Software-Intensive Product Engineering in Start-Ups

A Taxonomy

Eriks Klotins, Michael Unterkalmsteiner, and Tony Gorschek, Blekinge Institute of Technology

// *The Start-Up Context Map is a taxonomy of engineering practices, environment factors, and goals influencing the engineering process. This map aims to support further research on software engineering in start-ups and serve as an engineering decision support tool for them.* //

SOFTWARE START-UPS ARE small companies focusing on engineering and launching innovative products fast. Such companies are gaining momentum as an important part of the economy and are central for innovation.[1] However, their failure rate is high; sources suggest that 75 to 95 percent of start-ups fail before realizing their potential.[2,3] In 2015 alone, more than $429 billion was invested in start-ups; given an optimistic 75 percent failure rate, that amounts to $322 billion wasted on building unsuccessful products.[4,5] The high failure rates could be due to unfortunate market conditions, flawed business models, lack of commitment, or bad product ideas. However, the impact of inadequate software engineering practices on the failure rate is largely unknown and could be a significant contributor, next to marketing and business issues.[6,7]

Little is known about the engineering in start-ups. On the surface, start-ups are characterized by uncertainty, lack of resources, and high risk,[8] with product engineering and marketing being the top challenges.[6] However, this characterization isn't granular enough to identify specific challenges or to support adaptation of engineering practices from other, similar contexts and subsequently validate start-ups' specific engineering practices.

Some engineering challenges in start-ups can potentially be solved by adapting scaled-down practices from established companies.[7] However, a lack of understanding about the engineering context makes adaptation and validation of practices in start-ups difficult.[9] For example, some scaled-down agile practices try to address team collaboration issues, which might be negligible in a three-person start-up. So, ad hoc adaptation of engineering practices depends on the lead engineers' experience but can produce waste and often delivers unanticipated results.[10,11]

Understanding start-ups' engineering context will enable adaptation and development of engineering practices specific to start-ups. You can describe a context by, for example, listing and breaking down the factors influencing the engineering process.[12] However, simply listing the context factors isn't enough. The context description must be presented in a way that not only is useful for decision support in practice but also enables further research on engineering in start-ups.

Here, we present the *Start-Up Context Map*, a breakdown and description of factors influencing software-intensive product engineering in start-ups. (For more on software-intensive product engineering, see the sidebar.) By providing detailed descriptions of the factors influencing engineering processes, this map is also a decision support tool and a repository of state-of-the-art and state-of-the-practice knowledge. Along with the map, we present four use scenarios tailored for both practitioners and researchers. As far as we know, the Start-Up Context Map is the first attempt to systematize knowledge pertaining to software-intensive product development in start-ups.

Engineers and entrepreneurs can use the Start-Up Context Map to not only support engineering decision making but also learn from others and share their experiences with peers. Equally important, researchers can use the map to facilitate transfer of their research results to practice and get practitioners' input. In essence, the map can be seen as a Software Engineering Body of Knowledge for start-ups that's open, evolving as new experience is gained, and managed by the start-up community. Although the map is still in a draft state, there are no alternatives to support researchers and practitioners in decision making and devising better engineering practices for start-ups.[13]

SOFTWARE-INTENSIVE PRODUCT ENGINEERING

By *software-intensive product engineering*, we mean applying well-understood practices in an organized way to evolve a product containing a nontrivial software component from idea to market, within cost, time, and other constraints.

The engineering process is influenced by a set of circumstances—the context. Understanding the context is crucial to understanding the engineering process.[23]

The Map's Evolution

As a research framework, we used the industry–academia collaboration model that Tony Gorschek and his colleagues proposed (see Figure 1).[14] The model illustrates the Start-Up Context Map's evolution from inception to the envisioned future. White bubbles represent completed steps; gray bubbles show the map's envisioned evolution.

Start-ups' high failure rate poses an economic problem for the software industry (see step 1 in Figure 1). On the basis of literature surveys and the study of start-up experience reports, we formulated the working hypothesis that engineering inadequacies could be contributing to these high failure rates (step 2). Although start-ups adapt established companies' practices, such practices aren't necessarily suited for start-ups because they try to address challenges not present in start-ups and disregard start-up-specific challenges.[6,7] This finding highlights the need to identify and develop, evaluate, and validate more start-up-specific software engineering practices.

The Start-Up Context Map (step 3) aims to characterize start-ups' engineering context, thus supporting decision making in start-ups and providing a foundation for developing start-up-specific engineering practices. We used other context models as inspiration for parts of the map. We discuss the map's construction and use in detail later.

