements Engineering in Global Software Development (2002-2008)

Global software development (GSD) is now an established mode of creating software, but as recently as the late 1990s it was just emerging in practice. Our contributions to this field began in early 2000 with our field study of requirements engineering (RE) in two international, multi-site organizations. One of the first investigations of RE in global software development, this work became highly cited. The questions we asked included:

1. *What is the effect of geographical, temporal and cultural distance on RE activities and stakeholder interaction?*
2. *Does distributed RE require different processes to address these challenges -- and what should these processes look like?*

This work brought significant contribution to both research and practice of requirements engineering in GSD. By answering RQ1 we described specific RE challenges and a theoretical model of the impact of distance on requirements management activities, as well as outlining recommendations for improved RE practice in global teams. Our RE'02 [18] paper was invited for publication in RE Journal [1]; the two have a combined Google Scholar citation figure of over 350 in Jan 2014.

A three-year (2002-2005) longitudinal field study of RE process improvement at the same organization (Unisys) addressed RQ2 and enabled us to describe both (i) the implementation of RE practices to account for challenges associated with distributed RE, and (ii) a methodology for the systematic analysis of the payoffs that improved RE processes bring to distributed software projects. Our research results, published in the Transactions on Software Engineering in 2003 [5] and the Journal of Empirical Software Engineering [2], represent some of the first quantitative evidence to support the hypothesis that effective RE can lead to improved productivity, quality and risk management. The methodology in our study is described in a recent book on Case Study Research in Software Engineering[1] , and is included in the discussion of design theories and issues of generalizability from case study research by Wieringa and colleagues[2]. Empirical studies by other researchers that draw on our methodology include studies of the relationship between requirements quality and project success, and communication flow in business process redesign, as well as redesign success.

Acknowledging that collaborative tools, besides software processes and practices, are an integral part of requirements engineering activities in global teams, in this line of further research we also explored:

3. *What is the appropriate computer-mediated collaborative environment to support requirements engineering elicitation and negotiations in global teams?*

In collaboration with researchers from University of Bari, Italy, we developed and evaluated empirically a number of prototypes for requirements elicitations and negotiations in distributed teams [7,11]. Drawing on theories of computer-mediated communication, common ground and media richness, we developed a research framework that addressed aspects specific to requirements elicitations and negotiations and which informed the analysis, design and empirical evaluation of our tools.

Insights from these first studies shaped our subsequent questions and approaches in significant ways:

a. In market-driven software development, RE involves a large pool of stakeholders playing different roles, and having diverse backgrounds, expectations, needs and communication challenges. Their collaboration is key to project success – especially in distributed projects [5,6] – raising the question: *How do these various stakeholders collaborate for requirements discovery, negotiation and management?* (pursued in Theme 2),
b. If small development teams have significant challenges coordinating requirements work: *How do large software teams achieve effective socio-technical coordination in global teams?* (pursued in Theme 3)
c. If GSD skills are becoming essential to our graduates, *How can we teach our CSC students the skills of Global Software Engineering and effective Requirements Engineering in distributed teams?* (pursued in Theme 4)

Doctoral students: D. Mallardo and F. Calefato (University of Bari, Italy), J. Chisan (MSc, UVic)
Industrial partners: Unisys

[1] P. Runeson, M. Host, A. Rainer, B. Regnell: Case study research in Software Engineering: Guidelines and Examples, Wiley, 2012.

[2] R. Wieringa, M. Daneva, N. Condori-Fernandez, The Structu*re of Design Theories, and an Analysis of Their Use in Software Engineering Experiments, Int'l Symp. on Emp. Soft. Eng. and Measurement (ESEM), 2011

Theme 2: Requirements-driven collaboration: a requirements perspective on coordination (2007-2012)

Software development draws upon the knowledge, expertise and contribution of diverse stakeholders (e.g., business analysts, designers, developers, testers). We investigated the collaboration within teams that involved multiple stakeholder roles and whose work was shaped by project requirements (which we termed *requirements-driven collaboration*). We found that communication patterns across multiple stakeholder roles do not align with technical dependencies, but timely analysis of communication improves team configuration and use of domain knowledge.

