Important remarks:

- These example research proposal are in their final form, the draft proposal (first step) is shorter.
- The exact format may vary in the current course. Refer to the kick-off slides of the current course.
- Since the Sample 1 is a specific example of Topic 1, do NOT use the same or a related research question for your own research proposal if you target Topic 1.

Table of Contents



## 1. Sample Proposal 1

# Productivity and remote work: Can improved team cohesion through scheduled coffee breaks help productivity?

An in-situ study on a one-to-one coffee-break scheduling bot.

[redacted]
University of Zurich
Zürich, Switzerland

[redacted]
University of Zurich
Zürich, Switzerland

## 1 INTRODUCTION

Due to the spread of COVID-19 globally, many countries instructed companies to have their employees work remotely. Many teams who had previously worked face-to-face were forced to adjust to collaborating and communicating remotely, introducing several new challenges for software developers. Researchers have already begun studying the phenomena and its effects on workers in different aspects such as well-being, eating behaviour or physical activity [1, 8, 46, 65]. Furthermore, multiple employees have noted that their productivity was affected by the shift of the workplace as well. To analyze this, Bao et al. performed a case study, where they concluded that productivity was affected in positive and negative ways [4], a finding confirmed by Russo et al. [46] who analyzed developer well-being and productivity. Ralph et al. [44] concluded with the help of a questionnaire survey that well-being and productivity are not only correlated, but also currently being negatively impacted by the COVID-19 crisis. A decrease in productivity can generally be caused by multiple factors, such as: *bigger need for self discipline; decreased collaboration with others, more loneliness, higher burnout, more communication friction and more scheduled meetings.* [4, 50].

Whereas the previously mentioned papers look at the impact and productivity on the individual level, Miller et al. [41] analyse how the pandemic has negatively impacted team productivity, by discussing how the ability to reach milestones, team culture and communication have changed. Productivity and team factors such as interconnectedness, team culture, team cohesion and team identity have shown themselves to be strongly correlated [60, 67]. The term **team cohesion** describes the measure of attraction of the group to its members. Highly cohesive teams are more cooperative, effective and productive in achieving goals [48]. The remote nature of WFH settings and the asynchronous collaboration sometimes introduced by it, has had considerable effects on team related factors. Considering that developers spend 45% of their work time collaborating [22], the decreasing trend in team factors and its effects are problems which we need to design solutions for.

Our research builds up on previous papers and studies on how technology can improve productivity through supporting the aforementioned negatively affected team factors, especially team cohesion. Remote work in the software development field will persist, even when companies move back on-site. The trend to work from home (WFH) has actually accelerated through the pandemic [38]. As Miller et al. [41] have previously noted in their study, regular WFH is not the same as pandemic forced WFH, however the situation still provides us with a natural experiment for many researchers to study.

We hypothesize that productivity can be improved through better team cohesion which can be reached with one-to-one coffee-breaks. Previous scholars have shown that team cohesion can be improved through social activities such as team building sessions, physical activity, social activities or team events [18, 51]. We distinguish between scheduled and spontaneous meetings due to the fact that in previous papers [4, 41, 50] it was concluded that many workers not only dislike the high amount of scheduled meetings but that it contributed negatively to productivity, and thus we hypothesize that an spontaneous meeting will feel more natural and less formal for participants. We pose the following research questions (RQ):

- RQ1: Can one-to-one coffee-breaks improve team cohesion?
- RQ2: Can the perceived productivity be improved through better team association caused by one-to-one coffee-breaks?
- RQ3: Do spontaneous coffee-breaks have a bigger impact as opposed to scheduled ones?

To answer our research questions, we provide an extensive literature overview on productivity, WFH, team cohesion and their relation. This is followed by an in-situ case study, where we schedule one-to-one coffee-breaks with the help of a bot. We gather qualitative data with the help of interviews and surveys which will be performed at the end of the day, end of the week and post study.

## 2 RELATED WORK

Related work can broadly be classified into four categories: developer productivity, remote work and productivity, team factors and social breaks.

### 2.1 Developer Productivity

There has been a steady incline in research around the topic of productivity in the software development field, as it may lead to faster development speed and also higher developer satisfaction [20, 30]. One fundamental problem which keeps arising in studying developer productivity is defining and factors that encapsulate ***productivity*** [59]. There are several studies which explore how to quantify productivity [17, 62, 68]. Some example of such quantification are:

- number of tasks per month [68]
- number of source line codes per hour [17]
- ratio of written code lines and spent effort [25]
- days taken to respond and solve a modification request [14]

Inspired by Meyer et al. [39], this paper adopts their approach of analysing perceived productivity, which encapsulates the developers themselves think about productive versus non-productive work. Through observation and a wide-scale survey, they managed to show how developers assess their own productivity and which activities they categorize as highly productive, such as coding, and less productive, such as meetings. There has also been work on the predictors of productivity such as years of experience [55] or relevant skill-sets [35] Another aspects which has gained significant traction is work fragmentation. Not only is developer work highly fragmented and divide in short sessions, but is also highly interrupted due to meetings, blocking tasks and unexpected requests from co-workers [39]. The impact of these factors on work performance has been extensively studied [16, 27, 43, 56]. While sometimes interruptions are necessary, e.g. to switch tasks in order to unblock a co-worker, they can lead to lower performance [2].

There has been an abundance of work addressing the problem of individual productivity through software tools and methods [32, 40, 49, 54, 66, 69]. For instance, Young-Ho et al. came up with TimeAware [32], a tool to increase personal productivity through positive and negative framing of daily work activities. They concluded that the negative framing had no impact, but positive framing had a positive impact. Züger et al. address productivity by minimizing flow interruption occurring in the office with the help of colorful lights, indicating availability or a state of flow [69]. Tseng and colleagues try to address the problem from a different angle. Through creating a healthy balance between work and breaks, they try to minimize cyberloafing and thus improve performance [54].

## 2.2 Work From Home

Research on WFH has increased in the recent years [5, 19, 29, 36], due to its many benefits for employees as well as for the employers. It allows employees to eliminate their commute and save money for food or transportation. At the same time, it allows companies to hire the best talent from any location. Despite the many advantages, there are several challenges have arisen. Workers may find themselves working increased hours [19] and thus increase the risk of burnout or developer disengagement [37]. Teamwork can be significantly impacted by remote work [9, 24, 61]. Wagstrom et al. found the temporal distribution of teams had a significant negative impact on communication response time [61]. Furthermore, Butler et al. further studied the challenges faced by developers during the pandemic and their correlation to job satisfaction. They found out that two of the biggest challenges were an overload of meetings and feeling overworked [11].

