Leiden University

Master Computer Science

The effects of music listening on program comprehension

Name:	Jacob Jonkman
Studentnr:	s1145509
Date:	April 4, 2020
1st supervisor:	Dr. Ir. Felienne Hermans
2nd supervisor:	Dr. Tessa Verhoef

MASTER THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Abstract

Developing software takes a lot of time. Estimating the time it takes to complete a project is notoriously difficult, and catching up with delays is not always as simple as putting more developers on the job. Because of this, developer productivity is an important area of research. Related research has shown music listening to influence humans performing many different tasks in many different ways, causing both positive and negative effects. The effect of listening to music on someone trying to program is therefore not discernible from related research. One important part of a programmer's job is that of code comprehension, where an existing piece of code is studied closely to discern the inner workings of it, before changes can be made to its behavior. In this thesis, we explore the effects of listening to music on the program comprehension of Computer Science students. 38 participants were asked to analyse short programs and answer what they thought the output of these programs would be. Half of them did this while not listening to any music, while the other half was asked to listen to music of their own choosing. Although a number of close to significant effects were found, no significant effects were found on either the measured time on task and number of correctly answered questions, nor were any effects found on participant-reported measures like perceived distraction and relaxation, except for perceived focus, where music listening had a significant detrimental effect on perceived focus.

Chapter 1: Introduction

Listening to music is a popular way to get through parts of the day. Whether it is for relaxation, motivation, or simply to provide background noise, people like to listen to music in their daily lives for many distinct reasons. Music streaming platforms like Youtube, Spotify and Soundcloud have seen significant growth over the last few years [24], making it easier to listen to whatever kind of music people want to, whenever they want to, which has made music an even more ubiquitous part of life. However, it has been found that music can act as a distraction, preventing people from fully focusing. Many researchers have therefore tried to understand the subtle cognitive effects that can be caused by listening to music, but these efforts have often reached contradictory conclusions compared to related research. Because of this, the influence of music listening on cognitive processes is still regarded as an open research area.

1.1 Cognitive effects of music listening

One of the best known contributions to the field is the 'Mozart Effect' as found by Rauscher, Shaw, & Ky [22]. This research found that participants that were first exposed to 10 minutes of Mozart's music, scored higher on a spatial reasoning IQ test task compared to participants that were exposed to either 10 minutes of relaxing music or silence. This finding is well-known, but is often misinterpreted as having proven that listening to music makes one smarter or more intelligent. However, this result is not sufficient to prove that music can influence the intelligence of listeners, since this research focuses solely on spatial reasoning, which is only a part of the equation when trying to determine one's intelligence, thus limiting the practical uses of this finding [18]. Furthermore, although the results of this study have been replicated by the same researchers [23], other attempts have failed to produce similar findings [27].

The challenge to generalize the original findings of Rauscher, Shaw & Ky to a more broadly applicable result, might be explained due to the fact that many cognitive processes are influenced by listening to music, and not all of them are influenced in the same way. For example, background music was found to degrade performance on a working memory test, i.e. the forward digit span test, in which a sequence of numbers is presented to a subject, who afterwards has to recall the numbers in the correct order [5]. Here it was also found that the perceived distraction of their participants was a poor predictor for the resulting test scores, signifying a difficulty for people to correctly assess their own distraction levels. Threadgold, Marsh, McLatchie & Ball [30] found that background music decreased the creativity of test subjects, irrespective of whether participants normally listened to music while studying or whether the music induced a positive mood or not. Salamé & Baddeley [25] found that background music disrupts immediate verbal memory, with a particularly strong effect of vocal music. Furthermore, the effect of music listening can also be influenced by traits of the individual. For example, Furnham & Bradley [10] found that background music had a detrimental effect on immediate recall in a memory test, as well as having a detrimental effect on test scores of introverts on a delayed recall test and a reading comprehension test. Extraverted participants, however, only experienced a detrimental effect on the immediate recall test, suggesting that introverts have a different reaction to background music than extraverts.

There have also been studies which have reported positive influences from listening to background music. For example, music has been found to improve focus and productivity in early childhood students [28]. Others have found no significant effect of music listening at all, for example on the process of learning and retention of neologisms [16]. In this research, efforts were made to control for intelligence, extraversion and musical education, with all experimental groups showing no significant differences in any of these measures. Participants were presented with 160 neologisms - words or expressions that are not (yet) accepted as being a part of a language, in this case completely new words with either weak or strong semantic associations to existing words - and were asked to respond to each word whether they had seen it before during the experiment. No significant effects were found. As a limitation of the study, the researchers suggested that listening to music that someone likes might lead to a more positive effect. Special care was taken here to select participants that were not familiar with the given music, something they theorize might have had a negative effect on the results, and leave up to future work to prove.

Most research however, has come to more nuanced conclusions, in which more in-depth findings are presented which take into account certain factors of the human condition. For example, Giannouli, Kolev & Yordanova [12] assessed verbal working memory and phonologically-cued semantic retrieval with background music, but only in participants without a musical background, and made a distinction between younger and older participants. They tested this using musical excerpts of three composers of classical music: Mozart, Vivaldi and Glass. It was found that brief exposure to music did not have a significant positive effect on verbal working memory, with a transient impairment after the music of Vivaldi. In contrast, listening to Vivaldi led to an improvement of phonologically-cued semantic retrieval, but only for the younger participants, while participants who listened to Mozart produced decreased test scores. From this, it is suggested that different composers, and perhaps even different compositions, can cause different effects on cognitive functions. By manipulating the tempo and intensity of a piece of background music, Thompson, Schellenberg & Letnic [29] found that specifically fast and loud background music affected reading comprehension, further signifying that the specifics of a piece of music matter when trying to predict the effects of listening to a piece of music.

Another study investigated the effect of both experimenter-selected and participant-selected music on driving game performance [6]. Here, it was found that "participants exposed to their self-selected music were most efficient, perceived lowest distraction, highest enjoyment, liking and appropriateness, and experienced a reduction in tension-anxiety", while the exact opposite was true for those participants that were exposed to experimenter-selected music. It is suggested that this could be caused by the subjective relationship one has with one's own music, which can cause context appropriate arousal and association. This supports the theorized limitations of the work of Jäncke & Sandmann [16], where no effects of music listening was found, but the observed effect was suggested to be limited by the choice to use unfamiliar music during the experiment.

