Third Generation Industrial Co-production in Software Engineering

Tony Gorschek and Krzysztof Wnuk

Abstract Industry-academia collaboration is one of the cornerstones of empirical software engineering. The role of researchers should be developing new practices and principles that enable industry in meeting the engineering challenges today and in the future. This chapter describes the third generation of industrial co-production in software engineering that includes seven steps. The co-production model and experiences associated with its use represent deep and long-term co-production with over thirty companies, many of which are still active partners in SERL

1 Introduction

Software is at the core of almost every product and service today. Doing your taxes, handling your bank errands, driving a car, or even booking a dentist appointment; all powered by software. Software has created unprecedented benefits for companies to be more effective, efficient and to create smarter products to compete in the marketplace. Software also enables the creation of completely new types of business. However, as more companies are created, and existing companies transform into software-intensive companies, the amount of software, and the amount of software development organizations, is exploding (Gorschek, 2018).

Software is also increasing in size, complexity, and interactions. This puts new demands on how software is conceived, developed, evolved and maintained; in essence software engineering. Striking a balance between creating business and customer value,fast and sustainable, whilst keeping costs as low to maximize market competitiveness.

The role of researchers can and should be developing new practices and principles that enable industry in meeting the engineering challenges of today and the future.

Department of Software Engineering
Blekinge Institute of Technology, Karlskrona, Sweden SE-371 79
e-mail: Tony.Gorschek@bth.se e-mail: Krzysztof.Wnuk@bth.se

For clarity. The terms “new practices and principles” contain any method, model, framework, practice, tool, ways-of-working, and so on that enable the engineering, evolution and long-term asset management of software and/or software intensive products and services. From now on, all these practices and principles are called Solution(s).

The role of a software engineering researcher is the application of a scientifically based and valid methodology to develop, validate, and transfer said solution. In this context, defining usable and useful is critical. Usable in it’s base form denotes if a solution can be used for its purpose, and to what extent. Useful denotes to what degree a solution delivers value during said use, and to what extent. A significant part of “proving” that your solution is good or not involves measuring if it is usable and useful. This will become apparent later in this chapter.

One way to develop and transfer research results and solutions to industry is to use co-production as a collaboration approach. The term *co-production* refers to the collaborative work of researchers and practitioners in industry to identify challenges, and devise solutions that can be used in practice (Sannö et al., 2019). Co-production has many origins and can be associated with for example action research (Rapoport, 1970; Hult and Lennung, 1980). However, co-production for the purposes of this chapter is more of a macro framework of research methods in which many other micro-methodologies (e.g., case studies, action research, experimentation, etc.) can be used in combination to achieve co-production.

This chapter is centered around a co-production model that was devised, tried, and refined over a period of fifteen years at the Software Engineering Research and Education Lab (SERL-Sweden) at Blekinge Institute of Technology, Sweden. The co-production model and experiences associated with its use represent deep and long-term co-production with over thirty companies, many of which are still active partners in SERL (Gorschek et al., 2006; Wohlin et al., 2012). We also provide some general lessons learned on knowledge and technology transfer and publication strategy.

2 The Three Generations of Software Engineering Research

The **first** generation of software engineering research was greatly focusing on developing and establishing theoretical underpinnings of the discipline. Since software engineering originates from computer science, many early advancements originate from computer science research and offer increased understanding of challenges in developing and maintaining large software systems (Naur and Randell, 1969). Later on, software engineering continued to grow and develop theories and models in many sub-areas, e.g., software architecture and decomposition (Parnas, 1972) or management of software projects (Brooks Jr, 1995). Experimentation and lab validation dominate in the first generation of software engineering, supported by experience reports and essays about challenges and methods for managing large software projects.

The **second** generation of software engineering research broadens the research topics within software engineering and the empirical methods used to work with these topics. For example, software engineering researchers start to use interview studies and grounded theory as methods for qualitative data analysis and reasoning (Robson and McCartan, 2016). Prominent authors offer guidelines how to conduct experiments (Wohlin et al., 2012), case studies (Runeson et al., 2012), surveys (Punter et al., 2003) or even construct theories in software engineering (Sjøberg et al., 2008), see also Chapters 3,4 and 5. Detailed guidelines on what method to use in what circumstances are offered by Lethbridge et al. (2005), supported with guidelines of using advanced statistics in software engineering (Arcuri and Briand, 2011). What characterizes the second generation of software engineering research is a lack of long term commitment from the studied company and a discrete nature of the studies. Phenomena are often studied under a limited period and proposed solutions are rarely deployed to studied organization and used without the assistance of the researchers.

3 The Model for Co-production in Software Engineering

The seven steps described below are not necessarily sequential or determinant in order or effort spent. Rather for the purposes of clarity presented in a chronological manner. The focus of the chapter is also on the co-production itself and not the overall research methodology. That judicious application and use of research methodologies is assumed in every step, but not detailed more than necessary for the purpose of describing co-production, see Figure 1.

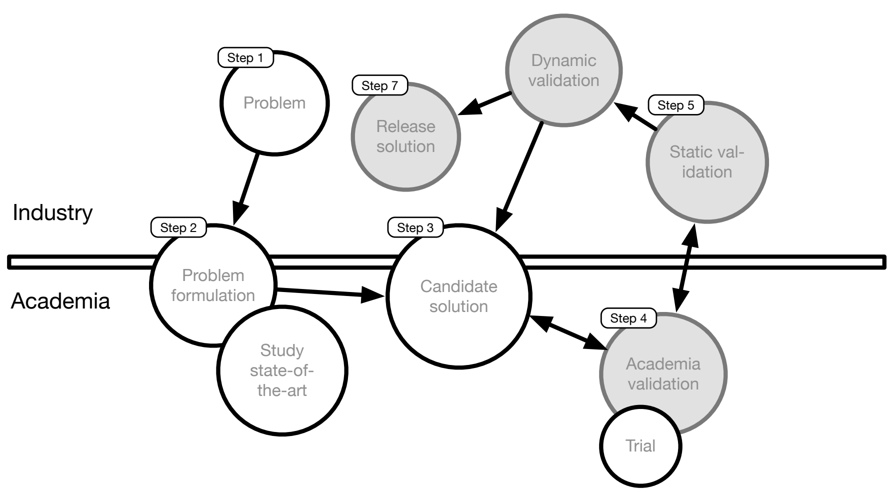

Fig. 1 The steps of the Industrial Co-production in Software Engineering.

