

One thousand and one stories: a large-scale survey of software refactoring

Yaroslav Golubev, Zarina Kurbatova, Eman Abdullah AlOmar,
Timofey Bryksin, Mohamed Wiem Mkaouer

ESEC/FSE 2021: Proceedings of the 29th ACM Joint Meeting on
European Software Engineering Conference and Symposium on the
Foundations of Software Engineering August 2021 Pages 1303–1313

発表者：松尾 和輝

背景

- リファクタリングは、高品質なソフトウェアを維持し、技術的負債を減らすための重要なツールである
- 最近の統合開発環境（IDE）には、名前の変更、要素の移動、抽出などを含むいくつかのリファクタリングがツールとして実装されており、これらの定義済みコード変換を自動的かつ安全に適用する方法を開発者に提供している[32, 33, 36]
- 最近の調査では、開発者が自動リファクタリングツールの採用に消極的で、いまだに手動でコードをリファクタリングしているという報告がいくつかある[17, 35]

目的とアプローチ

- 目的

- 開発者が実際にどのようなリファクタリングを使用しているかを考察し、開発者が**IDE**を使用してコードをリファクタリングする際に直面する現在のユーザビリティ上の課題に対する認識を高めること

- アプローチ

- アンケートを実施、アンケートの結果を用いて分析

回答者の選定

- IntelliJベースのIDE（IntelliJ IDEA, PyCharm, Clion, WebStormなど）[31]の有料会員の中から、調査の案内をメールで受け取ることに同意した人を選択、合計で2万人の開発者が参加
- 19860通のメールが配信され、回答者には3週間のアンケート期間が与えられた
- 1330件の回答があり、そのうち1,183件が回答済み、回答率は6.0%、回答完了率は88.9%だった

アンケート内容

Table 1: The summary of survey questions. Grey numbers near certain questions indicate that the presence of this question conditionally depends on another question, specified by the number

I. Background	
Question 1	How many years of coding experience do you have?
Question 2	What programming languages do you regularly use?
II. Familiarity with refactorings	
Question 3	In the past month, how often have you performed any code refactoring?
Question 4 (3)	Have you renamed anything in your code or project structure in the past month?
Question 5 (3)	Have you moved any code from one location in the project to another?
Question 6 (3)	During this time, did you ever refactor code for an hour or more in a single session?
III. Refactoring approaches	
Question 7	For the following scenarios, please select all the approaches you have used in the past month. (Renaming a class, method, variable, or symbol / Extracting a method or a variable from existing code / Moving code to another file)
Question 8 (7)	In these scenarios, what were your main reasons for not using the IDE refactoring feature?
Question 9	For the following scenarios, please select all the approaches you have used in the past month. (Inlining a variable or method / Changing the signature of an existing function / Moving a method up or down the class hierarchy)
Question 10 (9)	In these scenarios, what were your main reasons for not using the IDE refactoring feature?
IV. IDE refactoring features	
Question 11	How familiar are you with the following IDE refactoring features? (RENAME FILE, CLASS, METHOD, SYMBOL, ETC. / EXTRACT METHOD, VARIABLE, COMPONENT, ETC. / MOVE / INLINE VARIABLE OR METHOD / CHANGE SIGNATURE / PULL UP OR PUSH DOWN MEMBER)
Question 12 (11)	Please think about the last several times you used IDE refactorings. How happy were you with the overall experience?
Question 13 (12)	Please tell us a bit more about your experience.
Question 14 (11)	How often do you undo or revert an IDE refactoring action because you're unhappy with the result?
Question 15 (14)	The last few times you undid or reverted an IDE refactoring feature, what were the reasons?
V. Previewing refactorings	
Question 16	When refactoring the code, do you find it useful to preview all of your changes before applying them?
Question 17 (16)	For what types of changes do you find this feature most useful?
Question 18 (16)	What are your main reasons for wanting to see a refactoring preview?
Question 19 (16)	What are your main reasons for not finding a preview useful?
VI. Final thoughts	
Question 20	Please share any thoughts or feedback you have about using the IDE refactorings.

