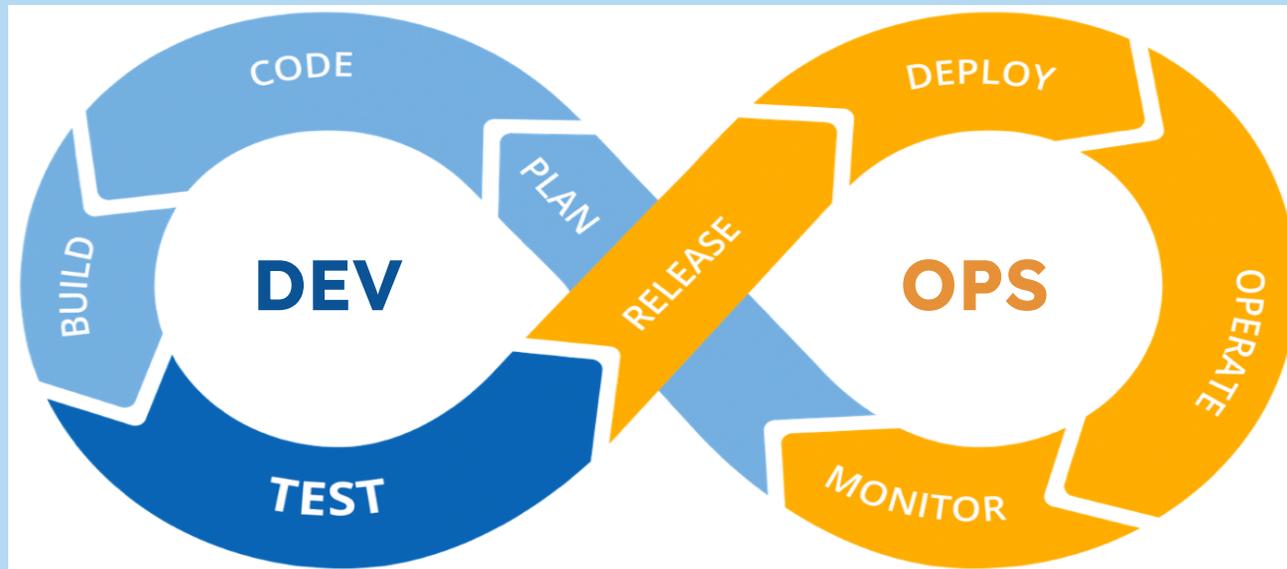


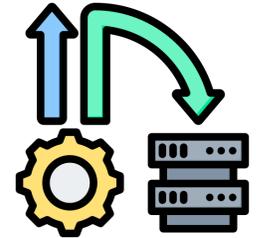
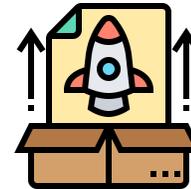
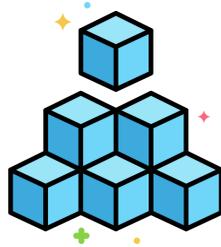
# L'USINE LOGICIELLE



# CYCLE DEVOPS

- Améliorer la communication entre les développeurs et l'exploitation afin de réduire le temps de mise sur le marché d'un produit
- Proposer des bonnes pratiques destinées à répondre au besoin croissant d'industrialisation et de normalisation du système d'information
- Sécuriser et stabiliser les mises en production
- Limiter les interventions humaines : sources principales d'erreur
- Se reposer sur des outils et construire **une usine logicielle**





MONITOR

PLAN

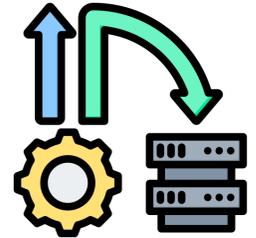
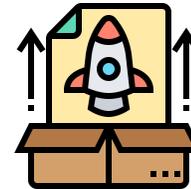
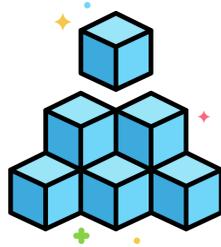
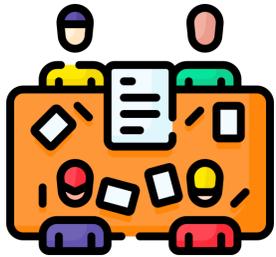
CODE

