

Sensibilisation aux bonnes pratiques techniques du Software Craftsmanship : Lego® à la rescousse !



Isabelle BLASQUEZ
[@iblasquez](https://twitter.com/iblasquez)

Alice BARRALON
[@a_barralon](https://twitter.com/a_barralon)

Manifesto for Software Craftsmanship

Elever le niveau

En tant qu'aspirants Artisans du Logiciel, nous relevons le niveau du développement professionnel de logiciels par la pratique et en aidant les autres à acquérir le savoir-faire. Grâce à ce travail, nous avons appris à apprécier :

Pas seulement des logiciels opérationnels,
mais aussi **des logiciels bien conçus.**

Pas seulement l'adaptation aux changements,
mais aussi **l'ajout constant de la valeur.**

Pas seulement les individus et leurs interactions,
mais aussi **une communauté de professionnels.**

Pas seulement la collaboration avec les clients,
mais aussi **des partenariats productifs.**

C'est à dire qu'en recherchant les éléments de gauche, nous avons trouvé que les éléments de droite sont indispensables.

© 2009, les sous-signés.
Cette déclaration peut être copiée librement, quelle que soit la forme,
mais seulement dans sa totalité, y compris cette remarque.

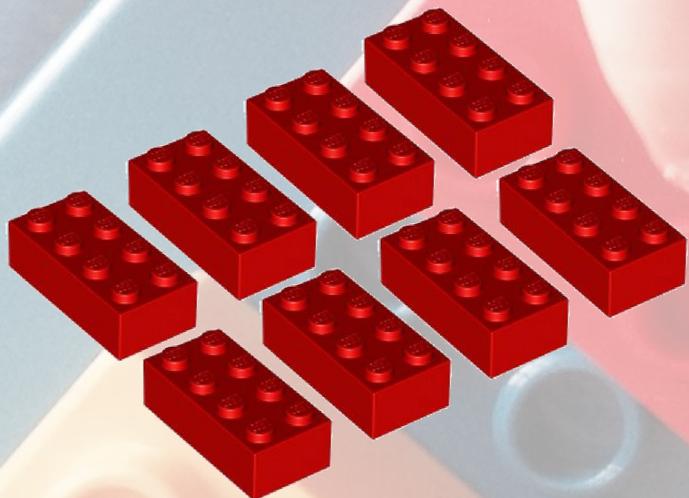
**A l'image du potier
qui travaille son argile en continu,
le développeur travaille son code
en permanence.** Benoit Gantaume



***Le jeu est comme un engrais pour la croissance du cerveau.
Il est fou de ne pas l'utiliser. Stuart Brown***

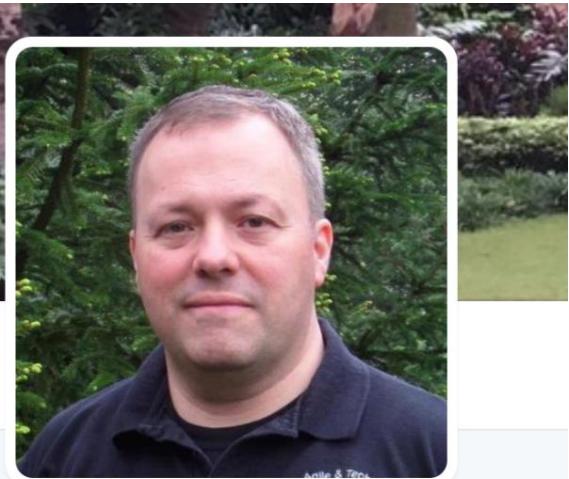
Il faut jouer pour devenir sérieux. Aristote

La puissance des LEGO®



⇒ **915 103 765 combinaisons possibles**

Adapté d'une idée originale de :



Mike Bowler

@mike_bowler

Agile & technical coach/trainer, LEGO
Serious Play facilitator, speaker, dad

📍 Smith-Ennismore-Lakefield

🔗 gargoylesoftware.com/mike_bowler



Bryan Beecham

@BillyGarnet

Anzeneer@Industrial Logic, Speaker,
#HumanRefactor Guide, Guitarist.
Passionate about the improvement of
companies, teams and individuals.

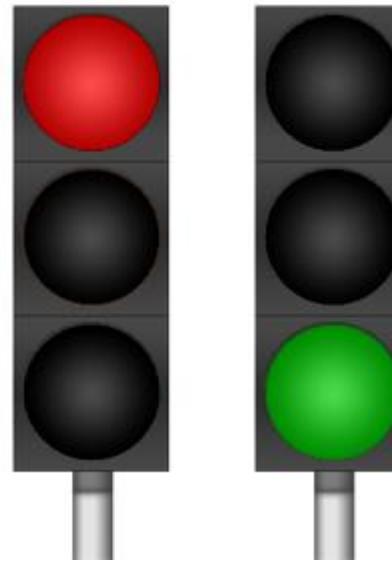
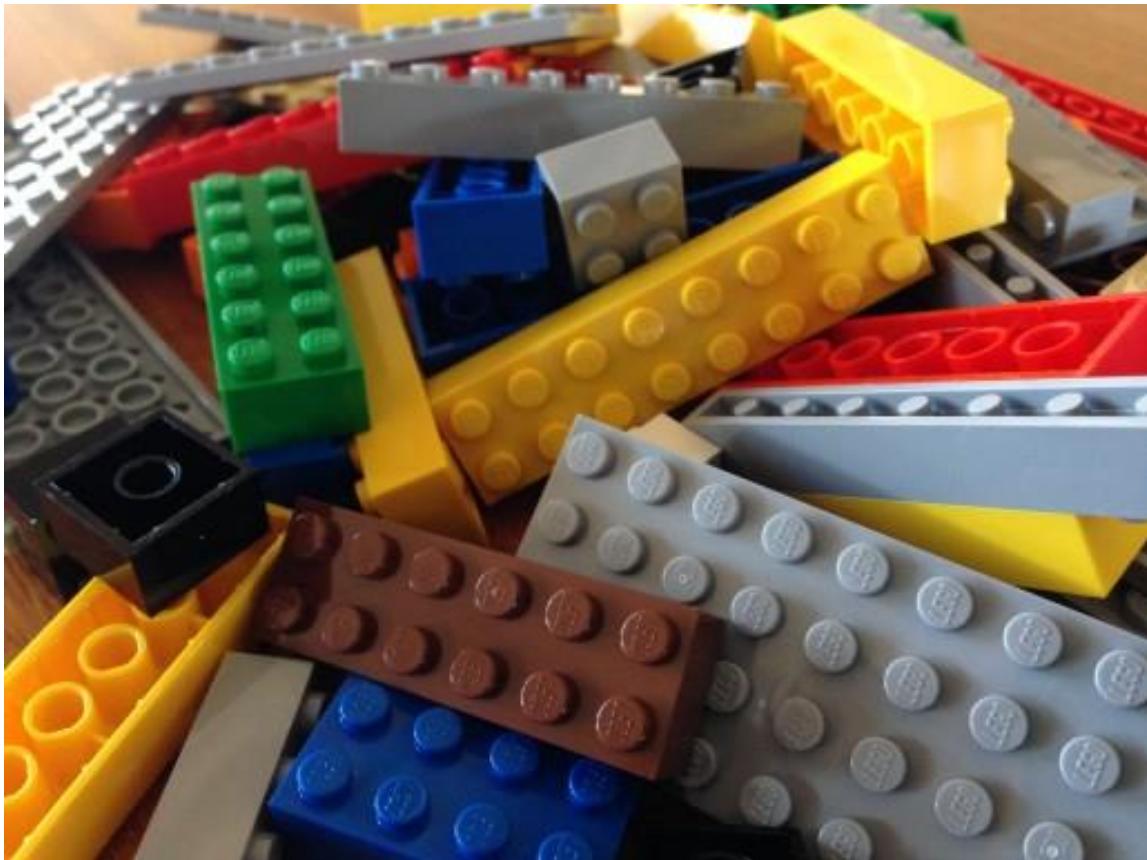
📍 Canada