回答者の背景

- Q1：コーディング経験について
 - 27.8%の回答者が16年以上，合計48.1%が10年以上，79%が5年以上の経験を持っていることが示された．なお，「コーディングの経験はありません」と回答した方は2名（0.2%）
- Q2：よく用いるプログラミング言語について
 - 回答者が慣れ親しんだ言語を3つまで選択
 - JavaScript（46.6%），Python（28.3%），PHP（27.2%），Java（24.4%），SQL（21.4%），これらの言語は，さまざまなソフトウェア工学分野で使用されている最も一般的なプログラミング言語の一部である[11].
 - その他の言語は，Typescript，C/C++，C#，Go，Rubyなどが挙げられた

リファクタリングに対する親しみ

- Q3：リファクタリングを行う頻度について

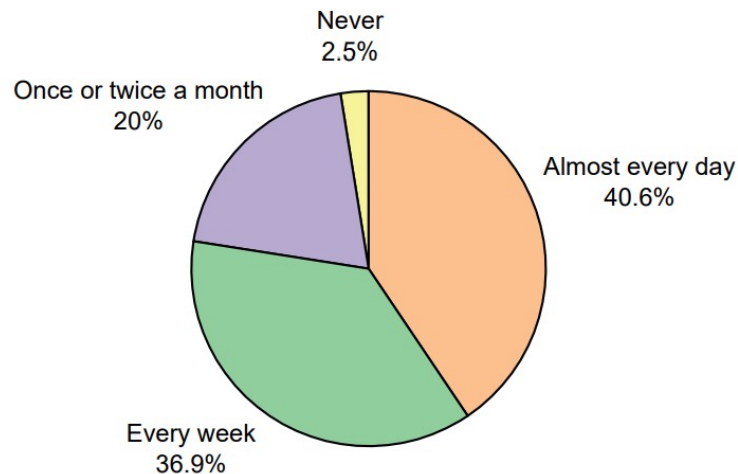


Figure 1: The answers to Question 3: *In the past month, how often have you performed any code refactoring?* (Out of 1,181 respondents)

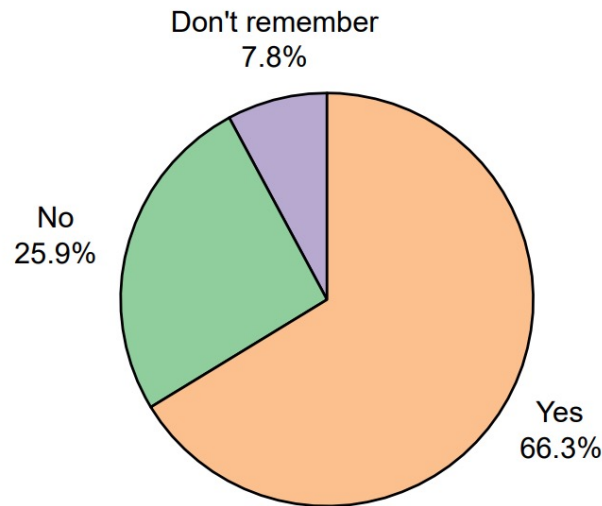
- 40.6%がほぼ毎日、36.9%が毎週リファクタリングを行っている
- 月に1・2回、全くしないと答えた266人（22.5%）の回答者に対しては、リファクタリングを知らない可能性があるため、追加の質問（Q4, Q5）を行う

リファクタリングに対する親しみ

- Q4：過去 1 カ月の間に、コードやプロジェクトの構造で何か名前を変更したか？
- Q5：プロジェクト内のある場所から別の場所にコードを移動したか？
 - 回答者266名のうち84.2%がコードの名前を変更し、78.2%がコードの要素を移動していた.
 - これらの質問に肯定的な回答をしなかった回答者は14名のみであり、彼らはリファクタリングに関する質問の対象者ではないため、これ以上の回答は考慮しなかった

