Chapter 4
Critical Digital Services

An Under-Studied Safety-Critical Domain

John Allspaw

Abstract While the COVID-19 pandemic has brought new attention to how essential Internet-connected services are to society's functioning, there continues to be a dearth of research about how these critical digital services (CDSs) are operated, maintained, and delivered from a cognitive work, human factors, and safety science perspective. Efforts to anticipate what the future of work will look like must consider the challenges and opportunities this now critical domain faces. The conditions are favorable for making progress in studying work in this domain. A small (but growing) community of practitioners in software engineering and operations are enthusiastic about exploring and understanding the *cognitive work* they engage in every day. There is also (at least, currently) a relative *absence* of regulatory or procedural barriers that would otherwise hamper productive exploration by researchers.

Keywords Critical digital services · Software engineering · Online operations · Automation

4.1 Introduction

It might appear self-evident, perhaps even banal, to say that *all* modern businesses rely on Internet-connected services. The societal need to adapt to the global COVID-19 pandemic has highlighted how much organizations have come to rely on distributed software applications in many facets of everyday life. For example, to follow social distancing policies (which prevented working in offices), many adapted by working from their homes using video conferencing and other collaboration tools (such as messaging and chat services). Many smartphone and tablet users who have viewed these applications as *conveniences* suddenly realized how integral they have become for keeping some of the most fundamental societal functioning alive in many countries worldwide. Services such as:

J. Allspaw (✉)
Adaptive Capacity Labs, New York, USA
e-mail: allspaw@adaptivecapacitylabs.com

- food and medicine delivery apps;
- social media and community organizing/communication services;
- news and broadcasting channels;
- information, data, and guidance published by governments;
- new channels of coordination between research organizations working on vaccines.

all became critical to supporting both local and global communities.

While the COVID-19 pandemic certainly played a critical role in highlighting how essential these services are, the importance of these digital services has been growing for some time.

Many (if not all) high-tempo and high-consequence industries (energy, transportation, medicine, aviation, etc.) have experienced surprising disruptions significant enough to be later seen as *seminal* accidents (Three-Mile Island, Chernobyl, Tenerife airport disaster, etc.). These events presented such a major shock to the prevailing ideas on what constitutes risk and safety that they upended previous beliefs held by those studying or practicing work in those domains. As David Woods has said, these events revealed that "things didn't work the way we thought they did" (personal communication, July 20, 2021).

The industries associated with these now well-known accidents learned *the hard way* that controlling risk of accidents required understanding the "messy details" of real work done by people in their given roles and the organizational framing (management, leadership, etc.) that influences that work.

Critical digital services has yet to experience its "Three-Mile Island" event. Is such an accident necessary for the domain to take human performance seriously? Or can it translate what other domains have learned and make productive use of those lessons to inform how work is done and risk is anticipated for the future? The answers to these questions are currently unclear, but we can start by exploring what opportunities and challenges CDS has that contrast with other domains.

4.2 Growing Criticality

If we look closely at those domains traditionally known as "safety–critical" mentioned above, we will find a great deal of dependence on software services delivered across the Internet.

For example, if electronic medical health record systems are unavailable or experiencing latency, this disruption can significantly impact hospital operations, which then require staff to implement workarounds. Unfortunately, in many cases, these workarounds must be in place without knowing how long they will be necessary.

In the U.S. alone, e-commerce sales grew 20-fold between 2000 and 2019. The estimated value of the U.S. digital economy was USD 1.35 trillion in 2017. Rates of data transfer (in aggregate) grew from 0.1 terabytes per second (TB/s) in 2002 to 45 TB/s in 2017 and are expected to reach 150 TB/s by 2022.

