



Conceptualizing the Transition from Agile to DevOps: A Maturity Model for a Smarter IS Function

Aymeric Hemon¹(✉), Barbara Lyonnet¹, Frantz Rowe^{1,3},
and Brian Fitzgerald²

¹ University of Nantes, LEMNA, Chemin de la Censive du Tertre,
Bâtiment Erdre, 44322 Nantes, France
{aymeric.hemon, barbara.lyonnet,
frantz.rowe}@univ-nantes.fr

² University of Limerick, LERO, Castletroy, Limerick V94 T9PX, Ireland
brian.fitzgerald@ul.ie

³ SKEMA Business School, KTO, Sophia Antipolis, France

Abstract. Agile development approaches have become the norm for almost all software development now. While agile approaches can deliver more frequent releases of working software, it quickly became apparent in many organisations that they were not able to leverage these frequent releases due to the disconnect between the development and operations functions, with the latter typically responsible for releasing software to customers. This resulted in the move towards closer integration of these functions through the DevOps movement. As the trend towards digitalisation continues, companies are increasingly implementing DevOps. We propose a maturity model for this agile to DevOps transition with three levels: agile, continuous integration, continuous delivery. Based on an in-depth case study in an organisation which has several years' experience of DevOps, we identify a fundamental disruption in the soft skills and competences that software teams are expected to possess, and in the patterns of collaboration among teams. The latter is especially salient for release managers, project managers, production engineers and even architects. Arguably, smartness may be characterized as being flexible, teaming up with people who have a different profile, belonging to a different function, and delivering more quickly what had been designed. In light of this, we argue that DevOps leads to greater smartness for the Information Systems (IS) function.

Keywords: DevOps · Agile · Roles · Skills · Collaboration

1 Introduction

Agile methods seek to ensure a close link between the customer and developers to ensure that software meets market needs. Agile methods also strive for a more rapid release schedule. However, while agile methods can achieve a more frequent cadence of development of software, a bottleneck has emerged in that the Operations function (Ops), who coordinate the actual release of software in organizations, are typically not

aligned with the Development function (Dev). A release could take weeks to be launched. Consequently, organizations were not able to achieve faster software releases to customers. In order to solve this problem, Debois advocated a tighter integration between the Dev and Ops functions which is termed DevOps [1–3].

Emerging as it did from practice, there is not a great deal of work thus far outlining the conceptual or theoretical underpinnings of DevOps. A similar situation occurred in the case of agile methods, which was also practice-led, and some important definitional work appeared later [4]. We believe that many companies are in the transition from agile to DevOps, and we suggest three maturity levels to reflect this: Agile, Continuous Integration, Continuous Delivery. Each level builds cumulatively on the previous one. We analyse five key job roles in DevOps (Release Manager, Architect, Product Owner, Department/Project Manager and Production Engineer) and describe the key collaborations across these and other DevOps roles. We then describe the skills for each of these roles, dividing them into ‘hard’ and ‘soft’ skills [5, 6]. We consider how soft skills (SSk) vary by level of maturity.

The paper is structured as follows. In Sect. 2, we discuss other relevant work in this area. Section 3 presents our research method and Sect. 4 our findings. Finally, Sect. 5 discusses the implications of our findings for research and practice.

2 Literature

2.1 Roles, Skills and Competencies of Information Technology (IT) Jobs

“A skill is a combination of ability, knowledge and experience that enables a person to do something well” [7]. Skills are specific to a domain and developed by practice [7]. Hard skills refer to the ability to perform, to do something well, using knowledge, techniques, practices. Robles [8] defines “soft skills as intangible, nontechnical, personality-specific skills that determine one’s strengths as a leader, facilitator, mediator, and negotiator” (see Table 1). This paper will focus on SSk.

Gallivan et al. [5] identify a “long history of IT-skills studies” which suggest that soft or non-technical skills are more important. Wong [6] also emphasises the relative importance of SSk – flexibility, adaptability, motivation and good communication. Non-technical skills include interpersonal, leadership, organization, independence/motivation, and creativity skills. However, studies found that non-technical skills were far less valued than technical skills in recruitment advertisements. They concluded that despite the emphasis on hiring well rounded employees with good business knowledge and SSk, the recruitment process focuses on “hard skills” because they were easier to screen. Thus, they confirmed the “recruitment gap” that had been identified earlier [9, 10]. This lack of recognition is harmful as some SSk have been identified as important in agile software teams [11]. In addition, Wiedemann et al. [12] investigated key capabilities of DevOps teams which could foster competitive advantage. They identified seven key capabilities (Change Readiness, Decision Making, Culture, Collaboration, Intrapreneurship Skills, Agile Project Management, IT Technical Skills, Continuous Skills). However, they mixed capabilities with skills without determining precisely which skills were a source of competitive advantage. In their study, they

Table 1. Soft Skills (SSk) as identified by Robles (2012)