リファクタリングに対する親しみ

- Q6：1 時間以上連続で、コードのリファクタリングを行ったことがあるか？



- リファクタリングに精通している1,145人に対して質問
- 3分の2が1度に1時間以上リファクタリングを行ったと回答

Figure 2: The answers to Question 6: *During this time, did you ever refactor code for an hour or more in a single session?* (Out of 1,145 respondents)

リファクタリングのアプローチ

- Q7：過去一ヶ月の間に「リネーム」、「抽出」、「移動」それぞれのリファクタリングを行ったか？リファクタリングを行う際にどのような手段（自動/手動等）で行ったか？

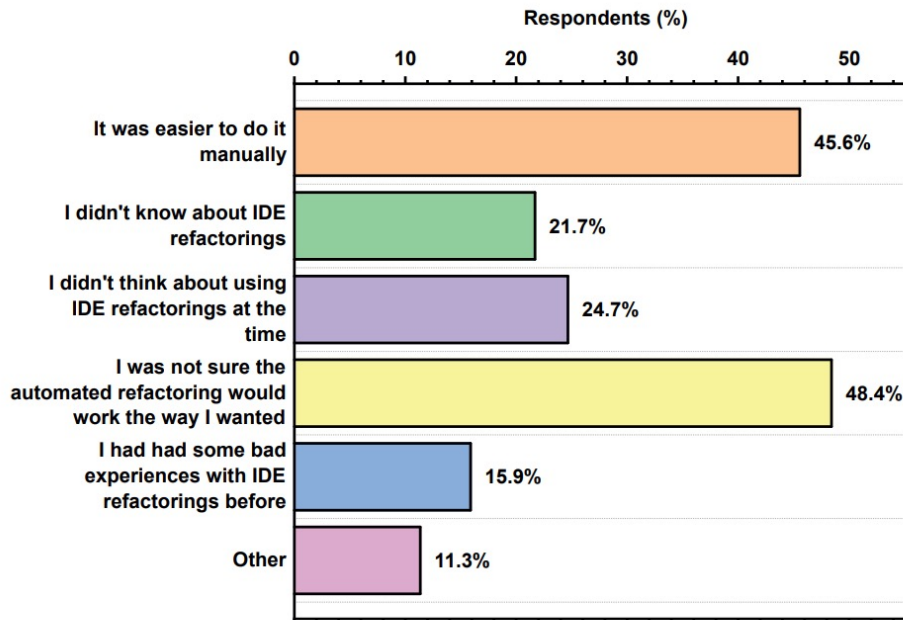
	Used IDE refactoring	Used Find and Replace	Used Copy and Paste	Edited manually	Didn't have this scenario
Renaming a class, method, variable, or symbol	85.8%	46.2%	21.3%	29.8%	0.3%
Extracting a method or a variable from existing code	54.7%	20.7%	30.4%	33.2%	10.7%
Moving code to another file	38.6%	12.2%	57.5%	30.8%	4.4%

(a) The answers to Question 7: For the following scenarios, please select all the approaches you have used in the past month. Several approaches can be selected for each refactoring, the percentages show the proportion of respondents who chose each option. (Out of 1,167 respondents)

- リネームに関しては、自動で行うと回答した人が多い
- リネームは単純な作業であるため、検索と置換で行う人も多かった
- 抽出も半数は自動で行う人が多い
- 移動に関してはコピペで行う人が多い

リファクタリングのアプローチ

- Q8：IDEの自動リファクタリング機能を使わない理由は？



(b) The answers to Question 8: *In these scenarios (Renaming, Extracting, Moving), what were your main reasons for not using the IDE refactoring feature?* (Out of 1,014 respondents)