BUILD

TEST

RELEASE

DEPLOY  
OPERATE



MONITOR

PLAN

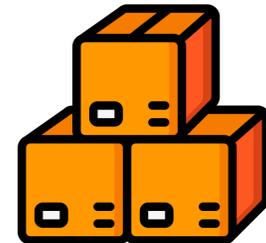
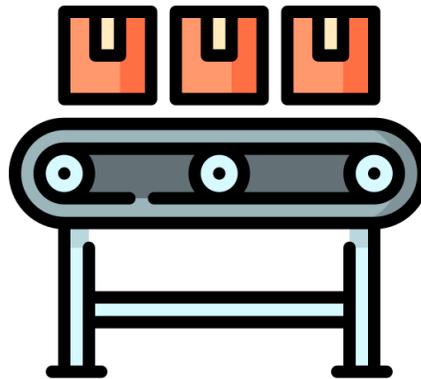
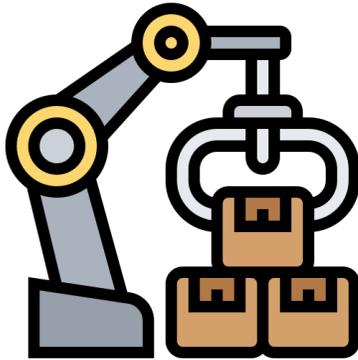
CODE

BUILD

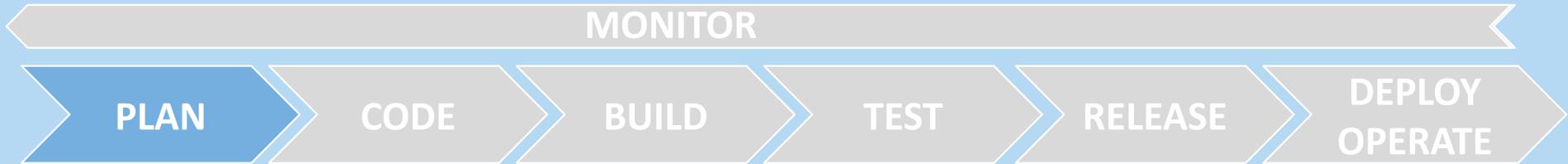
TEST

RELEASE

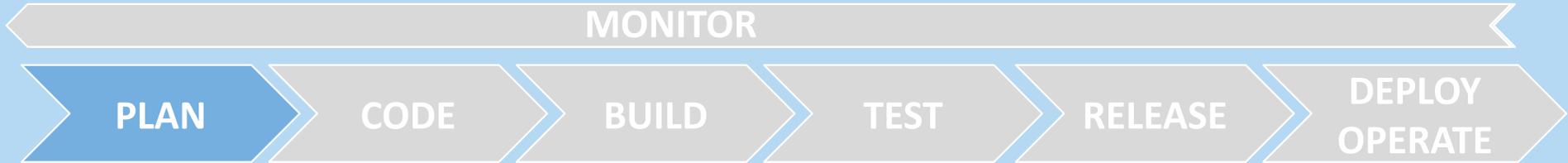
DEPLOY  
OPERATE



# PLAN : PLANIFIER LE TRAVAIL

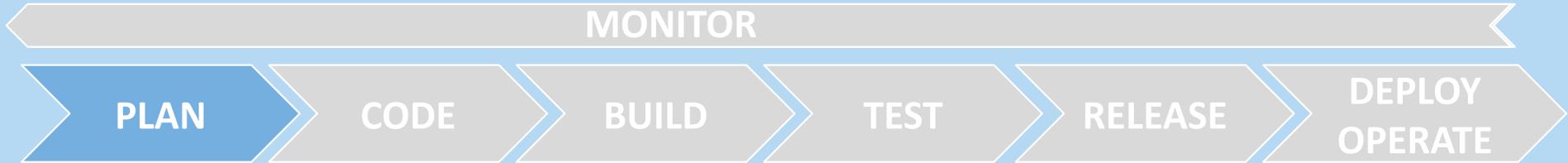


# PLAN : PLANIFIER LE TRAVAIL

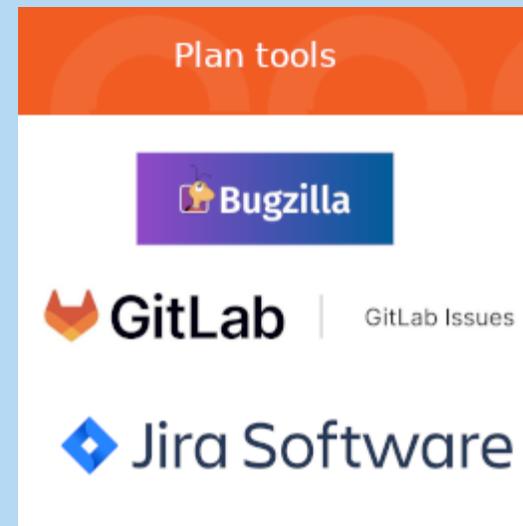


- Planifier le travail / gestion de projet
- Plan de production
- Planification des itérations
- Documentation
- Mise en place d'évolutions
- Gestion des anomalies

