The Importance of Coding Styles within Industries

Stijn Broekhuis
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
s.b.broekhuis@student.utwente.nl

ABSTRACT

There exist common conventions and styles that try to improve the readability and maintainability of software code. It is common for industrial projects to work with these coding practices. Teams adopt or adapt coding styles, and in some cases, they are mandatory. This means coding practices are an integral part of software development. This research will try to find a greater understanding of coding styles within industries. Firstly, a literature review of coding styles with non-semantic consequences and their effects was done. Then, a survey was used to find information about how coding styles are used in companies and finding a difference in the coding styles by looking at the experience. Nearly all participants used coding styles such as Naming Conventions, Brackets and Indention rules.

Keywords

Coding Conventions, Coding Styles, Programming

1. INTRODUCTION

The skills of a good programmer are not something to be taught. Learning to write good code is achieved often by starting with reading good code. Making code readable is, therefore, an important factor in programming. Besides being able to make better code, readable code is essential for programmers that work in a team to work together. As everyone needs to be able to understand what sections of code do without spending too much time. As a result, coding conventions and styles were made. One of the first coding standards is the book "The Elements of Programming Style" [6] written by Kernighan, Brian W. and Plauger, Phillip James. The book showed examples of real programs and how they could be improved. Many of the 'lessons' that Kernighan and Plauger stated, such as; *Say what you mean, simply and directly.* and *Write clearly - don't sacrifice clarity for "efficiency."* are still relevant today.

Coding styles such as the location of a '{' or using spaces around operations, do not have an objective reason why some are better than others. These are often based on preference and what is common in the language. In Java, the naming conventions of variables are *usually* written with `camelCase` [10, 13, 15, 5]. However, in Python this is often `snake_case` [17, 5] which originates from the language C.

Besides different coding styles between languages, many IDEs (Integrated development environment) offer options to change these styles to fit the preference of the user. When a project involves multiple people it can then result in every user having their own coding styles. To solve this problem a there exists a file format called .editorconfig[1]. Many IDEs allow their coding style settings to be overwritten by a file named `.editorconfig` that is stored in a project. Since the file is within a project, all members of the project will have the same .editorconfig file and thus maintain a consistent coding style (because their IDEs will warn or even cause errors if they do not).

To manage and fix bugs within code, it must be readable. It is known that some coding practices affect the readability of code [1, 2, 8] in both positive and negative ways. For a company it is useful to have conventions that improve the readability of their code. Coding styles with non-semantic consequences are styles that do not affect how the code operates, it only affects how code looks. This study focuses on the non-semantic consequences as it interesting to see the reason why they are used. Other styles that have semantic consequences are useful as they can improve the code's speed, testability, maintainability, or debugability. It is expected that developers in industries have higher standards for clean code than a project made by a single person. The goals for this research are as follows:

Goal 1: Develop a greater understanding of the effects of coding conventions and the common differences in popular languages.

Goal 2: Develop a greater understanding of the importance and usage of coding styles with non-semantic consequences in industries.

To achieve these goals, the following research questions (RQ) are used as a basis:

RQ1: In what areas do non-semantic coding styles differ in popular languages such as Java, Python, JavaScript and C#?

RQ2: How do non-semantic coding styles found in **RQ1** influence the quality/readability of the code?

RQ3: How do employees of industries use coding styles found in **RQ1**? Are they aware of the influences found in **RQ2**?

[1] Learn more: https://editorconfig.org/

35th Twente Student Conference on IT July. 2nd, 2021, Enschede, The Netherlands.

This study conducted literature research and a survey to find out how coding styles are generally used. The survey will ask employees what coding styles and conventions they use and their experience and opinion about using them. The results will give insight into the frequency and necessity of coding practices and how they are executed.

2. RELATED WORK

This section contains related work about coding conventions. Coding conventions have been proposed since the 1970s. Well known examples are authors are Kernighan and Plauger. Most papers focused on semantic consequences, although some did partly focused on readability.

In 1974, Kernighan, Brian W. and Plauger, Phillip James [6] published the first edition of *The Elements of Programming Style*. According to them, it has been common for programmers to make unreadable code that was a pain to maintain, so-called 'write-only code', but "...students, teachers, and computing professionals now recognize the importance of readable programs." The book gave many lessons on how to write quality code by using examples of real software and improving them.

In 2016, H. G. Koller [7] wrote in their master thesis about the effects of Clean Code on understandability. Their research showed, unexpectedly, little effect on the understandability when using Clean Code. Using Clean Code only had a positive effect on how long changing current functionality took.

