SILESIAN UNIVERSITY OF TECHNOLOGY PUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2023
ORGANIZATION AND MANAGEMENT SERIES NO. 176

LITERATURE REVIEW OF CONTINUOUS DELIVERY: RESEARCH DIRECTIONS FOR CRITICAL INFRASTRUCTURE SOFTWARE PROJECTS

Piotr GODZIEWSKI

Wroclaw University of Science and Technology; piotr.godziewski@gmail.com, ORCID: 0000-0001-8912-2052
Nokia Poland; piotr.godziewski@nokia.com

Purpose: The purpose of this work is to draw future research directions on how to ease adoption of continuous delivery (CD) for business-to-business (b2b) critical infrastructure products. CD is a recognized software lifecycle management practice reducing go-to-market time, strengthening customer feedback loop, and improving product quality. Telecommunication networks, considered critical infrastructure, are sensitive to changes in delivery models.
Design/methodology/approach: Literature review was performed by combining bibliometric analysis and the own model gauging telecom software vendors' interest in shaping CD practices across the industry.
Findings: The research is skewed toward engineering practices excellence. Little is spent on the customer challenges. Transformation slowdowns are attributed to product teams.
Research limitations/implications: Some software vendors, especially smaller ones, may prefer not to publish the outcomes before validating them with the customers. This work looked at publicly available materials therefore not capturing the picture of internal corporate experimentation on continuous delivery.
Practical implications: Scientists should seek access to customer perspective. Sales, services, and business managers may be invaluable proxies of such information.
Originality/value: This work nudges the community to shift focus from R&D excellence to change management at customer interface, and to deal with CD model industrialization aspects.

Keywords: Continuous delivery, critical infrastructure, devops, agile, telecommunication networks.

Category of the paper: Literature review.

1. Introduction

Telecommunication networks are strategic for proper functioning of states, public safety organizations, enterprises, and citizens. COVID-19 pandemic emphasized the importance of critical communication infrastructure as digitization enabler. 5G, the fifth-generation mobile broadband technology, introduces new services and use cases. For example, home broadband was the most appealing 5G application among 52% respondents surveyed in (The Mobile Economy 2021, 2021). Introducing novel services puts speed of experimentation in the center of business case modeling for communication service providers (CSP). Value hypothesis must translate to product capabilities with short go-to-market time to gain competitive advantage. System complexity requires all network segments (i.e., radio access, transport, core) to be adaptable to the changing business demand. Providing small, iterative, frequent product changes to customer is the merit of continuous delivery. The practice is widely used in business-to-consumer (b2c) space. Its principle has much in common with the culture of intelligent fast failures and reuses many techniques known from commercializing rapid innovations (Czerwinska-Lubszczyk et al., 2022). When we replace end user of digital product with business entity, the CD concept gets more sophisticated but still feasible to serve the go-to-market time requirement.

Networks fall under the classical, and legislative, definition of critical infrastructure. (Cybersecurity & Infrastructure Security Agency, 2020) identified 16 critical infrastructure sectors, telecommunication networks among them, which *are considered so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof*. (Presidential Policy Directive -- Critical Infrastructure Security and Resilience, 2013) called for actions to secure communications systems due to *the enabling functions they provide across all critical infrastructure sectors*. We found similar definition in (Communication from the Commission to the Council and the European Parliament - Critical Infrastructure Protection in the Fight against Terrorism, 2004). Regulatory requirements such as emergency call handling, network coverage, service availability, are all in the center of CSP operations. Commercial acceptance of major software change becomes highly restrictive process. This stands in opposition to continuous delivery concept as we know it from b2c products.

Network infrastructure is also at the forefront of companies striving for extreme automation and digitization. 4G and 5G allow building private networks for exclusive use by the enterprises. This trend prioritizes improved security and privacy for mission-critical communication.

The aim of this article is to review the studies which connect the realms of product engineering and managing its commercialization. At this point, it is hard to say if existing research base is enough to determine the recipe for industrializing CD concept in b2b environment of critical network infrastructure. In the next section, we will look at the research question supporting the goal. Next, the method will be presented describing repeatable data collection protocol, bibliometric analysis, and own model evaluating research activity of telecom software suppliers. The results will cover literature mapping and walkthrough of the most influential groups of literature. This will be completed by the analysis of research work affiliated with telecom software vendors. Discussion will highlight the focal points of the analyzed literature set. Finally, we will point out limitations of this work and wrap up future research directions.

The originality of the material comes from looking at the scientific knowledge base through the prism of actors at the customer interface, i.e., stakeholders responsible for the ultimate delivery to the customer. Earlier literature studies focused on the applicability of CD models in product organizations.

1.1. Research question development

Continuous delivery is a long-established practice in software industry. Often, as in (Ståhl et al., 2017), it refers strictly to activities in research & development (R&D) department. Deployment to customer and release to end users happen afterwards (Johanssen et al., 2018) classified CD as the core element of continuous engineering. More holistic definition appeared in (Humble, Farley, 2010). The CD concept binds the phases of building, testing, and deploying, for the end goal of delivering software more frequently. Continuous delivery, whether defined as R&D process or a holistic delivery framework, is merged with technical, cultural, and management aspects of the product development organization.

RQ: We will ask to what extent the CD research invite non-engineering topics driving go-to-market strategies and operating models at the customer interface? This emerged from observing the disconnect between continuous delivery capabilities in product line, and the ability to commercialize such continuous value flow toward customer. Our context is the b2b nature of critical networks software.

