Booting and Rebooting Academia-Industry Collaborations within Software Engineering Courses

Andrea Capiluppi, *Member, IEEE,*, and Letizia Jaccheri, *Senior Member, IEEE*

***Abstract*—Collaboration between academia and industry for education is a common practice, and several academic institutions already benefit from that. While best practices have been shared on how to run these collaborations, the process of starting these collaborations is less clear. Even long established schemes, like the one running at the Norwegian University of Science and Technology (NTNU) for 50 years, struggle to remember how it originally started, or how to keep it running after the original initiators quit.**

The aim of this paper is to propose the steps to start and sustain a collaboration framework between academic courses and industrial collaborators, and in the context of Software Engineering. The framework was originally conceived, and deployed from scratch, at the University of Groningen (RUG): so far it has successfully engaged several industrial partners, and several dozens of projects deployed. The framework and its steps are also usable to reboot other similar collaboration schemes, especially when (like at NTNU) their originators have left the institution.

***Index Terms*—industry collaboration, framework, case study**

I. Introduction

When the first author moved to a new university and wanted to start a cooperation with the local IT industry, he had no idea of how to start/boot it. When the second author inherited, from its historical founder, one of the most prestigious software engineering course in her country, she panicked that she would not be able to sustain it and bring it into the future. The idea of this paper was born from a conversation about how to boot and reboot academia-industry collaborations within software engineering courses in a systematic and informed way.

The importance of academia-industry links is well understood: several academic researchers and practitioners have studied the phenomenon, with the aim to 'close the loop' between what is taught in universities and what is actually requested by industry ([5], [6], [9]).

Benefits and advantages of these collaborations have been analysed in detail, and in diverse settings, universities and courses: scholars and practitioners usually report *case studies*, involving the description of how specific university courses helped students establishing connections with real industry needs ([2]–[4], [7]–[12]), or have a better feel of how their skills aligned with the industrial needs.

All these case studies are important to show how the academia-industry collaboration should be a central aspect of higher education, rather than a complement to it. Nonetheless, there are still gaps in understanding how a working collaboration scheme could be applied (e.g., repeated) to another setting; or how a new one could be kick-started in case there wasn't one before. Further gaps in the literature, or the state of practice, creep up when the academics originating the scheme are no longer available: at NTNU, for example, the Software Engineering Customer Driven course was run for 50 years [1], but the original process details are now lost in time, and need to be revamped or adjusted to a new, more modern process.

In this paper, we present the steps contained in a collaboration framework that was initiated at RUG. The scheme was started from scratch, with the aim to establish and maintain industry links, and within the boundaries of academic courses focusing on Software Engineering. The framework is based on practical steps that have been distilled from trial-and-error: the main objectives of these steps are to create enough trust between academics and industrial collaborators, and to make the scheme sustainable over the years.

We articulate this paper as follows: section II describes the framework and its four steps (Planning, Preparation, Execution, Follow-up). section III describes its deployment to boot the academia-industry collaboration at RUG, while section IV describes how the same framework was used to reboot an existing collaboration scheme at NTNU. section V provides some lessons learned and conclusion.

II. Framework

The phases to boot and reboot our academia-industry collaboration are divided in four major phases: i) Planning, ii) Preparation, iii) Execution and iv) Follow-up. Each phase involves a number of steps and stakeholders, and it is described in detail below.

A. Phase one – Planning

The planning phase has various objectives: first, trying to evaluate whether this framework can be applied in the first place within an institution; second, isolating the courses (at the BSc or MSc level) that might be possible to integrate in this framework; third, create internal collaboration between the course and other existing outreach activities. We have identified three steps in this phase, that are necessary as a preliminary work before booting a similar scheme.

PL-1: *Isolate the Learning Objectives (LOs) that will be part of the industry-academia collaboration*. This is a very important preliminary step: the course coordinator chooses which LOs will be developed as academic outputs, and which ones will be developed as industrial deliverables. From other existing schemes, and also from ours, we noticed that industrial partners will be mostly interested in working code, whereas Software Engineering courses have typically a broader set of objectives, including soft skills.

