

A Vision of Experimental Process Improvement

Ingo Weber¹ and Jan Mendling²

¹ NICTA, Sydney, Australia*, `firstname.lastname@nicta.com.au`

² Wirtschaftsuniversität Wien, Vienna, Austria, `firstname.lastname@wu.ac.at`

Abstract. Software systems that support Business Process Management are in wide-spread use. They play an important role in facilitating process automation and process improvement. Yet, there is hardly any insight into whether the implementation of a *supposedly improved process model* leads to an *actual improvement in the process*. The research problem this paper addresses is: how can we determine if a new variant of a process model is an improvement over a previous variant, with respect to relevant measures? To this end, we suggest to build on recent software engineering concepts from the DevOps movement, and to develop novel techniques that provide the infrastructure for assessing in how far a specific business process change manifests an improvement.

1 Introduction

Business Process Management Systems (BPMSs) provide an infrastructure for process automation, integrating human tasks and software services. One of their advantages is that process improvements can be quickly put into operation. A challenge is, however, that there is currently no support to test the often implicit assumption that a modification of a process actually represents an improvement. The point that not all improvement ideas lead to positive effects is underlined by a study of changes to Microsoft’s website: only one third of the ideas had positive and significant effects, another third no impact, and one third was even harmful[13]. The lack of support for testing redesign hypotheses is problematic. Given the current architecture of BPMSs, it is not possible to conduct a fair comparison between the old process and the new process since they are not operational at the same point of time. That means, doing post-hoc analysis of data generated from the old process being operational in time interval $[t(n - 1), t(n)]$ and the new process running from $[t(n), t(n + 1)]$ is biased towards the respective conditions of each time interval.

Concepts from DevOps, which aim to bring software development (Dev) and operations (Ops) closer together, could help to address this problem. “DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality” [3]. To ensure high quality, DevOps employs various testing methods. One of these is *A/B testing*, where two variants (*A* and *B*) are both deployed and receive a share of the production workload while being monitored closely.

* NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

The monitoring data is then used to compare the effectiveness of the two variants, for instance in the form better conversion rates. How to apply methods like A/B testing to process improvement has not been studied yet.

2 State of the Art

The research laid out in this paper relates to four areas of research on business process management, which we discuss in the following.

Process Improvement. Approaches for process improvement can be classified into analysis techniques, redesign techniques, and overall procedural models. These relate to various metrics of performance. The most prominent *analysis techniques*[8] include value-added analysis and techniques such as the cause-and-effect diagram. Once weaknesses are identified, methods for *designing an improved version* of a business process need to be utilized, for instance based on redesign heuristics or the work systems theory[17,2]. *General procedures* for process improvement cover prototyping of the new process, its simulation, and general systems testing techniques[11]. Note that none of these procedures explicitly tests the efficiency and effectiveness of the redesigned process, e.g., in terms of important performance dimensions for process improvement like time, cost, or flexibility[6].

Process Mining. The term *process mining* refers to various techniques for relating event logs and process models[1]. *Automated process discovery* is the act of deriving a process model from event logs. At the heart of process discovery are algorithms that generate process models which meet various quality criteria including fitness, simplicity, generalization and precision. The issue of variation is also a research topic in this area: techniques for discovering process variants have been investigated[5]. *Conformance checking* refers to techniques that assess if a given model fits an event log and vice versa. The main techniques for conformance checking are token replay and alignment. Few works exist around changes in models and process mining, e.g.[10]. Conformance checking of logs against partly fitting models, such as new process variants, requires *matching* activities of the process model with the events in the event log, which is related to work on process model similarity and matching[12].

Testing Processes. Classically, testing of executable workflows has been approached following a two-level paradigm: testing and verification focuses first on the business process, and second on the implementation of each of the activities [14]. The recent years have seen several contributions on testing processes, as surveyed in[4,7]. The focus of most reviewed works is on technical matters of *correctness*. A perspective that embraces *process improvement* is missing completely. Recently proposed concepts in software development, delivery, and operation are referred to under the umbrella term *DevOps*[3]. Automated testing plays a big role in implementing DevOps practices, because automated tests are fast to run and provide quality assurances. Executable business process models can be seen as a software artifact, and hence DevOps practices could be applied in principle. How to benefit from the characteristics of processes is an open question.

Experimental Research on Business Process Management. Recent years have witnessed a growing stream of experimental research on BPM. This

includes works on factors that influence the quality of a process specification. Research on *process modeling grammars* has investigated quality characteristics, actual use, and factors of comprehension[9,16]. The theoretical foundation of experimental research on process model comprehension are often found in theories of cognition[18]. Factors of comprehension performance include, e.g. model complexity and ontological deficiencies[15,16]. Experimental research that investigates the effectiveness of process improvements is still scarce.