4.3 Growing Consequences

We need not look further than daily news channels for demonstrations of this criticality. The effects of outages, degradations, and other types of disturbances can range from a short-lived inconvenience to unrecoverable, viability-crushing events for companies. These disruptions can also have devastating *cascading* consequences long after responders have restored service. Consider complications that can exist in the aftermath of:

- **airline reservation systems** (effects on crew scheduling, flight logistics, etc.) [11];
- **stock exchanges** (effects on equity pricing, financial solvency, etc.) [10];
- **electronic medical health records** (effects on clinical procedures, bed capacity, etc.) [15];
- **entertainment and sporting events** (effects on fan behaviors, crowd controls, etc.) [13];
- **retail businesses** (effects on regional economics, employment, etc.) [5].

4.4 The Landscape of Roles and Skills

To make informed projections about the challenges that CDSs are likely to face in the future, it is worthwhile to describe how roles and professions have evolved over time.

In the early days of the web, much of the work involved with designing, building, and operating Internet-connected services was typically done by *generalists*, most coming from a programming background. Once the web (and the technologies that powered it) took on new capabilities and the global userbase grew, specializations began to appear. The generalist role of "webmaster" disappeared and new ones emerged that fell into two categories: *application development* (focused on user-facing) functionality and *systems administration* (focused on underlying operating systems and other infrastructure.)

In the early 2000s, Web sites took on more interactive functionality. Advances in network connectivity, web-specific programming languages, new standards, and other new technologies fueled the development and growth of even more complicated uses. Web *sites* became web *applications* and the use of databases to store and retrieve data exploded. Database administration became a specialized role, and as e-commerce businesses grew in use and scale, so did the need for network and security engineering roles.

Specialist roles and skills continue to emerge, but they continue to sit roughly in two main areas. "Front-end" engineers focus largely on software related to the functionality, design, and representation of user-facing interfaces. "Back-end" engineers focus on the technical infrastructure that supports and enables the front-end

applications to work, such as operating systems, languages, databases, and other architectural components that are typically invisible to users of the service.

These specialized skills and roles evolve and adapt very effectively—practitioner communities grow and organize in a bottoms-up and organic fashion (via collaborating on open-source code, conferences, etc.), rather than via centralized professional organizations like those found in aviation or medicine.

While there has been some effort to provide formal training, accreditation, and licensure of these skills [12], they have not been successful. There has been much debate about the need for formalized training and licensure for these roles. The most significant barriers are seen as the relative immaturity of the field (despite its growing criticality) and what assurances licensing could practically give in such a fast-moving field of practice.

A notable aspect in CDS worlds that contrast with others is that quite often, the *designer* of technology is also the *operator* or *user* of the technology. This is in stark contrast to what is found in other domains. Take, for example, medical monitoring devices found in clinical settings. These are not designed (or even modified in ways beyond the built-in controls) by the doctors or nurses who use them every day, but by manufacturers at a distance. In CDS environments, engineers often create new software tools for their own use or modify the existing tools they use. Engineers who find a need for new functionality in the tools they use will simply write that new functionality into the tool, sometimes on the spot.

Why and how does this matter? It means the feedback between user and designer in the same individual is quite fluid (and largely tacit) and interfaces can evolve at a rapid pace as a result. This tool-creating and modifying capacity in the CDS domain is rather unique and represents a great advantage over work done in environments where physical limitations and laws (gravity, thermodynamics, chemical reactions, etc.) govern what can and cannot be modified.

4.5   What Does the Future of Work in CDS Look Like?

Lisanne Bainbridge's seminal paper, *Ironies of Automation* [4], is almost 40 years old and is perhaps more relevant today than it was when it was published. In the 2010s, specialized communities of practice in the tech industry began to emerge, such as "DevOps[1]" and "SRE.[2]" New topics which described the lived experience these practitioners encountered in their work started to appear in conference talks, blog posts, and mailing list discussions. These included subjects such as:

- difficulties with designing and handling alarms or alerts to reduce false positives;
- exploration of novel on-call rotation structures and schedules;

---

[1] Approach aiming at conciliating software development (Dev) and operations (Ops).

[2] Site Reliability Engineer.

- new generations of diagnostic tools to help engineers understand their application and system behaviors;
- approaches to onboard new and early career engineers into such complex environments;
- indicators of occupational burnout and activities that may mitigate it;
- (and many others).