3.1 Step 1: What is the problem?

Never, ever, use the word "problem". Industry partners have challenges and opportunities for improvement. The first thing to remember is that companies are people, people in groups have politics and culture. Be aware of politics, learn and respect the culture, and make allies.

Establishing contacts, selling yourself and the research collaboration, and getting access to a company that is willing to work with you can be seen as *Step 0*. *Step 0* is not really covered in this chapter focusing on the co-production. However, it can be as simple as giving an invited seminar on research and challenges overall, to spark the interest of engineers and managers listening.

One overarching thing we should never forget, and which can be utilized with great success is that a researcher is not an employee, nor a consultant. This might make access and the way into a company harder, but it also gives you the element of trust. A researcher can capitalize on being an external, unbiased party, that focuses on solving problems and supplying solutions, rather than developing products or working at the company in question. In addition, a researcher's work follows a set of rules (ethics and methodology), and any results are scrutinized by not only the industrial partner but also peers (peer-review). Handled correctly this can give credibility that can be leveraged for attention and trust during a collaboration between you as a researcher, and your industrial partner(s). The assumption for the purposes of this chapter is that you have established the relationship with your industrial partner, and at least secured the commitment to start working together in a broad area.

Step 1 (see Fig. 1) is about identifying potential improvement areas (challenges) based on industry needs. In its purest form, this is done by performing a process assessment or exploratory investigation of some sort. This phase consists of four main activities.

3.1.1 Activity 1: Select an overarching area or direction

It is impossible to assess an entire organization from all areas and perspectives. The size and possible findings are too great. Moreover, you as a researcher have a research focus (requirements engineering, testing, architecture, and so on). This is probably the first delimitation in area. Further refinement and narrowing can be relevant and beneficial depending on circumstances. Initial workshops with your industrial partner to discuss areas that require improvement can also be a good way forward. The challenge here is to make sure you do not select what to do at this point. The company telling you what to do exactly often leads to working on symptoms rather than actual cause/effect items. Symptoms are not the same as long-term challenges.

3.1.2 Activity 2: Plan and execute the "assessment"

Like all terms and concepts subject to fashion, process assessment and improvement (SPI) is by today's standards seen as old. However, the core methodology supplied by SPI frameworks like CMMI or uniREPM (Chrissis et al., 2003; Svahnberg et al., 2015), or even the original PDCA (Edwards, 1986) and QIP, are still very much valid and useful. The purpose of the assessment is threefold.

(1): The assessment should be planned and anchored at an appropriate level of the industry partner organization. How long will it take? What is your population and sample (what (selection) of products will you study, what roles and selection of practitioners having this role will you interact with, and so on) of your study? What methodology (see below) will you use and why? This needs to be planned and written down and agreed upon before you start. Typically, you will have a reference group or contact person at the company that opens doors and advises you. In order for this group or individual to help you, it is your job to be professional and offer all the "sales support" this person needs to get the resources and access needed for the assessment. A plan is such a thing.

(2): finding out what the challenges are (a.k.a. problems), how they manifest, their source both in origin and observation, their impact, and potential for being addressed. The process assessment models out there are often a combination of traditional methods used in exploring a phenomenon like case-study research (on an abstract level), and interviews, workshops, and document and artifact analysis on a detailed level. Further, the method used to analyze the data collected needs to be planned. Choose a combination of methodologies as appropriate. Remember, the assessment's purpose is to narrow down what to focus your research on, thus is a part of the research itself, and the results from such an assessment are research results in them-selves (e.g., interview results about challenges in an area, document analysis about the currently used processes and methodologies).

A critical part of the assessment is to use triangulation and root-cause analysis. Triangulation implies that you use more than one source to identify and describe challenges (e.g., documents and interviews and artifacts that show the same phenomena but from different perspectives and different sources). A root-cause analysis is also a good way to dig deeper and separate cause from symptom (e.g., "we have problems with bugs" can have a root cause of inadequate requirements analysis). Examples of dedicated research focused assessment frameworks and their use as well as result reporting can be found in (Svahnberg et al., 2015; Pettersson et al., 2008; Pernstål et al., 2012). A good practical example of triangulation and root cause analysis can be found here (Pernstål et al., 2015; Pernstål et al., 2019).

(3): as you are performing the assessment you are getting to know your partner. You meet people, more importantly they meet you. Trust is built, especially in the assessment stage as you listen, take notes, and try to understand. The relationship built here is crucial for a long-term relationship with the industrial partner. Relationships are always between you as a researcher, and one individual at a time at the partner site. Establish and take care of these relationships, you will need them down the line.

3.1.3 Activity 3: Report and collect

As the assessment is completing and analysis is underway (largely done) you need to report your findings in several ways and formats. Papers and technical reports can be two such forums. But this is not enough in any way. Most important is to give feedback to the ones that participated in your assessment. This can take the form of one or several seminars and/or workshops. Results, analysis and implications are presented, root-cause analysis is clarified and dependencies between findings are made clear. A good rule of thumb is that the audience already know what they told you. Repeating that is not relevant. Added value is in the form of your analysis and findings based on what they showed you and told you. By this stage, you as a researcher should know more about the challenges and their causes than any one of the people that participated in the assessment.

By reporting back to the same people you met in the assessment you show them that their investment has given results in the short-term (in the form of findings), and you as a researcher should also utilize this opportunity to get additional input and confirmation (or identify errors) in the assessment. A great way to do this is to ask for it explicitly as a part of the seminar/workshop. Surveys post-seminar can also be a complement.

