Student Response to Teaching of Memory Cues and Resumption Strategies in Computer Science Classes

Noah John
Colorado State University
njohn@rams.colostate.edu

Jaime Ruiz
Colorado State University
jgruiz@cs.colostate.edu

ABSTRACT

Programming is a creative process that requires the ability to concentrate and juggle multiple concepts simultaneously in one's mind. Existing research shows there is a tangible cost when a programmer is interrupted as the programmer must recover the context of his work and refocus on the task at hand. However, CS students are rarely taught about interruptions and how to manage them. Instead, teaching tends to focus only on technical concepts. In addition, there is little research on interruptions with respect to CS students. Therefore, our research examines what happens when CS students are taught about interruptions and how to cope with them.

The objective of this paper is to determine if CS students are affected by interruptions, what knowledge CS students possess regarding memory cues and resumption strategies, and what their opinion is of this material. We conducted a study with approximately two-hundred undergraduate CS students to answer these questions. Our study was comprised of an initial questionnaire, a seminar on memory cues and resumption strategies, and a follow-up questionnaire. Our results demonstrate that CS students are affected by interruptions, but 73% of students report not knowing methods to mitigate them. After learning about memory cues and resumption strategies, students report that the material was useful and that they want to study it. Their most significant feedback is that they have a strong desire to include these techniques in CS curriculums, reporting a mean score of 7.78 out of 10, where 0 signifies strong disagreement and 10 signifies strong agreement.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer science education; K.3.2 [**Computer and Information Science Education**]: Curriculum; H.1.2 [**User/Machine Systems**]: Human Factors

General Terms

Human Factors

SIGCSE '15 March 04 - 07 2015, Kansas City, MO, USA

ACM 978-1-4503-2966-8/15/03 ...$15.00.
http://dx.doi.org/10.1145/2676723.2677234.

Keywords

interruptions; task resumption; cues; students; teaching

1. INTRODUCTION

Computer science students are taught a myriad of topics over the course of their education. In an undergraduate curriculum, it is common to see classes on programming languages, programming paradigms, data structures, algorithms analysis, and operating systems. As a student, it is important to learn the fundamental concepts taught in these classes to become a knowledgeable software engineer. Furthermore, a large portion of the work CS students do is programming-related. Whether the language is Java, C++, Scheme, or assembly, programming represents a majority of a CS student's homework and is the primary benchmark for a student's performance. And while the technical details and concepts, such as object hierarchy or C++ pointer syntax, are conveyed to students in class, significantly less is conveyed about the art of programming. At its core, programming requires strong logical reasoning, the knack for solving problems, and smart design skills. However, it is also a creative process that requires the ability to concentrate and juggle multiple concepts simultaneously. Additionally, there is a tangible cost when a programmer is interrupted as the programmer must recover the context of his work and refocus on the task at hand. On average, industry software engineers need fifteen minutes to recover from an interruption and return to a focused and productive state [28]. While there has been extensive research on the topic of interruptions, their effects, and methods for minimizing their impact [20], very little has reached CS curriculums.

In contrast, consider an English curriculum. One of its goals is to produce proficient and eloquent writers capable of composing various types of prose. Writing is recognized as a creative process that requires concentration and the managing of multiple ideas at the same time. English students, along with a majority of the general public, are familiar with the term "writer's block", which refers to a condition when an author is unable to produce new material. Furthermore, writing teachers dedicate some of their time to discussing this barrier and how to overcome it. Informal discussion of techniques and experiences has been deemed useful by the writing community and writer's block is one of the concepts covered in an English curriculum. While writer's block is not analogous to a programmer being interrupted and then having to resume a task, there are parallels. In both cases, the person is unproductive for a period of time and must find a way to overcome this state. Additionally, as a writer must remember his train of thought, a programmer must recover his context and his intended next step. We discuss these topics with English students. But how would CS students react if they were taught about interruptions and how to cope with them?

The objective of this paper is to determine if CS students are affected by interruptions, what knowledge students possess regarding memory cues and resumption strategies, and what their opinion is of learning this material.

Our method consisted of conducting a study with approximately two-hundred undergraduate CS students. First, we surveyed their programming habits, their work, and their previous knowledge on memory cues and resumption strategies. Next, we gave a seminar on interruptions, memory cues, and resumption strategies to these same students. Finally, six weeks later, we conducted a follow-up survey to gather their opinions on the seminar and examine if it had any perceived effect on their work.

