**Two Types of Code/Peer Reviews**

1. **Formal Inspections (Fagan Inspections)**  
   Formal inspections are highly structured and systematic peer review processes designed to detect defects early in the development cycle. This method involves several defined roles, including a moderator who oversees the process, an author who created the code, reviewers who evaluate the code, and a scribe who records defects [1]. The inspection follows a detailed sequence: planning, overview, preparation, the inspection meeting itself, rework by the author, and a follow-up to ensure corrections were made.  
   **When Used:**  
   Formal inspections are typically used in the development of critical systems, such as aerospace control software, medical device firmware, or financial transaction systems domains where software defects can lead to significant harm or cost. They are also essential in environments that require compliance with industry regulations.  
   **Why:**  
   This approach is used because it significantly reduces the number of bugs before the testing phase, enhances software quality, and ensures strict adherence to coding and documentation standards. The thoroughness of this method is ideal for projects where safety, reliability, and traceability are paramount.
2. **Pull Request (PR) Reviews (GitHub/GitLab Style)**  
   Pull request reviews are an informal, asynchronous type of peer review common in modern version control platforms like GitHub and GitLab [2]. Developers submit code changes via pull or merge requests, which are then reviewed by team members who comment, suggest changes, or approve the code.  
   **When Used:**  
   This type of review is widely used in Agile and DevOps workflows, particularly in continuous integration/continuous deployment (CI/CD) pipelines, and in collaborative open-source projects.  
   **Why:**  
   PR reviews offer a faster, flexible feedback loop that encourages team collaboration without the need for real-time meetings. They support iterative development and are well-suited for dynamic, distributed teams.

Sources:

1. <https://www.atlassian.com/agile/software-development/code-reviews>
2. https://smartbear.com/learn/code-review/what-is-code-review/