Initially, we addressed broad research questions about this collaboration:

4. *How can the collaboration of many stakeholder roles be studied?*
5. *What are the communication patterns in requirements-driven collaboration?*

In a chapter [15] of the *Collaborative Software Engineering Book*[1] we introduced the concept of requirements-driven collaboration and a methodology to study the collaboration of a diverse set of stakeholders (RQ4). We then applied our methodology in a series of field studies of small but distributed software teams at Dell (RQ5). Our publications at RE'08 [24] and RE'11 [30] document patterns of communication and information flow in cross-functional teams whose coordination was driven by project requirements. The type of information brokers that could emerge in teams working on interdependent requirements, or the observation that requirements changes are most frequent types of communication in requirements-driven collaboration are examples of empirical findings with implications for both theories and tools for effective knowledge management in multi-stakeholder teams.

Our findings also suggested a complex interplay between stakeholder communication and other project factors (e.g., organizational structures, development processes), and which affects stakeholders' collaboration and ultimately project success. We thus pursued emergent research questions:

6. *How do organizational structures (communication structures, distribution of domain knowledge) and development processes affect the collaboration within multi-stakeholder teams?*
7. *Can we leverage information from stakeholders' communication to improve project success?*

We studied the interplay between communication structures and the distribution of domain knowledge, and we considered the relationship of these to how different stakeholder roles coordinate their technical dependencies in software projects (RQ6) [34]. Our study contributes to an understanding of the interplay of organisational structures and work dependencies when conceptualized at the requirements-level in upstream project phases, much earlier than code generation. The results have implications (i) for team configuration and effective knowledge management in outsourcing projects, and (ii) for further research that considers requirements-driven dependencies in socio-technical coordination.

To address RQ7, we used data mining tools and machine learning in the automated analysis of large online communication repositories in projects such as IBM's Rational Team Concert (RTC)[2]. In an RE'12 publication [32] we described our approach to identifying communication patterns in stakeholder collaboration based on a large set of user stories. Using a data-driven approach we also identified a catalogue of six communication patterns derived from this dataset. Our approach is intended to support project managers to identify, through timely analysis of communication patterns, requirements for which development has become problematic. We validated the patterns we identified and their relationship to project success with RTC developers and managers. A recent submission under review at TSE [13] describes our approach together with its supporting machine learning infrastructure, the communication pattern catalogue we developed, and empirical evidence of how it supports enhanced risk management activities in the current RTC development processes.

Doctoral students: Sabrina Marczak, Irwin Kwan
Post-doctoral fellows: Dr. Eric Knauss
Industrial partners: Dell, IBM

[1] A. Finkelstein, J. Grundy, A. van der Hoek, I. Mistrík, and J. Whitehead (Eds.), *Collaborative Software Engineering*, 2010

[2] https://jazz.net/

Theme 3: Socio-technical coordination in large software teams (2006-ongoing)

Our initial studies of RE practice at Unisys identified significant challenges in the collaboration of geographically distributed stakeholders. Still, the actual developer teams were small and mostly physically co-located in Australia and New Zealand. In this line of research we investigated the coordination of *large, distributed* developer teams. We studied the complex interplay of the social and technical aspects of their work, identified new patterns of communication that should be considered in measures of socio-technical coordination, and developed and empirically validated a number of tools to support efficient coordination.

