The effect of Python Version Upgrades on the Compilability of Code Snippets posted on Stack Overflow

Shiyu Yang[1,a)] Tetsuya Kanda[1,b)] Katsuro Inoue[2,c)]

Abstract: Stack Overflow(SO) is a Q&A site for programmers that have accumulated a wealth of code snippets. In 2008, Python 3.0 was released, a new language version not backward compatible with Python 2. This means that code snippets written in Python 2 may not be compiled directly by Python 3. This issue may affect the compilability of Python code snippets on SO. In this work, we use the Python compliance analyzer, PyComply, to parse Python code snippets on SO. The Python code snippets used for this study were from SOTorrent, an open dataset based on the official SO data dump. Using the PyComply parsing results, we investigate the effect of the Python version upgrades on the compilability of Python code snippets on SO. We found that Python version upgrades affected the compilability of Python code snippets on SO, with evidence that the release of the new Python version inhibits the development of the old versions and the trend of code snippets responding to newer versions increases over time.

1. Introduction

Stack Overflow is a Q&A site for programmers which provides a platform for programmers to exchange information and share knowledge. Users can post questions, answer questions, and search for content that interests them on Stack Overflow. Users usually describe their questions or answers by attaching code snippets. As of June 2022[*1], Stack Overflow has accumulated over 23 million questions and 34 million answers. These numbers are increasing every day.

These questions and answers on Stack Overflow accumulate a large number of code snippets. This vast amount of ready-to-use code snippets provides developers with an easy way to find solutions to daily programming problems. Nowadays, copying code examples from Stack Overflow is common [2].

While searching for the required code snippets on Stack Overflow is convenient, recent studies have shown that code snippets can be toxic [7], obsolete [11] [12] as well as low-quality [10] and lead to software quality issues [11] [5], license violations [1] or migration of security vulnerabilities [4] [9].

There are many reasons why code snippets on Stack Overflow are problematic. One is due to the use of outdated programming language features in the code snippet.

Programming languages are not set in stone. Existing programming languages constantly evolve to meet new needs and thus gain longevity. Popular programming languages often use versions to indicate their evolution, with newer versions usually representing more mature forms of the language. Many popular programming languages have backward compatibility, which means that programs compiled with an earlier language version can be compiled with a later version and exhibit the same behavior as the previous version. However, the release of Python 3.0 made the Python language break this rule; Python 3 is not backward compatible with Python 2.

The lack of backward compatibility for Python 3 can cause problems for Stack Overflow users. For example, a user found a code snippet that meets his/her needs and wants to reuse it in his/her project. However, this code snippet is written in Python 2 and may not be compilable if used directly in the user's project written in Python 3.

To our knowledge, no studies have investigated the Python language versions of Python snippets on Stack Overflow. We are interested in investigating the compilability of Python code snippets on Stack Overflow for each Python version, and how Python version upgrades affect the compilability of Python code snippets. Knowledge of the compilability of Python code snippets may provide insight into new research directions and tool support.

In this paper, we extracted 2,475,559 code snippets from SOTorrent whose post tag contained "Python" and filtered 307,788 code snippets whose compilability changed with the Python version for research. We used a Python compliance analyzer, PyComply [6], to parse the code snippets we studied to understand the effect of compilability of Python code snippets on Stack Overflow due to Python version upgrades. We organized our study by answering the following research questions:

- **RQ1:What are the available Python version ranges for Python code snippets on Stack Overflow?**
  The code snippets whose available version range spans Python 2 and Python 3 account for 42.0%, those whose avail-

[1] Osaka University
[2] Nanzan University
[a)] yangsy@ist.osaka-u.ac.jp
[b)] t-kanda@ist.osaka-u.ac.jp
[c)] inoue599@nanzan-u.ac.jp
[*1] https://data.stackexchange.com/

able version range is only in Python 2 account for 39.6%, and those whose available version range is only in Python 3 account for 18.4%.

- **RQ2:How Stack Overflow users respond to each Python release, and what are the differences between Python 2 and Python 3?**
  We found that new versions are released and get a response shortly afterward. The release of the new Python version inhibits Stack Overflow users' response to older versions. The existing data only allow us to explore the differences between the user responses of Python 2 and Python 3 versions from a partially feasible perspective.
- **RQ3:How are Python 2-only compilable Python code snippets and Python 3-only compilable Python code snippets distributed on Stack Overflow?**
  The number of Python 2-only compilable Python code snippets increased year by year starting in 2008, peaked in 2015, and began to decline year after year. The number of Python 3-only compilable Python code snippets has generally trended upward. Nevertheless, among the code snippets investigated by RQ3, the Python 2-only compilable Python code snippets are about twice as large as the Python 3-only compilable Python code snippets.

The rest of the paper is organized as follows. Section 2 presents the background. Section 3 introduces our study approach. Section 4 presents the results of our research questions. Section 5 discusses the implications and limitations of our study. Finally, Section 6 concludes the paper.

2. Background

2.1 Stack Overflow

Stack Overflow is a popular Q&A site for programmers. Developers can post questions, answer questions, and search and browse for the content of their interest on the site.

Original posters use tags to describe the topic of the question[*2]. Tags are a means of connecting experts with questions they will be able to answer by sorting questions into specific, well-defined categories. Each question must have at least one tag and can contain a maximum of 5 tags. Original posters need to specify tags when creating questions. In our data collection process in Section 3, we use tags to determine the data we need to extract. Figure 1 shows an example of a post on Stack Overflow[*3]. The original poster asked how to access the index in 'for' loops, and attached three tags to describe the topic of the question.

