|  |  |
| --- | --- |
| ステージングエリア | ・ファイルをコミットする前に変更内容を一時的に登録しておく場所  ステージングエリアはなぜ必要か？  ・1つのコミットは1つの意味を束ねることで，あとからの履歴の修正が容易になる．  ・関連性の強い変更のみを選別して焦点が明確なスナップショットを作成してからコミットを行うことができるようにするため．  ・関連性の強い変更だけをまとめてステージングすることで，論理的に整理されたコミットを保つことができる． |
| スナップショット | ある時点でのソースコードや、ファイル、ディレクトリ、データベースファイルなどの状態を抜き出したもののこと |
| コミットメッセージ | 1行目：コミットする変更内容の要約を記述  2行目：空行  3行目：変更の概要やその箇所を詳細に記述 |
| リモートリポジトリ | 他の人が参照できるリポジトリ |
| ローカルリポジトリ | 自分の手元にあるリポジトリ |
| プッシュ | ファイルをローカルリポジトリ→リモートリポジトリにコピーすること |
| フェッチ | リモートリポジトリのファイルをローカルリポジトリに取り込むこと |
| ブランチ | ブランチを使うことで開発を分岐させることができる |
| チェックアウト | ブランチを切り替えること |
| マージ | 各ブランチの開発内容を併合すること |
| ブランチとプッシュ | ローカルリポジトリで開発を進め，一段落したら開発中のブランチの内容を，リモートリポジトリに同期する．  ・ブランチを指定してプッシュする仕組みによって，開発者はローカルリポジトリで好きなブランチを作成して開発を進めることができる． |
| プル | リモートから各ブランチの最新の情報をフ取り込み(フェッチ)  +ローカルリポジトリの特定ブランチに併合(マージ)すること |
| ブランチを用いたチーム開発の流れ | 1. メインブランチからトピックブランチを作る 2. 機能実装を行い，コミットする 3. 定期的にトピックブランチをリモートリポジトリにプッシュする 4. 定期的にメインブランチの内容をフェッチしてローカルリポジトリにマージする 5. トピックブランチをメインブランチにマージする 6. リポートリポジトリのメインブランチにプッシュする |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |