From Idea to Impact: Survival Guide for Successful Products

Christof Ebert, Vector Consulting

Silvia Abrahão, Universitat Politècnica de València

V. S. Mani, Siemens Healthineers

Digital Object Identifier 10.1109/MS.2024.3363540
Date of current version: 11 April 2024

SOFTWARE IS THE driving force for innovations. However, many businesses miss out on opportunities to turn ideas into products and services with impact. An idea needs to create impact, or it is a waste. However, we often face two disconnected streams, with academic research being focused on the idea, while industry focuses on impact. Most academic organizations are not aware of industry and market needs. Churning out papers seems more relevant than getting an idea to a viable product. On the other hand, companies struggle to transfer research, and rather duplicate it in their own labs. Ideal product innovation is collaborative, continuous, and market oriented. In a highly competitive world, going from idea to impact is more than an iterative process from a need to a solution. It is an ongoing fight for survival. The articles in this theme issue provide many lessons learned of how to survive.

Innovation: From Idea to Impact

Imagine two campers sitting in front of their tent when suddenly a bear attacks. One camper just starts running barefoot. The other gets into his fancy sneakers and shouts to the first, she would be faster with shoes. The former shouts back that though there is no way to run faster than the bear, it is sufficient to be faster than her comrade. The rest is bad luck or a nice meal, depending on perspective. The same holds for software innovation. Many ideas do not get impact because too much time is wasted from research to results.

Software-driven innovation is the call of the day. Yet, many attempts to create innovative products fail. Both industry and academia face challenges in transferring technology effectively. Organizations, both academic and industrial, remain in their silos with familiar networks and practices, but eventually lose traction in enabling adoption. There is no simple cookbook recipe for innovation; however, the sharing of experiences reveals invaluable insights[1,2]:

- Create new needs.
- Build on an existing platform.
- Use an agile team.
- Proof value.
- Grow incrementally.

Many ideas do not get impact because too much time is wasted from research to results.

To illustrate these principles from idea to impact, let us look to a software idea with big impact. In the early seventies, Ray Tomlinson enhanced a simple messaging service toward the first electronic mail. Working on ARPANET (Advanced Research Projects Agency Network), a network connecting computers and a predecessor of today's Internet, he discovered that users communicated by messaging to other users on the same computer. He thought beyond this simplistic pattern and anticipated sending asynchronous messages to any computer: worldwide. Tomlinson used the "@" to indicate a destination in the format <username>@<name of computer>, which is essentially how e-mail has been addressed ever since.[3] When inventing the e-mail and innovating collaboration software, Tomlinson followed the above five success factors. He stimulated new needs to allow people asynchronous communication. He used an existing platform rather than waiting for something fancy and perfect. He worked with a small team and incrementally improved the service. He proved value by bringing the results to his community, where he, for instance, coached a woman who used e-mail to create one of the first newsletters, and eventually became his wife.

Our ambition with this theme issue is to share such experiences from global leaders and thus show how to learn and translate these experiences to your own environments.

Bringing Ideas to Practice and Products

Start with the end in mind to continuously iterate between novel ideas and potential impact. The idea-to-impact funnel is fueled by ideas that must be tested and improved. This means collaborative work, creative and iterative, rather than the traditional research silos and sequential pipelines.

While transferring research know-how to create a successful software product or service, organizations must understand the market need, technology, the value proposition, and the limitations of the technology. Developing software products with enduring value needs close collaboration along the idea-to-impact feedback loop. Figure 1 shows this loop with three phases and how it stimulates the creation of innovative products.

The first phase develops the idea and concept. Research will yield

knowledge about the feasibility of the idea. This phase includes analyzing markets, and benchmarking to identify industry trends, market needs, potential partners, and competitors.[1] It also involves packaging and transferring initial prototypes using technology transfer mechanisms, such as collaboration. By transferring knowledge and technologies, organizations can leverage existing expertise and resources to develop new products, services, and processes. Feedback is relevant and must be bidirectional, i.e., from research to product and from markets to research.