2. Method

Bibliometric analysis was performed to cover large set of literature positions ranging in thematic scope (Donthu et al., 2021). Screening, inclusion and exclusion criteria, and data cleansing followed recommendations from (Barends et al., n.d.). Mapping and reporting, including use of tools, followed (Linnenluecke et al., 2020). The protocol was augmented with selecting publications affiliated with commercial software suppliers. This step, if executed in isolation, would have been highly biased and of little value. However, the goal was to verify if, and how, telco vendors invest in continuous delivery research. This method combined the realms of academic research and industry.

The search phrase had to be broader than the continuous delivery term to capture interlinked terms such as continuous release, deployment. Wildcards (*, $) are used to account for lexeme variations. Title, abstract, and keywords fields were analysed. Screening resulted in 9876 publications.

Timeframe was limited to last ten years (2012-2022). Oztemel & Gursev (2020) provided the synthesis of prior work, highlighting cloud computing as the key catalyst of new delivery models. Software-intensive projects and customer aspects were the focus thus software and customer phrases were explicitly included. Only proceeding papers, articles, and early access publications were filtered. Categories irrelevant for the study were excluded, leaving subjects of computer science, business, management, and operations research.

Table 1.
Data collection protocol

Web of Science		**Scopus**	
Search phase	**Dataset size**	**Search phase**	**Dataset size**
Initial screening			
("contin*s deliver*" OR "contin*s deploy*" OR "contin*s releas*" OR "contin*s exploration" OR "contin*s experiment*") (Topic)	4244	TITLE-ABS-KEY ("contin*s deliver*" OR "contin*s deploy*" OR "contin*s releas*" OR "contin*s exploration" OR "contin*s experiment*")	5632
Inclusion criteria			
2012-2022 (Year Published)	2649	PUBYEAR > 2011 AND PUBYEAR < 2023	3313
software OR customer (All Fields)	567	ALL (software OR customer)	1045
Document type: • Proceeding Paper, • Article, • Review Article, • Early Access.	566	Document type: • Conference Paper, • Article, • Conference Review, • Review.	1007
Language: • English.	561	Language: • English.	984
Exclusion criteria			
WoS Category NOT: • Computer Science (and all its subcategories), • Telecommunications, • Business, • Management, • Operations Research Management Science, • Engineering Multidisciplinary, • Multidisciplinary Sciences.	399	Subject area NOT: • Computer Science, • Decision Sciences, • Business, Management, and Accounting, • Social Sciences, • Multidisciplinary.	508
Merging and duplicates removal			
Data set size: 674			

Source: own work.

Web of Science and Scopus results were exported to BibTeX files, converted to xlsx format using the bibliometrix library of R Studio based on (Moral-Muñoz et al., 2020). Merging with duplicates removal was performed in R Studio. The output xlsx file was sent to biblioshiny for analysis and visualization.

The final WoS and Scopus queries, last row of Table 1, were modified by one more inclusion criterion to extract literature affiliated with major telecommunication vendors. The subset included publications associated with Ericsson, Nokia, Cisco, and Huawei. They are among the market leaders.

3. Results

The literature was dominated with conference papers (472 items), followed by articles in peer-reviewed journals (159 items). The ACM International Conference Proceedings Series along with The International Conference on Software Proceedings were the most used conferences, while The Information and Software Technology by Elsevir was the top peer-reviewed journal. Figure 1 presents the evolution of research space over time and its distribution across four major sources.

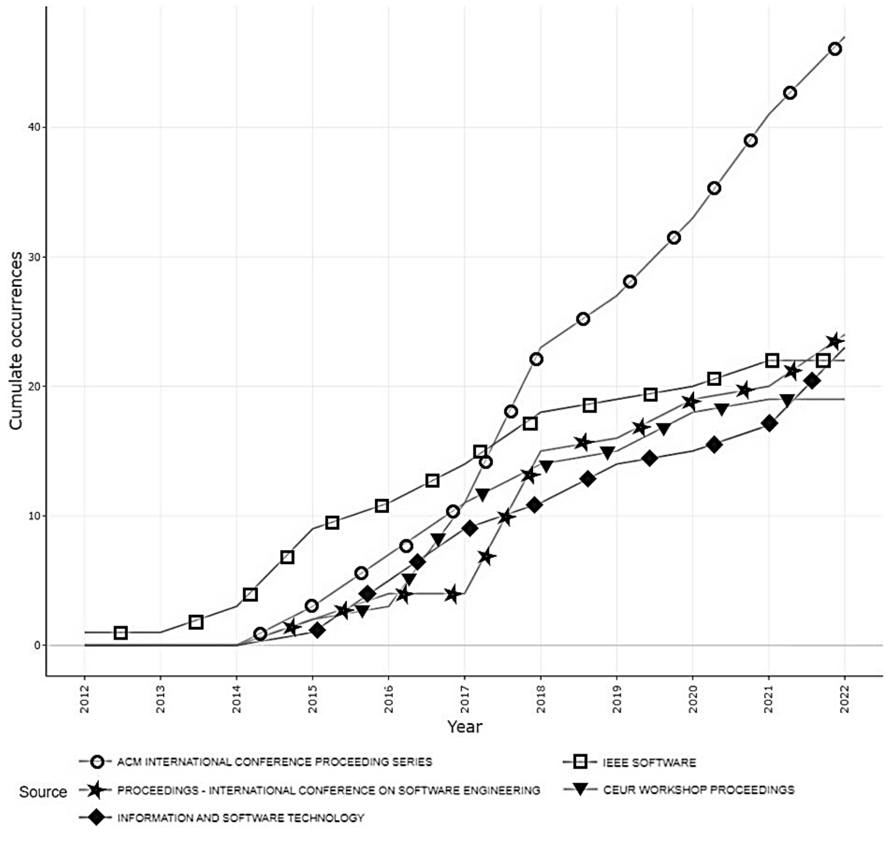

Figure 1. Major sources of continuous delivery literature.