Rationale: The framework presented here should disrupt as

little as possible the academic provision of a course, otherwise it will be too complicated to get started. The Software Engineering course at RUG, for example, requires students to work on the standard documents of the software development life cycle (SDLC): requirements, design, code and testing. Other departments might have different LOs, compendia or guidelines on how their Software Engineering courses run. Using existing LOs of the course (as is) for the industrial collaborations, instead of adding new ones, makes for an easier planning and faster boot of our proposed framework.
Stakeholders: course coordinator.

PL-2: *Interact with the local Business Liaison Officers (BLOs) and the student associations*. An academia-industry collaboration framework cannot live in isolation: universities have normally outreach programs, alumni and student associations that maintain some type of industry collaborations. In this step the course coordinator establishes a direct connection with the existing internal staff and department(s) and student associations that facilitate the collaboration with industries. The objective of this step is to find the best way to complement, instead of overlap, other existing outreach activities.
Rationale: Most universities already have BLOs and ad-hoc departments that establish connections between academic staff and industry partners. Also, student associations often provide valuable links for students who wish to participate in extra-curricular activities, especially in collaboration with an industry. The framework presented here is rooted from the ground up, and starting from software-related problems expressed by partnering industries. Therefore it is reproducible in other courses, as long as industry needs can be expressed as adjacent to the LOs of a course.
Stakeholders: course coordinator, BLOs, student association(s).

PL-3: *Create a professional organisational structure behind the scheme.* The framework should not live in isolation from other outreach initiatives within the university (i.e., PL-2). Similarly, the framework should be independent from the course coordinator, or past, current and future students. This should guarantee continuity in case the originators of the scheme leave the academic institution. Therefore, in this step the course coordinator, with the help of the student associations, establishes an academic entity that acts on behalf of this academia-industry scheme. This entity has a profile on professional networks (i.e., LinkedIn[1]) in order to (i) recruit further collaborators; (ii) connect past and present students who participated in the courses; and (iii) keep an ongoing dialogue with partnering industries.
Rationale: in our experience, it is very important to detach the course in our scheme from the professional profile behind it. In the long run, this has had the effect of creating an ecosystem of past and present students, industrial partners and additional courses that can be included in the same scheme. As a result, establishing such an entity has reinforced trust, recurring business and increased collaboration with new industrial partners.
Stakeholders: course coordinator, student association(s).

[1]https://www.linkedin.com/UnICorug/

B. Phase 2 – Preparation

There are a few activities that need to be run before running a similar scheme for the first time (and only once), in order to set up the resources needed for establishing a fruitful collaboration. The maintenance, expansion and upgrades of these resources are part of the follow-up phase.

PR-1: *Create a welcome pack to inform industrial partners and BLOs.* In this activity the course coordinator prepares a document detailing the purpose of the collaboration, the benefits for the students and the industrial collaborators. The timeline of the collaboration, the duties of the parties and the deliverables are also described. If other courses need to be included in the scheme, a similar document should be prepared by the respective course coordinators.
Rationale: This step was distilled after a lot of trial-and-error: the description of the RUG collaboration scheme was initially achieved meeting potential collaborators (online or in person), and explaining on a one-to-one basis the purpose of the collaboration. This approach was effective at generating proposals, but time-expensive: thus we created a welcome pack, containing the description of the scheme, and the benefits for the industrial collaborators. It is now common practice to connect to new interested parties with this welcome pack.
Stakeholders: course coordinator.

PR-2: *Create a proposal form for industrial partners to fill.* The academia-industry collaboration scheme that we propose is rooted in industry needs, rather than based on academic research. Therefore the course coordinator creates a form (either online or as an editable document) to collect software-based needs and circulates it to industrial collaborators. The form should be kept to a minimum: in our experience, this has increased the likelihood of a collaboration: (i) contact details of the industrial collaborator; (ii) the description of the problem of their software-based project; and (iii) the technology stack required to perform the project.
Rationale: It has become clear that bottom-up, problem-driven projects are an effective way to engage industrial interest. Software-based needs are easy to capture by means of one or two paragraphs of a *problem description*; and most projects would benefit a specific *set of technologies*, that better aligns with the technological setup at the partner's premises. Most interested companies benefit from the description of past project, so completed projects and their forms come handy for future interactions.
Stakeholders: course coordinator.

