

Task Interruptions in Requirements Engineering: Reality versus Perceptions!

Zahra Shakeri Hossein Abad, Guenther Ruhe
Department of Computer Science
University of Calgary, Calgary, Canada
Email: {zshakeri, ruhe}@ucalgary.ca

Mike Bauer
Arcurve Inc.
Calgary, Canada
Email: mike.bauer@arcurve.com

Abstract—Task switching and interruptions are a daily reality in software development projects: developers switch between Requirements Engineering (RE), coding, testing, daily meetings, and other tasks. Task switching may increase productivity through increased information flow and effective time management. However, it might also cause a cognitive load to reorient the primary task, which accounts for the decrease in developers’ productivity and increases in errors. This cognitive load is even greater in cases of cognitively demanding tasks as the ones typical for RE activities. In this paper, to compare the reality of task switching in RE with the perception of developers, we conducted two studies: (i) a case study analysis on 5,076 recorded tasks of 19 developers and (ii) a survey of 25 developers. The results of our retrospective analysis show that in ALL of the cases that the disruptiveness of RE interruptions is statistically different from other software development tasks, RE related tasks are more vulnerable to interruptions compared to other task types. Moreover, we found that context switching, the priority of the interrupting task, and the interruption source and timing are key factors that impact RE interruptions. We also provided a set of RE task switching patterns along with recommendations for both practitioners and researchers. While the results of our retrospective analysis show that self-interruptions are more disruptive than external interruptions, developers have different perceptions about the disruptiveness of various sources of interruptions.

Index Terms—Requirements Engineering, Task Interruptions, Multitasking, Task Switching, Empirical Software Engineering

I. INTRODUCTION

More than ever, multitasking is thrust upon software engineers by commercial pressures, such as budgetary constraints; customer support models that value supporting old versions and therefore have ad-hoc resourcing demands; services-based business models that strive to reduce non-billable time; or human-resources situations such as illness of a key team member or vacant team positions. A task switching may be caused by a change in task priority, a question from a coworker, a scheduled meeting, or a task blockage resulting from an unavailable resource needed for development [1]. Regardless of the source, task switching imposes a cognitive load and can be detrimental to the primary task, particularly in cases where at least one cognitively demanding task is involved in the task switching process. However, this does not mean that task switching is a bad thing. To the contrary, sequential multitasking often allows us to perform tasks effectively and it may foster efficiency in certain circumstances [2] (e.g. knowledge transfer, managing the blocked tasks).

In the entire software lifecycle, Requirements Engineering (RE) is possibly the most data and communication intensive area [3], crossing many social and organizational boundaries. The high level of complexity and cognition in RE activities, along with the key role of this phase in the success of software development projects, raises this key research question for further investigation: “*How does task switching impact RE activities?*”. Over the last decade, a considerable amount of research into the cognitive aspects of software engineering (e.g. Section II-B) has been undertaken. For example, [4]–[6] aim to assist the RE process by reducing the cognitive load of requirements elicitation and communication. However, the body of research on RE is lacking in understanding the concepts and the disruptiveness of interruptions in RE activities. This understanding can help inform how requirements engineers’ productivity can be impacted by task switchings and what conditions make these interruptions more disruptive.

This paper reports on a mixed methods study of task switching and interruptions in the area of RE. We conducted a manual retrospective analysis on 5,076 recorded tasks of 19 developers to compare RE tasks with other software development tasks in terms of their vulnerability to interruptions; and to understand and explore how RE tasks are influenced by interruptions. Further, a survey of 25 professional software developers was conducted to understand developers’ perceptions and reasons behind task switchings and interruptions.

From both studies, we found that context switching, interruptions with a different priority, and afternoon interruptions make RE interruptions more disruptive. This paper makes the following contributions:

- 1) it presents the results of a manual retrospective analysis of 5,076 recorded tasks of 19 professional software developers, comparing RE tasks with other task types in terms of their vulnerability to interruptions, and specifically investigating the disruptiveness of RE interruptions,
- 2) it presents the results of a survey of 25 professional software developers, comparing the perceptions of developers to the repository analysis results,
- 3) it synthesizes the results of both studies and provides a set of “RE task switching” patterns and practical recommendations for practitioners by which they might better manage their RE task switchings and interruptions.

We provide background information about our study in Section II, followed by our Research Method (Section III). Our study design, including the data collection, preparation and analysis, is discussed in Section IV. We discuss the main contributions of our study and provide a set of recommendations related to each contribution in Section V. Limitations of this study are discussed in Section VI, followed by conclusion and research agenda in Section VII.

II. BACKGROUND

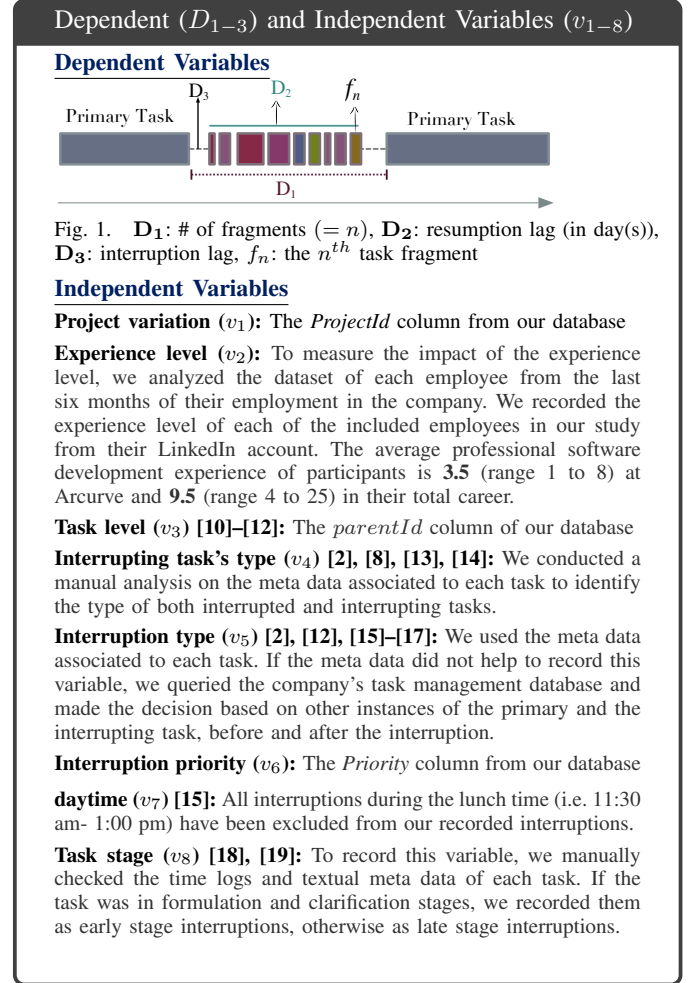
In this section, we first describe concepts related to task switching and interruption. Then, we review the related work, which addressed interruption analysis in software engineering.

