GIT STRUCTURE

		advantages	disadvantages
Git flow branching		Allows to work on multiple releases in parallel Each release in tagged and individually tested Allows multiple developers to work on the same feature Allows for jumping between work for current and future releases	Not suitable for continuous delivery or continuous deployment many branches to maintain can load to a technical debt build-up
Trunk based development	Emphasizes continuous integration and delivery by keeping the main branch stable at all times. Developers use feature flags or toggles to control the visibility of new features. Developers work directly to the main branch, making smaller and frequent commits	Simplicity: ideal for small teams fast feedback: teams can quickly identify issues and resolve them easy to use: minimal setup	Lack of control: does not provide strict control over the development process Risk of problems: undisciplined teams that keep feature branches open for weeks risk serious problems when trying to merge not suitable for complex projects
Github flow	Developers work on a single branch (main/master). Developers create feature branches to work on specific changes. Once the feature is complete it is merged back into the main branch through a pull request	Simplicity: ideal for small teams Fast feddback: the team can quickly identify issues and resolve them Easy to use: minimal setup	Lack of control: does not provide strict control over the development process risk of problems: undisciplined teams that keep feature branches open for weeks risk serious problems when trying to merge not suitable for complex projects
Release branching	Use to manage the	Bug fixes	Complexity: can be

	release of new features and bug fixes. Creation of a separate branch from the main development branch to isolate the changes intended for a specific release.	configuration updates isolation: teams focus on the release without worrying about changes to the main branch stability: allows teams to ensure that the release	complex and time consuming Risk of merge conflicts: especially if changes are made to both the main branch and the release branch at the same time maintenance overhead: time consuming
Three flow branching	Uses 3 branches: master, candidate and release	Reduced complexity	Avoids feature branches: more difficult to track changes