Around 2004 collaboration tools were being developed purposefully to support large, distributed software teams (most notably IBM's Rational Team Concert (RTC) development environment – originally referred to as Jazz), and we established what has since become a long-standing research collaboration with the RTC team in Ottawa, Canada. We conducted a series of empirical studies of socio-technical coordination in the RTC project itself (the project team developing the RTC uses the tool for its own development) and started with *broad research questions*:

8. *Does distance still matter with the use of more advanced collaboration tools?*
9. *What role do social interactions play in the success of large software projects (i.e. integrations)?*
10. *What is the interplay between the socio and technical dimensions of software development work and what is its impact on project success?*

we explored whether the advanced collaborative tools help overcome challenges of distance[1]. We explored the nature of socio-technical relationships in the RTC project, and the coordination patterns in resolving issues and handling large software integrations across multiple geographically distributed teams. We documented industrial best practices and in particular the role that particular social structures played in dealing with the effects of distance [8,9,10,25,26]. These results have direct practical implications for the management of global teams and the design of supporting collaboration tools. Our field studies (e.g., [22]) provided detailed accounts of coordination breakdowns due to poor management of technical dependencies and their changes in large software teams, and have informed the design of change awareness tools (e.g. Hattori and Lanza, 2010[2]), or recommendation approaches (Holmes and Walker, 2010[3]).

We also used data mining techniques to study the role played by communication alone in software integrations [26] and found that, with no other information on developers' work dependencies, developer communication emerged as a strong predictor for the integration outcome. This was an important result, given that any other factors related to integration failures studied before were of a technical nature. This result inspired our subsequent investigation of the socio-technical alignment in relation to integration outcome (RQ10) in the RTC project. We extended existing measures of socio-technical congruence and tested the relationship between socio-technical gaps and integrations at IBM [10]. Our results suggest that current socio-technical congruence measures should also consider indirect communication and emergent communicators (unaccounted for in previous measures) in large software integrations. Our invited paper in *IEEE Software's* Voice of Evidence column [12] describes what we currently know about socio-technical congruence in software projects.

Having acquired this knowledge of large-scale coordinations, our next goal (ongoing) has been to implement actionable knowledge for coordination, i.e., developing real-time support for efficient coordination in large teams:

11. *How can real-time socio-technical coordination be supported effectively?*

In collaboration with our industrial partners, we developed and evaluated a number of prototypes for coordination recommenders [38,39,40,45,47]. These prototypes leverage automated analysis of the developers' work practices, dependencies in the code and current technical tasks, as well as historical information on project outcomes, in order to make recommendations for efficient coordination on current tasks. Our recent publications at FSE [35,48] reports on our latest technique developments and also identifies that the properties of tasks that developers work on are as important in informing coordination needs as their work practices.

Doctoral students: I. Kwan, A. Schroeter, K. Blincoe, E. Kalliamvakou, A.Borici, J. Ell and L. Panjer
Post-doctoral fellows: Dr. Timo Wolf
Industrial partners: IBM, GM

---

1. Herbsleb, J. Global Software Engineering: The future of socio-technical coordination, IEEE Future of Software Engineering (FOSE'07)
2. Hattori, L. and M. Lanza. Syde: a tool for collaborative software development, Proc. of IEEE ICSE'10, 235-238
3. Holmes, R. and R.J. Walker. Customized awareness: recommending relevant external change events, Proc. of ICSE'10, 465-474

Theme 4: Frameworks for teaching and assessment of learning of GSD (2005-ongoing)

As an educator, Dana is passionate about employing our research insights in teaching our graduates the skills necessary to succeed in global teams. Having had the research immersion into a global team at Unisys prior to my UVIc appointment, in 2005 she designed, in collaboration with University of Bari, Italy and University of Technology Sydney, Australia, a Global Software Engineering course in which students from the three universities learned and practiced software engineering in real distributed software projects. The course emphasized requirements management, design and collaboration with remote customers, as these activities introduce the highest need for communication in distributed projects. This experience informed the research question:

12. *How can we teach and assess learning of GSD skills in our classrooms?*

In collaboration with Assoc. Prof. Allyson Hadwin in UVic's Educational Psychology Department, we began researching methods for the assessment of learning GSD skills, and our ICSE'06 publication [50] describes a framework of required GSD skills, as well as course activities and methods for assessing learning of these skills in GSD courses.