1.2 Programmer productivity

The impact of external factors, like music listening, on productivity is very relevant to the field of software engineering, since developing software is expensive. Developer productivity, and finding those factors which influence it, is an active research topic with real implications for businesses. As far as we know, there has only been one paper that studied the effect of music on programming specifically [19]. In that paper, the researchers studied the effects of music on programmer productivity by following professional developers in their work during a five-week experiment. During the first week, the participants were told "to do what they normally did in respect to music listening". During the second, third and fifth weeks, they were given plenty of means to listen to music, and were told to "Listen to the music when you want to, as you want to". In the fourth week, participants were asked to not listen to any music. It was found that the self-reported "Quality of Work" measure decreased during weeks 2, 3 and 4, reaching its lowest point in week 4 when music listening was prohibited, and then in week 5 returned close to the baseline that was reported in week 1. A similar trend was found for the "Time on Task" measure, in which the fourth week produced the worst results, while in week 5 the results returned close to the baseline reported in week 1. This result is explained by the existence of a certain learning curve involved with listening to music, where those programmers who were not used to listening to music during their work had to get used to it. The study suggested that music might help programmers to positively alter their mood when needed, if they know how to do so, thus increasing their quality of work.

When working on a code-base, there are many code-centric factors which influence the speed with which developers will be able to make changes. A large part of the difficulty of maintaining a code-base comes from having to be intimately familiar with the specifics of a piece of code. Care should therefore be taken to make the code of any software project easily understandable. For example, the presence of linguistic antipatterns in source code (for example, a variable whose name suggests that it is of type boolean, while it actually is not) have been shown to significantly increase the cognitive load of developers [8]. An example of something that can help a program's readability, or greatly hamper it, are good naming conventions, where for example single-letter or abbreviated identifier names have been shown to reduce productivity compared to word identifiers [14]. Refactoring has been defined as "a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior". Use of refactoring tools has been shown to increase developer productivity through making it easier to maintain and change existing code [20].

Apart from the specifics of a code base, many human-centric factors also influence programmer productivity. For example, research has shown that interrupting programmers can be a costly thing to do, where interrupted programmers sometimes take up to 10-15 minutes to reach the point where they can continue working again, time which is for example spent trying to rebuild context which was lost upon interruption [21]. It has also been found that happier employees are in general more productive. This has also been empirically found to be true for software developers specifically [13]. Stress has also been found to be a large cause for software development faults [11]. All of these issues might be mitigatable through the use of background music. For example, background music has been shown to be able to lower stress levels [17]. Furthermore, listening to music with headphones on can signal the desire to not be disturbed by colleagues, which might result in more uninterrupted stretches of work.

One important part of the software development life-cycle, is the process of bug-fixing. In order to be able to fix a bug, a programmer often needs to extensively study a piece of code to develop a sense of understanding about the function and the specifics of the code, often using both top-down (first deriving a general sense of what the purpose of the code is before refining their understanding) and bottom-up (examining individual statements of a program and grouping them into semantic chunks) strategies [9]. Could music listening help a programmer in the endeavour of comprehension, or does music act as a distractor in this case?

1.3 Research questions

Music has been found to have a wide variety of effects on humans, for example decreasing creativity in test subjects [30] and affecting reading comprehension [29], but also increasing feelings of happiness [13] and reducing stress [17]. In this research, we will study the effects of listening to music while programming. To be exact, we will be answering the following research questions:

1. How does listening to music influence a programmer's ability to understand pieces of code? By conducting an experiment in which participants will be asked to comprehend a program, either with or without listening to music, we will study the effects of music listening on program comprehension.

2. How do programmers perceive their distraction while listening to music while trying to understand a piece of code? Alley & Greene [5] reported a mismatch between how distracted participants felt and how distracted the data actually showed they were while completing a working memory test. We would like to see if we can replicate this result with programmers.

3. What effect does the chosen style of music have on code comprehension? Participants will be able to choose their own music during the experiment, since Cassidy & MacDonald [6] found that participant-selected music had a positive effect on task performance compared to experimenter-selected music. We are also interested in mimicking the work conditions of a typical programmer as much as possible, which in this case means letting the participants choose their own music. We would however like to know if there are any discernible patterns between the kind of music participant choose and the results that they produce. For example, do people who choose music without lyrics score differently than participants with music that does contain lyrics?

Chapter 2: Methods

We conducted a controlled between-subjects experiment with two experimental conditions: Music (listening to music during the experiment) and Silence (not listening to any music during the experiment). The participants were distributed over the experimental conditions in a pseudo-random manner, ensuring equal sample sizes for both conditions as well as placing the same amount of men and women in both groups. Participants were asked to complete a code comprehension test and to fill in a questionnaire afterwards. As compensation for completing the experiment, the participants had a chance to win one of three €25,- gift cards.

2.1 Participants

We conducted our experiment with students studying at the Leiden University. Specifically, we conducted the experiment with bachelor students that have successfully completed at least one programming course in which they were taught the programming language C++. This meant that students from bachelor programs like Computer Science, Computer Science and Economics, and Mathematics were eligible to participate. Through this requirement, we excluded first-year bachelor students, since the experiments were conducted in the fall. When considering these first year students, there are large differences in experience between students. Some students have been programming for years when they start their study, while others have never written a line of code in their life. By not considering these first year students, and only considering students who have passed at least one programming course, we can control for this possible gap in experience, thereby reducing the possible differences between different participants somewhat. We asked the participants whether or not they have successfully completed at least one of the first-year programming courses beforehand, where not having achieved a passing grade was reason to exclude a participant. We used a sample size of 19 participants per experimental condition, for a total of 38. Both groups consisted of 15 male students and 4 female students. Although the application and questionnaire as well as all the instructions of the experiment were written in English to allow international students to participate, all participants were native Dutch speakers.