When posting a question on Stack Overflow, the original poster can paste a code snippet related to the question into the body of the post to describe their question in detail. Each question can receive multiple answers from different answerers. Answerers can also post answers with code snippets to explain their answers. As shown in Figure 1, code snippets are added to the question and the answer for detailed descriptions.

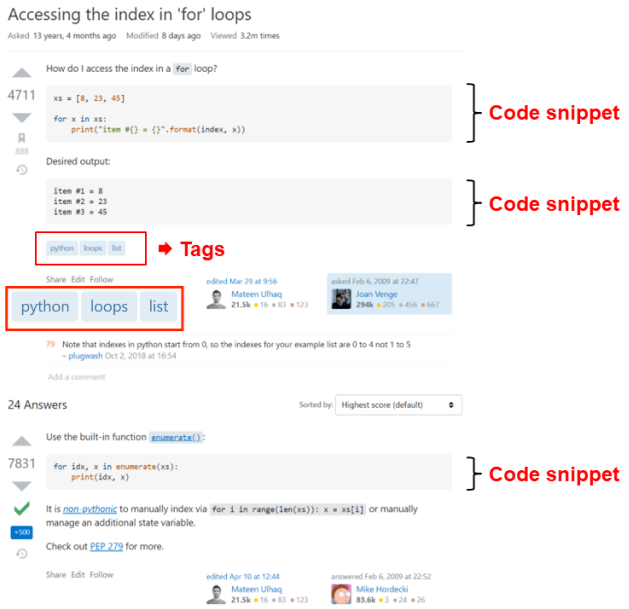


Fig. 1 An example of a question on Stack Overflow

2.2 Python Language Evolution and Backward Compatibility

Programming languages need to evolve in response to external and internal factors continually. Without evolution, the language may become less competitive and even be phased out. Versions are a common way for popular programming languages to express evolution, and later versions usually represent a more mature form of the language. Many popular programming languages have maintained backward compatibility during their evolution to avoid raising concerns among developers about software product compatibility, meaning that software developed in an earlier version of the language can be compiled with a later version and express the same behavior as the previous version.

However, Python is one of the notable exceptions that breaks the backward compatibility rule. Python 3, starting with 3.0 are not backward-compatible with Python 2 from 2.0 to 2.7. In other words, programs developed in Python 2 may not be compiled by Python 3 without modification.

Starting with Python 2.0, released in October 2000, the Python language continued its linear development until Python 2.5 [6]. Development then branched out, with Python 2.6 and Python 3.0 being released in October and December 2008, respectively. Python 3.0 is not backward compatible with previous Python versions. Despite the Python 3.0, the Python team decided to support both development branches due to the large number of users who continued to use Python 2. Since then, versions of the Python language have been split into two series.

However developing Python 2 and Python 3 at the same time makes it hard to improve Python. There are improvements Python 2 can't handle. So Python officials decided to phase out Python 2 and work on making Python 3 better and faster. This way they can help Python users by improving Python faster.The Python community has decided to discontinue support for Python 2. Official support for Python 2.7 ended on January 1, 2020. This marks the end-of-life of Python 2 [13].

[*2] https://stackoverflow.com/help/tagging
[*3] https://stackoverflow.com/questions/522563

As mentioned before, many code snippets have been accumulated on Stack Overflow. There may be code snippets written in various Python versions among these code snippets. Because official support for Python 2 has been discontinued,code snippets on Stack Overflow written in Python 2 are at risk of becoming obsolete if they are not modified to be compilable for Python 3. Therefore, it is necessary to investigate the Python code snippets on Stack Overflow to determine their compilability for each Python version.

2.3 PyComply

To investigate the impact of the transition from Python 2 to Python 3 on Python applications, Malloy et al. [6] developed a Python compliance analyzer, PyComply. It is based on an approach that exploits grammar convergence to generate parsers for each of the major versions in Python 2 and Python 3 and conducted an empirical study on the Qualitas corpus, a selection of Python applications.

Input to PyComply is the Python grammar for the version under study together with a Python program or test case; output from the tool includes the statistical information such as pass/fail result for each file or the number of Python 3 features that PyComply recognized. The core of PyComply is the grammar formalism used to define the Python syntax, along with the parser actions inserted into the grammar to facilitate identifying the Python 3 features.

The Python developers make a test suite available for each Python version. In addition to using correctness preserving grammar transformations to build their parsers, they also validated the parsers by comparing the number of test cases that their PyComply parsers pass with the number of test cases that the Python parsers passed numbers were the same. Moreover, the fact that their parsers recognize the same test cases that the Python parsers recognize substantiates the validity of their investigation.

3. Study Approach

3.1 Data Collection

To understand the effect of Python version upgrades on the compilability of python code snippets on Stack Overflow, we first need to obtain code snippets written in Python from the questions and answers posted on Stack Overflow.

As we introduced in Section 2, when an original poster posts a question on Stack Overflow, tags are added to describe the topic of the question. In addition, the content of the now posted code snippet may vary from that of the initial posting due to editing. We focused on the most recent posted data currently available to users to investigate the compilability of the current code snippet. Based on these, we used the following two criteria to identify the code snippets required for this study:

(1) Code snippets from posts containing the tag “Python”.
(2) Posting data is up to date.

