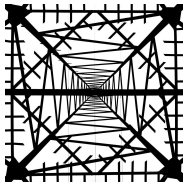
Les dépendances

AFUP Montpellier - Meetup du 4 octobre 2018 à Kaliop

Qui je suis



Julien Vinber

- Développeur depuis +15 ans
- Coordinateur AFUP Montpellier
- Développeur pour 2S2I en mission pour La Poste

Mail : julien@vinber.fr
Twitter : @julienVinber
LinkedIn : julienVinber

Un petit test

Un petit test

Record max :
Apnée statique avec **inhalation** d'oxygène
24 min 03 s

https://fr.wikipedia.org/wiki/Apn%C3%A9e_(sport)

100%

Vous êtes tous dépendant.

100%

Votre code, produit, logiciel, site… **EST DÉPENDANT**

Petite définition.

État, situation d’un sujet, qui n'a pas son autonomie par rapport à un autre.

En informatique, cela se traduit par le fait que notre code peut ne plus fonctionner sans que nous en soyons directement les responsables.

La dépendance est un risque.

Le rôle du développeur et de le minimiser

Comprendre et anticiper

Facteur externe (hors dev)

Dépendance à l'infrastructure.

- Ce n’est pas notre problème en soi.
- Mais cela va finir par arriver.

Solution :

- Plan de reprise

Dépendance au service.

On ne peut rien faire pour les autres. Et en particulier on ne peut gérer leur indisponibilité.

Dépendance au service.

Pas de solution unique :

- Il faut donc évaluer :
  - Les risques de pannes
  - Les conséquences d'une panne

Solution possible :

- Ignorer l'information.
- Avoir nos caches pour consommer une donnée non fraiche.
- Prévoir un plan de reprise pour des actions bloquantes.

Facteur externe (dev)

Dépendance aux langages.

Oui le langage que nous utilisons et aussi un risque.

- Bug
- Faille de sécurité
- Adaptation à une plateforme.

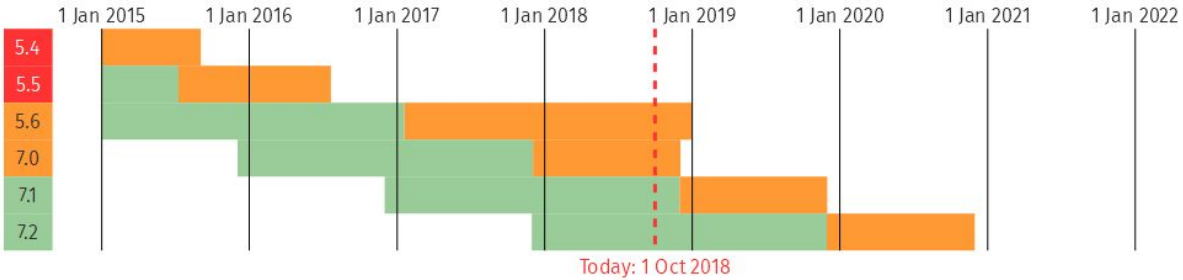
Dépendance aux langages.

Solution :

- Faire attention au langage exotique
- Faire attention aux versions et leur maintenance.

Dépendance aux langages.

Petit rappel :



Key

Active support	A release that is being actively supported. Reported bugs and security issues are fixed and regular point releases are made.
Security fixes only	A release that is supported for critical security issues only. Releases are only made on an as-needed basis.
End of life	A release that is no longer supported. Users of this release should upgrade as soon as possible, as they may be exposed to unpatched security vulnerabilities.

http://php.net/supported-versions.php

Dépendance aux framework

Nous avons les mêmes risques qu'avec le langage :

- Bug
- Faille de sécurité
- Adaptation à une plateforme.

Dépendance aux framework

Solution :

- Faire attention aux Framework exotique.
- Faire attention aux versions et leur maintenance.
- Penser à une évolution constante plutôt que de grosse migration que l'on ne fait jamais.

Dépendance aux framework

Quelques exemples :

- Symfony 2.* qui entre en security fixes à la fin du mois et fin de vie l'année prochaine
- Laravel : plus de support pour les versions < 5.5
- ...

Dépendance aux librairies

Toujours la même chose.

Mais en pire.

Dépendance aux librairies

Une petite histoire :

Azer Koculu publie un projet JavaScript baptisé Kik

Kik est un nom déjà utiliser.

Il s'ensuit des histoires juridiques

=> Mars 2016 le développeur retire tous ces projets de npm

Dépendance aux librairies

Entre autres le projet LeftPad :

- 11 lignes de code

Dépendance aux librairies

Entre autres le projet LeftPad :

- 11 lignes de code
- 2,5 millions de téléchargements lors du dernier mois.
- Des milliers de projets bloqués, directement ou à cause de dépendance de dépendance.

https://www.zdnet.fr/actualites/leftpad-un-module-vous-manque-et-tout-est-depeuple-39834650.htm

Dépendance aux librairies

Conclusion :

Posez-vous la question, le risque en vaut-il la chandelle ?

Dépendance aux librairies

Solution :

Se poser les questions :

- Qui à écrit cette librairie ?
- Puis-je m'y fier si je ne suis pas à 100 % dans le cas nominal ?
- Puis-je la réécrire aussi bien que la librairie et en combien de temps ?
- Qui vas en assurer le support et en combien de temps ?