🔗 billygar.net

Version originale :

About the LEGO technical practices exercises : <http://www.gargoylesoftware.com/ex>

Matière première ...



Images : <http://www.boost.co.nz/blog/2013/11/what-does-lego-and-building-high-quality-software-have-in-common/>
<http://www.passetoncode.fr/cours/feux-tricolore>
<http://www.bbbpress.com/2015/06/drama-game-index-card/>

Simple, pour commencer ...

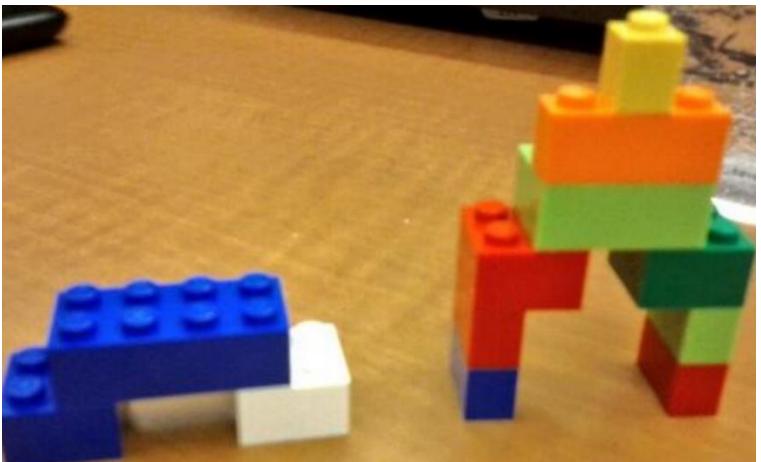


3 minutes

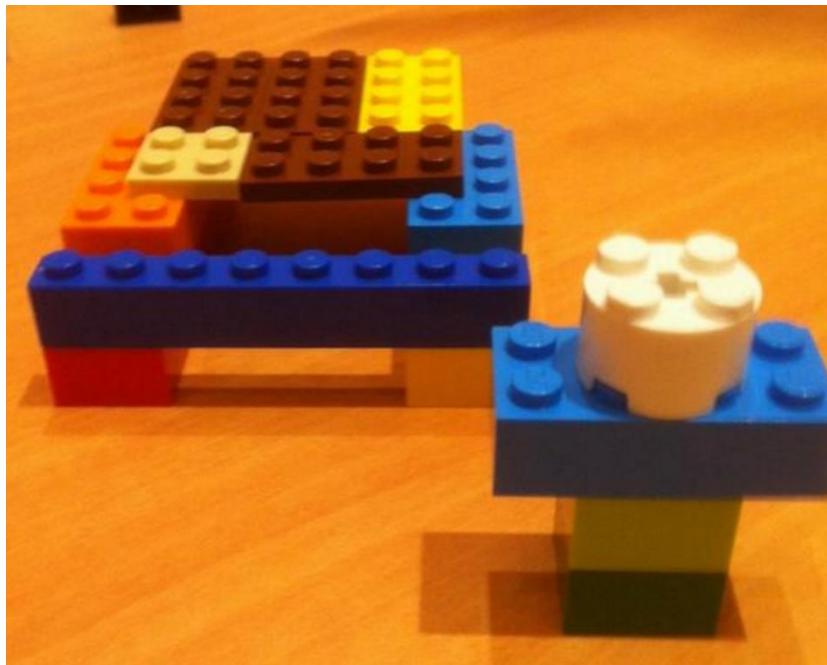
Construire une maison et une personne



Admirez et partagez !



 #TDDLego



Extraits de : <https://twitter.com/sahithyaW/status/428217265667198976>
<https://twitter.com/JeffHoover/status/515550469175799808>
<https://twitter.com/iGeaux/status/515550799045607424>
<https://twitter.com/iGeaux/status/515554621629415424>

Isabelle BLASQUEZ – Alice BARRALON

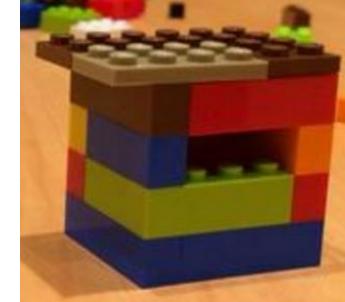
Debriefing



Valeur
métier

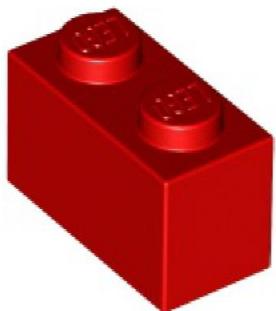


ET

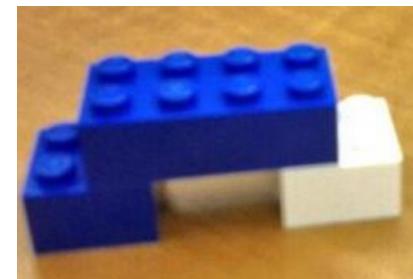


Coût

Temps ? Ressources ?