We use SOTorrent to extract Python code snippets on Stack Overflow, which is available at Zenodo[*4]. SOTorrent is an open dataset based on the official Stack Overflow data dump, which provides access to the version history of Stack Overflow content at the level of whole question posts and individual text or code blocks [3]. SOTorrent has been continuously updated with many versions since its creation. When we started our study, the latest version of SoTorrent was SOTorrent20_03 as of March 15, 2020. Based on our criteria above, we ended up with 2,475,559 code snippets. For the subsequent PyComply parsing, we dealt with formatting issues that prevent code snippets from being parsed by PyComply, such as redundant indentation, and inconsistent indentation.

[*4] https://zenodo.org/record/3746061#.YdpzI2jP02w

3.2 Code Snippet Analysis

This subsection introduces how to use PyComply to parse a Python code snippet to identify its Python version. We present code snippet analysis in two parts, preprocessing and PyComply parsing.

3.2.1 Preprocessing

Before the PyComply parsing, we need to build the corresponding PyComply for all Python 2 and Python 3 versions investigated in this study.

When Malloy et al. [6] ran PyComply for the eight major releases in Python 2 and the seven major releases in Python 3 and examined the pass rates for the applications, they found that the pass rates for versions 2.1, 2.3, 3.2, and 3.4 were the same as the corresponding previous version for all applications. To investigate whether a similar situation might arise when parsing Python code snippets on Stack Overflow with PyComply, we performed a PyComply parsing test. We ran PyComply for the same eight major releases in Python 2 and the seven major releases in Python 3. When we examined the pass rates of Python code snippets, we found that the pass rates for versions 2.1, 2.3, 3.2, and 3.4 were the same as the corresponding previous version for all Python code snippets. This result is consistent with the findings of Malloy et al. [6]. Therefore, we omitted these versions in this study, showing only six major releases in Python 2 and five major releases in Python 3.

3.2.2 PyComply parsing

In this study, a code snippet is considered compilable for a Python version if it can pass PyComply parsing for that Python version. Otherwise, the code snippet is deemed uncompilable for that Python version.

We parsed the previously obtained 2,475,559 code snippets with each Python version of PyComply and recorded the pass/fail results for each snippet. Based on the parsing results, we found that 1,173,905 code snippets failed to be parsed by all Python versions of PyComply, i.e., these code snippets are uncompilable for all Python versions. There are many reasons why these code snippets are uncompilable for all Python versions, such as programming errors, and pseudocode. In other words, they are not caused by the Python version upgrade we would like to investigate. Therefore, we removed these code snippets from this study. In addition, we found that 993,866 code snippets could pass the parsing of PyComply for all Python versions, i.e., these code snippets were compilable for all Python versions. This may be the case because they are not using some features that would affect the compilability of code snippets vary between Python ver-

sions. While these code snippets are helpful for users, this study focuses on code snippets that are (only) compilable in certain versions, i.e., code snippets whose compilability changes depending on the version of Python. We believe that excluding code snippets compilable for all Python versions can make the changes in the data clearer when studying specific research questions. Therefore, we removed these code snippets. Finally, the remaining 307,788 code snippets were the subject of our study.

The pass rates for the remaining 307,788 code snippets are shown in Table 1. The first column of Table 1 is the posting year of the Python code snippets. One row for each year, and one column for each Python version, ranging from 2.0 to 2.7 in Python 2 and from 3.0 to 3.6 in Python 3. The last column shows the number of Python code snippets each year. The data in all but the first and last columns show the percentage of these Python code snippets that passed each version of PyComply in the corresponding year.

3.3 Case Study

As described in Section 2, in 2008, the successive release of Python 2.6 and Python 3.0 started the path of branching Python versions. Therefore, when investigating the research questions, we analyzed the timelines of Python 2 and Python 3 separately to avoid the interference caused by the overlap in the release schedule of Python 2 and Python 3.

In addition, we used the same dataset when investigating RQ1, RQ2, and RQ3. We obtained 307,788 code snippets by filtering out code snippets that are compilable for all Python versions and uncompilable for all Python versions, as mentioned before.

3.3.1 RQ1:What are the available Python version ranges for Python code snippets on Stack Overflow?

This study would like to investigate the lifespan of code snippets on Stack Overflow that are uncompilable in some Python versions. The available python version ranges for Python code snippets may somewhat represent the survival period of code snippets.

One thing to note about the available python version ranges for Python code snippets is that the available python version ranges may not be continuous but are subject to breaks. For example, if a code snippet is compilable for Python 2.4, uncompilable for Python 2.5, but compilable for Python 2.7. The available python version range for that code snippet is 2.4-2.7. Just because a code snippet is uncompilable for a newly released Python version does not mean that it is entirely deprecated, and many other Python versions are available simultaneously. We only need to focus on the earliest available Python version and the latest available Python version for a code snippet. Based on the above, we define the criteria for available Python version ranges as follows:

(1) The available Python version range of a code snippet is from the oldest Python version to the latest Python version that the code snippet can pass the corresponding PyComply parsing.
(2) There can be breaks between available Python version ranges.