More recently, with an increasing adoption of agile development methodologies in distributed projects, we started a research project that investigates the usefulness of agile processes as a teaching tool in our GSD courses. In collaboration with Prof. Casper Lassenius of Aalto University, Finland, we are conducting a long-term, longitudinal study of the use of agile methods in GSD learning. The question we are addressing is:

13. *Are distributed agile methods, with their emphasis on iterative development, customer involvement, and continuous communication, an effective way of teaching GSD?*

We have taught two editions of a course across UVic and Aalto University and used Distributed Scrum as the development methodology in our GSD course project ([46]). The course has been very successful and highly rated at both universities. Our ICSE'13 publication [55] describes our instructional design as well as teaching assessment techniques, and provides empirical evidence of the GSD learning in the course. The course has been introduced into our regular software engineering curriculum in the department and will be offered yearly.

Doctoral students: A. Schroeter, K. Blincoe, A.Borici, P. Ratty (MSc)

Recent Theme 5: Requirements management in software ecosystems (2012-ongoing)

We live in a highly collaborative world, in the age of the platform[1]. Given our results in socio-technical coordination of large software projects, in this line of research we seek to recontextualize our work in even larger systems: we study software ecosystems as a new perspective for understanding the complexity of platform-based software development enterprizes that drive many large projects such as IBM's CLM[2] (Collaborative Lifecycle Management system, of which RTC is a part), along with open-source projects such as Gnome or Apache. As a first step, we are interested in how requirements for many interrelated products within the ecosystem emerge from the input of a large base of stakeholders or actors in the ecosystem, and how they are managed towards stakeholder satisfaction. Research questions we are asking include:

1. *How are stakeholders selected and integrated in requirements discovery and management processes in software ecosystems?*
2. *How are requirements elicited and prioritized to the satisfaction of many stakeholders in software ecosystems?*
3. *How are requirements mapped to actors (products) in the ecosystem and how are concerns that cross-cut products in the ecosystem managed?*

Recently we began a mixed-method study of IBM's CLM ecosystem. This ecosystem offers an interesting starting point to study requirements management in platform-based ecosystems, because it uses the concept of transparency[3] in an open commercial development model, in which features are derived and discussed in an open communication channel with all stakeholders in the ecosystem. We used extensive online document analysis, participant observations and interviews with CLM ecosystem managers. Our preliminary analysis describes processes of stakeholder management as well as requirements discovery, as well as forces and frictions inherent in the open communication channels used for discussions of features in the CLM product ecosystem. A publication is being prepared for RE'14.

In longer term, with this understanding from IBM's CLM ecosystem, we will move to studying how automated techniques of analysis of feedback from the community can benefit requirements discovery for products in software ecosystems. Studies have already identified that feedback from the user community can be mined for feature requests[4,5], although we know little about the effectiveness of the processes needed to support this feature discovery. Identification of features from large user feedback repository is only the beginning, the real value will be in techniques for the identification of key players in the ecosystem as well as their effective involvement in the requirements discovery and management for product success.

Doctoral Students: A. Knauss
Post-doctoral fellows: Dr. E. Knauss, K. Blincoe
Planned Publication venues: RE'14, ICSE'14

---

[1] Simon, P. The age The Age of the Platform: How Amazon, Apple, Facebook, and Google Have Redefined Business, 2011

[2] http://www-01.ibm.com/software/rational/alm/collaborate/

[3] L. Dabbish, H. C. Stuart, J. Tsay, J.D. Herbsleb: Leveraging Transparency. IEEE Software 30(1): 37-43 (2013)

[4] Pagano, D., W. Maalej, User Feedback in the AppStore: an empirical Study, Proc. of IEEE int'l Conf. on Requirements Engineering, 2013

[5] Galvis, L. and K. Windblath, Analysis of User Comments: An Approach for Software Requirements Evolution, Proc. of ICSE'13

Recent Theme 6: Run time discovery of contextual requirements for self-adaptive systems (2012-ongoing)