The map is a work in progress. So, we created an online tool (startupcontextmap.org) that supports collaboration and continuous development of the map. We describe the tool's functionality and usage scenarios later.

We linked 88 start-up experience reports to specific map nodes (step 4). This demonstrated the map's usefulness for systematizing unstructured information and validated its content and coverage. Statements from the experience reports complement the map with first-hand practitioner experience with specific contextual factors and engineering practices.

To complete step 4, we'll conduct workshops in the start-up research community (for example, the Software Startups Global Research Network; softwarestartups.org). We'll use insights from the workshops to update the map and prepare it for static and dynamic industry validation, shown as steps 5 and 6 in Figure 1. We expect to receive specific, actionable input from the workshops; at this point, it's difficult to provide a design of subsequent validation steps. However, our overall strategy is to seed the Start-Up Context Map such that it provides immediate

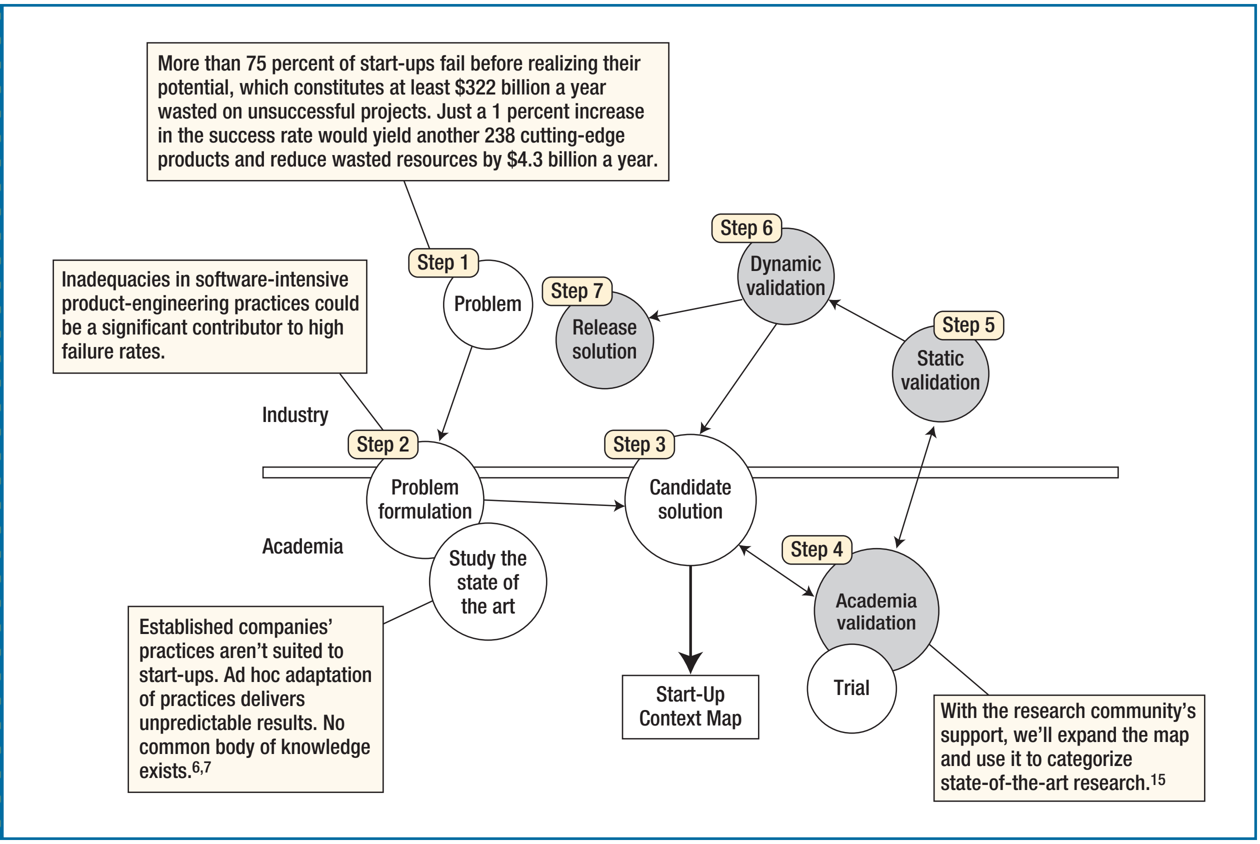

FIGURE 1. The Start-Up Context Map's evolution.[14] White bubbles represent completed steps; gray bubbles show the map's envisioned evolution.

and growing value for practitioners, stimulating the sharing of data that researchers can synthesize.