In 2018, R. M. dos Santos and M. A. Gerosa [2] experimented with certain coding practices and the impact on the readability of the code. Their research found that of the 11 assessed coding practices they tested, 8 had evidence that it affected readability.

In 2021, V. Lenarduzzi, V. Nikkola, N. Saarimäki and D. Taibi [9] researched the effects of code quality on pull request acceptance. They wanted to find out if GitHub pull requests had a higher chance of being accepted if the code was clean. They concluded that code quality does **not** affect the pull request acceptance.

3. RESEARCH METHODOLOGIES

Answers to the research questions mentioned above need to be found to achieve the goal of the paper. In Figure 1 is a simple diagram of how the research questions relate.

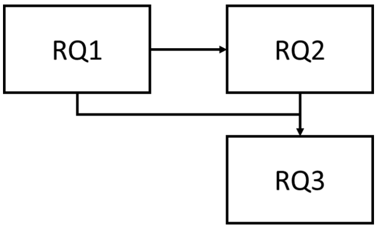

Figure 1. How the research questions correlate.

RQ1 can be answered by firstly defining the certain coding styles options that this study will focus on. These can be found by studying different coding style guides that can be found online. These coding styles might have similarities that overall give a better understanding of the code conventions and styles. For example, it is common for different languages to have a naming convention. A subset of these coding styles will be chosen to research their effects.

A literature review will be done on the effects of the coding styles found in RQ1. Searching for papers that state effects of individual coding styles. This will give an understanding of the importance of individual coding conventions and styles. Answering this question allows the research to continue with the next and final research question.

RQ3 will be answered with a survey that will be given to employees of different industries using Google Forms. The survey will have three sections. Firstly, general questions about what type of work they work in, how long they have worked, and what language they work in. Secondly, this section will focus on questions about coding styles. These coding styles will not be connected to specific languages. Thirdly, there will be a section about coding conventions and their effect on the quality of code according to the participants. Finally, with the answers found in the research questions, the goals will be completed and a conclusion for each of them can be made.

4. THE CODING STYLES

To find common areas that coding styles differ, one has to look at coding style guides. The research found coding guidelines by using the search engines DuckDuckGo and Google and using the terms "Coding Style Guide" and "Coding Conventions Guides". Six guides were chosen. These were based on the most used languages [4]. Using six guidelines gives an accurate representation of popular languages and what areas coding styles are often used.

Google Style Guides

The Google Styles Guides [5] are a collection of style guides that are used by Google and serve as a guide for programmers within the company who add to Google's code or for external programmers who wish to create a project that has the same style as Google. Because this is a collection of style guides, the study looked at the style guide of popular languages[4]. Unlike humans, the style guides have similar reasons for their existence. They help programmers to make code readable and/or look better. However, these are not rules, one is not required to follow them. If there is a scenario where violating the guidelines results in better code, then these rules do not apply. Additionally, style guides *might* be wrong on certain rules and forcing is therefore not a favourable idea.

Style Guide for Python Code 2001

The Style Guide for Python Code [17] is part of the Python Enhancement Proposals(PEP) with number 8 and is created by the community. The Style Guides makes it clear that its goal is to increase consistency and improve readability. This style guide also states that just as The Elements of Programming Style, [6], that code is more read than written. Just like the conventions in the Google Style Guides, they are more like guidelines, really.

Sun MS' Java Coding Style Guide 2000

The Java Coding Style Guide by Sun MicroSystems [15] is created by A. Reddy who worked/works at Sun MicroSystems. Unlike the Google Style Guides, this style guide was created for all Java developers and accepted feedback from anyone. The paper states that the importance and benefits of a consistent coding style are well known, although I wished it would cite a source for this as it would make my life easier. The guide was created from studying existing java code, such as the JDK code and C++ code. Again, Reddy makes it clear that users of the guide need to see the rules as guidelines.

MISRA-C 2004

The MISRA-C 2004 [11] is a coding conventions guide created by MISRA. It is an industry-standard when working with the language C. The original coding conventions

Table 1. Style Guides Mentioning a Style Convention.

