International Journal of Information Management xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

International Journal of Information Management

journal homepage: www.elsevier.com/locate/ijinfomgt

A case analysis of enabling continuous software deployment through knowledge management

Ricardo Colomo-Palacios[a], Eduardo Fernandes[b], Pedro Soto-Acosta[c,*], Xabier Larrucea[d]

[a] Department of Computer Science, Østfold University College, B R A Veien 4, 1783 Halden, Norway
[b] Meta4 Spain, Centro Europa Empresarial, Edificio Roma, C/. Rozabella, 8, 28290 Las Rozas Madrid, Spain
[c] Department of Management & Finance, University of Murcia, Campus de Espinardo, 30100 Espinardo, Murcia, Spain
[d] Tecnalia, Parque Tecnológico de Bizkaia. Calle Geldo, Edificio 700, E-48160 Derio, Bizkaia, Spain

ARTICLE INFO

Keywords:
Knowledge management
Continuous deployment
Continuous software engineering
DevOps
Case study

ABSTRACT

Continuous software engineering aims to accelerate software development by automating the whole software development process. Knowledge management is a cornerstone for continuous integration between software development and its operational deployment, which must be implemented using sound methodologies and solid tools. In this paper, the authors present and analyse a case study on the adoption of such practices by a software company. Results show that, beyond tools, knowledge management practices are the main enablers of continuous software engineering adoption and success.

1. Introduction

In order to preserve their competitive advantage, software producers need to deliver products and new features to customers as fast as they can. It is generally accepted that important problems in software delivery are rooted, among other aspects, in the disconnections among software development activities, causing delays in software delivery (Fitzgerald & Stol, 2017). This lack of connection lies not only on the technical side, where human aspects and knowledge management facets are some of the main areas to be improved. Continuous software engineering permits software features delivery at rates which a few years ago would have been considered unachieveable (Colomo-Palacios, Fernandes, Soto-Acosta, & Sabbagh, 2011, p. 4; O'Connor, Elger, & Clarke, 2017). This approach is based heavily on applying automation to the overall software development process (including code collaboration tools, verification, version control system, deployment and release management…) by using several tools. These tools act as structures in which different types of knowledge are coded and shared among software practitioners.

Like any other approach, continuous deployment presents benefits but also caveats. On the benefits side, the literature reports: Increased customer satisfaction, shorter time-to-market, higher developer productivity and efficiency, continuous rapid feedback and, finally, higher quality and reliability. With regard to the challenges, researchers found the wide panoply of tools available and their integration, organizational culture to be a hindrance to the transformation process and increased quality assurance efforts.

The continuous approach goes beyond the borders of traditional software development to reach the operational side as well. In this scenario, DevOps stands for a continuous integration between software development (Dev) and its operational deployment (Ops). DevOps efficiently integrates development, delivery, and operations, thus facilitating a lean and fluid connection of these traditionally separated silos (Ebert, Gallardo, Hernantes, & Serrano, 2016). Consequently, DevOps implies a cultural shift toward collaboration between development, quality assurance, and operations (Ebert et al., 2016). The success of DevOps is based on four principles (Humble & Molesky, 2011):

- Culture. Joint responsibility for the delivery of high quality software.
- Automation. Automation in all development and operation steps towards rapid delivery and feedback from users.
- Measurement. All process must be quantified to understand delivery capability and proposals of corrective actions should be formulated for improving the process.
- Sharing. Sharing knowledge enabled by tools is crucial.

Accordingly, knowledge management is one of the pillars of DevOps and must be implemented using sound methodologies and solid tools. The literature has reported specific knowledge management systems

* Corresponding author.
E-mail addresses: ricardo.colomo-palacios@hiof.no (R. Colomo-Palacios), eduardofer@meta4.com (E. Fernandes), psoto@um.es (P. Soto-Acosta), xabier.larrucea@tecnalia.com (X. Larrucea).

https://doi.org/10.1016/j.ijinfomgt.2017.11.005
Received 24 October 2017; Accepted 17 November 2017

designed and implemented to serve in DevOps scenarios (Wettinger, Andrikopoulos, & Leymann, 2015). Focusing just on the development side of DevOps, Knowledge management is seen as one of the cornerstones for software quality. These authors indicate that in the context of software quality, knowledge management comprises aggregation, distribution, visualization of data, and information and knowledge to support collaborating stakeholders in fulfilling their quality-related tasks and decisions (Del Giudice & Della Peruta, 2016).