PR-3: *Set up a web portal to make the proposal form, and other information, available to potential collaborators.* Collecting forms has been proven to be prone to errors, it is based on a string of messages or emails, it relies on one individual (typically the course coordinator) and it is difficult to track. Therefore the course coordinator, in collaboration with the IT department, should set up a website that showcases the collaboration scheme, provide the collaboration form to

fill, and accepts forms filled in by collaborators. GET and POST methods are embedded in the web portal to accept proposal forms.

Rationale: The creation of a dedicated web portal at RUG[2] and at NTNU [3] has gradually shifted the problem of engaging industrial collaborators from individual communication, to a collection of filled-in forms. Information on the scheme, testimonials and the possibility to upload the proposal form made the interactions easier. Moreover, the portal accepts project proposals all-year long, and it can include different courses, once additional forms are prepared.

Stakeholders: course coordinator, IT department.

C. Phase 3 – Execution

The execution phase starts before the beginning of the course. The main actors of this phase are the course coordinator and the teaching assistants, although feedback from the students and the industrial collaborators are also required. There are 5 activities to this phase, as follows:

E-1: *Vet the proposals before the start of the course.* During this activity the course coordinator examines the proposals received either via direct communication, or through the web portal as filled-in forms. Proposals that do not contain computer science topics, or do not lend themselves to enough coding, are discarded.

Rationale: This activity is pivotal to the smooth running of the course, and it requires a good amount of attention to detail. Project proposals could be too low-level (e.g., programming hardware controllers) or too simple for groups of students, although they could be viable projects for individual students. In these cases, the project proposals should be discarded. It could be still possible to use the proposal in the context of other courses, so communication with interested academics (even from different backgrounds and faculties) could provide a match for these discarded proposals. It is important to involve the assistants for two reasons: first if they are involved from the beginning they will have greater ownership and engagement, second they provide a fresh view about new technology and new topics since they are computer science PhD .

Stakeholders: course coordinator, other academics, and assistants.

E-2 *Publish the vetted proposals for students to bid for, and team formation.* Ahead of the course, the course coordinator publishes the vetted proposals in some form (either online, or on a shared space) for students to evaluate. Based on personal preferences and past personal experience, students are allocated to one of their projects of choice. Alternatively, the coordinator (as in the case of NTNU) can allocate projects to and students to groups.

Rationale: For a large (e.g., 150-200 students) course, there is a need for 30-40 project proposals to be worked on[4]. The amount of information and detail for students to assess, in order to evaluate all those project proposals, is very large: it is therefore important to let them have a good look before the course starts. Also, forming teams with members chosen randomly among the course students does not guarantee allocating the more skilled students to the right project. Thus, students express ('bid for') up to 5 choices on a shared spreadsheet: subsequently, teams of 5 students get selected from these choices, allowing each student to work on one of their choices. The decision of splitting the class in groups of 5 students is to align the projects to the phase of the Softare Life-cycle, and have a designated student to each phase: 1) Requirement officer, 2) Chief architect, 3) Lead developer, 4) Testing officer. In addition, a student will act as the 5) Scrum master. NTNU, on the other hand, operates with approximately 100 students divided into groups of seven or eight students: the size of the groups is mostly a practical matter, and constrained by the number of available projects.

Stakeholders: course coordinator, course students.

E-3 *Introduce and assess soft skills.* The course coordinator (or other lecturers) gives one lecture specifically on soft skills, group work and professionalism. The lecture is placed at the beginning of the term. Team work, effective communication and cooperation are monitored and facilitated by the teaching assistant during their weekly meetings.