A. Terminology

Interruptions are a form of task switching or sequential multitasking [2]. While there is some disagreement in the literature, there appears to be a general agreement, with plenty of theoretical evidence [2], [7], [8] about the cognitive cost of task switching, which appears on individual productivity. This theoretical evidence describes the **disruptiveness** of task switching and interruption in terms of the time cost and the cost they pose on developers' productivity. Memory-for-goals [7] is one of the major cognitive theories on interruptions, which explains switching to another task suspends the goal of the primary task and activates the secondary task goal. The time period between task switches is called **interruption lag** [1], [2], [9] (D_3 in Figure 1). In empirical studies, interruption lag is measured as the transitional interval between when a subject stops working on the primary task and when they start the secondary task. The timing of an interruption in respect to the primary task is a key aspect in the study of task interruptions [1], [2]. As discussed by Salvucci and Taatgen [2], a primary task can rehearse its problem state very briefly, usually for a few seconds. The longer the time between pausing and resuming the primary task, the more time required to reconstruct the problem state after the task has been resumed [8]. In this study, we define the **resumption lag** as the time (i.e. in day(s)) between interrupting and resuming the primary task (D_2 in Figure 1).

Complementing the theory of memory-for-goals, Salvucci and Taatgen [2] used the concept of **problem state** to explain the process of sequential task switching and the suspension and resumption processes. The *problem state* refers to the information required for performing a task, which can be utilized with only one task at a given time. What the current task is **-the goal-** and the information required for doing a task **-the problem state-** are maintained in distinct areas of the brain. This implies that the human's cognition only allows multiple goals to be active in their brain, not multiple problem states. Thus, task switchings which need to utilize the problem state at the same time contribute to a cognitive cost. However, there are some tasks during every software development project which are entirely reactive, such as answering an email, or a phone call, or when a manager pays a visit. In addition, there are some tasks that might need to utilize the problem

state resource but do not need to keep the information therein, such as daily stand-up meetings [2]. Thus, in this paper, we only consider task switchings as **interruptions** in which the suspended task needs to maintain the problem state. For instance, if a development task is interrupted by a meeting, we count it as an interruption but not for the reverse case.



An interruption initiated by the subject of the primary task to address a more critical task or a planned task from the backlog is called a **self-interruption**. Besides interrupting ourselves, an interruption can also be motivated by some external events in the environment (e.g. a colleague entering one's cubicle), which is called an **external interruption** [16]. Many researchers [2], [12], [15], [16] used this classification as a key factor to measure the effect and the time cost of interruptions.

Considering the limited cognitive flexibility of humans [20], nested task switchings causes mental congestion for keeping track of multiple states of tasks, which decays the goal of the primary task [15]. From existing interruption analysis studies and real-world examples [2], [15], it is clear that nested task switching causes a cognitive cost for reconstructing the problem state for every task switch which drives down the developers' productivity [21]. In this study, in addition to "interruption lag" and "resumption lag", we use the **number of task fragments** between pausing and resuming

the primary task as a factor to measure the disruptiveness of interruptions (D_1 in Figure 1).

B. Related Work

As discussed in the background section above, there are plenty of both practical and theoretical evidence from Human Computer Interactions (HCI), Psychology, and Computer Supported Cooperative Work (CSCW) that studied the impact of task switching and interruption on individual productivity (e.g. [2], [7], [8], [15]). In addition to the studies discussed in the background section, in this section, we mainly focus on related work that studied task interruptions in the area of software engineering. Vasilescu et al. [20] modelled the rate and breadth of developers' context switching behaviour and studied the effects of task switchings on developers' productivity. They found that the *rate* and the *breadth* (i.e. the number of projects) of task switching is an influential factor on individuals' productivity: developers who are involved in several projects generate more output than who are not (breadth). However, they found that frequent context switching during the course of a day has a negative impact on developers productivity (rate). They also surveyed developers to understand the main reasons for and perceptions of multitaskings in software development projects. Participants of this study describe the interrelationships and dependencies between projects as the most common reason for their task switchings. However, the results of this survey indicate that developers do not seem to be aware of the limits of multitasking (e.g. when to stop the task switching).

Meyer et al. [22] conducted a survey and an observational study to understand developers' perception of productive and unproductive work. While their observational data shows that participants performed significant task switching during a day, yet, they perceive their days as productive when they complete many or big tasks without significant interruptions or context switches. Further, Parnin and Rugaber [1] conducted a survey and a retrospective analysis on 86 programmers to understand the various strategies and coping mechanisms that developers need to manage interrupted programming tasks. They found that only a small percentage of interrupted programming tasks were resumed in less than a minute. Also, among all resumption strategies they proposed, an automatic tag cloud of links to recent source code, contextual reminders, lightweight annotation of a task, and task snapshots are the top four choices of developers who participated in these studies.

Chong and Siin [23] compared interruption patterns among paired and solo programmers. Their study indicates significant differences between the pair programmers and solo programmers in terms of the length, type, time, context of occurrence and strategies for handling work interruptions. Also, they found that a substantial number of interruptions during the workday are self-initiated and joint work may have potential support for interruption handling.

While the existing research provided a wealth of insight on task switching and interruptions, we could not find any study that investigated interruptions in the area of RE. Moreover, the comprehensiveness of our study in terms of the size of our

dataset and the number of dependent and independent variables makes it different from other investigations.

III. RESEARCH METHOD

To design our research methodology, we followed the guidelines provided by Runeson and Höst [24]. Our research methodology comprises of both exploratory (i.e. retrospective analysis) and descriptive (i.e. survey) approaches.

A. Context

We conducted this study in collaboration with Arcurve¹, Calgary's largest independent software services company. This collaboration greatly helped us to achieve our study goals for several reasons: (1) Arcurve is specialized in custom software development, integration, implementation and managed application services, supporting the full software lifecycle; (2) the workers are involved in multiple projects related to diverse industries (e.g. oil and gas, mining, agriculture, asset management, health services, supply-chain logistics, and education; and (3) Arcurve's software development and delivery processes follow industry-leading practices, with an emphasis on iterative development, collaboration, and incremental delivery.

B. Developing the Conceptual Framework

After developing the key concepts of our study, in this step, we developed the *conceptual framework*, including the key variables, factors and the potential relationships among them.

1) *Independent and Dependent Variables*: we identified a preliminary list of dependent and independent variables by conducting a comprehensive literature review on task switching and interruption analysis. These variables were pilot tested [25] on 7,770 recorded tasks of 10 employees to ensure the quality of our dataset and to identify the potential confounding variables, such as interruption source and type, experience level, and task stage. The datasets required for both studies were collected from Arcurve's task-based bug tracking and project management tool (i.e. Fogbugz²). The three dependent (D_{1-3}) and eight independent (v_{1-8}) variables of this study, the way we interpreted them in the course of our data analysis, and their corresponding literature references are listed in Figure 1.