Source: own work, biblioshiny software.

Jan Bosch, from Chalmers University of Technology, consistently co-authored high number of publications, most of which were the case studies with b2b and b2c companies. As illustrated on Figure 2, Helena H. Olsson was another key contributor and co-created much of the research with Jan Bosch. The two Swedish scientists leveraged proximity of large-scale software organizations in automotive, telecommunication, and military defence sectors. The sequence of their most relevant work started with the Stairway to Heaven model (Olsson et al., 2012), the conceptual roadmap of transitioning software organization through the stages of continuous practices, from integration, to delivery, and experimentation. The EMFIS model proposed in (Martensson et al., 2017) defined a maturity assessment matrix. Its components were decided based on interviews with practitioners from automotive and telecommunication companies, Saab and Ericsson. An interesting detour from engineering practices was found in (Ståhl, Bosch, 2017). It proposed the Cinders framework which was the collection of recommendations for documenting, investigating, and communicating about continuous integration and delivery systems across the R&D organization. More recently, the group changed its focus to continuous experimentation practices. The HURRIER process in (Mattos et al., 2020) came from the case study in Ericsson. It provided actionable techniques in four groups of activities:

- Project management of incremental development in R&D organization resulted in better availability of the software product.
- Internal product verification ensured end-to-end quality.
- Early validation was restricted to single customer, carefully selected based on customer relationship.
- Final validation with multiple customers took place during gradual rollouts.

The HURRIER framework promoted early exposure to field issues and required customer feedback to be embedded in the process. It incentivized shorter cycles of continuous experimentation. Sceptics of such transformation should note that the case study took place in the R&D of 4G product, key system of critical network infrastructure. A holistic view, called the Controlled Continuous Delivery (CCD), was provided in (Dakkak, Bosch et al., 2022). It connected success probability of continuous delivery adoption with type of customer segment, and stage of product lifecycle. Only small group of innovators and early adopters was likely to embrace high-frequency continuous delivery. The CD practice was believed to be more business relevant in introduction and growth phase of product lifecycle. When product matured, or even declined (e.g., 3G), it became less appealing to push for short CD cycles.

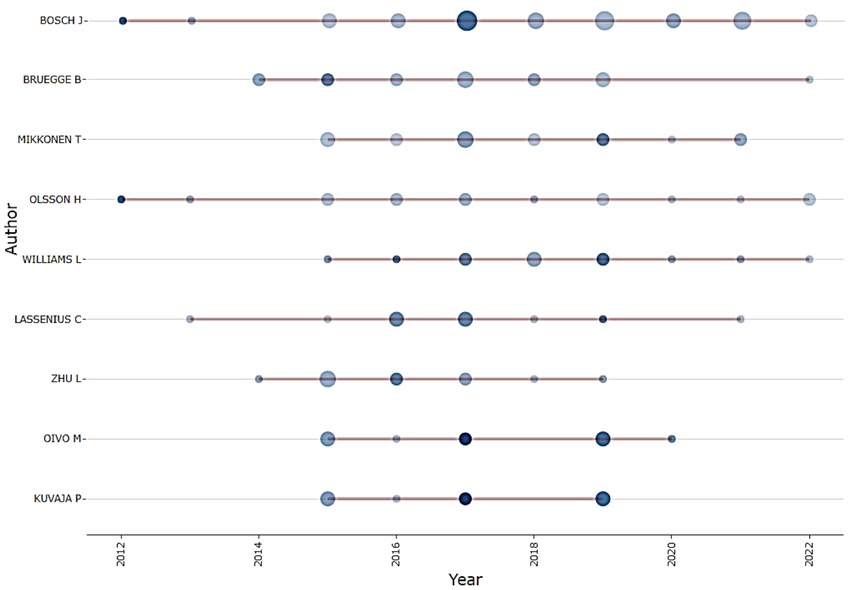

Figure 2. Publication intensity of the key contributors over time.

Source: own work, biblioshiny software.

The bulk of publications co-authored by Bernd Brügge was found less relevant in the b2b context. Higher education didactics, covered in (Alperowitz et al., 2016) and (Schmiedmayer et al., 2022), remains outside this review's scope but we should recognize its importance. Having young engineers experience CD way of working during student assignments, will likely ease CD adoption later when the graduates join companies or establish their own digital businesses.

Figure 3. Word cloud.

Source: own work, biblioshiny software.

