3. Concepts of GitHub, GitLab, and BitBucket.

GitHub: It is a Git-based version control platform. It has a simpler user interface that quickly gets developers in the flow of working with the Git algorithm. GitHub pushes and pulls requests and merges versions quickly. The free version of GitHub allows developers to work with private and public repositories, contribute to them, and collaborate. The major functionality of GitHub is repository branching and forking, pull and merge requests, and codebase cloning. The main functionality works quickly – perhaps, the fastest among the three. Developers can quickly upload files to remote repositories and receive immediate notifications from other contributors.

GitLab: GitLab was founded in 2011 as an alternative to GitHub and BitBucket. It is neatly packed into a great UI regardless of its versatility. GitLab provides seamless support to CI pipeline and Docker. Other code repositories offer CI features in their premium versions, but in GitLab, CI support is included in a free plan. Also, developers can host CI GitLab integration on their servers. However, accessing the main functionality and performing fast pushes and merges can be a struggle.

BitBucket: BitBucket was launched in 2008. It didn’t support Git-based version control up to 2011. This service provides a provision to host an unlimited number of private repositories for smaller teams. However, BitBucket’s UI was not as straightforward as GitHub and the functionality was underdeveloped. The main distinction of this platform is its integration with Jira and Asana. Developers can easily monitor BitBucket activity in Jira, measure their productivity, and connect organization boards to collaborative repositories.

4. Industrial practices of using Git.

Git is an integral part of industrial software development practices, providing version control and collaboration capabilities that are crucial for managing codebases, coordinating development efforts, and ensuring the stability and reliability of software projects. Git is a tool used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to track changes in the source code, enabling multiple developers to work together on non-linear development. The stages for a good workflow with Git are the following: Track master, create a feature branch, add our changes, commit our changes, add more changes, push the feature branch, create a pull request, get all changes, merge all the changes from master, select final commit, update remote branch, merge pull request, remove feature branch.

Feature branching is a common approach, allowing teams to develop multiple features concurrently in isolated branches without disrupting the main codebase. Merge strategies are carefully chosen to ensure smooth integration of feature branches. Collaborative development is facilitated through pull requests, providing a space for thorough code reviews before merging changes. Forking is employed for experimentation, allowing developers to test new ideas independently before proposing modifications to the main project. Git integrates with project management tools to unify workflows, linking code changes with relevant issues, tasks, and project milestones for enhanced traceability and accountability.

Strategic choices in selecting Git hosting platforms, like GitHub, GitLab, or Bitbucket, are made based on collaboration features and integrations, aligning with specific project and organizational needs. These refined practices underscore the importance of Git in modern industrial software development, emphasizing collaboration, automation, and the integration of tools and processes to deliver high-quality, reliable software.