Obtaining Insights Into the Interplay Between Systems and Software Engineering

Alexandr Vasenev[1], Johan Lukkien[1], Laura van Veen[1], Pieter Goosen[1], Richard Doornbos[1], Arjan Mooij[1, 2]
[1]TNO-ESI, Eindhoven, The Netherlands; [2] Zürich University of Applied Sciences, Winterthur, Switzerland
alexandr.vasenev@tno.nl

***Abstract*—The high-tech equipment industry is adopting approaches like Model-Based Systems Engineering (MBSE) and Scaled Agile Framework (SAFe) to improve their practices. Instantiations of these approaches do not necessarily align across disciplines. Given the importance of software aspects in systems, the interplay between software and other engineering disciplines should be carefully addressed. Decision makers are interested in mitigating any risks in the collaboration of specialists in different disciplines. This short positioning paper proposes an applied research approach to identify such risks for later mitigation. The approach is inspired by grounded theory (we constructed concepts from interviews) and systems thinking. Work still to do is to cross-check identified concepts with literature. This research is in progress and we welcome discussions and critical opinions.**

***Index Terms*—systems and software interplay, model based engineering**

I. Introduction

High-tech systems, such as production printers, lithography machines, and in-hospital medical equipment become increasingly advanced and complex. Software contributes to this growth in terms of the size and complexity of the software base, the number of developers and the total time spent. Software addresses many purposes: firmware, feature realization, functional modules (like data analysis), integration into customer infrastructure, connections to information systems, and 3rd party applications. This requires a good interplay between software engineering (SWE) and systems engineering (SE) disciplines within a company. This short paper proposes an approach developed and adopted in a running project to answer the research question: *How can high-tech companies obtain insight into the interplay between SE and SWE?* such as to identify risks. The methods relies on surfacing (common) concerns of professionals to decision makers. Outcomes of applying the method help to address the larger question of "How to improve the interplay of systems and software engineering?"

II. Current practices and trends

Improving the interplay of SE and SWE is a highly relevant topic for industrial companies and researchers. The International Council on Systems Engineering (INCOSE) recognizes topic's importance in forming several Working Groups to address it: Agile Systems & Systems engineering, System Software Interface, and the MBSE initiative. A system-software interface survey [1] highlighted problem areas: Expertise, Interface definition, Leadership, Process, Environmental & system, and Technology and Tools. While relevant, this is still rather general and does not provide actionable insight in context of specific companies, which is desirable for practitioners. For high-tech equipment companies some specifics are as follows. The *innovative nature* of the industry implies a search for novel solutions, often limited by physical system capabilities, e.g., speed of physical components. They may employ different project management approaches like "Rolling waves" used for SE and Agile software methods (e.g. SAFe) for SWE. These need to be synchronized, particularly for *Brown-field* engineering, to deal with legacy, integration and configure to order, and emerging qualities [2].

A *Platform-based* approach allows re-using assets, but poses challenges like "teams may achieve high commonality but fail to differentiate the products; teams may differentiate the products, but create products with excessive costs" [3]. Typically, the interplay of SWE and SE is behind this.

The trend of *increasing (and speed-up of) software-based product features* causes rapid hiring of software developers, who may lack domain knowledge of the company's product, or be unfamiliar with the company's traditional systems engineering approaches.

Overall, high-tech companies push limits of non-software disciplines and require synchronization on de-risking points, build on their legacy, and need to absorb the rapid growth of SW developers. To ensure continuous value delivery, the company may prefer evolutionary approaches over complete restructuring of their way of working. In this case, bottom-up approaches to identify improvement points (e.g., using grounded theory to construct concepts from interviews) can win over applying top-down frameworks (such as [4]).

III. Approach

Figure 1 illustrates our proposed (and ongoing) approach: as a first step, experienced in-company specialists are interviewed on the topic of *SW-SWE interplay* by an expert team familiar with modern methods and, preferably, in-company practices. After that, each individual interview is processed to construct a map of essential aspects the interviewee mentioned. The individual maps are then merged, and cross-related to concepts from the literature to construct an overall, potentially hierarchical, concept map. In this process, we identify how these concepts are influenced and influence each other, and how

The research is carried out as part of the Papillon program under the responsibility of TNO-ESI. The research activities are supported by the Netherlands Ministry of Economic Affairs and Climate, and TKI-HTSM.

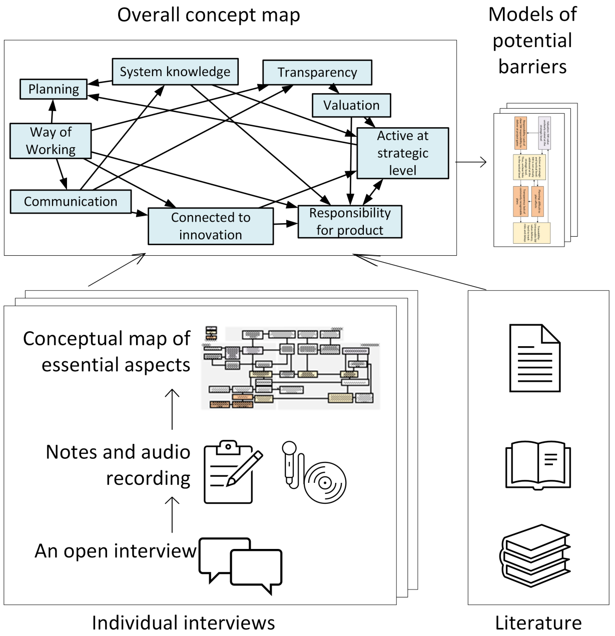

Fig. 1. A method overview.

they contribute to the important Key Performance Indicators (KPI's) of the company. This is the basis to identify potential risks and barriers of the SWE-SE interplay to be presented to decision makers.