To avoid the disruption caused by overlapping releases of the Python 2 and Python 3, we first need to investigate the available version ranges of Python code snippets in Python 2 and Python 3 separately. Figure 2 is a conceptual diagram of the version ranges

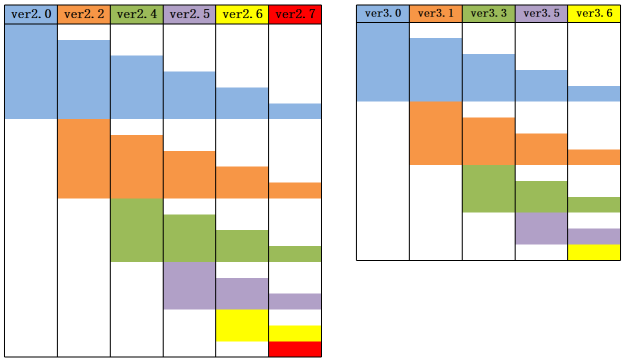


Fig. 2 Schematic diagram of the available Python version ranges

in Python 2 and Python 3, with the left and right parts representing Python 2 and Python 3, respectively. The first row of both parts shows the Python versions in the figure, each marked with a color. A rectangular block of color represents a range of Python versions in the rest of the figure, marked with the color corresponding to the Python version that the range starts with. There are 21 possible ranges in Python 2 and 15 possible ranges in Python 3. Taking Python 2 on the left as an example, the available version range represented by the orange-colored part of line eight is 2.2-2.4. Python code snippets in this range are uncompilable for Python 2 versions before Python 2.2 and after Python 2.4, but only for Python 2 versions in the range 2.2-2.4.

Based on the above, we investigate the available version ranges of Python code snippets within Python 2 and Python 3, respectively. Considering that there may be some code snippets whose available Python version ranges span Python 2 and Python 3, we combine the version ranges in Python 2 and Python 3 in pairs. If there are duplicate code snippets in the combination of the Python 2 range and the Python 3 range, it means that these duplicate code fragments are compilable for both Python version ranges. This way, we can get Python code snippets that are compilable across Python 2 and Python 3.

When we combine the available version ranges within Python 2 and Python 3, we use forms like [2.0,3.0-3.3] to represent our combinations. [2.0,3.0-3.3] means that the code snippets are compilable for Python 2.0 and versions in the range Python 3.0 to Python 3.3. We use ∅ to represent that there are no Python 2 versions or Python 3 versions available.

These combinations can be divided into three categories:

(1) `Ranges span Python 2 and Python 3`: Contains both the range of Python 2 and Python 3, such as [2.0,3.0-3.3]
(2) `Only in Python 2`: Contains only the range of Python 2, not the range of Python 3, like [2.0, ∅]
(3) `Only in Python 3`: Contains only the range of Python 3, not the range of Python 2, like [∅, 3.6]

3.3.2 RQ2:How Stack Overflow users respond to each Python release, and what are the differences between Python 2 and Python 3?

We would like to investigate how authors of Python code snippets on Stack Overflow would react and choose when faced with the constant release of new Python versions. Moreover, how this reaction behaves differently between Python 2 and Python 3. We can understand how Python versions are trending and interacting

Table 1 Pass rates for Python code snippets for each year for Python versions 2.0 through 3.6 (Removed the code snippets compilable for all python versions)

Year	ver2.0	ver2.2	ver2.4	ver2.5	ver2.6	ver2.7	ver3.0	ver3.1	ver3.3	ver3.5	ver3.6	# code snippets
2008	68%	72%	81%	89%	89%	90%	25%	25%	28%	28%	30%	220
2009	68%	70%	80%	85%	86%	88%	23%	23%	29%	29%	31%	2,150
2010	65%	67%	76%	84%	85%	87%	26%	26%	30%	30%	33%	4,689
2011	65%	67%	76%	85%	86%	88%	25%	25%	30%	30%	32%	8,674
2012	62%	64%	74%	84%	86%	88%	27%	27%	31%	31%	35%	15,682
2013	59%	61%	71%	84%	86%	88%	30%	30%	34%	34%	38%	25,980
2014	54%	56%	67%	82%	85%	87%	33%	34%	38%	38%	42%	32,392
2015	48%	51%	63%	81%	84%	86%	38%	39%	43%	43%	47%	38,354
2016	42%	45%	57%	79%	82%	85%	45%	45%	49%	50%	54%	42,720
2017	29%	33%	47%	74%	78%	82%	57%	58%	61%	63%	68%	47,501
2018	16%	20%	35%	65%	70%	74%	67%	68%	70%	74%	83%	46,476
2019	9%	13%	28%	59%	64%	69%	72%	72%	74%	78%	91%	36,783
2020	5%	10%	25%	55%	61%	65%	72%	73%	74%	77%	95%	6,167

by investigating the authors' responses to Python code snippets on Stack Overflow to each version.

In order to investigate how Stack Overflow users respond to each Python version, we first need to determine the release date of each python version, which can be obtained from Python.org[*5].

To observe how Stack Overflow users respond to each Python release, we can investigate how the Python code snippets responding to that Python version have evolved since each Python version release. We refer to the Python code snippets that are compilable for a certain Python version after it is released as the code snippets responding to that version, which is defined in detail as follows:

(1) The posting date for code snippets responding to a certain Python version needs to be after the release of that Python version.

(2) Code snippets that respond to a certain Python version must be compilable for that version, and uncompilable for the Python versions before that version.