3.1.4 Activity 4: Select

By this stage you often have a rather long list of "challenges" identified. One good practice is to ask as many as possible to prioritize the challenges identified in the assessment (perhaps as a part of Activity 3 above). This gives you and your industry partner reference group/contact(s) input as to what to focus on. Overall, there are three aspects that should dictate what challenges to focus your research around. Listed in order of importance: (1) the importance to the company or potential benefit if solved/addressed; (2) the dependencies between the challenges (you might have to solve X before Z, xor X needs to be solved together with M, and so on); (3) your interest area – the least important, but still relevant.

Running example step 1: The area for research is managing quality requirements. However, this is a large area within requirements engineering research and body of knowledge. The important part of the assessment is to understand what abstraction level and what organization we should study for improving quality requirement. A good idea here is to start from a focus group meeting with several role to discuss what is quality, what is the impact of quality on internal business and stakeholders and what benefits a company may achieve by investing in improving both the processes and product quality.

3.2 Step 2: Is it Researchable?

One of the most difficult things to do when working in co-production with industry is to separate consultancy and research. Industry partners will be very happy to have you do work for them, especially since you probably are very qualified, and cheap (your salary most likely comes from university). Overall the rule of thumb is that you do only research. Research is what you can publish (peer-reviewed venue). This chapter will not go into a long discussion about what does or does not constitute research, however, it is important that you are clear towards your industrial partner as this will come up.

The purpose of Step 2 is to delve deeper into one or several of the challenges in Step 1. This can be done in any number of ways but generally involves three activities.

Running example step 2 (is it researchable?): When discussing the area and the symptoms in step 1, you identified that a company lacks an adequate tool for the needs and education in managing quality requirements. The need to introduce a new (better) tool is highly prioritized by the company and You are asked to do that as a part of the research project. The answer is that installing and configuring a tool is no research and you should be clear in communicating that this is consultancy, maybe qualified work but not research. You should explain this to the company and move to the other challenge. For example, the information meta-model that the tool will use and the associated improvements to handing quality requirements could be a potential researchable area. Therefore, root cause analysis of the challenges is a very important step and it helps to define what is researchable and prepare conducting focused state-of-the-art study.

3.2.1 Activity 1: Problem formulation

Problem formulation involves a deepening of the root-case-analysis of the challenge(s) (see Step 1), but also breaking the challenge(s) into actual research problems. In its simplest form you want to (i) be able to describe the context of the challenge, (ii) the area(s) of research/study, and (iii) either identify a number of research questions or potentially refine into a hypothesis. The end result is a research direction with concrete research questions to answer.

3.2.2 Activity 2: Study state-of-the-art

Study state-of-the-art is often (to some extent) done in parallel with Activity 1. However, a more detailed and systematic review of state-of-the-art is often a part of reading up on a subject, positioning the problem formulation/research (from Activity 1) to other research and being able to explain how your research will contribute to the state-of-the-art. An example of a systematic literature review (SLR) done in conjunction with Step 2 can be found here Pernstål et al. (2013).

The literature review helps you to systematically obtain evidence about the area of interest and formulate conclusions or recommendations based on the analysis and synthesis of the evidence found in the selected papers. The process should be transparent and traceable and decisions taken during the literature review (e.g., what papers to include) should be clearly documented. Chapters 12 and 13 offer more details about systematic literature reviews.

A critical lesson in Step 2 is not to lose touch with your industrial partner. Doing a one-year SLR is a very bad idea. You need to keep in touch with your industrial partner, as well as present, discuss and validate your ideas and findings, even from this step. A rule of thumb is that Step 2 should not take more than 2-3 months of calendar time to not risk the industrial partner moving on. Another good idea is to find a recent SLR in the area of interest that contains a summary of the core papers in the area and perform snowballing on these articles.

Our experience shows that a faster systematic literature review (narrow and efficient methodology applied, see e.g., snowballing Wohlin (2014)) is preferable. The purpose here is not to be perfect, rather to gauge; are the challenges and subsequent problem formulation and breakdown likely to result in new research that complements what is already out there? Other chapters provide additional information around searching grey literature as so called multi-vocal literature reviews (Chapter 14), and rapid reviews covered in Chapter 13.

If the answer is yes, then you move on to Step 3 of the co-production model as fast as possible. If the answer is no on the other hand move to the next challenge(s) on the list. An overall tip for Step 2 is that even if much of the work is reading, you can do the reading at the company partners' offices. That way you are part of the environment, can talk to people, and be seen.

Selecting and detailing a research area is the topic for whole books and will not be covered here in any detail in this chapter. However, type of contribution is relevant for the discussion. The base question that any engineering scientist has to ask is; if a research area is well established and significant research has been conducted, why does my industrial partner have challenges/problems in this very area? The answer can be that the company simply is unaware of the many "solutions" available. If this is the case, then a very good idea is that you, as a researcher, help facilitate the transfer and adaptation of one or several already existing "solutions". This can be a research contribution as you can do research on the adoption and subsequently measure the usability and usefulness of a solution, even if it is not new or yours. There are many benefits this, not least presenting a solution to the partner company very fast. However, many, if not most, solutions out there (even if well published) have rarely been extensively tried in industry. However, let us say that the solution has been validated in industry, thus it appears to scale and show utility. Even then it most likely needs modification and adaptation to fit and be a solution to your partner company's challenges. All this is very relevant and publishable engineering research in itself.

