

Management Challenges for DevOps Adoption within UK SMEs

Stephen Jones Norwich Business School University of East Anglia Norwich, United Kingdom
stephen.j.jones@uea.ac.uk

Joost Noppen School of Computing Sciences University of East Anglia Norwich, United Kingdom
j.noppen@uea.ac.uk

Fiona Lettice Norwich Business School University of East Anglia Norwich, United Kingdom
fiona.lettice@uea.ac.uk

ABSTRACT

The DevOps phenomenon is gathering pace as more UK organisations seek to leverage the benefits it can potentially bring to software engineering functions. However substantial organisational change is inherent to adopting DevOps, especially where there are prior and established methods. As part of a wider piece of doctoral research investigating the management challenges of DevOps adoption, we present early findings of a six month qualitative diary study following the adoption of DevOps within a UK based SME with over 200 employees. We find that within our case study organisation, the DevOps approach is being adopted for the development of a new system used both internally and by customers. DevOps, conceptually, appears to be generally well regarded, but in reality is proving difficult to fully adopt. This difficulty is down to a combination of necessity in maintaining a legacy system, lack of senior management buy-in, managerial structure and resistance. Additionally, we are finding evidence of job crafting, especially with the software developers. Taken together, we put forward the argument that DevOps is an interdisciplinary topic which would greatly benefit from further management and potentially psychology oriented research attention.

Keywords

DevOps, Management Challenges, United Kingdom, SME, Case Study

1. INTRODUCTION

Traditional and modern (agile) software development methods tend to focus solely on the software development teams. In either case, once software is developed, it is typically passed to the IT operations team, who assume responsibility for its deployment, ongoing maintenance and support. This silo approach can introduce organisational problems, including a blame culture between both functions [8, 9, 10], communication difficulties [1] and delays in producing software updates [2, 6].

To mitigate, industry is increasingly moving towards integrating both the software development and IT operations functions. This integration is at the very heart of the DevOps approach, which emphasises a culture of collaboration through the harmonisation of the software development and IT operations functions [4, 8, 6]. Such integration has the aim to facilitate continuous deployment, which is the immediate implementation of new code and functions within software products and business information infrastructure [7, 3].

Our research seeks to define how company and management structure influences the support of DevOps in the delivery of high quality software systems and vice-versa. Within this paper, we present the early findings of a case study exploring the adoption of DevOps within the software

development function of a UK based SME with between 200 and 250 employees, and what this means from a business management perspective.

2. BACKGROUND AND CASE STUDY

Our case study focuses on a UK based SME with just over 200 employees. For the purposes of anonymity, we refer to our case study business as “the organisation”. Our case study focuses on the activities of the organisation’s soft- ware development and IT operations teams. Included in the case study focus are two systems. The first is the organisation’s legacy system, and is of critical importance, relied upon both internally, and as an eCommerce platform for customers. The second system, currently being developed, is the replacement for the legacy system.

The initial development of the legacy system was out- sourced, although the organisation did have one in-house software developer. The legacy system was generally developed and maintained in a haphazard manner, resulting in a highly coupled, monolithic and poorly documented code base. Taken together with the continued use of dated and in some cases, deprecated technologies, the state of the legacy system code base has had knock-on development and organisational effects. For instance, the development and deployment of updates often proves tedious, consuming substantial development time, and is often accompanied by lengthy de- lays and downtime. In particular, the downtime is concerning for the organisation, given their reliance on the system as part of their operation. In taking together these issues and increasing demands from both internal users and customers, the organisation has deemed the legacy system as no longer fit for purpose, requiring the development of a complete replacement.

In implementing the decision to develop a new system, the organisation was keen to avoid encountering the quality issues they did with the legacy system. These issues include the aforementioned monolithic architecture and a large amount of bugs, but also, very long delivery cycles with extended periods of downtime during deployment, in turn harming the business. The organisation recruited a software development manager, who in turn created a team of seven software developers and one test analyst, settling on a DevOps approach to their development in order to address these issues. All were recruited based on their experience with agile development methods, in particular, Scrum, and understanding of various technologies, namely, .NET, C#, Azure1 and JavaScript.

