

# Principes d'ingénierie en développement Web



## Bio du présentateur

# Benoit Piette

Benoit Piette est un spécialiste des technologies Internet depuis plus de 14 ans et architecte de solutions Web chez Tink Profitabilité numérique. Dans sa quête d'améliorer la qualité de ses interventions, il devient membre actif du W3Québec dès sa fondation en 2003 et en fût le président de 2006 à 2008. Il est expert invité au HTMLWG du W3C et participe à plusieurs conférences, dont Confoo.ca, MakeWebNotWar, Webéducation et Intracom (au Québec).

[benoitpiette.com](http://benoitpiette.com)

## W3Québec

Le W3Québec est un organisme à but non lucratif dont l'objectif est de promouvoir les bonnes pratiques et les standards ouverts en développement Web.

[W3Qc.org](http://W3Qc.org)

# Table des matières

1. En avant avec l'industrialisation!
2. Préoccupations transversales
  3. Preuves
4. Quelques pratiques

## Préjugés et industrialisation



## Encore des histoires

- On a souvent voulu se débarrasser du programmeur en le transformant en ouvrier
  - On a essayé de le remplacer par des analystes d'affaires
    - On a essayé de l'*oustourcer*

## Ouvriers ou concepteurs ?



## Qu'est-ce qu'un ingénieur ?

Un ingénieur (du latin ingenium) est un professionnel exerçant des activités de conception et de direction de projets, de réalisation et de mise en œuvre de produits, de systèmes ou de services impliquant des problèmes techniques complexes. Ces responsabilités supposent alors un ensemble de connaissances techniques d'une part, économiques, sociales, environnementales et humaines d'autre part, reposant sur une solide culture scientifique.

[[Source : Wikipedia](#) ]

- Organisation
  - Science
- Humain et social

## Objectifs

- Proposer des orientations / principes
  - Keep it simple
- Ne pas tomber dans une méthode particulière, il y a bien d'autres présentations pour cela
  - Je n'ai rien à vendre :-)
- Dans cette optique, nous resterons haut niveau

## **Objectif : Qualité ou monétisation ?**

- Malgré les processus, les projets sont en retards et au dessus du budget
  - Il y aura toujours des besoins contradictoires
    - Lesquels écouter ?

## **La difficulté de créer des principes d'ingénierie**

- Beaucoup de dogmes, peu d'élus
- Le problème revient toujours à l'humain
- Les orientations de l'industrie changent très souvent

## **Le principe de base en intégration Web**

- Séparation structure / présentation / comportement
- Comment : les techniques, qui risquent de changer

## **Courants actuels de l'industrie**

- Économie de biens virtuels
- Informatique en nuages
  - Web et Mobile
- Perspectives économiques incertaines
  - Communications sans vie privée
- Design et ergonomie / limites de la technologie
  - Touch computing

## **Courants actuels de l'industrie**

- Diminuer les coûts
  - On demande plus de qualité
- Les changements sont toujours aussi rapides
  - Industrialisation ?
    - Problèmes sérieux de sécurité, intégrité
    - Être à l'écoute des besoins de l'industrie

## **Il n'y a pas de "One size fits all"**

- Trop généraliser coûte cher en intégration.
- La solution très spécifique et temporaire devient permanente

## **Exemple de Keep It Simple**

# **Être idéaliste dans ce qu'on vise**

# **Être pragmatique dans ce qu'on fait**

## **La Maîtrise du sujet**

- La compréhension "ownership" d'un projet
- Pas tout-à-fait dans les sens de Scrum.
- Autant les aspects organisationnels que technologiques et humains.

## **Caractéristiques de la chaîne de montage d'application Web**

- Créativité artistique et ingénierie
  - Le produit est la page Web
- La chaîne de montage est l'application
- Nous travaillons à construire ou assembler cette chaîne de montage
  - Méta-chaîne de montage et méta-plan
    - L'effet papillon est important!

## **Qualité et préoccupations transversales**

- Gestion
- Accessibilité
  - Sécurité
- Performance
- Ergonomie utilisabilité, aspects cognitifs
  - Référencement SEO, SEM

## **Qualité et préoccupations transversales**

- Qualité et recherche de bogues
- Mise en place de normes de développement
  - Impact des choix d'architecture
    - Identification des risques
- S'empêcher de réinventer la roue

## Sécurité et OWASP

- Exemple (incomplet) de norme de sécurité basé sur OWASP

<b>Top 10</b>	Activité
A1 : Injection	<ul style="list-style-type: none"><li>● Utilisation selon les règles de l'art d'un ORM</li><li>● Toujours filtrer les entrées au système</li><li>● Revue de code hebdomadaire</li></ul>
A2 : Cross Site Scripting (XSS)	<ul style="list-style-type: none"><li>● Toujours filtrer les entrées au système</li><li>● Revue de code hebdomadaire</li><li>● Tests de sécurité applicatifs</li></ul>
...	