Since PyComply does not provide versions of Python before Python 2.0, it is impossible to determine whether code snippets compilable for Python 2.0 are uncompilable for versions before Python 2.0 using PyComply parsing; therefore, we do not investigate Python 2.0 in this research question.

3.3.3 RQ3:How are Python 2-only compilable Python code snippets and Python 3-only compilable Python code snippets distributed on Stack Overflow?

While developers can upgrade to Python 3, code snippets written in Python 2 on Stack Overflow may not be modified by their authors to make them compilable for Python 3. While studying obsolete answers on Stack Overflow, zhang et al. [11] found that only a tiny proportion of such answers are updated afterward when an obsolete answer is identified. Soni et al. [8]. explored how comments affect answer updates on Stack Overflow, using the SOTorrent dataset. Their results show that a large number of answers on Stack Overflow are not updated, even when they receive comments that warrant an update.

Therefore, we need to determine if Python 2-only compilable Python code snippets on Stack Overflow. If so, compare it to the Python 3-only compilable code snippets and analyze the differences.

[*5] `https://www.python.org/`

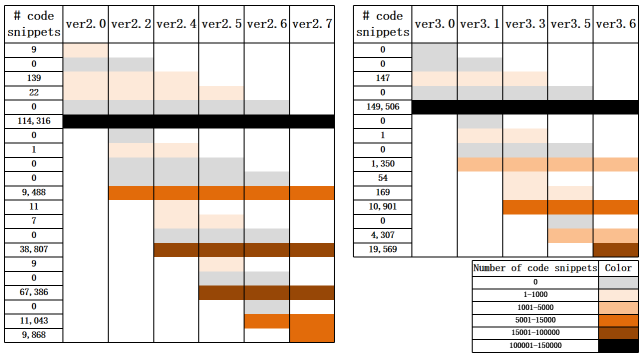


Fig. 3 Available python version range for Python code snippets on Stack Overflow

4. Case Study Results

4.1 RQ1:What are the available Python version ranges for Python code snippets on Stack Overflow?

To investigate the available Python version ranges for Python code snippets on Stack Overflow, we first investigated the available version ranges of Python code snippets within Python 2 and Python 3, respectively. The results are shown in Figure 3. The first column of the left and right subplots indicate the number of Python code snippets in each Python version range. The first row of the two subplots show each version of Python 2 and Python 3, respectively. A rectangular block of color represents an available Python version range in the rest of the figure. We divided the number of Python code snippets within each available Python version range into six groups, marked with different colors. The legend in the lower right corner of the figure shows the correspondence between the number of code snippets and the colors. For example, the color of the 2.0-2.2 range in the figure is gray, indicating that there are no compilable code snippets in that Python version range.

Second, we combined the available version ranges of Python 2 and Python 3 obtained in pairs. For each of the three categories of combinations as mentioned in Section 3, we investigated with the following methods:

(1) `Ranges span Python 2 and Python 3`: Taking [2.4-2.7,3.0-3.3] as an example, using the code snippets of [2.4-2.7] and [3.0-3.3] for comparison. The duplicate items of the two ranges are the code snippets that we need in the range [2.4-2.7,3.0-3.3].

(2) `Only in Python 2`: Taking [2.0, ∅] as an example, we first use the PyComply Parsing results to obtain code snippets that are uncompilable for all Python 3 versions. We then compare them with the code snippets in the range [2.0,2.0] to extract the duplicates. The duplicates are the code snippets we need.

(3) `Only in Python 3`: Taking [∅, 3.6] as an example, we first use the PyComply Parsing results to obtain code snippets that are uncompilable for all Python 2 versions. We then compare them with the code snippets in the range [3.6,3.6] to extract the duplicates. The duplicates are the code snippets we need.

The results we obtained are shown in Figure 4. The first column of the figure shows the available Python version ranges in Python 2, and the first row shows the available Python version ranges in Python 3. The values in the figure represent the number of code snippets in each available Python version range. We divided the number of Python code snippets within each Python available range into five groups, marked with different colors. The legend on the right side of the figure shows the correspondence between the number of code snippets and the colors. In total, there are 46 non-zero available Python version ranges. The 13 ranges in the last column belong to category 2, Only in Python 2. The total number of code snippets for this category is 121,784. The eight ranges in the last row belong to category 3, Only in Python 3. The total number of code snippets for this category is 56,682. The remaining 25 ranges are in category 1; ranges span Python 2 and Python 3. The total number of snippets for this category is 129,322.

We first analyze the data of category 2 and category 3. Most of the code snippets in category 2 are in the range [2.0-2.7, ∅]. This means that code snippets in this range are compilable for all versions of Python 2. Similarly, most code snippets in category 3 are in the range [∅, 3.0-3.6]. In addition, in category 3, most of the remaining code snippets except for [∅, 3.0-3.6] are in the range containing version 3.6, such as [∅, 3.5-3.6], [∅, 3.6]. Although category 2 is not so obvious, we can also see from the figure that, the code snippets except for the range [2.0-2.7, ∅] are mostly concentrated in the range containing version 2.7, such as [2.5-2.7, ∅] and [2.6-2.7, ∅]. Finally, the data for category 1 also show a similar pattern to the previous category 2 and category 3, as shown in the figure. Most of the code snippets are concentrated in the range of versions containing 2.7 and 3.6, such as [2.7, 3.1-3.6], [2.4-2.72.7, 3.3-3.6].