# PLAN : PLANIFIER LE TRAVAIL



- Planifier le travail / gestion de projet
- Plan de production
- Planification des itérations
- Documentation
- Mise en place d'évolutions
- Gestion des anomalies



# CODE : CRÉATION ET INTÉGRATION DU CODE DU PRODUIT

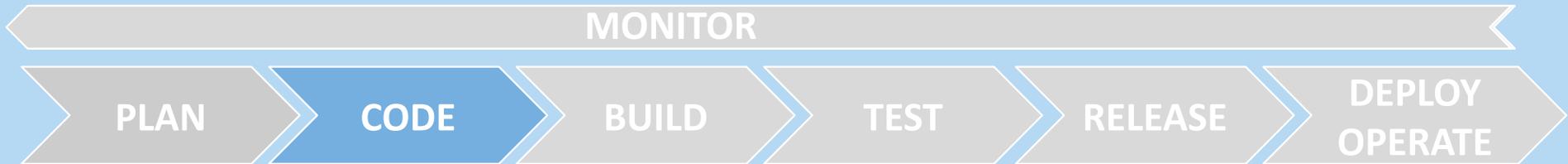


# CODE : CRÉATION ET INTÉGRATION DU CODE DU PRODUIT



- Créer, construire et produire le code
- Suivre l'évolution de ce code
- Gestion de configuration
- Gestion des branches, fusion
- Conventions sur le code

# CODE : CRÉATION ET INTÉGRATION DU CODE DU PRODUIT



- Créer, construire et produire le code
- Suivre l'évolution de ce code
- Gestion de configuration
- Gestion des branches, fusion
- Conventions sur le code



# BUILD : CONSTRUIRE SON PRODUIT



# BUILD : CONSTRUIRE SON PRODUIT

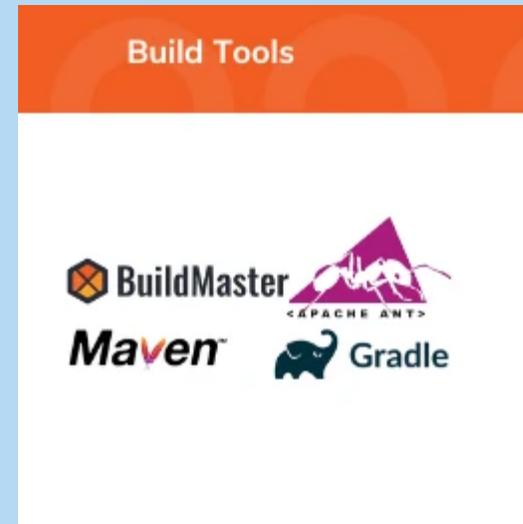


- Produire la forme "exécutable" du code
- Avoir une version "compilée"
- Produire des artefacts
- Construire automatiquement du code prêt à être déployer

# BUILD : CONSTRUIRE SON PRODUIT



- Produire la forme "exécutable" du code
- Avoir une version "compilée"
- Produire des artefacts
- Construire automatiquement du code prêt à être déployer



# TEST : TESTER LA VIABILITÉ DE SON PRODUIT



# TEST : TESTER LA VIABILITÉ DE SON PRODUIT

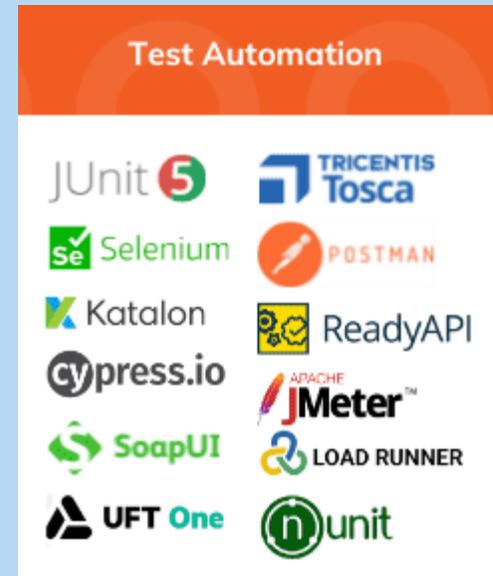


- Tests unitaires
- Tests d'intégration
- Tests fonctionnels
- Suite de tests automatique

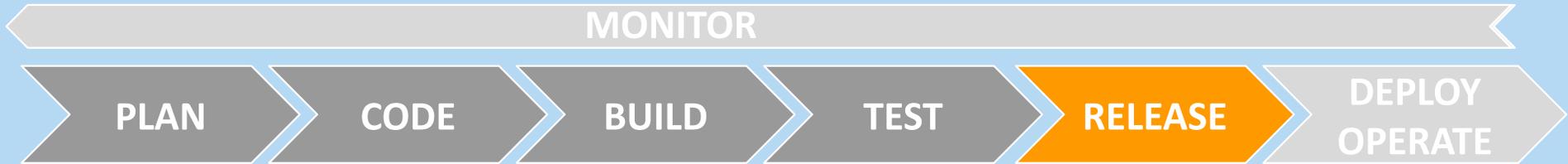
# TEST : TESTER LA VIABILITÉ DE SON PRODUIT