Mentions Style Convention	*Google Style Guides*	*Style Guide Python Code*	*Sun MicroSystems Java Coding Style Guide*	*MISRA-C 2004*	*Sun MicroSystems Java Coding Convention*	*StandardJS Coding Style*	*.Net C# Coding Style*	*C Style: Standards and Guidelines*	*Total*
Naming Convention	✓	✓	✓	✓	✓	✓	✓	✓	8
Bracket	✓		✓	✓	✓	✓	✓	✓	7
Indentation	✓	✓	✓		✓	✓	✓	✓	7
File Name	✓		✓		✓		✓	✓	5
Column Limit	✓	✓	✓		✓			✓	5
White space	✓	✓	✓	✓	✓	✓	✓	✓	8
Structure of a file	✓		✓		✓		✓	✓	5
Documentation	✓		✓	✓	✓			✓	5
File Organization	✓		✓		✓		✓	✓	5
Comments	✓	✓	✓	✓	✓	✓		✓	7

guide was in 1998 was created to help the automotive industry in the UK. The reach of the guidelines was greater than expected as other industries started using it and a revised version was created in 2004. The company did not stop making newer versions, but this research paper chooses to look at the 2004 version.

Sun MS' Java Coding Conventions 1997
The Java Code Conventions [10] were created in 1997. It was based on the Java Languages Specification from Sun MicroSystems. Note that this is different from the one motioned above, although it is otherwise very similar. It again states that coding conventions are important to programmers without giving a citation.

StandardJS Coding Style
StandardJS is a coding style guide, linter and formatter [16] for the programming language JavaScript. It can be added to projects to check if code follows their style. The style guide was created to stop people from discussing coding styles and get work done. Although there are ways to customize it, their intention is to stop programmers from doing so.

.Net C# Coding Style
Finally there is .Net C# Coding Style [14] is a style guide that is used for .Net and C# code. Their style guide used as the default style settings for the IDE Visual Studio.

C Style: Standards and Guidelines
This book was originally published in 1991 by D. Straker. He was frustrated with code that was not readable in his opinion. He wanted code to be better and wrote a book. This book is now available for free online[2].

Results
Common areas in each guide were marked down. Whether an area was mentioned in another guide is was noted. Six common areas were found, as seen in Table 1, these were chosen to continue working with. The study did not use *file organization* and *documentation* as it was used less and had no implication to writing code or was similar to another style. *Structure of a file* was later omitted due to the lack of effects found.

Naming Conventions
These explain how variables, class-names, methods, etc. should be formulated. This includes what characters can be used and the structure of them.

```
private exampleName(){}
// or
private _example_name(){}
```

Brackets and Indentations
Some languages use brackets to specify sections of code. Examples of these are *Allman* and *K&R.*

```
if (x)
{
    // Allman
}
if (x) {
    // K&R
}
```

Indentation style decides how blocks of code contain more white space behind the deeper the code goes. Indentation can be done via tabs or spaces. Because brackets and indents are very closely connected, they will be combined for the research.

Column Limit
The column limit defines how far code lines can go, this does not stop programmers from crossing this line but removes the frequency of having to scroll sideways to read code. The column limit is often 80 or 120 characters. Note that programmers often use a font that is mono-space, meaning that all characters have the same width.

[2] http://synque.com/cstyle/index.htm

White Space

White space in this research is defined as whether certain characters need to have space before and/or after.

```
function(int a, boolean b){
    x = (a*3)+3; // No spaces
}
function (int a,boolean b) {
    x = (a * 3) + 3; // With Spaces
}
```

Comments

Comments are notes written by the programmer that are not part of the documentation. They are there to explain what the code does in the place it was written. The orange text in the examples above are comments, they do not affect code.

5. THE EFFECTS

As previously stated, to learn the effect of the coding styles found in the previous section, a literature study will be held. Using FindUT and Google Scholar and the terms "*Coding Style*" "*Readability*" "*Coding Conventions*", information about the effect of coding styles were searched. All of them needed to test coding styles and specify what coding styles were being tested. This was needed to find individual effects of each coding style. Finding applicable papers that clearly stated effects of coding styles with non-semantic consequences was difficult and this resulted in only 4 paper to work with.

Impacts of Coding Practices on Readability

Written by R. M. dos Santos and M. A. Gerosa, this paper studied the effect of coding styles on readability by showing participants two pieces of code. Participants were asked to select the more readable code. Results of different coding styles could then be measured.

A Study of Different Coding Styles Affecting Code Readability

Written by T. Lee and J. B. Lee and H. P. In, this paper tested if the effects of violating coding conventions affected the readability of code using Coding Conventions from CheckStyle and PMD.