Rationale: Relatively to the university of Groningen, this experience is a unique opportunity for students to engage in group work involving a real industrial case study. During the inital boot of the RUG scheme (2019-2020) we observed that some behaviours were not as expected, especially considering attendance to meetings, and communication with the client. This lecture has so far served as a placeholder for students to digest the ground rules of the industrial placement. At NTNU, there is a tradition to invite external experts to give a lecture about team dynamics and one about presentation techniques. The lectures include practical exercises.

Stakeholders: course coordinator.

E-4 *Manage support and monitoring of groups.* This activity is organised by the course coordinator, but it is delegated teaching assistants. The course coordinator also act as supervisor of at least one group. Two weekly meetings are organised: one between the teaching assistant and the group, and one between the industrial collaborator and the group. Moreover there are bi-weekly meetings between all the group leaders and the teaching team and bi-weekly meetings of the teaching team.

Rationale: The weekly meetings have two different purposes: the first has a monitoring purpose, and the second serves as a progress report to the industrial collaborator. The meetings with the group leaders serve as a feedback mechanisms between the groups and the teaching team. It is important that the course coordinator also acts as a supervisor in order to learn together with the students and the assistants about new technology and new challenges that arise each year.

Stakeholders: course coordinator, course students, teaching assistants and industrial collaborators.

E-5 *Assess students on the stated learning outcomes.* The

[2]http://UnICo.web.rug.nl/

[3]https://sbs.idi.ntnu.no/tdt4290

[4]These numbers include the possibility of drop-out students, who enrolled in the program but then decided not to take part in it.

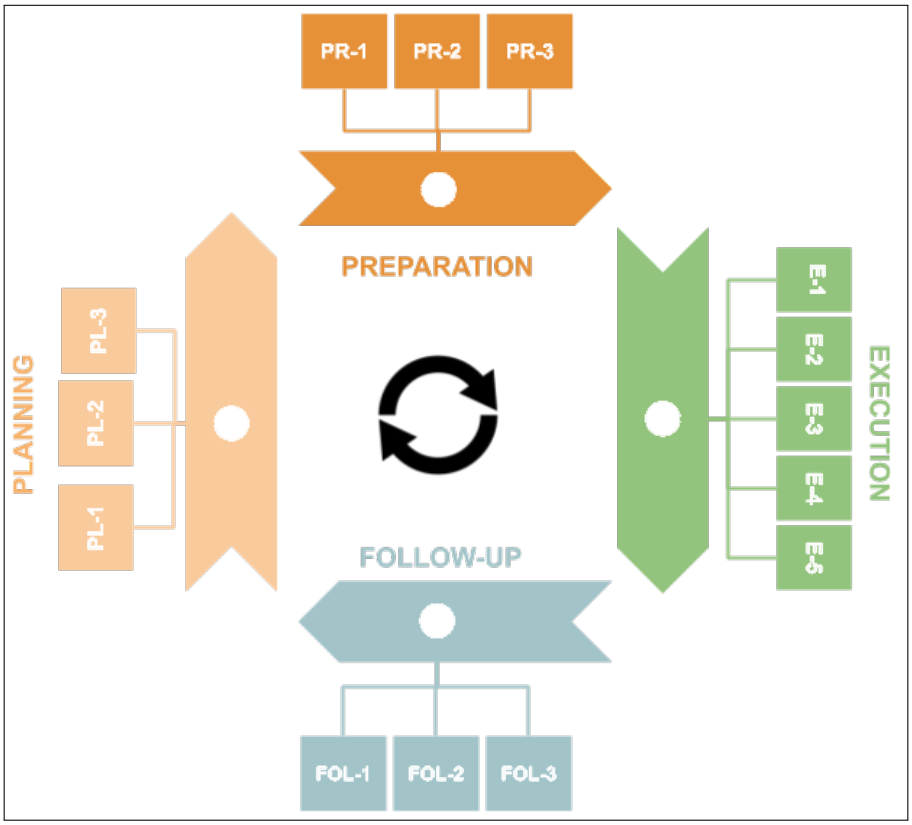

Fig. 1. Timeline of the phases and activities of the proposed framework

course coordinator evaluates the groups according to the University policies based on the deliverables selected in the activity **PL-1** above: for the final grade, less weight is given to "successful" or "unsuccessful" projects. It is good practice to organize a presentation day in which the groups present their process and product to the evaluators. The coordinator has the responsibility to make the evaluation criteria available to the students from the beginning of the course and to the evaluators.