Running example step 2 (selecting an area): The challenge is "quality requirements" that lead to later misunderstandings (in the company resulting in requirement-based defects found late in product development)". The root-cause shows (among other things) (i) inadequacies in specification where quality requirements are often neglected, and (ii) inadequacies in communication in relation to requirements between practitioners, roles, departments etc. You start reading up and find a lot of research in both of these fields (hundreds, if not thousands, of papers and methods/tools etc. already published). On this level, it seems like it will be hard for you to contribute. Two choices exist. One, you can move on to another area where less research is done and it is easier to contribute, or two, you can delve deeper and see if you can contribute even if the area is well researched. Maybe the problem is in decision making about quality requirements? If so, you should study how a requirement is decided upon since its inception (both quality and functional), what roles and involved, what artifacts are produced and who prepares, takes and influences the decisions made about these requirements.

Outcomes from Step 2 result often in a hybrid. There are often many potential solutions, but few are actually validated and tested in a real environment (shown to be usable, useful and scale). These solutions can, however, act as input and inspiration to you in your work to solve your partner company's challenges. This is all very good research practice if done correctly.

3.2.3 Activity 3: Formulate a research plan

Once you have an idea of what research to start with, whether it be to develop something completely new, use existing research, or a hybrid, you need to develop a plan for the next steps. These steps are basically the same as the ones described in this co-production model.

3.3 Step 3: Solution or not solution?

The co-production model has a rather large step between Step 2 and this Step 3. Step 3 is the "do the research" step. From Step 2 we should have an idea of research direction, area, and details on the initial parts of a plan. This research could be further empirical studies (so more investigation deeper, and not a “solution”). Studying a phenomenon in industry is relevant, valuable and publishable.

Industry partners can be shown that further study can be directly beneficial to the company and provide direct improvements. However, even after further study, some sort of solution will be part of the research and change the nature of the research activities from descriptive (what we can observe and what are the challenges or problem associated with the phenomenon we are exploring) to prescriptive (can we develop a way to improve the current way of working). For the purposes of this chapter, we will focus on creating and delivering some sort of solution.

At the core of any solution is a continuous collaboration with industry partners while developing the solution. This is critical. Sitting at university for months, then visiting the company to show them what you have is most often not a good idea. What you want is to involve company practitioners in the work. This can take the form of formal workshops, work sessions, meetings and so on Santos and Travassos (2009). However, informal discussions are at least as beneficial and important. Sit at your partner company site. Discuss your research and ideas during a lunch break. This is a great way to get input and also build trust and acceptance of any solution you eventually come up with, as it was “made here” at the company site, with constant input and feedback. This can be seen as a type of action research Pernstål et al. (2015).

Running example step 3: : You came to a conclusion that a a new way for the partner company to specify and communicate quality requirements that could hold the potential to remove significant amounts of misunderstandings (usefulness), but at the same time not increase level work needed or formality (usability and scalability). You call this way of quality requirements specification/communication “QRImP”. You are sure that the improved communication part is a way to improve decision making about quality requirements since it will provide improved rationale for decisions and help decision makers to perform better work.
QRImP was developed based on challenges identified (Step 1), it was refined and formulated for research (Step 2), and you developed a solution in the form of QRImP (Step 3), in collaboration (many coffee machine discussions, workshops and meetings) with many practitioners and also colleagues at university over months of work. Now what? Well, what you actually have developed is a candidate solution that needs to be “tested” and improved. This is described in Step 4 through Step 6, as these steps are validation and improvement incrementally within each steps and evolutionary as the steps progress.

3.4 Lab-based (static) validation and improvement in academia

Most of the time Step 3 (developing the candidate solution), Steps 4, and 5 happen simultaneously, or at least overlap to some extent. For reasons of clarity, we describe the steps in sequence. Step 4 is validation in academia. You devise a "solution" of some sort, whether it be a method, model, framework, way-of-working, organizational improvement, tool or equivalent, and try it out on so called "toy examples" and/or use students as subjects to try it out. Albeit being a smart way to do initial validation of a solution, it also has many inherited flaws, inconsistencies and scalability problems.

Many researchers critique this type of lab-based scaled-down validation as it does not represent an industrial scale, context or application using practitioners subjected to time and resource pressure. Our experience is that utilizing scaled down scenarios and/or students or even fellow researchers as subjects is beneficial. Any use of industry resources poses a cost and disturbs the real production environment. This should be avoided if non-production environments can be utilized to catch the same incompatibility with the candidate solution. Wohlin et al. (2012) offers a comprehensive overview and guidelines how to plan, conduct and report experiments in software engineering.

Doing a short experiment with colleagues, or students, in a controlled environment can not only catch items that simply don't work, but you can also measure usability and usefulness in an accurate manner as you can do for example a controlled experiment and compare your candidate solution to another way of working.

Lab-based validation is one step and not the end of validation, despite evidence to the contrary looking at peer-reviewed papers in many fields (where an experimental validation is sufficient for publishing in many top venues). However, since software engineering is an applied engineering science, ending at the lab validation stage is especially troubling. The only reason to end at the lab validation step in software engineering is if the solution fails so badly that there is no reason to carry on.

The data collected during lab validation can and should be used to improve the solution. In many cases, this implies simplification of the solution and making it less useful and more usable. This might be contrary to your training as a scientist and engineer; however, it is wise to remember that a perfect solution that no one uses is a meaningless solution from the company's point of view.

Lab validation is a perfect opportunity to try to balance usefulness and usability as well as scalability of a solution. Does it take the students two hours to specify and coordinate one requirement? Well, if your industrial partner has a hundred requirements to handle per year (e.g., slow system evolution, safety critical applications for example), your solution may be acceptable. However, if your industry partner has thousands of requirements per year, which change constantly as competition and market circumstances evolve, then they will be lucky to have five minutes to spend on a single requirement during initial specification and communication.

A notable problem inherent in software engineering research is that problems can be very complex, but any solution has to be simple. This might sound obvious, but publishing "simple solutions" might not be that easy as reviewers do not see the entire evolution. The contribution of creating and refining a solution that is usable and useful in industry and used for real requires a lot of work and time and often does not result in many more publications. The point of taking the next step is working towards refining and transferring the solution to reality and measuring how well your solution works; not inventing new solutions or more advance concepts.

