

A C-Level Perspective on the Role of Developers in the Creative Process of Software Startups

Fostering Developer Creativity and Participation in Innovation

Kiev Gama^{ID}, Universidade Federal de Pernambuco

Cleidson R. B. de Souza^{ID}, Universidade Federal do Pará

Alexander Nolte^{ID}, Eindhoven University of Technology

George Valença^{ID}, Universidade Federal Rural de Pernambuco



// Creativity is essential for software startups as it drives innovation and solutions to market needs. Through interviews with C-level startup executives, this study highlights how software developers contribute to creative activities from ideation to user validation. //

CREATIVITY IS A cornerstone of success in software startups, driving innovation and enabling companies to develop solutions that meet market needs. Startups thrive on creative problem-solving and the ability to adapt quickly to changing environments. To streamline and enhance creativity, various processes and methods have been developed and are widely adopted (e.g., lean startup, design thinking, design sprint). They are typically market-centered and user-centered, focusing on rapid idea validation and iterative development, which is the essence of agile development.

Behind the many models and processes around creativity, there are key shared concepts like divergent and convergent thinking, and the notion of problem and solution space, which form a general perspective of the creative process.¹ The purpose of divergent thinking is to create choices, while convergent thinking

Digital Object Identifier 10.1109/MS.2025.3541435

Date of publication 14 February 2025; date of current version 11 April 2025.

focuses on eliminating options and making choices.² Exploring the problem space leads to a shared understanding of the problem before an actual development process starts. Moreover, it leads to the exploration of the solution space, thus promoting a creative ideation process to select the most viable path among the many possible alternatives.³

Creativity in software engineering has been predominantly related to requirements engineering (e.g., generate novel requirements), but, in general, practitioners seem often unaware of various techniques—being mostly limited to brainstorming—for enhancing and assessing creative performance, as pointed out in the most comprehensive review we identified on that topic.⁴

The typical deductive-rationalist thinking style of IT engineers leads to an approach of having a given problem and, based on logic, deducing a suitable solution.³ This approach differs from the creative mindset used in design thinking,^{2,5} which aims to explore a problem–solution space more openly and bring a multiperspective comprehension that helps to deal with the ambiguity of problems.

In software startups, activities related to creativity are often left to individuals with a background in product design and user research, many times leaving software developers behind or only involving them late in the process.⁶ The decision of whom and how to involve personnel in such creative processes is typically taken by executives.⁷ This consequently leads us to ask the question: How do C-level executives perceive the role of software developers in the creative process within startups?

While creativity is often examined from a developer's perspective

in literature,⁴ it is less frequently explored from the viewpoint of those who run companies or manage products. C-level professionals are uniquely positioned to understand creativity from a strategic perspective. In this article, we investigate how these executives perceive their teams' role in achieving the desired level of creativity for their products. Specifically, we explore the perspectives of chief executive, technology, and product officers (CEOs, CTOs, and CPOs) in software startups regarding the role of software developers in the creative process. When referring to such processes in this article, we are mainly considering activities around innovative thinking that range from ideation to concept development and user validation. Although creativity could also span more software-intensive activities (e.g., testing, programming, software reuse), C-level executives may be less aware of such operational details.

To answer our research question, we conducted an interview study with key people from startups at varied maturity stages from Europe and the Americas. Our analysis led to findings such as the importance of analyzing competition to creatively find new ideas for startups, as well as the difference on the role of developers in user-driven and tech-driven startups. Based on our findings, we provide recommendations for practitioners to take more advantage of developers in the creative processes of their software startups.

Creativity in Software Startups

Design thinking is a widely adopted method for structuring creativity in many companies. In design thinking, much of the design

activity—especially conceptual design—can be unplanned, intuitive, and ad-hoc.⁵ This is similar to how startups operate when designing a software product. In terms of processes, startups often adopt a flexible approach where software development, design, and market research are intertwined⁸ and where practices are chosen opportunistically rather than based on a specific methodology.⁹ Their processes are thus typically characterized by frequent pivots and failures.¹⁰ They often resort to a combination of lightweight methodologies for flexibility and fast releases, which keeps them agile and responsive and shortens the time taken from idea conception to production.