In addition, WFH has been found to have an impact on individual software development productivity. The majority of work the authors could find focused on improving productivity through tools of individuals one-site, and only a few qualitatively or quantitatively analysing productivity in remote settings [3, 13, 33]. To the authors' best knowledge, there have only been a few studies which specifically address developer productivity in remote settings [4, 20, 44]. Raph et al. [44] conducted an online questionnaire with over 2,000 responses from developers around the world. They highlight how well-being and productivity are very correlated, and currently decreasing, they set up a table of support actions employers can reference to improve worker well-being such as the organization paying for equipment to work from home, reassurance that they understand why their performance is currently lacking or reassurance that they will keep their job. Through a quantitative analysis by Bao and colleagues, they highlighted how some developers actually experienced an incline in productivity, but they also concluded that those developers who did experience a decrease in productivity, named the following reasons: more home demands; a need for self discipline; and decreased collaboration with others [4]. Smite et al. conducted a case study with a large international company with offices in Sweden, USA and the UK. They specifically analyzed how WFH has impacted developer productivity, satisfaction and collaboration. They report benefits and challenges and highlight problems such as more loneliness, higher burnout, more communication friction and more scheduled meetings. They also suggest different measures to improve work culture but also to balance individual and team productivity [50].Overall it seems like WFH can have a positive and negative influence, but there are certain factors that can steer it either way.

## 2.3 Team Cohesion on Productivity

Through systematic review, Wagner and Ruhe managed to show which main factors predict developer productivity and categorized them into technical, such as product complexity, and soft, such as work environment. They concluded that camaraderie, team identity and cohesion are among the main factors contributing to productivity [60]. These factors can increase individual productivity and job satisfaction as well as the team performance [57]. Additionally, Goncalves et al. found out that developers spend 45% of their work time collaborating, making those factors even more relevant to be studied [22]. Johnson et al. add to this arguing software developers profit less from WFH than other office jobs do, due to the importance of collaboration and communication in this field [28]. In a survey conducted by Ford et al. [20], participants which reported a decrease in *"Communication ease with colleagues"*, *"Effectiveness of communication with colleagues"*, *"Quality of scheduled meetings"*, *"Positive interactions with their team"*, and *"Knowledge flow within their team"* were more likely to report lower perceived)productivity, team productivity and work satisfaction.

Yang et al. show how individuals and teams in companies experienced a decrease in interconnectedness during WFH and also reported a lessened strength of ties between colleagues. They also conclude that work performance is correlated to the strength of the ties and thus expect a decrease in productivity [67]. This confirms the study conducted by Waber et al. in 2010 who measured individual, quantitative productivity of employees with and without collective breaks in a US call center. They concluded that team cohesion is closely and positively related to individual productivity of the employees [58].

On a broader level, not only individual productivity has been analysed in the remote setting, but also team productivity. As mentioned above, Miller et al. [41] took a closer look at the software team's culture and productivity during the pandemic. They realised that important team factors, such as social connection and communication have suffered. These factors have been proven to positively influence team productivity according to Bhardwaj and Rana [7]. Bezerra et al. [6] add that collaboration is key to improve team communication and thereby also the teams' productivity.

Team cohesion appears to be more important than ever in these times, however there is a vast amount of confusing and contradicting literature discussing how to conceptualize and measure team cohesion. According to Salas et al. who conducted an exhaustive literature research on measuring team cohesion with a focus on its relevance for team performance, we seem to get the best results by acknowledging that team cohesion is multidimensional, of which the task and social dimension gives most clarity. Although it is also multilevel, focusing on team level should provide enough insight into the team [47]. They suggest several different methods to put this into practice: using data analysis tools to scan different methods of communication (MsTeams, Slack, etc) [10, 21, 26], using some kind of sociometric or physiological measurements to analyze body language and interaction behavior [21, 42], or having an external observer to estimate a team's cohesion [15], aside from the more traditional metric systems like interviews and surveys. Additionally, Salas et al. point out, that it is best to measure team cohesion over time.

## 2.4 Social Breaks

There are many positive aspects of taking a break during a work day, whether they are social breaks, individual breaks, whether it's a break for some physical activity or a sedentary break [28, 34, 52, 54, 58, 63]. Cambo et al. designed BreakSense, a tool to encourage mobility during breaks [12]. Lebbon and Hurley concluded that up to 15% of leisure time during work hours has a beneficial impact on productivity [34]. Wegener et al. went as far as to label breaks at work as a dynamic boundary between work an leisure time, being neither fully leisure and fully work [64], since they are both a place for recreation and productivity.

Breaks can have a direct and an indirect link to an individual's productivity, on one hand it's a space where ideas can be formed informally and naturally [52] and on the other hand it's a big factor contributing to social and personal well-being. Johnson et al. investigated the effects of the workplace on productivity and found that when software engineers take regular breaks, it increases their motivation, productivity and morale [28] and is often mentioned as a productivity strategy. Adding to that, coffee breaks are a vital part of a workday, allowing employees to talk about their work demands and it's an opportunity for supporting each other and forming social bonds [52, 53]. Microsoft also highlighted in their study how these social breaks or informal and spontaneous meetings spark creativity ideas and foster productive collaboration. However such meetings can be harder to achieve remotely [53]. They also reported that many missed the lack of the '*hallway chats*' and '*dropping by someone's desk*' during remote work. Studies report that workers are struggling to find alternatives to the casual, spontaneous *"water-cooler"-conversations* with other co-workers. An interesting note to add here was that this so called *"watercooler"-conversations* allowed employees to spontaneously and coincidentally meet and start conversations with other co-workers with whom they usually would not plan to meet with or who happen to be in a completely different department[11, 53]. In a diary study, 67% of workers reported that their need for spontaneous interaction was not being met during remote work [45]. Miller et al. [41] show that this lack of informal communication and the resulting decreasing team cohesion was seen as negatively impacting productivity.

Waber et al. have previously found that giving employees joint breaks versus individual breaks increases their team cohesion and thus improves their individual, quantitative productivity. [58]. Stroebaek concludes that social breaks are so vital to individual and team well-being and team performance and that they should be a part of the daily routine in a work day.

There have been some discussions in the WFH field whether scheduled social breaks solely increase stress due to the higher workload due to the many scheduled meetings [53]. Research shows that software developers are increasingly under pressure due to higher amount of scheduled meetings [4, 41, 50]. In one survey of over a hundred information workers across a variety of industries, 57% said their meeting load had increased [53]. Adding to that, the challenge of recognizing the right time for a spontaneous break has been proven to be difficult as previous studies have shown [31, 32, 54].

Thus we would like to adopt an idea previously used by Züger et al. [69]. With the help of the status indication on their respective work channel (e.g. Slack, Teams), we want to detect availability of a worker in order to match them up for a coffee-break with another available worker. In this way, workers can be randomly scheduled with other workers of the company, even workers who are outside of their department. We hypothesize that this method will allow a more accurate and natural re-creating of on-side coffee breaks.