A number of tools and technologies are in use at the organisation with the development infrastructure heavily Atlas- sian2 based, with Bitbucket, HipChat and Jira. In addition, Jenkins is used to automate unit tests, with the anticipation to aid in continuous deployment further down the line. With Azure’s continuing development3, the software development manager also anticipates the use of containers for sandboxing software deployments and to aid in continuity where any deployment was to fail.

The software development team is based at the organisation’s headquarters and are in close proximity to two systems administrators, who form the organisation’s IT Operations team.

3. METHODOLOGY

To understand the DevOps adoption at the organisation, we are undertaking a qualitative diary study with the soft- ware development and IT operations teams. Our study commenced in late February 2016, and will span at least six months.

The diary study utilises the same tools the organisation is using for its software development activities. In particular, Bitbucket and the git protocol is being used to submit weekly diary entries

written in markdown. The repository is private, and is a useful platform for the collection and tracking of qualitative data. The diaries themselves are open reflections, and a set of guiding questions are provided to aid each participant.

We are supplementing the diary study with semi-structured interviews at the beginning, during and at the end of the study period. Our objective for the interviews is to probe diary entries so as to extract more information.

Additionally, participant attrition is an ongoing concern and with such a longitudinal study, the interviews therefore also serve as a useful control mechanism for diary participation and a potential alternative data collection method if necessary. The process is illustrated in figure 1.

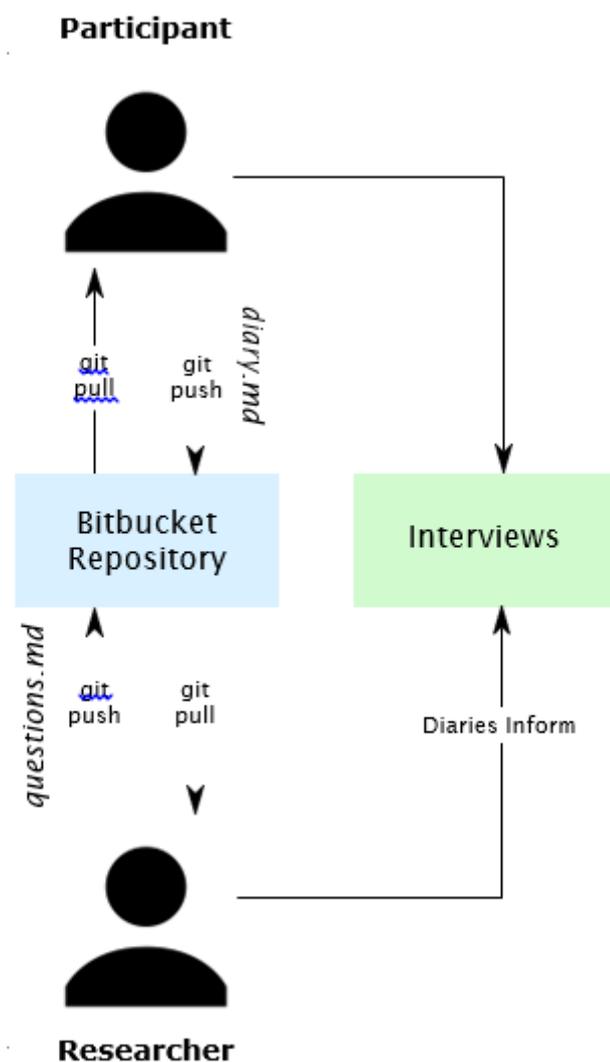


Figure 1: Process for diary study and interviews.

In analysing the data, we take both the submitted diary entries and interviews. We seek to identify emerging practises and trends, drawing comparisons with what has been reported in the literature. Finally, we interpret the findings in order to produce conclusions.

4. INITIAL FINDINGS

Our initial findings are based on the entrance interviews, and 29 diary entries from the first month of the study. We structure our initial findings based on the observations from the data, specifically focusing on the adoption of DevOps at the organisation.

4.1 Adoption of DevOps

The organisation's adoption of the DevOps approach is primarily based on the software development manager's personal knowledge and organisational objectives for the new system. In his interview, the software development manager expressed that he feels responsibility for making the new system a reality, despite senior management having little interest in the actual methods employed. Yet, the organisation is keen to ensure the new system is of high quality, easily maintainable, with the ability to rapidly deploy new features and updates. Such objectives sit well with the software development manager, who has taken methodological steps to ensure the development of the new system is better managed.