•(CS) Communication – oral, speaking capability, written, presenting, listening
•Courtesy – manners, etiquette, business etiquette, gracious, respectful
•(FS) Flexibility – adaptability, willing to change, lifelong learner, accepts new things, adjusts, teachable
•Integrity – honest, ethical, high morals, has personal values, does what’s right
•(ISk) Interpersonal Skills – nice, personable, sense of humor, friendly, nurturing, empathetic, has self-control, patient, sociability, warmth, social skills
•Positive attitude – optimistic, enthusiastic, encouraging, happy, confident
•Professionalism – businesslike, well-dressed, appearance, poised
•(RS) Responsibility – accountable, reliable, gets the job done, resourceful, self-disciplined, wants to do well, conscientious, common sense
•(TS) Teamwork – cooperative, gets along with others, agreeable, supportive, helpful
•Work Ethic – hard working, loyal, initiative, self-motivated, on time/attendance

suggested some ways to foster collaboration within DevOps team but did not specify the extent of such collaboration. In a companion study focusing on new forms of collaboration, Wiedemann [13] highlighted assimilation stages of innovation within DevOps teams but did not consider the question of the extent of the collaboration. Hence our research questions:

- To what extent are soft skills perceived as important by IS Function members transitioning from agile to DevOps?
- What are the implications for collaboration when transitioning to DevOps?

2.2 Agile, DevOps and IT Jobs

Agility is defined as “the continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment” [14]. Agile methods arise from the inability of conventional methods, i.e. Waterfall, to give satisfaction in a changing environment [15]. DevOps initiative was launched to extend the movement towards the agile by including Operations and Quality. The idea was to solve the problem of bottleneck present when Development teams were delivering to Operations faster and more frequently. Thus, agile methods form a base for DevOps. Agile methods impact teams as well as jobs on management styles, collaboration, control, new skills set, training or recruitment [16–18]. DevOps philosophy leads to build bridges between Development and Operations teams. DevOps foster the creation of cross-functional teams where each team member need to consider and anticipate the job to be done by other members. For example, developers need to understand real-world production environment where their colleagues will release the code. In the same way, Operations need to integrate the way Developers will produce the code, will test it and will build the delivery package. It is therefore fundamental in a DevOps configuration to increase

test automation [19] to optimize end-to-end deployment processes. Indeed, “automation is the key to efficient collaboration and tight integration between development and operations” [20].

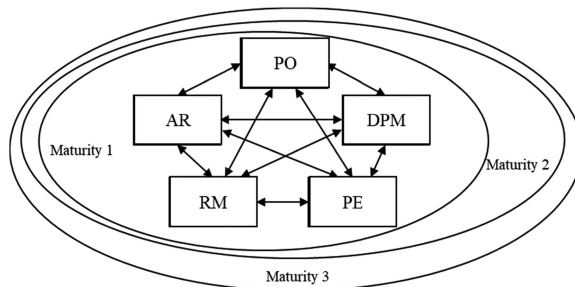
2.3 Roles, Competencies of IT Jobs in the Transition from Agile to DevOps

Humble et al. [21] presented four core values for DevOps: Culture, Automation, Measurement and Sharing. In the recent DevOps handbook, Kim et al. [22] suggest that “a high-trust culture that enables all departments to work together effectively, where all work is transparently prioritized and there is sufficient slack in the system to allow high-priority work to be completed quickly”. In the IS literature, Ghobadi and Mathiassen [23] study knowledge-sharing barriers in agile software teams. Knowledge sharing is important in these teams as it is essential for collaboration across different specialties. They focus on four jobs: user representative, project manager, developer and tester. While tester can be part of development teams, they can also be considered as part of the operations. In this paper, we focus on the jobs which are a priori most impacted by DevOps in terms of skills and competencies. In consultation with industry experts, we selected five jobs of core interest: (1) product owner (a more commonly used title than user representative), (2) managers, (3) architects, (4) production engineers who perform the tests as Ops (in fact developers also perform tests in a DevOps mode), (5) release managers. We did not consider developers in this paper. Although developers are certainly also impacted in the way they share knowledge and collaborate, we did not see particular high challenges for them in moving towards DevOps, while greater challenges were identified and anticipated for others. This view was confirmed by the industry experts. The main argument was that the core activity for developers is coding and that while they would have to take into account additional factors when coding in a DevOps context, the coding role would still be central, and they would not have to evolve as much as others.

The **Product Owner (PO)** role is essential when working in agile with Scrum. POs play a dual role, as representative of the client needs, but also with a real operational role that links the business to project management. POs are responsible for optimizing the value of what development teams produce. Autonomy is essential for POs to succeed and their decisions must be respected by all stakeholders [24, 25]. There is some debate as to whether POs can also act as project managers (PM). They can definitely be managers as indicated in the Scrum guide, “Product Owner is the sole person responsible for managing the Product Backlog” [25]. However, there is no clear indication regarding project management in the Scrum method. The Scrum guide mentions that “Scrum Teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish their work, rather than being directed by others outside the team” and “Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person” [25]. Therefore, this means implicitly that self-organization is compulsory which avoids the possibility for the team to be managed by a PM from the team or external to the team. Theoretically in the Scrum method, POs are managers, but not PM. POs are associated with business ownership and not project management even if they are accountable for

the product backlog management and the final product delivery. However, in reality a PO can also assume the function of PM. Beyond, the distinction between the PO and the PM, some studies mix the PM profile with Scrum Master (SM) profile to identify Scrum Product Owner competences [26].