Using a coding standard to improve program quality

Written by X. Fang, this paper analyzed different java projects that either used or did not use a coding style. Their effects were shown in graphs that displayed the frequency of comments in a file and the number of files based on file size.

Assessing the Value of Coding Standards: An Empirical Study

Written by C. Boogerd and L. Moonen, this paper evaluated the effects of violating MISRA-C rules and the actual faults of programs.

Results

The results of these papers were analyzed to find the effects of the coding styles found in RQ1. Unfortunately, the effects of commenting were limited. The results are shown in Table 2. This table shows if a coding style increased or decreased the readability/quality of code. The rows are different coding styles while the columns show the different papers. The cells then show how the coding style either improved or decreased the readability and quality of the code. Despite the limited findings, the table implies that coding styles are not universally positive.

6. THE SURVEY

6.1 Survey Structure

Using the coding styles found in RQ1 and effects in RQ2 a survey was made. This survey was made in Google Forms. The survey was split into 3 sections. These were: General Information, Coding Styles, and Coding Styles Effects.

Before the first question, information was given to the participants about this study. This includes what this study sees as coding styles and conditions of the survey. The survey stated that it does not collect emails, the data is anonymous and is used for research. It also contained an email address for questions.

General Information

This section had some general questions about the participant for a better insight into what type of participants had filled in the survey.

QA1: What best fits your current role in your field?

This was a multiple-choice question (Project Manager, Developer/Programmer, Tester). It gave a better insight into what type of worker filled in the survey.

QA2: How long is your experience in programming?

A multiple-choice question that had the options: 0-2, 3-5, 6-9 or 10+ years. If a person has worked for a longer time it gives a better indication of their experience with working with coding styles.

QA3: What is the primary focus of your work?

Options for this question were: *Manufacturing*, *Defence & Security*, *General Software Development*, *Web Development & Business/IT*, and *Healthcare.* These were chosen as they were common fields in Development. This question will give a better insight into how these industries differ. Additionally, an 'Other...' option was given if none of the fields above applied.

QA4: Do you work in the following languages?

As the study has learned in RQ1 often languages have their own standards, this question tells information about what coding styles the participant might be familiar too. For this question multiple answers are possible. The languages that were available to choose from were: *C (++)*, *C#*, *Java*, *JavaScript / TypeScript*, *Python* and *'Other...'*. Participants were asked not to write down 'Other' of any other language applied above.

QA5: What IDE do you use?

Many IDEs have built-in coding styles, therefore it is insightful to know what IDE a participant uses. This question allows for multiple answers, and the options were: *JetBrains (IntelliJ, PyCharm, WebStorm, CLion, etc.)*, *Visual Studio (Code)*, *Eclipse*, *Notepad++* and *'Other...'*. Again, *'Other...'* should only be filled in if none applied.

Coding Styles

This section is about whether the participants use coding styles, whey and how they use coding styles or if they that they do not use coding styles, why they do not.

QB1: Do you work with a coding style

This question has 4 options; "*Yes, I have a coding style from work.*", "*Yes, I have my own coding style.*", "*Yes, I work with the default coding style of my IDE.*" and "*No, I do not work with a coding style.*" For these questions, all answered 'yes' skipped the next question, participants that said 'no' were transported to the next question.

QB2: Do you have any reason for not using a coding style?

As not using a coding style is very interesting, it is an optional open question to allow full explanation by the

Table 2. The effects of Coding Style on Readability and Quality [2, 12, 3, 1]

Coding Style	Impacts of Coding Practices on Readability	A Study of Different Coding Style Affecting Code Readability	Using a coding standard to improve program quality	Assessing the Value of Coding Standards: An Empirical Study
No Specific Style			**Improved**: Using these coding styles reduced the number of comments that were written in the code.	
Naming Convention	**Improved**: Use of dictionary words.	**Improved**: Using a constant naming convention.		**Decreased**: Limiting characters for identifiers to 31. Inner scopes using the same name for identifiers as outer scope.
Brackets and Indentations	**Improved**: A blank line after curly braces. **Decreased**: Opening braces on the same line as clause.	**Improved**: Using spaces instead of tabs. **Decreased**: Opening braces on the same line as clause.		
Column Limit	**Improved**: A 80 chars column limit			
White Space	**Indifferent**: Blank lines separating related instructions	**Decreased**: Using a white-space after something, such as a comma or statement.		
Comments				**Improved**: No /* inside comments.

participant. Afterwards, the participant is sent to the section called '*Effects of Coding Styles*'. Even though these participants do not use a coding style, they might know about the effects.

