Large-scale Refactoring Survey

Start of Block: Large-scale Refactoring Survey

Large-scale Refactoring Survey We are a group of researchers at Carnegie Mellon University seeking to better understand how often software developers engage in **large-scale refactoring**, the difficulties they encounter during that refactoring, and if/what kind of tooling is needed to reduce the costs and complexity of large-scale refactoring. **Refactoring** is the process of restructuring software without changing its external behavior. Small-scale refactorings described by Martin Fowler (e.g., rename, move function, extract class) are often used independently, but sometimes form part of a larger-scale effort to improve existing software. That larger-scale effort is the focus of this study. The insights from this study will be used to guide future research in automated refactoring assistance and inform the construction of tools that effectively assist developers with software evolution. This survey will take approximately fifteen minutes

Informed Consent

Thank you for your time today. This fifteen-minute survey is part of a research study conducted by the Carnegie Mellon University Software Engineering Institute and is funded by the office of the Under Secretary of Defense.

Purpose The goal of this study is to identify how often software developers engage in large-scale refactoring, the difficulties they encounter during that refactoring, and if tooling is needed to reduce the costs and complexity of large-scale refactoring. The insights from this voluntary research study will be used to guide future research in automated refactoring assistance and inform the construction of tools that effectively assist developers. **Procedures** In this survey, you will be asked to respond to a small number of questions that seek to understand the practical challenges you observe regarding large-scale refactoring. The survey is expected to take about 15 minutes of your time. Please remember to not reveal anything that is both identifiable and private about yourself or others. **Participant Requirements** Participation in this study is limited to individuals age 18 and older. You are being invited to participate because we believe you have worked in developing, maintaining, and evolving software in some capacity. **Risks** The risks and discomfort associated with participation in this study are no greater than those ordinarily encountered in daily life or during other online activities. There is a potential risk of breach of confidentiality. **Benefits** There may be no personal benefit from your participation in the study but the knowledge received may be of value to software engineering practices. **Compensation & Costs** There is no compensation for participation in this study. There will be no cost to you if you participate in this study. **Future Use of Information** In the future, once we have removed all identifiable information from your

data, we may use the data for our future research studies, or we may distribute the data to other researchers for their research studies. We would do this without getting additional informed consent from you. Sharing of data with other researchers will only be done in such a manner that you will not be identified. **Confidentiality** By participating in this research, you understand and agree that Carnegie Mellon may be required to disclose your consent form, data and other personally identifiable information as required by law, regulation, subpoena or court order. Otherwise, your confidentiality will be maintained in the following manner: Your data and consent form will be kept separate. Your consent form will be stored in a secure location on Carnegie Mellon property and will not be disclosed to third parties. By participating, you understand and agree that the data and information gathered during this study, including quotations from your answers, may be used by Carnegie Mellon and published and/or disclosed by Carnegie Mellon to others outside of Carnegie Mellon. However, your name, contact information and other direct personal identifiers will not be mentioned in any such publication or dissemination of the research data and/or results by Carnegie Mellon. Note that per regulation all research data must be kept for a minimum of 3 years. The federal sponsor has the right to access all research records. **Right to Ask Questions & Contact Information** If you have any questions about this study, you should feel free to ask them by contacting the Principal Investigator Dr. Robert Nord, Principal Researcher, Software Engineering Institute, Carnegie Mellon University, 4500 Fifth Avenue, Pittsburgh, PA, Phone: 412-268-1705. Email: info@sei.cmu.edu. If you have questions later, desire additional information, or wish to withdraw your participation please contact the Principal Investigator by mail, phone or e-mail in accordance with the contact information listed above. If you have questions pertaining to your rights as a research participant; or to report concerns to this study, you should contact the Office of Research integrity and Compliance at Carnegie Mellon University. Email: irb-review@andrew.cmu.edu . Phone: 412-268-1901 or 412-268-5460. **Voluntary Participation** Your participation in this research is voluntary. You may discontinue participation at any time during the research activity. You may print a copy of this consent form for your records.

Survey Online Consent

Q1.1
I am age 18 or older. I have read and understand the information above. I want to participate in this research and continue with the survey.