For startups, the primary challenge, especially in early phases, lies in building a viable and marketable product.¹⁰ Consequently, requirements engineering and software development often run in parallel, maintaining agility. Despite the creative demands inherent in these processes, the role of developers as key contributors to startup innovation and creativity remains underexplored. Prior studies suggest that developers are frequently engaged late in the process,⁶ and they commonly focus on implementing ideas and suggestions that others have developed, which frequently require additional clarification.¹¹ This study investigates the role of developers not only as implementers but as active contributors to the creative processes driving innovation in software startups.

Research Method

We conducted an interview study with CEOs, CTOs, and CPOs of software startups at varied stages (initial, growth, stabilization, and

maturity¹²). Executives are uniquely positioned to understand how creativity is fostered and implemented across the company, since they typically set the overall strategy of a startup, create the culture that can make or break creativity and ultimately decide if a creative idea moves from idea to product.⁷ Moreover, it is reasonable to assume that in software-oriented startups the CEOs, CTOs, and CPOs are often actively involved in software development and have a software development background themselves. Thus, we conducted a study with these C-level executives, focusing on their perspective on creativity, which is more strategic while still being informed by technical viewpoints. We recruited participants through referrals from our professional network. Out of 30 invitations, 17 people

accepted to participate (13 men and four women). We organized 30-minute online interviews with participants from Brazil (12), United States (2), France (1), Estonia (1), and Ireland (1). We purposefully selected invitees that represented a diverse variety of startups related to business value (from small and average ones to a US\$1 billion-valued unicorns), size (3 to 1,000+ employees), experience (from one to 14 years in the market), and focus (user-driven and technology-driven), as illustrated in Figure 1.

During the interviews, we first asked participants about their profile and the profile of their startup (role and personal background, startup size, phase, and product) before asking them to identify a product or idea they recently worked on. We asked them to describe the process

of how the product or idea was developed. In addition, we prompted them to reflect on the process, the alignment of the process with agile methods, and the involvement of developers in the creative process.

Two authors conducted the interviews, which were recorded and transcribed using an AI-automated tool, with manual verification to correct minor errors. In addition, the interviewers wrote analytic memos during and immediately after each interview to capture initial impressions, observations, and emerging themes. After transcribing the interviews, we followed a thematic analysis strategy in that we created codes to label relevant interview excerpts based on our research focus on creativity and the role of developers in the creative process after familiarizing ourselves with the data. We