## **Accessibilité**

- Principe d'amélioration progressive
- Normes gouvernementales en accessibilité (ex: SGQRI-008 au Québec)
  - Revue du design / code

## **Performance et robustesse**

- Calcul des besoins en performance
- Identification des goulots d'étranglement
  - Tests de performances
  - Identification des fuites de mémoire
- Mise en place d'environnements reflétant le plus possible la réalité

## **Il est question d'optimisation**

- S'assurer que les bases sont respectés
- Ensuite, on optimise selon les besoins du projet

**Toujours avoir une preuve en main**

## **Rechercher le maximum de confiance**

- Peu importe les outils et méthodes
  - Selon le besoin (et le budget)
- Idéalement, la preuve est mathématique / scientifique et représente la réalité

## **L'effort nécessaire**

- On ne peut pas toujours éviter la complexité
- Il n'y a peut-être que les fous qui ne doutent pas
- Mais si le doute nous empêche d'agir, il faut douter du doute
- La meilleure solution, c'est la vôtre (and stick with it)

## **Stratégie de découpage**

- Identifier et éliminer les risques le plus tôt dans le projet
- Identifier les priorités : la valeur pour le client
  - Faire attention au surmenage
  - Contraintes de projet

**Être à l'écoute - l'humain est le centre de  
toute organisation**

## **Développement durable, pérennité et durée de vie**

- Les technologies changent rapidement
- Les méthodologies évoluent, et tout le monde pense différemment
  - Trop d'opinions
- Établir des normes de développement et guérites de validation

# L'importance du prototype

# Gestion des sources

# Tests unitaires et automatisés

# **Builds automatisés / intégration continue**

## **La place de l'analyse , architecture / modèles**

- Assurer la maîtrise du sujet

## **Tout prendre avec un grain de sel**

- Avec toutes ces contradictions...
- On a le droit d'être passionné par ce qui nouveau
- On a le droit d'écouter ceux qui y sont passés avant nous
- On réinvente souvent la roue (des fois elle est un peu plus vite, mais vaut-elle la peine)

## **Design patterns**

- Je n'en ferai pas la liste, mais...
- Faut au minimum être au courant.
  - Séparation des préoccupations
    - Modèle vue contrôleur

## **Refactorisation**

- Quand elle est nécessaire
- Contrôler la dette technologique

## **Séparer les environnements**

- Développement, tests, assurance qualité
  - Pré-production, staging, production

# Déploiements

## Vérité et preuves

- L'objectif d'une preuve est d'augmenter les chances de succès d'un projet, pas de cacher les problèmes en se donnant un sentiment que tout va bien.
- Niveau de confiance des preuves selon le contexte
  - Pour valider : normes et guérites
  - Preuves par expérience ? Selon le contexte

## **Unification**

- Normes de développement (niveaux CMMI)
  - Rechercher la répétition du succès

## **Culture d'entreprise et gestion du changement**

- Changement d'orientation
- Croissance de l'entreprise - industrialisation à petite échelle
  - Attention à l'inflexibilité

# Le méta-plan

## **Gestion du temps**

- Point de vue développeur
- Organisé, mais pas trop

## Les risques

- Minimiser les hypothèses non vérifiées et les incertitudes
  - Les éliminer le plus tôt dans un projet
    - Encore une fois, le prototype aide
    - Se méfier de "Ça devrait marcher !"

## Livrer!

- Toujours garder en tête que l'objectif est de livrer
- (préférentiellement en temps et budget et avec le niveau de qualité visé)
- Même quand le temps et le budget sont dépassés
  - Garder le focus, ne pas abandonner!
  - Contrôler la dette technologique

**Ne pas oublier le travail d'équipe!!**

## Récapitulation

- Keep it simple
- Il n'y a pas de "One size fits all"
- Être idéaliste dans ce qu'on vise
- Être pragmatique dans ce qu'on fait

## Récapitulation

- La confiance en nos preuves doit être le plus élevée possible
  - Suivre les bonnes pratiques de base (tests, etc)
- Nous sommes plus que des exécutants et ne jamais l'oublier!
  - Ne jamais laisser tomber!

**Pespectives : est-ce qu'on est encore dans l'ouest ou est-ce que le train vient de passer ?**

## Droits d'auteur

- Ce contenu est déposé intégralement sous licence Creative Commons de type BY-SA.
  - Les images contenues dans cette présentation appartiennent à leurs propriétaires respectifs et sont utilisées selon la doctrine de l'utilisation équitable (fair use). (En gros, je les enlève si vous le voulez)

## Pour continuer

- [W3Qc](#): Les standards Web au Québec!
- D'autres ressources à venir

## Commentaires? Questions?

- Merci!
- Mon site Web personnel, <http://benoitpiette.com>
- Continuons notre discussion sur Twitter  
(<http://twitter.com/benoitpiette>)
- N'hésitez pas à me contacter :  
[benoit.piette@w3qc.org](mailto:benoit.piette@w3qc.org)

©2010-2011 Benoit Piette, ([CC BY-SA](#)).