- ○ Yes (1)
- ○ No (2)

Skip To: End of Survey If I am age 18 or older. I have read and understand the information above. I want to participate in... = No

End of Block: Large-scale Refactoring Survey

Start of Block: Respondent Demographics

Experience

Q2.1 How many years of experience do you have in the software industry?

- ○ Less than three years (1)
- ○ Between three and ten years (2)
- ○ Ten or more years (3)

Q2.2 What tools do you currently use for refactoring at any scale?

_____

_____

_____

_____

_____

We define ***large-scale refactoring*** as restructuring software, without introducing functionality, for the purpose of improving non-functional quality or changing architecture. Large scale involves either pervasive changes across a code base or extensive changes to a substantial element of the system (*e.g.*, greater than 10k LOC). One example is the need to partition legacy monoliths into smaller pieces to create separately deployable, scalable, and evolvable units. Another example is restructuring interfaces and communication patterns to enable replacement of a legacy feature by an improved or less proprietary alternative.

Q2.3 According to the above definition, on how many occasions during your career have you participated in a large-scale refactoring?

- ○ I have never participated in a large-scale refactoring (1)

- ○ I have participated in one large-scale refactoring (2)
- ○ I have participated in two to four large-scale refactorings (3)
- ○ I have participated in five or more large-scale refactorings (4)

End of Block: Respondent Demographics

Start of Block: Software system

Software Project

Recall the software project that you spent the most amount of time working on during your career.

Q3.1 Do you still work on that software?

- ○ Yes (1)
- ○ No (2)

Q3.2 For how many years have you worked on that software?

- ○ Less than one year (1)
- ○ Between one and three years (2)
- ○ Between three and five years (3)
- ○ Between five and ten years (4)
- ○ More than ten years (5)

Q3.3 How old was that software when you started working on it?

- ○ I was there at the very beginning (1)

- ○ It was less than one year old (2)
- ○ It was between one and five years old (3)
- ○ It was between five and ten years old (4)
- ○ It was between ten and twenty years old (5)
- ○ It was more than twenty years old (6)
- ○ Unsure (7)

Q3.4 Roughly how often are new features released for that software?

- ○ Several times a month (1)
- ○ Once per month (2)
- ○ Several times a year (3)
- ○ Once per year (4)
- ○ Less frequently than once per year (5)
- ○ Unsure (6)

Q3.5 Roughly how large was the codebase for that software in terms of lines of code?

- ○ Less than 10k LOC (1)
- ○ 10–100k LOC (2)
- ○ 100k–1M LOC (3)

- ○ 1.1M–10M LOC (4)
- ○ 10+ M LOC (5)

Q3.6 Which of the following describes your role(s) when working on that software? Check all that apply.

- ☐ Software engineer (1)
- ☐ Software architect (2)
- ☐ Testing engineer (3)
- ☐ DevOps engineer (4)
- ☐ Systems engineer (5)
- ☐ Technical lead (6)
- ☐ Project manager (7)
- ☐ Business/requirements analyst (8)
- ☐ Other: (9) _____

Q3.7 What type of organization did you work for when working on that software?

- ○ Industry (1)
- ○ Government (2)
- ○ Government contractor (3)

- ○ Academia (4)
- ○ Other (5)

End of Block: Software system

Start of Block: Experiences performing large-scale refactoring

We define ***large-scale refactoring*** as restructuring software, without introducing functionality, for the purpose of improving non-functional quality or changing architecture. Large scale involves either pervasive changes across a code base or extensive changes to a substantial element of the system (*e.g.*, greater than 10k LOC). One example is the need to partition legacy monoliths into smaller pieces to create separately deployable, scalable, and evolvable units. Another example is restructuring interfaces and communication patterns to enable replacement of a legacy feature by an improved or less proprietary alternative.

Q4.1 According to the above definition, have you ever participated in a large-scale refactoring for that software?

- ○ Yes (1)
- ○ No (2)
- ○ I don't know (3)

Skip To: QID112 If According to the above definition, have you ever participated in a large-scale refactoring for th... = Yes

Skip To: End of Block If According to the above definition, have you ever participated in a large-scale refactoring for th... = No

Skip To: End of Block If According to the above definition, have you ever participated in a large-scale refactoring for th... = I don't know

Refactoring Involvement

Consider the most substantial large-scale refactoring that you were involved in for that software.

