

Architecture logicielle, composants et frameworks

M1 MIAGE Université Paris 1 Panthéon-Sorbonne

2020-2021



JAKARTA EE



L'architecte Logiciel



UNIVERSITÉ PARIS 1

PANTHÉON SORBONNE

Définitions

- Années 1990: Un architecte logiciel élabore les règles et les standards de fabrication du logiciel sans nécessairement être affecté à des tâches de développement.

Mais...

- 1998 : Naissance des communautés Free Software (FSF) et Open-Source (OSI): le rôle d'architecte est occupé par un Lead technique, qui continue à faire ses preuves en tant que développeur.

Donc, la vision et la valeur du métier d'architecte logicielle ne sont pas claires et on doit d'abord **s'interroger sur la définition de l'architecture logicielle**

L'architecture logicielle



UNIVERSITÉ PARIS 1

PANTHÉON SORBONNE

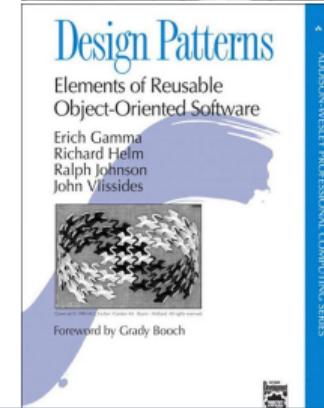
Tentative de définition: standards



- C'est un terme emprunté à l'architecture de bâtiments, mais cette définition se rattache à des concepts complètement différents.
- Standard ANSI/IEEE 1471:2000 : *l'organisation fondamentale d'un système sous la forme de ses composants, leurs relations entre eux et à l'environnement et les principes qui gouvernent sa conception et son évolution*

Tentative de définition: practitioners

- Ralph Johnson[1] + Martin Fowler: *la compréhension commune des développeurs experts sur la conception du système*[2]
 - l'architecture comme un phénomène social
 - les artefacts d'architecture (diagramme, documentation...) ne sont finalement qu'une représentation imparfaite de cette compréhension commune



[1] Johnson, Ralph, and John Vlissides. "Design patterns." Elements of Reusable Object-Oriented Software Addison-Wesley, Reading (1995).

[2] Fowler, Martin. "Who needs an architect?" IEEE Software 5 (2003): 11-13.

Tentative de définition 2

- L'ensemble des choix techniques qui doivent être pris tôt dans le projet

Tentative de définition 2

- L'ensemble des choix techniques qui doivent être pris tôt dans le projet
- Les choix techniques sur lesquels on ne voudrait pas trop se tromper.

Tentative de définition 2

- L'ensemble des choix techniques qui doivent être pris tôt dans le projet
- Les choix techniques sur lesquels on ne voudrait pas trop se tromper.
- Les choix techniques qui sont difficiles à changer

Tentative de définition 2

- L'ensemble des choix techniques qui doivent être pris tôt dans le projet
- Les choix techniques sur lesquels on ne voudrait pas trop se tromper.
- Les choix techniques qui sont difficiles à changer
- Finalement, quels sont les choix techniques pour lesquels il est difficile de revenir en arrière?
 - choix du langage de programmation (habituellement, ce n'est pas considéré comme un composant clé de l'architecture!)

Tentative de définition 3

- Finalement nous avons les 2 aspects:
 - Les choix techniques pour lesquels il est difficile de revenir en arrière
 - Compréhension commune de la conception
- Ce qui nous donne

Choix techniques
difficiles à changer

La compréhension commune
de la conception



FIGURE 1 – tentative de définition

Tentative de définition 3

- Finalement nous avons les 2 aspects:
 - Les choix techniques pour lesquels il est difficile de revenir en arrière
 - Compréhension commune de la conception
- Ce qui nous donne

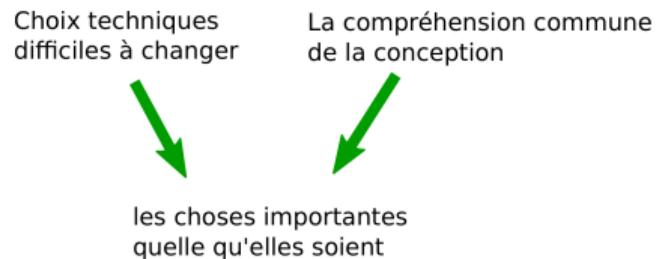


FIGURE 2 – tentative de définition

- Finalement, lorsqu'on se pose des questions sur l'architecture, il faut d'abord se poser la question de ce qui est vraiment important pour le système!