2) *Exploratory Factor Analysis*: due to the complexity of interpreting all combinations of our dependent and independent variables (i.e. $8 \times 3 = 24$), we used the **Binary Exploratory Factor Analysis (BEFA)** approach. This method facilitates easier interpretation of the results by identifying the underlying dimensions of a dataset and classifying the variables based on these dimensions. To implement the BEFA approach, we manually analyzed 4,093 task logs (included in 5,076 task logs) of our dataset to explore 70 interruptions/task switchings per employee. We chose a cut-off of 0.32 for a statistically meaningful rotated factor loading, following the rule of thumb proposed by Tabachnick et al. [26]. Table I presents the Varimax-rotated matrix between the factors and

¹www.arcurve.com/

²<https://shop.fogcreek.com/FogBugz/>

TABLE I
FACTOR MATRIX SHOWING FACTOR LOADINGS ON THE VARIABLES

Variables (possible values)	Factor1	Factor2	Factor3
Project variation (same/different) (v_1)	0.50	0.27	-0.07
Employees' experience (less/more) (v_2)	0.81	-0.13	-0.14
Task Hierarchy (main/sub-task) (v_3)	-0.87	0.18	-0.46
Interruption type (self/external*) (v_4)	0.09	0.87	0.26
Task type (same/different) (v_5)	0.07	0.61	0.08
Priority difference (same/different) (v_6)	0.25	-0.38	-0.05
Interruption time (morning/afternoon) (v_7)	-0.10	-0.28	0.39
Task Stage (early/late) (v_8)	0.01	0.02	0.38

*The bold values present the value of each variable that has been used for implementing the BEFA method

variables after the factors have been sorted by their absolute loading values.

Looking at the factor loadings of the variables in this table, we observe that **Factor1** has high loadings, shaded in gray, on variables (v_{1-3}), which describes the *contextual* aspects of interruptions, such as other ongoing projects and tasks in the company, and the experience level of the employees. **Factor2** has high loadings on variables (v_{4-6}) which explain **interruption characteristics**, such as the type of the interrupting task, priority, and interruption source. Likewise, **Factor3** has high loadings on variables ($v_{3, v_{7-8}}$) and describes the **temporal** aspects of interruptions. To interpret and name this factor, we only used v_7 and v_8 as v_2 is assigned to Factor1.

C. Research Questions

Our study was guided with the following research questions:

RQ1 (RE vs other tasks): Regarding the disruptiveness of interruptions, is there a significant difference between RE and other software development activities in terms of the context, type, and timing of interruptions?

RQ2 (RE interruptions): How and to what extent RE interruptions are influenced by the context, type, and timing of interruptions?

RQ3 (Cross-factor analysis): How do cross-factor combinations of variables v_{1-8} impact RE interruptions?

IV. STUDY DESIGN

A. Data Collection and Preparation

During the four months of data collection and preparation, we hired three Research Assistants (RA) in the area of software engineering and trained them over the first two months to familiarize them with the data preparation process. After the first two months, we ran a pilot study on the prepared dataset and reported the results [25] to our industry partner. Following the feedback we got from this step, we revised our data extraction form and started data extraction and preparation phases from scratch. The data extraction form and a sample dataset collected for one employee are available on the website of the first author³. For each employee, we recorded 100 interruptions due to the high level of details in our data extraction form.

³<http://wcm.ualgary.ca/zshakeri/projects>

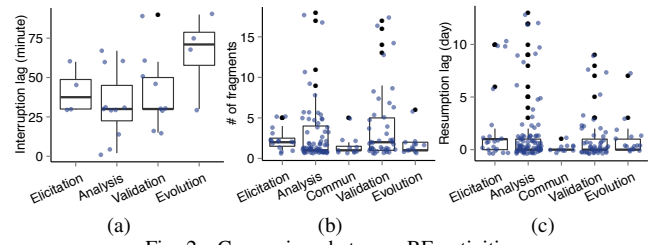


Fig. 2. Comparison between RE activities

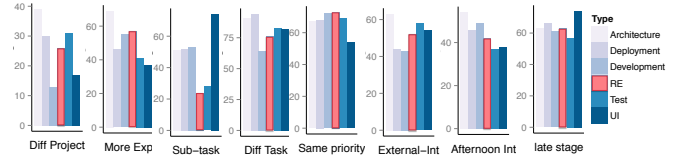


Fig. 3. RQ1- Comparison of frequency (percentage) of each variable between RE and other task types

B. Data Analysis Methods

To analyze and interpret the results of RQ1 and 2, we used **statistical hypothesis testing**, where the null hypotheses for each of these questions were formulated as follows:

- **[RQ1]** $H_0(1, v_i D_j, type_k) = < variable_i >$ does not make a significant difference between the $< disruptiveness_j >$ resulted from task interruption of RE and $< type_k >$ tasks.
- **[RQ2]** $H_0(2, v_i D_j) =$ In the context of RE task interruptions, $< variable_i >$ does not make a significant impact on $< disruptiveness_j >$.

Where $i \in \{1, 2, \dots, 8\}$ and $j = 1, 2, 3$. Moreover, to answer RQ3, we used the Pearson cross-factor correlation analysis along with statistical tests. We used $Q-Q$ plots [27] to assess the normality of the distribution of our dataset. As we could not confirm that the underlying populations follow a normal distribution, we used the *Kruskal-Wallis* test to examine the null hypotheses posed for RQ1-3. To properly address RQ1, if the overall statistically significant difference in group means was shown by the *Kruskal-Wallis* test, we used the *Kruskal-Wallis post-hoc* test for pairwise multiple comparisons to examine each pair separately. To implement our statistical tests, we used the **PMCMR**⁴ package of **R**.

C. User Survey

Complementary to our retrospective analysis, we sent an online survey to 50 employees at Arcurve to understand developers' perceptions of task switchings and interruptions. We used Survey Monkey⁵ to design the online survey and collect responses. The survey had 10 main questions (18 sub-questions) including multiple choice, Likert scale, and open-ended questions. We asked participants about their job roles, development experience in general, and the interruption factors which influence their productivity⁶. We received 25 (50% response rate) responses. The average professional software development experience of participants was 12.5 (± 10.5)

⁴<https://cran.r-project.org/web/packages/PMCMR/PMCMR.pdf>

⁵<http://www.surveymonkey.com>

⁶<https://wcm.ualgary.ca/zshakeri/files/zshakeri/re-interruption-survey.pdf>

TABLE II

RQ1- COMPARISON BETWEEN RE AND OTHER SOFTWARE DEVELOPMENT TASKS IN TERMS OF THE DISRUPTIVENESS OF VARIOUS FACTORS

Pairs	Interruption Context									Interruption Type									Interruption Time					
	different project (v_1)			less experience (v_2)			sub-task (v_3)			different type (v_4)			self-interruption (v_5)			different priority (v_6)			afternoon Int. (v_7)			late stage (v_8)		
	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
Kruskal-Wallis	0.003	0.01	0.32	0.1	0.01	0.01	1e-4	4e-4	0.03	1e-3	3e-4	0.04	0.01	0.003	0.04	4e-6	3e-5	0.05	0.01	0.04	0.01	0.03	0.01	0.2
RE-Architecture	0.04	0.02*	0.3	0.1	0.02	0.2	4e-5	1e-4	0.2	0.01	0.01	0.8	0.17	0.1	0.7	0.05*	0.04*	0.7	0.03	0.02	0.4	0.05*	0.1	0.3
RE-Development	0.1	0.1	0.8	0.2	0.1	0.001	7e-5	0.001	0.01	0.01	0.03	0.03	0.2	0.2	0.3	5e-5	2e-5	0.5	0.1	0.05*	0.03	0.5	0.3	0.3
RE-Test	4e-5	3e-5	0.3	0.1	0.03*	0.1	6e-5	0.001	0.1	4e-4	1e-5	0.04*	0.01*	0.01	0.1	0.002	0.002	0.2	0.07	0.01*	0.7	0.1	0.03	0.6
RE-UI Design	0.07	0.08	0.3	0.1	0.2	0.1	1e-6	2e-4	0.4	0.01	0.03	0.01*	0.1	0.1	0.06	0.001	0.003	0.05	0.02*	0.01*	0.06	0.06	0.06	0.1
RE-Deployment	0.3	0.6	0.01*	0.17	0.2	0.04	0.01	0.01	0.02	0.2	0.6	0.01	0.8	0.7	0.05	0.4	0.5	0.1	0.7	0.5	0.06	0.03	0.3	0.1

*: The p-value of the alternative value of the corresponding variable. For instance, if ($v_2 = \text{more exp}$), there will be a significant difference between RE-Test in terms of D_2

years. To analyze the results of the survey, we (i) iterated through the open-ended responses using the grounded theory approach [28], to interpret them and to identify participants' perceptions about productivity and their suggestions about the interruption tool; (ii) applied descriptive statistics and Spearman's rank correlation test methods to quantitatively investigate the survey data. This study has been approved by the University of Calgary Conjoint Faculties Research Ethics Board (CFREB⁷).

