***Q5)***

***Scenario 1: During a release cycle, you find that the number of bugs reported in the new***

***features are significantly higher than in the previous releases. The development team claims that***

***the features were developed according to specifications.***

***Task: Write a response detailing how you would handle this situation, including steps to identify***

***the root cause and possible solutions to reduce the number of bugs in future releases***

This is an area where Agile software development methodology works great

In most of such situations, the key areas to note is that developers and testers are working for the same goal and as a team. With that in mind, I will gather up the following points and hold a meeting to clarify the bugs and causes:

**Prioritization**

1. One of the key importance for testers is setting severity of bugs.
   1. This is the order in which developers start organizing where to start fixing
   2. Major bugs in a released product are frowned upon, while minor bugs can be fixed with hot-patches
2. Identify recurring patterns – If a similar bug has happened in another section, it could a problem with the unit. Unit-tests as helpful to avoid such issues.

**Clarify Requirements**

1. Ambiguous or Misinterpreted requirements are one of the causes for bugs that are not agreed upon.
   1. Setup ‘*Expected Outcome*’ with illustrations if it demonstrates the thought process better

**Test Coverage:**

1. Was there a change in test case coverage?
   1. Perhaps the testing team has evolved and is using a more comprehensive testing procedure
   2. This might bring more minor bugs to light which were not seen before in the previous release cycle

**Environment discrepancy**

1. Is the environment similar during development and testing?
2. Did the development team take into account multiple testing environments?
3. Do the APIs work with legacy system (if applicable)?

***With the following sorted out, it is important to clarify these during a Review meeting. This would help to figure the root cause of the bugs and aid in planning how to improve the system in the future.***

***Scenario 2: You are part of a QA team that is transitioning from manual to automated testing.***

***However, some team members are resistant to this change, citing a lack of experience in***

***programming.***

***Task: Describe how you would address this challenge to ensure a smooth transition to***

***automated testing while keeping the team motivated and engaged.***

While programming is natural for many and enjoy it like myself, I do find some are worried about it.

This would be a task to provide the employees **confidence in their abilities**, offer them help with **training**, conducting seminars/workshops to **teach with examples**.

1. **Open communication**: Discuss the importance and necessity of automated testing (its pros and cons). While manual testing cannot be replaced fully, automated testing takes a lot of load off the engineers.
2. **Mentoring/Training**: Help with real-world examples and example scripts to train the engineers. Make it easy for a beginner to enter into the field of automation
3. **Start small**: Make the tasks into small sub-sets to allow implementation by newly trained engineers. It would be accepted more readily if they are not overwhelmed
4. **Positive Incentive**: Most employees would work when they know the results products a positive outcome. Create a supportive and positive culture.

In most practical cases, engineers who are inclined to programming take on automated test creation tasks, while others find their strengths to be in manual testing where domain knowledge and testing experience help to find niche bugs.