A look at the word cloud on Figure 3 tells us that researchers see the DevOps concept closely related with CD. Its strong presence in the dataset deserves explanation. In (Debois, 2008), the author, by many considered the father of DevOps, called for a novel way of managing software projects by integrating infrastructure work (e.g., setting up the underlaying hardware) and operations work (e.g., supporting customer issues) into software engineering project. Today, popularity of the DevOps concept makes it more than obvious that there should be a structured bond between development and operations. It was an emerging concept in 2008. (Leite et al., 2019) provided systematic literature review in four dimensions: people management, process and project management, product delivery, and engineering practices. The authors mapped multiple concepts associated with DevOps culture, walked through the toolset for achieving the required system architecture readiness, and pointed out implications to organizational management and operations research. The interlinking of DevOps and CD was evident in the definition developed by the researchers, i.e., understanding DevOps as collaborative and multidisciplinary effort which enables continuous delivery of high-quality software. (Claps et al., 2015) analysed not only technical but also social challenges when adopting CD. In that case study, the authors were convinced about the need for future research at the business level, to explore headwinds faced by customer during CD roll-out.

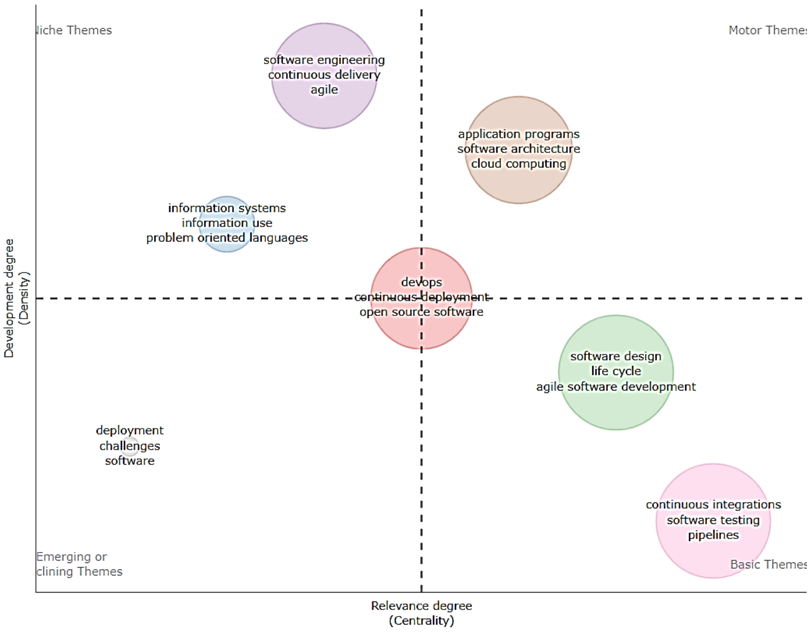

Figure 4. Thematic map of the degrees of relevance and development maturity.

Source: own work, biblioshiny software.

Figure 4 informs us about the relationship between thematic clusters, their relevancy and maturity. Centrality reflects the strength of relationships to other clusters. The higher centrality, the more relevant something is across the research space. There were three themes, with interaction to the rest of literature, stronger than the central devops cluster:

- Testing cluster made of continuous integration, software testing, and CI/CD pipelines.
- Design cluster with references to life cycle management, and agile project management.
- Architecture cluster including cloud computing, and software architecture.

The second dimension is the density which defines the strength of relationships within a thematic group. The higher density, the more developed and mature something is. The design and testing clusters were less mature (lower density) compared to the architecture theme as they presented more basic terms. We may think of them as enablers, therefore appearing in most of the studies (high centrality). On the contrary, the architecture topic was classified as a motor theme because of its maturity (high density) and its relevance (high centrality). Software engineering cluster, including continuous delivery and agile, was the most developed group, but less relevant for the rest compared with devops cluster, including continuous deployment.

Twenty-nine publications were left after filtering for Ericsson-affiliated work. They were reduced to twenty after removing contextual duplicates i.e., follow-up studies which added nuance on top of the original work.

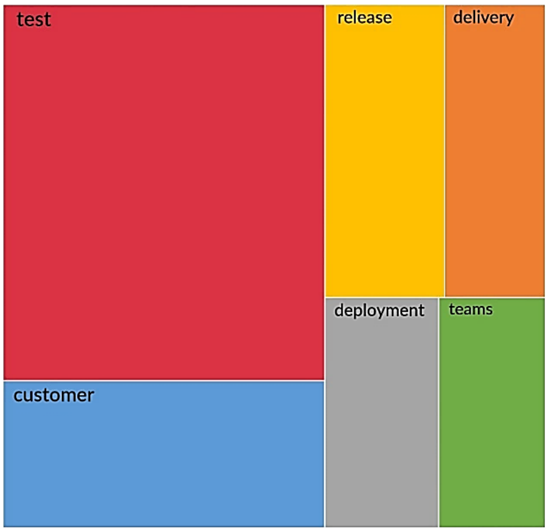

Figure 5. Heatmap of coding references in Ericsson-affiliated research.

Source: own work, NVivo software.

More than 60% of coded statements on **Błąd! Nie można odnaleźć źródła odwołania.** referred to testing, either exploratory with customer, or continuous integration. There was much focus on embedding exploratory tests in day-to-day R&D work by promoting session-based testing in development teams, and scenario-based testing with end users (Mårtensson et al., 2017). Not only it improved the understanding of requirements, but acted as a catalyst of frequent, short-cycle test rounds with customer. That allowed cultivating continuous collaboration culture – important mindset element to start talking about continuous delivery. Publications related to managing testing activities created a sequence of proposals:

- The Cinders framework provided recommendations on documenting, investigating, and communicating continuous integration pipeline system across R&D (Ståhl, Bosch, 2017),
- The EMFIS model was a maturity model, evaluating state of continuous integration practices (Mårtensson et al., 2018). Any gap between developer's perception and process owner's assumption was especially valuable to the transformation leaders.
- The TAS model (Mårtensson, Ståhl et al., 2019) had its roots in a classical test pyramid. There are different stakeholders associated with each testing level (e.g., unit testing,

component testing, system testing). The model analysed their needs and recommended how to adjust stakeholder management with outputs from various test levels.