The second phase evaluates and develops the actual innovation. The novel concept is assessed to determine if the idea is suitable for markets and if the intellectual property needs to be protected. The most relevant thing here is to create and address real needs. Products fail if there is no need. After the intellectual property is appropriately protected, a marketing and commercialization plan for the concept is defined and a proof-of-concept is developed. The concept is tested and hardened to survive real application. Marketing must be planned to achieve a successful go-to-market.[2] Sales and service must be addressed internally or with ecosystems. Software will not be perfect when first delivered, and if initial users do not get support, they will turn away.

The third phase targets the industrialization and market entry of the product. Start-ups, incumbents, and software centers would benefit from open innovation and cocreation initiatives. Joint innovation labs, digital innovation hubs, open innovation platforms, and demonstration platforms are proven facilitators for idea to impact.[4,5] Such initiatives allow research and data sharing, as well as the codesign and cocreation of products, along with their piloting and testing, which help organizations to reduce costs and accelerate development.

Once the product is released, the way forward depends on multiple factors, such as the market demand, the invention's stage of development, and further evolution plans, such as releases and feature activation. For the product to have an impact on industry and society, it may take several iterations of the phases to translate idea to impact. While Figure 1 is not a defined pattern and certainly depends on factors—such as company size, market, and innovation degree—it emphasizes the need to use a wide range of channels to foster collaborative innovation.

This theme issue presents several good examples of the idea-to-impact feedback loop. We present four successful cases of technology transfer and one case of adoption that connect research and practice. Rather than rewriting the abstracts of the articles in this theme issue, we distilled their essence for fast access. Table 1 provides the overview on the five selected articles together with a mapping to the idea-to-impact cycle and the major take-aways. The last

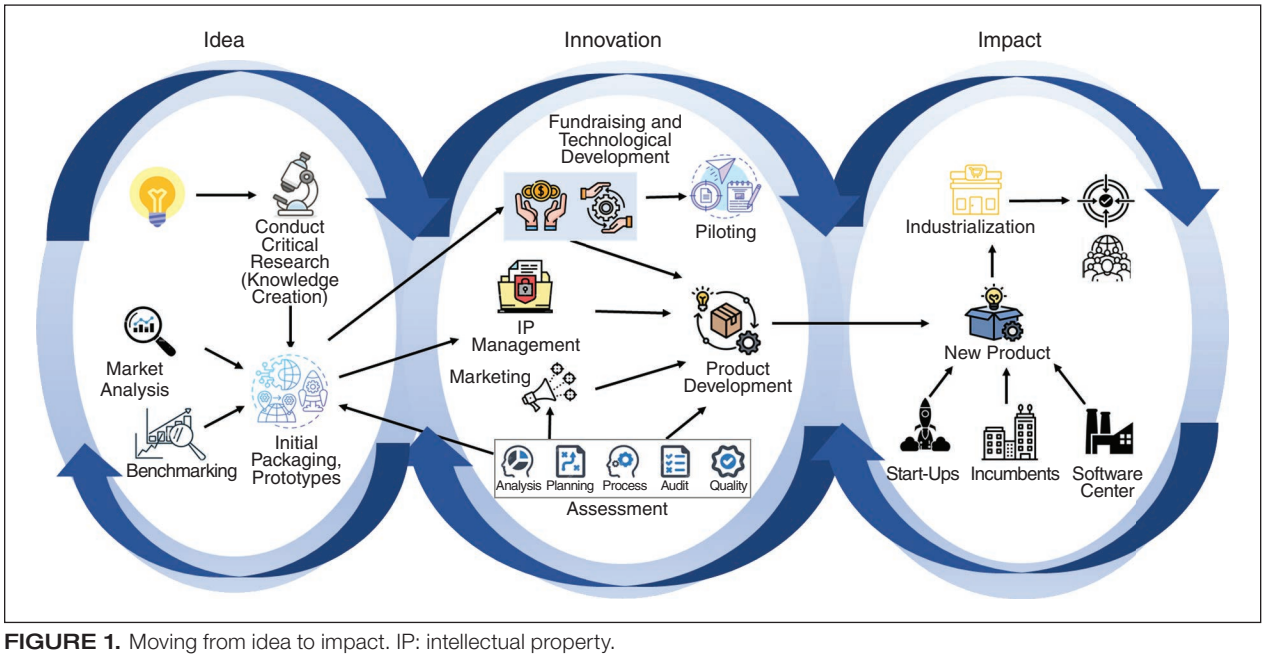

FIGURE 1. Moving from idea to impact. IP: intellectual property.