Pourquoi se soucier d'architecture logicielle 1

- Chef de projet: “*On doit souvent mettre moins l'accent sur la qualité pour produire plus de fonctionnalités.*”

Pourquoi se soucier d'architecture logicielle 1

- Chef de projet: “*On doit souvent mettre moins l'accent sur la qualité pour produire plus de fonctionnalités.*”
- Équipe technique: *il faut faire du code de qualité, c'est notre métier, ce sont nos valeurs, notre art, notre savoir-faire*

Pourquoi se soucier d'architecture logicielle 1

- Chef de projet: “*On doit souvent mettre moins l'accent sur la qualité pour produire plus de fonctionnalités.*”
- Équipe technique: *il faut faire du code de qualité, c'est notre métier, ce sont nos valeurs, notre art, notre savoir-faire*
- Avec cet argument, l'équipe technique perd à tous les coups:
 - **Finances vs savoir-faire**
- Donc il faut trouver une valeur financière à une architecture de qualité.



Qualités d'un système

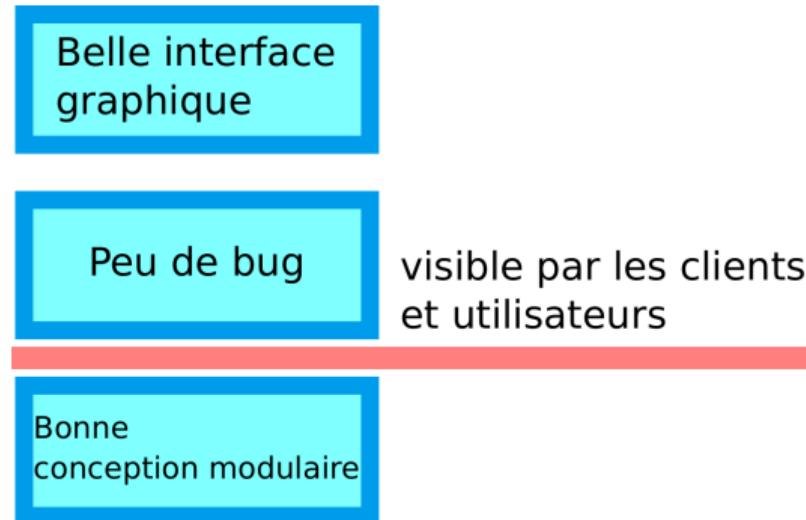


FIGURE 3 – Qualité visible vs invisible