The notion of **Manager (DPM for Department & Project Manager)** is very large, and the literature contains many definitions. In our conception, a manager is a person with the responsibility to deal with resources, tangible and intangible, to serve a specific objective. The Managers may be project leaders or simply team leaders, or both. It is simply a feature of the hierarchy that often places these two roles in the same person. Others like SM are either not always present or are played by the more traditional role of developer or PM. Team Managers (TM) include different professions: development/operational or supervisory team leader, and qualification-integration manager. PM define and manage an IT project from conception to delivery to get an optimal result for customer requirements of quality, performance, cost, time and security. The **architect (AR)** role is evolving. The AR is often brought to develop multiple skills, whether it is a technical AR, a software-application AR, or a functional AR. On the Ops side, **Production Engineers (PE)** or production integrators and testers are responsible for production, operations, incident monitoring, support and user support. PEs participate in the development of architectural files and the production of applications. PEs also provide the expertise and support for incident resolution. Finally, on the Ops side, the **Release Manager (RM)** is key to ensuring the success of projects. RMs are responsible for the deployment processes. They follow the different versions and coordinate between the development and test teams and the deployment teams. The RM is associated, in the French context, with the function of PM Implementation. This is an essential activity but the scope is questioned by DevOps. We investigated to what extent SSk were perceived as important by these five roles when transitioning from agile to DevOps and the implications for collaboration (See Fig. 1).



Mat.1: Agile moving towards DevOps / Mat.2: Continuous Integration /
Mat.3: Continuous Delivery

Fig. 1. Existing 5 Roles related to 3 Maturity Levels

3 Methodology

This study was conducted in a large European services firm with thousands of IT Staff using agile methods since 2010 and DevOps since 2016.

We followed case study method combining interviews, observations and documentation [27]. We studied 5 job roles and their scope of collaboration in 11 project teams, which involved 54 in-depth face to face individual interviews (these were recorded and lasted 90 min on average). About DPM, we interviewed six PM and six TM. Interviews were conducted by three interviewers and were subsequently transcribed and coded. Codification was triangulated to verify if codes were consistent and similar. This can be viewed as an embedded case study, where job roles are embedded in teams and are centered on collaborative relationships with other job roles.

First, a review of the literature identified the appropriate criteria for selecting the sample of 54 individuals. Second, we observed and interviewed four teams for 15 days (mat. 2 and 3) to compare elements in the literature and those in practice, i.e. size of the team, agile to DevOps maturity (as per our model), and level of externalization. Third, we conducted a series of interviews with 12 strategists in IS and Human Resources, to incorporate their vision and to validate the selected sample.

Three criteria were retained to select the sample: 1- the size of the project measured in terms of number of staff assigned to the project [28] with small project equal to or less than 14 people and large project larger than 15 people; 2- the outsourcing policy and more precisely the degree and nature of outsourcing on the project, e.g. we realized that fixed-price contracts can pose specific problems when working in agile mode. We contrasted two situations: some project activities carried out in work package and fixed price mode, and project without outsourcing or with an outsourced technical assistance contract; 3- the level of maturity in terms of the agile to DevOps transition as presented in Sect. 2.3 above. All theoretically possible configurations allowed us to identify 12 distinct types of projects. This document covers 11 projects whose analysis has been completed. The case of fixed-price outsourcing, maturity level 2 and large project was not completed. Then, agile stage: agile can be considered maturity level 1 in the transition to DevOps, because it facilitates more frequent releases as development becomes more iterative. However, DevOps is not realized because Dev and Ops continue to work in silos with limited sharing, common culture and automation of releases. Maturity level 2 is that of Continuous Integration: the Ops function has to be aligned with the Dev function and both begin to perform various tests (unit and non-regression tests) [29] which are synchronized with code development. Lastly, maturity level 3 is that of Continuous Delivery Stage: integration tests with the other components, end-to-end tests, performance tests, user acceptance tests are then performed by Ops and co-designed with the Dev function [30] (Table 2).

Documents, observations and interviews performed in the first stage considerably helped us in interpretation of the 54 interviews in the second stage. Collaboration scope was parsed by interviewing individuals to express with whom they worked the most. While it is susceptible to bias, we did not want to induce other biases by repeating this question for each possible job role they could have collaborated with. Collaboration enrichment was examined through responses to whether and how individuals were

Table 2. Distribution of 54 interviews per contingencies

Project Size: 25 Small (min: 5; max: 14) / 29 Large (min: 15; max: 80)
Type & Degree of Sourcing: 24 Outsourced Fixed-Price/ 30 Other (Internal or other)
Agile Maturity: Mat.1: 19 / Mat.2: 15 / Mat.3: 20

satisfied or unsatisfied in our semi-structured interview. To investigate the skills issue in the 54 interviews, we asked what skills (know-how, behaviour) were necessary for the evolution of their jobs. We then analysed SSk detecting the presence of any subcategories in keeping with Robles [8] set of SSk (cf. Table 1).