Our work in stakeholder interaction (Theme 2) combined with expertise acquired in data mining techniques and the increasing trend toward mobile devices, has generated interest in how to support the changing needs of mobile device users at run time. Complementing existing research in context-aware and self-adaptive systems, we are investigating 'reflective approaches'[1] to the discovery of contextual requirements for socio-technical systems at run-time. We are currently addressing a first research question:

4. *What are the contextual requirements in socio-technical systems, and can the context of use be modeled for discovery of contextual requirements at run-time?*

We are exploring the use of data mining techniques in the analysis of system's context and of usage at run-time. We have been working with a large dataset generated by a team of four athletes using a mobile coordination device on a 72-day boat trip across the Atlantic Ocean. Preliminary analysis results indicate the usefulness of data mining approaches to identify frequent patterns in the context changes that allow the operationalization of contextual requirements at run time. In our next steps we will refine our techniques and investigate how to identify these contextual requirements and associated architectural changes automatically at run-time. Our long-term research goal is to develop techniques for modeling and automatic identification of contextual requirements at run-time.

Doctoral students: A. Knauss, A. Rook (MSc)
Planned publication venues: SEAMS'14, RE'14, ICSE'14

[1] Finkelstein, A. and A. Savigni, A framework for requirements engineering for context-aware services, Proc. of Int'l Workshop From Software Requirements to Architectures, 2001
Bencomo, N., J.Whittle, P. Sawyer, and A. Finkelstein. Requirements reflection: requirements as runtime entities, Proc. of ICSE 2010

SELECTED PUBLICATIONS (STUDENT COLLABORATORS ARE BOLDED) – as of Jan 2014

Journal publications

1. Damian, D., and D. Zowghi, Requirements challenges in multi-site software development organizations, *Requirements Engineering Journal 8,* pp. 149-160, 2003.
2. Damian, D., D. Zowghi, L. Vaidyanathasamy, and Y. Pal, An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software, *Int. Journal of Empirical Software Engineering,* 9(1-2), pp.45-75, Mar. 2004.
3. Damian, D., L. Vaidyanathasamy, and Y. Pal, Requirements Engineering and Downstream Software Development: Findings from a case study, *Journal of Empirical Software Engineering*, 10(3), pp. 255-283, 2005.
4. **Layman, L**., L. Williams, D. Damian, and H. Bures, Essential communication practices for extreme programming in a global software development team, *Journal of Software and Technology*, 48(9), pp. 781-794, Sept. 2006
5. Damian, D., and **J. Chisan**, An empirical study of the complex relationships between requirements engineering processes and other processes that lead to payoffs in productivity, quality and risk management, *IEEE Transactions on Software Engineering*, 32(7), pp. 433-453, 2006
6. Damian, D., Stakeholders in global requirements engineering: lessons learned from practice, *IEEE Software*, Mar./Apr. 2007
7. Damian, D., F. Lanubile and **T. Mallardo,** On the need for mixed media in distributed requirements negotiations, *IEEE Transactions on Software Engineering,* vol. 34, no. 1, pp. 116-132, Jan. 2008
8. **Wolf, T., T. Nguyen** and D. Damian, Does distance still matter?, *Journal of Software Process: Improvement and Practice*, 13(6), pp. 493-510, 2008
9. **Wolf, T., A. Schröter**, D. Damian, **L. Panjer and T. Nguyen**, Mining task-based social networks to explore collaboration in software teams, *IEEE Software*, Jan./Feb., pp. 58-66, 2009
10. **Kwan, I., Schroeter, A**. and D. Damian, Does socio-technical congruence have an effect on software build success? A study of coordination in a software project, *Journal of Transactions on Software Engineering*, 37 (3), 307-324, 2011
11. **Calefato**, F., D. Damian and F. Lanubile. Computer-mediated communication to support distributed requirements elicitation and negotiation tasks, *Journal of Empirical Software Engineering*, 17(6), 2012, 640-674
12. **Kwan,** I., M. Cataldo and D. Damian, Conway's law revisited: The evidence for a task-based perspective, *IEEE Software*, 29(1), 90-93, 2012
13. **Knauss, E.,** D. Damian, J. Cleland-Huang and R. Helms, Patterns of Requirements Clarifications, *Journal of Requirements Engineering (REJ)*, submitted 2013, accepted pending minor revisions