Qualités internes vs Externe

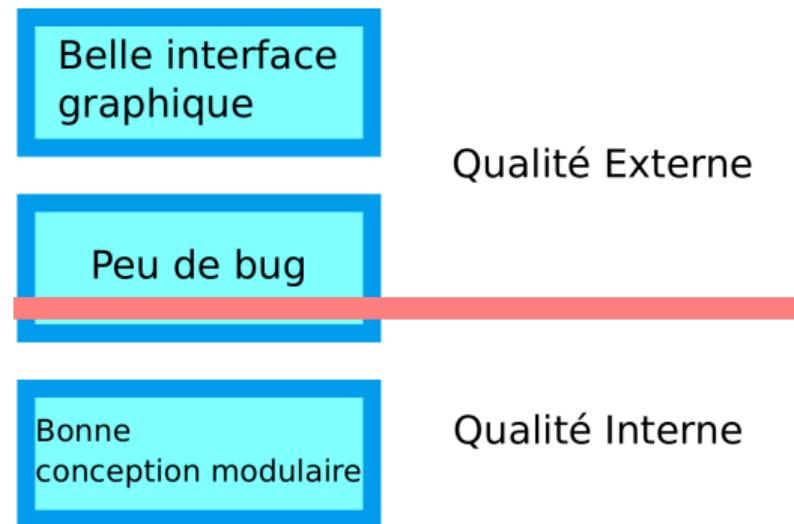


FIGURE 4 – Qualité interne vs externe

- Architecture = Qualité interne
- À fonctionnalités équivalentes, est-ce que vous achèteriez un logiciel 100€ ~~de plus~~ SIONNE sa qualité interne est meilleure?

Qualité interne = le long terme

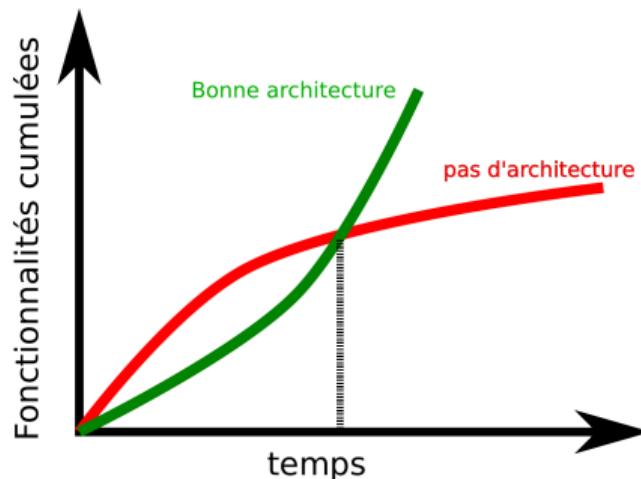


FIGURE 5 – Hypothèse de l'endurance de la conception

- <https://martinfowler.com/bliki/DesignStaminaHypothesis.html>
- <pub> Envie de prouver cette hypothèse? et pourquoi pas l'étudier dans votre mémoire de M1 </pub> ?

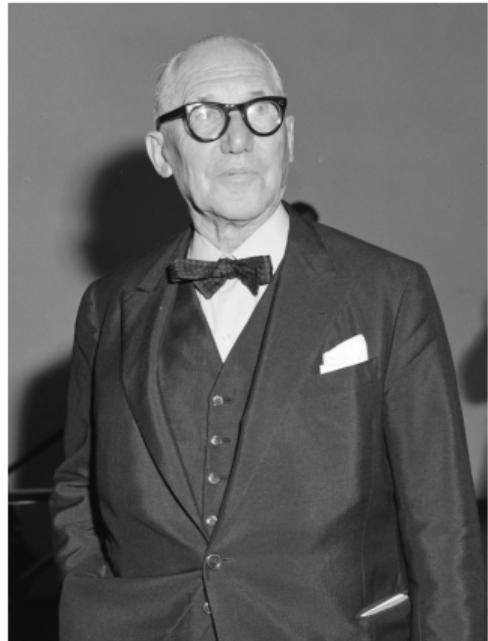
Qualité interne = le long terme

- Peut être est-il préférable d'acheter le logiciel plus cher avec une meilleure architecture
 - de nouvelles fonctionnalités délivrées plus vite par le vendeur
 - cas de l'intégration continue
 - patches de sécurité
- **Voici la raison économique poussant vers une architecture de qualité**
- Hypothèse 2 : le break-even apparaît en quelques semaines.

Comment être un bon architecte?