Answer to RQ1: We found that the available version ranges of Python code snippets whose compilability varies with Python version changes in this study can be divided into three categories. The code snippets whose available version range spans Python 2 and Python 3 account for 42.0%, those whose available version range is only in Python 2 account for 39.6%, and those whose available version range is only in Python 3 account for 18.4%.

4.2 RQ2:How Stack Overflow users respond to each Python release, and what are the differences between Python 2 and Python 3?

Studying the growth of Python code snippets responding to each Python version release based on the release date of each Python version can help us investigate how Stack Overflow users respond to each Python release and the differences between Python 2 and Python 3.

Range	3.0-3.3	3.0-3.6	3.1-3.3	3.1-3.6	3.3	3.3-3.5	3.3-3.6	3.5-3.6	3.6	∅
2.0	0	1	0	0	0	0	0	0	0	8
2.0-2.4	0	0	0	0	0	0	0	0	0	139
2.0-2.5	0	0	0	0	0	0	0	0	0	22
2.0-2.7	103	60	0	99	0	155	9,340	0	0	104,550
2.2-2.4	0	0	0	0	0	0	0	0	0	1
2.2-2.7	0	8,450	0	0	0	0	42	0	0	996
2.4	0	0	0	0	0	0	0	0	0	11
2.4-2.5	0	0	0	0	0	0	0	0	0	7
2.4-2.7	23	34,059	0	4	0	3	335	0	0	4,383
2.5	0	0	0	0	0	0	0	0	0	9
2.5-2.7	12	58,313	0	7	0	10	332	0	0	8,712
2.6-2.7	2	8,713	0	0	0	0	98	0	0	2,230
2.7	0	7,915	1	1,198	0	0	47	0	0	707
∅	7	31,995	0	42	54	1	707	4307	19,569	-

Number of Python code snippets	Color
1-500	
501-5000	
5001-10000	
10001-100000	
100001-200000	

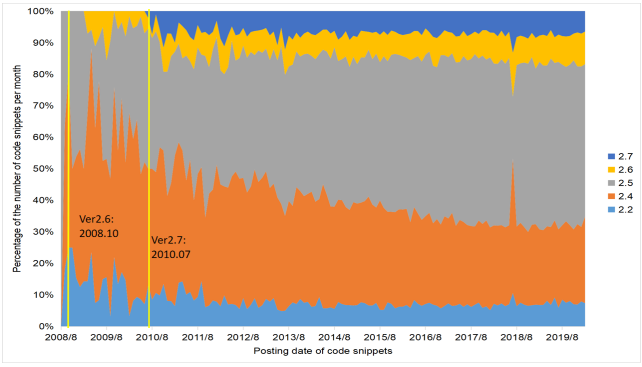
Fig. 4 Available version ranges

To improve the experimental accuracy in this research question, we divided the code snippets responding to each version by month. Figure 5 is the percent stacked area chart for Python 2 and Python 3. The release date of each Python version is marked with a yellow line in the figure. The horizontal axis represents the posting date of code snippets. The vertical axis shows the percentage of the number of code snippets per month.

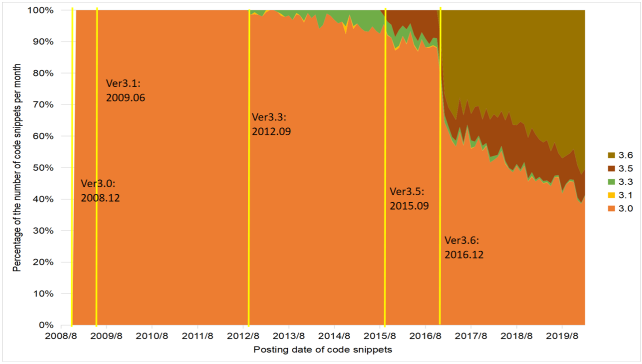
Figure 5(a) shows the growth of code snippets responding to each version of Python 2. Since August 2008, 2.2 and 2.4, released in October 2000 and December 2001 respectively, have shown an overall decreasing trend in their share of the total. 2.5 was released in September 2006 and took a more dominant position alongside 2.4 about two years after its release. 2.6 was released in October 2008, and code snippets responding to its release appeared about six months later. 2.6 also impacted 2.5 for a while after it gained a response, causing its share to continue to drop. 2.7 was released in July 2010 and quickly gained response after its release, and its response code snippets began to grow. 2.7 also affected other Python 2 releases after its releases, such as 2.5 between August and December 2010, and saw a relatively significant drop.

Figure 5(b) shows the growth of code snippets responding to each version of Python 3 versions. 3.0 was released in December 2008, and had a monopoly for quite some time as the Python 3 was starting. 3.1 was released six months later, but it had almost no presence. 3.3 was released in September 2012. Although 3.3 was quickly responded to and gave 3.0 some impact, it was still not very strong. Immediately after 3.5 was released in September 2015, 3.0 and 3.3 dropped significantly, but 3.0 still held a great advantage. Until December 2016, the release of 3.6 brought a massive hit to versions such as 3.0, which saw a steep drop in share. 3.5 also took a hit but regained its vigor and grew modestly after a while. Together with 3.6, the newer Python 3 versions gradually took over the dominance of 3.0.