The Start-Up Context Map

We conducted a literature review to find existing categorizations of goals, practices, and environmental factors from other contexts (see the sidebar "Snowball Sampling"). From the 1,206 discovered papers, we identified nine studies proposing a categorization of engineering context. However, none of the existing models were broad enough to cover the context map's envisioned scope—start-up goals, environment, and practices. The existing models overlapped and were even sometimes incompatible with each other.

Lacking a solid starting point, we created the map structure by brainstorming, plugging existing models into our structure, and searching for additional studies supporting each map node. During construction, we continuously cross-validated the map with the existing models to ensure that all those models' relevant context factors were placed in the map.

The Map's Structure

Figure 2 shows an overview of the context map, which comprises three main context factors: the *goals*, *environment*, and *practices*. These three factors affect each other and must be considered together. Goals drive all decisions in a start-up, decisions are made in a specific environment that influences them, and the decisions pertain to using specific practices in a specific environment.

We broke down each main context factor into subfactors and leaf nodes, creating a tree-like structure. Currently, the map holds 26 subfactors pertaining to goals, 77 subfactors pertaining to the internal and external environments, and 138 practices. In Figure 2, each segment's width denotes how many subfactors it contains.

Each leaf node contains a granular description of the subfactor, examples from practice, references to related work, and practitioners' experience (see Figure 3). Details supported by literature are referenced; details still requiring validation aren't. In this way we ensure a clear connection between the map and supporting literature. Moreover, by denoting less certain details of the map, we invite collaborators to investigate them.

Validating the Map

We performed several activities to validate the map's completeness and usefulness. Completeness refers to what extent the map covers all relevant aspects of software engineering in start-ups. Usefulness refers to what extent the map is suited for categorizing information regarding software engineering in start-ups.

Completeness. We invited members of the Software Startups Global Research Network to collaborate on validating the map. We conducted a joint workshop to review the map and reach consensus regarding whether it covered all relevant aspects of software engineering in start-ups. The 10 workshop participants first reviewed the map offline and provided suggestions for improvement. We compiled their feedback and discussed unclear items in an online meeting.

From the feedback, we adjusted the map's scope and updated its contents. For example, we added subfactors related to technical debt and improved the descriptions of the subfactors related to software testing and construction.

We used the improved map to devise a case survey for collecting data on engineering practices in start-ups (we discuss the use scenarios in more detail later).

This analysis revealed two things. First, the map sufficiently describes software engineering in start-ups to a level that enables assessment of the engineering context and utilized practices. Second, there's substantial interest in the research community to collaborate on the map.

Usefulness. We performed a static validation and applied the map to 88 start-up experience reports, each written by one of the principals of a start-up. Although such reports contain valuable experience lessons, their unstructured nature makes them difficult to benefit from.

We applied our map to categorize statements in the experience reports. First, we segmented the reports by paragraph. Then, we analyzed each report paragraph by paragraph to identify statements pertaining to specific engineering factors. We mapped higher-level statements to higher-level nodes and mapped more-specific factors to leaf nodes. We mapped paragraphs addressing multiple factors or ambiguous statements to multiple nodes.

In case we identified a statement that couldn't be clearly mapped to a factor, we considered expanding the map with a new factor.

As a result of the mapping, we linked 876 statements from the reports to 69 context factors. During the mapping, we identified and added context factors. For example, we now differentiate between a product's internal view and external

SNOWBALL SAMPLING

Snowball sampling is a reference-based method to systematically discover new papers referenced by a starting set.[24] We applied the method in the following four steps.

DEFINING A START SET

We started with a paper by Kai Petersen and Claes Wohlin[9] and looked for other similar papers aiming to categorize context. We consulted colleagues and looked into different research areas for papers, aiming to characterize context. The final set consisted of five papers from the start-up,[16] agile-development,[25] software process improvement in small organizations,[26] and industrial-context[9] research areas.

DISCOVERING PAPERS

We performed multiple iterations of forward snowball sampling until we discovered no new relevant papers.

SCREENING

We screened the discovered papers to remove duplicates, sources that weren't peer reviewed, and papers that didn't list engineering context factors.

SANITY CHECK

We performed a backward snowball-sampling iteration to ensure we hadn't missed any relevant paper.