OU



OU



OU



?



Si on vous avait dit que chaque *point* de brique de LEGO® coutait 1000€, est-ce que cela aurait influencé votre design ?

⇒ « **juste assez** » (pensée Lean)

Extrait : <https://twitter.com/iGeaux/status/515554621629415424>

<https://twitter.com/sahithyaW/status/428217265667198976>

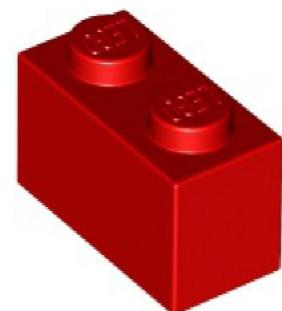
Pièces disponibles sur : <http://shop.lego.com/>

Isabelle BLASQUEZ – Alice BARRALON

Bonnes Pratiques : deux principes indispensables

KISS

Keep It Simple Stupid



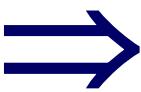
vs



YAGNI

You Ain't Gonna It

Une livraison KISS pour notre problème ...

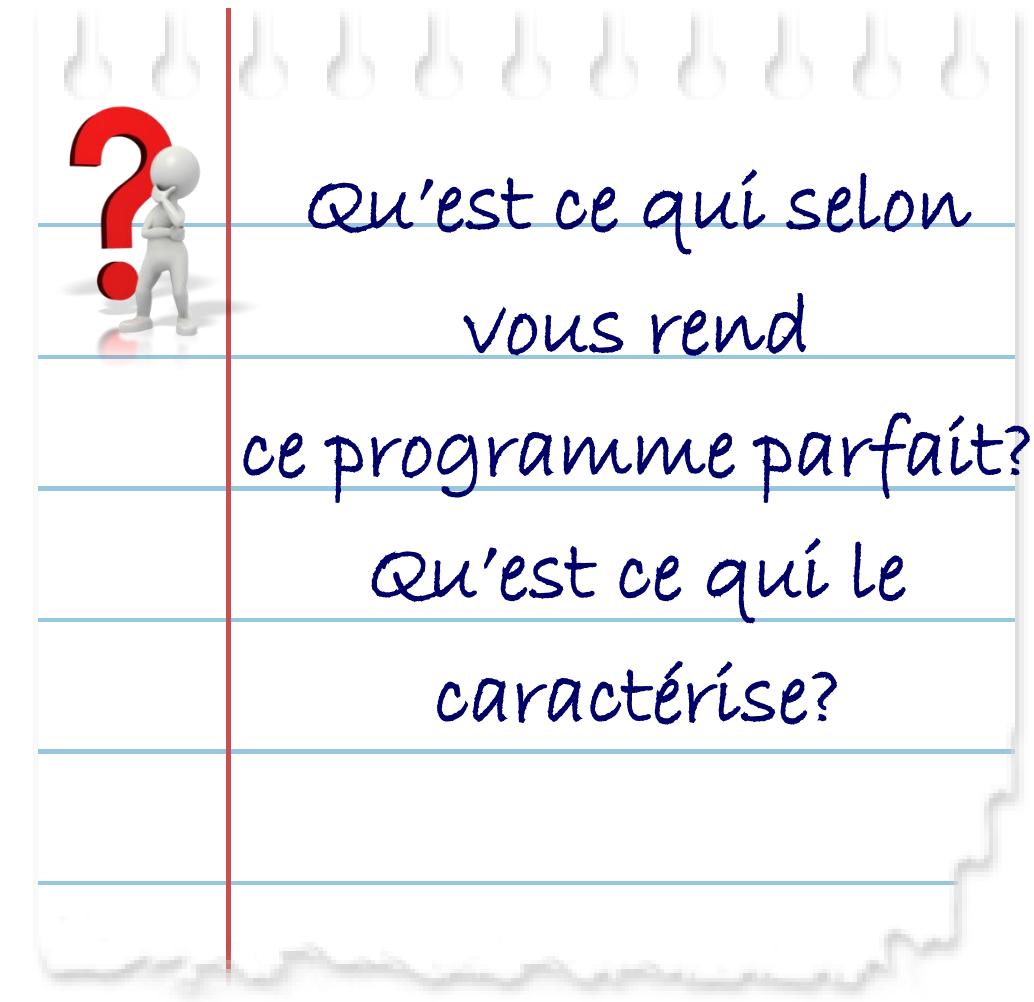
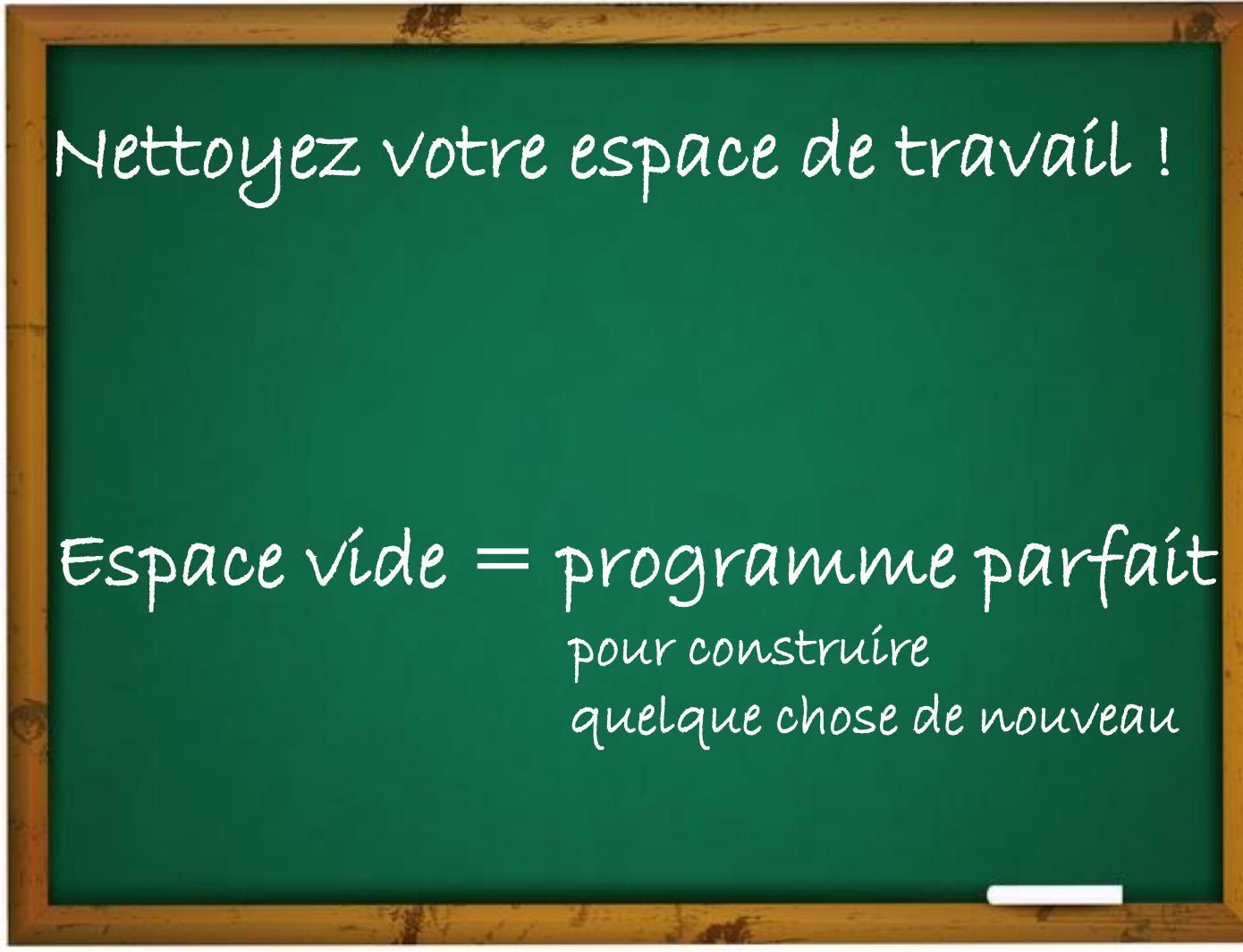


