Omniscient DevOps Analytics

Damian A. Tamburri[1], Dario Di Nucci[2]
Lucio Di Giacomo[3], and Fabio Palomba[4]

[1]TU/e - JADS, The Netherlands – [2]Vrije Universiteit Brussels, Belgium
[3]Guardia di Finanza – Comando Regionale Trentino Alto Adige, Italy
[4]University of Zurich, Switzerland

Abstract. DevOps predicates the continuity between Development and Operations teams at an unprecedented scale. Also, the continuity does not stop at tools, or processes but goes beyond into organisational practices, collaboration, co-located and coordinated effort. We conjecture that this unprecedented scale of continuity requires predictive analytics which are **omniscient**, that is, (a) transversal to the technical, organisational, and social stratification in software processes, and (b) correlate all strata to provide a live and holistic snapshot of software development, its operations, and organisation. We elaborate on this conjecture and illustrate it with an example scenario.

1 Idea Outline

Omniscient - [om-nish-uh nt], adjective — *"having complete or unlimited knowledge, awareness, or understanding; perceiving all things."* — [Cit. Oxford Dictionary]

DevOps is a set of practices aimed at accelerating the lead-time between a change and its operational availability to end-users [1]. Since its early inception, DevOps has radically shifted the way of conceiving software processes as well as the production of software artefacts; sample DevOps practices include the incremental or radical intermix of Dev- and -Ops professionals in the same team, using application lifecycle automation tools to enable continuous delivery, or designing failure-first software architectures to learn a "proper" architecture from runtime operations monitoring. Nevertheless, tracking and evaluating the effective monetary, technical, organisational, and social gains connected to any single DevOps practice over others, or compared to the previous way of working is still a challenge for several reasons. For example, there is still no definite way to shift from classical software engineering practices to DevOps lest by incurring in great costs[1] that cannot be estimated up-front. Moreover, there is still a lack of actionable metrics to measure the impact, risks, and gains connected to every single practice let alone any of their combinations. Conversely, a proliferation of *ad-hoc* measurement and monitoring solutions exist mainly configured to sustain each organisational scenario in each company.

[1] `https://jaxenter.com/true-cost-devops-adoption-138287.html`

We argue that, given the scope of its proclaimed shifts, DevOps also deserves a radical shift in the means and mechanisms that software people need to employ in tracking DevOps pains and gains. In fact, DevOps practices altogether aim at improving all aspects of software production, operation, and evolution, with small, and steadfast devices of technical (e.g., tools), organisational (e.g., co-operation practices), and social (communication practices) nature. Secondly, achieving DevOps involves a shift of culture towards *failure* — products should fail fast, fail observably, fail quantifiably, fail safely, and more. In summary, we observe that DevOps: (a) entails changes along all possible activities in all possible layers around software (see Fig. 1), and (b) focuses on failure. From this observation we conclude that DevOps requires measurable, fine-grained, complete, constant awareness over every*thing* and every*one*, to achieve an understanding of what dimension (social, organisational, or otherwise) is influencing what else, for the purpose of constant, continuous improvement.

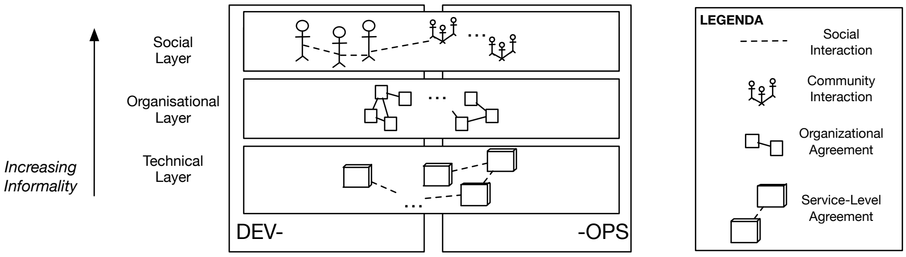

Fig. 1. Omniscient DevOps Analytics - a social layer accounts for the people and communities to be monitored; an organisational layer for the organisational agreements and protocols to be tracked; a technical layer for the product variables to be observed.

We refer to the above awareness condition with the term **omniscience**, and argue that omniscience can be achieved via specific predictive analytics [2] frameworks that: (a) are able to constantly analyse sensory data over all contracts, agreements, and measurable quantities from layers in Fig. 1; (b) constantly and statistically relate the elicited analyses at multiple levels of granularity (e.g., person vs. team vs. unit vs. organisation ...) and transversally to all layers.

The ultimate goal of such omniscient DevOps analytics toolkits shall be to sustain the perpetual improvement cycles at the basis of the DevOps philosophy, offering at every step, a clear overview over what needs to be improved along which measurable improvement dimension, and in which layer.

References

1. Bass, L., Weber, I., Zhu, L.: DevOps: A Software Architect's Perspective. SEI Series in Software Engineering. Addison-Wesley, New York (2015)
2. Yang, Y., Falessi, D., Menzies, T., Hihn, J.: Actionable analytics for software engineering. IEEE Software **35**(1) (2017) 51–53