Rationale: It is very important for the students to understand that the collaboration with the industrial partners is necessary, but not sufficient, to pass the course. The learning objectives need to be assessed whether the collaboration delivered a working software or not. Groups may fail if the deliverables lack the required quality; individual students may fail if their contribution to the group work was insufficient. At NTNU the reports must be evaluated by external evaluators. At NTNU customers and assistants are also invited to provide feedback that the evaluator can take into consideration.

Stakeholders: course coordinator with help of the assistants.

D. Follow-up

This phase runs after the completion of the course: wrapping up the experiences of both companies and students ensures that the next iteration of the course will benefit a better process, and an even more streamlined approach.

FOL-1 *Gather feedback from companies.* The course coordinator sets up a questionnaire to evaluate how the companies perceived the scheme. The questionnaire can be put online (e.g., CrowdTech, SurveyMonhey, Google Forms) and reminders set for collaborating partners to fill in.

Rationale: The feedback from the industrial partners is used to understand how students behaved in the work settings, and how professional their interactions were conducted: it serves the purpose to feed back into the **E-3** activity, dedicated to introduce the needed soft skills for this course. The industrial feedback does not serve the purpose to change the grade of the groups or individual students: the academic side of this process is handled by the course coordinator, with input from the teaching assistants. At NTNU customers are invited to a meeting in the middle of the semester to address ongoing issues and to a preparation meeting before semester start.

Stakeholders: course coordinator, teaching assistants.

FOL-2 *Evaluate if further work is needed on completed projects.* The course coordinator assesses whether the projects would benefit additional work, in terms of refactoring, missing tests or further features.

Rationale: most of the received proposals are generally too large in scope for one group to complete, and one of the first tasks of the students is to negotiate what is feasible in the time frame. The features that cannot be completed are placed in a Won't Do category (using the MOSCOW technique of requirements priorisation): the course coordinator checks is

these could be used for further work.
Stakeholders: course coordinator.

FOL-3 *Solicit more projects, and for different courses:* the course coordinator checks the status of the collaborations. If that was a successful project, the course coordinator decides whether the industrial partner could be interested in further projects within the same scheme, and communication for the next round of the course is established. Repeated partnerships are noted as long-term collaborations. At NTNU a database of existing customers is maintained and social media is used (linked in) to disseminate knowledge about the course and possibility to propose projects.
Rationale: The long term objective of this scheme is to create an ecosystem of students and industrial collaborators that act within the same goal: achieve a long term collaboration between academia and partnering industries. Industrial collaborators may get removed from the scheme, if the collaboration did not meet the expectations of students.
Stakeholders: course coordinator, other academic staff.

III. Booting an academia-industry collaboration scheme (RUG)

In this section we discuss how the template was created to boot the academia-industry collaboration scheme at RUG; and we identify the main descriptors for a selected number of activities.

- **[PL-1]**: marking criteria were developed for the following deliverables: (i) Requirements document; (ii) Design and architecture document; (iii) Source code; (iv) Testing. Students submitted all these documents at the end of the first block, gathered formal feedback on each, and submitted their final versions at the end of the course.
- **[PL-3]**: the UnICo[5] LinkedIn page was established prior to the run of the course, in collaboration with the local student association (Cover, the study association for Artificial Intelligence and Computing Science at RUG[6]), in order to connect current and past (i.e., alumni) students and industrial collaborators.
- **[PR-2]**: a first draft of an editable PDF was circulated via email to the interested collaborators. Afterwards, the PDF form was included on the web portal.
- **[PR-3]**: the UnICo web portal was set up at the end of the first iteration, in an attempt to make the process of collecting proposals as independent as possible. The website warns about new proposals being uploaded from interested parties, with an email to the course coordinator. Before the set up of the UnICo portal, 11 companies were involved with an overall 14 projects (2019-2020); in the last iteration of the course (2022-23), UnICo obtained 33 projects from 24 companies.
- **[E-1]**: the course does not have a lot of rejected proposals. Rather, the participants are asked to redefine the scope of their proposal, if it does not precisely suit the needs of the Software Engineering course.
- **[E-4]**: each TA manages 2 to 3 groups, and weekly meetings are held between all TAs and the course coordinator to check and monitor the progress of all projects.