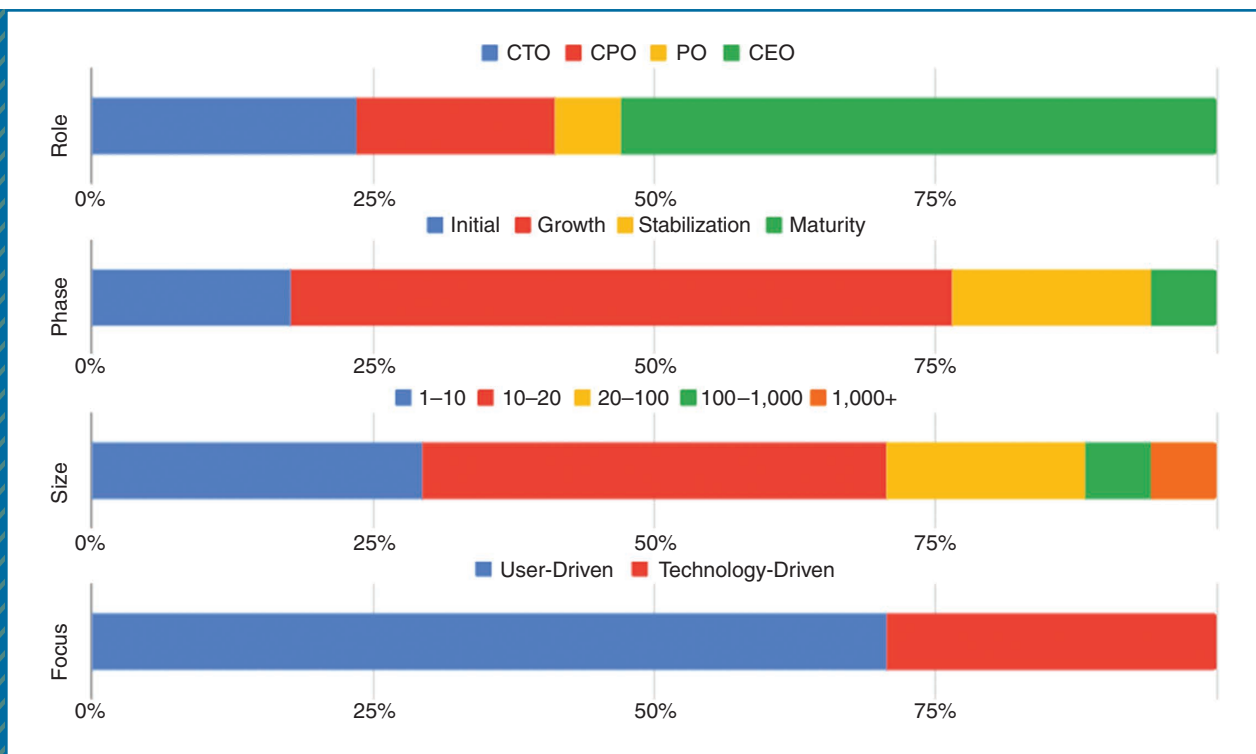


FIGURE 1. Profiles of interviews (role) and their startups in regard to phase, size, and focus (n = 17).

then identified codes under common themes (e.g., sources of inspiration, creative process) related to our research focus. Iterating the procedure, we refined the themes and added new ones as they emerged arriving at a final set of five themes: (1) sources of creative inspiration, (2) creative processes, (3) creativity in user-driven startups, (4) creativity in technology-driven startups, and (5) large language model (LLM) usage. The whole process was conducted collaboratively and iteratively among the authors. After the analysis, we performed member checking by sharing findings and recommendations with 15 of the interviewees, among whom seven provided feedback. Their responses brought reinforcement to our findings and also highlighted the importance of diverse sources of creative inspiration, particularly noting that market experience is essential for exploring the problem space and serves as a complement to competition-inspired creativity. In the following section, we present our updated findings after getting feedback from our interviewees.

Findings

In this section, we discuss the most relevant finding from each of the aforementioned themes and their connection to key aspects of creativity: the exploration of problem–solution space applying both divergent and convergent thinking. Other findings (e.g., strategies for customer involvement, prototyping approaches, A/B testing) were common knowledge and deemed less relevant to report. We further discuss how startups adopt a flexible approach for their creativity, development and business activities, with most decisions being made

opportunistically.^{8,9} Our interviewees did not mention any specific creativity methodology or process being followed. Instead, they elaborated on how they combined different approaches, with recurring mentions of broader methodologies, including lean startup, design thinking, and customer development.

Competition-Inspired Creativity

A notable pattern among the studied startups was the use of competitive testing or benchmarking. Analyzing competitors helps teams understand market gaps and refine the problem space by spotting unaddressed user needs. To gain insights on potential solutions, startups adopt approaches that closely resemble established techniques that encourage divergent/convergent thinking in defining new product features.¹³ Divergent thinking can be explored through strategies to variate existing concepts (e.g., substitute, combine, adapt, modify/magnify, put to other uses, eliminate, reverse/rearrange). These variations help to brainstorm novel ideas, and to focus and refine selected concepts into viable features (convergent thinking). For instance, participant P15, CEO of a user-driven startup, dedicates one day per week to studying competition by searching online or testing products. Highlighting the role of competition in this creative process, P10, the CTO of a tech-driven unicorn startup, shared:

“Indirect sources of inspiration are competition, potential competitors. Other startups, when they emerge, have a specific mission to address a particular problem, [...] they solve a problem that we don’t solve today, or that we solve poorly, or that we understand poorly.”

There was less developer involvement on benchmarking in user-driven startups, while in tech-driven startups, as in the case mentioned by P11, the engineering team is responsible for analyzing competition to identify new ideas:

“For example, sometimes we benchmark competitors. [...] We look at all of them, analyze, and compare. [...] Or sometimes, we say, ‘We need new ideas, let’s do an overview of the techniques they are using.’”