IV. Case study

We adopted the outlined approach to interview (1) a SWE specialist and (2) SE specialists from three recently finished projects in a high-tech company. Additionally, we interviewed a program manager and a process improvement specialist. Each of the individual interviews was conducted by a team of three specialists: one SE and one SWE researcher, complemented with either a competence development or management specialist. This ensured coverage of the topic from diverse perspectives to counter the risk of "silo" interpretations.

We then constructed an overall concept map aimed at understanding the SWE position. The categories are: *Way of working* (e.g. SAFe), *Responsibility for product and innovation* (referring to involvement in the product), *Active at strategic and tactical level* (referring to involvement in company directions), *Transparency* (expectations and assumptions), *System and domain knowledge* (extent of understanding the system to build), *Communication* (all aspects), *Valuation* (perceived value in terms of contribution to the overall product), *Planning* (of activities, of activating teams).

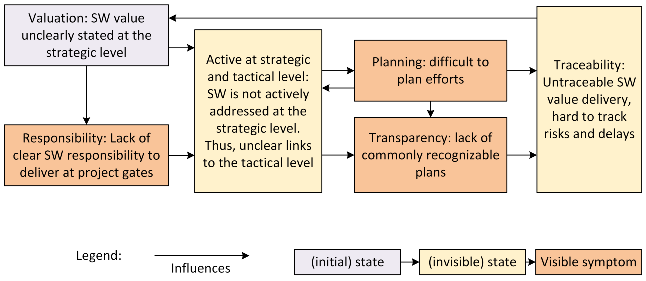
Valuation: SW value unclearly stated at the strategic level
Active at strategic and tactical level: SW is not actively addressed at the strategic level. Thus, unclear links to the tactical level
Planning: difficult to plan efforts
Traceability: Untraceable SW value delivery, hard to track risks and delays
Responsibility: Lack of clear SW responsibility to deliver at project gates
Transparency: lack of commonly recognizable plans
Legend:
Influences
(initial) state
(invisible) state
Visible symptom

Fig. 2. A potential barrier (example).

Afterwards, we constructed risk models. As an illustration, Figure 2 shows a cyclic dependency on SWE topics that do not receive needed attention. Risks identified in a bottom-up way bear weight as they are inputs from professionals who experience them directly. These risks provide recognizable barriers to managers and process developers, who can introduce measures (e.g., reviews, process gates, or responsibility structure) to mitigate them.

More complex dependencies can sometimes be projected to system archetypes described in [6]. For instance, "short-term business concerns vs. long-term new product features" can be seen through the lens of the archetype "Fixes That Fail". Also, difficulties a software group experiences in platform development can be seen through the archetype "Tragedy Of The Commons". Currently, we are studying this.

V. Discussion

We outlined an approach to identify barriers to the fluent interplay between software and other engineering disciplines. Inspired by the grounded theory, we identified relevant concepts in a bottom-up manner from individual interviews. We complemented them, by considering known frameworks and specifics of the high-tech equipment domain.

Incorporating quantifiable elements into the concept map is a promising future direction to spot improvement opportunities in practice. For instance, SAFe 6.0 (released in March 2023) [5] introduced three measurement categories (Competency, Flow, and Outcomes) that can be integrated into the overall concept map.

We invite parties, interested in model-based development and software-system interplay, to discuss and adopt this and similar approaches. In our view, exploring the interplay between software and systems disciplines is crucial for removing barriers on the spot and providing momentum for adopting model-based system and software practices.

References

[1] S. Muscarella, M. Osaisai, and S. Sheard (2020). Systems and Software Interface Survey. In INCOSE International Symposium (Vol. 30, Issue 1, pp. 1305–1323). Wiley.

[2] TNO-ESI, "MBSE in the High-Tech Equipment Industry", TNO Report 2022 R11504, online: http://resolver.tudelft.nl/uuid:aad3da36-9766-4e64-9214-bdf2e014dc2f, accessed: Jun 2023.

[3] D. Robertson and K. Ulrich. "Platform product development." Sloan management review 39.4 (1998): 19-31.

[4] R.T. Pascale and A.G. Athos (1981). The art of Japanese management. Business Horizons, 24(6), 83-85.

[5] What's new in SAFe 6.0? (2023), https://scaledagileframework.com/whats-new-in-safe-6-0/, accessed: Jun 2023.

[6] M. Branz, A. Farrell, M. Hu, W. Liem, and E. Ballard, E. (2021). System Archetypes. Methods Brief Series 1.07: Systems Thinking Foundations. Social System Design Lab: St. Louis, MO.