V. RESULTS

This section summarizes the key findings of the two-stages research, along with detailed discussion and proposed recommendations associated with each finding.

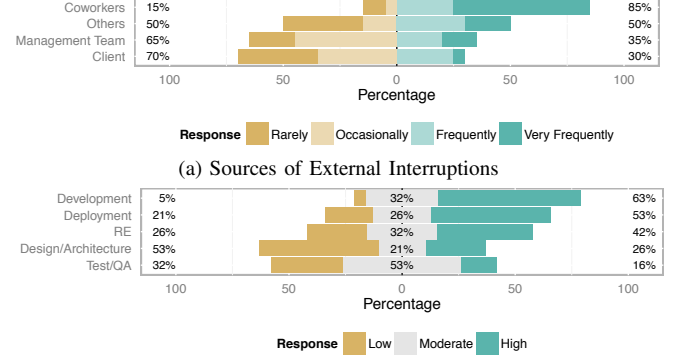
A. Preliminary Results

RE activities: According to the Kruskal-Wallis test, comparison between RE activities showed no statistically significant differences in the disruptiveness factors we defined in Section II-A (D_{1-3}). Thus, in the rest of this section, we consider all RE activities as RE tasks. However, the results of the related descriptive statistics (Figure 2) show that, to some extent, requirements elicitation is more vulnerable to task interruptions compared to other RE activities, except for the interruption lag for requirements evolution tasks.

A productive task: We asked each participant how they define and measure their "productivity" on a certain task. The participants predominantly described their productivity of a certain task as the rate of output (e.g. the number of requirements completed), as in: "I define productivity of a certain task by being able to complete as much of it as possible in the shortest amount of time accurately, and with enough detail to not have to revisit it". 13 (54%) participants explicitly stated that they measure their productivity as an effort spent against the expected effort. However, 4 (31%) of these respondents stated that it is not possible to accurately estimate a task's effort and it is only possible by comparing it to previous, similar tasks.

B. RQ1- RE versus Other Software Development Tasks

1) *Overview of Comparisons:* Figure 3 illustrates the comparisons between RE and other tasks in terms of the frequency of all independent variables of this study. We observe from this figure that, compared to other types of tasks, the majority of RE and testing interruptions occur to high-level tasks.



(b) The vulnerability of different task types to interruptions

Fig. 4. Interruption type and source

Moreover, among all RE interruptions; 24% were sub-tasks, 26% were caused by a different project, 57% were carried out by employees with a higher level of experience, 76% were caused by a different task type, 73% were caused by a task with the same priority, 53% were motivated by some external events, 42% occurred in the afternoon, and 63% occurred in later stages of their development.

Finding 1-1: We further investigated the main sources of interruptions for RE tasks. Among 53% of external RE interruptions, 84% are caused by coworkers, 6% by clients, 6% by the management team, and 4% by other reasons (e.g. answering a phone call, or personal affairs).

Discussion 1-1: Our survey data supports finding 1-1, as 85% of participants chose *coworkers* as the most frequent source of their external interruptions (Figure 4a).

2) *Hypothesis Testing:* to answer this RQ, we posed 144 null hypotheses following the template presented in Section IV-B. Table II presents the p-value for each of these tests. Out of 120 Kruskal-Wallis post-hoc tests, 41 tests were rejected, colored in gray, among which 19 (46%) rejected tests are related to interruption characteristics (i.e. v_{4-6}), 17 (41.5%) tests are related to the interruption context (i.e. v_{1-3}), and only 5 (12%) of the rejected tests are related to the temporal aspects of interruptions (i.e. v_{7-8}). All of the rejected null hypothesis tests for RQ1 are illustrated in Figures 5.

Finding 1-2: In all (100%) of rejected hypotheses, RE tasks are more vulnerable to interruptions compared to other tasks.

Discussion 1-2: To further investigate this finding, complementary to the retrospective analysis, we asked the survey respondents to, regardless of the values of the independent variables, rank the vulnerability of main development tasks to interruptions. By vulnerability, we mean the negative im-

⁷<http://www.ucalgary.ca/research/researchers/ethics-compliance/cfreb>

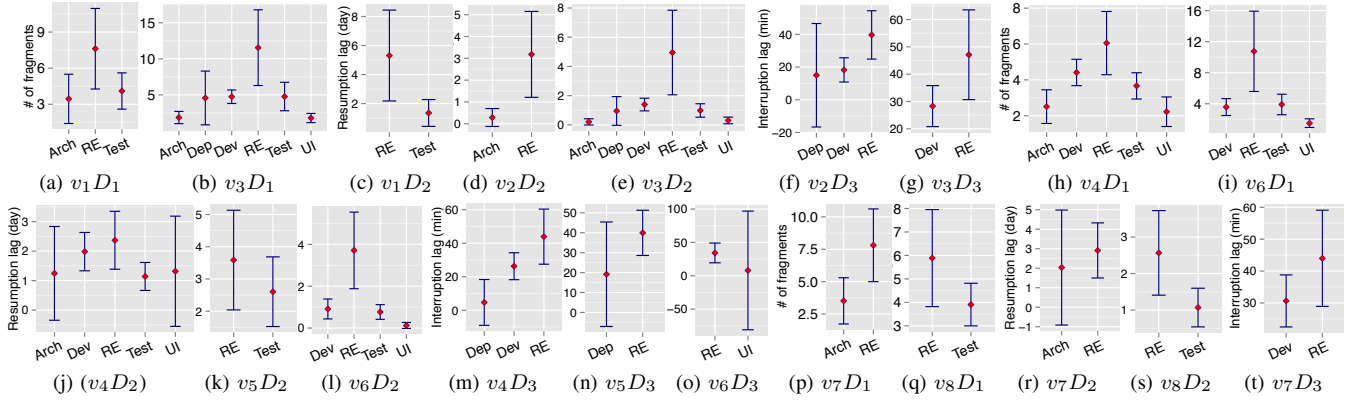


Fig. 5. 95% confidence interval of sample means for disruptiveness of interruption characteristics in RE tasks versus other task types

pect of task switching on their productivity after resuming the primary task. As illustrated in Figure 4b, although 11 (43%) participants chose RE tasks as highly vulnerable to interruptions, 16 (63%), and 13 (53%) found development and deployment tasks more vulnerable. This variation could be because participants answered this question according to their current job function. For example, a participant whose primary job function is **RE** stated that: “I have been in situations that I have gotten requirements confused between the two tasks”. Similarly, a **tester** stated that: “I find RE is more abstract and different contexts help ensure a well thought out approach while creating a specific test bed for a test scenario takes more area specific focus or is state dependent and may have to be restarted”. Moreover, we used different scales for these comparisons. While the retrospective analysis compares the impact of interruptions between RE tasks and other task types for specific values of our independent variables, the survey investigates these comparisons on a larger-scale, regardless of the values of the independent variables.