In spite of the importance of the topic, to the best of authors' knowledge, there are no research studies that go beyond the explanation of knowledge management tools on knowledge management factors in continuous software engineering or DevOps scenarios. This paper aims to bridge the gap in this important topic.

This case is structured as follows: Section 1 above contains a brief introduction to Continuous Software engineering, continuous deployment and DevOps. In Section 2, a background of the company in which the case study is conducted is presented. Section 3 presents the main aspects on the team leading the DevOps efforts based on continuous deployment. This is followed by Section 4, in which the research methodology for this case study is presented. In Section 4, the case study findings are analysed and discussed. Section 5 provides a discussion and describes the lessons learnt. Section 6 presents the main conclusions of the case study.

2. Company background

Meta4 is a world leader in human capital management solutions. Founded in 1991, Meta4 has more than 1300 clients in 100 countries. More than 18 million employees are managed via Meta4 software. In 2016, Meta4 made 63 million euro, 5% more than for 2015, achieving record takings through its line of cloud HR and payroll solution.

Meta4, with 950 employees worldwide, has branches in eleven countries, although the headquarters of the company is located in Madrid, Spain. Meta4 moved from on premise products to service-oriented cloud solutions. Cloud solutions have experienced a 26% increase in 2016, showing a clear market movement in that direction. For 2017, Meta4 forecasts a significant increase in sales from their cloud HR solutions, so continuing the company's upward trend of recent years. This leads to a new scenario for the company in which cloud solutions are emerging as the future of the company in terms of revenue but also in terms of business model and technological approach.

3. The DevOps team

This section begins by describing the scenario before the project started, after which the project scope and objectives are depicted.

According to Gartner, by 2020, 30% of global midmarket and large enterprises will have invested in a cloud-deployed human capital management suite. Meta4 started its efforts towards fully functional cloud solutions around a decade ago. However, it was not until 2013 when DevOps appeared as a possible solution to some of the issues associated with DevOps adoption. The DevOps team was formally established by 2015.

Today, the DevOps team includes ten workers and some occasional collaborators. Meta4 combines DevOps methods with more traditional integration and deployment approaches. Not all cloud features are dealt with by means of DevOps yet; a significant part of the core of the solution is still managed, developed and controlled under traditional approaches.

4. Case study research method

Given the nature of the project and the objectives of the case study, a qualitative research methodology was adopted. More precisely, researchers used the Grounded Theory (GT). Drawing on GT, researchers are able to investigate the organisation from a user-orientated perspective and an organisational perspective and extrapolate findings grounded in the data available. In our case, researchers conducted a set of semi-structured interviews with project group members identified by the project manager. Every interview was voice recorded and then transcribed. The transcriptions were used for the coding of data in the subsequent analysis phase.

5. Lessons learned

The lessons learned during the different phases of the case study can be classified into three different categories as follows: organizational matters, tools and people. In what follows, these areas will be reviewed and discussed.

5.1. Organizational matters

Given the nature of the changes in the organizational, usually the adoption of DevOps practices is not smooth (Zhu, Bass, & Champlin-Scharff, 2016). Literature has underlined the diverse challenges of DevOps adoption and the situation reported in this case provides empirical evidence for DevOps adoption challenging nature.

Respondents identified two kinds of pressure in the adoption of DevOps. The first set comprises external pressures. This is basically, the buzz towards the adoption of DevOps in industry fora at first. This goes beyond the "Technological mimetism" to follow international consultants' advices to follow "On the Rise" practices. At the same time, respondents acknowledge that the evidence of the availability and reported effectiveness and benefits of certain tools was another strong external pressure for adopting DevOps.

The second set of pressures is composed of internal forces. All software companies suffer from pressures of the customer to reduce release times while ensuring high quality. This normally leads to internal pressures from internal sales departments. In this sense, respondents reported that adopting DevOps was also recommended for improving cycle times and overall quality. According to respondents, there was already an established process of semi-automated deployment that managers wanted to improve. DevOps was also seen as a way to improve the whole process. Finally, the evolution of sales towards cloud led also to a separate way to deal with cloud deployments and a new way of tackling the problem naturally led to DevOps.