QB3: Does your coding style enforce:
Following a list of the coding styles found in RQ1 (Indentations & braces, naming conventions, white-spaces, comments, and column limits). For each coding style, it is asked to say *Yes*, *No* or *N/A (Not Applicable)*.

QB4: Is there another major aspect that was not mentioned in the previous question?
This optional question allows participants to name a coding style that they think was important and that the survey did not mention.

QB5: Do you follow the coding styles?
This yes or no question tells if participants take the coding styles seriously at all.

QB6: Could you explain why?
Following the previous question, this optional question gives the participant a chance to explain their reason for using coding styles before the survey asks about the effects found in RQ2.

QB7: Does your team use a tool for coding styles?
This multiple-choice question has three options: "*Yes, a mandatory tool*", "*Yes, a recommended tool*", and "*No, leave it up to the team*" This question gives insight into the freedom employees have when working with coding styles.

QB8: If yes, what tool?
This optional question is self-explanatory.

Effects of Coding Styles
This final section of the survey is about the effects of coding styles found in RQ2.

QC1: Do you agree with the following statements?
This question has the following statements: "*Coding styles are an important part of programming*", "*Following code styles are required where I work*", "*Coding styles make my code easier to read*", "*Coding styles make someone else's code easier to read*", "*Coding styles reduce the amount of comments in code*", and finally "*What coding style isn't important. Just using one is*". This last question was added to determine participants find, for example, using Allman brackets are better than K&R. For each statement participants can choose for "*Completely Agree*", "*Agree*", "*Disagree*", and "*Completely Disagree*". A "Neutral" option was omitted as this forces users to make a choice.

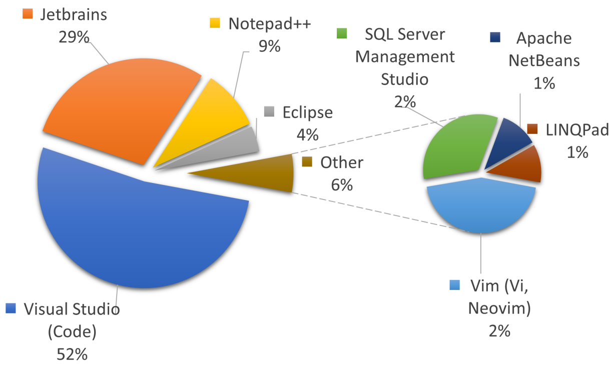

Figure 2. IDEs that participants use.

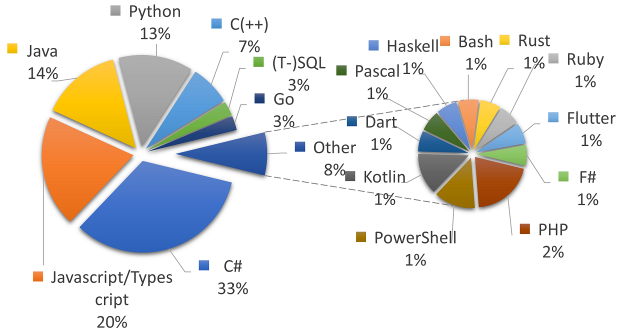

Figure 3. Languages of the participants.

6.2 Results

General Information

For this survey, we contacted several companies that have worked with the University of Twente before. We send out a request to send forward a survey to developers and programmers. The survey was online for 8 days before we closed it. The survey got 102 responses (95 Developers, 5 Project Managers, 2 Testers). The participants had an experience ratio of 6:30:23:42 for 0-2:3-5:6-9:10+ years. As seen in Figure 4, 97% of participants stated that they were using a coding style while programming. Following from Figure 2 and Figure 3, the most used language and IDE were C# and, unsurprisingly Visual Studio or Visual Studio Code. To the person who is still using Pascal, I hope you are doing well. 97% of participants that used C# used Visual Studio (Code) and 93% of participants that used Java used a JetBrains IDE. The results also found that only 42% of participants that do not work with C# used Visual Studio (Code). In contrast of 19% of participants that did not work in Java used a JetBrains IDE. Do note that this calculation does imply count preference. Unfortunately, question QA3 was not understood correctly. This resulted in participants answering what their job is within a company while the question wanted to know what their company focuses on. The results of this question could thus not be trusted or used.

Coding Styles Usage

From the 3% that did not use a coding style, none elaborated on why. In Figure 5, it shows that industries are likely to require using a tool for coding styles.