[5]https://www.linkedin.com/company/UnICorug
[6]https://www.svcover.nl/

IV. Rebooting an academia-industry collaboration scheme (NTNU)

In this section we discuss how the template was used to systematize the re-boot the academia-industry collaboration scheme at NTNU; and we identify the main descriptors for a selected number of activities.

- **[PL-1]**: marking criteria were formalized. The expected learning outcomes from the course include (i) **knowledge** (to give students practical experience in completing all phases of a major project); (2) **skills** (the ability to organize and implement major projects, as well as to document and present the results to a real customer); and (3) **general competence** (as the insights into project work and how groups can be used to solve complex computer technical problems).

Evaluation criteria are formalised in a way that students and evaluators know very well in advance how the different dimensions will be evaluated and by whom. *Product* criteria account for 30% of the final grade; *Team dynamics* criteria for 20%; *Product and process* criteria for 20%; and *Documentation* for 30%.

- **[PL-3]**: The coordinator of the course uses her LinkedIn page and tags the relevant entities to link to publish information about the course.
- **[PR-2]**: the compendium was circulated via email to the interested collaborators. Afterwards, the PDF form was included on the web portal.
- **[PR-3]**: the NTNU web portal[7] was established and linked to a Microsoft form in an attempt to make the process of collecting proposals as independent as possible. The website warns about new proposals being uploaded from interested parties, with an email to the course coordinator. The portal also points to the NTNU rules for Intellectual Property Rights and informs the customers that by filling the form they accept the NTNU framework for IPR The student has the copyright to the assignment he/she writes. Having the copyright means deciding whether the work should be made available to the public. It also means that it is the student who decides whether the thesis can be copied, but NTNU can take the necessary copies for carrying out censorship and archiving.
- **[E-1]**: the course used to have a lot of rejected proposals. After the adoption of this framework, we have decided to change policy and the participants are asked to redefine the scope of their proposal, if it does not precisely suit the needs of the Software Engineering course.
- **[E-4]**: each TA manages 2 to 3 groups, and weekly meetings were held between all TAs and the course coordinator to check and monitor the progress of all projects. A compendium for the TA has been developed. This includes questions that the TAs are encouraged to ask the groups.

[7]https://sbs.idi.ntnu.no/tdt4290

- **[FOL-1]**: the form to gather feedback from the industrial collaborators is made available in the replication package.

V. Lessons learned and replicability

Below we present the main lessons while running various iterations of the courses

