

HUOM! Tämä on alkuperäisen artikkelin rinnakkaistallenne. Rinnakkaistallenne saattaa erota alkuperäisestä sivutuksestaan ja painoasultaan.

PLEASE NOTE! This is an electronic self-archived version of the original article. This reprint may differ from the original in pagination and typographic detail.

Käytä viittauksessa alkuperäistä lähdettä:

Please cite the original version:

Dahlberg, T. & Lagstedt, A (2021). Introduction to the Minitrack on Co-Development with Software and Business Engineering Frameworks and Methods. *Proceedings of the 55th Hawaii International Conference on System Sciences*, (7464–7465). HICCS. doi: 10.24251/HICSS.2022.895

© 2022 The Authors'. Licensed under Creative Commons license Attribution-NonCommercial-NoDerivatives 4.0 International <https://creativecommons.org/licenses/by-nc-nd/4.0/>

## Introduction to the Co-development with Software and Business Engineering Frameworks and Methods Mini-Track

Tomi Dahlberg  
Turku School of Economics at University of Turku  
FIN-20014 University of Turku  
Turku, Finland  
tomi.dahlberg@utu.fi

Altti Lagstedt  
Haaga-Helia University of Applied Sciences,  
Ratapihantie 13 FIN-00520 Helsinki  
Helsinki, Finland  
altti.lagstedt@haaga-helia.fi

The history of software and information systems (IS) development frameworks, methods and tools exceeds 60 years. Their purpose is to increase the efficiency and effectiveness of software and IS development. New or improved versions of existing frameworks, methods and tools are launched continuously, both plan-driven (e.g. waterfall) and change-driven (e.g. agile). Despite of this, the success rate of software/IS development projects has remained at the modest 35-40 % level with only a slight 10-15 % success rate increase compared to the 25-30 % level of the mid 1980s. Moreover, a significant amount of software, IS and other IT-enabled solution development projects are carried out without the rigorous use of any development framework, method and tools.

At the end of the day, the purpose of IS development is not only to design and implement an IS. Within IS development, software development initiatives are embedded into or related to business development initiatives. Similarly, to software and IS development methods, also business development methods evolve continuously. Lean management, service design and business modelling methods are examples. Taken together, the scope and focus of software, IS and business development could be on an organizational function or process or on a product or a service. The scope and focus could, however, also be enterprise-wide like digital transformations or even cross-enterprise like multi-enterprise digital platforms. As a consequence, software/IS development needs to be integrated with business process development, re-engineering or digitalization/digitization of existing processes; development of new digital products, services or businesses; digitization of existing products, services and businesses; or the development of other types software/IS-enabled incremental and revolutionary innovations. In the resulting activities and projects, it is often necessary to include and align different kinds of competences, frameworks, methods, tools and even mindsets in order to control and synchronize work done in various subprojects or activities.

The modest success rates of software/IS and other development projects encourage both the academic

community and practitioners to search for better answers to the needs of software and business engineering. As a whole, we need to increase our knowledge base and theoretical understanding about the interrelations between software, IS and business development during the planning, development and run (such as maintenance and operation) phases of their lifecycles.

Software, IS and business development projects vary widely regarding their problem or solution complexity, the number of involved persons and actors, their novelty, their context, and / or their importance, just to mention few of the potentially relevant characteristics. Software/IS development could be conducted inhouse, co-sourced or outsourced. In addition to that, the participation of internal and external business professionals, software users, internal and external IT professionals may change during the lifecycle of software and business development from requirements engineering to IS/software roll-outs and the post-evaluation of an IS/software. As a consequence, an IS/software user organization may not be able or willing to assess all possibilities and limitations of the various software, IS and business development methods.

To overcome the above described limitations of conceptual and theoretical understanding, the goal of this mini-track, organized for the second time at HICSS 2022, is to offer research contributions that open up new perspectives and insights for better co-development with software/IS and business engineering frameworks, methods and tools. From eight submissions four papers were accepted into this year's mini-track. They offer alternative perspectives to the themes of the mini-track. All contributions were selected after extensive peer reviews, three after suggestions for revisions were completed. On the basis of reviewer comments, we deem that also papers that were rejected have the potential to become published by taking into consideration reviewers' comments. Thus, we thank the authors and the reviewers of the eight submissions for job well done and for their valuable contributions.

In their paper *Selection, Adaption and Use of IS and Business Development Methods in Digitalization Projects*, the mini-track co-chairs Dahlberg and Lagstedt together with Kiselev and Kautz reported the findings of survey and interview data from joint research conducted in Finland and Switzerland. The Finnish survey study (n=272) revealed that perceived success in the management of IT and digital business as a part of business management was strongly related ( $P > .9999$ ) to IS project success. The importance of simultaneous business and software/IS development sub-projects was one of the findings of 17 interviews conducted in the two countries. The paper also proposed a hybrid method evaluation tool.

The second paper *Calculating the Costs of Inner Source Collaboration by Computing the Time Worked* by Buchner and Riehle applied the design science method of Peffers et al. (2007) to calculate time spent on code contributions for usage within various cost related business processes in a multi-national corporation. The Cost Plus method suggested by the OECD is extended with the objective to better solve fiscal, most notably taxation, and administrative challenges met in software collaborations across organizational multi-country boundaries. The paper discloses the design science steps of the new method proposed.

The third paper *Spotlight on the Positives: How Do Information Technology Projects Achieve Cost Underruns?* by Alami, Madsen and Krancher investigated 12 public sector IT projects in Denmark that had been able to deliver agreed outputs with lower than budgeted costs. Empirical data was collected with 15 interviews and document analysis. Six projects were executed with plan-driven (waterfall), four with change-driven (agile) and two with hybrid (waterfall and agile combined) methods. The paper reports two conditions and four practices discovered in the sample that were related to cost underruns.

The fourth paper *Leveraging Service Design by bridging business and process modeling* by Pérez Blanco, Vara, Gómez, De Castro and Marcos approaches the themes of the mini-track from the perspective of design science. This highly practice-oriented paper investigated how two business model notations (Business Model Canvas, e3 value) and three business process notations (Service Blueprint, Process Chain Network and BPMN) could be used in a complementary way in practical design science work. The paper proposed several suggestions for the development of a modeling tool kit to this purpose.