Given the critical operational nature of the legacy system, it has been necessary for the organisation to run it in parallel with the development of the new system. Consequently, this necessitates time being allocated to maintaining the legacy system, in addition to the new system development activities. The entrance interviews revealed that every member of the software development team expressed a strong dislike to dealing with the legacy system given its poor overall quality, highly coupled nature and lack of available documentation. While the maintenance of the legacy system is now undertaken in an agile manner, with defined sprints, the initial in-house developer described the previous approach as "just something" and "not even being waterfall".

Of more concern is the limitation of the legacy system to deprecated technologies potentially harming the new system development due to the need for developers to "switch" between technologies. For deployment, a lengthy and manual process has to be undertaken, which results in the legacy system being unavailable for a number of hours, causing significant disruption to the organisation. Deployment downtime is greatly exacerbated when the legacy system deployment fails.

The software development manager envisages a DevOps approach, where both the software developers and systems administrators work in close collaboration as part of an integrated whole. In particular, he believes the organisation will substantially benefit from the implementation of continuous deployment, meaning updates to the new system code base can be rapidly tested and deployed to the release version while still maintaining a good level of quality of service (QoS) such as minimising downtime, but increasing speed and frequency of deploying new features and fixes. The software development manager strongly believes that such a working arrangement builds on the Scrum method being used for the new system development, thus echoing statements in the literature suggesting DevOps and traditional development approaches are incompatible [5].

The overall challenge of the DevOps adoption is further compounded for the new system given the necessity for both the software developers and systems administrators to learn new technologies, tools and methods. This is in addition to the ongoing maintenance sprints for the legacy system. This learning and acquisition of new skills is by no means an issue from the developer's perspective. All of the developers revealed during the first interview that exploring new technologies and methods was part of the role and something they enjoyed. Instead, the challenge focuses on the psychological aspects of having to "switch" or "interrupt" the new system development and learning with maintenance of the legacy system. One developer describes this challenge as "the biggest they face"

and metaphorically refers to switching between the legacy system and the new system as “having to put different hats on”. In all the entrance interviews, each developer expressed high preference to working on the new system, with developers commenting that they were “horrified” by the quality of the legacy system code base, and that you can’t do incremental changes when working on the legacy system, once you pull something out, it is like tugging on threads and it all starts to unravel”.

4.2 Management Structure and Resistance

We have reported our observations that the software development manager is effectively championing the organisation’s adoption of a DevOps approach with the new system. While enthusiastically promoting the approach, this work is undermined by a lack of business analysis. Such a crucial gap in skill for the organisation is frustrating for the software development manager, forcing him to undertake this role, despite him requesting the organisation hires someone especially for it. As a result, this represents a threat to the effectiveness of the DevOps approach being undertaken, and limits the extent to which the software development manager can support the DevOps associated changes, and individual employees in both the software development and IT operations functions.

Previously, we outlined that the organisation’s IT Operations team is made up of two systems administrators, both versed in Microsoft environments. The first (Sysadmin A) has just embarked on their career, while the other (Sysadmin B) is late career and is respected by Sysadmin A, despite being considered “old school”.

The adoption of DevOps at the organisation appears to be a bottom up process, led by the software development manager, thus senior management have yet to be convinced on the benefits it brings. In addition, there appears to be resistance from IT operations, who consider their roles to have nothing in common with those of the software developers.

This is further compounded by the management structure (see figure 2), where both the IT operations team and Software Development Manager report to the Head of Group Operations.

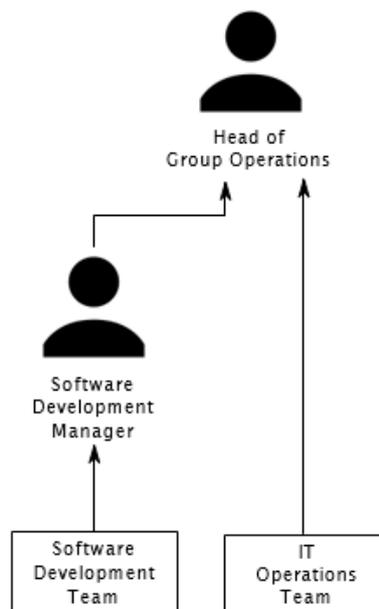


Figure 2: Current management structure for software development and IT operations within the organisation.

