ARTICLE IN PRESS

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

Please cite this article as: Colomo-Palacios, R., International Journal of Information Management (2017), https://doi.org/10.1016/j.ijinfomgt.2017.11.005

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