

# Systèmes d'information avancés

Sergey Kirgizov

ESIREM

# Structure du module

8 CM, 4 séances TD, 7 séances TP

Le lien :

[https://kirgizov.link/teaching/esirem/  
advanced-information-systems/](https://kirgizov.link/teaching/esirem/advanced-information-systems/)

Les notes :

- ▶ un contrôle intermédiaire
- ▶ un contrôle terminal
- ▶ un projet à faire pendant les TPs, autour de la compréhension de la blockchain

Concentrons nous sur l'étude des structures des données, des algorithmes, des méthodes et des logiciels qui permet le développement et fonctionnement de systèmes d'information.

# Cycle de vie d'un système d'information

## 1. Compréhension

# Cycle de vie d'un système d'information

1. Compréhension
2. Modélisation

# Cycle de vie d'un système d'information

1. Compréhension
2. Modélisation
3. Développement

# Cycle de vie d'un système d'information

1. Compréhension
2. Modélisation
3. Développement
4. Gestion

# Cycle de vie d'un système d'information

1. Compréhension
2. Modélisation
3. Développement
4. Gestion
5. Recyclage (?)

# Cycle de vie d'un système d'information

1. Compréhension
2. Modélisation
3. Développement
4. Gestion
5. Recyclage (?)

**en pratique, bien sûr, ce n'est pas tout à fait vrai**

# Biologie ?

- ▶ Systèmes d'information actuels sont très complexes et en évolution rapide.

# Biologie ?

- ▶ Systèmes d'information actuels sont très complexes et en évolution rapide.
- ▶ Comme des organismes biologiques.

# Biologie ?

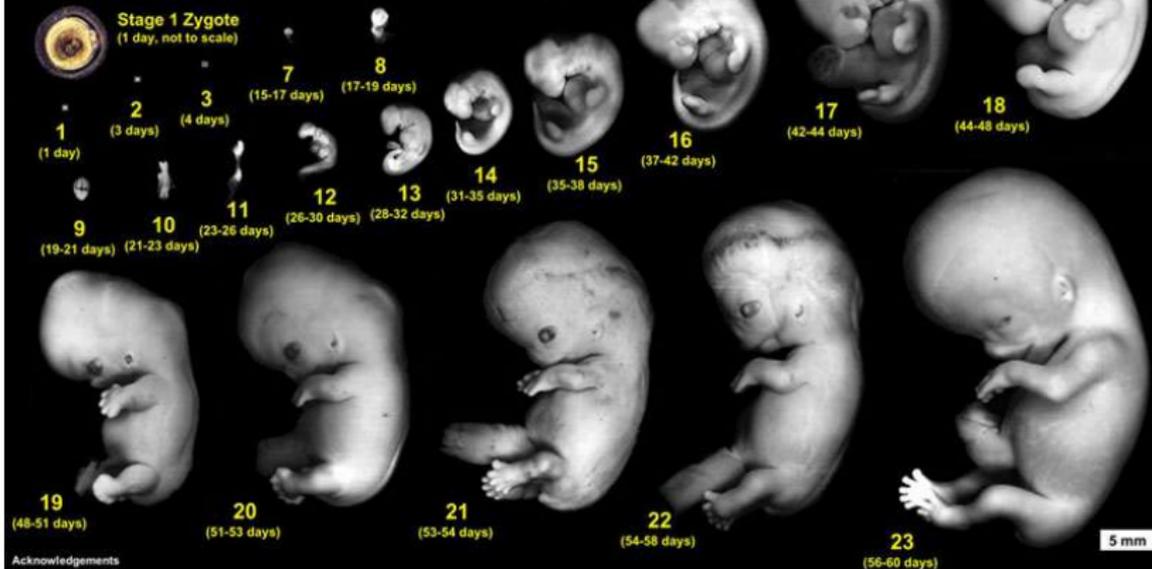
- ▶ Systèmes d'information actuels sont très complexes et en évolution rapide.
- ▶ Comme des organismes biologiques.
- ▶ Trop complexes.