2.2 Procedure

The experiment was conducted in the Snellius building of the Leiden University. Participants were seated in front of a laptop in an otherwise empty room. The laptops were all running Windows 10. Because the participants completed the experiment in silence (possible music was only played through a participant's pair of earphones), multiple students were able to participate in the experiment at the same time, within a single classroom.

After reading and signing an informed consent form, which can be found in Appendix C, the participants were then presented with instructions about the experiment that they were about to participate in. Here they were told that they would be looking at six code snippets, and that their task would be to fill in what they thought the program would print to the command line when executed. Participants were notified of the fact that they would be asked to fill in a questionnaire after the experiment. Participants in the Music condition were also asked to construct a playlist of music of their choosing which was about 20 minutes in length and not to remove their earphones during the experiment. Earphones were provided, but the participants were also allowed to use their own headphones or earphones. Then, they were presented with a sequence of six code snippets written in C++. Their task was to study these programs, and fill in what they believed the programs would print on the command line. Participants were not allowed to use any resources to help them answer the questions. The participants were told that they would be timed per code snippet, but they were also encouraged to take their time, so that they would not feel pressured to rush through the experiment. The questions were answered through a web application [2] that was written in Angular 8 [3], which stored the answers inside of a Firebase database [4], along with the number of seconds spent on each assignment. The application started a new timer every time the participant proceeded to a new question. This timer itself was not visible to the participant, again to prevent stress. The application did not allow participants to return to a previously answered question. A screenshot of our application mid-experiment can be seen in Figure 2.1

Figure 2.1: One code snippet of our application: determining whether a string is a palindrome.

We used the code snippets by Siegmund et al. [26]. In this research, functional magnetic resonance imaging (fMRI) was used to study programmers while they performed a series of program comprehension tasks. Several programs of about 20 lines of code each were constructed and then manipulated to include beacons (meaningful function names) or no beacons and pretty-printed layout or disrupted layout. The code snippets that were used in this research can be found on Github [1]. These code snippets were originally written in Java, but since Computer Science students at the Leiden University primarily learn C++ in their first year, we reimplemented these functions in C++. We took care to use the same function and variable names, comments, and coding style as were used in the original code snippets, but some small changes had to be made to make the programs compatible with C++. The C++ programs we used can either be found in Appendix A, or within our Github repository [2]. When crafting these programs, Siegmund et al. took care to both make the programs short enough that they could be comprehended within a small time window, as well as making sure the computations were not too complex so that they could easily be done by a human. We only used the 'bottom up' versions of the code snippets, with pretty-printed layout and no scrambled beacons, since these versions most closely match the way a regular program typically looks like. We did use obfuscated variable names, since these would provide too much information to the test subject when trying to determine the function of a program.

We used 6 different code snippets from the set of code snippets used by Siegmund et al. [26] in one of three categories: Programs that computed numerical properties (computing a list's average and counting the number of matching characters in two strings), boolean properties (is a string a palindrome and is a string a substring of a second string) and the manipulation of an input variable (converting a bitstring to an integer and stitching two strings together). All six code snippets that we used can be found in Appendix A. Participants were expected to spend no longer than 20 minutes answering all 6 questions.

Afterwards, participants were asked to fill in a questionnaire. They were asked to enter their age, gender, experience with programming, number of years that they had been studying and more. There were also questions about their personal preferences regarding listening to music and whether they knew how to play an instrument, as well as their experiences during the experiment. Here, we measured for example how relaxed the participants felt, how distracted they felt and whether they liked participating in the experiment through questions like "I felt relaxed during the experiment", which they had to answer on a 7-point scale where '1' corresponds with "Strongly agree" and '7' corresponds with "Strongly disagree". Finally, there was a checkbox indicating that a participant wanted to be updated about the results of the experiment, and an optional field for a participant's email address. The full questionnaire can be found in Appendix B.

Chapter 3: Results

3.1 RQ 1

In order to explore the effects of music listening on code comprehension, we conducted an experiment in which participants had to read C++ programs and determine the outcomes of said programs. No significant differences were found between the experimental groups in terms of age, number of years of experience in programming, number of years studying, being able to play a musical instrument, music listening habits and extra/introversion. These statistics can be seen in Table 3.1. However, both age and years of programming experience almost reached a significant difference, with significances of $p = 0.080$ and $p = 0.094$ respectively. The participants in the Silence condition were about one year older and had about half a year more experience in programming than the participants in the Music condition.

	Music M	Music SD	Silence M	Silence SD	p
Age	20.5	1.6	21.5	1.8	.080
Years studying	2.5	1.11	2.8	1.3	.504
Programming experience	.79	.7	1.16	.6	.094
Music while working[1]	2.47	1.020	2.95	1.311	.222
Music in spare time[1]	3.11	.937	2.95	1.129	.642
Able to play instrument[2]	.53	.513	.53	.513	1.000
Extra/introversion[2]	.506	.140	.414	.107	.298

Table 3.1: Statistics of our experimental groups

Using an independent-samples T-test, we found no significant effects of music listening on either the time needed to complete the experiment ($p = .741$) or the number of correctly answered questions ($p = .340$). Means and standard deviations can be found in Table 3.2. We also measured participant-reported statistics through a questionnaire. The results of this can be viewed in Table 3.3. A significant detrimental effect of music listening on participant-reported focus was found ($p < .05$). All other participant-reported measures were not significantly different, although a marginally significant positive effect of music listening on reported distraction ($p = .064$) was found.

[1]Where 1 corresponds with 'Never', 2 with 'Sometimes', 3 with 'About half of the time', 4 with 'Most of the time' and 5 with 'Always'.

[2]Where 0 corresponds with 'Not able to play an instrument' or 'Introvert', and 1 corresponds with 'Able to play an instrument' or 'Extravert'. Participants were able to answer that they were not sure about their introversion or extraversion, in which case their answers were omitted from the analysis.