- 少なくとも1回のリファクタリングで「検索と置換」,「コピペ」,「手動」と回答した1,014名に対して質問
- 手動で行う方が簡単, 自動リファクタリングが自分の思い通りに動くかどうかかわからないと回答した人が半数だった
- リファクタリングツールはシンプルで直感的なものにするべきである

リファクタリングのアプローチ

- Q9：過去一ヶ月の間に「インライン化」，「シグネチャの変更」，「メソッドのクラス階層の上下への移動」それぞれのリファクタリングを行ったか？リファクタリングを行う際にどのような手段（自動/手動等）で行ったか？

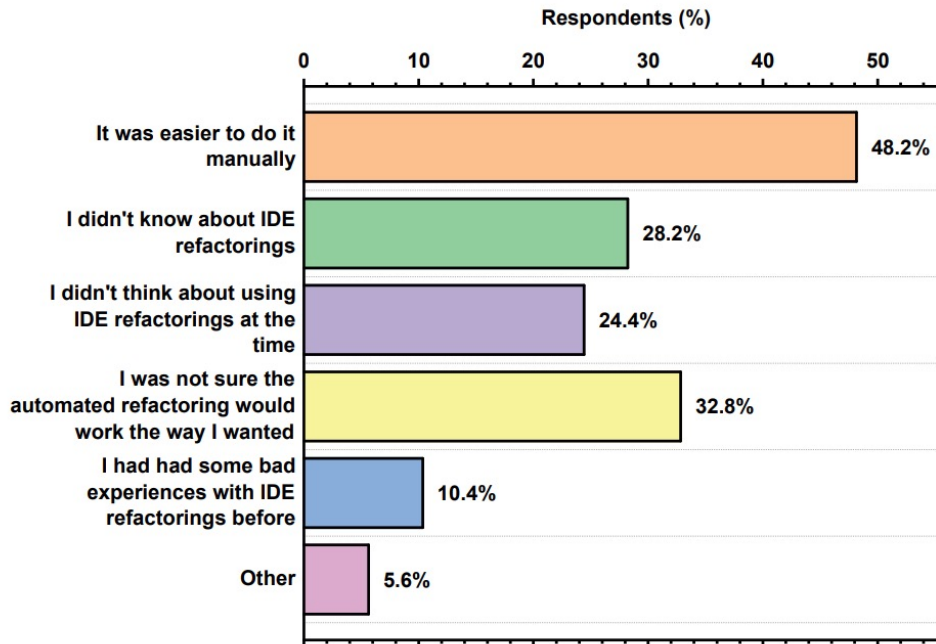
	Used IDE refactoring	Used Find and Replace	Used Copy and Paste	Edited manually	Didn't have this scenario
Inlining a variable or method	50.3%	11.7%	12.8%	25.4%	24.8%
Changing the signature of an existing function	53.6%	14.3%	9.8%	42.1%	10.6%
Moving a method up or down the class hierarchy	27.0%	6.4%	26.9%	27.2%	30.0%

- インライン化，シグネチャの変更に関しては，自動で行うと回答した人が多い
- シグネチャの変更は，手動で行うと回答した人も多い
- メソッドのクラス階層の上下への移動は行ったことがないと回答した人が多い

(c) The answers to Question 9: For the following scenarios, please select the approaches you've used in the past month. Several approaches can be selected for each refactoring, the percentages show the proportion of respondents who chose each option. (Out of 1,167 respondents)

リファクタリングのアプローチ

- Q10：IDEの自動リファクタリング機能を使わない理由は？



(d) The answers to Question 10: *In these scenarios (Inlining, Changing the Signature, Pulling/Pushing), what were your main reasons for not using the IDE refactoring feature?* (Out of 868 respondents)

- 少なくとも1回のリファクタリングで「検索と置換」,「コピペ」,「手動」と回答した868名に対して質問
- 手動で行う方が簡単と回答した人が多かった
- 次いで、自動リファクタリングが自分の思い通りに動くかどうかわからない, IDEのリファクタリングを知らないと回答した人が多かった