Book chapters

14. Singer, J., S. Easterbrook, M.-A. Storey and D. Damian, Matching methods to questions in empirical software engineering, in *Advanced Topics in Empirical Software Engineering*, Springer Verlag, J. Singer, F. Shull, and D. Sjoberg (Eds.), Springer, 2007
15. Damian, D., **I. Kwan and S. Marczak**, Requirements-driven collaboration: leveraging the invisible relationships between requirements and people, A. Finkelstein, J. Grundy, A. van der Hoek, I. Mistrík, and J. Whitehead (Eds.), *Collaborative Software Engineering*, 2010.
16. Damian, **D., S. Marczak and I. Kwan**, Collaboration Patterns and the Impact of Distance on Awareness in Distributed Requirements-driven Collaboration, in *Global Software and IT: A Guide to Distributed Development, Rightshoring and Outsourcing*, IEEE Computer Society Press, Wiley, USA, 2011
17. **Lim, S.,** D. Damian and A. Finkelstein, StakeSource: A Web2.0 Tool for Stakeholder Analysis in Requirements Elicitation, in *Managing Requirements Knowledge*, W. Maalej and A.K. Thurimella (Eds.), Springer, 2011

Conference publications

18. Damian, D., and D. Zowghi, The impact of stakeholders' geographical distribution on requirements engineering in a multi-site development organization, *Proc. of the 10th IEEE Int. Conf. on Requirements Engineering (RE'02)*, Essen, Germany, pp. 319-328, 2002
Best paper nomination and invitation to submit an article to the *Requirements Engineering Journal* – see [1]

19. Damian, D., and J. Chisan, Exploring the role of requirements engineering in improving risk management, *Proc. of Int. Conf. on Requirements Engineering, RE'05*, pp. 481-482, Paris, Aug. 2005.
20. Damian, D., F. Lanubile and **D. Mallardo**, The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations, *Proc. of Int. Conf. on Software Engineering (ICSE)*, Shanghai, May, 685-690, 2006
21. Damian, D., **S. Marczak and I. Kwan**, Collaboration patterns and the impact of distance on awareness in requirements-centred social networks, *Proc. of IEEE Int. Conf. on Requirements Engineering (RE)*, New Delhi, pp. 59-68, Oct. 2007.
22. Damian, D., **L. Izquierdo**, J. Singer and **I. Kwan**, Awareness in the wild: why communication breakdowns occur, *Proc. of IEEE Int. Conf. on Global Software Engineering (ICGSE)*, New Delhi, Oct. 2007.
23. **Calefato, F**., D. Damian and F. Lanubile, An empirical investigation on text-based communication in distributed requirements workshops, *Proc. of Int. Conf. on Global Software Engineering (ICGSE)*, New Delhi, Oct 2007
24. **Marczak, S.,** D. Damian, U. Stege, and **A. Schröter**, Patterns of information flow in interdependent requirements social networks and implications for requirements-driven collaboration, *Proc. of IEEE Int. Conf. on Requirements Engineering (RE'08),* Barcelona, Spain, Sept. 2008. *Acceptance rate: 22%*
25. **Nguyen, T., T. Wolf and** D. Damian, Global software development and delay: does distance still matter?, *Proc. of IEEE Int. Conf. on Global Software Engineering (ICGSE'08)*, Bangalore, India, Aug. 2008.
Best paper nomination and invited to submit to the Journal of Software Process: Improvement and Practice – see [8]
26. **Wolf, T., A. Schroeter**, D. Damian and **T. Nguyen**, Predicting build failures using social network analysis on developer communication, in *Proc. of IEEE Int. Conf. on Software Engineering (ICSE)*, Vancouver, pp. 1-11, May 2009. *Acceptance rate 12.3%*
27. Damian, D., **S. Marczak, M. Dascalu**, M. Heiss and A. Liche, Using a real-time conferencing tool in distributed collaboration: an experience reoprt from Siemens IT Solutions and Services, *Proc. of IEEE Int. Conf. on Global Software Engineering (ICGSE)*, Limerick, 2009
28. **Lim., S.L**, D. Damian and A. Finkelstein, Stakesource2.0: Using social networks of stakeholders to identify and prioritize requirements, Proc. of the *Int. Conf. on Software Engineering (ICSE),* Hawaii, 2011.
29. **Kwan, I.** and D. Damian, New ideas and emerging results: the hidden experts in software-engineering communication, *Proc. of Int. Conf. on Software Engineering (ICSE)*, Hawaii, 2011.
30. **Marczak, S**. and D. Damian, How Interaction between Roles Shapes the Communication Structure in Requirements-Driven Collaboration, *Proc. of the International Conference on Requirements Engineering (RE'11),* Trento, Italy, 2011
31. **Schroeter, A., J. Aranda** and D. Damian, To talk or not to talk: factors that influence communication around changesets, *Proc. of the 2012 ACM Conference on Computer Supported Cooperative Work (CSCW)*, Seattle, February 2012
32. **Knauss, E.,** D. Damian, **G. Poo-Camano** and J. Cleland-Huang, Detecting and classifying patterns of requirements clarifications, *Proc. of IEEE International Conference on Requirements Engineering (RE'12)*, Chicago, USA, 2012
33. **Aranda, J.,** D. Damian and A. **Borici,** Transition to Model-Driven Engineering: What is revolutionary, what remains the same, *ACM/IEEE 15th International Conference on Model Driven Engineering Languages & Systems (MODELS '12),* Austria, 2012
Best paper nomination and invitation to submit a paper to the *International Journal on Software and Systems Modeling* (SoSyM) – in preparation