Recommendation 1: We propose the recommendation associated with this finding as a set of comparison patterns:

⟨ RE-type_k ⟩ Comparison Patterns

Comparison Patterns: ⟨ (v_iD_j, RE - type_k) ⟩

Table II lists significantly different pairs and Figure 5 presents the details of these comparisons. Moreover, Figure 6 illustrates the percentage frequency of each pair of (*interrupting*, *interrupted*) tasks for various task types. For instance, most of the RE tasks are interrupted by *development* (25.7%), *PM* (25.1%) and *RE* (18.8%) tasks, respectively. Interestingly, most of the interruptions in *architecture* (26.6%) and *UI* (39.1%) are caused by RE tasks.

Interrupting Tasks	UI	Test	RE	PM	Develop	Deploy	Arch
UI	0	2	2	2.1	1.2	12.5	
Test	6.2	8	8.8	14.1	23.8	6.2	
RE	26.6	14	9.6	18.8	10.4	39.1	
PM	26.6	28	21.8	25.1	37.6	15.6	
Develop	4.7	4	8.5	7.9	3.5	6.2	
Deploy	26.6	38	43.3	25.7	21	17.2	
Arch	3.1	4	2.6	1.6	0.7	1.6	
Interrupted Tasks	6.2	2	3.3	4.7	1.7	1.6	

Fig. 6. RQ1- percentage of (interrupting, interrupted) task pairs

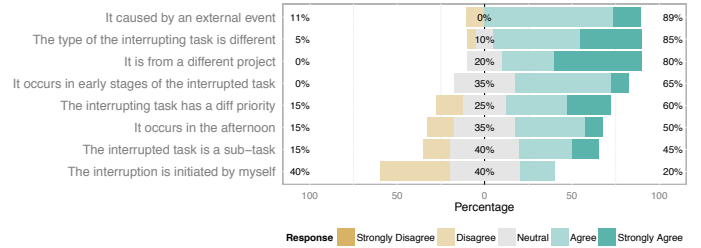


Fig. 7. Perceived negative impact of interruptions on RE tasks' productivity

C. RQ2- RE and Interruption Characteristics

To answer this RQ, we posed 24 null hypotheses following the template presented in section IV-B. We also asked survey participants to indicate, using a five-point Likert-type scale, their agreement to a number of variables of our study on their productivity after RE interruptions (Figure 7). Table III presents the results of the Kruskal-Wallis tests we ran to verify all hypotheses we posed for this RQ. Moreover, we found that none of the independent variables considered in our study significantly impact the **interruption lag** (D_3). In the rest of this section, we discuss the key findings and recommendations from analyzing this RQ.

Finding 2-1 (interruption context): According to the Kruskal-Wallis test results, in the context of RE interruptions, project variation ($H_0(2, v_1D_1)$, $H_0(2, v_1D_2)$), and the hierarchy level of the interrupted task ($H_0(2, v_3D_1)$), ($H_0(2, v_3D_2)$) make a significant impact on the number of fragments and the resumption lag. Moreover, the experience level of a developer makes a significant impact on the number of fragments resulted from these interruptions ($H_0(2, v_2D_1)$).

Discussion 2-1: While switching projects allows developers to use their time more efficiently, specifically in situations where they are blocked in one project, and provides them with opportunities for learning and knowledge transfer [20], switching context comes at a cost. The results of our retrospective analysis also reveal that context switching increases the cognitive cost of task switching by increasing the number of task fragments and the resumption lag (Figures 8 (a) and (g)). Our survey data also supports this finding, as 20

TABLE III
RQ2- THE RESULTS OF NULL HYPOTHESIS TESTING FOR RQ2

Interruption Context									Interruption Type						Interruption Time								
project variation (v_1)			experience (v_2)			hierarchy (v_3)			type (v_4)			self/external (v_5)			priority (v_6)			daytime (v_7)			task stage (v_8)		
D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
0.001	0.001	0.5	0.03	0.08	0.8	0.001	0.02	0.1	0.66	0.7	0.37	0.0003	1e-4	0.37	0.02	0.01	0.3	0.001	0.001	0.3	0.6	0.6	0.21

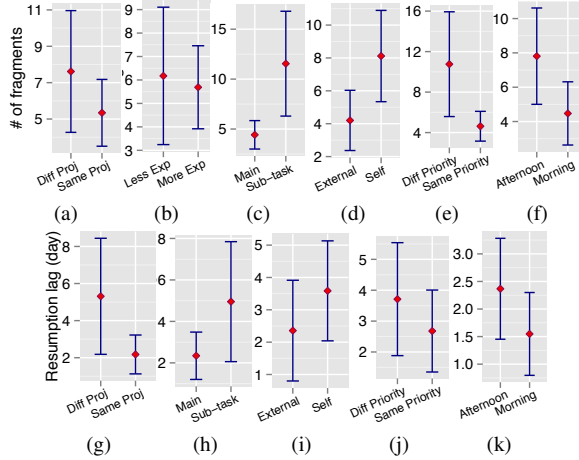


Fig. 8. RQ2- 95% confidence interval of sample means for disruptiveness of interruption characteristics in RE tasks

(80%) participants agreed that context switching negatively impacts their productivity; “*Switching gears completely affects my productivity more rather than just changing topics on the same project*”. Likewise, a recent study [22] concluded that context switching generally reduces the productivity of software practitioners.

Also, interruptions coinciding in low-level sub-tasks, as shown in Figures 8 (c) and (h), are more disruptive. This was reflected in a greater number of fragments and resumption lag after switching a sub-task rather than switching a main task. The most relevant study to this finding is the one by Salvucci et al. [12], where they discussed the level of the interrupted task in terms of the complexity of the task’s problem state (discussed in Section II-A) and the mental workload required for performing a task. They stated that the complexity of the problem state associated to a sub-task is higher than a main task due to having additional memory chunks. Thus, resuming lower-level tasks (e.g. requirement clarification) is more error prone and requires more effort and time. In addition, only 4 (15%) participants disagreed about the negative impact of interrupting sub-tasks, as in: “*minimal impact. Ideally, I am focused on the bigger picture*”. Looking at Figure 8 (b), we observe that interruptees with less experience resume their primary task after performing more task fragments. While our data population did not show a significant impact of experience on the length of resumption lag, we still believe that interrupting developers with less experience might have a negative impact on their productivity after resuming the task. This can be reflected in the cognitive cost of the greater number of fragments.

Recommendation 2-1 (a): If an RE task needs to be switched for some reasons like there is a lack of information, the interruptee is bored, or the knowledge of the interruptee is required by other teams, we recommend interrupters to

possibly ask their requests from coworkers who are working on the same project as they are. Furthermore, we recommend interruptees in cases of self-interruptions to switch to a task from the same project that has a less cognitive cost of task switching.

