

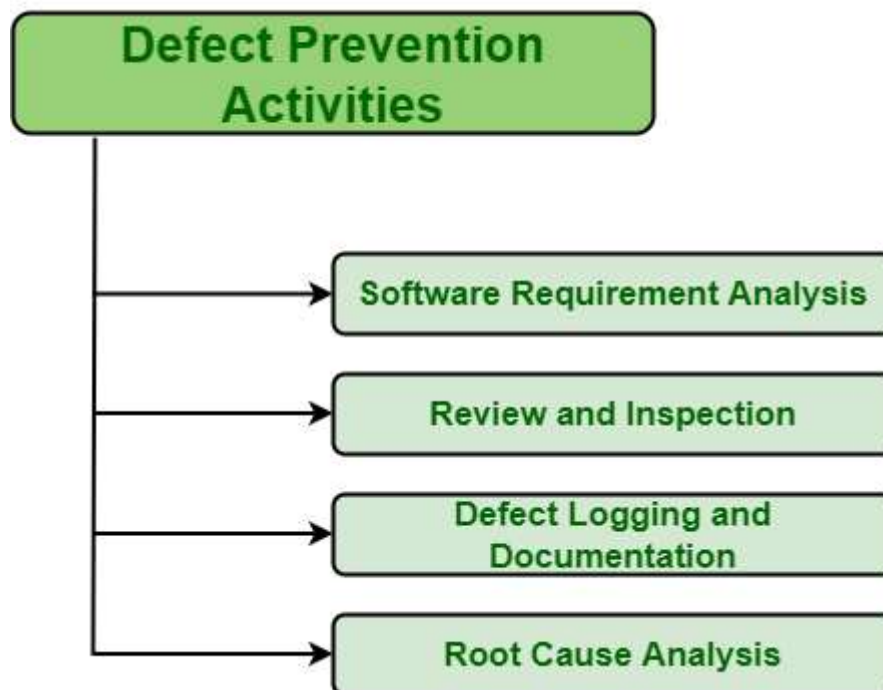
Defect Prevention Methods and Techniques

Defect Prevention

is basically defined as a measure to ensure that defects being detected so far, should not appear or occur again. For facilitating communication simply among members of team, planning and devising defect prevention guidelines, etc., Coordinator is mainly responsible. Coordinator is mainly responsible to lead defect prevention efforts, to facilitate meetings, to facilitate communication between team members and management, etc. DP board generally has a quarterly plan in which sets some goals at organization level. To achieve these goals, various methods or activities are mostly used and carried out to achieve and complete these goals.

Methods of Defect Prevention :

For defect prevention, there are different methods that are mostly used over a long period of time. These methods or activities are given below :



Defect Prevention Methods and Techniques

1. **Software Requirement Analysis** : The main cause of defects in software products is due to error in software requirements and designs. Software requirements and design both are important, and should be analyzed in an efficient way with more focus. Software requirement is basically considered an integral part of [Software Development Life Cycle \(SDLC\)](#). These are the requirements that basically describes features and functionalities of target product and also conveys expectations or requirement of users from software product. Therefore, it is very much needed to understand about software requirements more carefully, If requirements are not understood well by tester and developers, then there might be chance of occurring of issue or defect in further process. Therefore, it is essential to analyze and evaluate requirements in more appropriate and proper manner.
2. **Review and Inspection** : Review and inspection, both are essential and integral part of software development. They are considered as powerful tools that can be used to identify and remove defects if present before their occurrence and impact on production. Review and inspection come in different levels or stages of defect prevention to meet different needs. They are used in all software development and maintenance methods. There are two types of review i.e., self-review and peer-review.
3. **Defect Logging and Documentation** : After successful analysis and review, there should be records maintained about defects to simply complete description of defect. This record can be further used to have better understanding of defects. After getting knowledge and understanding of defect, then only one can take some effective and required measures and actions to resolve particular defects so that defect cannot be carried further to next phase.
4. **Root Cause Analysis** : Root because analysis is basically analysis of main cause of defect. It simply analysis what triggered defect to occur. After analyzing main cause of defect, one can find best way to simply avoid occurrence of such types of defects next time.
5. **Static Code Analysis**: To find any problems in the source code without running the program, use automated techniques for static code analysis. Before the code is even compiled, these tools can

detect typical programming errors, coding standard violations and other problems.

6. **Pair programming:** It involves two programmers sharing a workstation, where one writes code while the other goes over each line as it is written. This real-time communication between team members promotes constant feedback and helps in the early detection of mistakes.
7. **Test-Driven Development (TDD):** Prior to writing any code, write automated tests. This method helps identify errors early in the development process and guarantees that the code complies with the requirements.
8. **Training and Skill Development:** Investing in continuous training and skill development for team members is recommended. Skilled developers are better able to follow best practices and are less prone to make frequent mistakes.
9. **Checklists:** To make sure that crucial actions are not missed during the various stages of development, use checklists. Developers and reviewers can quickly confirm that established standards are being followed by using checklists as a reference.