

# Questionnaire for the Priority Ranked Inspection Process

Thank you for your participation. As a reminder, your participation in this research is voluntary. All references to data gathered will be made anonymously. Please indicate your level of agreement to the following statements the best of your ability. In the following statements, the terms Inspections and Code Reviews are used interchangeably.

1. Inspections are an important part of the CSDL development process. (circle one)

(1) Strongly Disagree (2) Disagree (3) Neutral (4) Agree (5) Strongly Agree

2. The most important goal of the CSDL inspection process is to remove defects. (circle one)

(1) Strongly Disagree (2) Disagree (3) Neutral (4) Agree (5) Strongly Agree

3. Which of the following would most likely contain the most critical defects: (circle all that apply)

- (1) Newly created code
- (2) Code that has no (or very few) unit tests
- (3) Code that has low coverage
- (4) Code that was developed by a new developer
- (5) Code that only one developer has worked on
- (6) Code that has a large number of dependencies
- (7) Old code
- (8) Other: \_\_\_\_\_

4. When I request an inspection, I generally volunteer code that is: (circle all that apply)

- (1) Newly created code
- (2) Code that has low coverage
- (3) Code that no one has seen before
- (4) Code that has no unit test
- (5) Old code
- (6) Other: \_\_\_\_\_

5. When I request an inspection, I use Hackystat to help me pick a piece of code to inspect? (circle one)

(1) Strongly Disagree (2) Disagree (3) Neutral (4) Agree (5) Strongly Agree

6. Inspection should only occur on newly created code. Old code that has already been released does not need to be inspected. (circle one)

(1) Strongly Disagree (2) Disagree (3) Neutral (4) Agree (5) Strongly Agree

7. Please provide a numerical ranking of the packages provided that represents what packages you believe should be inspected. For example, the following table shows that package `com.example.bar` is the highest priority package that should be inspected. And `com.example.bax` has the lowest priority ranking.

Ranking	Package
2	<code>com.example.foo</code>
1	<code>com.example.bar</code>
3	<code>com.example.baz</code>

Enter your own subjective rankings in this table. Please use values from 1 through 13 and do not use one number twice. After finishing the rankings we will discuss you justifications for those rankings.

Ranking	Package
	<code>org.hackystat.app.telemetry.analysis</code>
	<code>org.hackystat.app.telemetry.analysis.selector</code>
	<code>org.hackystat.app.telemetry.config</code>
	<code>org.hackystat.app.telemetry.config.core</code>
	<code>org.hackystat.app.telemetry.processor.evaluator</code>
	<code>org.hackystat.app.telemetry.processor.parser</code>
	<code>org.hackystat.app.telemetry.processor.parser.impl</code>
	<code>org.hackystat.app.telemetry.processor.query</code>
	<code>org.hackystat.app.telemetry.processor.query.expression</code>
	<code>org.hackystat.app.telemetry.processor.reducer</code>
	<code>org.hackystat.app.telemetry.processor.reducer.impl</code>
	<code>org.hackystat.app.telemetry.processor.reducer.util</code>
	<code>org.hackystat.app.telemetry.processor.stream</code>

**When finished wait for further instructions.**

After finishing the rankings we will discuss you justifications for those rankings.

Ranking	Justification
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

8. To the best of your knowledge, please provide the top 5 modules that you think need to be inspected and the top 5 modules that you think do NOT need to be inspected. Below is a list of all Hackystat modules.

Modules that need to be inspected		
Ranking	Module	Justification
1		
2		
3		
4		
5		

Modules that do NOT need to be inspected		
Ranking	Module	Justification
1		
2		
3		
4		
5		

- Hackystat Modules
  - hackyAnt
  - hackyCGQM
  - hackyCli
  - hackyDependency
  - hackyEclipse
  - hackyHPCS
  - hackyIssue
  - hackyJBuilder
  - hackyJupiter
  - hackyKernel
  - hackyPerf
  - hackyPRI
  - hackyPrjSize
  - hackyReport
  - hackyReview
  - hackyStatistics
  - hackyStdExt
  - hackyTDD
  - hackyTelemetry
  - hackyVCS
  - hackyVim
  - hackyZorro

9. To the best of your knowledge, please provide the top 5 workspaces that you think need to be inspected and the top 5 workspaces that you think do NOT need to be inspected. Attached is a list of all Hackystat workspaces.

Workspaces that need to be inspected		
Ranking	Workspace	Justification
1		
2		
3		
4		
5		

Workspaces that do NOT need to be inspected		
Ranking	Workspace	Justification
1		
2		
3		
4		
5		