4 Findings

4.1 Changes in SSk Perceptions When Considering Maturity Levels

Codification of interviewee skill perceptions allowed us to complete nine of the ten groups of SSk identified by Robles (2012). Only half of Robles [8] 10 groups of SSk are mentioned by interviewees for each maturity level. Five groups of SSk are under-represented if not absent within representations of Agile and DevOps teams whatever the maturity level, i.e. integrity, work ethic, courtesy. Among them, positive attitude clusters various skills, i.e. being optimistic, enthusiastic, confident, encouraging, and is only cited by PO and DPM at higher level of maturity (2 and 3). Beyond these results, we came across five groups of SSk represented in each maturity level.

Communication skills (CS) refer to the ability to communicate both orally and in writing, presenting and listening, and are present at each maturity level. A RM highlighted the importance of CS: “I think you have to know how to communicate” (RM, mat. 1). This perception is shared by many people on both sides, Dev and Ops, e.g. a PO (mat. 2): “You have to listen, so especially not having the posture of “I know, I have experience”, especially today”. A DPM (mat. 3) confirms the central position of communication saying that “Communication is still a fundamental basis of teamwork. And I think that often, either because we are partitioned (...) we are somewhat inclined to understand or integrate neither the constraints of the rest of the team nor other people from other teams”. We noted that the perception of CS was slightly higher in terms of diversity and frequency of mentioned skills when maturity level was going up. These previous verbatim show CS at mat. 1 covering basic CS principles as to convey an information between two or more stakeholders, then CS at mat. 2 with a stress on listening skills integrating humility, and finally CS at mat. 3 with a wider coverage and a higher degree of CS incorporating large understanding skills and their usage to interact with teams and integrate their constraints.

Interpersonal skills (ISk) are related to sociability, empathy, nurturing, friendly, patience; self-control capacity. As for CS, ISk appear at each maturity level. A PO underlined the importance of ISk to satisfy different stakeholders and reach final goal: “Coordination, of course, then relational, because everyone is not going necessarily in

the same direction or does not necessarily have the same objectives (...) On the one hand, the production and the operations managers ask to limit the number of releases or production launches (...) on the other hand, we have businesses that always ask for more.” (PO, mat. 1), a shared vision by a DPM (mat. 3): “Stakeholders analysis! I change partners very often. So, know how to evaluate a situation quickly, if I have someone who is supportive, or if someone who is perhaps a little more reluctant (...) You are asked to get closer to a certain number of people with whom you are not used to work with (...) so it requires human qualities.”. The perception of ISk was slightly larger in terms of diversity and frequency of skills when maturity level was higher.

Flexibility skills (FS) signify to be ready to change, to learn throughout the life, to accept new things, to adjust, to adapt. FS are present at all levels of maturity and on both Dev and Ops sides. Moreover, people from maturity level 3 seem to have a clearer and wider representation of FS. A PE (mat. 1) simply mentioned one sub-skill of FS group, adaptability: “Adaptation is perhaps most important, because it changes all the time, businesses, tools, applications and technologies”. At a similar position, a PE (mat. 3) asserted: “Everything evolves so quickly...it is better to know how to learn than to develop know-how because that will change, it is important to be open to the novelty”. A PO (mat. 2) stated that: “You have to feel and accept the error. One must accept sometimes and assume one has made a mistake, and it is a step backwards”. The skill is about adaptability but with a deeper and larger interpretation of learning and adaptability. Finally, a DPM (mat. 3) on a small project suggested: “You must be humble or agree that you don’t know...your neighbor knows better and will show you. (It is better) not to impose your vision either.” Meanwhile another DPM (mat. 3) confirmed this “The challenge is to change the subject - just able to juggle these topics. It necessarily requires great flexibility and adaptability. A questioning, because suddenly, you’ve been in the business a long time, have already proven worth in the past, and you are asked to work in an-other way”. We discovered that FS perception was broader and much more important in terms of diversity and frequency of mentioned skills when maturity level was higher.

Teamwork skills (TS) indicate a capacity to cooperate, to be supportive, collaborative, helpful. TS are intimately linked to Agile Principles, i.e. “Business people and developers must work together daily throughout the project”, as well as Agile Methodologies (Scrum, XP), Agile Practices (daily stand-up, sprint review) or DevOps philosophy breaking silos between Dev and Ops and building bridges and fostering new teams. As mentioned by a PO, it is important to “understand that we do not work each one in his own corner (...) it must be shared a minimum, that every-one knows where the others are (...) all the developers do not need to know where all the integrators are, but we can have a roughly correct and almost real-world view of where we are. to know how to adjust, to know how to project also.” (PO, mat. 1). From the Operations side, an RM highlighted a cultural change regarding Teamwork: “We do not ask for technical expertise anymore, because we really want people to stop solving the problem technically. (We prefer) to get the problem solved by a community. It is really a different positioning”. (RM, mat. 2), a vision confirmed by another Ops: “You really need to have the vision of belonging to the same team, and not to say: “Me, I do represent Production (Ops)”, to feel “powerful” because we (Ops) take the decision in fine to put the application into production.” (PE, mat. 3). Finally, a DPM on a small

project (mat. 3) demonstrated her way to become supportive and collaborative integrating fully her team: “You also have to be ready to join this team even if you are a manager of some of them. We (the DPM) must also know when to take a back seat.” When team members are really acculturated to work together, i.e. in advanced DevOps team, silos disappear. We figured out that the perception of TS was moderately larger in terms of diversity and frequency of mentioned skills when maturity level was higher.