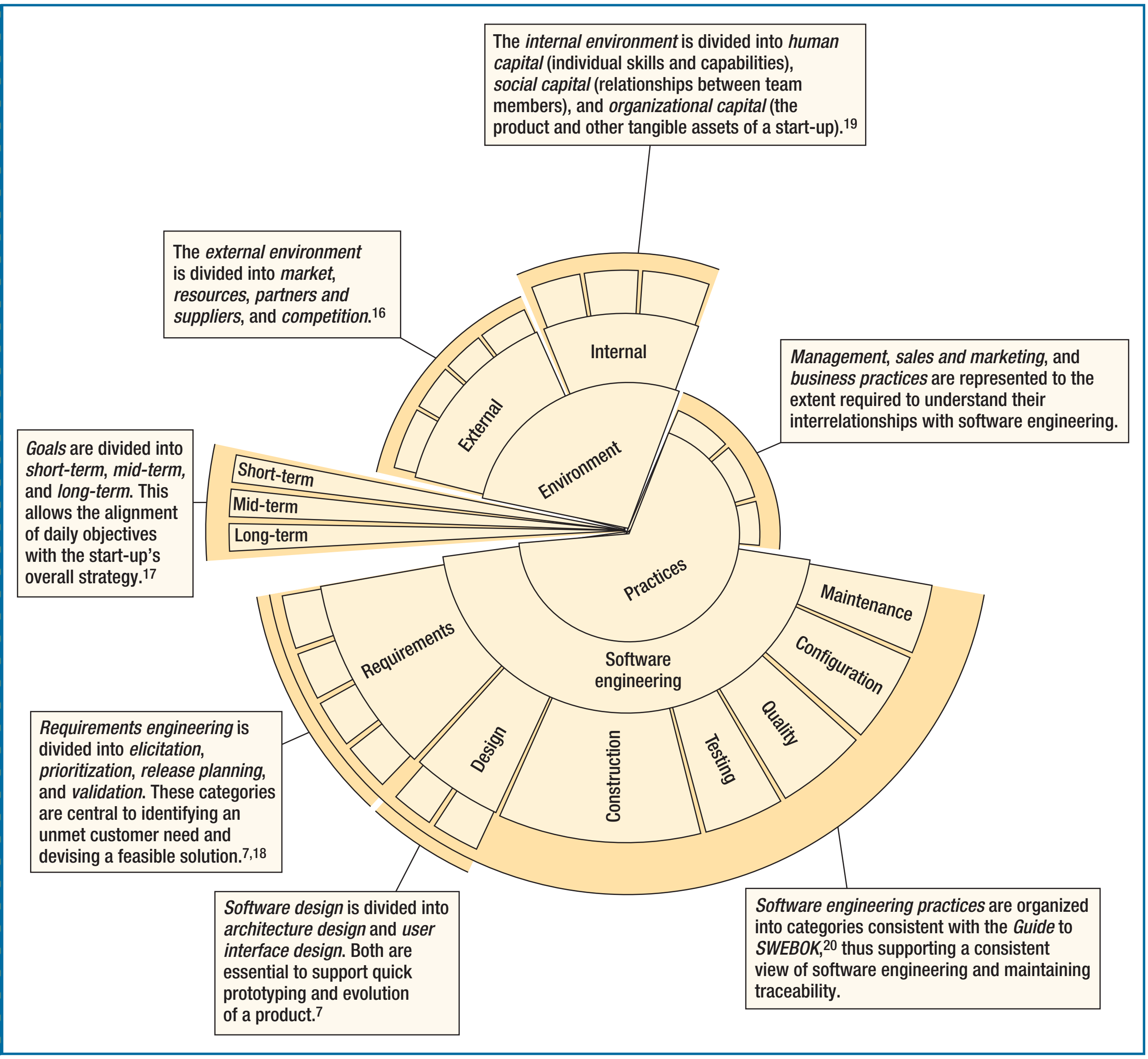

FIGURE 2. An overview of the Start-Up Context Map.

view. The internal view refers to the product's engineering view: architectural design, effort, and complexity. The external view represents how a customer sees a product—for example, features, usefulness, and suitability for a task.

This analysis revealed three things. First, the map is useful for systematizing experience reports. This systematization enables topical access to statements in the reports, thus supporting further analysis. The descriptions of practitioners' experience with a factor expand the overall description, making the map more useful to practitioners.

Second, practitioner experience reports are a useful source to identify new factors for the map, eventually also allowing pruning of the map if parts are never used.

Finally, using the map as an experience database is a feasible use scenario (we discuss this more in the next section).