As shown in Figure 5, the posting date of code snippets collected in SOTorrent start from August 8, 2008. Among the versions of Python 2, only Python 2.6 and Python 2.7 were released after August 8, 2008. The versions of Python 3 were all released after August 8, 2008. Our existing data support is insufficient to



(a) Growth of Python code snippets responding to Python 2 versions



(b) Growth of Python code snippets responding to Python 3 versions

Fig. 5 Growth of Python code snippets responding to each Python version

comprehensively compare the differences in user responses to all Python 2 and Python 3 releases studied in this study. Therefore, we only explore the differences from a feasible perspective.

Comparing Python 2 and Python 3, we found that Python 3 has grown more rapidly than Python 2 in the code snippets we studied. Python 2.7 was released in July 2010, but until 2020, 2.5 and 2.4 are still the dominant versions. And in Python 3, 3.6 quickly gained a dominant position as soon as it was released. This speed difference may be related to the official discontinuation of Python 2 support mentioned in Section 2.

Answer to RQ2: We found that new versions are released and get a response shortly afterward. The release of new Python version inhibits Stack Overflow users' response to older versions. The existing data only allow us to explore the differences between the user responses of Python 2 and Python 3 versions from a partially feasible perspective.

4.3 RQ3:How are Python 2-only compilable Python code snippets and Python 3-only compilable Python code snippets distributed on Stack Overflow?

To investigate RQ3, we utilized the code snippets compilable only for Python 2 (121,784) and the code snippets compilable only for Python 3 (56,682) obtained in RQ1.

Figure 6 shows the distribution of Python 2-only compilable Python code snippets and Python 3-only compilable Python code snippets. Since SOTorrent only includes data for the first two months of 2020, the overall number of Python code snippets in 2020 is less.

As shown in Figure 6, The number of Python 2-only compilable Python code snippets increased year by year starting in 2008,

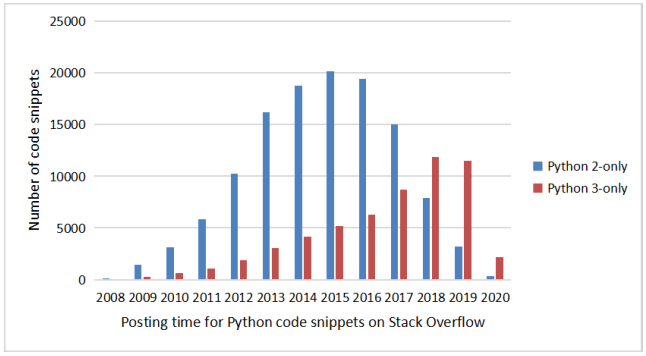


Fig. 6 Distribution of Python 2-only compilable Python code snippets and Python 3-only compilable Python code snippets

peaked in 2015, and has since begun to decline each year. On the other hand, there is an overall upward trend in Python 3-only compilable Python code snippets.

While the number of Python 3-only compilable Python code snippets is increasing, overall, the number of Python 2-only compilable Python code snippets far exceeds the number for Python 3-only compilable Python code snippets.

Answer to RQ3: The number of Python 2-only compilable Python code snippets increased year by year starting in 2008, peaked in 2015, and began to decline year by year after that. There is an overall upward trend in Python 3-only compilable Python code snippets. The Python 2-only compilable Python code snippets are about twice as large as the Python 3-only compilable Python code snippets.

5. Discussion

5.1 Implications

Through investigation, we found that the Python version upgrade affects the compilability of quite a few Python code snippets on Stack Overflow. Stack Overflow, as well as Stack Overflow users, should pay attention to this issue.

Suggestions for Stack Overflow: An automated tool can be built to identify the Python language version of existing code snippets on Stack Overflow or help users identify the Python version used in the posted code snippet in real-time as they create their questions or answers. In RQ3, we found that code snippets only compilable for Python 2 accounted for about 40% of the total number of code snippets we surveyed. Although the Python team discontinued support for Python 2 on January 1, 2020, there are still newly posted code snippets that are only compilable for Python 2 until 2020. An automated tool can be developed to identify possible Python versions of code snippets by analyzing their syntactic features as they are entered.

Suggestions for Stack Overflow Users: We recommend that Stack Overflow users provide information about the Python version used to write the code snippet when attaching code snippets to illustrate their questions or give answers. In RQ1, we observed that many of Stack Overflow's Python code snippets have different available Python version ranges and are not compiled by all Python versions. If users actively provide the Python version of the code snippet when posting or answering questions.

Users searching Stack Overflow for the required code snippet should also pay more attention to the information about the

Python version. Alternatively, identify the possible Python version of the snippet before utilizing the code snippet.

5.2 Threats to Validity

Our study is subject to limitations and threats to validity.

Limitations of PyComply: Parsing metrics of PyComply [6] parsing metrics are based on (static) syntactic observations, coarse-grained, and roughly categorize each Python file into compliant or not, without attempting to estimate the degree of compliance. While we believe that the level studied here is sufficient for our purposes, we should note that the parsing results of these PyComply cannot assert that any code snippet is 100% compliant with the Python version.

Limitations of Python code snippets: As we mentioned in Section 3, about 47.4% of the Python code snippets we obtained from SOTorrent could not pass the PyComply parsing for all Python versions, i.e., these code snippets are uncompilable for all Python versions. These code snippets contain programming errors, pseudocode, and other issues unrelated to the Python version upgrade that caused the parsing failure. However, there may also be featured in these code snippets related to Python version upgrades but are masked by errors caused by other unrelated issues. Because of technical and time constraints, we abandoned processing and studying this part of the code snippets in this study. This may pose a threat to the validity of our results.