# 3 METHOD

To answer the aforementioned research questions, a qualitative in-situ study with participants from a company still working remotely will be conducted. Our coffee-bot is designed to automatically schedule one-to-one coffee-breaks between two team members and offer them an optional topic or a fun question (e.g. "What is your favourite music artist?") to talk about in order to *"break the ice"*. As previously highlighted, employees face a high number of scheduled meetings during remote work. In order to analyze if spontaneous meetings have any kind of influence, we will split the study into two conditions: spontaneous and scheduled coffee-breaks. Thus, the participants will be split in two same sized groups A and B. To be able to detect the influence of the one-to-one coffee breaks, both groups will be subjected to a baseline week. In this week, our coffee-bot will not be in use, but we will still perform our planned survey at the end of each day to gather information on their perceived productivity and team cohesion. This will be done with the help of likert scale and open ended questions. Over the period of the following four weeks, participants will be experiencing our coffee-bot. Team A will have two fixed scheduled meetings during the week (e.g. Tuesday 11o'clock and Friday 14o'clock). Team B, will have spontaneous coffee-breaks whenever the coffee-bot detects two workers being free based on their status on Teams/Slack. The coffee-bot will not impose a time-frame on these meetings, but rather allow the participants themselves to decide its length. Participants will be prompted to fill out the same survey as previously at the end of the day and at the end of the week. Additionally, there will be a post-study interview with the participants once the four weeks are up to gather more in-depth insights and understand the role of coffee-breaks and its effect on team cohesion and productivity better. In contrast to perceived productivity which is an individual-level construct, we measure team cohesion on a team-level construct as suggested by Salas et al. [47]. We will introduce a team supervisor, in this case either the scrum master or project/team leader, depending on the closeness to the team members. The supervisor will answer the same questions before and after the study, based on the three-item team cohesion scale constructed by Harrison et al. which also uses the likert scale [23] similarly as Chang et al. [15] with a focus on the task and social dimension to provide insight whether the team's cohesion changed over time.

## REFERENCES


[1] Ammar, A., Brach, M., Trabelsi, K., Chtourou, H., Boukhris, O., Masmoudi, L., Bouaziz, B., Bentlage, E., How, D., Ahmed, M., Müller, P., Müller, N., Aloui, A., Hammouda, O., Paineiras-Domingos, L. L., Braakman-Jansen, A., Wrede, C., Bastoni, S., Pernambuco, C. S., Mataruna, L., Taheri, M., Irandoust, K., Khacharem, A., Bragazzi, N. L., Chamari, K., Glenn, J. M., Bott, N. T., Gargouri, F., Chaari, L., Batatia, H., Ali, G. M., Abdelkarim, O., Jarraya, M., El Abed, K., Souissi, N., Van Gemert-Pijnen, L., Riemann, B. L., Riemann, L., Moalla, W., Gómez-Raja, J., Epstein, M., Sanderman, R., Schulz, S. V. W., Jerg, A., Al-Horani, R., Mansi, T., Jmail, M., Barbosa, F., Ferreira-Santos, F., Šimunič, B., Pišot, R., Gaggioli, A., Bailey, S. J., Steinacker, J. M., Driss, T., Hoekelmann, A., and of the ECLB-COVID19 Consortium, O. B. Effects of covid-19 home confinement on eating behaviour and physical activity: Results of the eclb-covid19 international online survey. *Nutrients 12*, 6 (2020).

[2] Bailey, B. P., Konstan, J. A., and Carlis, J. V. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *INTERACT* (2001).

[3] Baker, E., Avery, G. C., and Crawford, J. Satisfaction and perceived productivity when professionals work from home.

[4] Bao, L., Li, T., Xia, X., Zhu, K., Li, H., and Yang, X. How does working from home affect developer productivity? – a case study of baidu during covid-19 pandemic, 2021.

[5] Baruch, Y. Teleworking: benefits and pitfalls as perceived by professionals and managers. *New Technology Work and Employment 15* (2000), 34–49.

[6] Bezerra, C., Coutinho, E., and Araujo, A. How do brazilian software development teams deal with working from home after a year of the covid-19 pandemic? In *Brazilian Symposium on Software Engineering* (New York, NY, USA, 2021), SBES '21, Association for Computing Machinery, p. 368–377.

[7] Bhardwaj, M., and Rana, A. Key software metrics and its impact on each other for software development projects. *SIGSOFT Softw. Eng. Notes 41*, 1 (feb 2016), 1–4.

[8] Bhattarai, M. *Working from Home and Job Satisfaction During the Pandemic Times.* 2020.

[9] Bird, C., Nagappan, N., Devanbu, P., Gall, H., and Murphy, B. Does distributed development affect software quality? an empirical case study of windows vista. In *Proceedings of the 31st International Conference on Software Engineering* (USA, 2009), ICSE '09, IEEE Computer Society, p. 518–528.

[10] Brzozowski, M., Sandholm, T., and Hogg, T. Effects of feedback and peer pressure on contributions to enterprise social media. pp. 61–70.

[11] Butler, J., and Jaffe, S. Challenges and gratitude: A diary study of software engineers working from home during covid-19 pandemic. In *ICSE SEIP* (May 2021).

[12] Cambo, S. A., Avrahami, D., and Lee, M. L. Breaksense: Combining physiological and location sensing to promote mobility during work-breaks. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, p. 3595–3607.

[13] Campbell, K. Flexible work schedules, virtual work programs, and employee productivity.

[14] Cataldo, M., Herbsleb, J. D., and Carley, K. M. Socio-technical congruence: A framework for assessing the impact of technical and work dependencies on software development productivity. In *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement* (New York, NY, USA, 2008), ESEM '08, Association for Computing Machinery, p. 2–11.

[15] Chang, S., Jia, L., Takeuchi, R., and Cai, Y. Do high-commitment work systems affect creativity? a multilevel combinational approach to employee creativity. *Journal of Applied Psychology 99*, 4 (2014), 665.

[16] Czerwinski, M., Horvitz, E., and Wilhite, S. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), CHI '04, Association for Computing Machinery, p. 175–182.

[17] Devanbu, P., Karstu, S., Melo, W., and Thomas, W. Analytical and empirical evaluation of software reuse metrics. In *Proceedings of the 18th International Conference on Software Engineering* (USA, 1996), ICSE '96, IEEE Computer Society, p. 189–199.

[18] DiMeglio, K., Padula, C., Piatek, C., Korber, S., Barrett, A., Ducharme, M., Lucas, S., Piermont, N., Joyal, E., DeNicola, V., and Corry, K. Group cohesion and nurse satisfaction examination of a team-building approach. *The Journal of nursing administration 35* (04 2005), 110–20.

[19] Felstead, A., and Henseke, G. Assessing the growth of remote working and its consequences for effort, well-being and work-life balance. *New Technology, Work and Employment 32*, 3 (2017), 195–212.

[20] Ford, D., Storey, M.-A., Zimmermann, T., Bird, C., Jaffe, S., Maddila, C., Butler, J. L., Houck, B., and Nagappan, N. A tale of two cities: Software developers working from home during the covid-19 pandemic, 2021.