Our results demonstrate that CS students are affected by interruptions and that they report the same resumption lag as industry engineers. We also see that 73% of students report not knowing about memory cues and resumption strategies. Finally, we find that students believe the material is useful and have a strong desire to learn these techniques.

2. BACKGROUND

There has been extensive research examining the effects of interruptions on productivity. A critical and foundational paper was published by McFarlane and Latorella [22] which thoroughly reviewed existing literature to argue its fundamental place in HCI research and how vitally important it is to understand interruptions and their effects as technology continues to evolve. They began by analyzing three application domains to show the necessity for considering interruptions in informed design decisions. They continued by summarizing their respective research that formalized and categorized interruptions with respect to tasks and summarizing their respective research into managing interruptions. They ultimately concluded with a discussion of existing strategies for mitigating the cost of interruptions and proposed that UI design was the most promising strategy, hence the importance of interruption research in HCI. Generally, research falls into four areas: interruption cost and effects, reasons for interruptions, interruption recovery, and coordinating interruptions. These four areas are summarized below.

2.1 Interuption cost and effects

One area of research has centered on the cost and effects of interrupting a person performing a task. Adamczyk and Bailey [1] conducted a user study to determine which are the least costly moments to interrupt someone's task. Hodgetts and Jones [12] [11] studied the effect of interruptions during different points in a problem-solving process. One objective is to leverage these findings to inform the design of an attention manager system [1]. Robertson *et al.* [26] studied the effects of *immediate-style* interruptions and *negotiated-style* interruptions on users debugging spreadsheet errors. Immediate-style interruptions are interruptions that require user action. Conversely, negotiated-style interruptions were defined in McFarlane's classification of interruptions [21] as interruptions that inform the user of a pending message but do not force them to acknowledge it immediately. Comparing the two styles, Robertson *et al.* found that negotiated-style were more effective.

Iqbal and Bailey [13] identified concepts from cognitive psychology that could be used to estimate the Cost of Interruption (COI) and Salvucci *et al.* [27] proposed a theoretical framework for defining the states of a multitasking continuum which included the states of interruption and resumption. Marulanda-Carter and Jackson [19] studied e-mail usage in a large international car rental company, measuring how e-mail interruptions affected productivity and the prevalence of e-mail addiction.

2.2 Reasons for interruptions

Some researchers have studied the reason for interruptions by observing the behaviors of people in an office and noting interruptions and their motivations. Czerwinski *et al.* [4] conducted a diary study of eleven experienced office software users and found that "returned-to" tasks were most difficult to switch to, required the most time to complete, and experienced the most interruptions (due to the task length). Szóstek and Markopoulos [28] observed the behaviors of people in an office and noted interruptions and their motivations. The study focused on face-to-face communication and highlighted factors such as urgency, employee hierarchy, and perceived availability.

Harr and Kaptelinin [8] thoroughly summarized existing research of interruptions and underlined the lack of attention paid to the social component of interruptions and the various dynamics that are in play. The paper identified four variables that affected the likelihood of an interruption occurring and proposed four related "ripple effects" that add additional cost to an individual interruption. In follow-up work, Harr and Kaptelinin [9] verified the effect of social context in a person's decision to interrupt someone else.

Jett and George [15] defined four types of interruption: intrusion, break, distraction, and discrepancy and detailed the positive and negative consequences for the interrupted person. Jin and Dabbish [16] defined a typology for self-interruptions. They ultimately identified seven types of self-interruption: adjustment, break, inquiry, recollection, routine, trigger, and wait. For each type, he observed their effects and duration.

2.3 Interruption recovery

Another area of research has been how one recovers from an interruption. Altmann and Trafton [2] studied the recovery process of users that were interrupted during the course of playing a computer strategy game and Jackson *et al.* [14] looked at the recovery time for e-mail interruptions of users of Microsoft Outlook in a software company.

There have been GUI prototypes that attempted to intelligently alert and interrupt the user (where the software itself has to interrupt the user) while assisting the user in quickly resuming his previous task after dealing with the interruption [6] and a prototype program that helps users organize related items on the Windows taskbar [4].

Parnin and DeLine [24] conducted a survey of software engineers to compile a list of cues that programmers use to resume a task that had been interrupted and then tested the usefulness of three memory cues: note-taking, a content timeline, and a degree-of-interest (DOI) treeview. In follow-up research, Parnin and Rugaber [25] looked at the techniques and strategies programmers use to resume an interrupted task.