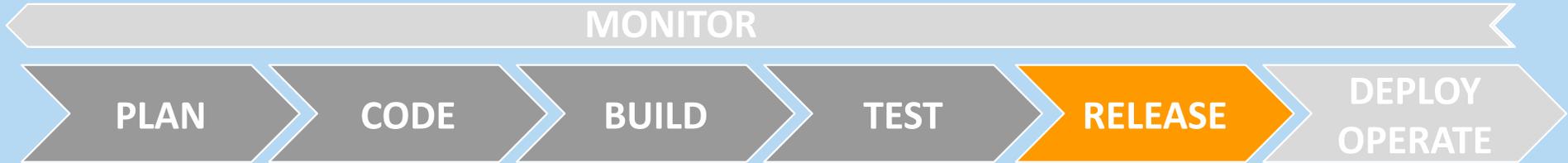
- Tests unitaires
- Tests d'intégration
- Tests fonctionnels
- Suite de tests automatique



# RELEASE: LIVRER LE CODE

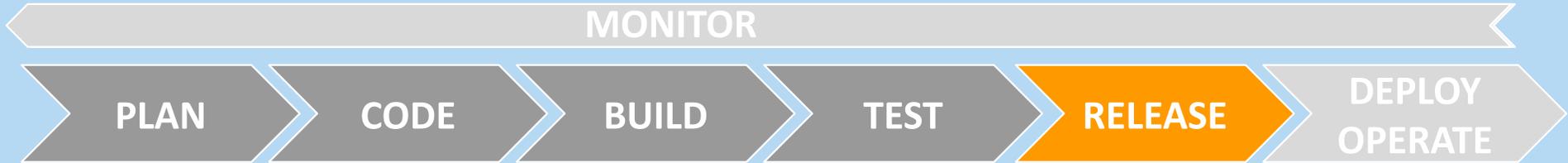


# RELEASE: LIVRER LE CODE

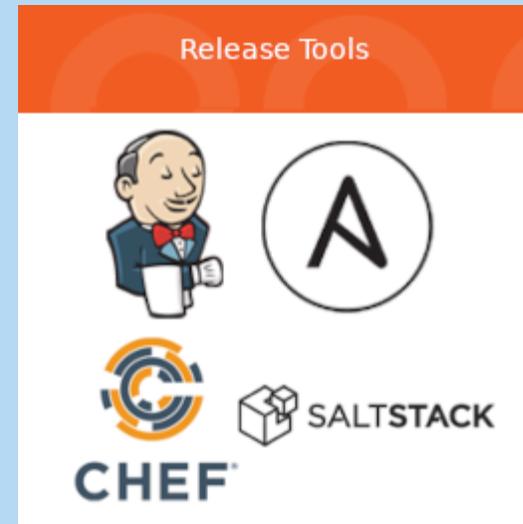


- Publier le code sous forme d'un livrable
- Publier/tagguer une version de logiciel
- Documenter en vue du déploiement
- Gérer les artefacts
- Phase intimement liée au déploiement

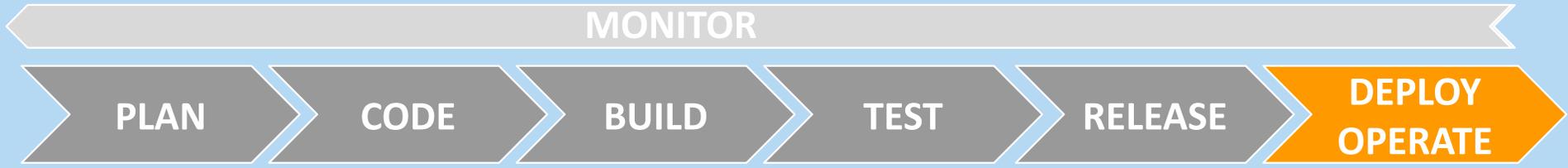
# RELEASE: LIVRER LE CODE



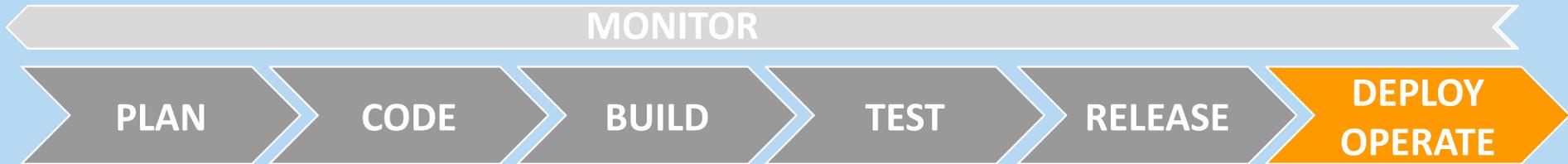
- Publier le code sous forme d'un livrable
- Publier/tagguer une version de logiciel
- Documenter en vue du déploiement
- Gérer les artefacts
- Phase intimement liée au déploiement