Table 1. Theme issue content and take-aways.

Article title	Scope	Highlights	Take-aways
The Future Is Already Here [A1]	Industry-academic research	• Asserts that innovations in software originate in industry not in academic research. • Declares the traditional view on the role of academic research in software is outdated. • Presents an alternative approach to academic research using the case study of the software center.	• Identify novel techniques and ways of working in companies. • Assess where the techniques can be used as is. • Validate techniques by piloting. • Spot limitations to resolve for quick adoption. • Align incentives for companies and researchers.
Connecting Research and Practice for Software Product Quality Evaluation and Certification: A Software Laboratory's 25-Year Journey [A2]	Evolve a business model from an idea	• Recounts the difficulties in transferring results of research on software quality to the industry. • Outlines the creation of a spin-off that became an ISO-accredited lab that provides software quality evaluation services. • Shows the lab evaluations have been used to certify software product quality.	• Work closely with the industry from the beginning. • Identify stakeholders for the solution and involve them from the start. • Be prepared to invest significant time and effort to overcome limitations in the research for market acceptance.
Innovating Industry with Research: eKnows and Sysparency [A3]	Full cycle of idea to impact in a company	• Narrates how to spin off a company from a research project. • Describes a software platform that enables building reverse engineering tools and document generators. • Outlines the development, which was guided by domain-specific requirements and a focus on supporting the reuse of components that analyze software in different programming languages.	• Focus on both piloting proofs-of-concept as well as differentiating the solution from competitors. • Identify the most suitable domain for a potential spin-off. • Restructure the code base for separation of the spin-off while enabling collaboration. • Structure the collaboration to enable the research team to deliver new advances and the spinoff in identifying and sharing industry needs. • Transfer best engineering practices along with the code.
Software Size Measurement: Bridging Research and Practice [A4]	Adoption of an idea to address a specific industrial need	• Investigates the limited adoption of functional size measurement methods despite being objective and consistent. • Gathers insights from firms experienced in size measurement to uncover industry expectations.	• Get guidance on how to measure functional size. • Follow the approach on how to use the results. • Relate functional size measurement to estimation and quality predictions.
From Research on Data-Intensive Software to Innovation in Data Spaces: A Search Service for Tabular Data [A5]	Agile start-up approach for full idea-to-impact cycle	• Presents a novel approach combining lean startup methodology and the last research mile recommendations to develop marketable products and services. • Identifies an opportunity for research-based advances in data-intensive software engineering. • Outlines the process, with clear definitions of done and key performance indicators, which was systematically followed.	• Plan beyond proof-of-concept, for proof-of-value and proof-of-use. • Consider the customer's problem, not the solution, as the most important item in the business. • Forge business perspectives by promoting collaboration between faculty members, researchers, and students to nurture academic entrepreneurs.

ABOUT THE AUTHORS

CHRISTOF EBERT is the managing director of Vector Consulting Services, 70499 Stuttgart, Germany. Contact him at christof.ebert@vector.com.

SILVIA ABRAHÃO is a full professor of software engineering at Universitat Politècnica de València, 46022 Valencia, Spain. Contact her at sabrahao@dsic.upv.es.

V. S. MANI is working on operation excellence in software engineering at Siemens Healthineers in Bengaluru, Karnataka 560100, India. Contact him at vs.mani@siemens-healthineers.com.

column of Table 1 is most relevant for industry practitioners, as it indicates what they should learn and apply to their own innovations.

Making a Lasting Impact

This theme issue examines the new and enduring challenges related to technology transfer in software engineering. Technology transfer is a key process that facilitates the transformation of ideas into products within ecosystems. By understanding its importance, learning from examples, following key lessons learned, and analyzing successful case studies, organizations can harness the power of technology transfer to drive innovation, economic growth, and societal progress.

The idea-to-impact process often takes too long, especially when crossing the boundaries between academic research and industry application. Agility and early prototyping with techniques, such as design thinking, will help to accelerate the process. Early and continuous feedback from markets not only ensures that the product targets the right segments with the right features and at the right price. It also ensures speed, which in today's global competition is a key success factor for any product.

