UNIVERSITÉ DE BATNA 2 MOSTEPHA BEN BOULAID

FACULTÉ DES MATHÉMATIQUES ET D'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

CHAPITRE 02

Cycle de vie de développement logiciel sécurisé
(SSDLC: Secure Software Development LifeCycle)

Dr. Ali BEDDIAF

PLAN

I. Introduction
II. Sécurité applicative
III. Menaces applicatives
IV. Cycle de vie: classique
V. Cycle de vie: méthodes agiles
VI. Cycle de vie: DevOps
VII. Cycle de vie sécurisé

INTRODUCTION

Motivation

- Toute application après son déploiement est menacée par tout type d'attaque provenant des utilisateurs potentiels et/ou des tierces personnes

- Ces risques menacent le bon fonctionnement de l'application ainsi que la confidentialité du métier

- C'est pourquoi il y a toujours le besoin de sécuriser l'application; « sécurité applicative »

- Elle doit intervenir tout au long du cycle de développement pour une meilleure efficacité et un coût minimal

SÉCURITÉ APPLICATIVE

Définition

- La sécurité des applications est un processus effectué pour appliquer des contrôles et des mesures aux applications d'une organisation afin de gérer le risque de leur utilisation

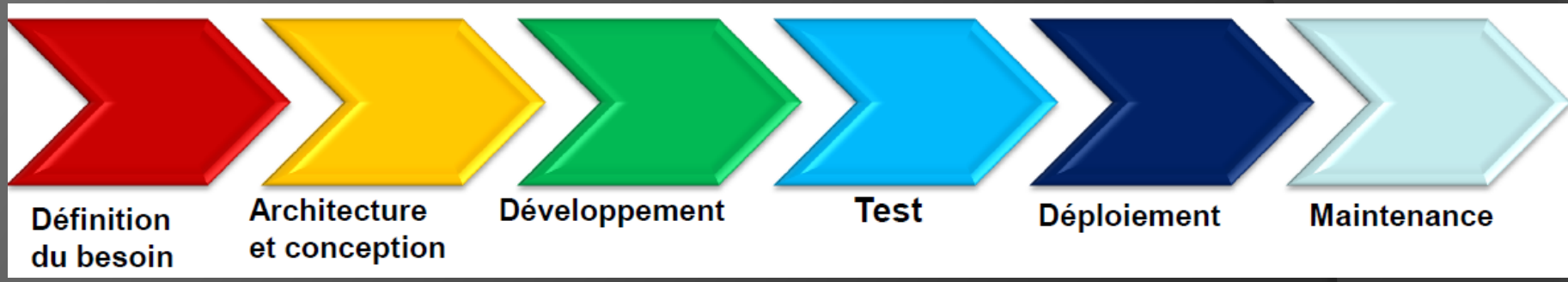
- Les contrôles et les mesures peuvent être appliquées à l' application elle-même (ses processus, les composants, les logiciels et résultats), à ses données (données de configuration, les données de l'utilisateur, les données de l'organisation), et à toutes les technologies, les processus et acteurs impliqués dans le cycle de vie de l'application

MENACES APPLICATIVES

Selon l'organisation à but non lucratif OWASP (Open Web Application Security Project) a listé les *dix* menaces les plus fameux que cours une application, à savoir:

- Injection
- Violation de gestion d'authentification et de session
- Cross Site Scripting (XSS)
- Références directes non sécurisées à un objet
- Mauvaise configuration de sécurité
- Exposition de données sensibles
- Manque de contrôle d'accès au niveau fonctionnel
- Falsification de requêtes intenses (CSRF)
- Utilisation de composants avec vulnérabilités connues
- Redirections et renvois non validés