One aspect mentioned by respondents is the benefits rooted in the partial adoption of DevOps practices in the company. Transitioning toward DevOps is much more complicated with evolving systems (Ebert et al., 2016). Taking this into account, the decision taken to start with a subset of the systems deployed is a feasible approach to minimize risks (personnel rejection, technical, compliance, legal…). This approach is seen as a spearhead in the adoption of DevOps practices in the company.

It is also worth noting that respondents have a perceived payback of the DevOps adoption. Respondents informed that, although there is no sound report of the cost-benefit of the initiative, their perception sets this time at one year. Although there is a need to conduct more rigorous studies in the matter, given the lack of relevant literature of the topic, these figures are in line with one of the few previous reports on the literature. In this work, (Ravichandran, Taylor, & Waterhouse, 2016a), the authors indicate payback period as 11 months for a DevOps project. However, it is also true that Meat4 is a medium-big company within the software industry, with a history of almost 30 years and, in cases like this, innovation adoption presents a quite different pace compared to start-ups. Thus, the capacity of the company to generate benefits from DevOps practices in such a short time is quite remarkable. However, it is also important to note that, in order to present sound metrics, managers in the company are now adopting tools that calculate the full economic impact of DevOps.

Regarding the effects of the initiative, respondents reported two kinds of perceptions. The first is the relatively limited impact of DevOps

on the company as a whole. DevOps is, for the moment, applied to the core technological components. Meta4 uses a metadata oriented product and this requires some more effort to migrate to standard DevOps mechanics. The DevOps impact will increase considerably when the metadata model is included in the process. A possible transition of the solution to a micro-services architecture (partial or total) could increase the importance of DevOps in all use-case scenarios.

This arena is leading us to two types of facts backed up by the DevOps concerns reported. Firstly, given that processes at Meta4 present a high level of maturity, this leads to continuous innovation and process and tools adoption in a faster pace. This aspect will be mentioned again later on in this paper. Secondly, the start-up mentality needed for DevOps (Ravichandran, Taylor, & Waterhouse, 2016b) is present in the organization and its teams. In the pure knowledge management sphere, the interaction-driven absorption traditional in start-up settings is complemented by the codification-driven approach enabled by tools. This aspect will be analysed in the next section.

5.2. Tools

As reported earlier, one of the main external pressures leading to DevOps adoption is the availability of tools. One part of the tools has to do with all elements encompassing an IT system in order to automate system's configuration, but also to code knowledge. Thus, given the need to count on multiple near-identical execution environments, infrastructure-as-code is the concept that keeps these environments consistent. Respondents agreed that, although there were several internal tools developed to control environments, commercial tools are now powerful and reliable tools and relatively widespread in the software community.

However, it is also true that Meta4 has also built internal tools to support certain tasks in the DevOps process. Given that DevOps is not a standard or a closed guide, but an approach to improve software processes, these internal tools are also valid to achieve certain goals. An example of this is the set of tools developed to register bugs and new requirements reported by users (sometimes in an unmanned way) and assign them to the different departments and teams (depending on priority and nature of the bug). This is one of the common processes of all software vendors and, according to respondents, it has been working for years with remarkable results. The DevOps team is also using this tool and sharing it with the rest of the company. Results show the importance of the process beyond specific tools. The tool was already implementing the agile principle to guide the communications between the two DevOps worlds: Development and Operations, and there were no need to adopt another tool to implement the same process. DevOps is more about the agile culture and knowledge-sharing than about specific tools.

This leads us to a new point worth highlighting. Some companies are already embracing some DevOps principles, no matter which specific set of tools they are using. The DevOps concept and its principles go beyond tools, although it is enabled by a panoply of tools available on the market. In a recent study on the topic (Lwakatare, Kuvaja, & Oivo, 2015), the four dimensions of DevOps are identified: collaboration, automation, measurement and monitoring. None of these dimensions are specifically tools, but they are all enabled by tools. DevOps is more a cultural shift for IT than a process or tools shift. This cultural shift is one of the DevOps four principles identified by (Humble & Molesky, 2011), although the other principle in the list – Sharing – is also relevant in our case According to the above authors, sharing, including knowledge-sharing, enabled by tools, is also crucial for DevOps adoption and success. In the case of the bug reporting and assignment tool, this knowledge sharing was already up and running, and there was no need to change it to embrace DevOps principles.