Une brique : le pouvoir de la métaphore au service de la simplicité



Extrait : <http://piwee.net/1-affiches-minimalistes-brique-lego-030516/Extrait>

Quid du programme informatique parfait ?



Quid du programme parfait ? (quelques éléments de réponse)



**Il marche !!!
Pas de bugs**

**Satisfait toutes
les exigences**

**S'exécute
extrêmement
rapidement**

**Implémenté
en un
rien de temps**

Ne coûte rien

**Très facile à
l'étendre
(de construire
au-dessus)**

**Très facile
à comprendre
et à maintenir**

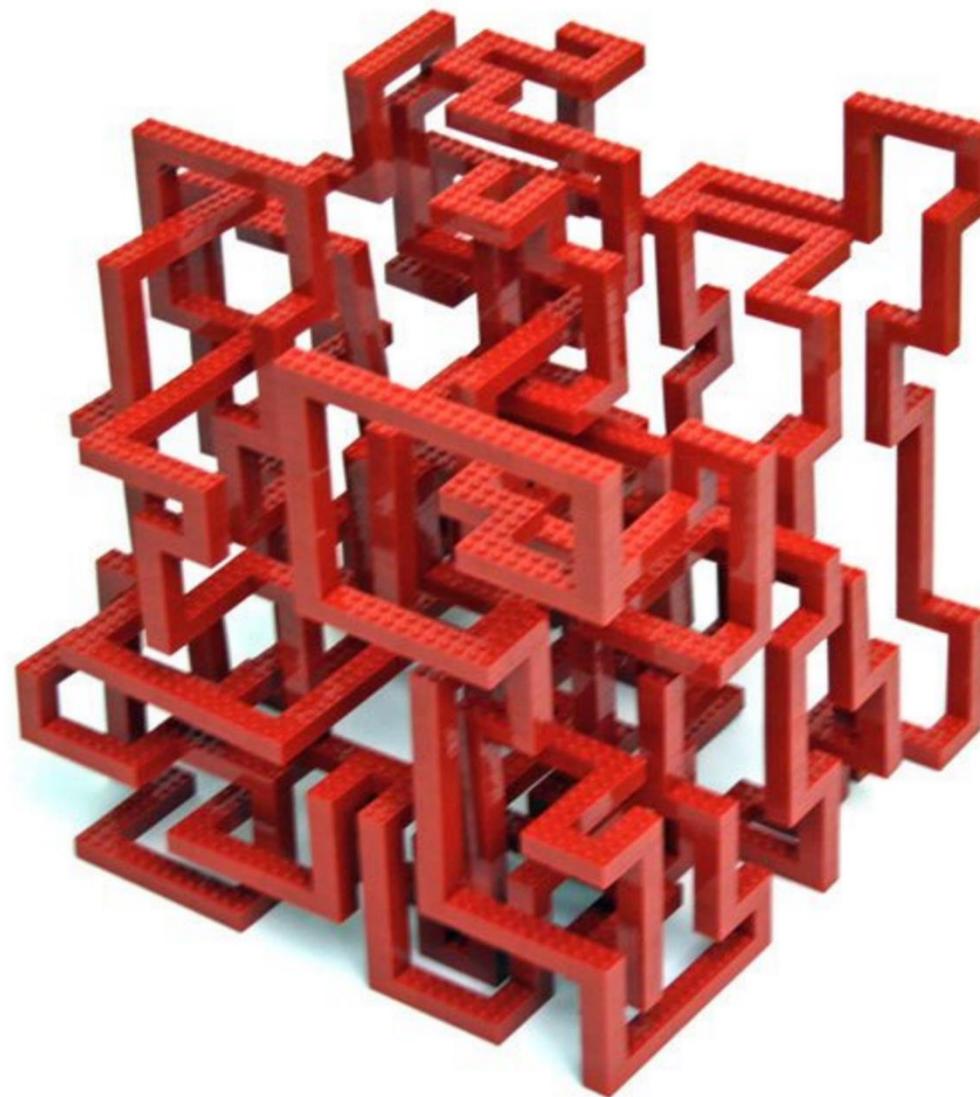
...