Using the Map

The use scenarios in Figure 4 aim to provide practical support to start-up practitioners and to connect start-ups with researchers, experts, and other stakeholders.

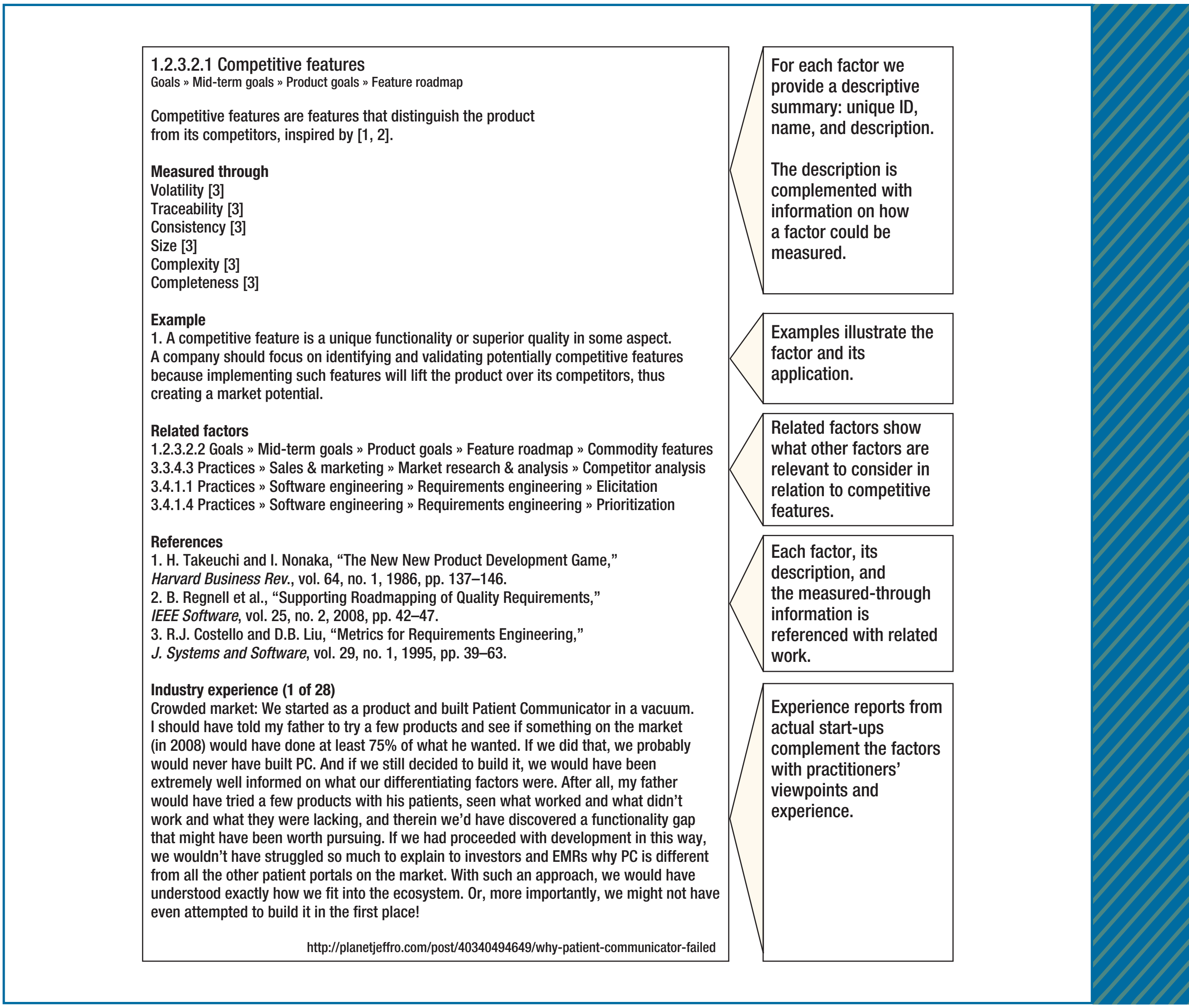

FIGURE 3. Leaf node granularity.

