A SYNERGISTIC RELATIONSHIP should exist between the global community of software practitioners and the community of software engineering researchers in academia, both of which are large and diverse. Practitioners could obtain more relevant research results, and researchers could better understand the type and scope of problems to examine. However, these communities are often disconnected, with the level of joint industry–academia collaborations (IACs) in software engineering still rather low, and these benefits not yet fully realized.

This chasm is beginning to close, through software engineering research conferences’ efforts to reach out to practitioners who can provide input and guidance to researchers. For example, during the 2018 International Conference on Software Engineering (ICSE), an Industry Forum encouraged collaboration (https://www.icse2018.org/track/icse-2018-Industry-Forum). The empirical-software-engineering community has also emphasized IAC in two ways:

- Academic and industry members interested in fostering IAC are supporting the International Software Engineering Research Network (ISERN; http://isern.iese.de).

In addition, others have previously written about this topic.1–3

IAC can range from a university collaborating with a company to collaborations among multiple universities and companies. These interactions can be formed and facilitated in multiple ways, such as

- companies awarding grants for a specific project,
- building on a long-term existing relationship to facilitate a new project,
- developing a series of projects through a long-term university-level collaboration with industry, or
- building a global network of partners to scale IAC connections.

Example 1: Documenting Architectural Decisions

Documenting architectural decisions has benefits for system design and evolution. Unfortunately, industrial adoption of this practice is low. Previous attempts at research prototypes failed because they were too labor intensive, lacked context-specific validation, or weren’t embedded with existing tools or design processes.

To address this low adoption rate and overcome some of these barriers, Paris Avgeriou and his team built industry-ready tool support for documenting five industrially validated architectural views. Here, the interaction involved one university and one company.

Finding the Connection

Avgeriou and his colleagues received a grant from ABB through its Software Research Grant Program, which funds university research collaborations through an open competition.
Description of Collaboration
To help ensure that the new tool would overcome some of the barriers faced by previous tools, Avgeriou and his colleagues undertook the following activities during tool design and development. First, they interviewed ABB system architects. Second, they analyzed system designs and documentation. Third, they developed the tool as an extension of Sparx Systems’ Enterprise Architect (ABB’s tool of choice). Finally, they interacted frequently with ABB to obtain early feedback.

After developing the tool, Avgeriou and his colleagues validated its usefulness and ease of use. They conducted two case studies of the tool during daily architectural tasks. The results showed that, although the tool was useful, its success depended on several contextual factors. To better understand these factors, they performed a follow-up longitudinal study in another unit of ABB. They adapted the tool to address that unit’s needs. They then evaluated the tool both during the initial phase of design exploration and two years later to identify any benefits of using the tool for the involved projects.

Lessons Learned
This project led to four key lessons learned:

- The tool found champions in the organization who appreciated its functionality and were willing to use it daily.
- The research team’s offering aligned well with the industrial team’s needs.
- Frequent interaction with company stakeholders was key to understanding their needs and gaining their support and trust.
- After the end of the official grant period, the research team needed to still spend resources (voluntarily) on the project, mostly to perform lightweight maintenance of the tool and support users.

Example 2: Automatic Bug Assignment
Processing the constant inflow of bug reports produced by large-scale software projects from the field and in an organization is a challenge. Before developers can fix bugs, the organization must assign the bugs, which is labor intensive and error prone.

The research question here was whether it’s possible to automatically assign bugs to teams by using historical bug data to train machine-learning techniques for classification. Here, the interaction involved two universities and one company.

Finding the Connection
Five factors enabled this interaction, some of which were mostly outside the control of academia or industry. First, an existing relationship helped remove some of the initial legal and social barriers that otherwise might have existed. Second, the collaboration benefited from a PhD student at the university and a PhD student at the company who had similar but complementary research interests in machine learning and automated bug assignment. Third, each student had independently begun working on tasks that would ultimately make the collaboration valuable. Fourth, the two students met serendipitously during a research conference far from their home country. Finally, the PhD supervisors and industrial manager were able to devise an approach in which the students published joint papers and completed their individual dissertations.

Description of the Interaction
Both students had conducted in-depth studies of bug reports at two separate companies. After meeting at the conference, they worked together on both datasets, comprising 50,000 bug reports, to develop a bug assignment algorithm. They developed experimental tools, using the open source machine-learning tool Weka. They evaluated several algorithm variants, using standard ten-fold cross-validation and a more realistic evaluation scenario with time windows. The experiments demonstrated an accuracy matching the current manual processes, so the company incorporated a simplified version of the bug assignment into its toolset. Throughout the core of the analysis process, the two students worked together in physical meetings at the industry site.

Lessons Learned
This interaction led to five key lessons learned:

- Build on existing or previous connections between the academic and industry partners.
- Find the right collaborators on both sides.
- The timing of the interaction must fit the requirements on both sides.
- It’s challenging to achieve a result using data from two companies that’s general enough for both companies. It requires trade-offs between each company’s needs and the available data.
- Going from research prototypes and an academic publication to a deployed solution takes time.
(The work was done from 2013 to 2014, the paper was published in 2016, and the internal tool was deployed in 2017 and 2018.)