IDEのリファクタリング機能

- Q11: 「リネーム」, 「抽出」, 「移動」, 「インライン化」, 「シグネチャの変更」, 「メソッドのクラス階層の上下への移動」それぞれのリファクタリングをどれだけ知っているか?
- リネームに関しては知っている人が多かった
- メソッドのクラス階層の上下への移動は知らない人が多かった
- 全てにおいて「過去にリファクタリングを行ったが、今後は行わない」と回答した人が少なかった

	Don't know about it	Know about it, but don't use it	Used it in the past, but will not use it again	Use it sometimes	Use it regularly
Rename file, class, method, symbol, etc.	5.1%	4.5%	2.1%	22.6%	65.7%
Extract method, variable, component, etc.	22.5%	18.8%	2.7%	25.7%	30.3%
Move	27.4%	15.7%	3.4%	27.5%	26.0%
Inline variable or method	30.3%	18.9%	2.1%	25.3%	23.4%
Change signature	26.8%	15.0%	2.7%	27.0%	28.5%
Pull Up / Push Down member	42.8%	20.7%	2.7%	19.3%	14.5%

Figure 4: The answers to Question 11: *How familiar are you with the following IDE refactoring features?* (Out of 1,167 respondents)

IDEのリファクタリング機能

- Q12：IDEのリファクタリングを使用した時，どれだけ満足したか

Table 2: The answers to Question 12: *Please think about the last several times you used IDE refactorings. How happy were you with the overall experience?* (Out of 1,078 respondents)

Answer	% of respondents
Not at all happy	0.3%
Not happy	1.6%
Neither happy nor unhappy	12.6%
Happy	56.9%
Very happy	28.6%

- Q11で「時々使う」または「定期的に使う」と回答した1,078名（92.4%）に対して質問
- 「非常に満足」，「満足」と回答した人が非常に多かった
- 否定的な回答を行った人に追加の質問（Q13）を行う

IDEのリファクタリング機能

- Q13：なぜ満足しなかったのか？
- 自由記述により回答
 - リファクタリングによってコードが壊れたり，新しいエラーが発生したから
 - DEのリファクタリングツールを使うと，手動でリファクタリングを行うよりも時間がかかる可能性があるから
 - IDEのリファクタリングツールを理解するのに時間がかかるから
 - 自動化ツールを信頼できないから
 - 自動リファクタリングを適用すると，特定のコーディングスタイル違反する可能性があるから
- 等

IDEのリファクタリング機能

- Q14：結果に満足できないために、IDEリファクタリングの操作を取り消したり、元に戻したりする頻度はどのくらいか？

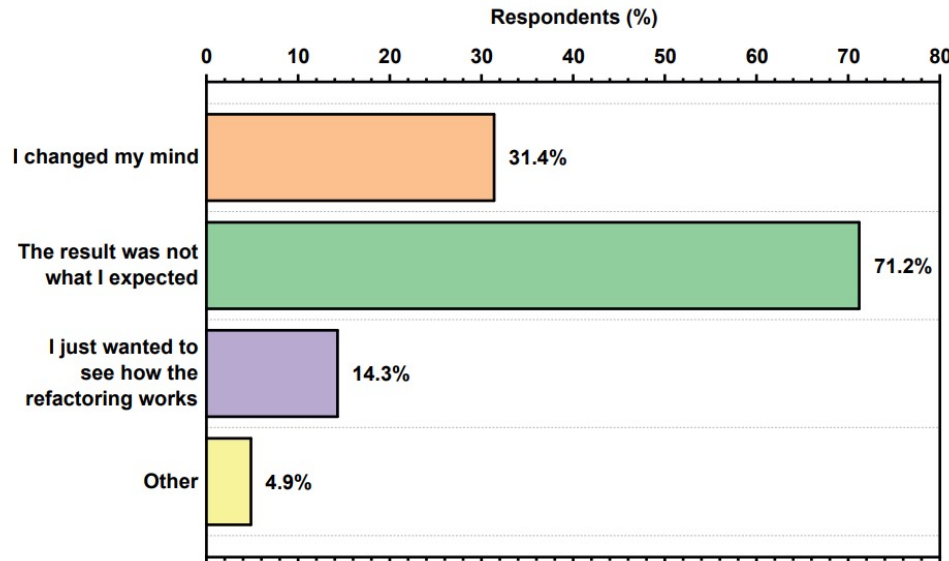
Table 3: The answers to Question 14: *How often do you undo or revert an IDE refactoring action because you're unhappy with the result?* (Out of 1,091 respondents)