A good indication of when to stop/move on from the lab validation is when you cannot learn more without adding the reality of industry to the validation (realistic scale, real practitioners as subjects, industry partner context and limitations). The limited and scaled down nature of lab validation can, however, be used to build a case for your industry partner.

During lab validation, the solution and instrumentation are refined in continuous collaboration with the company. The interaction with the company is realized in seminars and presentations of results from the lab validations. Step 5 often overlaps with step 4 as the focus is on building the trust of the company in the developed solution. The main potential is to build trust. Lab validation not only enables you to improve the solution but shows this to the industry partner – you are doing all you can to improve a solution that was co-produced with and by them, without wasting their time and resources. It also minimizes risk for the partner, as major problems associated with the solution can be caught before they invest in the next step.

Running example step 4: How well do the QRImP specification and coordination method work in comparison to another established way to achieve the same task? How can we measure that the effort dedicated to introducing QRImP would field expected benefits? How large should be the group of participants (roles, availability) so we can objectively measure the effects of introducing QRImP. How many quality requirements should be specify and decide upon with the help of QRImP? How to measure improved communication and narrowing some communication gaps with the help of QRImP?
When you develop the instrumentation for the lab-based validation you get a good idea on what is needed for later industrial validation in terms of training, manuals, templates, tools etc. to support and enable the use of QRImP. For example, you can measure the quality of requirements specified with the QRImP template, the communication issues (delays, decisions that need to be reversed) or other negative effects to see the effect of introduction of QRImP.

3.5 Step 5: Static-validation in industry

Static validation is often characterized by trying out your solution on a limited scale. The subjects are real industrial practitioners (you need to think about population and sample in relation to the future potential users of the solution), however the time and scale of application are limited.

Static validation is often a collection of activity steps and a progression towards more realistic scenarios. Below examples of this are described based on real cases in research. Please observe that these examples should be adapted to fit your contextual characteristics and case specifics. The purpose of the static validation is to get input to refine and improve the solution to the extent that the partner company wants to try it in their real production environment. Static validation is conducted at several levels described below:

Level 1: Workshop Sessions. This typically denotes dedicated work sessions where you plan and call a meeting. A handful of practitioners come, and you present your solution, give them initial training and tools (from nothing, a simple template, to an actual tool, depending on your solution), and let them try out the solution using real data (e.g., requirements in our example). Data collection here is both direct and indirect. You can measure things like task completion, defects, time to get proficient and so on.

Direct measurement is often done through interviews and/or a survey after they tried the solution to collect tips and views as well as their judgement. Direct measurement should not be discounted as not only can you get positive and negative (what worked, what did not), but also a comparison of your solution to the way the work today – a relative judgement. During the session, actual observation and light-weight logging Lethbridge et al. (2005) of behavior and work can be useful as a data collection tool in addition. A tip is to have one or two supporting scientists in these sessions. The main scientist is typically the workshop leader and will have a hard time observing and taking notes.

Level 1 is often carried out with multiple groups (as large groups are hard to observe; thus you might replicate the "study" several times. Weather you choose to change the solution (and/or instrumentation) between iterations is context dependent and presents pros and cons either way. Our experience is that the validity threats introduced (or sample size reduced as if you change instrumentation or perform several smaller studies) is preferable than running studies with a solution and/or instrumentation you know should be updated.

Level 2: Light-weight production. During workshops (where you optimally got a lot of input utilizing several methodologies) the solution is refined either iterative or in batches. Once you judge that more workshops will not yield different results (this can be observed via the data collected) you can move on to static validation closer to the production environment. This involves joining an actual production team on an appropriate level that works with areas relevant to your solutions utilization. The mandate to allow this comes from the issue that some of the same people participating in the previous workshops are the same production team/practitioners you now join. They use your solution to do their job. You as a researcher are on hand to support them and to some extent compensate for the learning curve of any new way-of-working. In this environment, it is much harder to be an external "observer" and collect data in an objective manner as you become embedded into the team. However, debriefings and observations are still important and possible. It is also possible to combine post-work interviews or surveys as a complement. This is still an activity to collect data to be used as input for improving your solution.

The practitioners' goals are aligned (they have a vested interest to really see if the solution works for real), but their primary goal is to do their work. Thus, any data collection is up to you in this phase especially as generally you are not supported by other scientists. The production environment is typically sensitive to disruption, and you as a scientist should embed yourself into the term to minimize distractions and biases. Having extra scientists present is typically impractical as a larger contingent of scientists disturbs the production environment even more than one.

It should be observed that static validation is very costly for an industrial partner. Person hours of the practitioners participating is only one part, the real cost for the company is best alternative investment. That is, what they could have been doing in their production environment, and all subsequent effects of this downstream are cancelled to work with you and help validate the solution. Often researchers forget that there is almost never any slack in an industrial production environment, thus it is always a trade-off. This insight should inspire a researcher to be very well prepared and maximize every opportunity for data collection aiming to improve the solution.

A central delivery of Step 5 is significant amounts of data on the usability and usefulness of the solution, in addition to the evolution of the solution during the static validation. This is critical from a research perspective, but also for the industrial partner in preparation for the next step. Step 5 also brings significant educational value since the participants gets improved understanding of the method and potential benefits when implemented in the organization.

Running example step 5: After receiving encouraging positive results from the QRImP lab validation (step4), you move on to static validation with industry (step 5). Here a large part of the instrumentation developed for the lab validation can be reused, together with the description of the desired participants and their required levels of expertise and availability. You then plan and execute a workshop session and invite the roles you decided are relevant and should use or be impacted by QRImP. You give the group some initial training in the method (with many simple examples), followed by a request to take some requirements from their environment and work with them using QRImP.