Doing Agile: Single Versus Dual-Track

Our interviewees highlighted that in their startups’ early stages, feature discovery and user validation were integrated directly into the development of a minimum viable product. In these phases, divergent and convergent thinking work in tandem, with product managers and designers both being tasked with exploring a wide range of possibilities (divergent thinking) and quickly converging on practical solutions (convergent thinking) through rapid feedback loops. For instance, P06 particularly emphasized lean startup¹⁴ principles when affirming that “in the beginning, it is practically chaos. We gather a lot of information [...] test and validate in practice [...] to create a solution that delivers value as quickly as possible.”

This flexible and fast-paced approach, blending divergent idea generation with convergent validation, is particularly suitable for smaller teams, but limits problem exploration since the solution is being quickly developed together. However, most startups we interviewed that had already passed the initial stage¹² adopted a continuous discovery and delivery model, often

referred to as dual-track agile.¹¹ In this model, ideation and development are done by the same team but work in different activity cycles. A part of a team, mostly composed of product and user interface designers, engages in creativity-centered activities on the discovery track, where they explore user-centered designs, brainstorm potential features (divergent thinking) and validate them with users (convergent thinking). This discovery “track” allows for deeper problem space exploration as it runs parallel to a delivery “track” where another part of the team (developers) focus on software development. Discovery and delivery activities have different timing. A product backlog works as a boundary object in this case. It is fed by the discovery track and consumed by the delivery track.

P03 exemplifies this dual-track approach by explaining how their previous startup ran one track focused on prototyping and another on product:

“In a previous startup that was already at a higher level of maturity, I started to separate prototypes from products. If I want to test a hypothesis, I create a separate prototype, usually with a control group, and if it worked or not, I would discard it or incorporate it into the final product. But this is when you already have a product with a large user base. To give you an idea, that startup (product) had almost a million users.” (P03 – CTO)

Dual-track agile¹¹ brings the possibility of teams working independently and focusing on their areas of expertise. However, such an approach can also create communication and

coordination issues among teams, since they work on each track in parallel as discussed by P12 below:

“[...] when you come out of a discovery activity like this, where you have such an in-depth creative process [...] you end up with a solution addressing various pain points. When you present it to the engineering and development team, they might say they can only implement about 25% of it. This creates an overhead because it means the other pain points won’t be resolved, sometimes because it involves a much larger architectural change and a longer development time.” (P12 – CPO)

Dual-track agile may not always be suitable. P03 notes that it can be unproductive for early-stage startups—“everything has to be done together.” Similarly, P13 shares that their 12-year tech-driven startup still thrives on a single-track approach, integrating divergent and convergent creativity within the same development workflow.

Developers are Feasibility Checkers in User-Driven Startups

In user-driven startups, product managers and designers lead the user-centered ideation phase exploring the problem space. However, developers’ high-level technical expertise can sometimes create a disconnect between the solutions they find straightforward to implement and what truly meets end-users’ needs. P06 highlighted this gap saying that “especially the people who are developing the product, their level of expertise with technology is very high. So, the type of solution they usually find practical to use in the software doesn’t generally have

a high level of empathy with the user.” This underscores the challenge of aligning technical solutions with user-centered design.¹¹ One way some teams address this is by involving developers in the early stages of product discovery, as the CTO of a game-focused platform explained:

“We usually brought in a developer to participate in the discovery phase [...] to help with the impact assessment. This impact mapping was always important during discovery for us to define things that were feasible [...] the devs were also very close to the solution. [...] our audience was young, and the people who worked with us were also young and liked to play games.” (P04 – CTO)

User-driven startups commonly discover product features through frequent validation and continuous experimentation. This process leverages divergent thinking by exploring various potential features and validating them through user-centered design approaches (e.g., A/B testing, validation with potential users, focus groups), which help to quickly validate user desirability and pivot a solution when needed. Considering the three principles of viability, desirability and feasibility,² product designers often focus on the first two while developers tend to focus on the third:

“[...] developers take the design that was created and then they kind of break down the whole structure. [...] Their main role is to verify feasibility. And not just feasibility, but also the productivity of that within the product. [...] We build a product, so it needs to

be easy to maintain, be scalable, secure, etc.” (P06 – CPO)

Such collaboration highlights how converging into achievable solutions relies on developers’ early involvement in assessing feasibility to avoid overhead. P12 explains the need of this *early* feasibility check:

“What you’re asking to do seems simple, seems feasible, but actually it isn’t. It means I will have to change the architecture of the state database if it happens. To deliver this, it will take about five or six months. Then, I asked ‘what can you manage to do to get close to this in two sprints?’” (P12 – CPO)