Recommendation 2-1 (b): Taken together, interruptions to higher-level tasks are more desirable since they minimize cognitive costs of task switching and thus decrease the number of task fragments, resumption time and the potential for resumption errors. We propose that it might be more efficient to complete a low-level RE task before the switch.

Recommendation 2-1 (c): In the context of interrupting or switching an RE task, to determine who to ask for help or more information about other tasks, we propose that it can be more efficient if interrupters ask their question to more experienced interruptees. To support this recommendation, a recent study [20] on task switching in software development teams, discussed the positive effects of knowledge transfers on productivity, which implies the value of experience as a source of knowledge and learning.

Finding 2-2 (interruption characteristics): According to the Kruskal-Wallis test results, in the context of RE task switchings and interruptions, the source of the interruption ($H_0(2, v_5 D_1), H_0(2, v_5 D_2)$), and the interrupted task priority ($H_0(2, v_6 D_1), H_0(2, v_6 D_2)$) make a significant impact on the number of fragments and the resumption lag.

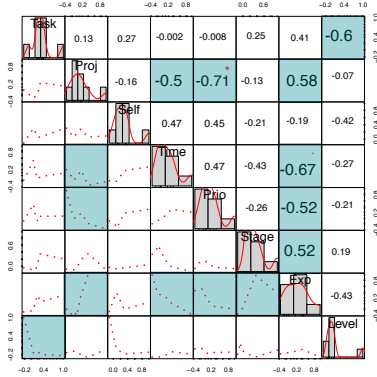
Discussion 2-2: As shown in Figure 8 (d), and (i), self-interruptions in the context of RE interruptions cause more task fragments and longer resumption lags. This finding is in line with the results of a controlled experiment that Katidioti et al. [17] conducted to contrast external interruptions with self-interruptions. Their study shows that self-interruptions introduce the extra cost of the decision. If these extra costs do not lead to a substantive reduction in the other costs of interruption, self-interruptions are more harmful than external interruptions. This is because they lead to greater resumption lags and have a negative impact on the duration of the interrupted task. However, our survey participants were not very positive about the effects of external interruptions on their productivity: only 9% agreed that self-interruptions negatively impacts their productivity, while this number for external interruptions is 89% (Figure 7). Moreover, 20 (80%) participants indicated that less than 30% of their interruptions are self-interruptions. To further investigate this difference, we hypothesized that there might be a correlation between the number of projects that participants are involved in and their response to this question. A Spearman’s rank correlation test shows no correlation between the participants’ agreement to the impact of self/external interruptions and the “# of projects”

TABLE IV

RQ3- CROSS-FACTOR ANALYSIS FOR RE INTERRUPTIONS (KRUSKAL-WALLIS TESTS)

	same(s)/ different(d) project (v_1)			sub/main task (v_2)			same (sp)/diff (dp) priority (v_3)			morning (m)/afternoon (a) (v_4)			early (e)/late (l) stage (v_5)		
	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
same(s)/ different(d) Project							0.001 (dp*)	0.002 (dp)	0.03 (dp)	0.01 (m)	0.048 (m)	X			
more(me)/ less(le) Experience	X	X	X				X	X	X	0.005(s)	0.01(s)	X			
same(st)/ different(dt) Task type	0.02 (me)	0.01 (le)	X				0.02 (me)	X	X	0.003 (le)	0.02 (le)	X	X	0.03 (e)	X
				X	X	X									
				0.001(dt)	0.01(dt)	X									

*: the recorded p -value is only related to the specific values mentioned in parentheses (i.e. the *controlled* variables that are used in RE Task Switching Patterns)

Fig. 9. Cross-correlation between v_{1-8} for interrupted RE tasks

they are involved in. Also, there is no correlation between the size of the primary project (in people) of participants and their agreement to these statements. However, there was a strong correlation between the participants' agreement to the negative impact of \langle context switching, task type \rangle , and self-interruption (context switching: $\rho = 0.61$, $p = 0.01$; task type: $\rho = 0.5$, $p = 0.03$). In addition, the Spearman's rank correlation test shows no correlation between the participants' agreement to the negative impact of external interruptions, the negative impact of context switching and type of the interrupting task.

Interestingly, according to the Kruskal-Wallis test results, the type of the interrupting task (v_4), does not make any significant impact on the disruptiveness of interruptions in RE tasks (p -value > 0.05). However, 21(85%) survey participants agreed that their productivity will be negatively impacted if the interrupting task is a task with a different type (e.g. development, test); "Tool/environment switching seems to burn more time than I usually account for". We aim to further investigate this variable in our planned replication study.

Recommendation 2-2: We propose that if an RE task needs to be switched, it can be more efficient if this switch is triggered by an external source such as coworkers or the management team, instead of leaving the decision to the interruptee.

Finding 2-3: according to the Kruskal-Wallis test results, in the context of RE interruptions, daytime ($H_0(2, v_7D_1)$, $H_0(2, v_7D_2)$) makes a significant impact on the number of fragments and the resumption lag.

Discussion 2-3: As shown in Figure 8 (f), and (k), in the context of RE interruptions, afternoon interruptions cause more task fragments and longer resumption lags. Of all recorded RE interruptions, 57% were interrupted in the morning among which 63% were resumed on the same day, while this number for afternoon interruptions is 39%. Similarly, our results show that about half (56%) of the afternoon interrup-

tions are self-initiated. This might be due to the additional pressure for completing more tasks before the end of the day. Further, 12 (50%) survey participants agreed and only 4 (15%) disagreed about the negative impact of afternoon interruptions, as in: "if I get interrupted later in the day I tend to use the existing interruption to get a coffee as opposed to waiting until I finish a current task". Also, a Spearman's rank correlation test shows a strong correlation between participants' agreement to the negative impact of afternoon interruptions and the size of the project they are involved in ($\rho = 0.5$, $p = 0.04$).

Although none of the survey participants disagreed about the negative impact of early-stage interruptions on their productivity (i.e. 65% (strongly) agreed, 35% neutral), our retrospective analysis does not show any significant impact of this variable (v_8) on RE interruptions (i.e. all p -values > 0.05), as in: "the early thought process in new development are key. If I swap in the middle of this, it will take longer to recreate the thought process since it is less developed mentally". On the other hand, several studies (e.g. [18], [19]) measured the cognitive cost of interruptions by looking at the point at which a task is interrupted. They found that interrupting a task during middle or end-task interruption points will result in longer interruption and resumption time. This implies the need for a further investigation on the impact of interruption point on RE interruptions.

Recommendation 2-3: Given the negative impact of self-interruptions on interrupting RE tasks (Finding 2-2) and the cost of afternoon interruptions, we recommend practitioners to minimize voluntary task switchings in later times of a day.

D. RQ3- Cross-factor Analysis

To answer this RQ, we conducted a cross-factor analysis of all of the variables listed in Table I in the context of RE task interruptions. As the first step, we performed a cross-factor correlation analysis among all of these variables (Figure 9) and excluded all the unsubstantial cross-factor combinations (i.e. $r < 0.5$ [29]). Out of all possible 28 combinations, we selected 7 combinations for further analysis, which are highlighted in Figure 9. As the next step, we tested the significant impact of each of these combinations on RE tasks (Table IV). To properly interpret the cross-factor impact of these combinations, we used a combination of Violin, Scatter, and Box plots for each of these combinations (e.g. Fig 10).

