

Quiz 05: Continuous Integration and Pair Programming

Due Mar 7 at 11pm**Points** 50**Questions** 11**Available** until May 15 at 11:59pm**Time Limit** None

Instructions

You **may** use the slides from the lecture and other sources to answer these questions. Please be sure to cite any references but be sure to answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	7,587 minutes	50 out of 50

Score for this quiz: **50** out of 50

Submitted Mar 7 at 10:22pm

This attempt took 7,587 minutes.

Question 1

5 / 5 pts

Describe two of the synergistic behaviors of Pair Programming

Your Answer:

The two synergistic behaviors of Pair Programming are:

- Pair Courage – It is the courage to start coding confidently knowing that someone is there to monitor you and help you if you get stuck or to help you improve your code. It's always helpful to get advice/feedback from others.
- Pair Learning – By working together we get to know how different the other person solves the given problem from you. This way we can learn and share new ideas, techniques, and tricks with each other. By learning new things and combining them with our knowledge we can increase our efficiency of working.

1. Pair pressure
2. Pair negotiation
3. Pair courage
4. Pair reviews
5. Pair debugging
6. Pair learning
7. Pair trust

Question 2

5 / 5 pts

Describe two of the myths of pair programming and explain why each is not true.

Your Answer:

The first myth of pair programming is that it will take twice the amount of time by two people. Whereas a single person can do the same task in half the time of the two people. This is not true because when we are working alone and if we get distracted from work there is no one to tell us to get back to work therefore we waste time whereas when someone is watching over us, we can't slack off or take a break or get distracted because the other person will always be there to tell us to get back to work. Hence our time completion of the given assignment will reduce significantly and might be even better than a single person working on it.

The second myth of pair programming is that the person programming won't be able to concentrate which is not true because when we are working in pairs it creates a certain sense of pressure on the person coding which in turn makes him more focused and concentrate for long period of time.

One myth for pair programming is that it only works effectively with certain people. This is a myth because diversity of thought is proven to improve innovation, and that people who refuse to take criticism are not working for the best of the project anyway, whether or not they are pair programming.

Another myth is that the individual has to share all of their accomplishments. This is a myth because many organizations do peer review, and because individuals can still have certain tasks assigned to them that they are credited with.

Question 3**5 / 5 pts**

Pair programming is not always productive. Identify one situation where pair programming may not be effective.

Your Answer:

If you have a very simple bug fix or a straightforward feature to implement, it can be expensive to put two programmers on that task. Additionally, it can be hard for two programmers to stay interested on a task that takes minimal thought-processing power. If you have an environment with a large test suit or integration process, pair-programming can further increase the resource expenditure. I would argue this is where pair programming struggles the most. Simple tasks can be overcomplicated by pair-programming.

slide 13: Pairs with mismatched experience may not be effective

Question 4

5 / 5 pts

Describe two best practices of Continuous Integration

Your Answer:

The two best practices of Continuous Integration are

- Test in production environment: -

We should always make an exact clone of the production environment this includes database software and version, operating system, libraries, IP address and port and hardware and only then we should start making changes and do the testing, because if we write code and do testing in a

completely different environment from our production there might be chances of having different test results from different environments.

- Developers frequent commit to mainline: -

A very good way to find out if there are any conflict between two developer code is by committing frequently. Merge issues may become very difficult to solve if stayed undetected for weeks. Whereas conflicts can be found very easily if commits are done frequently, and bugs can be resolved by doing self-testing. A general is to commit everyday if possible.

Maintain a single source code repository, Keep “everything” needed to build in the repository, Automate the build process, Automate testing of new builds, Frequent code commits from all developers, At least daily if not more frequent.

Question 5

5 / 5 pts

Compare and contrast deferred integration and continuous integration.

Your Answer:

Firstly, in deferred integration nobody knows when the integration will be completed. Whereas in continuous integration the integration is done in a single day since the team members integrate code almost daily.

Secondly, in deferred integration if bugs are found they usually take a lot of time to get fixed which in turn reduces morale of the developers which reduces their work efficiency. Whereas in continuous integration bugs take much less time to be fixed as compared to deferred integration.

Thirdly in deferred integration different parts of the project are built in different environment than the actual project which leads to different results when integrated with the main project. Whereas in continuous integration all the parts are built in an exact replica of the main project meaning they have the same database software and version, operating system, libraries, IP address and port and hardware.

Deferred Integration happens in plan driven methodologies. The integration only starts to begin after all of the development happens. This is much different that Continuous Integration where the build process is automated and happens during development so there is always a working product.

Question 6

5 / 5 pts

Describe the roles in pair programming.

Your Answer:

In pair programming two people are working together sitting in front of the same computer and they both have different roles. They are “Driver” and “Navigator”. The driver is the one who writes the code for the task to be completed. While the navigator is the one who watches over the driver and find/catches mistakes done by the driver. He also suggests alternative method to complete the given task and tests the code completed by the driver to find any bugs or errors. Both the driver and navigator switch roles often mainly every 15 to 20 minutes so they get to experience both the roles. This also increases productivity and efficiency. It works best if both of them are co-located and have same amount of work experience.

Question 7**5 / 5 pts**

What is pair debugging? Why does it help?

Your Answer:

Pair debugging basically means to find and fix bugs together on the code written by the developers. It helps in ways to learn different things such as Software test engineers get to learn about the architecture, design, and coding elements of the system. Developers get to learn about the nuances of debugging and testing. Pair debugging also provides an opportunity to identify some of the weak modules or components that need preventive maintenance. It also results in effective debugging because software test engineers learn more when they work along with developers. This provides them an opportunity to think beyond the obvious and create test scenarios with the right test data set.

Sometimes you need to describe the problem to someone else in order to solve it. Thinking out loud. An intelligent partner will ask questions that you should have asked yourself

Question 8**5 / 5 pts**

What is pair courage? Why does it help?

Your Answer:

Pair courage is the courage to start coding confidently knowing that someone is there to monitor you and help you if you get stuck or to help you improve your code. It's always helpful to get advice/feedback from others. It helps because we get to learn new things from our mistakes and the time to complete a task is significantly reduced because one developer's shortcoming is overcome by the other developer.

It is easier to get started if you know you have help. Feedback from your partner is encouraging. It's okay to admit you don't know something. Two is better than one.

Question 9**5 / 5 pts**

What is pair pressure? Why does it help?

Your Answer:

Pairs Pressure is a synergetic behavior of pair programming. It helps to keep both the developers focus on task. It helps by increasing the productivity of work because one is less likely to be distracted by other activities when someone is watching over. It creates a certain pressure on the person writing the code to work efficiently and not waste the other person's time. Since both the developers respect each other they both follow the standard processes more readily and do not procrastinate.

Pair pressure forces individuals to feel guilty about slacking off and taking frequent breaks because they have someone constantly working with them and watching over them. Having another person to engage with also keeps individuals' minds engaged so they focus more on coding standards as well.

Question 10**5 / 5 pts**

Williams and Upchurch report which of the following benefits for pair-programming for Computer Science students:

Correct!☒ Students complete assignments more quickly with higher quality**Correct!**☒ Students are happier and less frustrated☐ Students learn more slowly**Correct!**☒ Students are able to help each other to understand the concepts☐ I didn't read the paper**Question 11****0 / 0 pts**

"I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source."

Correct!

☒ True

☐ False

Quiz Score: **50** out of 50