The coding styles that we found in RQ1, Figure 6 shows that almost all participants use some coding style that involves indentations, braces, brackets and parentheses. However, commenting and column limits were less common.

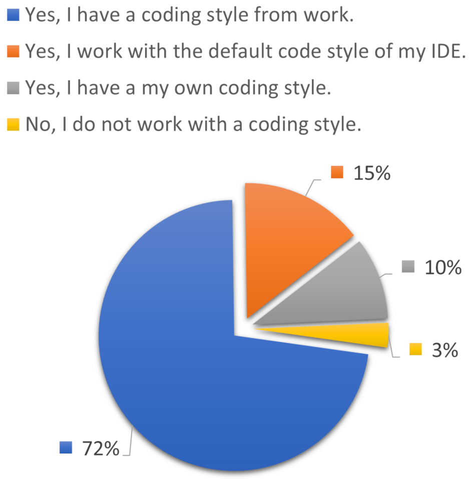

Figure 4. Participant answers to: Do you work with a coding style?

When asked why participants follow coding styles, the answers were processed. Non-ASCII characters were removed and everything was transformed to lowercase. Then all punctuation, numbers were removed and the answered were stemmed. This means that words such as "argued" and "arguing" became "argue". Then stop-words such as "the" and "a" were removed in addition to white-space around words. Finally, because this word was used a lot more than others, the word 'code' was removed, as it does not give additional insight. The frequencies of the words were then calculated and a word cloud was made, this can be seen in Figure 7. Noticeable words here are; "Enforced", "Readable", "Consistent", "Maintain".

The tools participants used were also analyzed and can be seen in Figure 8. This was done by filtering answers by hand and counting the frequency of each tool. ReSharper, a tool for Visual Studio and made by JetBrains, was the most commonly used tool. In the introduction, .editorconfig was mentioned as a way for programmers to sync the style settings using a .editorconfig file. These files overwrite the style settings of an IDE. Interestingly, .editorconfig was referenced in 9% of responses. A lot of unique tools were named, tools that were called more than once are in the other 8%.

Coding Styles Effect

In Figure 9, the effects found in RQ2 were asked to the participants. Participants overwhelmingly agreed that coding styles make code easier to read for both oneself and another. However, there was a disagreement whether code styles reduced comments in code as X. Fang's paper seems to suggest [3]. Participants mostly agreed that "What coding styles isn't important. Just using one is". However, almost 30% did not agree with this statement. Implying that they find some coding styles conventions are better than others. For the question "What coding style isn't important.." 71.5% agree with the statement. However, this percentage seems to increase with experience. 3-5 years participants agreed with 66.6%, 6-9 years agreed with 73.9% and 10+ agreed with 76.7%. 0-2 years had too few participants (6) to conclude from.

Finally, participants were asked if they had anything to

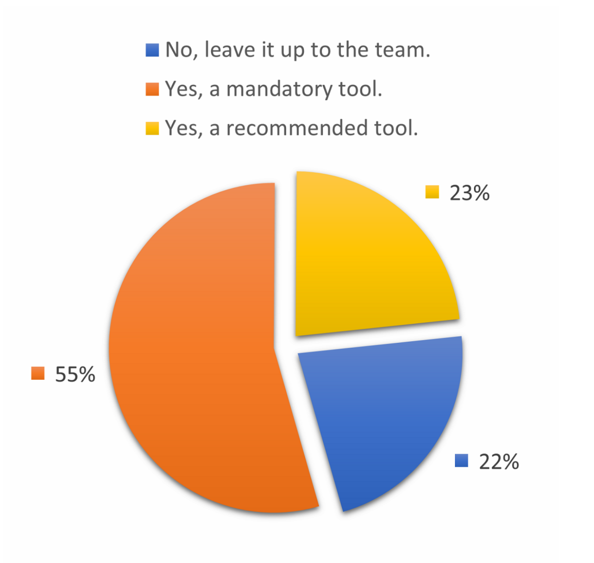

Figure 5. Participant answers to: Are coding styles required with a tool?