Finding 3 and Recommendation 3: We formulate the main findings of this RQ in forms of some **RE task switching patterns**. In the rest of this section, we briefly list (Table IV and Figure 9) and discuss these patterns.

Discussion 3: 10 (47%) participants agreed to the negative impact of both interruptions with different type, and interrupting sub-tasks [P4]. 11 (79%) and 10 (71%) of more experienced participants agreed with the negative impact of early-stage interruptions [P6] and context switching [P3], respectively. A participant with 12 years of experience stated: “When formulating initial ideas and solutions, I find when I am grouping multiple thoughts and they haven’t formed a concrete assertion, if interrupted, I have to start over. Once I have a generalized thought, I tend to resume quickly.”

RE Task Switching Patterns

$$\langle V_{ctl}, V_{ind}, Val_{dis}, D_{1-i} \rangle \quad i \in \{1, 2, 3\}$$

Where V_{ctl} and V_{ind} , denote the controlled and independent variables and Val_{dis} represents the value of the independent variable which makes an RE task switching more disruptive. Moreover, D_i represents the disruptiveness measures.

Example: $\langle (same\ project, daytime, afternoon), D_{1-2} \rangle$ states that when switching RE tasks on the same project, “daytime” makes a significant impact on the disruptiveness of these interruptions, and afternoon interruptions have a greater number of fragments and resumption lag comparing to task switchings that occur during the morning. In the case of context switching, daytime does not make a significant impact on the disruptiveness of RE task switchings [P1].

Patterns

- [P1] $\langle (same\ project, daytime, afternoon), D_{1-2} \rangle$
- [P2] $\langle (morning, project\ variation, different\ project), D_{1-2} \rangle$
- [P3] $\langle (more\ exp, project\ variation, different\ project), D_1 \rangle$
- [P4] $\langle (different\ type, task\ level, sub-task), D_{1-2} \rangle$
- [P5] $\langle (same\ priority, project\ variation, diff\ project), D_{1-3} \rangle$
- [P6] $\langle (early\ stage, experience, more\ experience), D_2 \rangle$
- [P7] $\langle (less\ experience, project\ variation, diff\ project), D_2 \rangle$
- [P8] $\langle (morning, experience, more\ experience), D_1 \rangle$
- [P9] $\langle (more\ experience, priority, different\ priority), D_1 \rangle$
- [P10] $\langle (less\ experience, daytime, afternoon), D_{1-2} \rangle$

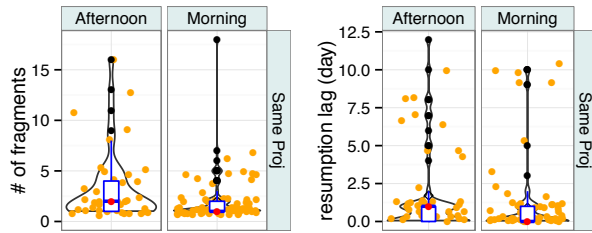


Fig. 10. [P1] $\langle (same\ project, daytime, afternoon), D_{1-2} \rangle$

Likewise, 10 (71%) experienced participants agreed that interruptions with a different priority negatively impacts their productivity on the primary task [P9]. 15 (75%) participants agreed that both context switching and priority difference negatively impacts their productivity [P5]; “depends on the task. It helps that it is the same app but if the tasks are from very different epics than it can have a negative effect”. Regardless of experience, 20 (80%) participants agreed about the negative impact of context switching [P3, P7]. The only difference between these two patterns is their disruptiveness factor (D). 3 (60%) less-experienced participants agreed on

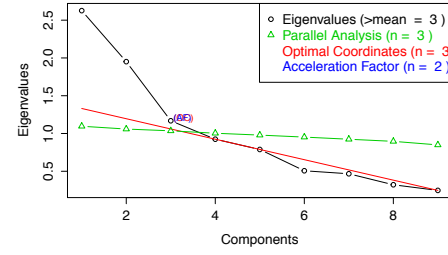


Fig. 11. Number of Factors (Scree Test)

the negative impact of afternoon interruptions: “I am more productive in the morning when no one is at the office (i.e. minimal distractions with no meetings)” [P10].

VI. THREATS TO VALIDITY

Conducting both studies on a particular company might question the generalizability of our findings. We attempted to address this threat by implementing the study on a fairly large dataset including various projects from different business domains and employees from different levels of experience. Moreover, applying a mixed method approach enabled us to triangulate findings obtained through the retrospective analysis with the results from the follow-up survey. Additionally, the employees’ dataset we used in our retrospective analysis was recorded during their normal real-world work, not during an experimental exercise.

Our data collection and preparation pose another threat to the validity of our results, since identifying the type of interruptions (i.e. self and external) and the type of the tasks in general (e.g. RE, architecture, development, test, ...) is not straightforward. To mitigate these risks, before the primary data collection phase, we pilot tested our data extraction method and the results of this stage have been validated by our industry partner. Moreover, the dataset associated with each employee was reviewed by at least two of the hired RA’s and the first author of the paper. To evaluate the reliability of our decisions for independent variables that have been recorded manually, we used the Cohens Kappa statistic, which calculates the degree of agreement between two raters. The calculated Kappa value was 0.96, which shows significant agreement, as stated by Landis and Koch [30].

Furthermore, the robustness and reliability of the measures (i.e. D_1, D_2) we used to study the disruptiveness of interruptions may affect the findings of this study. To define these underlying factors, we utilized the results of similar studies on interruption analysis (Section II-A). In addition, to ensure the reliability of these factors, we asked our survey participants about the negative impact of “higher number of task fragments (D_1)” and “longer resumption lags (D_2)” on their productivity. 23 (91%) and 21 (85%) participants agreed about the negative impact of both factors on their productivity after resuming the primary task. Further, to determine the number of factors we first used the *Scree test* (Figure 11) in conjunction with eigenvalues. Then, we manually checked the result of BEFA for the other number of factors (i.e. $n = 2, 4, 5, 6$). To ensure the reliability of the explored factors in our study and to minimize the impact of our data collection errors on these

factors, we used a fairly large dataset (i.e. 1,430 interrupted tasks) for implementing the BEFA method, which is in the range of “ ≥ 1000 (*excellent*)”, based on the sample size adequacy scale provided by Comerly and Lee [31].

VII. CONCLUSION AND RESEARCH AGENDA

To investigate the concepts and the disruptiveness of task switching and interruptions in the area of RE, we conducted two studies, a manual retrospective analysis on 5,076 recorded tasks of 19 employees and a survey with 25 professional software developers. From both studies, we found that context switching, the priority of the interrupting task, the interruption source and timing are the key factors that negatively impact RE interruptions. We also provided a set of RE task switching patterns along with practical recommendations for practitioners in the area of RE and software development in general.

