Why we didn't conduct studies in socio-technical resilience during the pandemic

Helen Sharp[1], Tamara Lopez[1], Michel Wermelinger[1], Mark Levine[2], Caroline Jay[3], Bashar Nuseibeh[1]
[1]The Open University UK; [2]Lancaster University UK; [3]Manchester University UK
{firstname.lastname}@open | lancaster | manchester.ac.uk;

Abstract. Conducting *in situ* studies to investigate socio-technical resilience in software development during the pandemic runs the risk of producing distorted results. To overcome this, we have changed the focus of our work in the short term, to create an analytical STR framework to identify salient factors, based on two sources. First, through an industrial collaboration investigating safety-critical collaborative work, and second through existing data sets of day-to-day software development activities. This paper outlines our rationale.

Introduction

The COVID-19 pandemic impacted working environments significantly and suddenly across the globe. Impacts on software development include gender inequality (Machado et al, 2020), productivity (Bezzera et al 2020), wellbeing (Ralph et al 2020) and communication (Oz et al, 2020). Our research focuses on socio-technical resilience (STR). It may seem that a significant pandemic would provide an ideal setting for our work, but major disruptions are not the most interesting aspect of resilience. It's the day-to-day activities and adjustments that reveal the secrets of how a system keeps running (Rasmussen, 1990).

As with other researchers the pandemic disrupted our plans for *in situ* developer studies. Initially we were resigned to postponing our work, or conducting it online, but as the pandemic continued, a different set of questions arose. In a crisis situation it's not clear which of the studied behaviours and situations will persist beyond the crisis. Studying software development at this time must address key questions:

How much of the data collected will still be relevant in a year's time or longer term? What might software development work look like post-pandemic? How can we pursue our research goals in STR before the pandemic has passed?

Socio-technical resilience in software development

Resilience can be defined in many ways. Although technical resilience is often characterised in terms of preventing failure, in our work resilience is used to refer to the intrinsic abilities a socio-technical system has to function in the face of change (Hollnagel, 2012). Resilience is thus a product of top-down efforts to ensure that systems work as planned (through procedures, policies etc.), and the bottom-up efforts of workers at the "sharp end" to keep systems working (Woods 2006). Resilience engineering provides a framework for understanding how people learn about, monitor and respond to situations that are both anticipated and unexpected. Studying resilience in software development from this perspective therefore requires *in situ* studies to gain an understanding of everyday activities at the micro level as well as procedures and policies at the macro level.

Will results obtained during the pandemic be relevant post-pandemic?

National lockdowns changed the way people work, shop, interact and socialise. Everyone spent more time at home and online, interacting through video conferencing systems. This applied to software developers too, but developers already spend a lot of their working lives online so what actually changed for them?

To help investigate this question we deployed a small-scale diary study for one section of our developer community. This ran over eight weeks (April to June 2020), with 17 participants. Each week focused on a different set of questions to maintain engagement, with prompts that asked how the pandemic was affecting working conditions (Lopez et al, 2021). Participants reported difficulties with working at home such as lack of space and increased childcare responsibilities, but most found the change straightforward in terms of the technical elements of their job. All agreed that communication changed, but not consistently for better or worse; both improved productivity and potential burnout were reported; concern about job security was common; and organizational changes affected well-being, both negatively through poorer social interaction and increased fatigue but also positively through improved flexibility and inclusivity.

This small study echoes others' findings in terms of communication (Oz et al, 2020), increased childcare responsibilities and interruptions (Machado et al, 2020), and also positive aspects of the changes (Butler and Jaffe, 2020). The main issues reportedly faced by developers are primarily not technical but social and organizational. This suggests that *in situ* studies focusing on STR will be overshadowed by specific contextual factors of the pandemic, like stress and anxiety caused by sudden home-working, by health issues and job security.

While these factors will have impact into the future, those impacts can be planned rather than suddenly imposed, and the specific context of the pandemic will not persist: organizational changes will evolve and stabilise, schools will re-open, and vaccines will be deployed. This led us to question whether data collected during the pandemic will be relevant afterwards.

What might software development look like post-pandemic?

Some aspects of day-to-day software development that were common pre-pandemic will persist post-pandemic. For example, code will be developed individually and through collaboration and community exchanges, development tools will evolve, and there will be interactions between people, tools and software. These aspects (and probably others) are relevant to our STR focus.

Other aspects of software development may differ, and several commentators have speculated about work after the pandemic. For example, McKinsey (2021) report on the long-term impact of the pandemic across several different work arenas and in eight economies including China, UK, France, Germany, Spain and the US. Their key findings emphasise the dimension of physical proximity, hybrid remote working (combining physical co-location and homeworking), the growth of digital technologies, importance of AI and automation, emphasis on skills and experience rather than academic degrees, and the likelihood of re-deployment from low-wage job categories. Madhavan (2021) also predicts an increase in hybrid working and digitization, and emphasises that a collaborative rather than an authoritarian style of management works better when staff are working at a distance.

