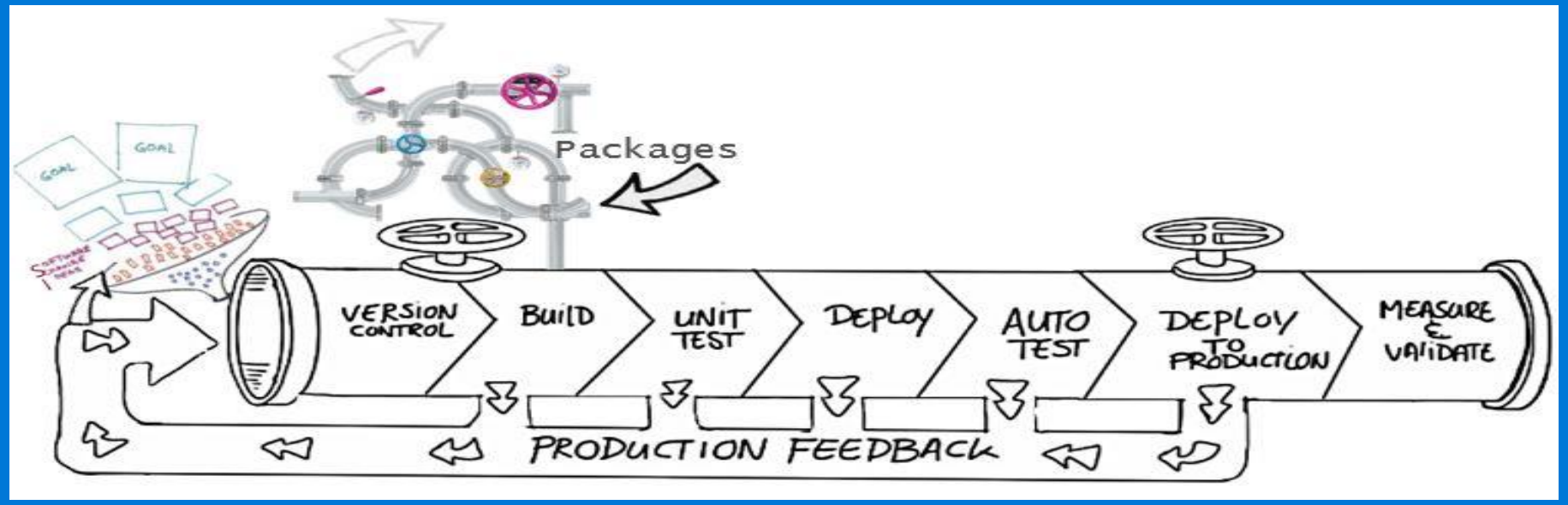
Contrôleur de code source (Source Contrôle)

Contrôleur de code source

- Source contrôle est la pratique **du suivi et de la gestion des modifications du code**
- Les sources contrôles fournissent un **historique** de développement du code et aident à résoudre les **conflits** lors de la fusion de contributions provenant de plusieurs sources.
- Le contrôle des sources protège le code source des dégradations occasionnelles suite aux erreurs humaines.
- Les avantages incluent: la réutilisabilité, la traçabilité, portabilité, l'efficacité, la collaboration et l'apprentissage.



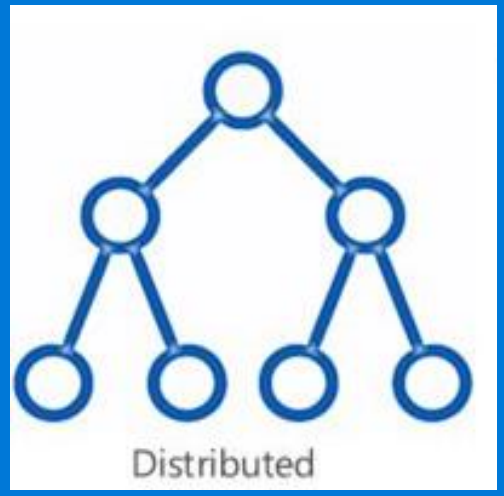
Contrôleur de code source

Bonnes pratiques pour le contrôle des sources :

- ✓ Faites de petits changements, avec des commit régulières.
- ✓ Ne pas commité des fichiers personnels.
- ✓ Mettre à jour souvent et juste avant de pousser pour éviter les conflits de fusion.
- ✓ Vérifiez votre changement de code avant de le pousser dans un référentiel, assurez-vous qu'il compile et que les tests réussissent.
- ✓ Portez une attention particulière à la validation des messages car ceux-ci vous indiqueront pourquoi un changement a été effectué.
- ✓ Lier les modifications de code aux éléments de travail (Work Item).