It is important to be at least 2 persons in the room so one can moderate and make sure it progresses according to the plan and the second person takes notes and collect any other relevant data. It is wise to run more than one workshop with two separated group and compare the results. Collect comments and improvement suggestions during the workshops to tune and improve QRImP. Ask for reasons and rationale for improvement to make sure that the improvements are harmonized with the overarching goal of the method

3.6 Step 6: Dynamic validation in industry

The goal of dynamic validation is to let industry use the solution uninterrupted by researchers. In this step there should be no action research, if you need to be on board you are still in the static validation step. Dynamic validation is critical as the true success of any co-production effort and solution is if an industry partner and the practitioners use the solution after the researchers leave the organization. Dynamic validation can be divided into three parts, and you can and often have to iterate within the activities.

3.6.1 Activity 1: Sell-in, buy-in, finding the right production instance

If you have worked closely with the industry partner throughout Steps 1-5, the move towards trying out your solution in a production instance (project, iteration, etc.) will be a natural next step. In some companies, there will still be a need for you to present your case, and to sell the usability and usefulness of your solution to middle-/senior management before they invest in dynamic validation. Again, the risk for the company at this stage is greater than in any other step. You need to train practitioners in using the solution, supply manuals, tool support, maybe even transfer data from other systems. This is a part of your preparation for dynamic validation.

Do you remember under Step 2 we had a discussion about the distinction between research and consultancy? The delivery of a tool (might be a commercial one supporting your solution), the training of using the solution (using the tool), and so on is consultancy (or at least not research). However, this is completely acceptable and mandated given that it is in preparation for dynamic validation of your solution. Thus, what is consultancy and what is not depends on the intent and context. The preparation for the dynamic validation is mostly on you, but the subsequent validation is all on the industry partner and the practitioners. Optimally you should not even be present in the environment at all.

3.6.2 Activity 2: Data Gathering

What is the research part of Step 6? Mostly preparation. You should instantiate a measurement program since you want to measure the usability and usefulness of the solution in a completely real use scenario in a real live production environment. In dynamic validation, practitioners use the solution without the researchers being there or being able to help, or do any changes in how the practitioners use the solution. The measurements you introduce need to collect as much data as possible pertaining to usability and usefulness with as little (close to zero) interference from the measurement being done. Any measurement interference in “normal” operations is a threat to validity and also constitutes a confounding factor Wohlin et al. (2012); Feldt and Magazinius (2010).

Exactly how measurement is done varies depending on your solution, the production environment and context of use Petersen and Wohlin (2009). For example, if there is tool support measurement on use, task completion etc. can be collected. If artifacts are produced (e.g., in our example, requirements) they can be tagged and saved in as a granular versioning/variant manner as possible. If other artifacts (code) result from the main solution artifacts (requirement) these are saved for later analysis. Going against the no interference policy, you can do some direct measurement, especially if use and artifact-based measurements are not possible.

For example, one of the practitioners can be the "responsible" for the new way-of-working (the solution). This person can then collect data as a part of this job. You can also present surveys (short and fast ones) in the use process. However, please observe that the practitioners not only have their jobs to do, but you are asking them to change how the work, and learn a new way (solution) at the same time. Any additional parts are pressuring them further during the production instance. After the production instance has ended (and the dynamic validation round has ended) you can do more direct measurement in various forms.

3.6.3 Activity 3: Data analysis and data reconstruction

As a validation round ends and data is collected, analysis of the impact of applying the solution is the focus. Measurements in relation to usability and usefulness are key, however, there are complementary parts. Improvement ideas and deal breakers (what was bad) are gathered through e.g., interviews and workshops. This should be done as fast as possible with as many as possible that were using the solution in the production environment.

The data collected and analyzed can result in one of three outcomes:

(i) the usability and usefulness are too low compared to previous ways-of-working. If this is the case you can either abandon the solution (that does not mean that it was bad research), or you can reverse back to Step 3 and re-design your solution.

(ii) the usability and usefulness are higher than the previous way-of-working. The company decides to progress to Step 7 (covered later in more detail).

(iii) the solution is promising, but needs work and improvement and more validation before it can move on to be used for real in the organization. This third outcome is the most common. Outcome one is the least common given that you have refined and progressed with the solution using co-production, and large surprises should have been caught earlier in the lab and static validation rounds.

Step 6 is often executed several times for several reasons. One of the reasons is that after the first static validation round you have tested the solution in one production environment only. Companies are heterogeneous in terms of what they do, and how they work. Thus, the solution might need several variants (instances) at different places to be eligible for more dynamic validation rounds. Dynamic validation completion depends on two goals. First, does the company have the confidence to move on to Step 7, and second, have you as a researcher exhausted the potential for data collection in relation to the solution (short-term).

Running example step 6: You need to start from creating a summary presentation of the results from the static validation that you can use to find the production instance that would be interested in continuously using QRImP. It is important here to remember the goals and the challenges and associate the results from static validation as clear indication that we are on the good path towards intended improvements. At step 6, you need to work intensively with sell-in activities but also remember what is researchable (please see step 2 for details).

For example, as a research you should not develop a tool unless it is a step towards gathering data for dynamic validation and later handing over the tool management to the studied company. Based on the validation in academia and static validation, you should establish a measurement program and decide about the frequency of data collection (how often would you measure if QRImP helps with managing quality requirements), and how often the results will be shared with the management team. You should also appoint a person that can be "responsible" for introducing the new "way-of-working" and making sure it is not forgotten or neglected.

3.7 Step 7: Solution Release

A company progressing to Step 7 is committing to using the solution without the assist or influence of the researchers. Your role in the end of Step 6 is of course significant research reporting (peer-review), but also significant reporting (mostly in the form of discussions and presentations) to the company on multiple levels. This gives the decision makers decision support input as to convincing management and themselves to adopt the solution.

The institutionalization of the solution in the company is not really part of the research, however it is in your interest to be supportive of these efforts for two main reasons. The future collaboration and research with the company is dependent to a degree, but more importantly, the ability to get access to long-term study of the solution post-release. In Step 6 we did try out the solution in a real environment, however, the long-term effects of the solution on the company (people, teams, products, and other artifacts) is largely unknown.