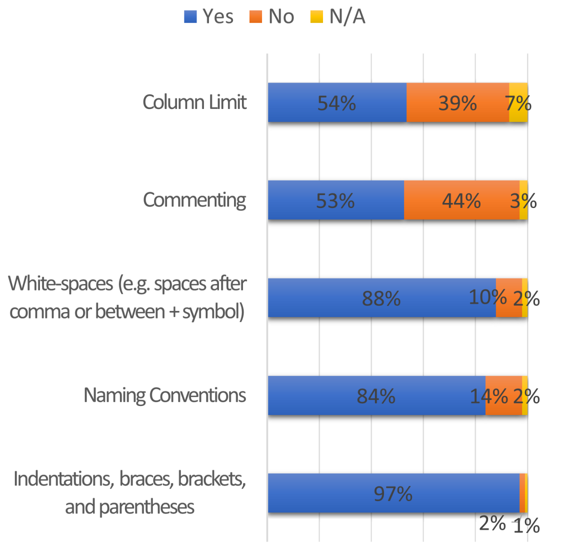

Figure 6. Participant answers to: Does your coding style follow?

add. Coding styles were often different per project. Besides some participants noticed that the study focused on coding styles with non-semantic consequences, there were not any interesting answers.

7. DISCUSSION

In this section, the results of the study will be discussed. Firstly, the goal was to develop a better understanding of the effects of coding styles and the common differences in popular programming languages. The literature study found that coding styles often have rules about indentation, brackets, comments, naming conventions and column limits. The effect of these areas was limited as most increased and decreased readability and quality. This could be because the topic is subjective. I hoped there were more effects found that could be tested during the survey. The survey showed that >90% of participants used coding styles for work. It is, therefore, reasonable to conclude that they have a critical role in industries. This could imply the necessity of teaching these coding styles to students.

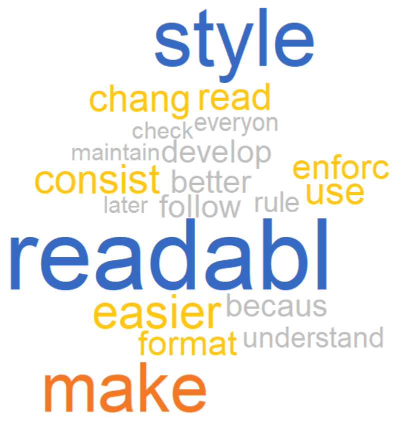

Figure 7. Top 20 word used in answers on why to use coding styles. The frequency of the word increases the size. Asked to participants before mentioning to participants about effect the study found.

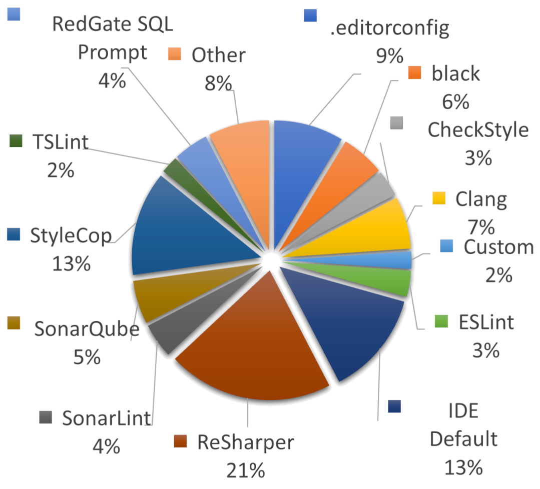

Figure 8. Participant answers to: What tool do you use for coding styles?

Originally, the study had a different third research question, but because of the misunderstanding of question QA3, it had to be changed as it could no longer be trusted to create a conclusion. From the survey, information about the usage of IDEs, programming languages, and tools of the participants was found. The answers were diverse as not a single IDE, language, or tool was exceptionally more popular than the rest. For IDEs, almost all C# programmers use Visual Studio (Code), and almost all Java users use a JetBrains product.

Results seem to suggest that with increased experience programmers seem to agree that using any coding style is important and not a specific one. As they were more exposed to different coding styles, they realized that the statement was true, otherwise, they would have solidified a preference for a certain coding style. The other statements did not show any correlation between experience and agreement. From this, I developed a better understanding of the importance and usage of coding styles with non-semantic consequences in industries. This means that

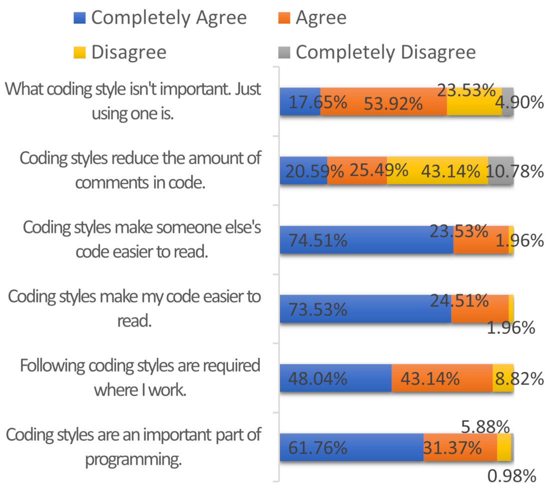

Figure 9. Participant answers to: Do you agree with the following statements?

the second goal of the study was achieved.