	Music M	Music SD	Silence M	Silence SD	p
Total time spent	797	250	770	260	.741
Number of correct answers	4.32	1.157	4.63	.831	.340

Table 3.2: Statistics for the time taken to complete the experiment as well as the number of correct answers

	Music M	Music SD	Silence M	Silence SD	p
Focused	3.58	1.68	2.47	1.31	**0.030**
Distracted	4.32	1.97	5.42	1.58	0.064
Relaxed	3.05	1.39	2.53	1.35	0.245
Stressed	4.58	1.58	5.16	1.57	0.264
Enjoyed	1.95	.52	2.32	.89	0.129
Not enjoyed	6.16	.69	6.00	1.00	0.574

Table 3.3: Reported variables

The assignments in our experiment were divided in three categories: numerical properties, boolean properties and input manipulation. We researched the differences between different question categories to see if the different types of questions were effected differently. The resulting mean values for time on task as well as the percentage of correct answers per category can be seen in Figure 3.2 and Figure 3.4. A series of T-tests were performed to analyse the potential differences between question categories, but no significant differences were found. The same was done for the individual assignments, the results of which can be seen in can be seen in Figure 3.1 and Figure 3.3. Again, no significant effects were found.

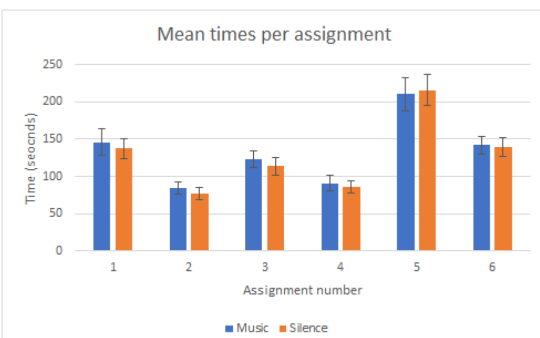

Figure 3.1: Mean times per assignment

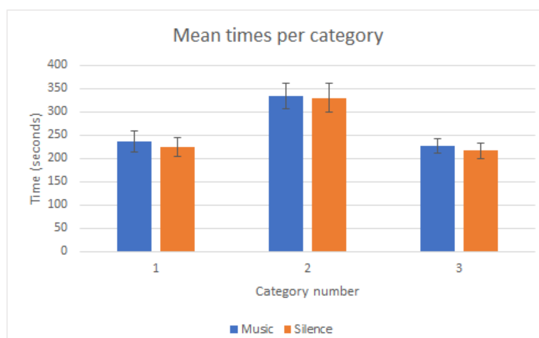

Figure 3.2: Mean times per category

Given the marginally significant difference in age and years of experience in programming between the two experimental conditions, we wanted to check for interaction effects for these parameters. Using a Linear Mixed Model approach, we determined that there were no significant interactions between time on task and age ($p = .570$) or years of experience ($p = .547$). There were also no significant interactions between the number of correctly answered questions and age ($p = .807$) or years of experience ($p = .730$).

A between subjects Multivariate Analysis of Variance (MANOVA) was carried out to determine whether combining our (measured and participant-reported) dependent variables would

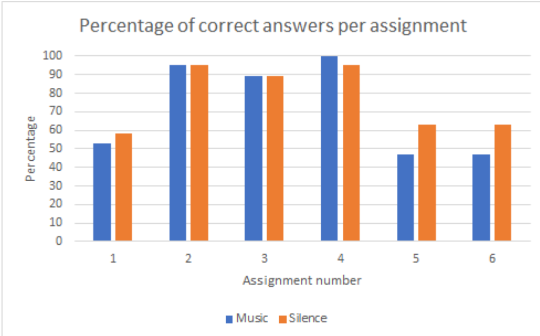

Figure 3.3: Percentage of correct answers given per assignment

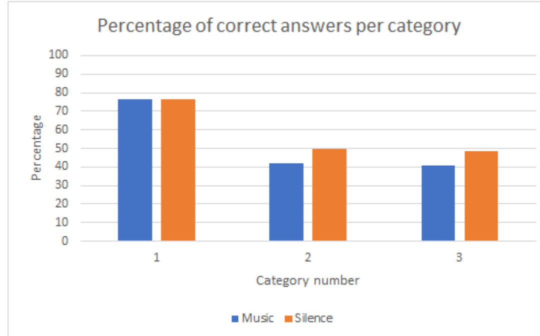

Figure 3.4: Percentage of correct answers given per category

produce a significant difference between our experimental groups. The test did not produce a significant difference between the Music and Silence conditions (F(8, 29) = 1.251, p = .306; Wilks' λ = .743). Generally speaking, listening to music does not seem to have had a significant effect on the program comprehension of our test subjects.

3.2 RQ 2

Our second research question was whether the perceived distraction participants experienced during the experiment was actually a predictor for their test scores. In order to answer this question, we used Pearson Correlation, which revealed a significant effect of reported focus on time on task ($r(36) = .39$, $p < .05$). No effect was found of focus on number of correctly answered questions ($r(36) = -.11$, $p = .512$). Perceived distraction did not produce a significant correlation on time on task ($r(36) = -.22$, $p = .184$) or number of correctly answered questions ($r(36) = .03$, $p = .837$). This result is mirrored by the finding in the previous section where reported focus was significantly lower among the participants that had listened to music, while a similar effect was not present for the measured variables.

3.3 RQ 3

In our survey, we asked participants who listened to music during the experiment to describe the music that they listened to through an open text field. We explored whether we could categorise these answers to see if we could discern patterns in performance or self-reported measures through doing so. Of the 19 participants that listened to music, 17 gave comments about the music they listened to. The first distinction we made, was whether the music contained lyrics or not. 5 participants listened to music without lyrics, while 12 participants listened to music with lyrics. No significant effects were found, although we did find two close to significant effects, on participant focus ($p = .087$) and relaxation ($p = .093$), where participants listening to music without lyrics felt more focused and relaxed. Next we looked at whether participants valued the music as being relaxing/happy/upbeat or not. Only 11 participants provided their opinion on this, with 4 participants stating that their music was active, noisy or not relaxing, and 7 stating their music was relaxing, mellow or calm. We again only found an almost significant effect on focus ($p = .061$), where participants listening to music they perceived to be relaxing felt more focused. Finally, we explored whether there was a difference between those participants that reported to have listened to pop music (6), or not (11). Here, a close to significant effect was found ($p = .090$), where pop music listeners felt more distracted.