2.4 Coordinating interruptions

The notion of coordinating interruptions is rooted to research done by Latorella [17] and McFarlane [20] where "coordinating interruption" was defined as the method one takes in interrupting someone's task. McFarlane [20] empirically evaluated four methods (immediate, negotiated, mediated, and scheduled) for coordinating interruptions and its results suggested that a method's effectiveness depended on the nature of the task being interrupted. He then built on this paper to evaluate four UI design solutions for coordinating interruptions [21].

Gievska *et al.* [7] looked at computer-mediated coordination and conducted an experiment that compared the performance of users that were interrupted randomly to that of users that were interrupted strategically using an interruption mediator. This paper established that workers were less distracted, were less annoyed, and found their work more manageable with mediated interruptions. Also, Kern *et al.* [18] had their study participants rank the personal and social disturbance level of various interruptions in differing environments. He then experimented with sensors to evaluate a person's personal and social interruptability and achieved a 98.1% recognition score for social interruptability. Palanque *et al.* [23] analyzed interruptions using model-based analytical techniques that could be used in designing interruption-tolerant systems.

2.5 Specialized interruption research

In parallel to the topics already mentioned, in should be noted that a significant portion of research into interruptions focuses on two disciplines: software engineering [3] [24] [25] [26] [29] and e-mail usage [14] [19].

We were unable to find research that investigated the resumption lag of CS students or the effect of teaching them about interruptions. This formed the basis of our motivation to pursue this study.

3. METHOD

Our research goal was to examine the effects on college students of presenting them instruction on interruptions, memory cues, and resumption strategies. More specifically, our objective was to support the following hypotheses:

- **H1** College students are affected by interruptions and have the same resumption lag as industry software engineers.
- **H2** College students' knowledge of memory cues and resumption strategies is limited.
- **H3** College students react favorably to being taught memory cues and resumption strategies in CS courses and believe they should be included in a college curriculum.
  - **H3a** The student's age will affect the student's response to being taught these topics.
  - **H3b** The student's programming experience will affect the student's response to being taught these topics.
  - **H3c** The student's years in college will affect the student's response to being taught these topics.
  - **H3d** The student's previous knowledge of memory cues and resumption strategies will affect the student's response to being taught these topics.
  - **H3e** College students will report that learning memory cues and resumption strategies benefited them outside of CS courses.

To accomplish this, the study had three components: an initial survey, a seminar on memory cues and resumption strategies, and a follow-up survey. The study involved 198 undergraduate CS students attending a public university. Their ages ranged from 18 to 39, their experience in programming ranged from a few months to ten years, and their college experience ranged from one semester to eight years. They were not compensated for participating.

The surveys and seminar were given to five different computer courses: one freshman-level course, one sophomore-level course, one junior-level course, and two senior-level courses. The first survey and seminar were given in the middle of the semester. The follow-up survey was given at the end of the semester, roughly six weeks later.

3.1 The initial questionnaire

The first questionnaire was given in the middle of the semester and consisted of 18 questions. Students completed the survey in 5 to 10 minutes. The survey gathered three types of data: background information on the student, how they worked on their assignments, and their prior knowledge of memory cues and resumption strategies. Once the questionnaires were completed, the seminar started.

3.2 The seminar on memory cues and resumption strategies

There was a seminar held during the lecture period of five different classes. The format was lecture-based, students were encouraged to interrupt with questions, and a white board was heavily used by the instructor. All seminars covered the same material and lasted 25 to 30 minutes depending on the number of student questions. The material was presented in the following order: Effects of Interruptions, Memory Cues and Resumption Strategies, Notion of a Brain Dump, Resisting to Coordinate Interruptions Immediately [20], Note Taking, Code Commenting, Window Arrangement, Roadblock Cues [25], Selective Suspension [10], and Concept Maps.[1]

3.3 The follow-up questionnaire

The second survey was given six weeks after the seminar. The survey consisted of two parts. Part one was comprised of six continuous rating scale questions. The first three questions had "highly not useful" on the left and "highly useful" on the right. The final three questions had "strongly disagree" on the left and "strongly agree" to the right. Participants were asked to mark a vertical line through a horizontal line scale to indicate their opinion for each of the six statements. We codified their answers by measurement, which gave us real numbers on the continuous interval $[0, 10]$. Figure 1 shows one of the questions. The statements centered on the seminar's effect on students. More details can be found in the subsequent section. The second part of the questionnaire asked for details about the effect of the seminar. It also contained two questions which were identical to questions from the initial questionnaire and asked students about their start and resumption lag times. Space was also provided to allow participants to leave additional comments.