[21] Gonzales, A. L., Hancock, J. T., and Pennebaker, J. W. Language style matching as a predictor of social dynamics in small groups. *Communication Research 37*, 1 (2010), 3–19.

[22] Gonçalves, M. K., de Souza, C., and González, V. M. Collaboration, information seeking and communication: An observational study of software developers' work practices. *J. Univers. Comput. Sci. 17* (2011), 1913–1930.

[23] Harrison, D. A., Price, K. H., and Bell, M. P. Beyond relational demography: Time and the effects of surface- and deep-level diversity on work group cohesion. *The Academy of Management Journal 41*, 1 (1998), 96–107.

[24] Herbsleb, J. D., Mockus, A., Finholt, T. A., and Grinter, R. E. An empirical study of global software development: distance and speed. *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001* (2001), 81–90.

[25] Hulkko, H., and Abrahamsson, P. A multiple case study on the impact of pair programming on product quality. In *Proceedings of the 27th International Conference on Software Engineering* (New York, NY, USA, 2005), ICSE '05, Association for Computing Machinery, p. 495–504.

[26] Hung, H., and Gatica-Perez, D. Estimating cohesion in small groups using audio-visual nonverbal behavior. *IEEE Transactions on Multimedia 12*, 6 (2010), 563–575.

[27] Iqbal, S. T., and Horvitz, E. Disruption and recovery of computing tasks: Field study, analysis, and directions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI '07, Association for Computing Machinery, p. 677–686.

[28] Johnson, B., Zimmermann, T., and Bird, C. The effect of work environments on productivity and satisfaction of software engineers. *IEEE Transactions on Software Engineering 477* (03 2019), 736–757.

[29] Johnson, S. K., Bettenhausen, K. L., and Gibbons, E. Realities of working in virtual teams: Affective and attitudinal outcomes of using computer-mediated communication. *Small Group Research 40* (2009), 623 – 649.

[30] Jorgensen, M., and Shepperd, M. A systematic review of software development cost estimation studies. *IEEE Trans. Softw. Eng. 33*, 1 (jan 2007), 33–53.

[31] Kaur, H., Williams, A. C., McDuff, D., Czerwinski, M., Teevan, J., and Iqbal, S. T. Optimizing for happiness and productivity: Modeling opportune moments for transitions and breaks at work. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2020), CHI '20, Association for Computing Machinery, p. 1–15.

[32] Kim, Y.-H., Jeon, J. H., Choe, E. K., Lee, B., Kim, K., and Seo, J. *TimeAware: Leveraging Framing Effects to Enhance Personal Productivity.* Association for Computing Machinery, New York, NY, USA, 2016, p. 272–283.

[33] Laihonen, H., Jääskeläinen, A., Lönnqvist, A., and Ruostela, J. Measuring the productivity impacts of new ways of working. *Journal of Facilities Management 10* (2012), 102–113.

[34] Lebbon, A. R., and Hurley, D. T. The effects of workplace leisure behavior on work-related behavior. *Journal of Behavioral Studies in Business 6* (2013).

[35] Li, P. L., Ko, A. J., and Zhu, J. What makes a great software engineer? *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering 1* (2015), 700–710.

[36] Malhotra, A., and Majchrzak, A. Enhancing performance of geographically distributed teams through targeted use of information and communication technologies. *Human Relations 67* (2014), 389 – 411.

[37] Maslach, C., and Leiter, M. P. Early predictors of job burnout and engagement. *The Journal of applied psychology 93 3* (2008), 498–512.

[38] Meister, J. The impact of the coronavirus on hr and the new normal of work, 2020.

[39] Meyer, A. N., Fritz, T., Murphy, G. C., and Zimmermann, T. Software developers' perceptions of productivity. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (New York, NY, USA, 2014), FSE 2014, Association for Computing Machinery, p. 19–29.

[40] Meyer, A. N., Murphy, G. C., Zimmermann, T., and Fritz, T. Enabling good work habits in software developers through reflective goal-setting. *IEEE Transactions*

*on Software Engineering 47*, 9 (2021), 1872–1885.

[41] Miller, C., Rodeghero, P., Storey, M.-A., Ford, D., and Zimmermann, T. Survey instruments for "how was your weekend?" software development teams working from home during covid-19. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)* (2021), pp. 223–223.

[42] Olguín-Olguín, D., and Pentland, A. Sensor-based organisational design and engineering. *International Journal of Organisational Design and Engineering 1*, 1-2 (2010), 69–97.

[43] Parnin, C., and DeLine, R. Evaluating cues for resuming interrupted programming tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, Association for Computing Machinery, p. 93–102.

[44] Ralph, P., Baltes, S., Adisaputri, G., Torkar, R., Kovalenko, V., Kalinowski, M., Novielli, N., Yoo, S., Devroey, X., Tan, X., Zhou, M., Turhan, B., Hoda, R., Hata, H., Robles, G., Fard, A. M., and Alkadhi, R. Pandemic programming: How covid-19 affects software developers and how their organizations can help. *ArXiv abs/2005.01127* (2020).

[45] Rintel, S., Wong, P., Sarkar, A., and Sellen, A. Methodology and participation for 2020 diary study of microsoft employees experiences in remote meetings during covid-19. Tech. Rep. 2020-10-FOW-SIM1, Microsoft Research, October 2020.

[46] Russo, D.and Hanel, P., Altnickel, S., and et al. Predictors of well-being and productivity among software professionals during the covid-19 pandemic – a longitudinal study. *Empir Software Engh 26*, 62 (2021).

[47] Salas, E., Grossmann, R., Hughes, A. M., and Coultas, C. W. Measuring team cohesion: Observations from the science. *Human Factors 57*, 3 (2015), 365–74.

[48] Severt, J., and Estrada, A. On the function and structure of group cohesion. 3–24.

[49] Sirera, J., and Velloso, E. Designing a tangible device for re-framing unproductivity. In *Human-Computer Interaction – INTERACT 2021* (Cham, 2021), C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, and K. Inkpen, Eds., Springer International Publishing, pp. 525–545.

[50] Smite, D., Moe, N. B., Klotins, E., and Gonzalez-Huerta, J. From forced working-from-home to working-from-anywhere: Two revolutions in telework.

[51] Smith-Ray, R., Mama, S., Reese-Smith, J., Estabrooks, P., and Lee, R. Improving participation rates for women of color in health research: The role of group cohesion. *Prevention Science 13*, 1 (Feb. 2012), 27–35. Copyright: Copyright 2012 Elsevier B.V., All rights reserved.

[52] Stroebaek, P. Let's have a cup of coffee! coffee and coping communities at work. *Symbolic Interaction 36* (2013), 381–397.