Notre produit

Dépendance au fonctionnel

Notre client connaît son métier.

Dépendance au fonctionnel

Mais il n’est pas pour autant informaticien.

Son métier change.

Dépendance au fonctionnel

Risque :

- La transcription du métier en demande informatique.
- La transcription de la demande en solution.
- Changement
  - Renommage
  - Changement fonctionne
  - ...

Dépendance au fonctionnel

Solution :

Il n'y a pas de solution miracle :

- Un travail de conception, conseil et validation avec le client.
- Une bonne définition des concepts mis en jour.
- Avoir toujours en tête que cela va changer.

Dépendance aux modèles

Une constante avec informatique :

Le changement.

Dépendance aux modèles

Quand on crée notre modèle, ce dernier correspond à une vérité de l'instant.

Mais cette vérité va changer.

Faire changer un modèle coûte cher. Donc on s'adapte.

=> Jusqu'à ce que cela devienne ingérable.

=> Jusqu'au blocage

Dépendance aux modèles

Solution :

- Tuer le dogme de la Sainte Base De Données détenant la vérité universelle.
- Avoir une vraie réflexion dès le début sur comment évoluer.
- Ne pas forcément se contenter d'un seul modèle.
- Dé corréler code et modèle.

Dépendance aux modèles

Exemple sur la corrélation code/modèles

```
$user->getStatus() === Status::ACTIF
```

…

```
function isActif() { return $this->getStatus() === Status::ACTIF }
```

```
$user->isActif()
```

Dépendance aux modèles

Bonus :

L'Event Sourcing

~~Etat~~ => Événement

Notre code

Le modèle Objet

L'Objet à pour but de réunir dans un seul endroit tout ce qui concerne un concept afin de ne pas mélanger.

Mais un programme va exister par les interactions entre objet.

=> Donc dès dépendance entre eux.

Le modèle Objet

Attention, sans Objet cela peut être encore pire…

Le modèle Objet

Solution :

Toujours penser que notre façon de faire peut être améliorée.

Aller voir :

- Design pattern
- Se renseigner sur les bonnes pratique.
- Chercher à en comprendre le sens.
- Être curieux et voir comment c’est fait ailleurs

Injection de dépendance

Un des Design pattern que l’on rencontre souvent.

Mais cela sert à quoi ?

L'**injection de dépendances** (*dependency injection* en anglais) est un mécanisme qui permet d'implémenter le principe de l'inversion de contrôle.

Il consiste à créer dynamiquement (injecter) les dépendances entre les différents objets en s'appuyant sur une description (fichier de configuration ou métadonnées) ou de manière programmatique. Ainsi les dépendances entre composants logiciels ne sont plus exprimées dans le code de manière statique mais déterminées dynamiquement à l'exécution.

Injection de dépendance

```
function luiParler(User $destinataire, string $message) {

	$mail = new Mail();

	$mail->setExpediteur(...);

	$mail->setServeur(...);

	...

	$mail->to($destinataire);

	$mail->message($message);

	$mail->send();

}
```

Injection de dépendance

```
function luiParler(Mail $mail, User $destinataire, string $message) {

	$mail->to($destinataire);

	$mail->message($message);

	$mail->send();

}
```

Injection de dépendance

```
function luiParler(MailInterface $mail, User $destinataire, string $message) {

	$mail->to($destinataire);

	$mail->message($message);

	$mail->send();

}
```

Injection de dépendance

```
function luiParler(BusInterface $bus, User $destinataire, string $message) {
	$bus->to($destinataire);
	$bus->message($message);
	$bus->send();
}
```

Injection de dépendance

```
function __contructor(BusInterface $bus) {
	$this->bus = $bus;
}
function luiParler(User $destinataire, string $message) {
	$this->bus->to($destinataire);
	$this->bus->message($message);
	$this->bus->send();
}
```

Injection de dépendance

```
Class GestionDependance

...

function __contructor(BusInterface $bus) {

       $this->bus = $bus;

}
```

Notre code

Oui tout dans notre façon même d'écrire notre code peut apporter des dépendances.

Exemple :

$maVariable == 0

$maVariable === 0

Si

$maVariable = false

$maVariable = '0'

$maVariable = ''

Le savoir

Dépendance au sachant

Quand on fait, nous acquérons un savoir.

- Savoir technique
- Savoir fonctionnel
- Savoir sur le pourquoi des solutions.
- ...

Dépendance au sachant

Un projet doit pouvoir survivre à des vacances

ou un changement d’équipe.

Dépendance au sachant

Solution :

- Documenter.
- Maintenir la documentation à jour.

Dépendance au sachant

Mais on ne lit pas les doc...

Dépendance au sachant

Solution :

- Ne pas oublier l'humain et la transmission de connaissance / expérience.
  - Pair programming.
  - Revue de code.
  - Accompagnement des nouveaux.
  - Eviter les spécialisation à outrance.
  - Des binôme par sujet.
  - Des mini présentation sur des sujet.
  - Des réflexions en interne pour trouver des solutions
  - ...

Un petit dernier pour la route.

Dépendance à l'indépendance

À vouloir être un ayatollah de l’indépendance on risque de ne jamais finir les choses.

Conclusion

Notre métier

La dépendance et un risque.

0 dépendance est impossible.

Nous devons choisir nos dépendances

En prenant en compte :

Le risque / Les conséquences / Le coût