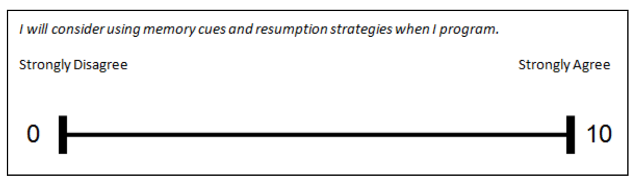

Figure 1: Example of continuous rating scale question from follow-up questionnaire.

4. RESULTS

The goal of this section is to support or reject the hypotheses defined in section 3 by presenting the results of the two questionnaires.

4.1 H1 - College students are affected by interruptions and have the same resumption lag as industry software engineers.

Students reported a mean resumption lag of 15.12 minutes (SD: 12.72). This is consistent with the fifteen minutes cited by industry software engineers in studies by DeMarco and Lister [5] and Solingen *et al.* [29]. Students were also asked what caused a session of work to be productive or unproductive. Table 1 shows a coalesced summary of student responses.

[1] Full description of seminar: http://noahjohn.us/research/memcues.html

Student Response	%
distractions	25%
understanding/clear requirements	24%
state of mind/mood	10%
focus	9%
time	6%
fatigue	5%

Table 1: Top contributors to productive/unproductive work for CS students.

The two most mentioned factors that affected productivity were distractions (25%) and understanding/clear requirements (24%). Distractions were primarily related to environment, which included noise, other people, and non-work technology. These factors were described by students in their feedback.

P76 *"Productive: no distraction from classmates/social media."*
P85 *"[affected by] distractions/loud talking in the computer lab."*
P119 *"If I don't have interruptions, I will be productive."*

We were encouraged by the comments on distractions because in many of the examples cited, the distraction was synonymous with a type of interruption and the seminar would directly deal with this topic. This provided evidence that CS students are affected by interruptions while doing their assignments. While this was expected, we felt it was important to have supporting feedback from students.

4.2 H2 - College students' knowledge of memory cues and resumption strategies is limited.

Nearly three-fourths (73%) of students reported not knowing memory cues or resumption strategies. Of the 27% that heard of these techniques, 42% said they learned about them in a psychology class and 14% did not remember where they learned about them. No student cited a CS class as a source for learning memory cues.

We also asked if they did anything prior to stepping away from an assignment that helped them refocus more quickly when they returned to work, which could indicate if they implicitly used memory cues and resumption strategies. The most common response (48%) was "no". The second most common response (27%) was that they left comments for themselves.

4.3 H3 - College students react favorably to being taught memory cues and resumption strategies in CS courses and believe they should be included in a college curriculum.

Part one of the follow-up questionnaire contained six continuous rating scale questions. For the first three questions (FQ1-3), 0 implied "highly not useful" and 10 implied "highly useful", and for the last three (FQ4-6), 0 implied "strongly disagree" and 10 implied "strongly agree". Table 2 contains a summary of student answers and Figure 2 displays a boxplot for their answers.

Students found the seminar useful (M: 6.24 SD: 2.04) and applicable to their programming assignments (M: 6.27 SD: 2.24). Students elaborated on these positions with their comments:

P17 *"I sometimes leave an assignment without finishing it, so it is helpful to create ways to jumpstart the work when I return."*
P18 *"[The techniques I learned] helped me reduce the time it took to get back into working after a break or interruption."*

Statement	Mean	Median	SD
I found the information in the memory cues seminar to be... (FQ1)	6.24	6.50	2.04
For my programming assignments, I found the information in the memory cues seminar to be... (FQ2)	6.27	6.5	2.24
For my non-CS assignments, I found the information in the memory cues seminar to be... (FQ3)	4.86	5.00	2.28
I will consider using memory cues and resumption strategies when I program. (FQ4)	7.41	7.75	2.02
I think memory cues and resumption strategies should be presented to CS students at some point of their curriculum. (FQ5)	7.78	8.00	1.93
I am interested in learning more about memory cues, resumption strategies, or similar topics. (FQ6)	6.34	6.53	2.40

Table 2: Summary of answers to part one of the follow-up questionnaire.