In our example, improving requirements communication might result in e.g., improved team coherence (soft factor), and improved ability to test customer value (hard factor). By helping with the company with instances (not doing it all but helping) you might be able to get them to let you also instantiate some sort of measurement program alongside it. This will enable you to monitor and collect data on a complete organizational scale, continuously. This can easily be motivated to the company (input for them too) when the measurement program is reasonably non-intrusive and low cost. The ability to perform such longitudinal studies Sjoberg et al. (2007) is as rare as it is a great opportunity. Step 1-7 completed implies you actually start over with a new set of challenges.

Running example step 7: You should work mostly with packaging your research results and the artifacts created so that the person "responsible" can smoothly take over. A significant amount of effort you should also decide to creating educational material about QRImP (recording lectures with examples from the company, slides, and supplementary material). Lastly, you should present the results from dynamic validation to several roles and levels of the organization to raise awareness and enable more easy technology transfer.

4 General Lessons Learned

This chapter presents some general lessons learned and experiences from applying the presented model. We focus here on clarifying what means a delivery and how the developed solution should be treated by both the researcher and industry. We also clarify how to establish an effective publication strategy when working with the model.

A word on Technology and Knowledge Transfer. The actual solution creation and subsequent refinement through validation is the visible delivery. However, just as much knowledge and improvement is "transferred" as an indirect effect of the co-production and collaboration work itself. The researcher learns about the industry partner's context, products, challenges, technologies, ways-of-working and more. At the same time practitioners at the company learn more about themselves and thus can change and improve things independent of the actual solution. Practitioners also learn things associated with the work. For example, training practitioners to try the solution out will improve their knowledge in the application field (become better at specifying requirements and/or communication and coordination in our example). This effect is hard for a researcher to measure. Purists will say that this in itself constitutes a confounding factor and validity issue. Engineering scientists say; any positive influence on state-of-practice is a good thing and our job.

A word on publications and papers. As you might have noticed very little has been voiced about "studies" and "papers" during the description of the co-production model. Working in real co-production and trying to base solutions on real problems and then devise solutions that eventually can and will be used in a real production environment without you as a researcher is very hard, prone to failure, and relatively speaking, harder to publish. You don't do studies and then a paper or two (a.k.a. "hit and run" research). You don't listen to industry, devise a solution that you apply on a toy example or subject students to and then stop and write paper after paper (toy level validation). If publications are your main goal, I would stay away from co-production in any real sense. However, if you want to work with industry and earn a status as an expert "problem solver", then co-production is vastly rewarding. As a researcher, you very seldom get to see the results of your work beyond some peer-recognition and citations. Seeing something you helped create working without your assistance, being used by strangers ten years after you released in a company is the real reward. If the practitioners using the solution have renamed it, use it in a different way than you originally intended, and don't know who you are, that is success.

All of this said, of course you can publish. Maybe you will not churn out papers at the speed of changing statistical methods on the same data set of open source data, but your papers will be much more relevant and contain real data and report on solutions you measure. We would like to avoid detailing the reporting part, but since younger researchers might read this chapter, we wanted to share examples of how you can plan out your publication parts while working in a co-production scenario.

Tips for reporting industrial co-production in software engineering
STEP 1: Finding challenges is, in essence, a detailed and deep empirical study of one or several macro cases (companies) and/or micro cases (dept. or divisions in a comp.). Real challenges, their descriptions and actual origin are in themselves research results. This especially if you apply a rigorous assessment methodology.
STEP 2: Investigating state-of-the-art, doing detailed root-cause analysis, and combining this with empirical data is relevant for publication. The quality of the publication depends on the scope and rigidness of the state-of-the-art investigation.
STEP 3: Formulating a candidate solution is often very interesting to convey in a position paper, spanning everything from a good workshop where your ideas to can be discussed, to actual heavier publications – depending on the novelty and potential of your candidate solution. Remember, you are basing its creation on input from industry even if the solution is not validated yet.
STEP 4: Lab validation can result in any number of publications, from student experiments to simulations and details on the solution. Refinement evolution based on lab validation is also interesting and relevant to base in a publication(s).
STEP 5: Static validation is often very well anchored, and you collect data from actual industrial practitioners in several iterations and phases of solution evolution. You also measure real usability and usefulness in this step, all relevant for publication.
STEP 5 and 7: The data collected on the use, usability, usefulness of the solution, and impact in industry is well worth publication, often resulting in several heavier papers.
OVERALL: As you progress through the steps you will notice that you might need to develop macro contributions so that you can move on and validate and improve the solution itself. These "macro" contributions can, for example, be: new ways of assessing a company (Step 1-2), new tools and tool support for the solution, new ways to measure usability and usefulness of your solution, and so on. All of these are also relevant in their own right and viable for publication.

5 Recommended Further Read

Wohlin et al. (2012) outlines the success factors powering the industry-academia collaboration. The authors highlight the usefulness of 14 factors, e.g., collaboration champion on site, buy-in and support from company management, researcher's visible presence and regular meetings.

Svahnberg et al. (2015) offer a framework for requirements engineering process assessment that is lightweight and offers comprehensive overview of the maturity level of an organization. The assessment brings practical examples of how to plan questions in various areas of software engineering, how to collect the answers and prepare the report to the companies.

Pettersson et al. (2008) offers detailed guidelines how to perform lightweight process assessment and improvement planning. It is the initial work that led to the creation of uniREPM Svahnberg et al. (2015) and brings practical viewpoints of process assessment and improvement.

Pernstål et al. (2012) offers a method for root cause analysis when performing software process improvement activities. Flex-RCA method for root cause analysis is used to delve deeper into challenges identified to find root causes as a part of the evaluation and subsequent improvement activities.

On the methodological stance, Wohlin (2014) outlines how to conduct snowballing literature studies in software engineering as an alternative to database search systematic literature reviews while Lethbridge et al. (2005) lists what data collection techniques to use in what empirical study in software engineering. They divide the techniques into the first-order techniques where the researcher has a direct context with study subjects and second and third-degree techniques where researchers study artifacts produced during software engineering activities.