Due to limitations, the participants were recruited from companies that are in the region of the University of Twente. Consequently, they may not represent development worldwide. The study was also limited by time, RQ1 focused on coding styles in general, opposed to finding a difference between languages. Further research could be done on the effects of coding styles with non-semantic consequences or the difference between languages. Additionally, since this study failed to do so, further research could look for a difference in coding styles between different industries such as Healthcare, Finance, Security and more.

8. CONCLUSION

In conclusion, many coding styles with non-semantic consequences have similar areas between languages. Many inform users on indentation, brackets, comments, naming conventions and column limits as seen in Table 1 and the participants of the survey had similar areas. The effects of these coding styles were inconclusive as effects can decrease and increase code quality and readability. The survey, however, found that coding styles serve an important role in programming industries independent of the language.

Almost all participants used a coding style, nearly half of them were given from work. For the effects of coding styles, participants agree that coding styles do increase readability but are indifferent whether it decreases commenting frequency. Programmers with more experience were also more likely to state that coding styles, in general, were important, and not a specific one.

9. REFERENCES

[1] C. Boogerd and L. Moonen. Assessing the value of coding standards: An empirical study. In *2008 IEEE International Conference on Software Maintenance*, pages 277–286, 2008. doi:10.1109/ICSM.2008.4658076.

[2] R. M. dos Santos and M. A. Gerosa. Impacts of coding practices on readability. *Proceedings - International Conference on Software Engineering*, pages 277–285, 2018. doi:10.1145/3196321.3196342.

[3] X. Fang. Using a coding standard to improve program quality. In *Proceedings Second Asia-Pacific Conference on Quality Software*, pages 73–78, 2001. doi:10.1109/APAQS.2001.990004.

[4] Github, 2020. URL: https://octoverse.github.com/.

[5] Google. Google style guide. URL: https://google.github.io/styleguide/.

[6] B. W. Kernighan and P. J. Plauger. *The elements of programming style*. McGraw-Hill, New York, 2nd ed. edition, 1978.

[7] H. G. Koller. Effects of clean code on understandability: An experiment and analysis. In *Effects of Clean Code on Understandability: An Experiment and Analysis*, 2016.

[8] T. Lee, J. B. Lee, and H. P. In. A study of different coding styles affecting code readability. *International Journal of Software Engineering and Its Applications*, 7:413–422, 09 2013. doi:10.14257/ijseia.2013.7.5.36.

[9] V. Lenarduzzi, V. Nikkola, N. Saarimäki, and D. Taibi. Does code quality affect pull request acceptance? an empirical study. *Journal of Systems and Software*, 171:110806, 2021. URL: https://www.sciencedirect.com/science/article/pii/S0164121220302090, doi:10.1016/j.jss.2020.110806.

[10] Sun Microsystems. Java code conventions, 1997. URL: https://www.oracle.com/technetwork/java/codeconventions-150003.pdf.

[11] MISRA. Guidelines for the use of the c language in critical systems, 2004. URL: http://caxapa.ru/thumbs/468328/misra-c-2004.pdf.

[12] P. W. Oman and C. R. Cook. A paradigm for programming style research. *SIGPLAN Not.*, 23(12):69–78, December 1988. doi:10.1145/57669.57675.

[13] J. Piater. A guide to coding style, 2005.

[14] .Net Platform. C# coding style, 2021. URL: https://github.com/dotnet/runtime/blob/c86d1905c3ad439820b6fd8d9fe50a1690dfb1c1/docs/coding-guidelines/coding-style.md.

[15] A. Reddy. Java™ coding style guide. *Sun MicroSystems*, 2000.

[16] StandardJS. Javascript standard style, 2021. URL: https://standardjs.com/.

[17] G. van Rossum, B. Warsaw, and N. Coghlan. Pep 8 – style guide for python code, 01-Aug-2013 05-Jul-2001. URL: https://www.python.org/dev/peps/pep-0008/.