Lastly, as a DPM (mat. 3) stated clearly: “the responsibility necessarily, to know to be responsible.”, thus Responsibility skills (RS) which covers two aspects. The first one is more evoked by PE: get the job done, and well done, conscientious, accountable. The second one is more perceived by DPM: trustworthy, astute, common sense imply to get the job done, to be conscientious, accountable, reliable. These skills are mostly cited by DPM, afterwards PE and to a lesser extent RM, and then few PO and very few AR. A large and complex PM suggested: “I want to say for commitment too. If they are self-organized, they must get in deep. They take a subject, they go to the end. They bring their ideas, they apply their ideas, and then when they fail, they repair. We must assume.” About conscientiousness, the sense of getting the job well done and being resourceful, a PE on a large project (mat. 3) declared: “You do not have to be an expert to find the solution. You must understand what’s going on. And about the know-how-to-be well, you must listen rather than talk.” We identified this same preoccupation of consciousness, reaching objectives with POs trying to stay the course: “do not do the wind vane. We can be wrong, but on the other hand, we do not return every day on what we said the day before. (...) It is disastrous for the progress of work.” (PO, mat. 1) We detected that the perception of RS was decreasing in terms of diversity and frequency of mentioned skills when maturity level was higher.

To conclude, we found that Agile maturity is a factor impacting SSk perception. First, FS are much higher in team members in a more mature agile environment (mat. 3). Secondly, the higher the maturity level, the more the presence and the diversity of CS and ISk increase. Third, TS and team spirit, are more present in a mat. 3 project compared to a mat. 2 or 1, which may seem logical but not obvious, particularly when considering the fundamentals of the agile approach which require a Teamwork culture. Fourth, surprisingly, RS are more present in a lower agility context (mat. 1) and decrease when maturity is higher. This could mean that in an agile environment, the sense of responsibility is more formally distributed, and in a mature DevOps configuration, the sense of responsibility is more shared within the team. In a DevOps environment, we can explain this result with Automation that could reduce and/or transfer responsibilities and therefore responsibility skills.

Type of Outsourcing may have an impact on SSk ‘perceptions’. Some competences, such as sense of responsibility or teamwork are more present when the project is internal. In contrast, outsourcing seems to be a factor which results in increasing FS and ISk. The perception of SSk seems to vary less with project size than with type of outsourcing. Overall maturity level is the most important.

4.2 Roles and Collaboration Analysis

To the question about the actors with whom they interacted the most, all interviewees cited more than 15 different co-worker’s functions. For maturity 1, projects, main

collaborations between Dev and Ops are less numerous (20) than those within Dev and within Ops (31). Collaborations crossing the two functions become more numerous than those internal either to Dev or to Ops (20 against 16) for projects in maturity 2, and then become balanced (31 against 32) for projects in maturity 3. This balancing with DevOps means that collaboration with members of other function are perceived as important as internal collaborations within Dev and Ops functions respectively.

Table 3 lists these main collaborations by role/profession; for example, 4 of the 10 RM reported the developer as a main co-worker. Some similarities and differences in collaborations are observed for the five key interviewed roles. As expected, the DPM reported the highest number of collaboration with different co-workers in comparison to the four other interviewed roles. This highlights their large scope of collaborations. Another important observation is the key role of the PO: all interviewed professions have mentioned their collaboration with the latter.

Table 3. Main collaborations reported by each role.

		PO (N=11)	DPM (N=12)	AR (N=10)	PE (N=11)	RM (N=10)	
Main Collaborations cited	DEV →	Product Owner (PO), N	2	12	10	11	10
		Department & Project Manager, N	9	12	10	11	10
		Architect (AR), N	3	1	1	4	2
		Scrum Master, N		3			
		Developer, N	8	6	8	10	4
		DevOps animator, N		1			
		Technical expert, N		1		2	1
		Quality expert, N	1				
		Multiple collaborators, N		2			
		Qualifier, N		1			
		Production Engineer (PE), N	1	4	2	1	8
		Admin. Infrastructure Director, N			1		
		System Engineer, N				1	
	OPS ←			3	2	4	3
	Release Manager (RM), N	2	1	6	6		

In Table 3, we can see more precisely the main collaborations between Dev and Ops. From the Dev side, we highlight collaborations between PO and RM, AR with operators or DPM with PE and operators. From the Ops side, we see a strong collaboration between PE and Developers, between RM and PO and with developers. Given the question asked, in traditional developments mode or in agile (mat. 1), these collaborations would not have been considered as main collaborations.

Moreover, all respondents have highlighted an improvement of the collaboration’s quality between co-workers in an agile context. As mentioned during the interviews, the setting up of daily meetings or sprint reviews, but also the use of collaboration tools (such as the Mingle) could partly explain this finding. Indeed, the setting up of these

daily meetings will rhythm the collaborations in a formal way. However, the frequency of meetings can be perceived as an overwork. For each role, we also describe the different forms of collaboration and types of actors involved.

Release Manager (RM). The RM is at the center of numerous collaborations within the company and in particular with the production team. All RMs underlined interactions with the project manager, generally on a weekly basis. This collaboration is, and must be, continuous to prioritize certain tasks and thus ensure the success of projects. For example, a RM explained (mat. 3): “every week we establish a point called the “Common Work Plan of Exploitability” where we exchange. We go through all actions, all activities in progress, and ... we see what has to be prioritized in the case of new actions”. These collaborations are often real driving forces to create a dynamic for all teams working on the same project. In addition, almost all RMs reported collaborations with production engineers (see Table 3). Their exchanges are frequent, (RM mat. 2): “we discuss certain points between us before going to meet the project manager. Given the slightest problem, one goes to see the other and vice versa”. It should be also noted that in this context, the setting-up of project group meetings provided the opportunity to gather RMs, developers and operators. The maturity level in the transition from agile to DevOps could influence the frequency of these collaborations. In fact, “The developers for me are hidden behind the project manager. But this evolves within the most advanced projects in the DevOps approach” (RM mat. 3). Thus, an advanced application level of agile and DevOps methods and concepts might naturally lead to establishing more interactions with developers.

Product Owner (PO). As expected, most POs reported collaborations with developers. Some of these interactions are required for the validation of the sprint during the demos. The maturity level in the transition from agile to DevOps has a direct impact on the frequency and quality of these interactions. Thus, a PO suggested that to feel part of an integrated team (mat. 3): “Today we are really an integrated team with marketing, developers.” In the case where several POs participate in the same project, they collaborate with each other daily via email or telephone and during the weekly “sales meeting”. This meeting is also a moment of exchange with the Product Manager and line managers. Moreover, some POs mentioned multiple exchanges with functional architects. In line with the various roles inherent to the PO roles, the majority of POs also reported collaborations with business sponsors or users, albeit not considered main ones. It should be noted that several POs highlighted the limited collaborative exchanges with supervisor-operators (mat. 2): “Those with whom I exchange the least, are probably exploitation”; “I have difficulties to express myself on this point, they (“supervisor-operators”) are more related to developers.”

Architect (AR). As in previous findings, the two main co-workers mentioned by the architects were PMs and developers (see Table 3). Most architects reported interactions with PMs to share information and views on a project. Collaborations were also reported with development team to exchange information, especially during sprint reviews and demos. An architect stressed the importance of this (mat. 3): “we have to be precise on how to proceed, and if feasible, establish a stronger, detailed collaboration with the technical manager”. These exchanges take place regularly. Several architects mentioned information exchange with the RM profession, especially to agree on norms, standards and good practices (mat. 2): “The (RM) is supposed to hand over to his team,

to read the technical architectural file”. An AR (mat. 3) highlighted the impact of the transition to DevOps on his work and collaborating: “it’s an approach allowing better understanding the role of each person involved in the project. In the end, each remains accountable; but decision-making in an architecture problem (...) will be more shared than before (...) thus, problems disappear.”

Production engineer (PE). The main collaborations cited by PEs are with the PM and developers (see Table 3). Thus, most PEs agree on the existence of a significant mutual collaboration with developers (mat. 2 and 3): “we inform each other, it’s natural...with the developers, we are becoming a unique team: the DevOps one.”

In addition, in their daily activities the PMs must provide applications to PEs so significant collaboration with shared responsibilities have been reported. However, some difficulties of co-operation have been mentioned, for instance a PE pointed out the following (mat. 2): “I was the one who explained what I expected in terms of documentation, whereas it’s their job.” PEs also discussed their collaborations with RMs. However, the frequency of these collaborations can vary a lot across projects. Indeed, in a context where the RM is entirely dedicated to the team’s projects, collaborations are frequent. A PE described this collaboration as follows (mat. 2): “In our organization, the role of RM was created because of administrative importance of our project, it’s useful to lighten our work (...) The role of the RM is to anticipate complex operations, to be sure we do not forget anything”. A PE (mat. 3) explained his view of the evolution of collaboration in a DevOps team: “the roles of PM and RM no longer exist in a DevOps team; At the development team level, the role of PM is operated by the Scrum Master; at the planning level, project management role is a crucial role covered by the team. And the role of RM is operated both by PO, technical expert, and production engineers, and operators”.

Department and Project Managers (DPM). PMs are at the center of multiple collaborations, both inter and intra-team. The manager profile collaborates with many actors (see Table 3). Most also mentioned collaborations with other PM and POs, in particular to discuss budget, tracking, prioritization of items. Several Managers also highlighted a close relationship with functional architects (mat. 2): “I work early with our functional architects to define and schedule future development.” In addition, although the PM is not a functional expert, he approves the conformity of the result.