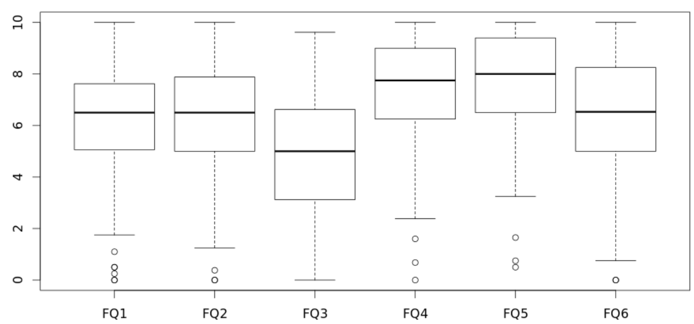

Figure 2: Blox plot of follow-up answers

More strongly, students said they would consider using the techniques (M: 7.41, SD: 2.02) and were highly in favor of teaching memory cues (M: 7.78, SD: 1.93). As one student articulated in his feedback:

P163 *"Seriously, try to get this stuff embedded in the CS courses. It is very important to be aware of this stuff."*

Also, they wanted to learn more about the subject (M: 6.34, SD: 2.40). For example, one student wrote:

P176 *"I'd like to know more about this topic; [I] wish it was covered in some CS course."*

Part two of the follow-up questionnaire focused on the seminar's effect and its answers are summarized in Table 3.

Question	Yes	No
Did the seminar change the way you think about programming?	57%	43%
Did you do anything differently during recent programming assignments?	43%	57%

Table 3: Summary of answers to part two of the follow-up questionnaire.

The 57% of students that cited a difference in thinking were asked to elaborate and provide an example. Here is a sample of what the students said:

P126 *"I always write a note to myself in my program now when I find a stopping place. This helps me resume my original thought process and helps the way I think about programming."*

P95 *"[The seminar] made me aware of the pitfalls of memory and productivity."*

While 57% of students stated they did nothing differently in their recent assignments, 5% of students mentioned their assignments were too short and/or easy to warrant using the cues and recognized their usefulness in the future with more complex assignments. As one student reported:

P66 *"It's [the seminar's topics] good stuff but I didn't use it this semester because the assignments weren't that hard. I'm sure I will next semester."*

Exploring the sub-hypotheses (H3a-H3e), Welch Two Sample t-tests were conducted on the six continuous rating scale questions (FQ1-FQ6) to determine if there was a difference between students that reported prior knowledge of memory cues and resumption strategies and students that did not report prior knowledge. Tests on FQ1 ($p < 0.05$), FQ2 ($p < 0.01$), and FQ4 ($p < 0.10$) showed statistically significant differences, which provided some support to H3d.

However, not all hypotheses were supported. Students did not report a noticable benefit to the seminar outside of CS courses (H3e) (M: 4.86, SD: 2.28). Analysis of variance was conducted to examine the effect of age (H3a), years in college (H3c), and programming experience (H3b) on FQ1-FQ6. Results showed no significant difference among different values of the dependent variable ($p > 0.10$). The only exception was an effect of programming experience on FQ5 ($p < 0.05$). However, post-hoc analysis with Tukey's HSD correction showed no distinguishable differences among different values of the dependent variable. Hence, our study did not demonstrate results that supported H3a, H3b, H3c, or H3e.

5. DISCUSSION

By showing that college students are affected by interruptions in similar ways to industry software engineers (H1), we were able to extend findings of previous research [5][29]. Students' average resumption lag was 15.12 minutes, which is consistent with the fifteen minutes industry software engineers reported.

In addition, we believe the feedback from the 198 students was meaningful and consistent. Respondents generally provided fair amounts of information on the questionnaires as evidenced by their feedback. Consistency was demonstrated by their answers. For example, in the initial survey, we included two questions that were intended to ask the same thing and only differ in wording. In one question, we asked what was the average length of time of a productive sitting for the student. In the other, we asked how long the student was able to "stay in the zone". We performed a Welch Two Sample t-test on the results of the former (M: 2.40, SD: 1.34) and the latter (M: 2.60, SD: 3.43) and found they were indistinguishable ($p > 0.50$).