[53] Teevan, J., Hecht, B., Jaffe, S., Baym, N., Bergmann, R., Brodsky, M., Buxton, B., Butler, J., Coleman, A., Czerwinski, M., Houck, B., Hudson, G., Iqbal, S., Maddila, C., Nowak, K., Peloquin, E., Fernandez, R. R., Rintel, S., Sellen, A., Smith, T., Storey, M.-A., Suri, S., Wolf, H., and Yang, L. The new future of work: Research from microsoft into the pandemic's impact on work practices. Tech. Rep. MSR-TR-2021-1, Microsoft, January 2021.

[54] Tseng, V. W.-S., Lee, M. L., Denoue, L., and Avrahami, D. *Overcoming Distractions during Transitions from Break to Work Using a Conversational Website-Blocking System.* Association for Computing Machinery, New York, NY, USA, 2019, p. 1–13.

[55] Tubío, Ó. D., Aranda, A. M., Uyaguari, F. U., Turhan, B., Misirli, A. T., Fucci, D., Oivo, M., and Juzgado, N. J. Empirical evaluation of the effects of experience on code quality and programmer productivity: an exploratory study. *Empirical Software Engineering 22* (2016), 2457–2542.

[56] van Solingen, R., Berghout, E., and van Latum, F. Interrupts: Just a minute never is. *IEEE Softw. 15* (1998), 97–103.

[57] Virtaneva, M., Feshchenko, P., Hossain, A., Kariluoto, A., Himmanen, J., Kaitila, P., Kultanen, J., Kemell, K.-K., and Abrahamsson, P. Covid-19 remote work : Body stress, self-efficacy, teamwork, and perceived productivity of knowledge workers. *SCIS 2021 : Proceedings of the 12th Scandinavian Conference on Information Systems 8* (2021).

[58] Waber, B. N., Olguin Olguin, D., Kim, T., and Pentland, A. Productivity through coffee breaks: Changing social networks by changing break structure. *SSRN Electronic Journal 6* (01 2010).

[59] Wagner, S., and Deissenboeck, F. *Defining Productivity in Software Engineering.* Apress, Berkeley, CA, 2019, pp. 29–38.

[60] Wagner, S., and Ruhe, M. A systematic review of productivity factors in software development. *ArXiv abs/1801.06475* (2018).

[61] Wagstrom, P., and Datta, S. Does latitude hurt while longitude kills? geographical and temporal separation in a large scale software development project. In *Proceedings of the 36th International Conference on Software Engineering* (New York, NY, USA, 2014), ICSE 2014, Association for Computing Machinery, p. 199–210.

[62] Walston, C. E., and Felix, C. P. A method of programming measurement and estimation. *IBM Syst. J. 16*, 1 (mar 1977), 54–73.

[63] Wegener, C., Meier, N., and Ingerslev, K. Drinking coffee at the workplace.

[64] Wegener, C., Meier, N., and Ingerslev, K. Drinking coffee at the workplace.

[65] Weitzer, J., Papantoniou, K., and Seidel, S. e. a. Working from home, quality of life, and perceived productivity during the first 50-day covid-19 mitigation measures in austria: a cross-sectional study. *Int Arch Occup Environ Health 94* (2021).

[66] Wu, D., and Tremaine, M. M. Knowledge worker adoption of time management tools: Satisfaction and perceived effectiveness. In *AMCIS* (2004).

[67] Yang, L., Holtz, D., Jaffe, S., Suri, S., Sinha, S., Weston, J., Joyce, C., Shah, N. P., Sherman, K., Hecht, B., and Teevan, J. The effects of remote work on collaboration among information workers. *Nature Human Behaviour* (2021).

[68] Zhou, M., and Mockus, A. Developer fluency: Achieving true mastery in software projects. In *Proceedings of the Eighteenth ACM SIGSOFT International Symposium on Foundations of Software Engineering* (New York, NY, USA, 2010), FSE '10, Association for Computing Machinery, p. 137–146.

[69] Züger, M., Corley, C., Meyer, A. N., Li, B., Fritz, T., Shepherd, D., Augustine, V., Francis, P., Kraft, N., and Snipes, W. Reducing interruptions at work: A large-scale field study of flowlight. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, p. 61–72.

## 2. Grade and Feedback for Proposal 1

Grade: 6.0
Feedback:

Overall, really nicely written and interesting content.

- Have an abstract as well for completeness
- Nice introduction, good storyline
- RQ1: there must be quite a bit of research on this already, try to state more clearly what would be novel about your work (or if it is about replicating something)
- The way RQ2 is stated it sounds like a difficult statistical analysis, it might almost be easier to just extend RQ1 and say does it improve team cohesion and productivity
- 2.1 is generally interesting. The points on fragmentation and measuring productivity are a bit mingled and it is not clear what to take away from the fragmentation/interruptions part. It could have been good to structure this section in: measuring productivity, productivity factors, and approaches to improve productivity
- State more explicitly how your work relates to the other work, e.g. in section 2.1, 2.2 and 2.3

From Alexander:

- Generally, very well structured and leading well to the main point of social breaks
- Very extensive list of related work

Content-related feedback:

- It might also be interesting to use a "Workplace Isolation" Questionnaire for measuring team cohesion. I can give you a master's thesis that might have some additional interesting references if you are interested. Greg W Marshall, Charles E Michaels, and Jay P Mulki. “Workplace isolation: Exploring the construct and its measurement”. In: Psychology & Marketing 24.3 (2007), pp. 195–223.

Minor editorial feedback:

- There are a few missing full stops, commas, and a few typos or missing letters, and remains of re-formulated sentences; another read-through would make sense (e.g. "Despite the many advantages, there are several challenges have arisen.")
- Double-check author names (e.g. “Raph et al. [44]” vs “Ralph”)

## 3. Sample Proposal 2

# Cognitive load threshold: An early detection system

[redacted]
University of Zurich

[redacted]
University of Zurich

## 1 MOTIVATION

Mockus & Herbsleb [14] stated in 2002 that: "Expertise is difficult to measure or observe directly". With technological advancements, lightweight sensors are becoming more ubiquitous and provide the ability to gather new insights about physiological factors. A very relevant physiological factor being cognitive load. Using such devices, our goal is to find a possible threshold in cognitive load. We define such a cognitive load threshold as the point where the participant is unable to proceed with the given task. While solving the task the individual bio metric data based on heart-, skin- and respiratory-metrics will be measured. Looking at perceived task difficulty and the bio metric data gathered, we will infer the cognitive load the individual experiences. A sample of tasks with different levels of difficulty will be acquired from a coding exercise website, Codewars.com [1] (simply referred to as "Codewars" in this proposal). As stated in cognitive load theory regarding learning, the factor of previous knowledge is shown to reduce the cognitive load [22]. While there has been successful research regarding the factor of expertise by Lee et al. [10] and Crk & Kluth [6] using technologies such as electroencephalography (EEG) and eye tracking, we argue that these technologies are still too invasive to provide a valuable insight for a real-life office setting. Subsequently, we have decided to only use the Empatica E4 wristband rather than any form of headset because we think that wristbands are more likely to be worn on a day to day basis. Furthermore, the ease of use that lightweight bio metric sensors provide, are a prerequisite to be applicable in a real-life office setting, thus we focus on metrics that are provided by lightweight bio metric sensors.