Chapter 4: Discussion

In this thesis, we explored the effects of listening to music on program comprehension. No significant effects were found of music listening on time on task or the number of correctly answered questions. This suggests that the act of listening to music does not enhance students' abilities to analyse and comprehend pieces of code nor does it impair their ability to do so. From this, we can draw a number of conclusions.

First off, our findings somewhat contradict the findings of the Mozart effect [22] [23], although the specific music the participants were subjected to during the experiment was of a different nature, which might explain the differences in results. Giannouli et. al [12] found evidence that different composers produced a different effect on working memory tests, which can be broadened to indicate that different music influences working memory differently. If this were to be the case, then surely our findings cannot realistically be compared to that of the Mozart effect, since no requirements were posed to the participants when they were selecting their music of choice. However, since numerous research has not been able to reproduce these findings either [27], we feel like we can safely disregard the findings of Rauscher, Shaw & Ky in this context.

Lesiuk [19] found that developers only experienced a positive change in their self-reported quality of work and time on task after listening to background music for a few weeks, where at first the background music even seemed to produce a negative effect on both of these measures. It is suggested that this might be caused by a certain learning curve or period where people have to get used to working while listening to music. 50% of our respondents reported to listen to music while studying or working either 'Never' (15.8%) or 'Sometimes' (34.2%), while only 31.6% said they did this 'most of the time' (26.3%) or 'Always' (5.3%). This might indicate that a similar period would be needed for the participants to get used to listening to music, with those participants that are used to music listening while working canceling out the observed effect. Using an independent samples T-Test, we found that there was indeed a significant effect of how often participants listened to music while working/studying on the time needed to complete the experiment ($p < 0.05$), where those participants that reported listening to music at least half of the time needed less time to complete the experiment, although the number of correctly answered questions was not influenced similarly ($p = 1.0$). From this, we can conclude that the findings of Lesiuk have merit, further adding to the theory of a learning period being present when listening to music while working in a programming environment. A similar effect was not present when comparing how often participants listened to music in their free time, indicating that there is a tangible difference between different ways of listening to music.

Using an independent-samples T-test, we found two significant effects of paticipant gender: on levels of stress ($p < 0.005$) and relaxation ($p < 0.05$) during the experiment. Female participants generally felt less relaxed and more stressed than the male participants. It is interesting to note that the female participants generally scored higher than male participants, but took longer to finish the assignments. Both of these results however, were not significant. How might we explain that the female participants experienced more stress during our experiment? Höhne & Zander [15] found that women in a Computer Science degree in general reported higher levels of belonging uncertainty than men, which was found to be predicted by a lower sense of self-efficacy in women, even though women generally graduated with higher grades. A feeling of unbelonging, low self-efficacy or a mistrust of one's skills could explain a more stressful experience while participating in our experiment. Indeed, where the male participants did not express any doubts about their skills, a number of the female participants did express doubts about their programming skills when asked to participate in our experiment, even though no significant difference between male and female participants was found in terms of actual skills. However, given the low number of female participants, we should be reluctant to draw any grand conclusions from this result about the differences between men and women in Computer Science.

Graziotin et al. [13] found a positive correlation between feelings of happiness and developer productivity. We theorized that music might positively influence the affective states of developers, in which case we might be able to find an interaction between the enjoyment and disenjoyment of participants on their test performance. Since we did not find a significant effect of music listening on self-reported enjoyment or disenjoyment, we might conclude that this was not necessarily a correct assumption. However, when studying these results more closely, we find that in general, our participants enjoyed the experiment quite unanimously. Only one participant reported to "Somewhat disagree" to the statement "I enjoyed participating in the experiment" (corresponding to a 5 out of 7), and the same can be seen for "I did not enjoy participating in the experiment", where only one person reported to "Somewhat agree" (3 out of 7). In both cases, around 80% of participants reported to either agree or strongly agree to the statement regarding enjoyment, and the other way around for disenjoyment. Given this irregular distribution of our data, it is no wonder that no effect is shown by our data, but it might be reason to not completely disregard the findings of Graziotin et al. Our experiment only took an average of 13 minutes per participant. It might be the case that if our experiment had taken longer to complete, the participants who were not listening to music would have become more bored by the experiment, lowering their enjoyment and possibly their concentration and test scores in the process.

Similar to the research of Alley & Greene [5], we found perceived distraction to be a poor predictor of test performance, both in terms of time on task, and the number of correctly answered questions. Interestingly, although the perceived focus was similarly a poor predictor of the number of correctly answered questions, it was a strong predictor for the time on task measure. First off, it is interesting to see that there is a difference between how focused participants reported to have felt and how distracted they felt, as one would presume that these two measures are mostly mirrored. When comparing participants' reported focus compared to their reported distraction, we can see that in general, participants judged their distraction levels to be a bit higher than their focus levels. From this, it seems like participants were more critical about their performance when the question was asked in a negative way.

It is also interesting to note that the time on task was accurately predicted by participant-reported focus, but not the number of correctly answered questions. This might indicate that participants mostly judged their own performance by how fast they completed the experiment. Why is it that people seem to struggle to correctly assess their distraction? In a related research [7], the effect of irrelevant speech was measured on digit span test scores. Here it was found that male participants had a harder time to correctly assess their own impairment than female participants did. Given the small amount of female students that participated in this study, it might be the case that this has skewed the observed effect even further.

We looked at the differences between vocal and instrumental music, relaxing and non-relaxing music, and pop-music and different kinds of music. For all three of these distinctions, no differences were found in terms of time on task or number of correctly answered questions. The self-reported measures did not produce any significant results either, although there were a number of results that were close to significant. Alley & Greene [5] found that vocal music was perceived as more distracting than instrumental music and silence. This result is somewhat mirrored by our finding that perceived focus was higher when listening to instrumental music compared to vocal music, although again, this effect was not quite significant. Similarly to our results, they did not find an effect of vocal or instrumental music on test scores.