However, it is important to be clear about the exploratory nature of these activities when involving developers. When developers are highly focused on feasibility, if expectations are not aligned, they may become uncomfortable handling creative ideas. This can be due to an erroneous assumption of having to implement everything that is discussed with users:

“We tried to get the development team more involved with clients and participating more directly in the [creative] processes [...] But people started getting very anxious and stressed out because of things like ‘there’s so much to do, the work never ends, and the client wants a lot of things’. [...] it generated anxiety, stress.” (P01 – CEO)

The novelty of products and features is essential for success, and achieving it requires balancing diverse perspectives through cross-functional collaboration, which is fundamental for effective product

development among project managers, designers, and developers. In a single-track approach, developers are closer to discussions about feasibility with designers. In dual-track, requirements and feasibility are monitored and assessed continuously by designers. Still, if developers and designers work in parallel on separate tracks, it can introduce new communication and coordination challenges that must be managed.

Developers Lead the Creative Process in Technology-Driven Startups

Our findings show that in tech-driven startups (e.g., cloud, security), where products are designed for highly technical users, developers play a central role in shaping product features. In these startups, developers are the ones who understand better the problem space and often drive feature development by exploring potential innovations (divergent thinking) and refining them into practical solutions (convergent thinking). P13 cites many important features that came straight from the development team, and P17 mentions frequently shipping features that were not user-validated “that the client did not know about yet and certainly will take advantage of that.” P11, the CTO of a tech-driven startup supports this:

“Why are the tech leads or engineering managers in our company those responsible for building and prioritizing the product? Because I believe in it. First, our product is technical. Second, they have to be on the front line. They need to listen and hear firsthand. [...] One aspect of conception that I think worked very well for us is not leaving this role solely to the responsibility of one product person, but practically to the whole team.” (P11 – CTO)

Software developers’ feedback and creativity are so important in tech-driven startups that P11’s startup organizes internal hackathons to make sure product people “listen” to the developers to find out their vision about the startup and even integrate hackathon results as products:

“So, we organize hackathons every year. [...] I was telling you about the big initiatives, which are more top-down. But what we also want to organize is grassroots innovation. When I say grassroots, I mean engineers who think that something is a really great idea that we should pursue. If we wanted to productize that, well, here’s what we would need to do. [...] the next hackathon always starts with showcasing all the features from the last hackathon that we managed to implement.” (P10 – CTO)

LLM Tools Have Limited Creativity Support

Multiple participants mentioned using LLMs in the creative process. This was mainly done in various aspects of both product conception and software development, with notable applications in creative processes. However, while participants reported leveraging LLMs to enhance divergent thinking, there was limitation in convergent thinking.

In the divergent-thinking aspect of creativity, our participants particularly mentioned LLMs being utilized to leverage the creative naming of products or features and in marketing campaigns. They were particularly useful to explore the problem space, being reported as a replacement for traditional search engines to understand the market, competitors, and user behavior. This is evident by

e.g., P16 mentioning that “at some point we are also using it (ChatGPT) to ask questions about specific markets.” Additionally, participants mentioned using LLMs to support content production, decision making, and ideation, streamlining tasks such as generating marketing texts, developing strategies, and sketching designs in the cases we studied.

Interviewees also mentioned using LLMs for convergent thinking but discussed their limitations regarding this aspect of creativity:

“It was helpful, but still, it’s very limited. We’ve just started using the paid ChatGPT and feeding the tool with a knowledge base related to the startup. The results are getting a bit better, but still limited. The way it articulates is very circular. It hits a ceiling in terms of creativity and then just keeps regurgitating more of the same.”
(P02 – CPO)

LLM-powered tools are also used in software development to assist in programming activities like coding, testing, etc. In this context, P01 mentioned using LLM tools for creativity, i.e., divergent thinking to generate alternative solutions for specific application programming interface usage.

Recommendations for Practitioners

Based on study findings, we list a set of resulting recommendations to leverage developers’ involvement in the creative process of software startups.

R1. Enhance competition-inspired creativity. Encourage developers to engage actively in competitive analysis. By analyzing market gaps and competitors, developers can better

understand the problem space and generate ideas through methods like SCAMPER. This promotes divergent thinking, allowing them to brainstorm potential solutions inspired by competitor analysis and refine ideas into actionable features through convergent thinking.