We would like to investigate whether expertise has an influence on the individual cognitive load threshold measured with lightweight bio metric sensors and if an early detection system for cognitive load based on an Empatica E4 wristband can be established and provide guidelines for an implementation in an office setting. We argue that lightweight bio metric sensors are easier to use, therefore having a higher chance of being implemented and used in an office setting. Which could potentially lower the barrier of entry for further research on bio metric data of professionals based on less invasive sensing devices. Introducing a new notion of a threshold for the cognitive load and providing a set of guidelines to create an initial foundation for an early detection system of the cognitive load threshold. Possibly to extend and analyse the implications of this early detection system, such as the frequency of crossing the individual threshold for cognitive load and how this impacts the software engineers short term activity (i.e. by staying in the flow) and their long term health.

## 2 RESEARCH QUESTIONS

We have formulated the following research questions we would like to address.

- **RQ1**: Can we estimate an individual threshold of cognitive load by testing the participants on coding tasks?
- **RQ2**: Is it possible to identify whether expertise can be sensed by lightweight bio metric sensors?
- **RQ3**: Can expertise influence the individuals cognitive load threshold?
- **RQ4**: What guidelines would an early detection system need to follow in order to provide a meaningful insight?

## 3 RELATED WORK

### 3.1 Lightweight sensors

Lightweight sensors are becoming less invasive and thus applicable in real-life environments, therefore providing an additional benefit for professional developers in their daily life. Previous literature has identified that Heart rate variability (HRV) and electrodermal activity (EDA) are related to cognitive load and therefore to the difficulty the individuals face when trying to complete a task [5, 7, 16, 23]. Similar metrics were also used by Fucci et al. [8]. They investigated that lightweight wearables were able to determine whether the participant was looking at prose related or code related tasks, as the latter is associated with a higher cognitive load. Müller & Fritz [15] found that wearables were able to determine code quality concerns and task difficulty by measuring the cognitive load. Thus, using lightweight sensors to detect cognitive load is a valid approach.

### 3.2 Developer Expertise

Lee et al. [10] identified that it is indeed possible to measure software developer expertise and task difficulty, using an electroencephalography (EEG) and an eye-tracker, having a heterogeneous experiment group from professionals to undergraduates. Crk & Kluthe [6] also found that expertise can be measured when performing a programming task with an EEG, looking at a group of 34 computer science undergraduates. More expertise in a programming language leads to a higher code comprehension skill and the ability to solve programming tasks faster as found by Lee et al. [11]. Other research done by Fucci et al. [8] were not able to identify expertise differences using a homogeneous experiment group and lightweight bio metric sensors. Hence, making developer expertise by measuring bio metric data a topic of ongoing debate.

### 3.3 Cognitive Load

*3.3.1 Cognitive Load Theory.* Sweller [21] has defined the following three types of cognitive load. One of these types cannot be manipulated due to the fact that this type is about the complexity of the material itself. To be more precise, Pollock et al. [19] says that the complexity of a piece of information depends on the amount of information a reader needs to know in order to fully understand the content in question. This type of cognitive load is called *intrinsic cognitive load* (ICL). It has been shown by previous research of Sweller et al. [22] that previous expertise is an important factor in terms of learning new material. The second type is called *extraneous cognitive load* (ECL) and describes the cognitive load that is required to handle how the information is presented. One could say, that this load does not necessarily contribute to the process of understanding the information, but rather to the process of how to understand the way the information is presented ([4], [17]). The third type, *germane cognitive load* (GCL), describes the cognitive load that is imposed by the effort to understand the material [17]. As mentioned, ICL cannot be manipulated by controlling how the content is presented. However, by adjusting the media and the design of the content, ECL and GCL can be manipulated and minimized [4]. The majority of literature concerning cognitive load theory resolves around the setting of learning new material. Hence, cognitive load theory aims to reduce ECL in order to optimize the learning process for the individual [17]. Paas et al. [18] has identified the following list of cognitive load impacting characteristics of the task and the individual.

- Task related factors
  - Task format
  - Task complexity
  - Use of multimedia
  - Time pressure
  - Pacing of instruction
- Individual related factors
  - Expertise level
  - Age
  - Spatial ability

We will discuss how our experiment attempts to homogenize these factors in order to reduce ECL more in detail in section 4.2. That way, we can focus our experiment on ICL.

*3.3.2 Measurement.* Menzen et al. [13] identified in their mapping study of research about biometric data in the software engineering field that cognitive load is the most researched area, followed by emotions. Implying that cognitive load is one of the most interesting research areas. Menzen et al. [13] also categorized the physiological factors to be measured, the most prominent being the brain, followed by the eyes and the skin. Only few are focusing on non-invasive measurement such as the heart rate or the skin temperature. Other research such as the one from Antony et al. [3] stated that the pulse rate is another indicator for task difficulty. Fritz et al. [15] suggested that bio metrics can be used to determine the perceived difficulty of code elements by looking at heart rate, respiratory rate and skin temperature. Task difficulty implies a heavy cognitive load as found by Sweller [20]. Therefore, we hypothesize that there is indeed a connection between these findings of task difficulty and cognitive load.

*3.3.3 Threshold.* As we have shown, there has been a lot of research conducted on how to measure cognitive load [8, 10, 15] and what its implications are in terms of cognitive load theory. However, we were not able to find any research concerning itself with finding an estimate of a tipping point where there is too much cognitive load. The tipping point being defined as the point on a cognitive load scale after which the task in question will not be completed successfully.

### 3.4 Psychological Detection Systems

There have been advancements in research of detection systems using biological signals, namely in the context of stress [2, 24, 25]. Other research not based on bio metric data is from Majumder et al. [12], suggesting the use of facial expression recognition, looking at different face features and states of eye opening to determine the emotions of the individual. Another one proposed by Kolakowska [9], to recognize emotions via keystrokes, this could be an addition to an existing emotion detection system but is not sufficient by itself. But there is no detection system in terms of cognitive load. One could say that a potential cognitive load detection system for software engineers would face the same problems as the stress detection system [2]. After all, both serve the same purpose, to ensure the mental well-being of workers. Such challenges could include privacy issues or to be more specific, data storage issues.

## 4 APPROACH

### 4.1 Participant selection

While [6, 8] used a rather homogeneous group of participants, we are going to take a different approach similar to [10]. In order to properly differentiate participants with a different level of expertise we have defined the following categories.

| Category | Amount of expertise | Example |
|---|---|---|
| 1 | little to none | undergraduate students |
| 2 | moderate | graduate students |
| 3 | large theoretical | post-doctoral researchers |
| 4 | large practical | software professionals |

**Table 1: Participant Categories**

All students and academics will be from a computer science faculty ensuring at least a basic understanding of the principle of coding. To obtain a large enough sample, we aim to have at least 10 participants for each category. It is also important to us that both genders are equally distributed. We are aware of the difficulties we are going to encounter using this participation constellation. However, we think that this categorization will yield promising and applicable results.