4.1 Limitations and shortcomings

This research seems to suggest that listening to music has no tangible effect on someone's ability to comprehend a piece of code. Can we generalize this result to mean that listening to music does not influence a programmer's efficiency or quality of work? Unfortunately, although this experiment does mimic a vital part of the software engineering process as stated in Chapter 2, it also misses some essential elements to completely mirror a programmer's work conditions. First of all, although the act of reading code and formulating a sense of understanding of the code is a vital step in the software engineering process - perhaps even fundamental, since it is the very first thing a programmer has to do before being able to make any change to a code base - these code snippets were really just short and isolated algorithms. Normally, a piece of code is not an isolated unit of computation, but is only a small part of a larger, more complex program. A piece of code might be executed through multiple function calls in different areas of a program, the output of a function might be used in unexpected ways, and many units of a program (i.e. modules or classes) keep track of their individual state, resulting in programs which cannot be seen without considering the complete encapsulating program. These problems and more can severely increase the difficulty of understanding a real-world program compared to the clean code snippets that we used in this experiment.

Another problem comes from the fact that in a programming job, the programmer will typically be stationed in a busy office, with many external distractions like people chatting, colleagues walking by or asking questions, phones ringing and so on. Compared to our test environment, a quiet classroom in the university building, the amount of distractions in such an environment is huge, while distractions have been found to influence a developer's productivity [21]. Compared to music, which can be familiar, regular, pleasant and does not necessarily ask for the programmer's active attention, these external distractions might have a larger influence on a developer's efficiency and quality of work. If music can drown out these external distractions in favor of itself which, according to our findings, does not significantly lower someone's ability to focus on their code, the actual effect of listening to music might be greater when tested in a real-world environment. This theory is supported by the research of Lesiuk [19], but is worth investigating more thoroughly.

Furthermore, this research is not sufficient to answer questions regarding the quality or productivity with which programmers actually write new code while listening to music. After attaining an understanding of a piece of code or a code base, the actual process of writing code is a fairly creative endeavour. Threadgold et al. [30] found a negative effect of listening to music on participant creativity. When considering this possible decrease in creativity in light of the possible greater focus as theorized in the previous paragraph, it is tempting to try and theorize which possible effect might overrule the other, but this is not something that can reliable be done with the current research.

When considering Figure 3.3, we notice that three of our assignments are correctly answered by at least 90% of our participants for both experimental conditions. These assignments correspond to the code snippets that stitch two strings together, determine the number of matching characters between two strings, and determine whether a string is a palindrome. This means that at most two participants per experimental condition did not answer these questions correctly. Half of our assignment might therefore have been too easy for most of our participants. This in turn might have suppressed the observed effect of music listening in our experiment, thus explaining our inability to find significant effects. When considering the other three questions in Figure 3.3, it seems like there might have been a perceived effect for the harder questions, but considering only these harder questions also did not produce a significant effect. Still, the fact that we used three assignments that were likely too easy for most of our participants, could be an explanation of our inability to find a significant effect.

We have found multiple results which approached significance but did not quite reach significance. This might be a result of the relatively small sample size (a total of 38 participants) that we used in this experiment. Combined with the fact that every participant only had to solve 6 assignments, we may have found a more potent effect had we increased either our sample size, the amount of assignments per participant, or both. Another limitation comes from the fact that we used mostly self-reported measures, which can be influenced by biases. For example, if someone has low confidence in their own capabilities, he or she might judge their distraction to be higher that it actually is.

We have found a number of almost significant effects of the specifics of the music on participant-reported measures. However, the way we asked participants about the specifics of their chosen music, is suboptimal. Our method, with just an open text question, did not produce very useful results. Instead, we should have constructed characteristics that we thought might produce interesting results, and specifically asked for those. For example, we could have asked every participant to quantify how relaxing they thought their music was on a scale of 1 to 5, to state whether they listened to vocal or instrumental music, whether it was high-tempo or low-tempo and so on. This would have produced more reliable distinctions and perhaps more reliable results. However, even if we had done this, our experiment still might not have been sufficient to answer our third research question, since only half of our participants would have provided data about these effects. A new study specifically about the different effects of different characteristics of music should be conducted to better understand how different kinds of music influence people. Such a study would not have to bother about including a Silence condition and would therefore have an easier time to gather enough data.

4.2 Future work

We did not find a significant effect of music listening on program comprehension in terms of accuracy or time on task. Our experiment should be repeated with a larger sample size in order to make sure that our findings are valid and reproducible. Perhaps more assignments should be added as well. We found a difference in the levels of stress between male and female students. Because Computer Science is such a male-centric field, it can be valuable to see if such an effect really exists, or if it is a result of the small amount (8 of 38) of female participants that completed our experiment.

Furthermore, additional research should be done concerning different parts of the software engineering work-cycle. We have explored the effects of music listening on code comprehension, but it is interesting to see if similar effects are found when concerning tasks like (architectural and front-end) design, writing tests and documentation, refactoring, and debugging. All of these tasks pose different cognitive requirements on the software engineer, and might therefore be influenced in different ways by music listening.

Finally, efforts could also be made to validate our test. Perhaps even more importantly: while we were conducting our literature review, we did not find any standardized tests regarding the measurement of code comprehension. Since it is an active research topic, it seems like a valuable contribution to the field to develop a validated and reusable test.

Chapter 5: Conclusion

In this research, we explored the effects of music listening on program comprehension. In order to do this, we conducted an experiment in which Computer Science students had to read short code snippets and had to answer what output they thought these short programs would produce, either while they were listening to background music of their own choosing, or while sitting in silence. No significant differences were found between our two experimental conditions, neither for time on task and the number of correctly answered assignments, nor participant-reported measures, namely stress, relaxation, distraction, focus, enjoyment and disenjoyment during the experiment. From this, we conclude that no tangible effect of music listening on program comprehension seems to be present. Students did not perform better or worse in either condition. However, this result can not be generalised to mean that music listening has no effect on programmer efficiency. Different parts of a software engineer's work might be affected differently by background music, and further research should be done in order to confidently conclude anything about whether it is wise or unwise to listen to music while programming.

