Informed Consent Block

Thank you for participating in the survey. We expect that this survey will take 5 minutes of your time.

Who we are:
Researchers at Oregon State University

Our Goal:
This survey is part of a research study to examine how developers resolve merge conflicts. Our goal is that by understanding the difficulties, we can ultimately help improve research and tool support in this area.

Please proceed to the next page to begin.

Demographics Block

What is your programming experience (including professional experience)?

Year(s)

0 5 10 15 20 25 30 35 40 45 50

What is the size of your primary team?

○ Myself

○ 2-5 team members

○ 6-10 team members

○ 11-25 team members

○ 26-50 team members

○ 51+ team members

What is your primary role?

○ Developer

○ Engineer

○ Architect

○ DevOps

○ Designer

○ Data Scientist

○ Chief Technology Officer

○ Other:

Awareness Phase

The following questions are about how you (and your team) become aware of merge conflicts.

Do you monitor for merge conflicts?

○ Yes

○ Sometimes

○ No

How do you monitor for merge conflicts? (Please list processes and tools that you use)

How do you determine the urgency of a merge conflict?

Planning Phase

The following questions are about how you approach resolving a merge conflict.

How often is code ownership a factor in your merge conflict resolution strategy?

○ Always

○ Most of the time

○ About half the time

○ Sometimes

○ Never

What is your first step in trying to understand code involved in a merge conflict?

Have you ever deferred responding to a merge conflict?

○ Yes

○ No

What factor(s) determine whether you defer responding to a merge conflict?

□ Size of the conflicting code

□ Complexity of the conflicting code

□ Number of conflicting code locations

□ Ownership of the conflicting code

□ Work schedule constraints

□ Approaching deadlines

□ Other:

What effect did deferring your response to a merge conflict have on the resolution of the conflict?

Evaluation Phase

The following questions are about how you evaluate whether your merge conflict resolution succeeded.

What condition(s) do you consider to be a successful merge conflict resolution?

- □ All tests pass
- □ Code successfully compiles
- □ Version Control System (VCS) warnings are gone
- □ Merged code is accepted into production codebase
- □ Code looks correct (i.e. visual test passes)
- □ Merged code is approved during code review
- □ Other:

What tools do you use to evaluate the results of performing a merge conflict resolution?

- □ Version Control Systems (e.g. Git, Subversion, CVS)
- □ Release Management Tools (e.g. Chef, Puppet, Salt)
- □ Continuous Integration (e.g. TravisCI, Jenkins, TFS)
- □ DevOps Tools (e.g. Nagios, Monit, Kabana)
- □ Program Analysis Tools (e.g. Coverity, CodeSonar, Roslyn)
- □ Other:

How often does your first attempt at resolving a merge conflict fail?

- ○ Very frequently
- ○ Somewhat frequently
- ○ Occasionally
- ○ Somewhat infrequently
- ○ Very infrequently

If your first attempt at resolving a merge conflict fails, what backup strategies do you use?