Students of all ages and experiences are of the strong opinion (M: 7.78, SD: 1.93) that this material should be part of their undergraduate CS curriculum. Furthermore, 57% changed the way they thought about programming as a result of the seminar and 43% did things differently. Additionally, a few more mentioned they would have done something new if their assignments were more challenging. These results tell us that college students react favorably to being taught memory cues and resumption strategies in CS courses and believe they should be included in a college curriculum (H3).

We expected the student's age, years in college, and programming experience to affect the student's response to being taught memory cues and resumption strategies (H3a, H3b, H3c). However, none of these factors affected students' responses significantly. The significant difference was between students with and without prior knowledge of memory cues and resumption strategies (H3d), where the quarter of students with prior knowledge finding the seminar less useful, yet they agreed that the material should be taught.

We also thought students would report a benefit to learning about memory cues and resumption strategies that extended to non-CS courses (H3e). However, our results showed that students were neutral, at best, to that statement.

While other disciplines, namely Psychology, touch on memory cues and resumption strategies, CS teachers in universities cannot rely on students being exposed to these subjects, as shown by the results to H2. Therefore, there is a clear knowledge gap that many CS students have. And, according to their feedback (H3), they desire to fill this gap and learn more.

Finally, we believe another takeaway is that while roughly half of students already intuitively utilized memory cues to some degree, typically in the form of code comments, the seminar reinforced those habits and encouraged students to take more steps to help reduce their resumption lag. As one student wrote:

P181 *"I've been doing something similar [to what was taught in the seminar], but this made it more concrete. I'd like to learn more."*

5.1 Limitations

In this section, we remark on limitations and possible counterarguments to our work.

First, we have not established to what extent memory cues and resumption strategies should be taught, only that students believed that they should be taught. Readers can use the description of the seminar in the method section to see what material was covered, but this is not intended to be a definitive guide of what CS teachers should adopt. Again, our objective was to show that teaching students about interruptions, memory cues, and resumption strategies is welcomed by students, not how it should be done in the classroom.

Secondly, our participants came from the same university. However, our institution is similar to other public state universities. While we believe there was sufficient diversity among our participants, we intend to conduct a second study across multiple education institutions.

Thirdly, students might have been biased to provide positive feedback because they interpreted positive feedback as being helpful. This occurred to the authors at the outset of the study, and clear verbal instructions were emphasized to the participants to be honest and that there were no right answers. Additionally, the course instructors reiterated to their classes that no favoritism or reward would be given to those who participated in the study.

6. FUTURE WORK

A natural progression of this research would be to establish what topics are the most valuable to cover with respect to interruptions, memory cues, and resumption strategies. While our seminar was well-received by students, we have no way of knowing if the seminar could be improved and to what extent material should be covered. Hence, we envision a new study to determine which topics are most valuable to share with students would be a logical next step.

We also believe there is a vast potential for research into "programming as a craft" that could take existing knowledge and research of the creative process and cognitive psychology and apply it to programming. We believe knowledge from these areas that is introduced into CS can only enhance the field and improve the education of its students.

7. CONCLUSION

In this paper we examined if students are affected by interruptions, what knowledge students possess regarding memory cues and resumption strategies, and what their opinion is of learning this material. We found that students were affected by interruptions, but nearly three-fourths of students reported not knowing about methods to mitigate them. After learning about memory cues and resumption strategies, students reported that the material was useful and that they wanted to learn about them. Their most significant feedback was that they had a strong desire to include these techniques in CS curriculums. Furthermore, their feedback, including their interest in learning more and their opinion that it is a worthwhile topic to be covered in class, was independent of their age, college experience, and work experience. Based on our findings, we hope this paper will open the door to more research into teaching students about the creative process, so that like English students, their knowledge extends beyond the technical details of their craft.

8. REFERENCES

[1] P. D. Adamczyk and B. P. Bailey. If not now, when?: The effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 271–278, New York, NY, USA, 2004. ACM.

[2] E. M. Altmann and J. G. Trafton. Timecourse of recovery from task interruption: Data and a model. *Psychonomic Bulletin & Review*, 14(6):1079–1084, 2007.

[3] J. Chong and R. Siino. Interruptions on software teams: A comparison of paired and solo programmers. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, CSCW '06, pages 29–38, New York, NY, USA, 2006. ACM.

[4] M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI '04*, pages 175–182. ACM, 2004.

[5] T. DeMarco and T. Lister. Peopleware: Productive projects and teams dorset house. *New York*, 1999.