Q4.2 What were the business goals of the refactoring? Check all that apply.

- [ ] Reduce the cost of software changes (e.g., bug fixes or new features) (1)
- [ ] Reduce the time to deliver new features and versions (2)
- [ ] Reuse features across systems (e.g., by extracting common services) (3)
- [ ] Replace existing features (4)
- [ ] Reduce reliance on unsupported or outdated technology (e.g., mainframes) (5)
- [ ] Other: (6) _____

Q4.3 What were the technical goals of the refactoring? Check all that apply.

- [ ] Improve understandability of the code (1)
- [ ] Switch from one technology or programming language to another (2)
- [ ] Migrate to a new architecture (3)
- [ ] Move to new deployment environments (e.g., cloud) (4)
- [ ] Enable granular deployment using DevOps (5)
- [ ] Improve use of automation (e.g., unit tests) (6)

☐ Other: (7) _____

Q4.4 How much of your time did you spend on each of the following activities during refactoring?

	N/A (1)	Least 1 (2)	2 (3)	3 (4)	4 (5)	Most 5 (6)
Determining where changes were needed (1)	○	○	○	○	○	○
Choosing what changes to make (4)	○	○	○	○	○	○
Implementing the changes (5)	○	○	○	○	○	○
Generating new tests (6)	○	○	○	○	○	○
Migrating existing tests (7)	○	○	○	○	○	○
Validating refactored code (inspection, executing tests, etc.) (8)	○	○	○	○	○	○
Re-certifying refactored code (9)	○	○	○	○	○	○
Updating documentation (10)	○	○	○	○	○	○

Other (11)	○	○	○	○	○	○

Q4.5 What other significant activities did you perform during refactoring?

_____

_____

_____

_____

_____

Q4.6 Approximately how many total staff days did the **team** spend on the refactoring effort

_____

Carry Forward Unselected Choices from "How much of your time did you spend on each of the following activities during refactoring?"

Q4.7 How challenging did you find each of these activities?

	Least 1 (1)	2 (2)	3 (3)	4 (4)	Most 5 (5)
Determining where changes were needed (x1)	○	○	○	○	○
Choosing what changes to make (x4)	○	○	○	○	○

Implementing the changes (x5)	○	○	○	○	○
Generating new tests (x6)	○	○	○	○	○
Migrating existing tests (x7)	○	○	○	○	○
Validating refactored code (inspection, executing tests, etc.) (x8)	○	○	○	○	○
Re-certifying refactored code (x9)	○	○	○	○	○
Updating documentation (x10)	○	○	○	○	○
Other (x11)	○	○	○	○	○

Q4.8 For the most-challenging activities that you identified, what made those activities challenging?

_____

_____

_____

_____

_____

Carry Forward Unselected Choices from "How much of your time did you spend on each of the following activities during refactoring?"

Q4.9 To what extent do you use tools for the following activities?

	Not at all (1)	Slightly (2)	Moderately (3)	Greatly (4)	Extensively (5)
Determining where changes were needed (x1)	○	○	○	○	○
Choosing what changes to make (x4)	○	○	○	○	○
Implementing the changes (x5)	○	○	○	○	○
Generating new tests (x6)	○	○	○	○	○
Migrating existing tests (x7)	○	○	○	○	○
Validating refactored code (inspection, executing tests, etc.) (x8)	○	○	○	○	○
Re-certifying refactored code (x9)	○	○	○	○	○
Updating documentation (x10)	○	○	○	○	○

Other (x11)	○	○	○	○	○

Q4.10 What tools, if any, did you use to assist your large-scale refactoring efforts?

_____

_____

_____

_____

_____

Q4.11 What kind of automation, if available, would have most improved your large-scale refactoring?

_____

_____

_____

_____

_____

Q4.12 To what extent were the goals of the refactoring achieved?

○ Fully achieved. (1)

○ Mostly achieved. (2)

- ○ Somewhat achieved. (3)
- ○ Not achieved. (4)
- ○ Unsure (5)

Q4.13 To the best of your knowledge, on how many other occasions have there been large-scale refactorings for that project during the time that you worked on it?