**Quand nous créons
un nouveau projet,
nous commençons
avec un
programme parfait**

**A chaque ajout
de code,
nous nous éloignons
de cet idéal.**

**Nous devrions nous efforcer
de tout mettre en œuvre
pour disposer
à tout moment d'un
logiciel aussi simple
et petit que possible
qui tend vers l'idéal.**

Vous avez dit : Simplicité ?



Crédits photo : Nathan Sawaya

Extrait du site : <http://www.topito.com/top-constructions-lego>

En savoir plus sur les oeuvres de Nathan Sawaya : <http://www.brickartist.com/> et <http://www.nathansawaya.com/>

Isabelle BLASQUEZ – Alice BARRALON

Bonne Pratique : Les règles de simplicité (eXtreme Programming)



Extrait : <http://c2.com/cgi/wiki?XpSimplicityRules>

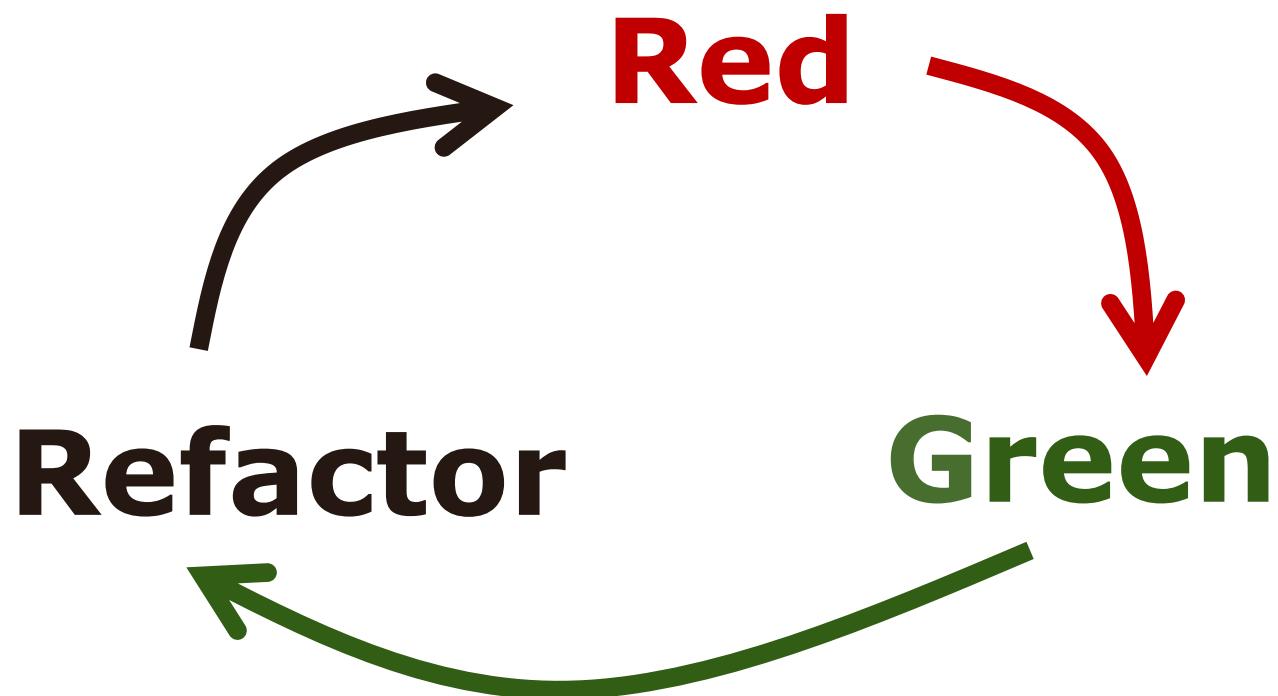
A lire également : <http://martinfowler.com/bliki/BeckDesignRules.html>
et <https://leanpub.com/4rulesof simplesdesign>



Test Driven Development (TDD) : à la Rescousse !

Test Driven Développement

But du TDD : « Clean code that works » Ron Jeffries



Itérations « baby-step » rapides
(de quelques secondes à quelques minutes)

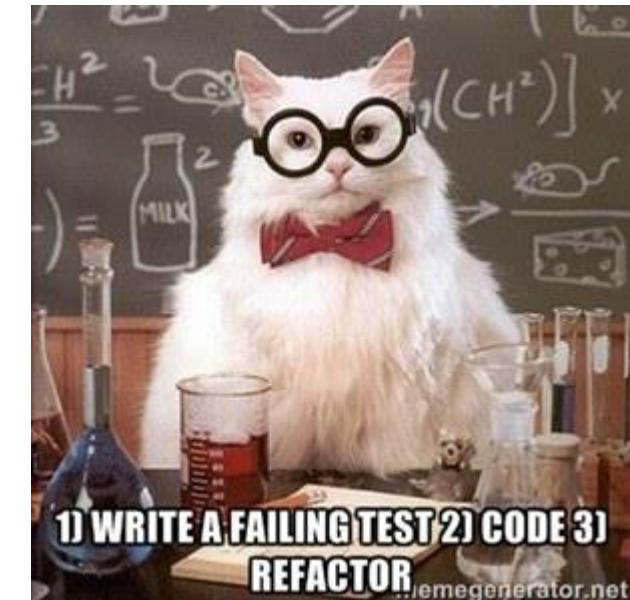
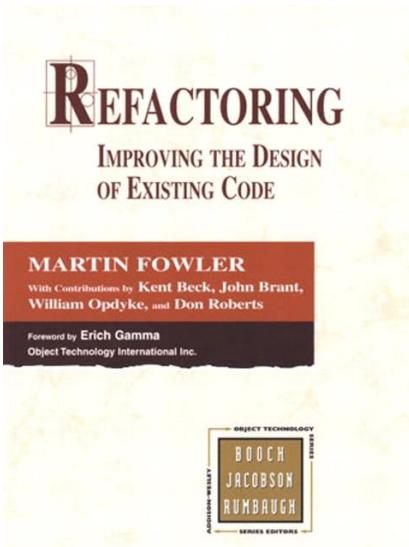


Image : <http://blog.octo.com/coder-a-pas-de-chaton-a-lecole-du-tech-lead/>

Refactoring : Définition



Un refactoring (remaniement) consiste à changer la structure interne d'un logiciel sans en changer son comportement observable (M. Fowler)

Pas de refactoring sans test !!!