This management structure facilitates disruption caused by IT operations resistance, which has occurred on several occasions. For instance, concerns and objections to using Azure were raised by one systems administrator, circumventing the software development manager. Such disruption has also trickled to the software developers, as evidenced in the diaries with growing frustrations and perceptions that the development team are having to do everything. Given the organisation's observed traditional hierarchical management structure, this 'IT operations loophole' needs to be closed. From this observation, it is clear that in order for a DevOps approach to work in the organisation, both the IT operations and software development teams need to report to the same line manager (see figure 3).

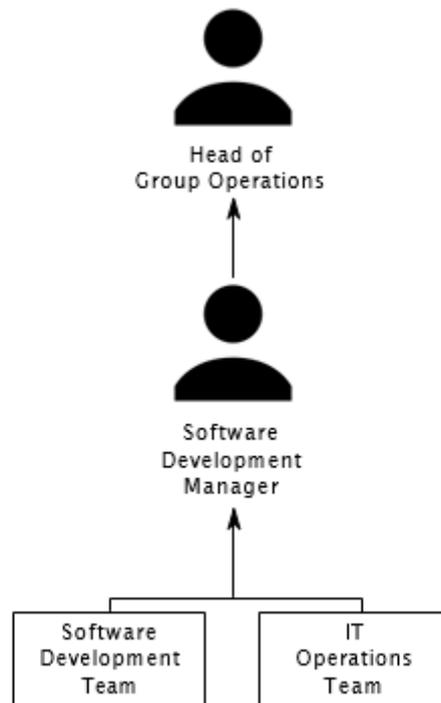


Figure 3: Suggested management structure for software development and IT operations within the organisation.

Another factor observed is the perceived seniority of Sysadmin B, given their longer career span. Despite their enthusiasm, Sysadmin A perceives himself as “a learner” and that Sysadmin B is “the systems administrator” at the organisation because he possesses greater experience. This perception of seniority and expertise also appears to have influence on the Head of Group Operations, who is, by his own admission, a non-technical person.

Of course, the limitation in these findings is that they are contextually specific, and may not apply to other organisations, especially where management structures, employee attitudes and skill sets differ.

4.3 DevOps - A Social Phenomenon

The issues observed at the organisation greatly portray DevOps as much a social and cultural phenomenon as it is technical one. The entrance interviews with both systems administrators revealed perceptions that their roles do not go beyond end-user support and hardware maintenance. During the initial interviews, Sysadmin A expressed desire to acquire additional

technical skills, especially with using a command line and git. Indeed, the Software Development Manager has arranged additional training for IT Operations with Microsoft PowerShell and git which Sysadmin A has taken up. Coupled with the available diary reflections, this could be interpreted as a management strategy to harness Sysadmin A's enthusiasm and career aspirations in order to get IT operations to take a more involved role in Azure PowerShell work and the new system's deployment activities. The inherent danger here is the inadvertent formation of a silo within IT Operations as both systems administrators, at this time, appear to have very different agendas for their career progression and beliefs as to what their roles encompass.

Conversely, in a diary reflection, Sysadmin B considers the use of a command line "was a dev responsibility as it involved coding", and as such is outside of the remit of his role. Indeed, the diary entry quotes that Sysadmin B considers his IT Operations role is just "supporting end users queries on software we develop plus supporting the hardware it runs on. The setting up of a VM or any configuration of a web server or database is the responsibility of development."

Taking together the entrance interviews and committed diary entries so far, there is already evidence that software development and IT operations silos do exist within the case study organisation. Indeed, these silos can, and do, present substantial problems to organisations. Despite DevOps seeking to harmonise both functions, and in turn, break apart these silos, adopting DevOps appears to be a challenge in itself for our case study organisation, especially given the observed management structure and IT Operations resistance.