34. Damian, D., R. Helms, I. **Kwan, S. Marczak,** and **B. Koelewijn,** The role of domain knowledge and hierarchical control structures in socio-technical coordination, *Proc. of IEEE Int. Conf. on Software Engineering (ICSE)*, San Francisco, May 2013.
35. **Blincoe, K**., P. Valetto and D. Damian, Do all task dependencies require coordination? The role of task properties in identifying critical coordination needs in software projects, *Proc. of the ESEC/ACM SIGSOFT International Conference on Foundations of Software Engineering (FSE) 2013*, St. Petersburg, Russia, 2013
36. **Knauss, E**., D. Damian and V:Issue:Lizer, Exploring Requirements Clarification in Online Communication Over Time, *Proc. of 35th International Conference on Software Engineering (ICSE '13*), 1327-1330, San Francisco, USA, 2013. Formal Tool Demonstration.
37. **Räty, P., B. Behm,** K. Dikert, M. Paasivaara, C. Lassenius and D. Damian, Communication Practices in a distributed scrum project, *Proc. of 4th International Conference on Collaborative Innovation Networks*, Chile, August 2013
    Best paper nomination and invitation to submit a paper to the *International Journal on Organizational Design and Engineering* – in preparation

Refereed workshop publications

38. **Panjer, L**., D. Damian and M.-A. Storey, Cooperation and coordination concerns in a distributed software development project, *Proc. of Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, Int. Conf. on Software Engineering (ICSE), Leipzig, May 2008.
39. **Schroeter, A., I. Kwan, L. Panjer** and D. Damian, Chat to succeed, *Proc. of Int. Workshop on Recommendation Systems for Software Engineering*, at ACM SIGSFOT FSE, Atlanta, Nov. 2008.
40. **Kwan, I**., and D. Damian, Extending socio-technical congruence with awareness relationships, *Proc. of the 4th Intl Workshop on Social Software Engineering* at ESEC/FSE, Szeged, Hungary, 2011
41. **Borici, A., A. Schroeter** and D. Damian, Embracing distributed work: distance shall matter less, *Proc. of Int. Workshop on The Future of Collaborative Software Development,* at the ACM Conf. on Computer Supported Cooperative Work (CSCW), Seattle, February 2012
42. **Schroeter, A., A. Borici** and D. Damian, Integrated development environments lead to departmental borders breakdown, *Proc. of Int. Workshop on The Future of Collaborative Software Development,* at the ACM Conf. on Computer Supported Cooperative Work (CSCW), Seattle, February 2012
43. **Knauss, A., A. Borici, E. Knaus** and D. Damian, Towards understanding requirements engineering in IT ecosystems, *Proc. of Intl. Workshop on Empirical Requirements Engineering (EmpiRE),* Sept, Chicago, 2012
44. **Aranda, J.** and D. Damian, A redefinition of roles and collaboration under model-driven development, *Proc. of the International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE),* at the International Conference on Software Engineering (ICSE'12), Zurich, Switzerland, June 2012