Bibliography

[1] *Github repository of code snippets* Retrieved from https://github.com/brains-on-code/paper-esec-fse-2017/, last visited November 13th, 2019.

[2] *Github repository containing the code for our application as well as our code snippets.* `https://github.com/jacob-jonkman/music-and-code`

[3] *Angular*. Retrieved from https://angular.io/, last visited November 13th, 2019.

[4] *Firebase*. Retrieved from https://firebase.google.com/, last visited November 13th, 2019.

[5] Alley, T.R. & Greene, M.E. (2008). The relative and perceived impact of irrelevant speech, vocal music and non-vocal music on working memory. *Current Psychology, 27*(4), 277-289.

[6] Cassidy, G., & MacDonald, R. (2009). The effects of music choice on task performance: A study of the impact of self-selected and experimenter-selected music on driving game performance and experience. *Musicae Scientiae, 13*(2), 357-386.

[7] Ellermeier, W., Zimmer, K. (1997). Individual differences in susceptibility to the "irrelevant speech effect". *The Journal of the Acoustical Society of America, 102*(4), 2191-2199.

[8] Fakhoury, S., Ma, Y., Arnaoudova, V., & Adesope, O. (2018, May). The effect of poor source code lexicon and readability on developers' cognitive load. In *Proceedings of the 26th Conference on Program Comprehension* (pp. 286-296). ACM.

[9] Feigenspan, J., Siegmund, N., & Fruth, J. (2011). On the role of program comprehension in embedded systems. In *Proc. Workshop Software Reengineering (WSR)* (pp. 34-35).

[10] Furnham, A., & Bradley, A. (1997). Music while you work: The differential distraction of background music on the cognitive test performance of introverts and extraverts. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition, 11*(5), 445-455.

[11] Furuyama, T., Arai, Y., & Iio, K. (1994). Fault generation model and mental stress effect analysis. *Journal of Systems and Software, 26*(1), 31-42.

[12] Giannouli, V., Kolev, V. & Yordanova, J. (2019). Is there a specific Vivaldi effect on verbal memory functions? Evidence from listening to music in younger and older adults. *Psychology of Music, 47*(3), 325-341.

[13] Graziotin, D., Wang, X., & Abrahamsson, P. (2013). Are happy developers more productive?. In *International Conference on Product Focused Software Process Improvement* (pp. 50-64). Springer, Berlin, Heidelberg.

[14] Hofmeister, J. C., Siegmund, J., & Holt, D. V. (2019). Shorter identifier names take longer to comprehend. *Empirical Software Engineering, 24*(1), 417-443.

[15] Höhne, E., & Zander, L. (2019). Sources of male and female students' belonging uncertainty in the computer sciences. *Frontiers in psychology, 10*, 1740.

[16] Jäncke, L., & Sandmann, P. (2010). Music listening while you learn: No influence of background music on verbal learning. *Behavioral and Brain Functions, 6*(1), 3.

[17] Kemper, K. J., & Danhauer, S. C. (2005). Music as therapy. *South Med J, 98*(3), 282-288.

[18] Kuepper-Tetzel, C. (2016). *Listening to Music while Studying: A Good or a Bad Idea?* Retrieved from http://www.learningscientists.org, last visited June 16th, 2019.

[19] Lesiuk, T. (2005). The effect of music listening on work performance. *Psychology of music, 33*(2), 173-191.

[20] Moser, R., Abrahamsson, P., Pedrycz, W., Sillitti, A., & Succi, G. (2007, October). A case study on the impact of refactoring on quality and productivity in an agile team. In *IFIP Central and East European Conference on Software Engineering Techniques* (pp. 252-266). Springer, Berlin, Heidelberg.

[21] Parnin, C. (2013). *Programmer interrupted* [Personal blog post]. Retrieved from https://blog.ninlabs.com/, last visited on July 1, 2019.

[22] Rauscher, F.H., Shaw, G.L., & Ky, K.N. (1993). Music and spatial task performance. *Nature, 365, 611*.

[23] Rauscher, F. H., Shaw, G. L., & Ky, K. N. (1995). Listening to Mozart enhances spatial-temporal reasoning: towards a neurophysiological basis. *Neuroscience letters, 185*(1), 44-47.

[24] Richter, F. (2019). *The Unstoppable Rise of Music Streaming*, retrieved from https://www.statista.com. Last visited September 5th, 2019.

[25] Salamé, P., & Baddeley, A. (1989). Effects of background music on phonological short-term memory. *The Quarterly Journal of Experimental Psychology Section A, 41*(1), 107-122.

[26] Siegmund, J., Peitek, N., Parnin, C., Apel, S., Hofmeister, J., Kästner, C., Begel, A., Bethmann, A. & Brechmann, A. (2017, August). Measuring neural efficiency of program comprehension. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 140-150). ACM.

[27] Steele, K. M., Bass, K. E., & Crook, M. D. (1999). The mystery of the Mozart effect: Failure to replicate. *Psychological Science, 10*(4), 366-369.

[28] Strachan, D. L. (2015). The space between the notes: The effects of background music on student focus.

[29] Thompson, W. F., Schellenberg, E. G., & Letnic, A. K. (2012). Fast and loud background music disrupts reading comprehension. *Psychology of Music, 40*(6), 700-708.

[30] Threadgold, E., Marsh, J.E., McLatchie, N., & Ball, L.J. (2019). Background music stints creativity: Evidence from compound remote associate tasks. *Applied Cognitive Psychology*.