- The ExET framework (Mårtensson et al., 2021a) was about visualizing exploratory testing outcomes and driving the corresponding enhancements in large-scale software projects.
- The MaLET model (Mårtensson et al., 2021b) provided step-by-step guideline on improving exploratory testing through permission governance, competence development, results distribution, and collaboration.
- Finally, (Ståhl, Mårtensson, 2021) pointed out that the test automation cannot be seen as an end itself. The next level is to focus on the most strategic, impactful test suites, with continuous benefit-vs-cost evaluation. The authors recommended investigating corporate tensions which might hinder managing the organization in one direction.

Deep dive to coding references under *Teams* category, on **Błąd! Nie można odnaleźć źródła odwołania.**, tells us that *development* and *exploratory test teams* are the ones most covered by research. What follows next, i.e., *customer support team lead* and *cross-functional teams*, is the evidence of research efforts expanding into non-development areas.

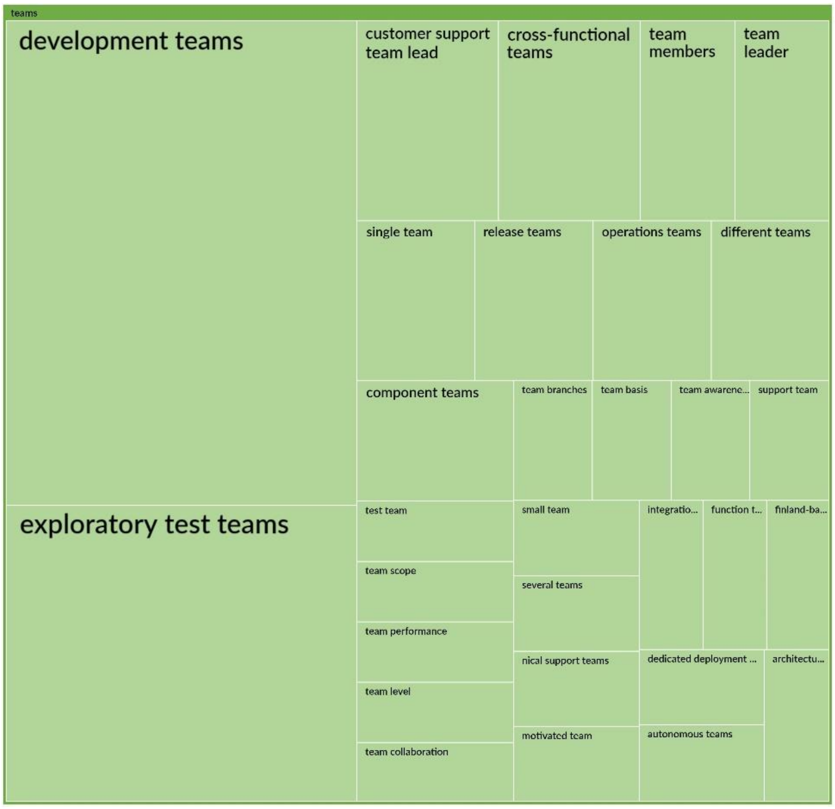

Figure 6. Heatmap of Teams codes in Ericsson-associated research.

Source: own work, NVivo software.

Table 2 presents the density of key coding references. Top five articles were the qualitative case studies performed at Ericsson. All except one were the interviews with practitioners.

Interviewees came from product line. In the most comprehensive campaign in (Mattos et al., 2020) the researchers interviewed customer solutions manager, specialist typically part of sales or pre-sales. All top five articles studied Radio Access Network product which is a common characteristic with the rest of Ericsson-affiliated literature. (Mårtensson et al., 2021a) provided the most mature structure of continuous delivery governance for all three mobile network generations, i.e., 3G, 4G, and 5G. It is also one of the few publications that augmented qualitative study with quantitative data. Performance indicators informing about CD process were analysed along with quality management metrics such as the number of defects at various development and delivery phases. According to the authors, customers willing to experiment with new features, early in the product lifecycle, had the highest chance of successful continuous delivery adoption. There were two other dimensions critical for CD transformation: risk management (e.g., managing deliveries in a limited low-risk network cluster, often called CD zone), and engineering excellence (i.e., development organization producing high-quality frequent software candidates for immediate delivery to the CD zone).