The map as a glossary. Use scenario 1 involves the map as a glossary. Typically, start-ups must identify relevant focus areas and avoid wasting resources on irrelevant activities. Moreover, a less experienced founder might not realize that a critical activity is neglected. One start-up founder reflected on the lack of understanding regarding a start-up's scope:

> *Looking back, when I started 99dresses fresh out of high school I was very naive and had zero idea what I was doing. In fact, I didn't even know what a start-up was! I just knew I wanted to solve a problem I personally experienced: having a closet full of clothes but still nothing to wear.*[21]

Even more-experienced founders could benefit from the map; for example, they might be biased toward their past experiences and overlook some context factors. A glossary

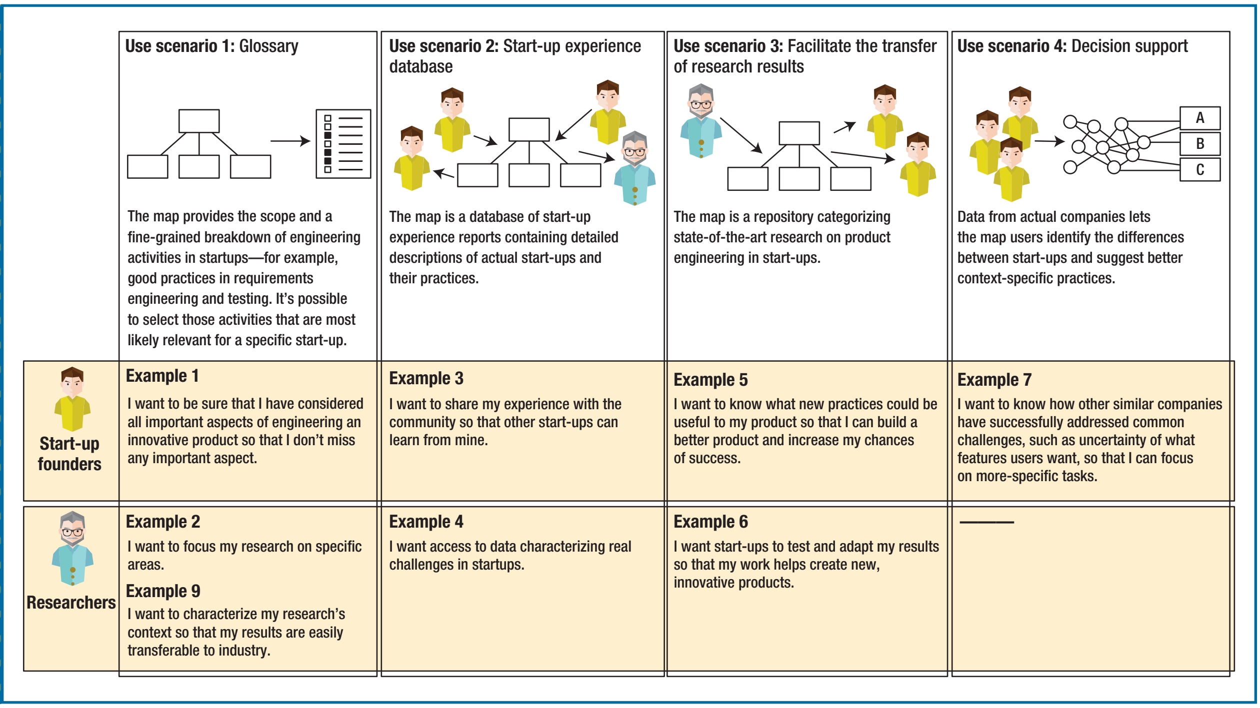

FIGURE 4. Four use scenarios for the Start-Up Context Map.

provides a lightweight decision support tool that both novice and seasoned start-up practitioners can use.

The Start-Up Context Map provides a basis for creating a comprehensive glossary of good engineering practices. An overview of the practices lets practitioners create their own methods. Use scenario 4 further elaborates identification of specific engineering methods. The glossary is community driven; entrepreneurs, start-up engineers, and start-up researchers decide what practices to include in the map. By having an overview of what goals, environmental factors, and practices could be relevant, a founder can make an educated decision to focus on or ignore certain context factors.

Researchers can use the map to identify gaps and specific areas for investigation—for example, by looking into the map's less developed areas. Equally important, researchers can use the map to characterize their research's context, thus enabling safe adaptation of research results. Through an understanding of the context regarding the application of a practice, successful transfer and replication of that practice can be enabled.

The map as an experience database. Use scenario 2 refers to using the map as an experience database. Although both practitioners and researchers value experience reports, these reports often don't contain enough structure and detail to be useful, and they're seldom collected in one place. Currently, the map indirectly holds the combined experience of 88 start-ups and is connected to research in the area, as we mentioned before.