[6] J. L. Franke, J. J. Daniels, and D. C. McFarlane. Recovering context after interruption. In *Proceedings 24th Annual Meeting of the Cognitive Science Society (CogSci 2002)*, pages 310–315, 2002.

[7] S. Gievska, R. Lindeman, and J. Sibert. Examining the qualitative gains of mediating human interruptions during hci. In *Proceedings 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada*, 2005.

[8] R. Harr and V. Kaptelinin. Unpacking the social dimension of external interruptions. In *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, GROUP '07, pages 399–408, New York, NY, USA, 2007. ACM.

[9] R. Harr and V. Kaptelinin. Interrupting or not: Exploring the effect of social context on interrupters' decision making. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, NordiCHI '12, pages 707–710, New York, NY, USA, 2012. ACM.

[10] E. Hemingway, G. Seldes, T. Dreiser, L. Hughes, and A. D. Ficke. *Monologue to the maestro: A high seas letter*. Esquire, 1935.

[11] H. M. Hodgetts and D. M. Jones. Resuming an interrupted task: Activation and decay in goal memory. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society (CogSci 2006)*, page 2506, 2006.

[12] H. M. Hodgetts and D. M. Jones. Reminders, alerts and pop-ups: The cost of computer-initiated interruptions. In *Proceedings of the 12th International Conference on Human-computer Interaction: Interaction Design and Usability*, HCI'07, pages 818–826, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] S. T. Iqbal and B. P. Bailey. Leveraging characteristics of task structure to predict the cost of interruption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 741–750, New York, NY, USA, 2006. ACM.

[14] T. Jackson, R. Dawson, and D. Wilson. Reducing the effect of email interruptions on employees. *Int. J. Inf. Manag.*, 23(1):55–65, Feb. 2003.

[15] Q. R. Jett and J. M. George. Work interrupted: A closer look at the role of interruptions in organizational life. *Academy of Management Review*, 28(3):494–507, 2003.

[16] J. Jin and L. A. Dabbish. Self-interruption on the computer: a typology of discretionary task interleaving. In *Proceedings of the SIGCHI '09*, pages 1799–1808. ACM, 2009.

[17] L. Kara A. Effects of modality on interrupted flight deck performance: Implications for data link. Technical report, 1998.

[18] N. Kern, S. Antifakos, B. Schiele, and A. Schwaninger. A model for human interruptability: Experimental evaluation and automatic estimation from wearable sensors. In *Proceedings of the Eighth International Symposium on Wearable Computers*, ISWC '04, pages 158–165, Washington, DC, USA, 2004. IEEE Computer Society.

[19] L. Marulanda-Carter and T. W. Jackson. Effects of e-mail addiction and interruptions on employees. *Journal of Systems and Information Technology*, 14(1):82–94, 2012.

[20] D. McFarlane. Coordinating the interruption of people in human-computer interaction. In A. Sasse and C. Johnson, editors, *Human-computer interaction, INTERACT*, volume 99, page 295, 1999.

[21] D. McFarlane. Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. *Hum.-Comput. Interact.*, 17(1):63–139, Mar. 2002.

[22] D. C. McFarlane and K. A. Latorella. The scope and importance of human interruption in human-computer interaction design. *Hum.-Comput. Interact.*, 17(1):1–61, Mar. 2002.

[23] P. Palanque, M. Winckler, J.-F. Ladry, M. H. ter Beek, G. Faconti, and M. Massink. A formal approach supporting the comparative predictive assessment of the interruption-tolerance of interactive systems. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '09, pages 211–220, New York, NY, USA, 2009. ACM.

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

[25] C. Parnin and S. Rugaber. Resumption strategies for interrupted programming tasks. *Software Quality Control*, 19(1):5–34, Mar. 2011.

[26] T. J. Robertson, S. Prabhakararao, M. Burnett, C. Cook, J. R. Ruthruff, L. Beckwith, and A. Phalgune. Impact of interruption style on end-user debugging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 287–294, New York, NY, USA, 2004. ACM.

[27] D. D. Salvucci, N. A. Taatgen, and J. P. Borst. Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1819–1828, New York, NY, USA, 2009. ACM.

[28] A. M. Szóstek and P. Markopoulos. Factors defining face-to-face interruptions in the office environment. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1379–1384, New York, NY, USA, 2006. ACM.

[29] R. van Solingen, E. Berghout, and F. van Latum. Interrupts: Just a minute never is. *IEEE Softw.*, 15(5):97–103, Sept. 1998.