### 4.2 Experiment setup

The participants will solve 4 coding tasks at the same time of day, preferably at 09:00 in the morning to guarantee that the participants solve the tasks under the same conditions and will each have a maximum of 30 minutes to solve the given tasks. On Codewars [1], tasks are created and judged by its users. Tasks are rated in terms of difficulty from a scale from one to eight, one being the hardest and eight being the easiest. By doing that, we should have a representative judgement of difficulty for each task. To conform with our Likert-scale of perceived difficulty, we choose to group difficulties into 4 categories.

| Difficulty level | Rating according to Codewars |
|---|---|
| Easy | 7, 8 |
| Rather Easy | 5, 6 |
| Rather Hard | 3, 4 |
| Hard | 1, 2 |

**Table 2: Difficulty levels**

The participants will not be told what the difficulty of the respective task is, to reduce potential biases. Each participant has to solve a task of every difficulty level. The participants will have access to the documentation of their chosen programming language but not the internet. That way, we can assure that the participants actually come up with a solution by themselves and do not simply copy someones code. Additionally, we want to prevent the participants accidentally stumbling upon the task (and its solution) on Codewars. We also want to avoid that participants already know the task and therefore, its solution. As a result, the participants will be able to read the task description and will then be asked if they have solved the given task before. If that is the case, they will be assigned another task of the same difficulty level. To further reduce differences and variability in the given difficulty level, we create a sample of possible tasks and discard the ones which show the highest variability. To tackle the influence of expertise on the experiment results we measure the expertise with lightweight bio metric data. This has not yet been proven conclusively, therefore we use the Empatica E4 wristband to either validate the findings of [8] that expertise cannot be measured with lightweight bio metric sensors or provide evidence that it is indeed possible based on our survey and experiment results. As expertise was measurable by looking at brain- and eye-related metrics [10].

As mentioned before, according to Paas et al. [18] cognitive load theory dictates a number of factors that can impact cognitive load. Since we want the focus to be on the difference in expertise, we have designed the experiment in a way that the following factors will be minimized. *Task format* and *Task complexity* will be the same for every participant and should therefore not be a difference. *Use of multimedia* and *Pacing of instruction* are not applicable to our experiment, since tasks will be exclusively formulated in text. *Time pressure* should also not be a determining factor since for every task a participant has 30 minutes which is more than enough to solve it. *Spatial ability* is not applicable in the context of our experiment. Finally, *Expertise level* is the factor in question and since none of our participants are either children or elders, *Age* is also not applicable to our experiment.

## 4.3 Data Collection

*4.3.1 Biometrics.* While the participants are solving the tasks, their bio metrics are being tracked using an Empatica E4 wristband, similar to [8] and [15]. We argue that the wristband is good enough to measure the relevant bio metric data that indicates the level of cognitive load. More specifically inspecting the participants skin-, respiratory- and heart related-metrics, such as EDA and skin temperature for the HT and HRV. By combining the findings from [15] & [20], we use the bio metric data to determine the task difficulty for the participant and link this perceived difficulty to the actual cognitive load the participant experiences.

*4.3.2 Survey.* Moreover, we will conduct a survey to address the perceived difficulty of each task. After each task, regardless of the fact whether the participant was able to solve the task or not, the participant will be asked to rate the difficulty of the task according to the difficulty levels in table 2. The time needed to answer the survey is not part of time allocated to solve the task. Whether the participant was able to solve the task or not will also be tracked.

## 4.4 Data Analysis

To predict a personal threshold we look at the individuals cognitive load levels as they complete tasks. The threshold will be the point on the cognitive load scale where the participant will not be able to successfully complete the task. To calculate the individual cognitive load threshold, a machine learning algorithm will be applied to analyse the participants bio metric data in EDA, HRV and HT. We will then cross reference the survey results to check whether the difficulty of the task was reflected in the bio metric data and thus in the cognitive load. The expertise will be another dimension for analysis, as this might impact the perceived difficulty and correlate to the experienced cognitive load during the coding task. To combine the findings and produce a set of guidelines for an early detection system, we plan to deploy the machine learning algorithms which have proven themselves as most optimal during analysis. These will train themselves based on the individual baseline data they will get from the users, either by completing the experiment or from other sources.

# 5 FURTHER RESEARCH

Replicating the results of this paper with a different set of bio metric sensors, mainly eye and brain related metrics, would help to further establish lightweight bio metric sensors as an alternative to the more invasive bio metric sensors, which are more frequently used in research but less often seen in practice. Furthermore, we would like to investigate how the surpassing of the cognitive load threshold impacts the software engineers. Looking at short term behavioral responses, such as not being able to focus on the task and sensing a feeling of frustration and long term issues such as mental health concerns, seen in burnouts and depression. Another option would be to extend existing stress detection frameworks by adding the dimension of cognitive load in order to render it more applicable for software engineers.

# 6 WORD OF HONOR

We [redacted], hereby declare that we have produced this work independently and have used no other than the listed tools and sources.

# REFERENCES


[1] [n.d.]. Codewars. https://www.codewars.com/. Accessed: 2010-10-22.

[2] Ane Alberdi, Asier Aztiria, and Adrian Basarab. 2016. Towards an automatic early stress recognition system for office environments based on multimodal measurements: A review. *Journal of biomedical informatics* 59 (2016), 49–75.

[3] Lisa Anthony, Patrick Carrington, Peng Chu, Christopher Kidd, Jianwei Lai, and Andrew Sears. 2011. Gesture dynamics: Features sensitive to task difficulty and correlated with physiological sensors. *Stress* 1418, 360 (2011), 312–316.

[4] Roland Brunken, Jan Plass, and Detlev Leutner. 2003. Direct Measurement of Cognitive Load in Multimedia Learning. *Educational Psychologist* 38, 1 (2003), 53–61. https://doi.org/10.1207/S15326985EP3801_7

[5] Michael Cheung and Stephen Porges. 1977. Respiratory influences on cardiac responses during attention. *Physiological Psychology* 5 (03 1977), 53–57. https://doi.org/10.3758/BF03335298

[6] Igor Crk and Timothy Kluthe. 2014. Toward using alpha and theta brain waves to quantify programmer expertise. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 5373–5376.

[7] Thomas Fritz, Andrew Begel, Sebastian C. Müller, Serap Yigit-Elliott, and Manuela Züger. 2014. Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development. In *Proceedings of the 36th International Conference on Software Engineering* (Hyderabad, India) *(ICSE 2014)*. Association for Computing Machinery, New York, NY, USA, 402–413. https://doi.org/10.1145/2568225.2568266

[8] Davide Fucci, Daniela Girardi, Nicole Novielli, Luigi Quaranta, and Filippo Lanubile. 2019. A replication study on code comprehension and expertise using lightweight biometric sensors. In *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*. IEEE, 311–322.