Table 2.
Number of coding references in Ericsson-affiliated literature positions

Item	Coding references					
	Customer	**Delivery**	**Deployment**	**Release**	**Teams**	**Test**
(Dakkak, Munappy et al., 2022)	33	2	8	10	5	1
(Issa Mattos et al., 2021)	19	1	8	10	3	7
(Mattos et al., 2020)	17	0	6	4	4	5
(Dakkak et al., 2021b) :	11	3	9	2	5	.5
(Dakkak, Bosch et al., 2022)	10	1	15	17	1	6
(Kasauli et al., 2017)	9	2	0	1	2	3
(Klotins et al., 2022)	8	15	1	5	0	13
(Dakkak et al., 2021a)	7	3	5	5	3	1
(Çalikli et al., 2018)	6	3	0	21	0	28
(Ståhl et al., 2016)	1	7	1	2	0	3
(Mårtensson, Ståhl, et al., 2019)	1	4	2	3	3	49
(Ståhl, Mårtensson, 2021)	1	3	3	3	0	29
(Ståhl, Bosch, 2017)	0	17	2	1	2	7
(Ståhl et al., 2017)	0	4	8	7	1	5
(Mårtensson, Stahl et al., 2019)	0	2	0	1	6	0
(Mårtensson et al., 2021a)	0	5	1	1	1	20
(Mårtensson et al., 2018)	0	4	2	3	9	21
(Mårtensson et al., 2017)	0	1	1	0	10	54
(Mårtensson et al., 2021b)	0	4	0	2	6	74

Source: own work.

Four articles were associated with Nokia Bell Labs, the telecom vendor's research arm. Two of them focused on security aspects during deployment phase (Martin et al., 2018; Combe et al., 2016). They analysed new attack surface introduced by deployment automation tools such as Docker, broadly used in CI/CD systems. Mijumbi et al. (2018) presented a model for predicting number of defects from the patterns of story point completion. Practitioners however may find such models too academic, and hard to apply in real-world software projects. Grohmann et al. (2019) proposed a machine learning model deriving application KPIs from platform KPIs. While those topics brought value to R&D, they did not connect with customer or commercial aspects of continuous delivery.

4. Discussion

We studied large set of academic work on continuous delivery with steady inflow of new publications. Industry practitioners are supported with non-scientific literature as well as plethora of academic papers, often published in collaboration with business. Systematic analysis revealed most of the work to be centered around variants of continuous engineering practices: continuous integration, continuous testing, continuous experimentation, continuous delivery. When they are put in use in product development organization, they form iterative sequence of managing software production: building, testing, deploying, releasing, and delivering. Such value stream depends more on the R&D culture than organizational productivity. This is why the research was highly coupled with Agile methodologies and DevOps culture, while there was little to no connection with operational phenomena at the customer interface. Both Agile and DevOps are critical forces shaping product line organizations of today and enable adoption of CD. This work however is put in the context of b2b delivery of critical network infrastructure products. In this case, R&D organizations rarely deliver their output directly to customer, and they may have limited knowledge about operating their product in the field.

We saw the language of literature dominated by the terms familiar to product development managers and experts. This implicates that the insights were skewed towards engineering processes and R&D organizations. Table 3 provides an exemplary mapping, which I developed in the course of data analysis, to quickly gauge whether an article was anchored in product development topics (e.g. engineering practices) or in business aspects (e.g. pre-sales, sales, services, customer relationship management).

Table 3.
Product line (R&D) and business teams speak different language

	Product line	Business teams
Mindset	– Agile, – DevOps.	– Go-to-market strategy, – Value-based selling.
	– Engineering challenges (e.g., feature development).	– Customer opportunities (e.g., upselling).
Project management governance	– Continuous, – Program management, – OPEX & CAPEX planning.	– On-demand, – Project management, – Topline and sales margin quality.
Tools, systems	– Customer environment configuration, – CD pipeline.	– Type of customer environment, – Digital delivery.
Practices	– Requirements engineering (system engineers).	– Customer engagement (technical pre-sales).

	– New product introduction, – Fault management.	– Customer acceptance, – Care services.
	– Release, – Deployment, – Delivery.	– Planning services, – Deployment services, – Integration.

Source: own work.

In the analysis phase, we talked about multiple models for managing continuous engineering flavors in product development organization. Those, derived from in-depth interviews and case studies of b2b large-scale software companies, provide actionable frameworks for R&D transformation leaders to drive continuous delivery adoption, at least in product line.

RQ: to what extend does the concept of continuous delivery invite non-technical, non-technological aspects which drive the actual delivery to customer? This review shows that the continuous delivery concept is considered mainly an engineering practice. It is true for the wide set of literature as well as publications associated with critical infrastructure vendors. On the other hand, the state of CD in telecommunication software projects suggests we may need to do things differently, to drive its adoption. To build upon existing knowledge base, new research should ask how to connect product development organizations, already fluent in continuous delivery practice, with customer frontend. Most recent academic work starts shifting in this direction.

5. Summary

We reviewed the state of academic work on continuous delivery in the context of critical infrastructure. Telecommunication sector requires software products of high architectural complexity, consisting of many interdependent subsystems, delivered from software vendor to communication service provider in b2b relationship. We started with three dimensions of criticality in network infrastructure. Requirement for agility drives shorter go-to-market time in consumer segment, giving CSPs competitive edge with higher throughput, better voice call quality, and new 5G services. Public safety and regulatory institutions pay extra attention to network reliability (e.g., five-nines availability). Security and privacy are the key business themes for enterprises interested in private network solutions (e.g., factory automation, campus networks).

Number of publications dictated the use of bibliometrics technique. First, we looked at the relevant body of knowledge retrieved from Scopus and Web of Science. The second part was to deep dive to academic work associated with telecom software vendors.

We looked at the literature review output through the lenses of author's professional experience. The focus of academic community has been on excelling continuous delivery practices in R&D. This is absolute pre-requisite to have product development capable of delivering high-quality software packages in short cycles. Handful of studies touched upon the processes associated with customer support, or product management. We concluded that engineering practices, increasing the chances of successful CD adoption, were comprehensively covered. Researchers may now pivot to specific types of software products (e.g., autonomous vehicles, intelligent electricity grids, telecommunication) and what it takes to enable continuous delivery in those b2b digitalization segments.