To conclude, the collaborative scope within the main collaborators clearly changed and is now more balanced. The perceived greater richness of collaboration reflects a better understanding among members. This greater richness appears to be more salient when projects are in maturity 3. Consequently, considering our findings on skills and collaboration, we could describe a three levels maturity model as follow. At maturity 1, there is a lower range of collaborations which could be explained by the DevOps philosophy itself opening collaboration to a new extent between Dev and Ops while Agile limits collaboration among Dev. Because agile methodologies ask for it, communication skills are very important and responsibility skills very developed as interpersonal skills. However, surprisingly, flexibility and teamwork are less present in their discourses. At maturity 2, the extent of collaboration is thus wider than for mat. 1. All the SSk are more represented at this level, except for RS which are decreasing. At maturity 3, the extent of collaboration is similar to mat. 2 or can slightly decrease. We explain this finding because when teams reach mat. 3, they know better that anyone

with who to collaborate to be more efficient, hence they limit or cancel collaboration with some stakeholders. SSk increase in comparison to mat. 2, except for RS, and this is particularly true for FS, TS and ISk.

5 Discussion and Conclusion

DevOps and Agile methods impact collaboration and existing IT skills; therefore, it is essential to understand the requirement for new skills [31] and new curricula for IT students [32, 33]. Companies struggle to determine the most important tasks for IT teams, specifically future Release/DevOps engineers [34]. Understanding the evolution of IT skills would help organizations to hire skills that fit their needs [10]. This is lacking in DevOps job advertisements which largely neglect SSk [35] and focus on hard skills, e.g. technical skills, thus confirming a recruitment gap [9, 10]. We sought to investigate what skills and competencies are required notably for the collaboration culture between Dev and Ops when a firm engages in Agile and DevOps.

Our study provides compelling evidence that the move from Agile to DevOps Maturity has an impact on perceived skills in use or in making. While all agile software projects require significant SSk, these are even greater in DevOps especially at mat. 3. Our study shows flexibility and interpersonal skills are important for agile teams [11], and also extends this to others SSk such as teamwork and communication and suggests that SSk are even more important for DevOps. Our findings are largely congruent with that of Wiedemann et al. [12] and complement with an in-depth analysis of SSk in Agile and DevOps teams. Beyond key capabilities highlighted by Wiedemann et al. [12], we identified characteristic features of DevOps teams.

Regarding collaboration, we showed that collaborative scope among the main collaborators has drastically changed and is now more balanced. Also, the perceived greater richness of collaboration reflects a better understanding of roles in the overall project. We provide greater clarity about collaborations within a DevOps team than previously available (e.g., [13]). Our findings align with Humble [21] who highlighted the importance of collaboration and shared responsibilities and more specifically with skills, experience and mindset of Ops people. This expression of greater richness seems more salient when projects are in continuous delivery (mat. 3).

Overall, these results show disruption in collaboration between agile and DevOps. The transition to DevOps facilitates the IS function becoming smart for three reasons. Firstly, adaptability is considered an indicator of smartness. Extending collaboration across functions within IS is a sign of adaptation where there is an evident need. The fact that delivery can be continuous at DevOps stage is also a sign of adaptation to a more demanding environment and of higher performance. DevOps is the agile approach whereby better sharing of information and developing a common culture can happen across different roles and jobs throughout the IS function as a whole which overcomes the traditional distinction between Dev and Ops.