Rather than focus discussions on computer science or programming concerns, it was clear these engineers became enthusiastic and curious about a myriad of topics that professionals from the world of safety would recognize as *human factors* [2].

Fast forwarding to 2021, the SRE and DevOps communities are now established as distinct fields of practice. In the past few years, there has been a faint trickle of research interest on the work these engineers do [1, 7, 9], but there is still much that is unclear.

4.6 What Can Industry Adaptation to COVID-19 Tell Us?

Organizations responsible for producing and operating CDSs needed to adapt quickly in the beginning of 2020 as the pandemic grew worldwide. From a business viability perspective, some online businesses faltered (such as travel-related services) while others saw unprecedented growth (telemedicine, for example).

By and large, the tech industry managed to adapt relatively well to the pandemic relative to other industries. Far and away, the greatest demonstration of this adaptation was adjusting the workforce to work from home and other alternate locations. As mentioned, Internet-capable communication and collaboration tools proved to be integral on this front. Prior to the pandemic, a few notable tech companies differentiated themselves as employers by declaring to be a "remote-friendly" business, with staff distributed across the globe [14]. While many other companies employed a small percentage of staff who worked outside of corporate offices, changing routines and practices that did *not* rely on in-person interactions was difficult.

It became quite clear there was a difference between "working from home" and working from home in a pandemic that required "lockdowns." For those with children who could not attend school or who were primary caregivers for those needing assistance, adjustments were not simply geographical in nature. Managing schedules and calendars became a much more difficult task, in addition to the fatigue generated by prolonged video conferencing.

Despite these multiple challenges, early analysis reveals increases in productivity, especially for those in scientific and technical roles [6].

4.7 A Critical yet Nascent Domain

There are several features of the critical digital services domain worth highlighting, in contrast to others. These qualities represent both challenges and opportunities to researchers.

4.7.1 Challenges

The first challenge is the relative age of the domain. While the invention of technologies and frameworks we now know as the World Wide Web took place in 1989, mainstream adoption began to accelerate several years later. Contrasting this history with transportation industries such as rail and aviation, it is much younger. Many studies taking place currently are novel, rather than building on a corpus of research spanning decades.

Another challenge is the somewhat insular nature of activity among practitioner communities. Despite much research attention brought to more theoretical topics surrounding computer science (such as machine learning and human–computer interaction), the work of software service *operations* tends to be viewed outside and quite different. This may bring groups to be reluctant to participate in research projects.

Modern software—especially applications that are deployed and operated in "open" environments such as the Internet—*cannot* be made to be free of bugs. The variety of usage and complexity of its mechanisms are simply too great to model sufficiently enough to provide assurance that accidents and outages would not happen. To a working engineer in the CDS domain, this fact is assumed in such a fundamental "sky is blue, grass is green" way that it is usually unspoken. Fundamental surprises [8] are common. The challenge for CDS is not that this is the case; it is that outsiders of the community are unaware of how uncertain software's behavior is and how well it can be tested prior to being deployed for use.

Perhaps the most significant challenge is how fast the criticality—and the expertise required to operate the technology to support it—is growing. Researchers looking to characterize and study this work will need to find ways to collect, collate, and synthesize the data they need more efficiently. This world being studied will not wait on twentieth-century research timeframes.

4.7.2 Opportunities and Advantages for Researchers

From a *cognitive work* research perspective, the tech industry's adaptation to work in a more distributed fashion during the pandemic did bring a potential opportunity that was not present prior to the pandemic: an unprecedented abundance of data that

might be available for analysis. Since teams working remotely with each other is mediated by software, data that researchers would want or need may be available.

Even the most basic features of current video conferencing, chat, and other collaboration tools include recording and/or logging actions taken, and utterances made by participants, at millisecond granularity. Collecting these *externalizations* for analysis historically was not possible without expensive audio and video recording equipment, and if they were not set up or recording a given exchange, the data were simply missed. The tools used for transcription and analysis have also dramatically improved the data researchers have available to them.