5.3. People

As reported before, the DevOps team is formed by ten members, although there is a set of professionals coming from other departments who join the team occasionally.

With regard to the recruitment process, three types of circumstances occur. The first group that makes up the DevOps team is the staff that come from the Quality Assurance (QA) and Operations department. These are people who volunteered to change their role and adopt DevOps practices in their everyday tasks. The second group is made up of people coming from the core development department, who develop the core of the application and build functionalities for specific purposes. Finally, the third group is made up of new hires, where the profiles sought are those that bring together the two worlds: Development and Operations.

Given that the company presents a deep-seated company culture after around 30 years in the market and a low rotation of personnel, the composition of the team is clearly not trivial. Two groups of workers represent the two cultures in DevOps and the two departments involved in the effort: software development and IT operations. A set of people from outside (but with relevant experience in the field) acted as the glue between them. This third group is intended to work on looking at both sides, mentoring the culture shift process.

Another characteristic mentioned by respondents is the technical, but also business, nuance presented by team members. This aspect is crucial to reduce cycles and improve the overall process. This requires balanced professionals able to understand business requirements and transform them into systems in an agile way. This facet is central for respondents and has been implemented in the new DevOps teams. Some of the people enrolled in the team have notable experience in the company, leading to a huge knowledge of the business. Some others have relevant experience in new DevOps tools and environment. This is makes a perfect match in the panoply of knowledge and experience gathered in the team.

DevOps is also about knowledge networks. In the pure Dev side, DevOps is agile. In this specific case study, several features of knowledge teams should be highlighted. The first was organizational support. The initiative was sponsored by top managers who gave their support to the team: all the necessary infrastructure for communication was implemented, and the creation of a community of practice was also facilitated by the company. The second aspect is networking. The low turnover and good personal relationships in the team led to good networking among members and a good integration of new members in the team. The third aspect is the team culture. Team culture was inherited from organizational culture. Organizational culture is strong and supports the integration of new personnel and ideas that count on the support of relevant people inside the structure. This was the case of DevOps.

In sum, regarding people aspects, the selection of the right internal resources, combined with new hires, the mix of business and technical skills, the support of relevant internal leaders and the formation of knowledge networks led the DevOps initiative to a moderate success inside the organization.

6. Conclusions

This case study illustrates the use knowledge management tools in the adoption of DevOps practices by a traditional software vendor as a way of efficiently integrating development, delivery and operations of cloud solutions. DevOps adoption drives a challenging cultural shift towards collaboration and knowledge-sharing between software development, quality control and operations. In this sense, several conclusions can be drawn from this case study. The need for implementing DevOps emerged when Meta4 moved from on premise products to service-oriented cloud solutions. It was a solution to some of the issues regarding cloud solution full functionality.

First, the pressures towards the adoption of DevOps are both internal and external. For instance, technological up to date, availability of technology, reported effectiveness and benefits were identified as the main external pressures. At the same time, DevOps adoption proves able to improve cycle times and quality. However, respondents suggest that although effects at a company level are limited, micro-service architecture could extend the benefits of DevOps beyond the technical side.

Second, given that DevOps is an approach to improve software processes based on an agile culture and knowledge-sharing, companies are able to embrace DevOps principles by using a widespread of knowledge-based tools. Indeed, the results from this study show that DevOps is more a cultural shift for IT than a process tools shift.

Finally, with regard to the DevOps adoption team, the results show that outside collaborators filled the gap between two traditionally independent departments: software development and IT operations. At the same time, top management support was crucial for boosting the networking between people with notable experience in the company and people of relevant experience in new DevOps tools.

This work fosters the empirical research and documentation of the lessons learned from companies that have adopted DevOps methods as well as the challenges and perceived paybacks of DevOps in terms of cycle times, overall quality and cost-benefits.

Acknowledgment

We would like to thank Fundación CajaMurcia for the support provided.

References

Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P., & Sabbagh, M. (2011). Software product evolution for intellectual capital management: The case of meta4 PeopleNet. *International Journal of Information Management, 31*(4), 395–399. http://dx.doi.org/10.1016/j.ijinfomgt.2011.04.001.