# Biologie ?

- ▶ Systèmes d'information actuels sont très complexes et en évolution rapide.
- ▶ Comme des organismes biologiques.
- ▶ Trop complexes.
- ▶ Parfois, on doit écrire programme B qui corrigent le résultat du programme A, car c'est plus facile de faire ça que comprendre et corriger le programme A.

# Carnegie Stages of Human Development

Dr Mark Hill, Cell Biology Lab, School of Medical Sciences (Anatomy), UNSW



#### Acknowledgements

Special thanks to Dr S. J. DiMarzo and Prof. Kohel Shiota for allowing reproduction of their research images and material from the Kyoto Collection and Ms B. Hill for image preparation.

© M.A. Hill, 2004

1. Un code développe une queue.

1. Un code développe une queue.
2. L'autre la supprime.

Nouvelles technologies

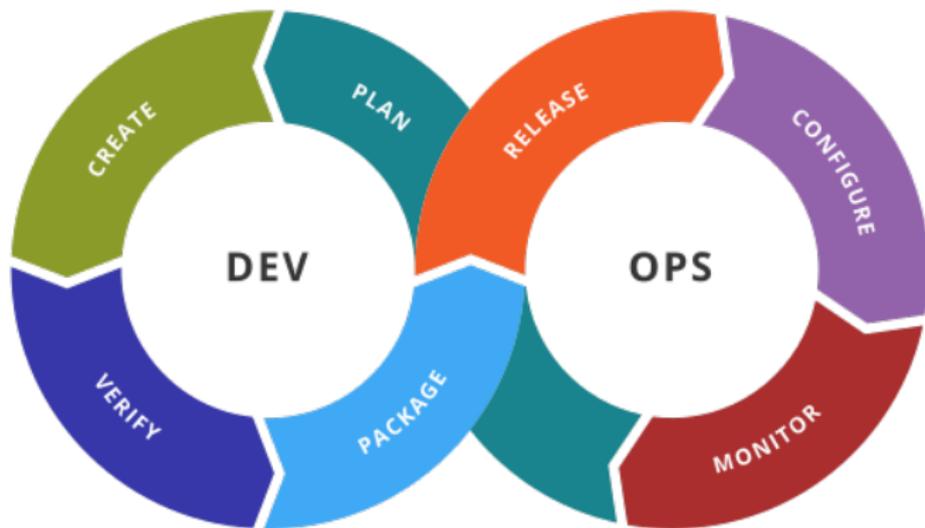
# Nouvelles technologies

- ▶ Massive Open Source Dev (npm, etc)
- ▶ Clouds
- ▶ Micro-services
- ▶ Conteneurisation : Docker, Kubernetes, etc



- ▶ Continuous integration, continuous delivery, continuous deployment (Circle, Travis, Jenkins)

Problèmes de sécurité et de confidentialité, de la vie privée  
GDPR



Changement culturel :

- ▶ Automatisation du processus de dev. et livraison.
- ▶ Écrire un code qui gère un autre code.
- ▶ Artisans vs. machines.

- ▶ chiffrement, signatures, autorité de certification, réseaux de confiance, PGP

- ▶ Introduction
- ▶ Tables de hachage
- ▶ Fonction de hachage cryptographique, intégrité, sécurité
- ▶ Systèmes de gestion de versions
- ▶ Systèmes de gestion de revue de code. Intégration continue. Livraison continue. Déploiement continu
- ▶ Nuages. Conteneurs. Micro-services
- ▶ Sécurité. Chiffrement. RSA. PGP.
  
- ▶ TD 1-2-3 — Fonctions de hachage
- ▶ TD 5 — Exemple de micro-service avec Python, Flask et HTTP
- ▶ TPs : un projet à faire, autour de la compréhension de la blockchain

Questions ?