Appendix A: Used code snippets

A.1 Assignment 1 - Compute array average

```
#include <iostream>
using namespace std;

int main() {
    int input[] = {4,2,3,5,1};
    int number1 = 0;
    int number2 = 0;

    while (number1 < sizeof(input)/sizeof(int)) {
        number2 = number2 + input[number1];
        number1 = number1 + 1;
    }

    float result = number2 / (float)number1;
    cout << result << endl;
}
```

A.2 Assignment 2 - Intertwine two strings

```
#include <iostream>
using namespace std;

int main() {
    string word1 = "abc";
    string word2 = "def";
    string result = "";

    if (word1.length() == word2.length()) {
        for (int i = 0; i < word1.length(); i++) {
            result = result + word1.at(i) + word2.at(i);
        }
    }
```

```
    cout << result << endl;
}
```

A.3 Assignment 3 - Is string a palindrome?

```
#include <iostream>
using namespace std;

int main() {
    string input = "abba";
    bool result = true;

    for (int i = 0, j = input.length() - 1; i < input.length() / 2; i++, j--)
        if (input.at(i) != input.at(j)) {
            result = false;
            break;
        }
    }

    cout << (result ? "true" : "false") << endl;
}
```

A.4 Assignment 4 - Count matching characters between two strings

```
#include <iostream>
using namespace std;

int main() {
    string string1 = "Magdeburg";
    string string2 = "Hamburg";

    int length;
    if (string1.length() < string2.length())
        length = string1.length();
    else length = string2.length();

    int result = 0;

    for (int i = 0; i < length; i++) {
        if (string1.at(i) == string2.at(i)) {
            result++;
        }
    }
```

```
    cout << result << endl;
}
```

A.5 Assignment 5 - Does string A contain string B?

```
#include <iostream>
using namespace std;

int main() {
    string word1 = "abcdefg";
    string word2 = "acd";
    bool result = false;
    for (int i = 0; i < word1.length(); i++) {
        for (int j = 0; j < word2.length(); j++) {
            if (i + j >= word1.length())
                break;
            if (word1.at(i + j) != word2.at(j)) {
                break;
            } else {
                if (j == word2.length() - 1) {
                    result = true;
                }
            }
        }
    }
    cout << (result ? "true" : "false") << endl;
}`
```

A.6 Assignment 6 - Convert binary to decimal

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    int number = 11;
    if (number < 0)
        return -1;

    int result = 0;
    int tempNumber = number;
    int variable = 0;

    for (int i = 0; tempNumber > 0; i++) {
```

```
        variable = tempNumber % 10;
        result = result + variable * (int) pow(2, i);
        tempNumber = tempNumber / 10;
    }

    cout << result << endl;
}
```

Appendix B: Questionnaire

1. I identify myself an a:
   - Male
   - Female
   - Other / Rather not say
2. How old are you?
3. Which study program are you following? (i.e. Computer Science, Computer Science and Economics, etc.)
4. How many years have you been studying your current study program?
5. How many years of experience do you have in programming?
   - 1 year
   - 2-4 years
   - 5 or more years
6. How often do you listen to music while studying/working?
   - Never
   - Sometimes
   - About half of the time
   - Most of the time
   - Always
7. How often do you listen to music in your spare time?
   - Never
   - Sometimes

- About half of the time
- Most of the time
- Always

8. Do you know how to play an instrument?
   - Yes
   - No
9. I would classify myself as an:
   - Introvert
   - Extravert
   - I am unsure
10. Do you want to be notified of the results of the study?
    - Yes
    - No
11. What is your email address? This is used to contact the winners of the raffle and to notify you of the results of the experiment if you choose to be notified of this. You can also leave this field empty
12. I felt focused during the experiment
    - Strongly agree
    - Agree
    - Somewhat agree
    - Neither agree nor disagree
    - Somewhat disagree
    - Disagree
    - Strongly disagree
13. I enjoyed participating in the experiment
    - Strongly agree
    - Agree
    - Somewhat agree
    - Neither agree nor disagree
    - Somewhat disagree
    - Disagree
    - Strongly disagree
14. I felt relaxed during the experiment

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Disagree
- Strongly disagree

15. I felt distracted during the experiment
    - Strongly agree
    - Agree
    - Somewhat agree
    - Neither agree nor disagree
    - Somewhat disagree
    - Disagree
    - Strongly disagree
16. I did not enjoy participating in the experiment
    - Strongly agree
    - Agree
    - Somewhat agree
    - Neither agree nor disagree
    - Somewhat disagree
    - Disagree
    - Strongly disagree
17. I felt stressed during the experiment
    - Strongly agree
    - Agree
    - Somewhat agree
    - Neither agree nor disagree
    - Somewhat disagree
    - Disagree
    - Strongly disagree
18. (Only shown if participant is in the Music condition) What kind of music did you listen to during the experiment? For example, what kind of genre, did the music contain lyrics and would you describe the music as relaxing or not?
19. Do you have any additional comments?

Appendix C: Informed consent form

Informed consent form

Thank you for participating in my experiment! In order to answer my research questions, I need to collect some data about you and your results. This consent form is meant to inform you about this. Therefore, please read this document carefully before signing.

The experiment

When programming, reading and comprehending existing pieces of code is essential if you want to develop new functionalities or change existing code to fix bugs. Code comprehension is an important research area, since it is valuable to know how we can influence it.

You will be completing a short experiment in which your code comprehension will be tested and measured. You will be shown a number of short C++ code snippets, which you will have to closely inspect, after which you will have to answer what you think the program will print to the command line. Afterwards, you will be asked to fill in a short questionnaire about you and your experiences during the experiment.

Data collection and processing

During the experiment, we will be measuring two variables about you:

- Which answers you give to the different questions.
- How much time you spend per question.

Using this data, we will construct two measurements of your code comprehension: how many correct answers you give, and the time you take to finish the assignments. Furthermore, we will use the data you provide through the questionnaire to arrive at further conclusions towards understanding code comprehension. All data will be anonymized, and will not be used for any further research. If you fill in your email address, this will be assembled in a list of participants and omitted from the rest of the data.

Termination of participation

Participation in this experiment is entirely voluntary, and participants are free to end their participation at any point during the experiment, without having to specify any reasons for this. Should you choose to not finish the experiment, your data will be deleted and excluded from the study. Should you decide that you want your data to be deleted after the experiment, you can send an email about this to *jacob.jonkman@gmail.com*, after which your data will be deleted.

Name	**Date**	**Signature**
_______________	_______________	_______________