Answer	% of respondents
Often (>75% of the time)	3.0%
Every other time (≈50% of the time)	6.2%
Sometimes (≈25% of the time)	32.3%
Rarely (≈5% of the time)	50.8%
Never	7.7%

- 減多に元に戻さないと答えた人が多い
- 次いで時々と答えた人が多かった
- 戻さないと答えた人以外のすべての人（1007人）に元に戻す理由を質問（Q15）

IDEのリファクタリング機能

- Q15： IDEリファクタリングの操作を取り消したり，元に戻したりする理由は何か？

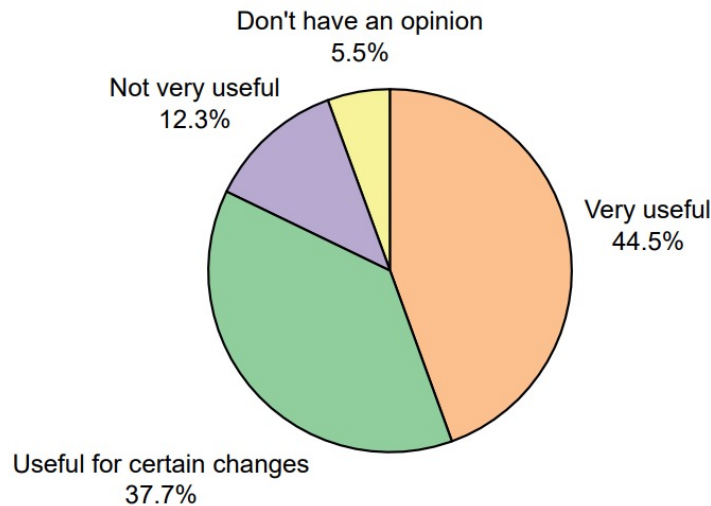


- 期待していた結果と違ったと答えた人が多い
- 次いで，気が変わったと答えた人が多い

Figure 5: The answers to Question 15: *The last few times you undid or reverted an IDE refactoring feature, what were the reasons?* (Out of 1,007 respondents)

リファクタリングのプレビュー

- Q16：コードをリファクタリングするとき、変更を適用する前にすべての変更点をプレビューすることが有用だと思うか？



- 非常に役に立つ、特定の変更のために役に立つ等、肯定的な回答をした人が多い
- 肯定的な回答をした人、否定的な回答をした人、それぞれに対し、追加の質問を行う（Q17, Q18, Q19）

Figure 6: The answers to Question 16: When refactoring the code, do you find it useful to preview all of your changes before applying them? (Out of 1,167 respondents)

リファクタリングのレビュー

- Q17：（肯定的な回答をした人）レビューが有用だと思うリファクタリングはどれか？
- Q17の結果はQ11の結果の右2列（下図）と似ていた

	Don't know about it	Know about it, but don't use it	Used it in the past, but will not use it again	Use it sometimes	Use it regularly
Rename file, class, method, symbol, etc.	5.1%	4.5%	2.1%	22.6%	65.7%
Extract method, variable, component, etc.	22.5%	18.8%	2.7%	25.7%	30.3%
Move	27.4%	15.7%	3.4%	27.5%	26.0%
Inline variable or method	30.3%	18.9%	2.1%	25.3%	23.4%
Change signature	26.8%	15.0%	2.7%	27.0%	28.5%
Pull Up / Push Down member	42.8%	20.7%	2.7%	19.3%	14.5%