Although the future is uncertain, aspects of pre-pandemic software development work will remain pertinent, and will continue to evolve, e.g. through different emphases on physical proximity, digitization, collaboration and automation.

How then to proceed?

Rather than perform *in situ* studies with developers to identify salient factors of STR in particular contexts, we are developing an analytical STR framework for identifying these factors. Two activities are underway. First, we have established an industrial collaboration with a partner in a safety-critical domain that is currently examining resilience within its socio-technical systems. We are using this connection to develop our theoretical understanding of STR. Second, to extract authentic scenarios from data collected in prior developer studies using distributed cognition. These data sets include individual and team working in co-located (Zaina et al, 2020), dispersed (Sharp et al, 2012) and hybrid contexts (Deshpande et al, 2016), increased digitization of physical story cards (Sharp et al, 2009) and collaboration (Sharp & Robinson, 2008).

We will apply this STR framework later to examine everyday practice post-pandemic. This, however, raises another question: How to tell when a "new normal" has stabilised sufficiently for *in situ* studies to continue?

References

Bao, L., Li, T., Xia, X., Zhu, K., Li, H., & Yang, X. (2020). How does Working from Home Affect Developer Productivity?--A Case Study of Baidu During COVID-19 Pandemic. *arXiv preprint arXiv:2005.13167*.

Bezerra, C. I., de Souza Filho, J. C., Coutinho, E. F., Gama, A., Ferreira, A. L., de Andrade, G. L., & Feitosa, C. E. (2020). How Human and Organizational Factors Influence Software Teams Productivity in COVID-19 Pandemic: A Brazilian Survey. In Proc BSSE (pp. 606-615).

Butler, J. and Jaffe, S. (2020) Challenges and Gratitude: A Diary Study of Software Engineers Working From Home During Covid-19 Pandemic, available from https://www.microsoft.com/en-us/research/uploads/prod/2020/07/NFW-Butler-Jaffe.pdf

Deshpande, A., Sharp, H., Barroca, L. & Gregory, A.J. (2016) 'Remote Working and Collaboration in Agile Teams', in *Proceedings of ICIS*, Dublin, pp 4543-4559

Hollnagel, E. (2012) Prologue *Resilience Engineering in Practice: A Guidebook*, 9-27 Ashgate Publishing, Ltd.

Lopez, T., Jay, C., Wermelinger, M. & Sharp H. (2021) How has COVID-19 pandemic affected working conditions for research software engineers?, doi:10.48420/14330807, from https://figshare.manchester.ac.uk/account/articles/14330807.

Machado, L. S., Caldeira, C., Perin, M., & de Souza, C. R. (2020). Gendered experiences of software engineers during the COVID-19 crisis. *IEEE Software*.

Madhavan, Naveen (2021) How COVID-19 changed the future, in *ITNow*, Spring 2021, 64-65.

McKinsey (2021) *The Future of Work after COVID-19 Exec Summary*, Feb 2021, from https:// www.mckinsey.com/featured-insights/future-of-work/the-future-of-work-after-covid-19

Oz, T., & Crooks, A. (2020). Exploring the Impact of Mandatory Remote Work during the COVID-19 Pandemic. https://osf.io/preprints/socarxiv/hjre6/

Ralph, P., Baltes, S., Adisaputri, G., Torkar, R., Kovalenko, V., Kalinowski, M., ... & Zhou, M. (2020). Pandemic Programming: How COVID-19 affects software developers and how their organizations can help. arXiv preprint arXiv:2005.01127.

Rasmussen, J. (1990) Human error and the problem of causality in analysis of accidents. *Phil. Trans. of the Royal Society of London. B, Biological Sciences*, *327*(1241), 449-462

Sharp, H., Giuffrida, R. & Melnik, G. (2012) 'Information flow in a dispersed agile team: a distributed cognition perspective', in *Proceedings of XP 2012*, Malmo, Sweden, May

Sharp, H., Robinson, H.M. & Petre, M. (2009) 'The Role of Physical Artefacts in Agile Software Development: two complementary perspectives' *Interacting with Computers* **21**(1-2) 108-116

Sharp, H. & Robinson, H. (2008) 'Collaboration and Co-ordination in mature eXtreme Programming teams' *International Journal of Human-Computer Studies,* **66,** 506-518

Zaina, L.A.M., Sharp, H. & Barroca, L. (2021) 'UX information in the daily work of an agile team: A distributed cognition analysis', *IJHCS*, 147 (March), 102574