# DEPLOY / OPERATE :



# DEPLOY / OPERATE :



- Déploiement des artefacts dans l'environnement adéquat
- Configuration de l'environnement et de l'architecture
- Mise en place d'une stratégie de déploiement (Rollout, Blue-Green, ...)
- Mise à disposition du logiciel aux utilisateurs
- Etape fortement liée à l'infrastructure
- Assurer l'opération du logiciel, le suivi de son usage, mise en place du retour continu

# DEPLOY / OPERATE :



- Déploiement des artefacts dans l'environnement adéquat
- Configuration de l'environnement et de l'architecture
- Mise en place d'une stratégie de déploiement (Rollout, Blue-Green, ...)
- Mise à disposition du logiciel aux utilisateurs
- Etape fortèment liée à l'infrastructure
- Assurer l'opération du logiciel, le suivi de son usage, mise en place du retour continu



# MONITOR :



# MONITOR :

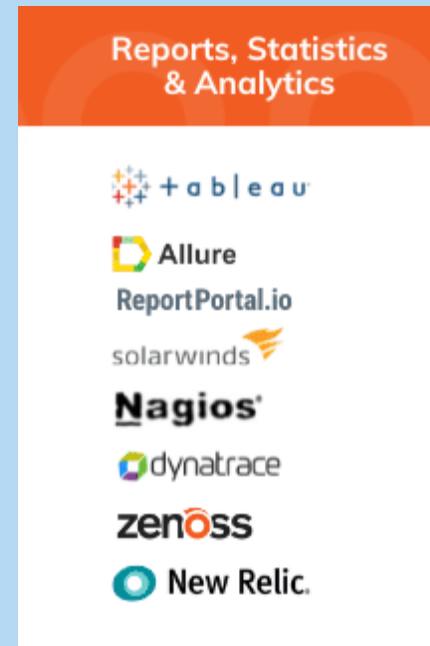


- Monitorer tout ce qui peut l'être : code, infrastructure
- Suivre les logs et les usages
- Etre dans une démarche d'amélioration continue

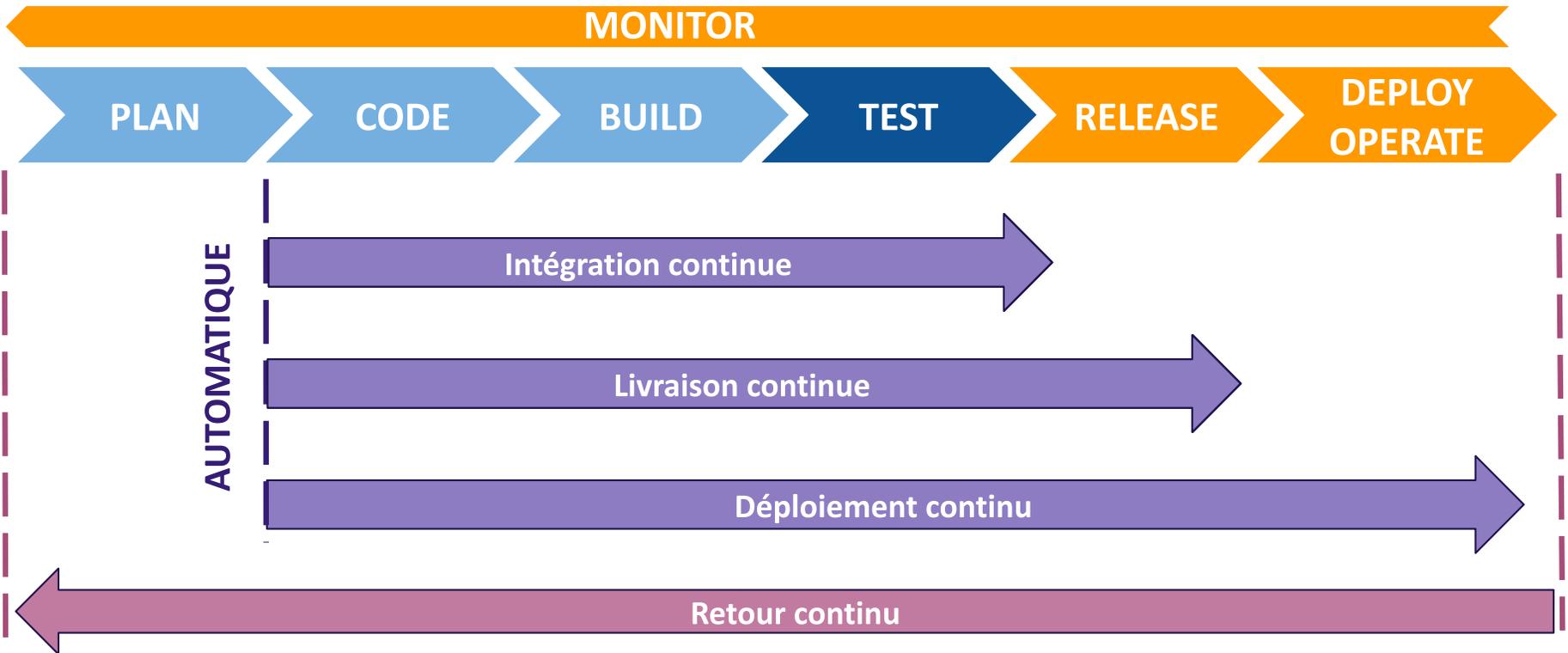
# MONITOR :