TDD en pratique

Pas de code de production sans test !

**Un nouveau test =
un nouveau comportement**

**Ecrire au plus vite un code
qui fait passer les tests**

Let's go ! Itération 1

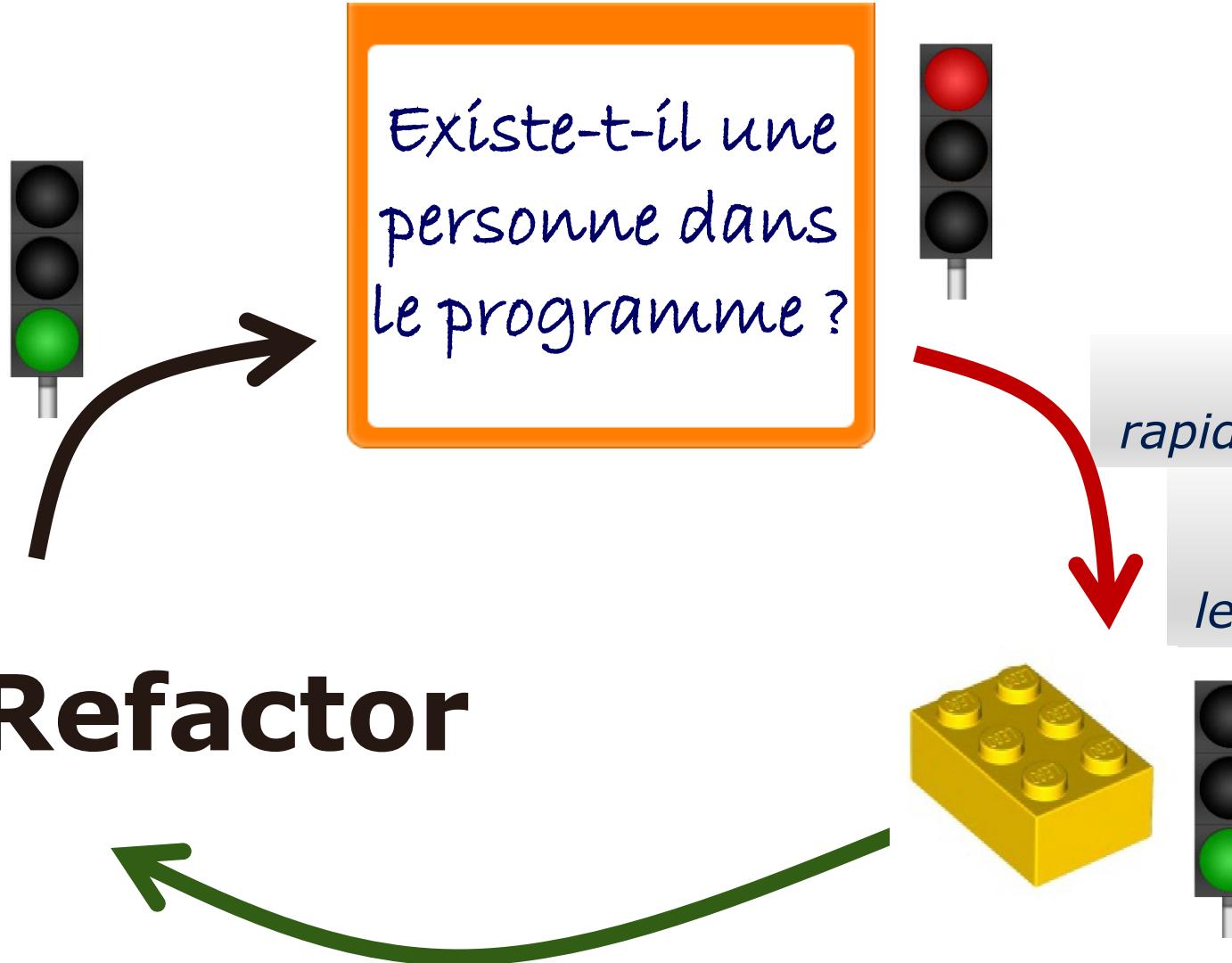
Toujours **un petit test en échec** pour commencer ...

... enfin,
« refactorer »
si nécessaire

Besoin de
simplifier la
conception ?

Y-a-t-il de la
duplication ?

Refactor



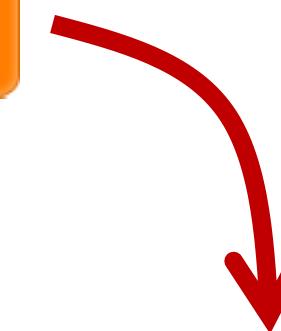
Let's go ! Itération 2

Lever votre carton vert lorsque l'itération est finie !

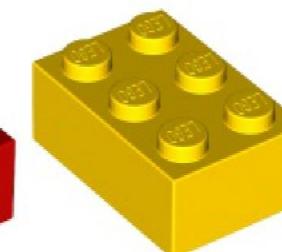
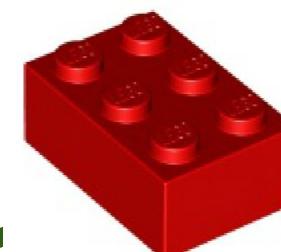


Besoin de simplifier la conception ?

Y-a-t-il de la duplication ?



