Technical Debt survey

1.
This survey is about how technical debt is managed and understood in software organizations. Your participation is important to allow us to correctly characterize the nature of the problem. The results will be used to inform recommendations for better technical debt management practices and tool support.

This survey is entirely voluntary. There are 20 questions. The survey should take about 15 minutes to complete. All information provided in the survey will be treated as confidential information and will not be shared.

This study is being conducted by the Software Engineering Institute at Carnegie Mellon University, a federally-funded research and development center. For questions please contact td@sei.cm.edu.

Demographic Information

2. Are you familiar with the concept of "technical debt"?

- ○ Yes
- ○ No

3. The term Technical Debt was coined by Ward Cunningham: "Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite... The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt."

Steve McConnell's definition of technical debt has increasingly been accepted: *"A design or construction approach that's expedient in the short term but that creates a technical context in which the same work will cost more to do later than it would cost to do now (including increased cost over time)."*

Do you wish to continue with the survey?

- ○ Yes
- ○ No

4. The questions which follow refer to the software system you are currently working on. If you are not currently working on a system, refer to the one you most recently worked on. If you are working on multiple systems, pick one that is representative.

5. What is (will be) the size of the system? (LOC = lines of code)

- ◯ less than 10k LOC
- ◯ 10–100k LOC
- ◯ 101–1M LOC
- ◯ 1.1M–10M LOC
- ◯ 10+ MLOC

6. What is the current age of this system, beginning from initial design and planning?

- ◯ Less than 1 year
- ◯ 1-2 years
- ◯ 3-5 years
- ◯ 6-10 years
- ◯ More than 10 years

7. What best describes your role on this project?

- ◯ Developer/Programmer/Software Engineer
- ◯ Tester/QA
- ◯ Team Lead/Organizational Lead
- ◯ Scrum Master
- ◯ Software Architect
- ◯ Project Manager/Product Owner
- ◯

- Business/Requirements Analyst
- Internal Consultant
- External Consultant
- Executive
- Other

8. What is the total number of people on the project (include technical staff and business staff)?

- Less than 5 people
- 5-9 people
- 10-20 people
- More than 30 people

9. What type of system are you developing (did you develop)?

- Real-time control system
- Data management system
- Embedded system
- Interactive Web site
- Data analysis system
- Modeling and simulation system
- Enterprise Information system
- Systems integration
- Other

What is Technical Debt?

10. In your own words, how would you define technical debt?

11. Please give an example of technical debt that had significant impact on your project. We would like to know its cause, symptoms and impact.

12. Thinking of your most recent project, how strongly do you agree with each of the following statements?

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
Technical debt is mostly a business concern	○	○	○	○	○
Technical debt is mostly a technical concern	○	○	○	○	○
Technical debt is both a business and a technical concern	○	○	○	○	○
Lack of awareness of technical debt was/is a problem	○	○	○	○	○
Incurring technical debt is strategically used to support the business objectives (e.g. low cost/short schedule)	○	○	○	○	○
Technical debt is explicitly tracked/managed	○	○	○	○	○
Dealing with the consequences of technical debt has consumed a painful chunk of project resources	○	○	○	○	○
Measuring, tracking, monitoring technical debt has consumed a painful chunk of project resources	○	○	○	○	○

13. The time it appears to be taking (took) to deliver software to the customer is/was:

- ○ Significantly longer than expected
- ○ Longer than expected
- ○ About average
- ○ Shorter than expected
- ○ Significantly shorter than expected
- ○ No outcome

14. The number of defects experienced by this project appears to be (was):

- ○ Much higher than expected
- ○ Higher than expected
- ○ About average
- ○ Lower than expected
- ○ Much lower than expected
- ○ No defects!

15. On this project, what is causing technical debt?

	Often	Sometimes	Never	Don't know/not applicable
Deployment and build infrastructure	○	○	○	○
Dependencies on external software packages	○	○	○	○
Obsolete code	○	○	○	○
Inadequate testing	○	○	○	○
Module dependencies	○	○	○	○

Low code quality				
Bad architecture choices				
Obsolete technology				
Dependencies on external team's code				
Other				
Code duplication or repetitive edits				

Managing Technical Debt

16. Who is aware of and who manages technical debt?

	Aware	Aware and manages	Not aware	Role does not exist
Developer/Programmer/Software Engineer				
Tester/QA				
Business Manager				
Team Lead/Organizational Lead				
Scrum Master				
Software Architect				
Project manager/Product owner				
Business/Requirements Analyst				
Internal Consultant				
Executive				
Other				

17. What metrics do you use to quantify technical debt on this project? Check all that apply.

- [ ] Estimated effort

- [ ] Story points
- [ ] Outstanding defects over time
- [ ] Cycle time, such as iteration or release duration
- [ ] Number of items in progress, such as tickets
- [ ] Change in number of features released per iteration (e.g., velocity)
- [ ] Cyclomatic complexity
- [ ] Defect density
- [ ] Test coverage
- [ ] Other process metric
- [ ] Other product metric
- [ ] Don't measure

18. What tools, if any, are used to analyze technical debt on this project?

19. At what point do/did you identify technical debt for this project? Check all that apply and add any others.

- [ ] As part of overall risk management process
- [ ] As part of development backlog by explicitly identifying technical debt items
- [ ] As part of the development backlog without explicitly identifying technical debt items
- [ ] Using tools periodically such as static analysis tools that provide technical debt measurement
- [ ] As a result of slowing project cadence
- [ ] Retrospectives
- [ ] Other

- [ ] Not identified

20. What strategies do you use to pay down technical debt or interest in this project? Check all that apply and add any others.

- [ ] We pay down technical debt only when it becomes an immediate road block.
- [ ] We prioritize for pay down per sprint.
- [ ] We integrate pay down with our release planning cycle.
- [ ] We use risk management to determine what debt should be paid down and when.
- [ ] Another team is responsible for prioritization and pay down.
- [ ] Other
- [ ] We do not pay down technical debt.

Q41.
How strongly do you agree with each of the following statements?

	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
Technical debt reifies an abstract concept	○	○	○	○	○
Technical debt is not simply bad quality but includes strategic architectural decisions	○	○	○	○	○
Technical debt can be introduced by context shift	○	○	○	○	○
Defects are not technical debt	○	○	○	○	○
Lack of process is not technical debt	○	○	○	○	○
New features not yet implemented are not technical debt	○	○	○	○	○
Technical debt implies dealing with both principal and interest	○	○	○	○	○

Technical debt assessment depends on future outcomes	○	○	○	○	○
Technical debt is not directly measureable	○	○	○	○	○
Technical debt should not be completely eliminated	○	○	○	○	○
Technical debt should not be treated in isolation from the software development context	○	○	○	○	○
Technical debt can be a wise investment	○	○	○	○	○

21. If you would like to be informed about the results of this research, please enter your email in the following box.

22. If you would be willing to participate in a follow-up interview (45 minutes) to share your perspectives and anectodes on managing technical debt, please enter your email in the following box.

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