References

1. Brunnert, A., et al.: Performance-oriented DevOps: a research agenda. ArXiv Prepr. ArXiv150804752 (2015)
2. Debois, P.: DevOpsDaysGhent. DevOpsDays, Ghent, Belgium (2009)
3. Debois, P.: DevOps: a software revolution in the making. *J. Inf. Technol. Manag.* **24**, 3–39 (2011)
4. Conboy, K., Coyle, S.: People over process: the implications of agile for IS skills and Human Resource Management. *IEEE Softw.* **28**, 48–57 (2009)
5. Gallivan, M.J., Truex III, D.P., Kvasny, L.: Changing patterns in IT skill sets 1988–2003: a content analysis of classified advertising. *ACM SIGMIS Database DATABASE Adv. Inf. Syst.* **35**, 64–87 (2004)
6. Wong, S., von Hellens, L., Orr, J.: Non-technical skills and personal attributes: the Soft Skills Matter Most. In: Proceedings of the 6th Australasian Women in Computing Workshop, Brisbane, Australia, pp. 27–33 (2006)
7. Boyatzis, R.E., Kolb, D.A.: From learning styles to learning skills: the executive skills profile. *J. Manag. Psychol.* **10**, 3–17 (1995)
8. Robles, M.M.: Executive perceptions of the top 10 soft skills needed in today’s workplace. *Bus. Commun. Q.* **75**, 453–465 (2012)
9. Todd, P.A., McKeen, J.D., Gallupe, R.B.: The evolution of IS job skills: a content analysis of IS job advertisements from 1970 to 1990. *MIS Q.* **19**, 1–27 (1995)
10. Trauth, E.M., Farwell, D.W., Lee, D.: The IS expectation gap: industry expectations versus academic preparation. *Mis Q.* **17**, 293–307 (1993)
11. Vivian, R., Tarmazdi, H., Falkner, K., Falkner, N., Szabo, C.: The development of a dashboard tool for visualising online teamwork discussions. In: IEEE/ACM 37th IEEE International Conference on Software Engineering, pp. 380–388. IEEE (2015)
12. Wiedemann, A.M., Schulz, T.: Key Capabilities of DevOps teams and their influence on software process innovation: a resource-based view. In: Proceedings of 23rd ACIS Americas Conference on Information Systems, pp. 1–10, Boston (2017)
13. Wiedemann, A.: A new form of collaboration in IT teams-exploring the DevOps phenomenon. In: Proceedings of the 21st PACIS Pacific Asia Conference on Information Systems, pp. 1–12, Langkawi (2017)
14. Conboy, K., Fitzgerald, B.: Toward a conceptual framework of agile methods: a study of agility in different disciplines. In: Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research, pp. 37–44. ACM, Newport Beach (2004)
15. Highsmith, J., Cockburn, A.: Agile software development: the business of innovation. *Computer* **34**, 120–127 (2001)
16. Conboy, K., Coyle, S., Wang, X., Pikkarainen, M.: People over process: key people challenges in agile development. *IEEE Softw.* **28**, 48–57 (2011)
17. Coram, M., Bohner, S.: The impact of agile methods on software project management. In: ECBS 2005 Proceedings, pp. 363–370. IEEE Computer Society, Greenbelt (2005)
18. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. *Commun. ACM* **48**, 72–78 (2005)
19. Lwakatare, L.E., Kuvaja, P., Oivo, M.: Dimensions of DevOps. In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 212–217. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18612-2_19
20. Wettinger, J., Breitenbücher, U., Leymann, F.: Standards-based DevOps automation and integration using TOSCA. In: Proceedings of the 2014 IEEE/ACM, pp. 59–68. IEEE Computer Society, London (2014)

21. Humble, J., Molesky, J.: Why enterprises must adopt DevOps to enable continuous delivery. *Cut. IT J.* **24**, 6 (2011)
22. Kim, G., Debois, P., Willis, J., Humble, J.: *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution, Portland (2016)
23. Ghobadi, S., Mathiassen, L.: Perceived barriers to effective knowledge sharing in agile software teams. *Inf. Syst. J.* **26**, 95–125 (2016)
24. Coyle, S., Conboy, K., Acton, T.: An exploration of the relationship between contribution behaviours and the decision making process in agile teams. In: *Proceedings of the 36th ICIS International Conference on Information Systems*, pp. 1–15, Fort Worth, TX, USA (2015)
25. Schwaber, K., Sutherland, J.: *The scrum guide*. Scrum Alliance 21 (2011)
26. Oomen, S., De Waal, B., Albertin, A., Ravesteyn, P.: How can Scrum be successful? Competences of the Scrum Product Owner. In: *Proceedings of the 25th ECIS European Conference on Information Systems*, pp. 131–142, Guimarães, Portugal (2017)
27. Yin, R.K.: Validity and generalization in future case study evaluations. *Evaluation* **19**, 321–332 (2013)
28. Rolland, K.H., Fitzgerald, B., Dingsøyr, T., Stol, K.-J.: Problematizing agile in the large: alternative assumptions for large-scale agile development. In: *Proceedings of the 37th ICIS International Conference on Information Systems*, pp. 1–21, Dublin, Ireland (2016)
29. Ståhl, D., Bosch, J.: Modeling continuous integration practice differences in industry software development. *J. Syst. Softw.* **87**, 48–59 (2014)
30. Chen, L.: Continuous delivery: overcoming adoption challenges. *J. Syst. Softw.* **128**, 72–86 (2017)
31. Tan, C.-H., Teo, H.-H.: Training future software developers to acquire agile development skills. *Commun. ACM* **50**, 97–98 (2007)
32. Bang, S.K., Chung, S., Choh, Y., Dupuis, M.: A grounded theory analysis of modern web applications: knowledge, skills, and abilities for DevOps. In: *Proceedings of the RIIT, 2nd Annual Conference on Research in Information Technology*, pp. 61–62. ACM, Orlando (2013)
33. Betz, C., Olagunju, A.O., Paulson, P.: The impacts of digital transformation, agile, and DevOps on future IT curricula. In: *Proceedings of the 17th Annual Conference on Information Technology Education*, p. 106. ACM, Boston (2016)
34. Kerzazi, N., Adams, B.: Who needs release and DevOps engineers, and why? In: *Proceedings of the International Workshop on Continuous Software Evolution and Delivery*, pp. 77–83. ACM, Austin (2016)
35. Hussain, W., Clear, T., MacDonell, S.: Emerging trends for global DevOps: a New Zealand perspective. In: *Proceedings of the 12th ICGSE International Conference on Global Software Engineering*, pp. 21–30. IEEE Press, Buenos Aires (2017)