Another significant advantage in this domain is the flip side of the challenge mentioned above: its relative short history. How is this domain like others, and in what ways? How is it different from other domains, and in what ways? Recent studies reveal that the expertise necessary to successfully navigate problem-solving maps to results found in other domains, but it seems quite clear that the landscape is an "open field" at this time, when it comes to understanding the multiple interleaved goals and concerns involved with coping with complexity in this domain.

This field (like many others written about in this volume) can be seen from a dual perspective [3]. An inspiring one since people have a highly refined expertise and novel mechanisms allow to bring that expertise to bear. A worrisome one because the configuration of technology and organization is often struggling to make this expertise effective.

Looking forward, it would behoove researchers to explore this domain in greater detail. As software services continue to become more critical to society's functioning, the future depends on it.

References

1. J. Allspaw, *Trade-Offs Under Pressure: Heuristics and Observations of Teams Resolving Internet Service Outages.* M.Sc. thesis, Lund Unviversity, Sweden, 2015. Retrieved from https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8084520&fileOId=8084521
2. J. Allspaw, Human factors and ergonomics practice in web engineering and operations: navigating a critical yet opaque sea of automation, in *Human Factors and Ergonomics in Practice* (2016), pp. 313–322
3. J. Allspaw, R.I. Cook, SRE cognitive work, in *Seeking SRE: Conversations About Running Production Systems at Scale*, ed. by D. Blank-Edelman (O'Reilly Media, 2018), pp. 441–465
4. L. Bainbridge, Ironies of automation. Automatica **19**, 775–779 (1983)
5. C. Connley, *Target Says Cash Registers Back Online and Customers Can Make Purchases Again After Systems Outage* (CNBC, 2019). Retrieved 26 May 2021, from https://www.cnbc.com/2019/06/15/targets-in-store-payment-is-system-down-impacting-stores-nationwide.html
6. E. Curran, *Work from Home to Lift Productivity by 5% in Post-pandemic U.S.* (Bloomberg, 2021). Retrieved 15 June 2021, from https://www.bloomberg.com/news/articles/2021-04-22/yes-working-from-home-makes-you-more-productive-study-finds
7. M.R. Grayson, approaching overload: diagnosis and response to Anomalies, in *Complex and Automated Production Software Systems*, ed. by D.D. Woods (The Ohio State University, 2018)
8. Z. Lanir, *Fundamental Surprise* (Decision Research, Eugene, OR, 1986)

9. L.M. Maguire, *Controlling the Costs of Coordination in Large-Scale Distributed Software Systems.* Doctoral dissertation, The Ohio State University, 2020
10. J. McCrank, *NYSE Shut Down for Nearly Four Hours by Technical Glitch* (Thomson Reuters, 2015). Retrieved from www.reuters.com/article/us-nyse-trading-idUSKCN0PI25A20150709
11. J. Mullen, *British Airways Computer Glitch Causes Big Delays at Multiple Airports* (CNNMoney, 2016). Retrieved from https://money.cnn.com/2016/09/05/news/companies/british-airways-computer-system-delays/
12. NCEES, *NCEES Discontinuing PE Software Engineering Exam* (2018). Retrieved 11 Nov 2021, from https://ncees.org/ncees-discontinuing-pe-software-engineering-exam/
13. J. Roberts, Hulu's world series stream crashed in the middle of Game 4. Fortune (2017). Retrieved 26 May 2021, from https://fortune.com/2017/10/29/world-series-hulu-crash-problems/
14. K. Schwab, More people are working remotely, and it's transforming office design. Fastcompany (2019). Retrieved 15 June 2021, from https://www.fastcompany.com/90368542/more-people-are-working-remotely-and-its-transforming-office-design
15. WKYT News Staff, *Software Issue Fixed, UK Healthcare No Longer Diverting Patients* (WKYT, 2019). Retrieved 26 May 2021, from https://www.wkyt.com/content/news/UK-HealthCare-diverting-patients-to-other-hospitals-citing-computer-issues-561140481.html