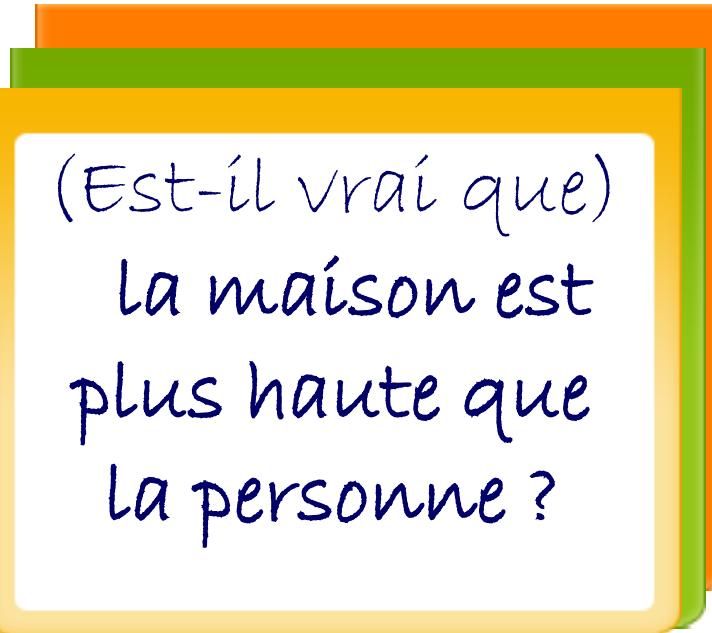
TOUS les tests doivent passés (succès = VERT)
pour garantir le comportement du programme



Refactor



Choix du prochain test ...



ou



?

```
assertTrue(maison.taille > personne.taille)
```

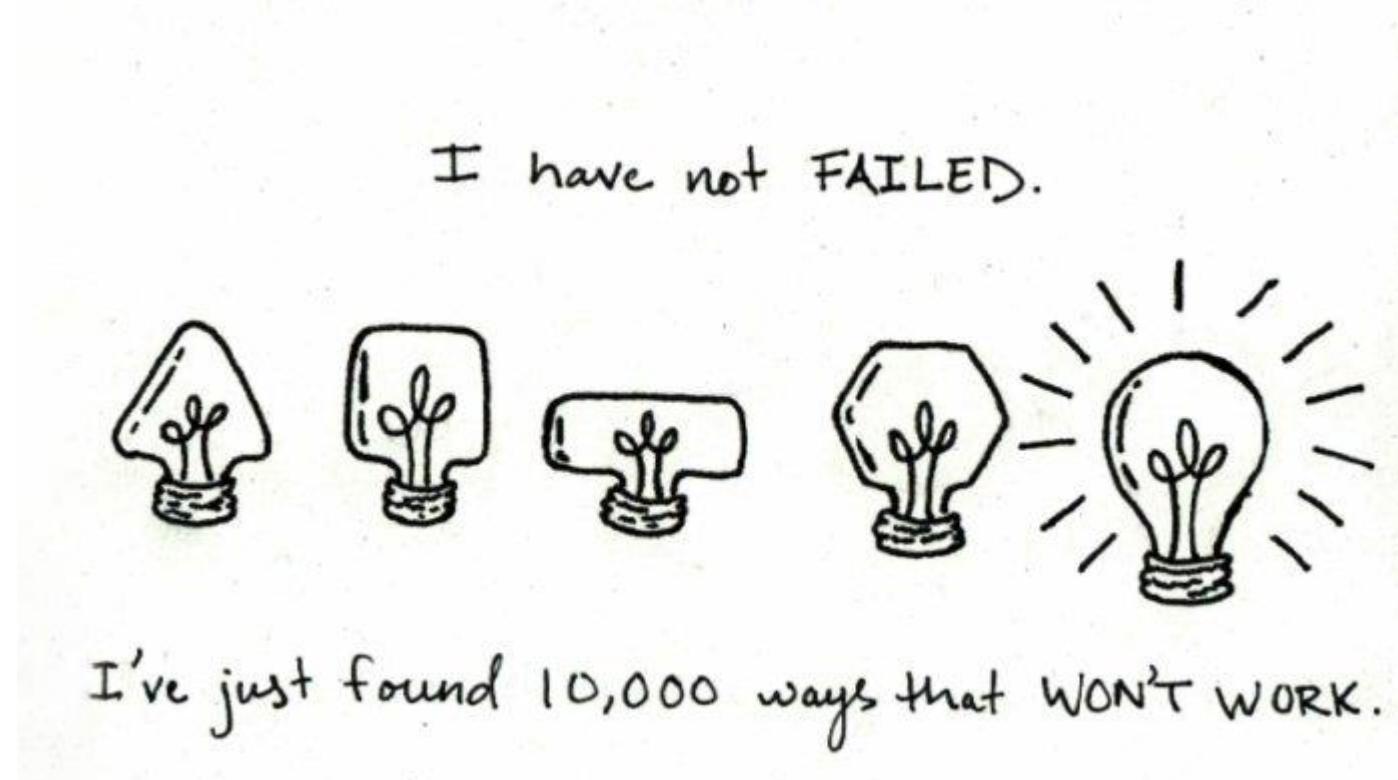


**D'un point de vue purement informatique,
pas encore de taille pour Maison et Personne**

Nouveau test - nouveau comportement : La maison doit être plus haute que la personne
On apprend un petit quelque chose de nouveau ...

Un échec est une réelle opportunité d'apprendre

If you're not failing, you're not learning. Bryan Beecham



If you're not failing, you're not innovating. Elon Musk (Tesla Motors)

Let's go ! Itération 3

Lever votre carton vert lorsque l'itération est finie !



*Besoin de simplifier la conception ?
Y-a-t-il de la duplication ?*

Refactor

La maison est
plus haute que
la personne ?