CYCLE DE VIE: CLASSIQUE

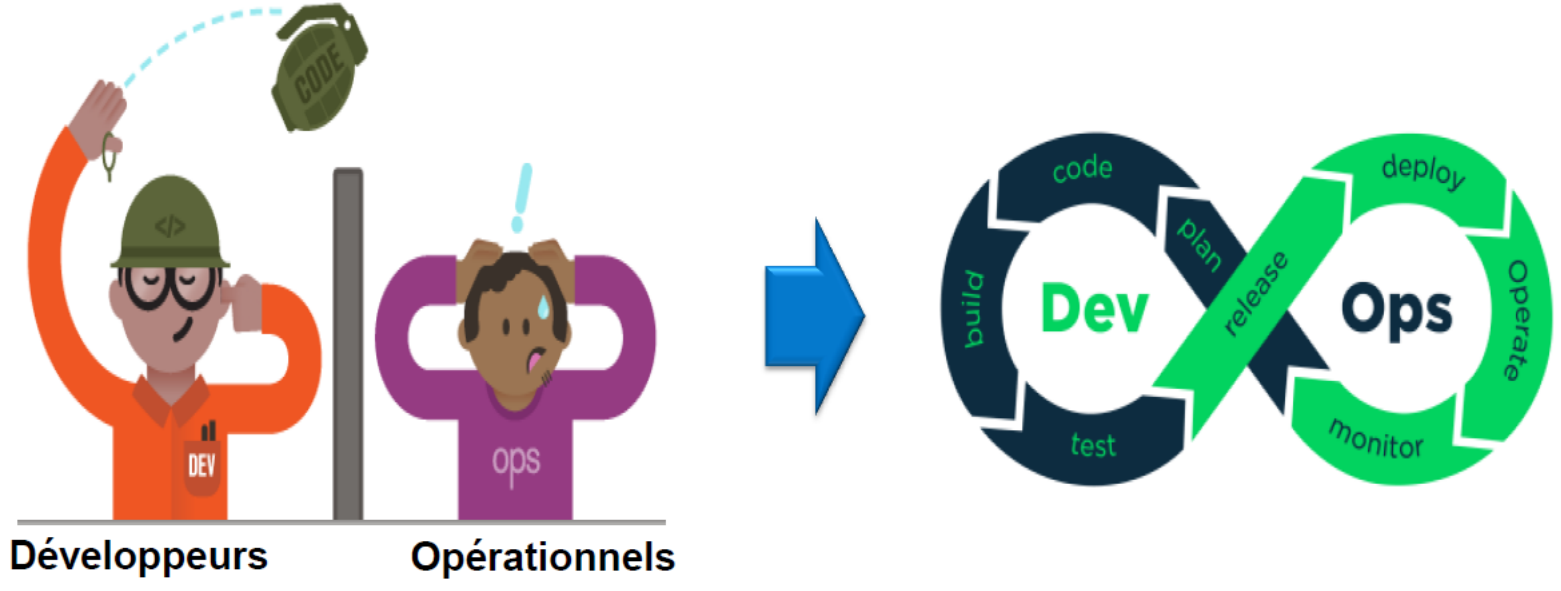


- C’est un enchainement des phases de développement de logiciel
- Une fois le logiciel livré, le client vu la variabilité de ses besoins qui sont des fois imprévisibles, il peut remettre en cause plusieurs phases ce qui rend la révision du logiciel très couteuse