Dans la prochaine section, nous allons aborder les 3 phases du travail de l'architecte

- La planification des systèmes
- La conception des systèmes
- Le construction des systèmes



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

La planification des systèmes



UNIVERSITÉ PARIS 1

PANTHÉON SORBONNE

La planification des systèmes

- Équipe & Cycle de vie du projet
- Définition du projet
- Périmètre du projet et estimation
- Interfaces externes du système

Équipe projet

Dénomination FR	Dénomination internationale	Techno
Architecte logiciel	Lead tech	UML, Word
Chef de projet	Project Manager	Excel, Gant
Chef de produit	Product Manager	PowerPoint, Excel
MOA	Business Analyst	UML, BPMN, word
Graphic designer	UI-UX specialist	Photoshop
Dev présentation	Dev front	HTML/CSS
Dev server	Dev back	JEE, C#
Spécialiste des données	data science engineer	R, Oracle, NoSQL
Administrateur base de données	DBA	Oracle
Administrateur Infrastructure	operation engineer	linux, bash
Ingénieur tests	Q&A engineer	Xunit, word



PANTHÉON SORBONNE

Tâches de l'Architecte Logiciel



Etre un architecte logiciel
n'est pas du tout stressant

John, 34 ans



UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Tâches de l'Architecte Logiciel

- Identification des technologies utilisées pendant le projet
 - langage, vendeur de solutions, dépendances tierce
- Recommandation des méthodes de développement au projet manager
 - Scrum, user-stories
- Conception et structure de l'application
- Définition du système (*qu'est-ce qui est important?*)
- Doc technique
- Guides de développement
- identification des tâches de développement
- support à l'équipe de dev
- code review
- suivi des couts + gestion RH

L'équipe de management / fonctionnelle



PANTHÉON SORBONNE

L'équipe de management / fonctionnelle

- Product Manager
 - S'assure que le système répond aux besoins du marché
 - **Épaulé par l'architecte sur la *faisabilité***
- Project Manager
 - Coordonne et planifie les tâches de l'équipe de dev
 - Communique sur l'avancement du projet à l'extérieur
 - S'assure que son équipe ait les moyens de travailler
 - **Épaulé par l'architecte sur l'*identification des tâches techniques***
- Business Analyst
 - Traduis un besoin utilisateur en exigences
 - Référence des développeurs pour les questions métier
 - **L'architecte doit s'assurer que les exigences sont adéquates**

L'équipe technique



L'équipe technique

- UI-UX specialist
 - fournit les éléments graphiques pour intégration au dev front
 - **L'architecte doit s'assurer de la faisabilité du design**
- dev front
 - responsable du développement de l'interface graphique
 - **L'architecte doit s'assurer de la maintenabilité de l'IHM**
- dev back
 - responsable du développement des règles métier, côté serveur
 - **L'architecte sert de mentor et le guide**
- spécialiste des données
 - responsables du cycle de vie des données dans le système
 - **L'architecte doit s'assurer que le modèle de donnée est adéquat**
- Test Engineer
 - S'assure de la cohérence des développements par rapport aux exigences à l'aide de moyens techniques

L'Équipe des opérations



L'Équipe des opérations

- DBA
 - fournis un environnement stable pour l'exploitation des données
- Operation Engineer
 - responsable des environnements de dev, test et prod, du déploiement et de la mise en production

Cycle de vie du projet

- Cycle de V, Cascade
- Méthodes agiles
- RUP

Agile/DEVOPS: vers une nouvelle vision de l'équipe?

- **Agile:** fusion des équipes dev + management + fonctionnelle
 - Les développeurs doivent comprendre le métier
 - Le métier doit rendre des comptes aux développeurs
 - La gestion de projet doit être plus participative
- **DevOPS:** fusion des équipes dev + opération
 - Le développeur doit prendre en compte le contexte opérationnel de leur développement
 - Les équipes d'opération doivent comprendre ce que fait le logiciel
- **Entreprises libérées:** une forme organisationnelle dans laquelle les salariés sont totalement libres et responsables dans les actions qu'ils jugent bon (eux et non leur patron) d'entreprendre
 - Forme ultime d'émancipation des salariés?
 - Aliénation encore plus grande à l'entreprise?
 - <pub> Envie d'étudier cette forme d'organisation? Et pourquoi pas dans votre mémoire de M1? </pub> ?