Software apps and centralized IT systems, such as content streaming, ecommerce, and search engines, are often taken as role models for fast DevOps-driven innovation. Yet they cannot be generalized to all markets. For instance, safety-critical products, such as robots—automotive, medical, and transport—are more challenging than a simple smartphone. Embedded systems cannot thrive on a fast DevOps rollback. They demand functional safety and very high reliability. In our highly regulated landscape, governance is pivotal. Standards matter as well as legal rules. Products with AI-based adaptive and learning algorithms must specify their behaviors to get approval and homologation. The underlying AI rules must be transparent and, in the case of self-learning, it must be ensured that rules that are still valid are not overwritten.

Each market follows their own rules and poses specific constraints toward idea-to-impact. Yet, they all have one thing in common: innovation needs execution. A good idea that arrives too late will lose against the second-best idea that arrives much earlier. As with the campers and the bear in our introduction, you must be faster than your competition. Continuously.

Acknowledgment

We thank the authors of the initial 26 submissions for this theme issue. We appreciate the feedback loop of the reviewers, who provided excellent guidance for the authors for more impact, and thank Editor in Chief Ipek Ozkaya for her guidance and support. Silvia Abrahão was supported by the Spanish State Research Agency under the UCI-Adapt project (Grant PID2022-140106NB-I00).

Appendix: Related Articles

A1. J. Bosch, "The future is already here," *IEEE Softw.*, vol. 41, no. 3, pp. 26–32, May/Jun. 2024, doi: 10.1109/MS.2024.3358417.

A2. J. Verdugo, J. Oviedo, M. Rodríguez, and M. Piattini, "Connecting research and practice for software product quality evaluation and certification: A software laboratory's 25-year journey," *IEEE Softw.*, vol. 41,

no. 3, pp. 33–40, May/Jun. 2024, doi: 10.1109/MS.2024.3357119.

A3. V. Geist, M. Moser, J. Pichler, and F. Schnitzhofer, "Innovating industry with research: eknows and sysparency," *IEEE Softw.*, vol. 41, no. 3, pp. 41–48, May/Jun. 2024, doi: 10.1109/MS.2024.3359458.

A4. T. Hacaloğlu, H. Ünlü, A. Yıldız, and O. Demirörs, "Software size measurement: Bridging research and practice," *IEEE Softw.*, vol. 41, no. 3, pp. 49–58, May/Jun. 2024, doi: 10.1109/MS.2024.3358079.

A5. A. Berenguer, O. Alcaraz, D. Tomás, and J.-N. Mazón, "From research on data-intensive software to innovation in data spaces: A search service for tabular data," *IEEE Softw.*, vol. 41, no. 3, pp. 59–66, May/Jun. 2024, doi: 10.1109/MS.2024.3359333.

References

1. C. Ebert, "Requirements BEFORE the requirements: Understanding the upstream impacts," in *Proc. 13th IEEE Int. Conf. Requirements Eng. (RE)*, Los Alamitos, CA, USA: IEEE Comput. Soc. Press, Aug. 2005, pp. 117–124, doi: 10.1109/RE.2005.57.
2. C. Ebert, "Software product management," *IEEE Softw.*, vol. 31, no. 3, pp. 21–24, May/Jun. 2014, doi: 10.1109/MS.2014.72.
3. J. Hicks. "Ray Tomlinson, the inventor of email: 'I see email being used, by and large, exactly the way I envisioned.'" The Verge. Accessed: Feb. 16, 2024. [Online]. Available: https://www.theverge.com/2012/5/2/2991486/ray-tomlinson-email-inventor-interview-i-see-email-being-used
4. T. Gorschek and K. Wnuk, "Third generation industrial co-production in software engineering," in *Contemporary Empirical Methods in Software Engineering*, M. Felderer and G. H. Travassos, Eds. Cham, Switzerland: Springer Nature, 2020, pp. 503–525.
5. C. Marijan and S. Sen, "Industry–Academia research collaboration and knowledge co-creation: Patterns and anti-patterns," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 3, pp. 1–52, 2022, doi: 10.1145/3494519.

Digital Object Identifier 10.1109/MS.2024.3376022