CYCLE DE VIE: MÉTHODES AGILES

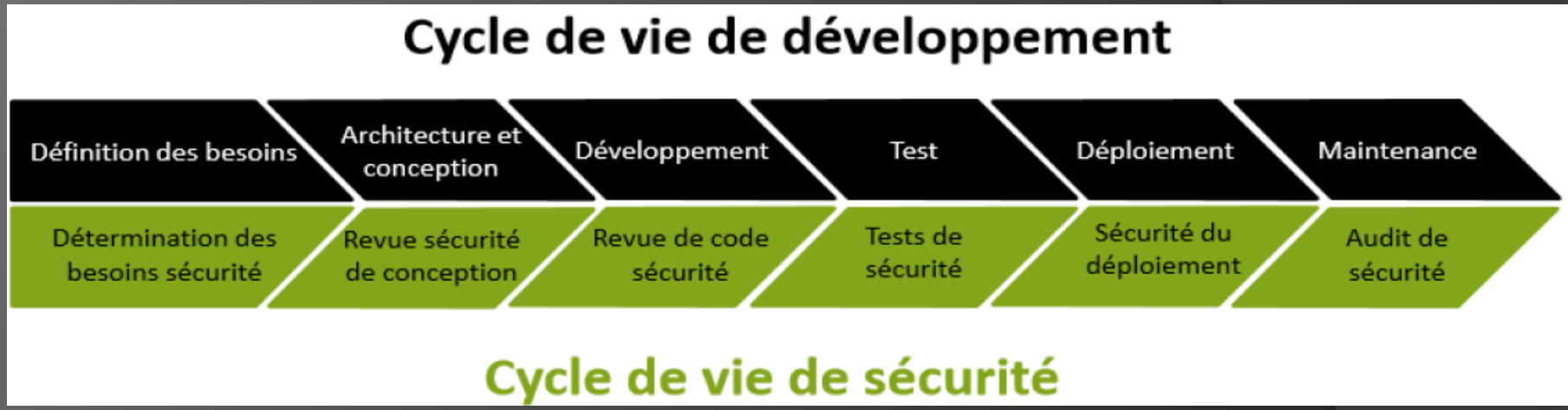
- Elles reposent sur un cycle de développement itératif, incrémental et adaptatif.
- Elles impliquent au maximum le client
- Une livraison précoce du logiciel au client
- Elles se basent sur la collaboration entre des équipes auto-organisées et pluridisciplinaires et leurs clients
- Exemples: Scrum et eXtreme Programming

CYCLE DE VIE: DEVOPS



- DevOps est une méthode de développement logiciel qui met en avant la communication, la collaboration, l'intégration, l'automatisation et les métriques entre développeurs et les administrateur système de l'infrastructure .

CYCLE DE VIE SÉCURISÉ



- Le SSDLC: Secure Software Development LifeCycle, est un processus continu contenant différents axes et étapes permettant d'assurer et augmenter le niveau de sécurité d'une application.
- Dans le SSDLC, les tests de sécurité sont effectués tout au long du processus de développement.

CYCLE DE VIE SÉCURISÉ

Frameworks

- Microsoft SDL: Security Development LifeCycle
- Cigital Touchpoints
- BSIMM
- OWASP OpenSAMM

CYCLE DE VIE SÉCURISÉ

- Connaître les différents types de vulnérabilités et les contremesures associées
- Comprendre l'enjeux de la sécurité
- Connaître les bonnes pratiques de sécurité

CYCLE DE VIE SÉCURISÉ

- L'application traite-elle de données sensibles (militaire, médicale, etc.) ?
- L'application est-elle exposée publiquement ?
- Exigences DICT
  - Disponibilité
  - Intégrité
  - Confidentialité
  - Traçabilité

CYCLE DE VIE SÉCURISÉ

- S'assurer que l'architecture proposée ne comporte pas de problème de sécurité:
  - Protocoles sécurisés
  - Méthode d'authentification/autorisation
  - Mécanisme de gestion des logs
  - Séparation des composants
  - Flux nécessaires

CYCLE DE VIE SÉCURISÉ

- SAST Analyse statique
  - VeraCode, Checkmarx, WhiteHatSecurity,IBM AppScan
- Revue de code
  - OWASP Code Review Guide
- Analyse des dépendences
  - Nexus, DependencyCheck, Jenkins

CYCLE DE VIE SÉCURISÉ

- DAST Analyse dynamique
  - VeraCode, Arachni, IBM AppScan , miniFuzz, AppVerifier, BinScop

CYCLE DE VIE SÉCURISÉ

- Eviter les incidents de sécurité liés à la configuration
  - Déploiement automatisé
  - Checklist de vérification