Del Giudice, M., & Della Peruta, M. R. D. (2016). The impact of IT-based knowledge management systems on internal venturing and innovation: A structural equation modeling approach to corporate performance. *Journal of Knowledge Management, 20*(3), 484–498. http://dx.doi.org/10.1108/JKM-07-2015-0257.

Ebert, C., Gallardo, G., Hernantes, J., & Serrano, N. (2016). DevOps. *IEEE Software, 33*(3), 94–100. http://dx.doi.org/10.1109/MS.2016.68.

Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software, 123*, 176–189. http://dx.doi.org/10.1016/j.jss.2015.06.063.

Humble, J., & Molesky, J. (2011). Why enterprises must adopt devops to enable continuous delivery. *Cutter IT Journal, 24*(8), 6.

Lwakatare, L. E., Kuvaja, P., & Oivo, M. (2015). *Dimensions of DevOps. Agile processes in software engineering and extreme programming*. Cham: Springer212–217. http://dx.doi.org/10.1007/978-3-319-18612-2_19.

O'Connor, R. V., Elger, P., & Clarke, P. M. (2017). Continuous software engineering—A microservices architecture perspective. *Journal of Software: Evolution and Process*. http://dx.doi.org/10.1002/smr.1866 n/a–n/a.

Ravichandran, A., Taylor, K., & Waterhouse, P. (2016a). *DevOps and real world ROI. DevOps for digital leaders*. Berkeley, CA: Apress139–150 Retrieved from https://link.springer.com/chapter/10.1007/978-1-4842-1842-6_9.

Ravichandran, A., Taylor, K., & Waterhouse, P. (2016b). *DevOps in the ascendency. DevOps for digital leaders*. Berkeley, CA: Apress3–14 Retrieved from https://link.-springer.com/content/pdf/10.1007/978-1-4842-1842-6_1.pdf.

Wettinger, J., Andrikopoulos, V., & Leymann, F. (2015). Automated capturing and systematic usage of DevOps knowledge for cloud applications. *IEEE international conference on cloud engineering*, 60–65. http://dx.doi.org/10.1109/IC2E.2015.23.

Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). DevOps and its practices. *IEEE Software, 33*(3), 32–34. http://dx.doi.org/10.1109/MS.2016.81.

Ricardo Colomo-Palacios Full Professor at the Computer Science Department of the Østfold University College, Norway. Formerly he worked at Universidad Carlos III de Madrid, Spain. His research interests include applied research in information systems, software project management, people in software projects, business software, software and services process improvement and web science. He received his PhD in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as Software Engineer, Project Manager and Software Engineering Consultant in several companies including Spanish IT leader INDRA. He is also an Editorial Board Member and Associate Editor for several international journals and conferences and Editor in Chief of International Journal of Human Capital and Information Technology Professionals.

Eduardo Fernandes received his Ph.D in Physical Sciences from the European Space Agency (ESA) / Universidad Complutense de Madrid (1995). He is currently Chief Architect and Technology Product Manager at Meta4. In this position he is responsible for strategic decisions which affect the Meta4 product line, for example PeopleNet. His main role is to analize technological market trends and technological frameworks to merge them to Meta4 products. Before reaches meta4 he was responsible for the development software team for the MiniSat-01 (ESA), the first Spanish mini-satellite (1997). While working on his master in physics he was responsible for the automation of the biggest telescope in Brazil, deeply involved in both software and hardware developments (1992).

Pedro Soto-Acosta is a Full Professor of Management at the University of Murcia (Spain). He attended Postgraduate Courses in Management at Harvard University (USA) and received his PhD in Business Economics from the University of Murcia. He serves as Associate Editor and Senior Editor for several mainstream journals including Decision Sciences, Electronic Commerce Research and Applications, Electronic Markets, Information Systems Management, Journal of Knowledge Management, Journal of Strategic Information Systems, and Computational Economics. His work has been published in journals such as Computers in Human Behavior, European Journal of Information Systems, Journal of Business Research, and Technological Forecasting and Social Change, among others. Further information is available at http://webs.um.es/psoto.

Xabier Larrucea holds a Ph.D, PMP, Executive MBA, and a computer engineering degree. He is a senior project leader and research scientist at Tecnalia and a part-time lecturer at the University of the Basque Country. His research focuses on areas such as safety-critical software systems, software quality assurance, software process improvement, empirical software engineering, metamodeling and technology strategy.