Contrôleur de code source

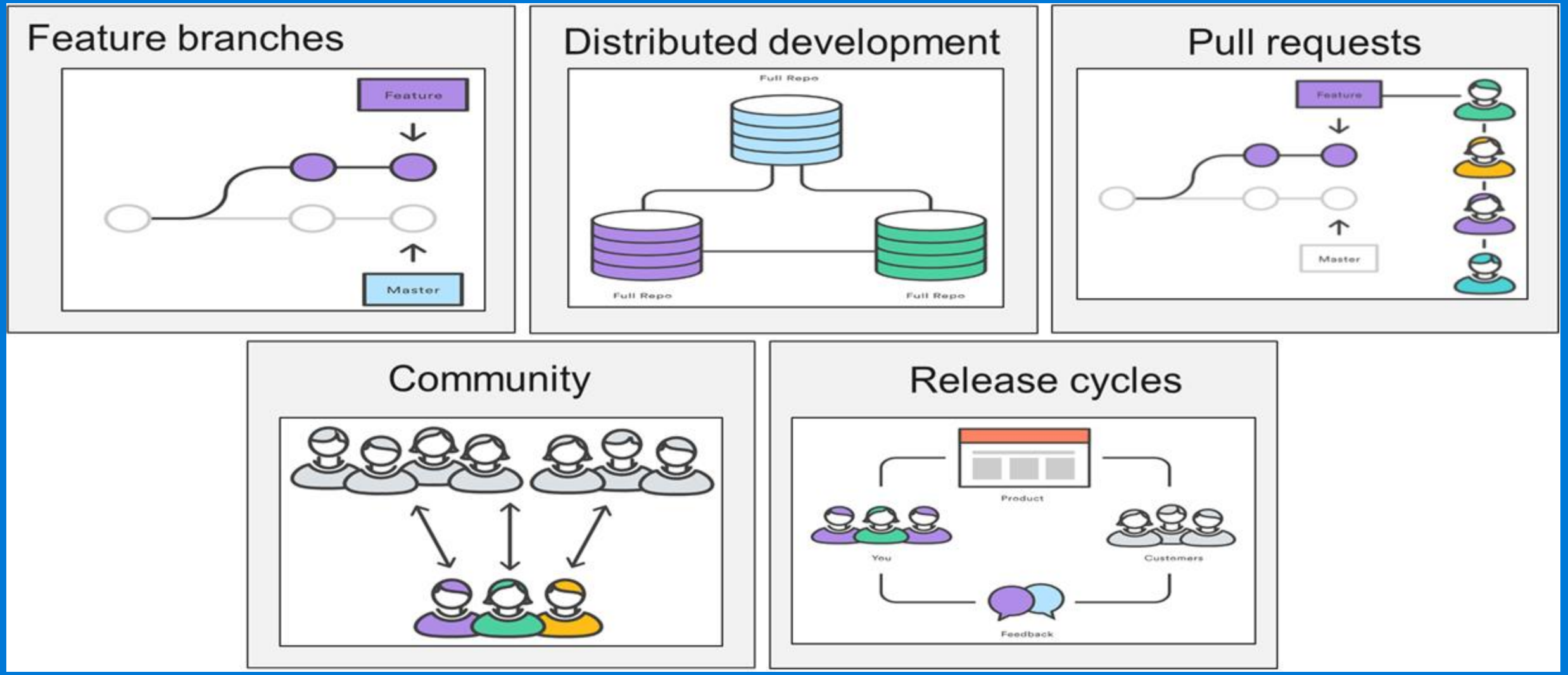
Contrôle de source distribuée :



Forces	Idéal pour
• Prise en charge multiplateforme (CrossPlatform) • Un modèle de révision de code convivial open source via des demandes de tirage (Pull) • Historique • Demande de validation « Pull Request » • Gestion des branches.	• Petite modification de code. • Évoluer grâce à l'open source. • Des équipes réparties. • Des équipes travaillant sur toutes les plateformes.

- Chaque développeur clone une copie d'un référentiel et à l'historique complet du projet.
- Les systèmes de contrôle de source distribuée courants sont Git et Azure DevOps Serveur

Contrôleur de code source



Mono vs Multi Repos

Un Repos est un endroit où l'historique de votre travail est stocké

Mono-repo - source code est dans un seul repository	• Déconseillé
Multiple-repo – chaque projet a son proper repository	• De meilleurs tests pour les développeurs • Complexité du code réduite • Révisions de code efficaces • Partage de composants communs • Refactoring facile

Version control workflow

Le contrôle de version a un flux de travail général que la plupart des développeurs utilisent pour écrire du code et le partager avec l'équipe

Ces étapes sont les suivantes:

1. Obtenez une copie locale du code s'ils n'en ont pas encore.
2. Apportez des modifications au code pour corriger les bogues ou ajouter de nouvelles fonctionnalités.
3. Une fois le code prêt, rendez-le disponible pour examen par votre équipe (Pull Request).
4. Une fois le code révisé, fusionnez-le dans la base de code partagée de l'équipe.

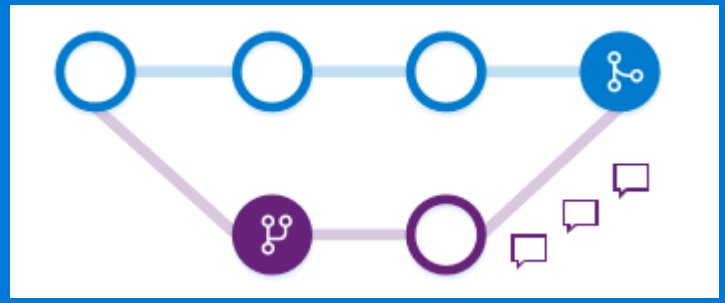
Pull Request

Pull Request est le processus collaboratif qui permet au reste de l'équipe de discuter des changements dans une branche et d'accepter de les fusionner une fois que tout le monde approuve.

Les Pull Request combinent la révision et la fusion de votre code en un seul processus collaboratif :

1. Après avoir corrigé un bogue ou une nouvelle fonctionnalité dans une branche,
2. Créez une nouvelle demande d'extraction,
3. Ajoutez les membres de l'équipe à la demande d'extraction afin qu'ils puissent examiner et voter sur vos modifications.

Utilisez les demandes d'extraction pour examiner les travaux en cours et obtenir des commentaires rapides sur les modifications.
Il n'y a aucun engagement à fusionner les modifications, car vous pouvez abandonner la demande d'extraction à tout moment.



Atelier

GIT – Azure DevOps