R2. Balance the choice between single- and dual-track agile approaches. Startups should reflect on the tradeoffs between single- and dual-track agile methods. In a dual-track approach, assigning developers to the discovery track can enhance early-stage creativity. Single-track may work better in smaller or less mature startups, fostering closer interaction between problem exploration and technical feasibility discussions, while dual-track can help established teams focus on divergent and convergent thinking more independently.

R3. Involve developers early in the creative process of user-driven startups. Include developers in the discovery phase to help with feasibility checks and to foster empathy toward end-users. Such early involvement reduces feasibility risks later on, as developers are better positioned to understand technical constraints. However, startups should be mindful of potential stress that can arise when developers feel pressured to implement every idea discussed. They should emphasize that the discovery phase is about exploration, not implementation. This helps watch for creativity being sacrificed at the cost of overly cautious

feasibility perspectives by developers.

R4. Foster developer-led creativity in tech-driven startups. In tech-driven startups, developers can drive the ideation process due to their deep understanding of the technical problem space. Holding internal hackathons or “innovation days” where developers present new ideas through crowd-sourced divergent thinking in the exploration of problem-solution space can increase creative outcomes. It also allows them to lead solution development that aligns with both technical and business goals.

R5. Use LLMs to explore creativity in the problem space. There are opportunities to use LLMs in the early ideation phases and for divergent thinking. They can expand brainstorming possibilities and assess market trends, augmenting possibilities of creativity during early phases of product discovery. However, reported shortcomings suggest these tools remain limited in convergent thinking, reducing their potential to fully support a divergent/convergent thinking cycle.

This study highlights a perspective on how C-level executives in startups leverage developers in fostering creativity, focusing on flexible practices that balance divergent and convergent thinking. Developers contribute significantly to both feasibility checks in user-driven startups and technical innovation in tech-driven ones. While LLMs offer some creative support, they remain limited

in divergent thinking. The findings' generalizability is limited by the small, qualitative sample. Future research should broaden startup diversity and include developers' perspectives for comparison with C-level views. 

References

1. W. Brenner, F. Uebernickel, and T. Abrell, "Design thinking as mindset, process, and toolbox," in *Design Thinking for Innovation*, W. Brenner and F. Uebernickel, Eds. New York, NY, USA: Springer-Verlag, 2016, pp. 3–21.
2. T. Brown, *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. New York, NY, USA: Harper Collins, 2009.
3. T. Lindberg, C. Meinel, and R. Wagner, "Design thinking: A fruitful concept for IT development?" in *Design Thinking*, C. Meinel, L. Leifer, and H. Plattner, Eds. New York, NY, USA: Springer-Verlag, 2011, pp. 3–18.
4. R. Mohanani, P. Ram, A. Lasisi, P. Ralph, and B. Turhan, "Perceptions of creativity in software engineering research and practice," in *Proc. 43rd Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Piscataway, NJ, USA: IEEE, 2017, pp. 210–217, doi: [10.1109/SEAA.2017.21](https://doi.org/10.1109/SEAA.2017.21).
5. N. Cross, *Design Thinking: Understanding How Designers Think and Work*. Oxford, U.K.: Berg, 2011.
6. T. Sporse, A. Tklich, N. B. Moe, and M. Mikalsen, "Understanding barriers to internal startups in large organizations: Evidence from a globally distributed company," in *Proc. IEEE/ACM Joint 15th Int. Conf. Softw. Syst. Processes (ICSSP) 16th ACM/IEEE Int. Conf. Global Softw. Eng. (ICGSE)*, Piscataway, NJ, USA: IEEE, 2021, pp. 12–21, doi: [10.1109/ICSSP-ICGSE52873.2021.00011](https://doi.org/10.1109/ICSSP-ICGSE52873.2021.00011).

ABOUT THE AUTHORS



KIEV GAMA is an associate professor at the Universidade Federal de Pernambuco (UFPE), 50740-560, Recife, Pernambuco, Brazil. His research interests span specific topics in software engineering, including collaborative and human aspects, education, and broader interdisciplinary topics, such as open innovation. Gama received his Ph.D. in computer science from Université de Grenoble. He is a member of the IEEE Computer Society, ACM, and the Brazilian Computer Society. Contact him at kiev@cin.ufpe.br.



CLEIDSON R. B. DE SOUZA is associate faculty at the Universidade Federal do Pará, 66075110, Belém, Brazil. His research interests include collaborative and human aspects of software engineering and generative artificial intelligence for software engineering. de Souza received his Ph.D. in information and computer sciences from the University of California, Irvine. Contact him at cleidson.desouza@acm.org.



ALEXANDER NOLTE is an assistant professor at the Software Engineering and Technology cluster, Eindhoven University of Technology, 5612 AZ, Eindhoven, The Netherlands and an adjunct faculty at the Software and Societal Systems department, Carnegie Mellon University, Pittsburgh, PA, 15213, USA. His research takes place at the intersection of software engineering and human–computer interaction and focuses on understanding and developing means to support collaboration in corporate, entrepreneurial, civic, educational, and scientific settings. Nolte received his Ph.D. in information systems from the University of Duisburg-Essen. Contact him at a.u.nolte@tue.nl.



GEORGE VALENÇA is an associate professor at the Department of Computing, Universidade Federal Rural de Pernambuco (UFRPE), 52171-900, Recife, Pernambuco, Brazil. He currently works as a chief scientist for technical cooperation agreements with the Alana Institute, São Paulo/São Paulo/Brazil, 05416-000 and with the Court of Accounts of Pernambuco, Recife/Pernambuco/Brazil, 50050-910. He leads the Sociotechnical Aspects on Software Platforms (ASPAS) group. His research interests include ethics in software development, children's rights by design, and human–computer interaction. Valença received his Ph.D. in computer science from Universidade Federal de Pernambuco (UFPE). Contact him at george.valenca@ufrpe.br and @prof.georgevalenca on Instagram.

7. T. M. Amabile, *How to Kill Creativity*. Brighton, MA, USA: Harvard Business School Publishing, 2018.
8. F. Dobrigkeit, D. de Paula, and N. Carroll, “InnoDev workshop: A one day introduction to combining design thinking, lean startup and agile software development,” in *Proc. IEEE 32nd Conf. Softw. Eng. Educ. Training (CSEET)*, Piscataway, NJ, USA: IEEE, 2020, pp. 1–10, doi: [10.1109/CSEET49119.2020.9206184](https://doi.org/10.1109/CSEET49119.2020.9206184).
9. N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, “Software development in startup companies: A systematic mapping study,” *Inf. Softw. Technol.*, vol. 56, no. 10, pp. 1200–1218, 2014, doi: [10.1016/j.infsof.2014.04.014](https://doi.org/10.1016/j.infsof.2014.04.014).
10. S. S. Bajwa, X. Wang, A. Nguyen Duc, and P. Abrahamsson, “‘Failures’ to be celebrated: An analysis of major pivots of software startups,” *Empirical Softw. Eng.*, vol. 22, no. 5, pp. 2373–2408, 2017, doi: [10.1007/s10664-016-9458-0](https://doi.org/10.1007/s10664-016-9458-0).
11. T. Sedano, P. Ralph, and C. Péraire, “Dual-track development,” *IEEE Softw.*, vol. 37, no. 6 pp. 58–64, Nov./Dec. 2020, doi: [10.1109/MS.2020.3013274](https://doi.org/10.1109/MS.2020.3013274).
12. M. Crowne, “Why software product startups fail and what to do about it. Evolution of software product development in startup companies,” in *Proc. IEEE Int. Eng. Manage. Conf.*, 2002, vol. 1, pp. 338–343, doi: [10.1109/IEMC.2002.1038454](https://doi.org/10.1109/IEMC.2002.1038454).
13. C. Müller-Roterberg, *Handbook of Design Thinking*. Seattle, WA, USA: Kindle Direct Publishing, 2018.
14. E. Ries, *The Lean Startup*. New York: Crown Business, 2011, pp. 2016–2020.



IEEE Security & Privacy magazine provides articles with both a practical and research bent by the top thinkers in the field.

- stay current on the latest security tools and theories and gain invaluable practical and research knowledge,
- learn more about the latest techniques and cutting-edge technology, and
- discover case studies, tutorials, columns, and in-depth interviews and podcasts for the information security industry.



computer.org/security