- *Diversity of projects*: by opening up to the local industries, these schemes get quickly involved in very different projects related to software development. Projects ranged from a full deployment of the frontend and the backend components; to a redesign of the only backend components; to the evaluation of diverse technologies (e.g., machine learning techniques, specialised hardware) for a routine internal task. In each case, it is important to keep the groups focused on the academic deliverables (requirements, design, etc), which are the shared pillars of any software engineering project.
- *Sustainability issues*: by opening up to the established IT industry, local start-ups and public sector entities, these schemes tend to get involved to address sustainability issues. We found that customers' project descriptions moderately addressed social sustainability, including gender diversity, for specific target groups. Technical sustainability is also addressed by a little more than half of the overall projects. However, we observed that no effort was made to address environmental and economic sustainability.
- *Aligning industrial needs to course offerings*: booting this framework at RUG, in the past four years we were able to observe what are the current needs of industrial partners, and how those match the technologies and pipelines currently taught in class (e.g., Java). For instance, we observed that most projects required either C++ or Python, whereas the commonly taught language across several courses is currently Java. This finding can be used to inform other course coordinators about the ongoing requests by industrial collaborators, and how that could influence the course offerings.
- *Steps and their relevance*: after booting at RUG, the infrastructure has now been tested and is robust for reuse. We realised that both when booting (RUG) and rebooting (NTNU) a similar scheme, **E-1** is still the most laborious one in the whole process. Most new collaborations require further clarifications and meetings between the parties. Repeated partnerships have to be preferred since the industrial partners already understand the scheme and their level of involvement.
- *Implementation of the scheme in different contexts*: although RUG and NTNU are only two scenarios where the collaboration scheme is being applied, its basic infrastructure is deployable in other contexts. One of the most visible differences between RUG and NTNU is based on what students need to work on at the respective locations. Although RUG focuses on the elements of the software life cycle, NTNU gives more weight to *team dynamics* (for example, how well decision-making is carried out in a collective manner, or how well roles are defined and maintained throughout the project) and *process aspects* (for example how the scope, time, communication, resources and risks are planned in advance, traced and managed during the project). This proves that the scheme is flexible enough to accommodate diverse learning criteria.
- *Non-Disclosure Agreements (NDAs)*: industrial collaborators may require students to sign an NDA, but only in those cases where industrial source code is shared with the students. Although this has become a common request at RUG, at NTNU we want to give priority to open projects. In both cases, we believe that this aspect should be formally explained in class, and a lecture was developed at RUG to describe the different types of NDA, and what they imply for who signs them while at NTNU these aspects are described in the portal and in the compendium.

VI. summary

References

[1] Rudolf Andersen, Reidar Conradi, John Krogstie, Guttorm Sindre, and Arne Sølvberg. Project courses at the NTH: 20 years of experience. In *Conference on Software Engineering Education*, pages 177–188. Springer, 1994.

[2] Shan Barkataki and Bolton Tom. Adding excitement to student projects: Try web based industry collaboration. In *2003 Annual Conference*, pages 8–164, 2003.

[3] David Carrington, Paul Strooper, Sharron Newby, and Terry Stevenson. An industry/university collaboration to upgrade software engineering knowledge and skills in industry. *Journal of Systems and Software*, 75(1-2):29–39, 2005.

[4] RC Edwards. Mechanical engineering technology senior projects–partnering with industry to enhance the students' capstone experience. *Technology Interface, Fall*, 2006.

[5] Yuanyuan Fan, Ana Evangelista, and V Indumathi. Implementation of industry inputs in higher education courses. In *2021 International e-Engineering Education Services Conference (e-Engineering)*, pages 9–16. IEEE, 2021.

[6] Vahid Garousi, Görkem Giray, Eray Tüzün, Cagatay Catal, and Michael Felderer. Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software*, 156:65–83, 2019.

[7] Márcia Juvane, Paul Spiesberger, Karl Pinter, Raoul Vallon, and Thomas Grechenig. Assessing mozambique's software industry to foster local universities-industry collaboration. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1740–1747. IEEE, 2020.

[8] Timothy C Lethbridge, Steve Lyon, and Peter Perry. The management of university–industry collaborations involving empirical studies of software engineering. In *Guide to Advanced Empirical Software Engineering*, pages 257–281. Springer, 2008.

[9] Syeda Wajiha Naim, Syed Asif Ali, Sumaira Hussain, and Basit Hassan Qureshi. A collaborative model to reduce gap between it industry and academia (cmrgia). *IJCSNS*, 19(5):118, 2019.

[10] Claudia Raibulet and Patricia Lago. Industrial project-based course on service oriented design: experience sharing. In *Proceedings of the First International Workshop on Designing and Running Project-Based Courses in Software Engineering Education*, pages 20–24, 2022.

[11] J Barrie Thompson and Helen M Edwards. Preparing graduate students for industry and life long learning: A project based approach. In *IFIP World Conference on Computers in Education*, pages 292–301. Springer, 2009.

[12] E. Venson, R. Figueiredo, W. Silva, and L. C. M. Ribeiro. Academy-industry collaboration and the effects of the involvement of undergraduate students in real world activities. In *2016 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, Oct 2016.