Start-up founders are often keen to disseminate their experience. As one start-up founder stated,

> *One thing I'll be doing more of is writing about my experience. Partially because it's therapeutic, but also because if there's a silver lining in all of this (and there is), it's that I can help educate others about a path fraught with hardship, but rewarding nonetheless.*[22]

Our map provides a backbone for structured experience reports. Start-up engineers can add their experience with specific factors directly to the map (see Figure 3), enabling structured access to their report. In this way, other practitioners can easily use an experience report because they can easily locate its relevant parts. Researchers can use the same structure to gather all experiences pertaining to a specific context factor and use that information as input for their research.

We used the context map to devise the case survey we mentioned earlier, which explores how start-ups employ engineering practices and what the start-up engineering context's characteristics are. We'll use the survey results to update the map with best practices.

The transfer of research results. Use scenario 3 supports the transfer of research results from academia to start-ups. State-of-the-art practices from academia are created through a rigorous research process, have passed certain validation and technology transfer stages (see Figure 1), and are accompanied by sufficient description that enables educated decisions regarding whether a practice is applicable in a specific start-up case.

Researchers can use the map and add their state-of-the-art results under specific nodes. Consequently, start-up engineers can easily locate cutting-edge practices to address a specific issue they encounter. Combined with use scenario 2, this enables collaboration between start-ups and researchers.

The map as a decision support tool. Use scenario 4 describes using the map for decision support. The characterization of the start-up context enables identification of similarities between start-ups and between their goals, environments, and applied practices. The decision support system uses start-up environment characteristics and goals as input and suggests engineering areas or environmental factors to focus on for a particular start-up. An inference engine and a knowledge database for decision support are based on start-up experience reports from use scenario 2. However, creating such a decision support tool requires many experience reports.

Thus, proven combinations of practices to tackle common challenges can be bundled for reuse in other similar start-ups. Such bundles or patterns can be created by start-up experts and used by practitioners. The goal is to evolve patterns over time. Examples of such patterns include the Lean Startup, Extreme Programming, Scrum, and various models that support innovation in small organizations.

Discussion. These use scenarios aim to reduce uncertainty by providing engineering decision support and facilitating the development of new start-up context-specific engineering practices. The Start-Up Context Map, by enabling informed decisions, can therefore increase start-ups' likelihood of success in engineering innovative products.

The Start-Up Context Map can be seen as a start-up body of knowledge, similar to the *SWEBOK Guide* (*Guide to the Software Engineering Body of Knowledge*) and *PMBOK Guide* (*A Guide to the Project Management Body of Knowledge*), but managed and developed by the start-up community in collaboration with interested researchers.

Our intention is to provide a core structure and the necessary tools for the map to be further validated, adapted, and evolved by the community. To support this, we launched the continuously evolving online tool at startupcontextmap.org.

We leave you with these key points:

- A structured approach to collecting start-up experience reports enables synthesis of best engineering practices for start-ups.
- Knowledge of which practices work and which don't under specific conditions is essential to tailor an engineering process to suit individual start-ups' needs.
- The Start-Up Context Map is a community-driven project to collect and categorize good start-up-specific engineering practices.

References

1. S. Srinivasan et al., "Venture Capital: Fueling the Innovation Economy," *Computer*, vol. 47, no. 8, 2014, pp. 40–47.
2. S. Blank, "Why the Lean Start Up Changes Everything," *Harvard Business Rev.*, vol. 91, no. 5, 2013, p. 64.
3. D. Gage, "The Venture Capital Secret? 3 Out of 4 Start-Ups Fail," *Wall Street J.*, 20 Sept. 2012.
4. *2H 2015 European Middle Market Report*, PitchBook, 10 Nov. 2015; pitchbook.com/news/reports/2h-2015-european-middle-market-report.
5. *U.S. Middle Market Report Q4 2015*, PitchBook Data, 2015.
6. C. Giardino, S.S. Bajwa, and X. Wang, "Key Challenges in Early-Stage Software Startups," *Agile Processes in Software Engineering and Extreme Programming*, 2015, pp. 52–63.
7. E. Klotins, M. Unterkalmsteiner, and T. Gorschek, "Software Engineering in Start-Up Companies: An Exploratory Study of 88 Start-Ups," submitted to EMSE, 2016, pp. 1–19.
8. C. Giardino et al., "What Do We Know about Software Development in Startups?," *IEEE Software*, vol. 31, no. 5, 2014, pp. 28–32.
9. K. Petersen and C. Wohlin, "Context in Industrial Software Engineering Research," *Proc. 3rd Int'l Symp. Empirical Software Eng. and Measurement* (ESEM 09), 2009, pp. 401–404.
10. A. Yau and C. Murphy, *Is a Rigorous Agile Methodology the Best Development Strategy for Small Scale Tech Startups?*, tech. report, Dept. of

ABOUT THE AUTHORS

ERIKS KLOTINS is a software engineering PhD student at the Blekinge Institute of Technology. His thesis focuses on software engineering practices in start-ups. Klotins received a master's in software engineering from the Blekinge Institute of Technology. Contact him at eriks.klotins@bth.se.