Wohlin et al. (2012) offers a comprehensive overview how to plan, conduct and report experiments in software engineering. The authors provide detailed guidelines with examples about how to plan an experiment, select dependent and independent variables, subjects and objects as well as instrumentation.

6 Conclusion

The co-production model presented here is one way of achieving co-production. The idea is not to follow it like a blue-print, but rather use it to get inspiration and understanding of the incremental and also step-wise nature of building a solution, but also in building trust and commitment from your industrial partner. There are many downsides to working in close collaboration with industry. The academic reward systems are not really gauged to reward success that is actually defined as making a real difference in reality.

The number of publications, H-index, funds acquired, doctoral students supervised, courses thought, and peer-recognition can all be good but do not necessarily have anything to do with co-production. So why do we do it? This is a long conversation and we can only speak for ourselves. We are engineers. We love solving problems. As researchers, we get to try to solve really tricky and complex problems. As engineering scientists this means solving complex problems by devising solutions that actually work in an applied setting. Not in theory, not maybe sometime in the future.

Acknowledgements We would like to thank all current and past members of SERL and collaborating companies for inspiration in writing this chapter.

References

Arcuri, A. and L. Briand (2011). A practical guide for using statistical tests to assess randomized algorithms in software engineering. In *2011 33rd International Conference on Software Engineering (ICSE)*, pp. 1–10. IEEE.

Brooks Jr, F. P. (1995). *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E*. Pearson Education India.

Chrissis, M. B., M. Konrad, and S. Shrum (2003). *CMMI guidlines for process integration and product improvement*. Addison-Wesley Longman Publishing Co., Inc.

Edwards, D. W. (1986). Out of the crisis.

Feldt, R. and A. Magazinius (2010). Validity threats in empirical software engineering research-an initial survey. In *Seke*, pp. 374–379.

Gorschek, T. (2018, February). Evolution toward soft(er) products. *Commun. ACM 61*(3), 78–84.

Gorschek, T., P. Garre, S. Larsson, and C. Wohlin (2006, Nov). A model for technology transfer in practice. *IEEE Software 23*(6), 88–95.

Hult, M. and S.-A. Lennung (1980). Towards a definition of action research: A note and bibliography. *Journal of Management Studies 17*(2), 241–250.

Lethbridge, T. C., S. E. Sim, and J. Singer (2005, Jul). Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering 10*(3), 311–341.

Naur, P. and B. Randell (1969). Software engineering: Report of a conference sponsored by the nato science committee, garmisch, germany, 7th-11th october 1968.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM 15*(12), 1053–1058.

Pernstål, J., R. Feldt, T. Gorschek, and D. Florén (2019, Mar). Flex-rca: a lean-based method for root cause analysis in software process improvement. *Software Quality Journal 27*(1), 389–428.

Pernstål, J., A. Magazinius, and T. Gorschek (2012). A study investigating challenges in the interface between product development and manufacturing in the development of software-intensive automotive systems. *International Journal of Software Engineering and Knowledge Engineering 22*(07), 965–1004.

Pernstål, J., R. Feldt, and T. Gorschek (2013). The lean gap: A review of lean approaches to large-scale software systems development. *Journal of Systems and Software 86*(11), 2797 – 2821.

Pernstål, J., T. Gorschek, R. Feldt, and D. Florén (2015). Requirements communication and balancing in large-scale software-intensive product development. *Information and Software Technology 67*, 44 – 64.

Petersen, K. and C. Wohlin (2009). Context in industrial software engineering research. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pp. 401–404. IEEE.

Pettersson, F., M. Ivarsson, T. Gorschek, and P. Öhman (2008). A practitioner's guide to light weight software process assessment and improvement planning. *Journal of Systems and Software 81*(6), 972–995.

Punter, T., M. Ciolkowski, B. Freimut, and I. John (2003). Conducting on-line surveys in software engineering. In *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings.*, pp. 80–88. IEEE.

Rapoport, R. N. (1970). Three dilemmas in action research: With special reference to the tavistock experience. *Human Relations 23*(6), 499–513.

Robson, C. and K. McCartan (2016). *Real world research*. John Wiley & Sons.

Runeson, P., M. Host, A. Rainer, and B. Regnell (2012). *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons.

Sannö, A., A. E. Öberg, E. Flores-Garcia, and M. Jackson (2019). Increasing the impact of industry–academia collaboration through co-production. *Technology Innovation Management Review 9*(4).

Santos, P. S. M. d. and G. H. Travassos (2009). Action research use in software engineering: An initial survey. In *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pp. 414–417. IEEE Computer Society.

Sjøberg, D. I., T. Dybå, B. C. Anda, and J. E. Hannay (2008). Building theories in software engineering. In *Guide to advanced empirical software engineering*, pp. 312–336. Springer.

Sjoberg, D. I. K., T. Dyba, and M. Jorgensen (2007). The future of empirical methods in software engineering research. In *2007 Future of Software Engineering*, FOSE '07, Washington, DC, USA, pp. 358–378. IEEE Computer Society.

Svahnberg, M., T. Gorschek, T. T. L. Nguyen, and M. Nguyen (2015, Mar). Uni-repm: a framework for requirements engineering process assessment. *Requirements Engineering 20*(1), 91–118.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pp. 38. Citeseer.

Wohlin, C., A. Aurum, L. Angelis, L. Phillips, Y. Dittrich, T. Gorschek, H. Grahn, K. Henningsson, S. Kagstrom, G. Low, P. Rovegard, P. Tomaszewski, C. van Toorn, and J. Winter (2012, March). The success factors powering industry-academia collaboration. *IEEE Software 29*(2), 67–73.

Wohlin, C., P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén (2012). *Experimentation in software engineering*. Springer Science & Business Media.