Future studies could develop in two directions. More questions about customer interface will be useful. That means targeting pre-sales, sales, market teams, and the corresponding practices, roles, organizations. Cross checking new findings with established opinions of the product development community could reveal gaps in end-to-end operating models.

Models and recommendations, which we discussed, were mostly based on case studies. Scientists with access to industry could increase the quality of research by employing qualitative methods such as in-depth interviews and focus groups. Quantitative analysis will require access to sensitive corporate data (e.g., installed base information, business value behind product features, go-to-market time). Data completeness, due to lack of systematic data collection mechanism in place, may limit its use. In this case, even partial data models could enrich studies which typically miss measurable outcomes.

References

1. Alperowitz, L., Dzvonyar, D., Bruegge, B. (2016). *Metrics in Agile project courses.* Proceedings - International Conference on Software Engineering, 323-326. https://doi.org/10.1145/2889160.2889183
2. Barends, E., Rousseau, D.M., Briner, R.B. (n.d.). *CEBMa center for Evidence-Based Management Rapid Evidence Assessments in Management and Organizations*. www.cebma.org/guidelines/.
3. Çalikli, G., Staron, M., Meding, W. (2018). *Measure early and decide fast: Transforming quality management and measurement to continuous deployment.* ACM International Conference Proceeding Series, 51-60. https://doi.org/10.1145/3202710.3203156.
4. Claps, G.G., Berntsson Svensson, R., Aurum, A. (2015). On the journey to continuous deployment: Technical and social challenges along the way. *Information and Software Technology, 57(1),* 21-31. https://doi.org/10.1016/j.infsof.2014.07.009.
5. Combe, T., Martin, A., Di Pietro, R. (2016). To Docker or Not to Docker: A Security Perspective. *IEEE Cloud Computing, 3(5),* 54-62. https://doi.org/10.1109/MCC.2016.100.
6. Communication from the Commission to the Council and the European Parliament - *Critical Infrastructure Protection in the fight against terrorism* (2004). Testimony of

European Commission. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex: 52004DC0702

7. Cybersecurity & Infrastructure Security Agency (2020). *Critical Infrastructure Sectors*. https://www.cisa.gov/critical-infrastructure-sectors.
8. Czerwinska-Lubszczyk, A., Grebski, M.E., Grebski, W., Jagoda-Sobalak, D., Krawczyk, D., Kuzior, A., Wolniak, R. (2022). *Creativity and Innovativeness in Psychology and Management, Vol. 1.* Towarzystwo Naukowe Organizacji i Kierownictwa.
9. Dakkak, A., Bosch, J., Olsson, H.H. (2022). *Controlled Continuous Deployment: A Case Study From The Telecommunications Domain.* ACM International Conference Proceeding Series, 24-33. https://doi.org/10.1145/3529320.3529323.
10. Dakkak, A., Mattos, D.I., Bosch, J. (2021a). *Perceived benefits of continuous deployment in software-intensive embedded systems.* Proceedings - 2021 IEEE 45th Annual Computers, Software, and Applications Conference, COMPSAC 2021, 934-941. https://doi.org/10.1109/COMPSAC51774.2021.00126.
11. Dakkak, A., Mattos, D.I., Bosch, J. (2021b). *Success Factors when Transitioning to Continuous Deployment in Software-Intensive Embedded Systems.* Proceedings - 2021 47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, 129-137. https://doi.org/10.1109/SEAA53835.2021.00025.
12. Dakkak, A., Munappy, A.R., Bosch, J., Olsson, H.H. (2022). *Customer Support In The Era of Continuous Deployment: A Software-Intensive Embedded Systems Case Study.* Proceedings - 2022 IEEE 46th Annual Computers, Software, and Applications Conference, COMPSAC 2022, 914-923. https://doi.org/10.1109/COMPSAC54236.2022.00143.
13. Debois, P. (2008). *Agile infrastructure and operations: How infra-gile are you?* Proceedings - Agile 2008 Conference, 202-207. https://doi.org/10.1109/Agile.2008.42.
14. Donthu, N., Kumar, S., Mukherjee, D., Pandey, N., Lim, W.M. (2021). How to conduct a bibliometric analysis: An overview and guidelines. *Journal of Business Research, 133*, 285-296. https://doi.org/10.1016/j.jbusres.2021.04.070.
15. Grohmann, J., Nicholson, P.K., Iglesias, J.O., Kounev, S., Lugones, D. (2019). Monitorless: *Predicting performance degradation in cloud applications with machine learning.* Middleware 2019 - Proceedings of the 2019 20th International Middleware Conference, 149-162. https://doi.org/10.1145/3361525.3361543.
16. Humble, J., Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.* Addison-Wesley Professional.
17. Issa Mattos, D., Dakkak, A., Bosch, J., Olsson, H.H. (2021). The HURRIER process for experimentation in business-to-business mission-critical systems. *Journal of Software: Evolution and Process*. https://doi.org/10.1002/smr.2390.
18. Johanssen, J.O., Kleebaum, A., Paech, B., Bruegge, B. (2018). *Practitioners' eye on continuous software engineering: An interview study*. ACM International Conference Proceeding Series, 41-50. https://doi.org/10.1145/3202710.3203150.