- ○ None (1)
- ○ One large-scale refactoring (2)
- ○ Two to four large-scale refactorings (3)
- ○ Five or more large-scale refactorings (4)
- ○ Unsure (5)

End of Block: Experiences performing large-scale refactoring

Start of Block: Refactoring was desired but did not occur

Refactoring Wanted

Q5.1 Have you ever wanted to perform a large-scale refactoring but were unable to?

- ○ Yes (1)
- ○ No (2)

Skip To: End of Block If Have you ever wanted to perform a large-scale refactoring but were unable to? = No

For the remaining questions, think back to the last time that you wanted to perform a large-scale refactoring but were unable to.

Q5.2 For what business reasons did you want to perform a large-scale refactoring? Check all that apply.

- [ ] Reduce the cost of software changes (e.g., bug fixes or new features) (1)
- [ ] Reduce the time to deliver new features and versions (2)
- [ ] Reuse features across systems (e.g., by extracting common services) (3)
- [ ] Replace existing features (4)
- [ ] Reduce reliance on unsupported or outdated technology (e.g., mainframes) (5)
- [ ] Other: (6) _____

Q5.3 For what technical reasons did you want to perform a large-scale refactoring? Check all that apply.

- [ ] Improve understandability of the code (1)
- [ ] Switch from one technology or programming language to another (2)

- [ ] Migrate to a new architecture (3)
- [ ] Move to new deployment environments (e.g., cloud) (4)
- [ ] Enable granular deployment using DevOps (5)
- [ ] Improve use of automation (e.g., unit tests) (6)
- [ ] Other: (7) _____

Q5.4 For what reasons did your organization decide not to perform the large-scale refactoring? Check all that apply.

- [ ] The anticipated value was too low (1)
- [ ] The anticipated cost to perform refactoring was too high (2)
- [ ] The risk of introducing errors during refactoring was too high (3)
- [ ] New features were prioritized over refactoring (4)
- [ ] Staff with sufficient knowledge and skills were not available (5)
- [ ] Refactoring could not be completed quickly enough to meet goals (6)
- [ ] Refactoring would be too disruptive to other development efforts (7)
- [ ] Other: (8) _____

Display This Question:

If If For what reasons did your organization decide not to perform the large-scale refactoring? Check a... q://QID131/SelectedChoicesCount Is Greater Than 0

Carry Forward Selected Choices from "For what reasons did your organization decide not to perform the large-scale refactoring? Check all that apply."

Q5.5 How important were the reasons in your organization's decision not to perform the large-scale refactoring?

	Unsure 0 (1)	Least 1 (2)	2 (3)	3 (4)	4 (5)	Most 5 (6)
The anticipated value was too low (x10)	○	○	○	○	○	○
The anticipated cost to perform refactoring was too high (x1)	○	○	○	○	○	○
The risk of introducing errors during refactoring was too high (x4)	○	○	○	○	○	○
New features were prioritized over refactoring (x5)	○	○	○	○	○	○

Staff with sufficient knowledge and skills were not available (x6)	○	○	○	○	○	○
Refactoring could not be completed quickly enough to meet goals (x7)	○	○	○	○	○	○
Refactoring would be too disruptive to other development efforts (x8)	○	○	○	○	○	○
Other: (x9)	○	○	○	○	○	○

Q5.6 What consequences, if any, did you observe from not performing the refactoring? (*E.g.*, “feature delivery took longer than expected”)

_____

_____

_____

_____

_____

End of Block: Refactoring was desired but did not occur

Start of Block: Further comments

Closing Comments

Q7.1 What are the strengths and weaknesses of the refactoring tools, if any, that you currently use?

_____

_____

_____

_____

_____

Q7.2 How useful would you find a tool that automatically suggests a collection of refactorings that would solve a problem that you specified (*e.g.*, isolate this portion of code so that it can be independently deployed)?

- ○ Not at all 1 (1)
- ○ 2 (2)
- ○ 3 (3)
- ○ 4 (4)
- ○ Extensively 5 (5)

Q7.3 If you have any further comments on this survey, please write them in the following box.

_____

_____

_____

_____

_____

Q7.4 If you check any of the following, please enter your email address below.

☐ Keep me informed about the results of this research. (1)

☐ It is OK to contact me for a follow-up interview (60 minutes) on large-scale refactoring. (2)

Q7.5 Email address (optional):

_____

Please click below to submit the survey.

End of Block: Further comments