Design of Dynamic Control Portfolios in Scrum Projects

Thomas Fischer[1][0000-0001-9891-7797], Alan Hevner [2][0000-0003-4953-3900] and René Riedl[1, 3][0000-0003-2468-5860]

[1] University of Applied Sciences Upper Austria, 4400 Steyr, Austria
[2] University of South Florida, Tampa, FL 33620, USA
[3] Johannes Kepler University, 4020 Linz, Austria
{thomas.fischer, rene.riedl}@fh-steyr.at
ahevner@usf.edu

In software development projects, control mechanisms are implemented to ensure that expectations related to software qualities and project management efficiencies are met [1]. These control mechanisms, which manifest in the form of development practices (e.g. burndown charts, daily stand-up meetings), are used to regulate the behavior of individuals in the development team [2]. Importantly, such control mechanisms do not operate in isolation, but are integrated in portfolios of controls, which comprise more formal control mechanisms (e.g. project plans, mile-stones), more informal control mechanisms (e.g. self-organized teams, rituals and ceremonies), or a mix of both [3].

As software development projects are not static, but subject to constant changes (e.g. in requirements or their priorities), so are the control portfolios employed during the lifetime of such a project [1, 4]. In recent reviews, Wiener et al. [1] and Cram et al. [5] highlight the paucity of research into the dynamic changes of the control portfolio over the project life cycle. To understand how the process of dynamic control portfolio adaptation works, we argue that it is important to know (i) what drives portfolio adaptations (i.e. triggers of change) and (ii) in what ways the portfolio can be adapted (i.e. types of change).

Regarding *triggers of change*, in their recent review, Cram et al. [5] present five main dimensions that can be used to analyze control in IS projects, including (i) the *control environment*, which roughly translates to control selection antecedents as discussed by Wiener et al. [1] (e.g. environmental uncertainty), (ii) *control outcomes*, which are related to project-level performance (e.g. software quality, delivery on time and/or in budget); (iii) *control experiences*, which are related to individual-level effects of control enactment; (iv) *control mechanisms*, which are related to the characteristics of control practices (e.g. their support of control modes); and (v) *control executions*, which are related to the enactment and adaptation of control in an IS project over time which is the focus of our research interest.

Regarding the *types of change*, we argue that adaptations of practices may be the first way to gradually evolve the overall control portfolio rather than more radical shifts towards a portfolio with totally different dominant control modes. This could be reflected in practices not being used in accordance with guides for a certain methodology, but rather in a deviant fashion.

Based on exemplary research [6, 7] and recent reviews [1, 5], we make seven testable propositions related to dynamic control portfolio adaptations:

- *Proposition 1*: Control mode support of a practice influences the type of change that can be applied to it, with practices supporting a wider range of control modes being more likely adapted than added or removed during the project lifecycle.
- *Proposition 2*: A balance between the benefits of having a large control portfolio (e.g. related to control mode support) and the efforts caused by maintaining this portfolio (e.g. number of meetings) is a main consideration for control portfolio adaptation.
- *Proposition 3*: Differences in control perceptions between controller and controlee roles can hamper the initiation of control portfolio adaptations and, in particular, produce more intrusive changes (e.g. practices being removed from the portfolio).
- *Proposition 4*: Control outcomes are easier to quantify (e.g. project performance) as compared to control experiences (e.g. developer motivation), which can lead to control outcomes having a stronger impact on portfolio adaptation.
- *Proposition 5*: The negative effects of an inappropriate control portfolio (e.g. on project performance or developer motivation) become visible only with a time delay (e.g. too much formal control will lead to negative effects on developer motivation over time, not instantaneously).
- *Proposition 6*: Each development practice in the control portfolio has unique characteristics that will facilitate or hinder adaptations
- *Proposition 7*: The chosen development methodology influences which practices are seen as mandatory parts of the portfolio and which ones are not.

Thus, if the control portfolio is adapted proficiently and in timely ways, we can assume that control outcomes as well as control experiences in the SD process will be balanced and a level of satisfaction on the controller's end (regarding control outcomes) and on the controlee's end (regarding control experiences) can be achieved. In future studies, we will focus on software development projects using Scrum as their main development methodology to investigate the presented propositions and the design processes that lead to effective and efficient control portfolio adaptations.

References

1. Wiener, M., Mähring, M., Remus, U., Saunders, C.: Control Configuration and Control Enactment in Information Systems Projects. Review and Expanded Theoretical Framework. MIS Quarterly 40, 741–774 (2016)
2. Kirsch, L.J.: The Management of Complex Tasks in Organizations. Controlling the Systems Development Process. Organization Science 7, 1–21 (1996)
3. Kirsch, L.S.: Portfolios of Control Modes and IS Project Management. Information Systems Research 8, 215–239 (1997)
4. Kirsch, L.J.: Deploying Common Systems Globally. The Dynamics of Control. Information Systems Research 15, 374–395 (2004)
5. Cram, W.A., Brohman, K., Gallupe, R.B.: Information Systems Control. A Review and Framework for Emerging Information Systems Processes. Journal of the Association for Information Systems 17, 216–266 (2017)
6. Harris, M.L., Collins, R.W., Hevner, A.R.: Control of Flexible Software Development Under Uncertainty. Information Systems Research 20, 400–419 (2009)
7. Harris, M.L., Hevner, A.R., Collins, R.W.: Controls in Flexible Software Development. Communications of the Association for Information Systems 24, 757–776 (2009)