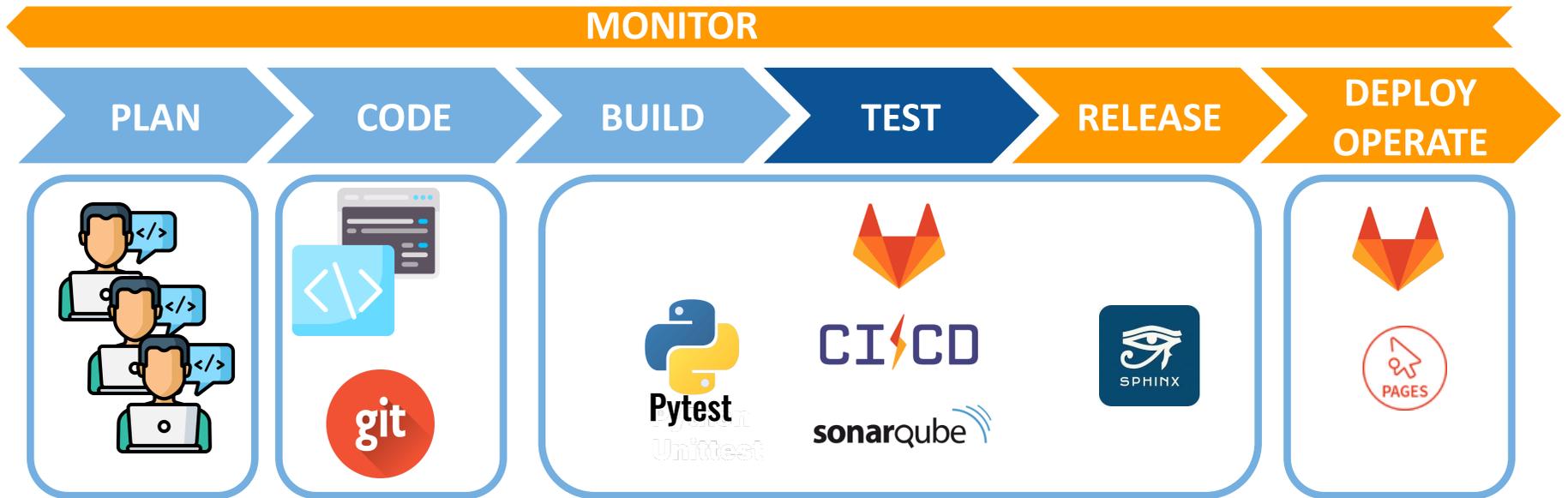
- Monitorer tout ce qui peut l'être : code, infrastructure
- Suivre les logs et les usages
- Etre dans une démarche d'amélioration continue