5. RELATED WORK AND FUTURE RESEARCH AGENDA

The literature is growing, with many case studies now published of various tools and approaches associated with DevOps. Work has been published examining the impact of technical and social debt associated with continuous deployment [3]. However, our work differs given it is not just focused around continuous deployment, as it also seeks to explore the various managerial and business issues surrounding DevOps adoption.

We are undertaking a second diary and interview based study in a different UK organisation which will begin in June 2016 so as to draw a comparison between two UK organisations adopting the DevOps approach. Both case studies will be analysed and compared against each other, as well as key findings from a systematic review of the growing DevOps literature. The overall objective of this research is to identify patterns of management and social challenges, and to identify best practises.

Job Crafting is defined as "the physical and cognitive changes individuals make in the task of relational boundaries of their work" [11, 179]. It follows, therefore, that these tasks are at the core of the employee-employer relationship, and job crafting is about the shaping of the task boundaries, be it physical or cognitive, and/or the relational boundaries. Three specific types of job crafting are put forward: Task, Relational and Cognitive [11]. Task crafting refers to job crafting where work is completed in a more timely fashion and where a job's meaning can change so the employee becomes a guardian or mover of a project. Relationship job crafting occurs where employees see their job as a vital part of an integrated whole. Finally, cognitive job crafting is where employees change their perception of work to not just be about delivering high quality outputs.

By its nature, job crafting is a phenomenon often studied with a longitudinal approach. We seek to better understand the social and cognitive aspects of DevOps adoption within our case study organisation. Our aim is therefore to build upon the early findings presented in this paper by pursuing further research using a job crafting theoretical lens.

6. CONCLUSION

In this paper, we have presented the initial findings of an in-depth and ongoing diary study exploring the adoption of DevOps within a large, UK based SME. Presently, we conclude that DevOps is very much an interdisciplinary topic. The management structure of the organisation needs to be addressed, as presently, the resistance being exhibited by IT Operations is undermining the software development manager's remit. From a quality standpoint, the necessity of maintaining legacy systems is one of the biggest issues the developers face. Moreover, this interrupts their learning of new technologies, pertinent to the new system, and as such, could introduce quality issues with the new system's code base.

Our research is still in the early stages, yet has already yielded interesting management and software engineering insights. Furthermore we believe that DevOps has a substantial management component, and is in desperate need of further business management related research focus to fully explore the phenomenon.

7. REFERENCES

- [1] L. Bass, R. Jeffery, H. Wada, I. Weber, and L. Zhu. Eliciting operations requirements for applications. In Proceedings of the 1st International Workshop on Release Engineering, pages 5–8. IEEE, 2013.
- [2] L. Chen. Continuous delivery: Huge benefits, but challenges too. *Software, IEEE*, 32(2):50–54, 2015.
- [3] G. G. Claps, R. B. Svensson, and A. Aurum. On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software Technology*, 57:21–31, 2015.
- [4] N. Cook, D. Milojcic, and V. Talwar. Cloud management. *Journal of Internet Services and Applications*, 3(1):67–75, 2012.
- [5] S. Hosono and Y. Shimomura. Application lifecycle kit for mass customization on paas platforms. In *Services (SERVICES), 2012 IEEE Eighth World Congress on*, pages 397–398. IEEE, 2012.
- [6] S. W. Hussaini. Strengthening harmonization of development (dev) and operations (ops) silos in it environment through systems approach. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 178–183. IEEE, 2014.
- [7] Y. Liu, C. Li, and W. Liu. Integrated solution for timely delivery of customer change requests: A case study of using DevOps approach. *International Journal of U- & E-Service, Science & Technology*, 7(2):41–50, 2014.
- [8] M. Loukides. *What is DevOps?* O'Reilly, 2012. Sebastopol, USA.
- [9] A. Tseitlin. The antifragile organization. *Communications of the ACM*, 56(8):40–44, 2013.
- [10] M. Walls. *Building a DevOps Culture*. O'Reilly, 2013. Sebastopol, USA.
- [11] A. Wrzesniewski and J. E. Dutton. Crafting a job: Revisioning employees as active crafters of their work. *Academy of Management Review*, 26(2):179–201, 2001.