Figure 4: The answers to Question 11: *How familiar are you with the following IDE refactoring features?* (Out of 1,167 respondents)

リファクタリングのレビュー

- Q18：（肯定的な回答をした人）レビューを見たい理由は何か？
 - すべてが思い通りに行われていることを確認するため
 - コードが壊れないことを確認するため
 - 自動機能に対する不信感に対抗するため
 - 変更中にコードの読みやすさをチェックし、リファクタリングを適用しても悪化しないことを確認するため
 - 自分が何かを忘れていないかどうかを確認するため

リファクタリングのレビュー

- Q19：（否定的な回答をした人）レビューが役に立たないと思う主な理由は何か？

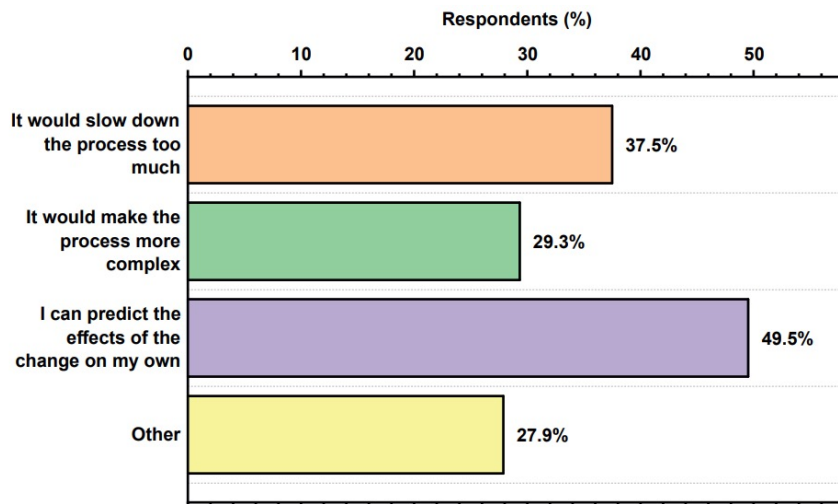


Figure 7: The answers to Question 19: *What are your main reasons for not finding a preview useful?* (Out of 208 respondents)

- 変更の影響を自分で予測できるからと答えた人が多い
- 次いで、IDEの処理が遅くなるから、プロセスが複雑になるからと答えた人が多い

最終的な考え

- Q20： IDE のリファクタリングを使用した感想やフィードバックについて
 - リファクタリングを利用するために、リファクタリングについて学ぶのが難しい
 - プログラミング言語によって、リファクタリングの動作が違う
 - 特定のソースファイルやコードをリファクタリングできないようにすれば良いのでは
 - 便利だけど仕組みを理解するのが難しい
 - 手動で行う方が速い

まとめ

- 開発者のリファクタリング使用に関する研究
- 開発者にアンケート調査を行い，その結果を分析
- ほとんどの開発者が毎日リファクタリングを行っている，また1度に1時間以上リファクタリングを行っている
- リネームは自動で行う人が多い，他のリファクタリングに関しては，検索と置換，コピー等手動で行う人が多い
- プレビューは有用であると答えた人が多い

所感

- 回答者がどれくらいの規模のプログラムを開発しているのかがわからない
 - プログラムの規模によって、自動リファクタリングの必要性が変わるのではないか
- ややこしいリファクタリングの方が手動で行う人が多いことが意外
 - 自分のイメージではややこしいリファクタリングの方が自動で行った方が楽なのではないかと思っていた