# CYCLE DEVOPS ET INTÉGRATION CONTINUE



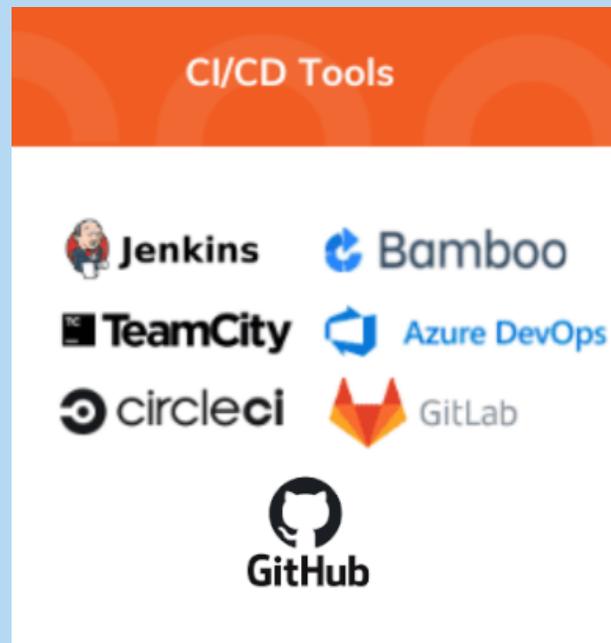
# CYCLE DEVOPS ET TP



# UNE MYRIADE D'OUTILS

<b>Application Lifecycle Mgmt.</b> JIRA, mingle, Trello, Visual Studio, Team Foundation Server, PivotalTracker, Basecamp, PHABRICATOR, asana	<b>SCM/VCS</b> git, SUBVERSION, mercurial, GitHub, Bitbucket, Gogs, GitLab, GitBucket	<b>Testing</b> Selenium, Jasmine, MOCHA, GAUNTLT, OWASP, ZAP, Gatling, KARMA, JUnit, FitNesse, Nunit, QUnit, Test NG, cucumber	<b>Deployment</b> Octopus Deploy, XL DEPLOY, RUNDECK, Capistrano, urban{code}, NOLIO, SSH, JUJU, ElasticBox, Spinnaker	<b>Cloud / IaaS / PaaS</b> amazon web services, heroku, Dokku, Flynn, Microsoft Azure, CLOUDFOUNDRY, Google Cloud Platform, rackspace, openstack, DEIS, appfog, OPENSHIFT, Engine Yard
<b>Communication &amp; ChatOps</b> slack, HipChat, #irc, flowdock, Microsoft Teams, RYVER, Mattermost, ROCKET.CHAT, COG, Nestor, LITA, HU-BOT	<b>CI</b> wercker, snap, TeamCity, Jenkins, Bamboo, drone.io, circleci, Travis CI, go, CODESHIP	<b>Testing</b> Galen Framework, LOAD IMPACT, JMeter, BlazeMeter, pytest, SERVERSPEC, Browsersync, Pa1ly, specflow, Newman, Unit.net	<b>Config Mgmt./Provisioning</b> puppet labs, CHEF, ANSIBLE, SALTSTACK, CFEngine, PowerShell DSC, VAGRANT, TERRAFORM	<b>Orchestration &amp; Scheduling</b> MESOSPHERE, MARATHON, SWARM, kubernetes, Nomad, RANCHER, MESOS
<b>Knowledge Sharing</b> github:pages, jekyll, Confluence, HUGO, Mark down, Read the Docs, apiblueprint, RAML, FLARUM, OPEN API INITIATIVE, graphviz, Discourse, reddit	<b>Build</b> sbt, Gradle, GRUNT, Maven, gulp, docker, Nant, Apache ANT, PACKER, MSBuild, Leiningen, Rake	<b>Testing</b> Galen Framework, LOAD IMPACT, JMeter, BlazeMeter, pytest, SERVERSPEC, Browsersync, Pa1ly, specflow, Newman, Unit.net	<b>Artefact Management</b> QUAY, DOCKERHUB, DOCKER REGISTRY, Bower, Jfrog Artifactory, python Package Index, nuget, archiva, npm, Nexus	<b>BI / Monitoring / Logging</b> logstash, elasticsearch, splunk, Vector, kibana, DATADOG, ZABBIX, APPDYNAMICS, GRIPK, ZIPKIN, Google Analytics, x-pack, SENTRY, Prometheus, New Relic, VIZCERAL, dynatrace, PINPOINT, Runscope, sensu, Atlas, RIEMANN, API metrics, Grafana, RAYGUN, STATSD, graphits, Airbrake.io, Rollbar, pagerduty, 100C, beats, OpsGenie, Keen IO

# DES OUTILS QUI COUVRENT DE NOMBREUX ASPECTS



## LES OUTILS, C'EST PAS SUFFISANT

- le DevOps / L'usine logicielle ce n'est pas seulement des outils
- C'est une démarche à adopter
- C'est une démarche à construire avec des objectifs et des moyens
- C'est une démarche itérative et évolutive

# MISE EN PLACE D'UNE USINE LOGICIELLE

# MISE EN PLACE D'UNE USINE LOGICIELLE

- Il faut prévoir la démarche au coeur du projet, ne pas la gérer à la fin
- Par conséquent, prévoir du temps pour cette thématique
- Ne pas vouloir tout faire d'un coup
- S'appuyer sur les outils et les bonnes pratiques existants
- Ne pas croire que les outils vont tout résoudre pour vous
- Ne pas sous-estimer la phase de mise en place

## LA MISE EN PLACE PAR ÉTAPES

1. Définir un référentiel de bonnes pratiques de code
2. Adopter une gestion centralisée du Code Source
3. Mettre en place des tests automatisés
4. Documenter l'architecture du code et le code
5. Déployer la partie CI/CD
6. Monitorer tout élément automatisé