External validity: The generality of our findings poses a danger to external validity. We focused on Stack Overflow in this study, and our findings may not apply to other Q&A sites because of the differences in mechanisms. To mitigate this threat, we should study more Q&A sites in the future.

Furthermore, unlike Malloy et al.'s analysis using Qualitas corpus [6], the Python files contained in it are parsed on a per-application basis. Our research objects are unrelated code snippets, and many of them are only short fragments, lacking syntactic features that can be used as metrics. This makes our findings less generalizable and perhaps not applicable beyond the Q&A site.

6. Conclusion

In this study, we want to investigate the effect of Python version upgrades on the compilability of Python code snippets on Stack Overflow. By analyzing the compilability of Python code snippets on Stack Overflow for different Python versions, We found that Python version upgrades had an impact on the compilability of Python code snippets on Stack Overflow, as evidenced by: 1) About 40% of the Python code snippets on Stack Overflow whose compilability changed with the Python version in this study are uncompilable for Python 3. 2) The release of new Python version inhibits Stack Overflow users' response to older versions. 3) The trend of code snippets responding to newer versions increases over time. Based on our findings, we offer the following suggestions: 1) An automated tool can be built to help users identify the Python language version of code snippets in Stack Overflow posts. 2) We recommend that Stack Overflow users provide information about the Python version used to write the code snippet when attaching code snippets to illustrate their questions or give answers. 3) Users searching Stack Overflow for the required code snippet should also pay more attention to the information about the Python version in the posts.

There are two possible directions for future work. First, we want to investigate factors that affect the compilability of code snippets on Stack Overflow due to Python version upgrades. For example, the syntax usage of specific Python versions. In addition, we would like to investigate the reasons for the failure of the code snippets filtered out in this study that are uncompilable for all Python versions.

Acknowledgments This work was supported by JSPS KAKENHI Grant Numbers JP18H04094 and JP19K20239.

References

[1] An, L., Mlouki, O., Khomh, F. and Antoniol, G.: Stack Overflow: A code laundering platform?, *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, IEEE Computer Society, pp. 283–293 (online), DOI: 10.1109/SANER.2017.7884629 (2017).

[2] Baltes, S. and Diehl, S.: Usage and Attribution of Stack Overflow Code Snippets in GitHub Projects, *Empirical Softw. Engg.*, Vol. 24, No. 3, pp. 1259–1295 (online), DOI: 10.1007/s10664-018-9650-5 (2019).

[3] Baltes, S., Treude, C. and Diehl, S.: SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets, *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, IEEE Press, pp. 191–194 (online), DOI: 10.1109/MSR.2019.00038 (2019).

[4] Fischer, F., Böttinger, K., Xiao, H., Stransky, C., Acar, Y., Backes, M. and Fahl, S.: Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security, *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 121–136 (online), DOI: 10.1109/SP.2017.31 (2017).

[5] Fischer, F., Böttinger, K., Xiao, H., Stransky, C., Acar, Y., Backes, M. and Fahl, S.: Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security, *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 121–136 (online), DOI: 10.1109/SP.2017.31 (2017).

[6] Malloy, B. A. and Power, J. F.: Quantifying the Transition from Python 2 to 3: An Empirical Study of Python Applications, *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, IEEE Computer Society, pp. 314–323 (online), DOI: 10.1109/ESEM.2017.45 (2017).

[7] Ragkhitwetsagul, C., Krinke, J., Paixao, M., Bianco, G. and Oliveto, R.: Toxic Code Snippets on Stack Overflow, *IEEE Transactions on Software Engineering*, Vol. 47, No. 3, pp. 560–581 (online), DOI: 10.1109/TSE.2019.2900307 (2021).

[8] Soni, A. and Nadi, S.: Analyzing Comment-Induced Updates on Stack Overflow, *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, pp. 220–224 (online), DOI: 10.1109/MSR.2019.00044 (2019).

[9] Verdi, M., Sami, A., Akhondali, J., Khomh, F., Uddin, G. and Motlagh, A. K.: An Empirical Study of C++ Vulnerabilities in Crowd-Sourced Code Examples, *IEEE Transactions on Software Engineering*, Vol. 48, No. 5, pp. 1497–1514 (online), DOI: 10.1109/TSE.2020.3023664 (2022).

[10] Wu, Y., Wang, S., Bezemer, C.-P. and Inoue, K.: How Do Developers Utilize Source Code from Stack Overflow?, *Empirical Software Engineering*, Vol. 24, pp. 637–673 (2019).

[11] Zhang, H., Wang, S., Chen, T.-H., Zou, Y. and Hassan, A. E.: An Empirical Study of Obsolete Answers on Stack Overflow, *IEEE Transactions on Software Engineering*, Vol. 47, No. 4, pp. 850–862 (online), DOI: 10.1109/TSE.2019.2906315 (2021).

[12] Zhou, J. and Walker, R. J.: API Deprecation: A Retrospective Analysis and Detection Method for Code Examples on the Web, *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pp. 266–277 (online), DOI: 10.1145/2950290.2950298 (2016).

[13] Sunsetting Python 2, Available at `https://www.python.org/doc/sunset-python-2/`. Accessed: 2022-06-27.