For more details about this work and its results, see the related publication.7

Example 3: Software Quality Improvement
The US Army Corps of Engineers’ Construction Engineering Research Laboratory (CERL) is leading the development of the Sustainment Management System (SMS), which aids maintenance of military installations. CERL has teamed with Montana State University’s TechLink Software Engineering and Analysis Laboratory (TSEAL) to research, develop, and deploy new techniques in industry to address the SMS software’s quality. TSEAL focuses on researching empirical quantitative techniques to measure the software’s overall quality and on operationalizing the findings as tools for CERL staff and contractors.

To date, TSEAL has developed dashboard technology to help measure various quality-assurance (QA) metrics, assess technical debt, and explore new approaches to measuring cybersecurity. It also has used AI techniques to improve decision making about maintenance and day-to-day operations of military bases and to provide functional testing for critical SMS software functionality. Besides the work’s direct impact on industry, the research team has shared the results through peer-reviewed publications.8,9

Here, the interaction has involved one university and one company.

Finding the Connection
This collaboration has resulted in a win–win relationship. From the industrial perspective, the benefits include the following:

• CERL receives recognition for supporting academia.
• CERL receives improved QA analysis of its SMS software.
• CERL can adopt software at a significantly reduced price.

From the academic perspective, the benefits include the following:

• Undergraduate, master’s, and PhD students receive support.
• Undergraduates can work in a professional environment on products that are actually used.
• Faculty receive support for important research.

Lessons Learned
Some of the lessons learned from interacting with government organizations include the following:

• The strict (but necessary) process can result in challenging interactions that require patience and persistence.
• Communication channels can be difficult owing to the presence of multiple proxies between engineers.
• Significant cultural gaps (including professional skill and professional and social maturity) exist between professional software practitioners, managers, academics, and students across government, industry, and academia. These gaps must be bridged.
• Convincing the industrial partners to change their practice and use the academic tools is challenging and can take months.

Example 4: Requirements-Based Testing
This study addressed how to use defect taxonomies to build a requirements-based testing approach. The first problem was designing such an approach. The second problem was whether the new approach was more effective and efficient than standard requirements-based testing.10 In addition, the approach was applied to review requirements.11 Here, the interaction involved one university and one company.

Finding the Connection
In this case, the industrial partner’s test manager initiated the collaboration. The test manager saw the need for improved testing capabilities. So, the test manager asked a researcher to perform informal testing, owing to the researcher’s expertise. This personal collaboration led to additional collaborations and even a personal friendship.

Description of the Interaction
The partners interacted intensively via telephone conferences and joint visits. They also shared a joint repository that allowed knowledge transfer to flow in both directions. The primary academic and industrial contacts even developed a model for mutual knowledge transfer between industry and academia based on the collaboration.12

Lessons Learned
This collaboration led to four key lessons learned:

• Personal commitment was needed from both the researcher and the industrial test manager. (Proper management support could be challenging.)
What’s Next?

IAC has many dimensions and formats. For example, projects such as HELENA (Hybrid Development Approaches in Software Systems Development; https://helenastudy.wordpress.com) and NaPiRE (Naming the Pain in Requirements Engineering; http://www.napire.org) have formed a network of researchers to scale access to industry and data collection, through a series of global surveys. Some projects live in between applied research and consulting, such as the ones Daniel Méndez Fernández and Birgit Penzenstadler conducted on requirements-engineering improvements.¹³

In the end, IAC on software engineering is crucial to advance the field. For example, this year’s ICSE witnessed impressive industry attendance and several ways to interact, including the Software Engineering in Practice track and the inaugural Industry Practice track and the inaugural Industry Forum. Several talks mentioned that researchers should spend more time in industry and that industry could benefit from academia solving clearly defined and existing problems.

Also, effort is ongoing within the ISERN community to identify IAC patterns and ways to leverage such interactions. A whole session at this year’s ISERN meeting will focus on this topic.

It’s important to understand the challenges of and lessons learned from these examples, to leverage them as good practice aiming at impactful results. So, we finish with two key takeaways:

• Choosing the appropriate size and complexity of the selected case projects was important and a challenge.
• Stepwise refinement of the approach by analytical and empirical methods was necessary.
• It was useful to apply quantitative and qualitative methods as well as more than one study type (a case study and an experiment) to collect the relevant measures.

Acknowledgments

We thank Paris Avgeriou, Michael Felderer, Clemente Izurieta, Marco Kuhrmann, Daniel Méndez Fernández, Birgit Penzenstadler, and Per Runeson for their contributions to the summaries in this article.

References


ABOUT THE AUTHORS

JEFFREY C. CARVER is a professor in the University of Alabama’s Department of Computer Science. Contact him at carver@cs.ua.edu.

RAFAEL PRIKLADNICKI is an associate professor in the School of Technology and the director of the Science and Technology Park (Tecnopuc) at the Pontifical Catholic University of Rio Grande do Sul, where he also leads the MuNDDoS research group. He chairs IEEE Software’s advisory board. Contact him at rafaelp@pucrs.br.