[9] Agata Kołakowska. 2013. A review of emotion recognition methods based on keystroke dynamics and mouse movements. In *2013 6th International Conference on Human System Interactions (HSI)*. IEEE, 548–555.

[10] Seolhwa Lee, Danial Hooshyar, Hyesung Ji, Kichun Nam, and Heuiseok Lim. 2018. Mining biometric data to predict programmer expertise and task difficulty. *Cluster Computing* 21 (03 2018). https://doi.org/10.1007/s10586-017-0746-2

[11] Seolhwa Lee, Andrew Matteson, Danial Hooshyar, SongHyun Kim, JaeBum Jung, GiChun Nam, and HeuiSeok Lim. 2016. Comparing programming language comprehension between novice and expert programmers using eeg analysis. In *2016 IEEE 16th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, 350–355.

[12] Anima Majumder, Laxmidhar Behera, and Venkatesh K Subramanian. 2014. Emotion recognition from geometric facial features using self-organizing map. *Pattern Recognition* 47, 3 (2014), 1282–1293.

[13] Juliano Paulo Menzen, Kleinner Farias, and Vinicius Bischoff. 2020. Using biometric data in software engineering: a systematic mapping study. *Behaviour & Information Technology* (2020), 1–23.

[14] Audris Mockus and James D. Herbsleb. 2002. Expertise Browser: A Quantitative Approach to Identifying Expertise. In *Proceedings of the 24th International Conference on Software Engineering (ICSE '02)*. Association for Computing Machinery, New York, NY, USA, 503–512. https://doi.org/10.1145/581339.581401

[15] Sebastian C Müller and Thomas Fritz. 2016. Using (bio) metrics to predict code quality online. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. IEEE, 452–463.

[16] Takao Nakagawa, Yasutaka Kamei, Hidetake Uwano, Akito Monden, Kenichi Matsumoto, and Daniel German. 2014. Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: A controlled experiment. https://doi.org/10.1145/2591062.2591098

[17] Fred Paas, Alexander Renkl, and John Sweller. 2003. Cognitive load theory and instructional design: Recent developments. *Educational psychologist* 38, 1 (2003), 1–4.

[18] Fred Paas, Juhani E Tuovinen, Huib Tabbers, and Pascal WM Van Gerven. 2003. Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist* 38, 1 (2003), 63–71.

[19] Edwina Pollock, Paul Chandler, and John Sweller. 2002. Assimilating complex information. *Learning and instruction* 12, 1 (2002), 61–86.

[20] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science* 12, 2 (1988), 257–285.

[21] John Sweller. 1999. Instructional Design. *Australian Educational Review* (1999).

[22] John Sweller, Slava Kalyuga, Paul Ayres, and Paul Chandler. 2003. The Expertise Reversal Effect. *Educational Psychologist* 38, 1 (2003), 23–31. https://doi.org/10.1207/S15326985EP3801_4

[23] J.A. Veltman and Anthony Gaillard. 1998. Physiological workload reactions to increasing levels of task difficulty. *Ergonomics* 41, 5 (1998), 656–669.

[24] Jacqueline Wijsman, Bernard Grundlehner, Hao Liu, Hermie Hermens, and Julien Penders. 2011. Towards mental stress detection using wearable physiological sensors. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 1798–1801.

[25] Jing Zhai and Armando Barreto. 2006. Stress detection in computer users based on digital signal processing of noninvasive physiological variables. In *2006 international conference of the IEEE engineering in medicine and biology society*. IEEE, 1355–1358.


## LIST OF TABLES

## 4. Grade and Feedback for Proposal 2

Grade: 4.75
Feedback:

Overall the idea of a threshold of cognitive load at which an individual is "overwhelmed" is an interesting idea, but you need to provide more background, motivate the research better and better discuss related work in the introduction to state what's novel about your idea. Some of the questions you pose have already been addressed in studies and you need to talk about this. In its current form, the research proposal is quite rough: the motivation is missing for the specific threshold, the relation to other work in the area is not stated clearly and in many places (related work, data analysis, data collection), the information is very high-level and more details should be provided.

- Please provide an abstract
- Motivation/Introduction:
    - How is cognitive load related to expertise?
    - Motivate why your research is of interest, not just what you do; also state what others have done so far
    - Why would you want to examine whether expertise has an influence on cognitive load? Hasn't that been established in the past and what is the related work with respect to this topic
    - How is "cognitive load" defined? Provide a definition and more context/background on it!
- Related Work:
    - Overall, the categorization needs to be motivated and an overview needs to be provided. Also, the current structure is repetitive and the reader has to put work into understanding how it all relates. Finally, there is very little detail provided on a lot of the research.
    - Provide an introduction to it and state how you divided up your related work and why this categorization makes sense
    - 3.1: what do you refer to as "lightweight"? Also, is this solely for measuring cognitive load or also other aspects? The overview here is very rough and more details on the related work in this section should be provided
    - 3.2: provide more context of why expertise is of interest and how it is defined; again, this paragraph should provide more details on the related work and be introduced
    - 3.3: definition comes too late; state how the factors of Paas are related to the three types of cognitive load
    - What's the difference between 3.1 and 3.3.2?
    - 3.3.3: why do you think there is such a tipping point?
    - How is 3.4 different to 3.3.2 or 3.1
    - Related work should also state how your approach is different/novel with respect to the related work, currently, that's not really been done.
- Approach:
    - Expertise is with respect to a task: you first need to introduce the task or the domain to talk about expertise; further, why is theoretical vs practical of importance
    - State what kind of tasks the tasks on codewars are and why they are good tasks
- Data collection & analysis
    - What does HT stand for (abbreviation not introduced)

- Be more specific which exact metrics you are going to use, currently it's very rough, also on how you would analyze it or segment it
- Survey should also show up in the procedure / setup section
- Overall, the sections are very rough and based on related work, you should be able to provide more details on what exactly you are going to do in these steps

- Editorial feedback
  - English should be revised and improved; we recommend proof-reading before submitting
- Additional comments
  - The participant selection as base for the expertise might not be sufficient. There might be undergraduate students with lots of practical programming experience, and post-doctoral researchers with very little depending on their area of expertise. Also there might be software professionals that just started after finishing their studies.
  - Your assumption that not being able to solve a task means that the cognitive load threshold was reached might not hold. There might be cases where people understood the problem in another way, and until the time limit were not able to follow down the right path. Does that mean they reached their cognitive load threshold?
  - For titles, the capitalization rules suggest capitalizing your titles as follows: Cognitive Load Threshold: An Early Detection System. (e.g. https://capitalizemytitle.com/)
  - It might be interesting to check for research on "cognitive overload"