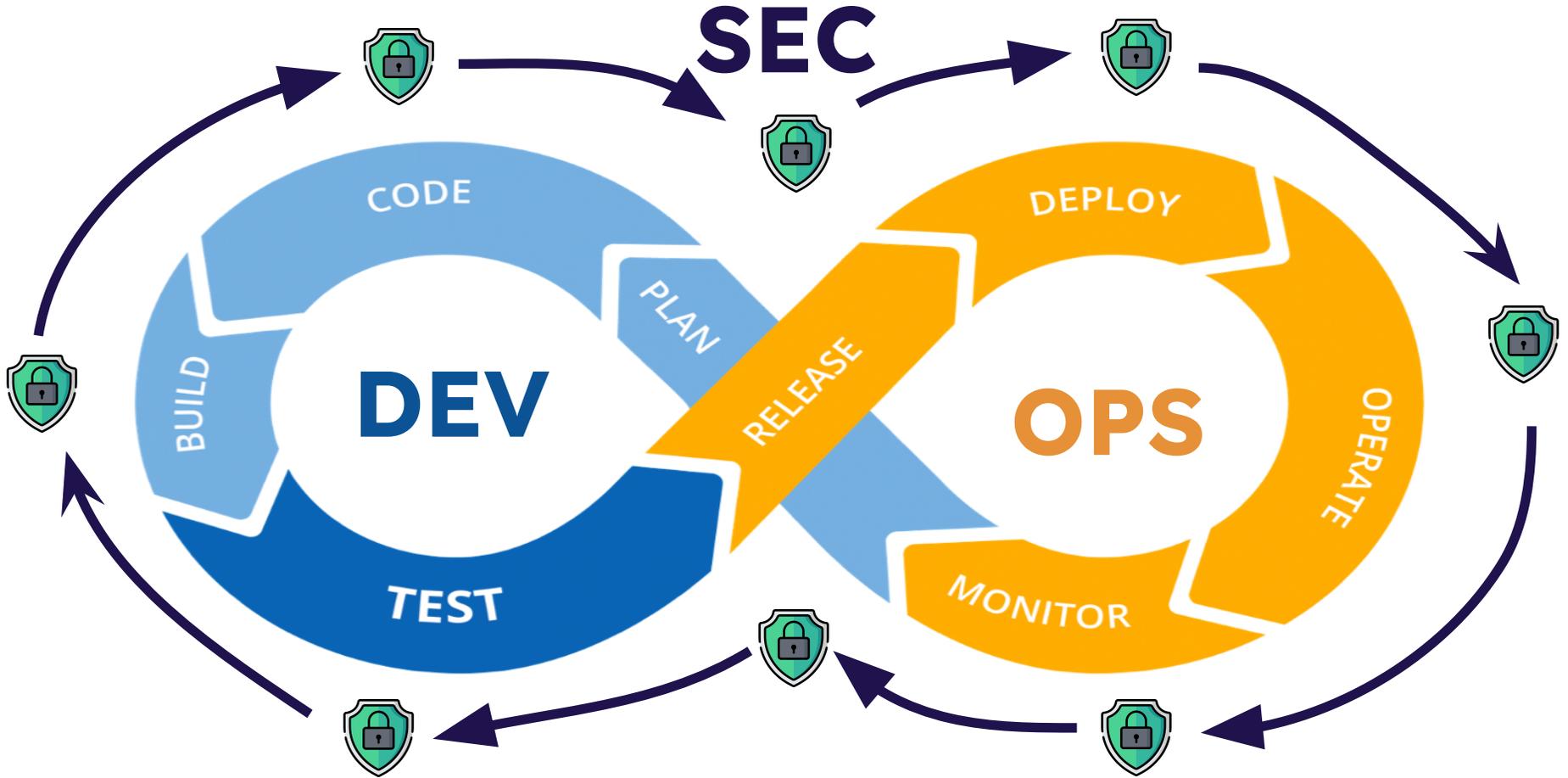
## DÉFINIR UN RÉFÉRENTIEL DE BONNES PRATIQUES

- Sensibiliser les équipes à la Qualité Logicielle
- Échanger sur les bonnes pratiques
- Former les développeurs débutants
- Mettre en place des règles métiers, des conventions de nommage

## POUR LA MISE EN PLACE DE LA CI/CD

- Ne cherchez pas à créer votre pipeline d'un seul coup
- Automatisez déjà vos tests / vos livraisons localement
- Examinez les logs de votre CI/CD même quand ça marche
- Réévaluez vos priorités et objectifs au cours du projet

## POUR ALLER PLUS LOIN, LE CYCLE DEVSECOPS



# LE CYCLE DEVSECOPS

