

Assignment #2 Sample solution

Due date: 10/10/2023 at 23.55

Question #1

Given is the following statement, while version control systems have advanced in the last couple of years, fully automated merging remains a challenge and is typically not recommended.

Is the above statement correct – briefly justify your answer (max 50 words)

Yes.

While fully automated merging might be able to check for syntactical correctness it cannot check for semantical correctness. Also if we have conflicting code regions (meaning statements which were modified in parallel), fully automated merging might override changes made by another users.

Question #2

In a recent department meeting your new boss made the following statement. I just read a research paper which included the following statements. “The key objectives of Continuous Integration is (1) to reduce the frequency of builds and therefore (2) making it easier to identify and fix programming and integration errors”

Clearly indicate, for each of the two statements, if you agree / disagree with it and correct them if necessary. Briefly justify your answer (1-2 sentences).

Disagree

Key objective (1) is wrong. It is just the opposite. The goal is to increase the frequency of builds, in order to make it easier to identifier and fix any potential problems.

Key objective (2) is correct. Given the more frequent integrations, fewer changes have been made since the last successful build, therefore making it easier to localize the problems, since only fewer code changes have to be analyzed.

Question #3

You are attending a workshop and one of the presenters makes the following statements. 1.) Dependency management systems such as Maven will help managing build dependencies and the automation of the build process. 2.) Such build management system will also remove technical debt, by eliminating all built errors.

Clearly indicate, if you agree / disagree with the above statement. Briefly justify your answer (1-2 sentences).

- 1.) This is correct. Build management tools allow for the automation of the build process and manage the dependency needed to build a system.
- 2.) Build errors are not technical debt. Technical debt is not related to system faults or errors. And build management tools do not address issues of technical debt. They are typically bad coding practices which will make the comprehension and longterm maintenance of software systems more difficult.

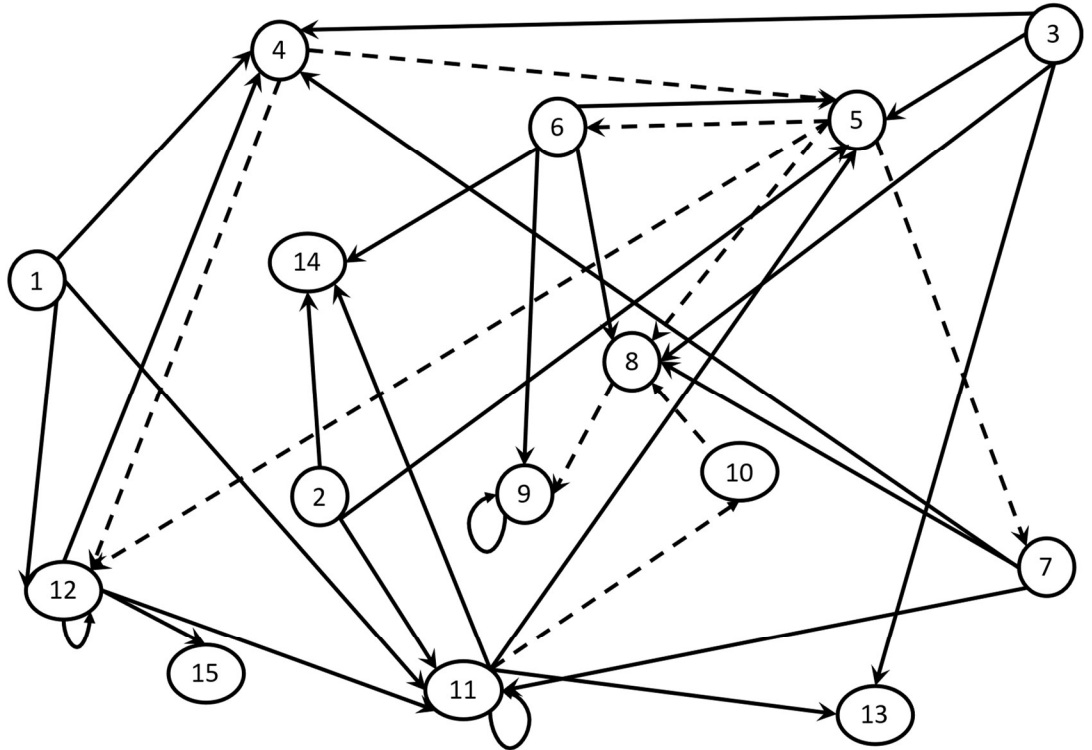
Question #3

Given is the following program and PDG

```

VOID MAIN ()
{
1.  CIN>>X;
2.  CIN>>Y;
3.  CIN>>Z;
4.  IF (Y<Z)
    {
5.      WHILE (Z>X)
        {
6.          Y = 100;
7.          CIN >>Z;
8.          IF (Z<Y)
            {
9.              Y = Y+Z;
            }
10.         ELSE
            {
11.             Z=Z+X+Y;
            }
        }
    }
12. X++;
}
13. COUT<<Z;
14. COUT <<Y;
15. COUT <<X;
}

```



Verify the above static PDG for correctness – **Note:** In situation where you have a transitive control dependency you can ignore it => A is control dependent on B and B is control dependent on C then you do not have to include the control dependency between A and C.

Indicate in the table next page what is missing/wrong