45. **Borici**, **A., K. Blincoe, A. Schröter,** G. Valetto and D. Damian, ProxiScientia: toward real-time visualization of task and developer dependencies in collaborating software development teams, *Proc. of Int'l workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, ICSE'13, June 2-9, Zurich, Switzerland, 2012
46. Damian, D., C. Lassenius, M. Paasivaara, **A. Borici** and **A. Schröter**, Teaching a globally distributed project course using scrum practices, *Proc. of Int'l Workshop on Collaborative Teaching of Global Software Development (CTGDSD),* at ICSE'13, Zurich, Switzerland, 2012
47. **Simpson, B., E. Kalliamvakou, N. Lambert** and D. Damian, Aduno: real-time collaborative work design in a shared workspace, *Proc. of Int'l Workshop on Cooperative and Human Aspects of Software Engineering (CHASE*) at ICSE 2013, San Francisco, USA, 2013
48. **Blincoe, K**., G. Valetto, and D. Damian. Uncovering Critical Coordination Requirements through Content Analysis, *Proc. of Int'l Workshop on Social Software Engineering (SSE),* at SIGSOFT FSE'13, 2013

Teaching-related publications

49. **Damian, D**., B. Al-Ani, D. Cubranic and L. Robles, Teaching requirements engineering in global software development: a report on a three-university collaboration, *Proc. of Workshop on Requirements Engineering Education and Training,* Paris, Aug. 2005.
50. **Damian, D**., A. Hadwin and B. Al-Ani, Instructional design and assessment strategies for teaching global software development: a framework, *Proc. of Int'l Conference on Software Engineering (ICSE)*, 685-690, 2006
51. Scott, T., **D. Damian** and A. Hadwin, Enabling shared understanding: implications for instructions and learning in global software development, *Proc. of International Conference on Educational Media (EDMEDIA),* Vancouver, June 2007.
52. **Damian, D**., M. Petre, M. Miller and A.F. Hadwin, Instructional strategies in the EGRET course: an international graduate forum on becoming a researcher, *Proc. of the Western Canadian Conference on Computing Education (WCCCE)*, May, Vancouver, 2012
53. **Damian, D**., and A. Borici, Teamwork, coordination and customer relationship management skills: as important as technical skills in preparing our SE graduates, *Proc. of Int'l Workshop on Software Engineering Education with real life examples (EduRex)*, June 2-9, Zurich, Switzerland, 2012
54. **Damian, D**., C. Lassenius, M. Paasivaara, A. **Borici** and A. Schröter, Teaching a globally distributed project course using Scrum practices, *Proc. of Int'l Workshop on Collaborative Teaching of Global Software Development (CTGDSD*), June 2-9, Zurich, Switzerland, 2012
55. Paasivaara, M., C. Lassenius, **D. Damian**, P. Raty and A. Schroeter A, Teaching students global software development using distributed Scrum, *Proc. of Int'l Conf. on Software Engineering (ICSE'13*), 2013