Although our investigation provides new insights on RE task switchings and interruptions, our results raise additional research questions. Using visualization techniques as visual aids and tool support to control context switches and interruptions is one of these notable unanswered questions. For instance, *filling gaps* and *continuity* in the storytelling approach, as we discussed in our previous work [32] (presented at RE 2016), increases developers’ awareness of their transition between various tasks and may help developers in retrospection on their own productivity. We also asked our survey participants about the main features of a tool which would help them to have less disruptive interruptions. The majority of participants opined that the ability to record the next step of the primary task before switching to the new task should be the key feature of this tool, as in: “*It should allow me to make a quick note on what the next step was before the interruption so I could read it back and instantly know what the next step is*”. A stopwatch, timeline of a task, and possibly a historical view of the primary task are other features of a tool suggested by survey participants.

As our future work, we aim to replicate this study in other contexts and to extend our investigation by looking closer at the impact on productivity and projects’ success as dependent variables. In addition, we plan to further investigate the task type by applying requirements classification techniques (as in [33]), and consider the task size as another independent variable, as suggested by our industry partner. Moreover, measuring the cognitive load of interruptions and proposing visualization techniques [34] to reduce the cognitive demands of task interruptions are other goals for replicating this study.

VIII. ACKNOWLEDGEMENT

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant 250343-12 and by an Engage Grant. Also, we sincerely thank Arcurve Inc. for providing their comprehensive task management dataset and all Arcurve employees for their valuable input to our research and willingness to cooperate.

REFERENCES

- [1] C. Parnin and S. Rugaber, “Resumption strategies for interrupted programming tasks,” *Software Quality Journal*, vol. 19, no. 1, pp. 5–34, 2011.
- [2] D. D. Salvucci and N. A. Taatgen, *The multitasking mind*. Oxford University Press, 2010.
- [3] Z. S. H. Abad and G. Ruhe, “Using Real Options to Manage Technical Debt in Requirements Engineering,” in *IEEE 23rd International Requirements Engineering Conference (RE’15)*, 2015, pp. 230–235.
- [4] O. Karras, S. Kiesling, and K. Schneider, “Supporting Requirements Elicitation by Tool-Supported Video Analysis,” in *IEEE 24th International Requirements Engineering Conference (RE)*, 2016, pp. 146–155.
- [5] P. Ghazi and M. Glinz, “An Exploratory Study on User Interaction Challenges When Handling Interconnected Requirements Artifacts of Various Sizes,” in *IEEE 24th International Requirements Engineering Conference (RE)*, 2016, pp. 76–85.
- [6] P. Ghazi, Z. S. H. Abad, and M. Glinz, “Choosing Requirements for Experimentation with User Interfaces of Requirements Modeling Tools,” in *25th IEEE International Requirements Engineering Conference (RE’17)*, 2017.
- [7] E. M. Altmann and J. Trafton, “Memory for Goals: An Activation-based Model,” *Cognitive Science*, vol. 26, no. 1, pp. 39 – 83, 2002.
- [8] T. Gillie and D. Broadbent, “What makes interruptions disruptive? a study of length, similarity, and complexity,” *Psychological Research*, vol. 50, no. 4, pp. 243–250, 1989.
- [9] E. M. Altmann and J. G. Trafton, “Task interruption: Resumption lag and the role of cues,” DTIC Document, Tech. Rep., 2004.
- [10] D. D. Salvucci and P. Bogunovich, “Multitasking and monotasking: The effects of mental workload on deferred task interruptions,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 85–88.
- [11] C. Speier, I. Vessey, and J. S. Valacich, “The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance,” *Decision Sciences*, vol. 34, no. 4, pp. 771–797, 2003.
- [12] 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*, ser. CHI ’09. ACM, 2009, pp. 1819–1828.
- [13] E. Cutrell, M. Czerwinski, and E. Horvitz, “Notification, disruption, and memory: Effects of messaging interruptions on memory and performance.” IOS Press, 2001, pp. 263–269.
- [14] S. M. Hess and M. C. Detweiler, “Training to Reduce the Disruptive Effects of Interruptions,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 38, no. 18. SAGE Publications, 1994, pp. 1173–1177.
- [15] G. Mark, V. M. Gonzalez, and J. Harris, “No task left behind?: Examining the nature of fragmented work,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’05. ACM, 2005, pp. 321–330.
- [16] V. M. González and G. Mark, ““constant, constant, multi-tasking craziness”: Managing multiple working spheres,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’04. ACM, 2004, pp. 113–120.
- [17] I. Katidioti, J. P. Borst, and N. A. Taatgen, “What happens when we switch tasks: Pupil dilation in multitasking,” *Journal of experimental psychology: applied*, vol. 20, no. 4, p. 380, 2014.
- [18] M. Czerwinski, E. Cutrell, and E. Horvitz, “Instant messaging: Effects of relevance and timing,” in *People and computers XIV: Proceedings of HCI*, vol. 2, 2000, pp. 71–76.
- [19] C. A. Monk, D. A. Boehm-Davis, and J. G. Trafton, “The attentional costs of interrupting task performance at various stages,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, no. 22. SAGE Publications, 2002, pp. 1824–1828.
- [20] B. Vasilescu, K. Blincoe, Q. Xuan, C. Casalnuovo, D. Damian, P. Devanbu, and V. Filkov, “The Sky is Not the Limit: Multitasking Across GitHub Projects,” in *Proceedings of the 38th International Conference on Software Engineering*. ACM, 2016, pp. 994–1005.
- [21] H. Sanchez, R. Robbes, and V. M. Gonzalez, “An empirical study of work fragmentation in software evolution tasks,” in *IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering*, 2015, pp. 251–260.

- [22] A. N. Meyer, T. Fritz, G. C. Murphy, and T. Zimmermann, "Software Developers' Perceptions of Productivity," in *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, ser. FSE 2014. ACM, 2014, pp. 19–29.
- [23] 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*. ACM, 2006, pp. 29–38.
- [24] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2008.
- [25] Z. S. H. Abad, G. Ruhe, and M. Bauer, "Understanding Task Interruptions in Service Oriented Software Development Projects: An Exploratory Study," in *Proceedings of the 4th International Workshop on Software Engineering Research and Industrial Practice*, ser. SER&IP '17. IEEE Press, 2017, pp. 34–40.
- [26] B. G. Tabachnick et al, "Using multivariate statistics," 2001.
- [27] M. B. Wilk and R. Gnanadesikan, "Probability Plotting Methods for the Analysis of Data," *Biometrika*, vol. 55, no. 1, pp. 1–17, 1968.
- [28] A. Strauss and J. M. Corbin, *Grounded Theory in practice*. Sage, 1997.
- [29] D. E. Hinkle, W. Wiersma, and S. G. Jurs, "Applied Statistics for the Behavioral Sciences," 2003.
- [30] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.
- [31] A. L. Comrey and H. B. Lee, *A first course in factor analysis*. Psychology Press, 2013.
- [32] Z. S. H. Abad, M. Noaen, and G. Ruhe, "Requirements Engineering Visualization: A Systematic Literature Review," in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, 2016, pp. 6–15.
- [33] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider, "What Works Better? A Study of Classifying Requirements," in *Proceedings of the 25th IEEE International Conference on Requirements Engineering (RE'17)*, 2017.
- [34] Z. S. H. Abad, A. Shymka, J. Le, N. Hammad, and G. Ruhe, "A Visual Narrative Path from Switching to Resuming a Requirements Engineering Task," in *2017 IEEE 25th International Requirements Engineering Conference (RE'17)*, 2017.