19. Kasauli, R., Knauss, E., Nilsson, A., Klug, S. (2017). *Adding Value Every Sprint: A Case Study on Large-Scale Continuous Requirements Engineering.*
20. Leite, L., Rocha, C., Kon, F., Milojicic, D., Meirelles, P. (2019). A survey of DevOps concepts and challenges. *ACM Computing Surveys, Vol. 52, Iss. 6.* Association for Computing Machinery. https://doi.org/10.1145/3359981.
21. Linnenluecke, M.K., Marrone, M., Singh, A.K. (2020). Conducting systematic literature reviews and bibliometric analyses. *Australian Journal of Management, Vol. 45, Iss. 2,* pp. 175-194). SAGE Publications Ltd. https://doi.org/10.1177/0312896219877678.
22. Mårtensson, T., Ståhl, D., Bosch, J. (2017). *Exploratory testing of large-scale systems – Testing in the continuous integration and delivery pipeline*. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10611 LNCS, 368-384. https://doi.org/10.1007/978-3-319-69926-4_26.
23. Martensson, T., Stahl, D., Bosch, J. (2017). *The EMFIS model - Enable more frequent integration of software.* Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, 10-17. https://doi.org/10.1109/SEAA.2017.31.
24. Mårtensson, T., Ståhl, D., Bosch, J. (2018). Enable more frequent integration of software in industry projects. *Journal of Systems and Software, 142*, 223-236. https://doi.org/10.1016/j.jss.2018.05.002.
25. Mårtensson, T., Ståhl, D., Bosch, J. (2019). Test activities in the continuous integration and delivery pipeline. *Journal of Software: Evolution and Process, 31(4).* https://doi.org/10.1002/smr.2153.
26. Mårtensson, T., Stahl, D., Martini, A., Bosch, J. (2019). *Continuous architecture: Towards the goldilocks zone and away from vicious circles*. Proceedings - 2019 IEEE International Conference on Software Architecture, ICSA 2019, 131-140. https://doi.org/10.1109/ICSA.2019.00022.
27. Mårtensson, T., Ståhl, D., Martini, A., Bosch, J. (2021a). Efficient and effective exploratory testing of large-scale software systems. *Journal of Systems and Software, 174.* https://doi.org/10.1016/j.jss.2020.110890.
28. Mårtensson, T., Ståhl, D., Martini, A., Bosch, J. (2021b). *The MaLET Model - Maturity Levels for Exploratory Testing*. Proceedings - 2021 47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, 78-85. https://doi.org/10.1109/SEAA53835.2021.00019.
29. Martin, A., Raponi, S., Combe, T., Di Pietro, R. (2018). Docker ecosystem – Vulnerability Analysis. *Computer Communications, 122,* 30-43. https://doi.org/10.1016/j.comcom.2018.03.011.
30. Mattos, D.I., Dakkak, A., Bosch, J., Olsson, H.H. (2020). *Experimentation for business-to-business mission-critical systems: A case study*. Proceedings - 2020 IEEE/ACM International Conference on Software and System Processes, ICSSP 2020, 95-104. https://doi.org/10.1145/3379177.3388902.

31. Mijumbi, R., Okumoto, K., Asthana, A., Meekel, J. (2018). *Recent Advances in Software Reliability Assurance*. Proceedings - 29th IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2018, 77-82. https://doi.org/10.1109/ ISSREW.2018.00-27
32. Moral-Muñoz, J.A., Herrera-Viedma, E., Santisteban-Espejo, A., Cobo, M.J. (2020). Software tools for conducting bibliometric analysis in science: An up-to-date review. *Profesional de la Informacion, Vol. 29, Iss. 1*. El Profesional de la Informacion. https://doi.org/10.3145/epi.2020.ene.03.
33. Olsson, H.H., Alahyari, H., Bosch, J. (2012). *Climbing the "Stairway to heaven" - A mulitiple-case study exploring barriers in the transition from agile development towards continuous deployment of software*. Proceedings - 38th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2012, 392-399. https://doi.org/10.1109/SEAA.2012.54.
34. Oztemel, E., Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing, Vol. 31, Iss. 1,* pp. 127-182). Springer. https://doi.org/10.1007/s10845-018-1433-8.
35. *Presidential Policy Directive -- Critical Infrastructure Security and Resilience* (2013). Testimony of Office of the Press Secretary. https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil.
36. Schmiedmayer, P., Chatley, R., Bernius, J.P., Krusche, S., Chaika, K., Krinkin, K., Bruegge, B. (2022). *Global software engineering in a global classroom*, 113-121. https://doi.org/10.1145/3510456.3514163.
37. Ståhl, D., Bosch, J. (2017). Cinders: The continuous integration and delivery architecture framework. *Information and Software Technology, 83,* 76-93. https://doi.org/10.1016/ j.infsof.2016.11.006.
38. Ståhl, D., Mårtensson, T. (2021). *Won't Somebody Please Think of the Tests? A Grounded Theory Approach to Industry Challenges in Continuous Practices*. Proceedings - 2021 47th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2021, 70-77. https://doi.org/10.1109/SEAA53835.2021.00018.
39. Ståhl, D., Mårtensson, T., Bosch, J. (2017). *Continuous practices and Devops: Beyond the buzz, what does it all mean?* Proceedings - 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, 2017-January, 440-448. https://doi.org/10.1109/SEAA.2017.78
40. *The Mobile Economy 2021* (2021). GSMA Intelligence.