MICHAEL UNTERKALMSTEINER is a senior lecturer in the Blekinge Institute of Technology's Department of Software Engineering. His research interests include software repository mining, software measurement and testing, process improvement, and requirements engineering. Unterkalmsteiner received a PhD in software engineering from the Blekinge Institute of Technology. He's a member of IEEE. Contact him at michael.unterkalmsteiner@bth.se; www.lmsteiner.com.

TONY GORSCHEK is a professor of software engineering at the Blekinge Institute of Technology. His research interests include requirements engineering, technology and product management, process assessment and improvement, quality assurance, and practical innovation. Gorschek received a PhD in software engineering from the Blekinge Institute of Technology. He's a member of IEEE and ACM. Contact him at tony.gorschek@bth.se.

Computer and Information Science, Univ. of Pennsylvania, 2013.

11. C. Giardino et al., "Software Development in Startup Companies: The Greenfield Startup Model," *IEEE Trans. Software Eng.*, vol. 42, no. 6, 2016, pp. 585–604.
12. D. Kirk and S.G. MacDonell, "Investigating a Conceptual Construct for Software Context," *Proc. 18th Int'l Conf. Evaluation and Assessment in Software Eng.* (EASE 14), 2014, article 27.
13. N. Paternoster et al., "Software Development in Startup Companies: A Systematic Mapping Study," *Information and Software Technology*, vol. 56, no. 10, 2014, pp. 1200–1218.
14. T. Gorschek et al., "A Model for Technology Transfer in Practice," *IEEE Software*, vol. 23, no. 6, 2006, pp. 88–95.
15. M. Unterkalmsteiner et al., "Software Startups—a Research Agenda," *e-Informatica*, vol. 10, no. 1, 2016, pp. 89–123; www.e-informatyka.pl/attach/e-Informatica...10/eInformatica2016Art5.pdf.
16. S. Chorev and A.R. Anderson, "Success in Israeli High-Tech Start-Ups; Critical Factors and Process," *Technovation*, vol. 26, no. 2, 2006, pp. 162–174.
17. L. Lehtola et al., "Linking the Business View to Requirements Engineering: Long-Term Product Planning by Roadmapping," *Proc. 13th IEEE Int'l Conf. Requirements Eng.* (RE 05), 2005, pp. 439–443.
18. J. Melegati et al., "Requirements Engineering in Software Startups: A Grounded Theory Approach," *Proc. 2nd Int'l Workshop Software Startups*, 2016.
19. C. Wohlin et al., "A General Theory of Software Engineering: Balancing Human, Social and Organizational Capitals," *J. Systems and Software*, Nov. 2015, pp. 229–242.
20. *The Guide to the Software Engineering Body of Knowledge*, ver. 3.0, IEEE CS, 2014.
21. N. Durkin, "My Startup Failed, and This Is What It Feels Like ...," *Female Founders*, blog, 23 June 2014; medium.com/female-founders/my-startup-failed-and-this-is-what-it-feels-like-c5d64b3ae96b.
22. C. Poole, "Today My Startup Failed," *Chris Hates Writing*, blog; chrishateswriting.com/post/74083032842/today-my-startup-failed.
23. D. Kirk and S.G. MacDonell, "Categorising Software Contexts," *Proc. 20th Am. Conf. Information Systems* (AMCIS 14), 2014.
24. C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering," *Proc. 18th Int'l Conf. Evaluation and Assessment in Software Eng.* (EASE 14), 2014, article 38.
25. T. Chow and D.-B. Cao, "A Survey Study of Critical Success Factors in Agile Software Projects," *J. Systems and Software*, vol. 81, no. 6, 2008, pp. 961–971.
26. T. Dybaa, "Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context," *Proc. 9th European Software Eng. Conf. Held Jointly with 11th ACM SIGSOFT Int'l Symp. Foundations Software Eng.*, 2003, pp. 148–157.