3 Approach

A core idea in this paper is to make A/B testing applicable for whole process variants. This is, however, not a straight-forward task and comes with various challenges. First, general A/B testing methods consider a black-box model of processing: requests are received, responses emitted, metrics are collected, and conclusions are drawn. The behavioural properties of the process variants (process models A and B) such as concurrency and exclusiveness of activities can help to better understand the variation in usage. Simulating the execution of a new variant is also feasible for a known artifact, i.e., process model variant B, in contrast to the original variant A. Second, treating the process variants as black boxes usually requires a duplication of the whole application stack: one whole stack with layers like the database tier, business logic tier, and the Web tier for A, and one whole stack for B. This increases complexity, cost, time, and resources for both configuration of isolated stacks and for operating them. In contrast, two process variants could be deployed into the same BPMS with efficient traffic splitting – thus requiring only a single stack. Third, by knowing the process models it is possible to design monitoring solutions, metrics collection and aggregation, and analytical methods on a generic level, for all kinds of process models. This may lead to reduced overhead in A/B testing: a suite of monitoring and analytic methods can be provided, fitting many models. It also enables the design of a suite of methods based on sound empirical methods for experimental design. The research problem is that concepts and techniques supporting these ideas are missing, as are insights into their merits.

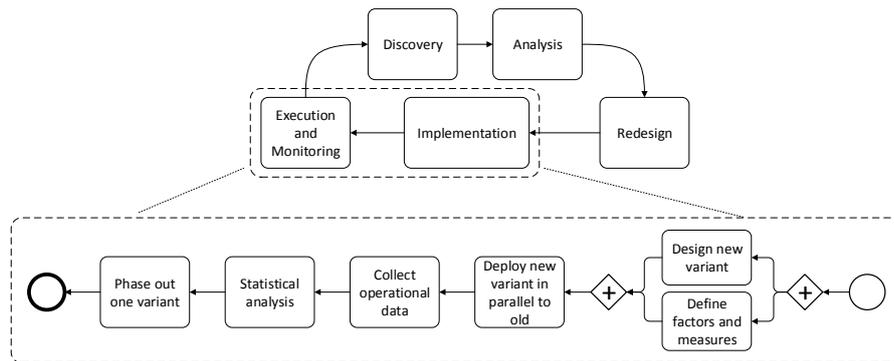


Fig. 1. Extended BPM Lifecycle

We suggest to address this research problem with a research framework for *experimental process improvement*. We believe this can be achieved by extending the BPM lifecycle and the respective support of BPMSs with capabilities to run process variants as part of a proper experimental design. [Figure 1](#) shows a lifecycle model that integrates possible extensions in the implementation phase. Research on this topic can build on three pillars: business process management, software testing, and empirical software engineering. Possible contributions include a novel approach for comparatively testing the efficiency and effectiveness of executable business process variants, both in simulation and operation, according to explicit design hypotheses using sound statistical methods.

References

1. van der Aalst, W.M.P.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer (2011)
2. Alter, S.: *Work system theory: overview of core concepts, extensions, and challenges for the future*. *Journal of the Association for Information Systems* (2013)
3. Bass, L., Weber, I., Zhu, L.: *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional (2015)
4. Bozkurt, M., Harman, M., Hassoun, Y.: *Testing and verification in service-oriented architecture: a survey*. *Software Test., Verif. and Rel.* 23(4), 261–313 (2013)
5. Buijs, J.C., Reijers, H.A.: *Comparing business process variants using models and event logs*. In: *EBPISM*. pp. 154–168 (2014)
6. del-Río-Ortega, A., Resinas, M., Cabanillas, C., Cortés, A.R.: *On the definition and design-time analysis of process performance indicators*. *Inf. Syst.* 38(4) (2013)
7. Dogan, S., Betin-Can, A., Garousi, V.: *Web application testing: A systematic literature review*. *Journal of Systems and Software* 91(0), 174 – 201 (2014)
8. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*. Springer (2013)
9. Evermann, J., Wand, Y.: *Toward formalizing domain modeling semantics in language syntax*. *IEEE Transactions on Software Engineering* 31(1), 21–37 (2005)
10. Günther, C., Rinderle, S., Reichert, M., van der Aalst, W.M.P.: *Change mining in adaptive process management systems*. In: *OTM Conferences*. pp. 309–326 (2006)
11. Kettinger, W.J., Teng, J.T., Guha, S.: *Business process change: a study of methodologies, techniques, and tools*. *MIS quarterly* pp. 55–80 (1997)
12. Klinkmüller, C., Leopold, H., Weber, I., Mendling, J., Ludwig, A.: *Listen to me: Improving process model matching through user feedback*. In: *BPM* (2014)
13. Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J., Melamed, T.: *Online experimentation at Microsoft*. In: *Workshop on Data Mining Case Studies* (2009)
14. Leymann, F., Roller, D.: *Production workflow: concepts and techniques*. Prentice Hall (2000)
15. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: *Seven process modeling guidelines (7PMG)*. *Information and Software Technology* 52(2), 127–136 (2010)
16. Recker, J., Rosemann, M., Green, P.F., Indulska, M.: *Do ontological deficiencies in modeling grammars matter?* *MIS Quarterly* 35(1), 57–79 (2011)
17. Reijers, H.A., Mansar, S.L.: *Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics*. *Omega* 33(4) (2005)
18. Sweller, J.: *Cognitive load theory, learning difficulty, and instructional design*. *Learning and instruction* 4(4), 295–312 (1994)