Effort minimum pour que tous les tests passent au VERT

Green

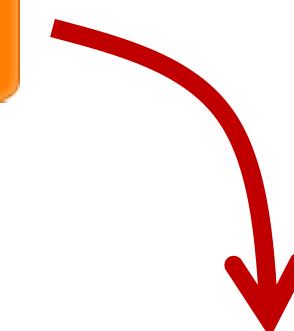
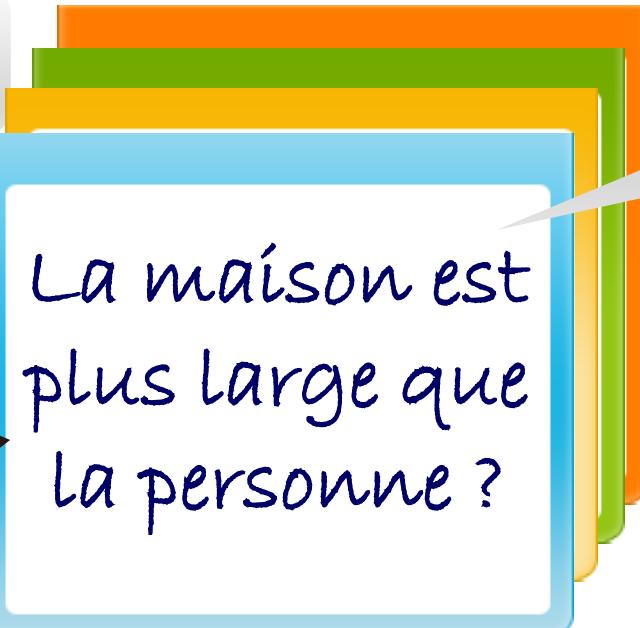
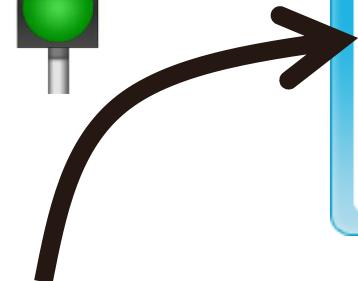


Let's go ! Itération 4

Lever votre carton vert lorsque l'itération est finie !

A chaque nouveau test,
nous apprenons
de plus en plus sur
les besoins de
l'utilisateur ...

Besoin de
simplifier la
conception ?
Y-a-t-il de la
duplication ?



Effort minimum
pour que
tous les tests
passent au **VERT**



Refactor

Green



Quel intérêt de créer une maison **et** une personne, s'il n'y a pas d'**interactions** entre ces deux objets ?

*Nouveau test doit être ajouté si et seulement si il **apporte de la valeur à l'utilisateur***

La personne peut entrer dans la maison ?

Let's go ! Itération 5

Lever votre carton vert lorsque l'itération est finie !



Besoin de simplifier la conception ?
Y-a-t-il de la duplication ?



Effort minimum pour que tous les tests passent au VERT

Refactor

Green



Prêt pour une nouvelle mission?

Détruire vos constructions

Nettoyer votre espace de travail



Indispensable REFACTORING pour minimiser la dette technique !

Descriptif original de l'atelier disponible sur : http://www.gargoylesoftware.com/ex/technical_debt

Image : https://twitter.com/LEGO_Group/status/710496008795336704

Isabelle BLASQUEZ – Alice BARRALON

Instructions



Exigences données pas à pas
(sous forme de condition de tests)

Une nouvelle exigence toutes les 30 secondes

Contrainte :
REFACTORING INTERDIT !!!

Deux briques assemblées ne pourront en aucun cas être séparées
c-à-d
aucune des briques utilisées ne pourra être enlevées

Exigence n°1



30 secondes

Exigence n°2



30 secondes

Exigence n°3

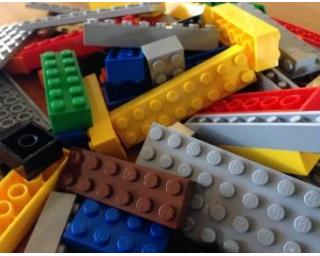


La maison doit être entièrement fermée.

Ajouter également des murs intérieurs

30 secondes

Exigence n°4



30 secondes

Exigence n°5



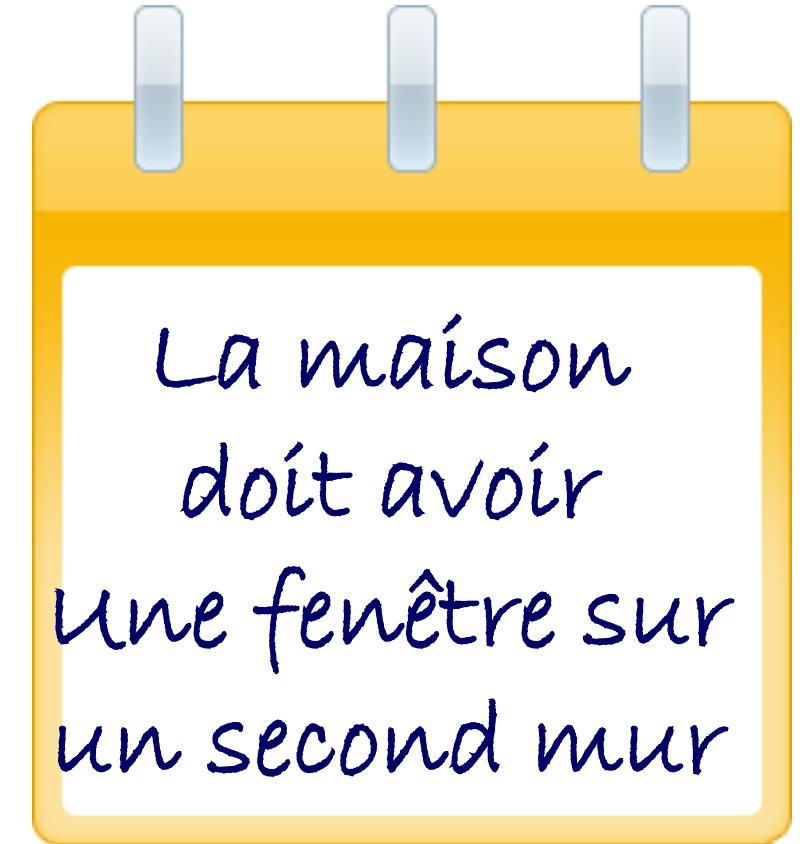
30 secondes

Exigence n°6



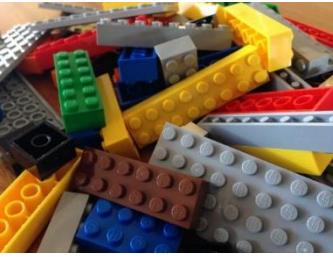
30 secondes

Exigence n°7



30 secondes

Exigence n°8



*Les toits en pente
ne se vendent plus !*

*La maison
doit avoir
maintenant
un toit plat*

60 secondes

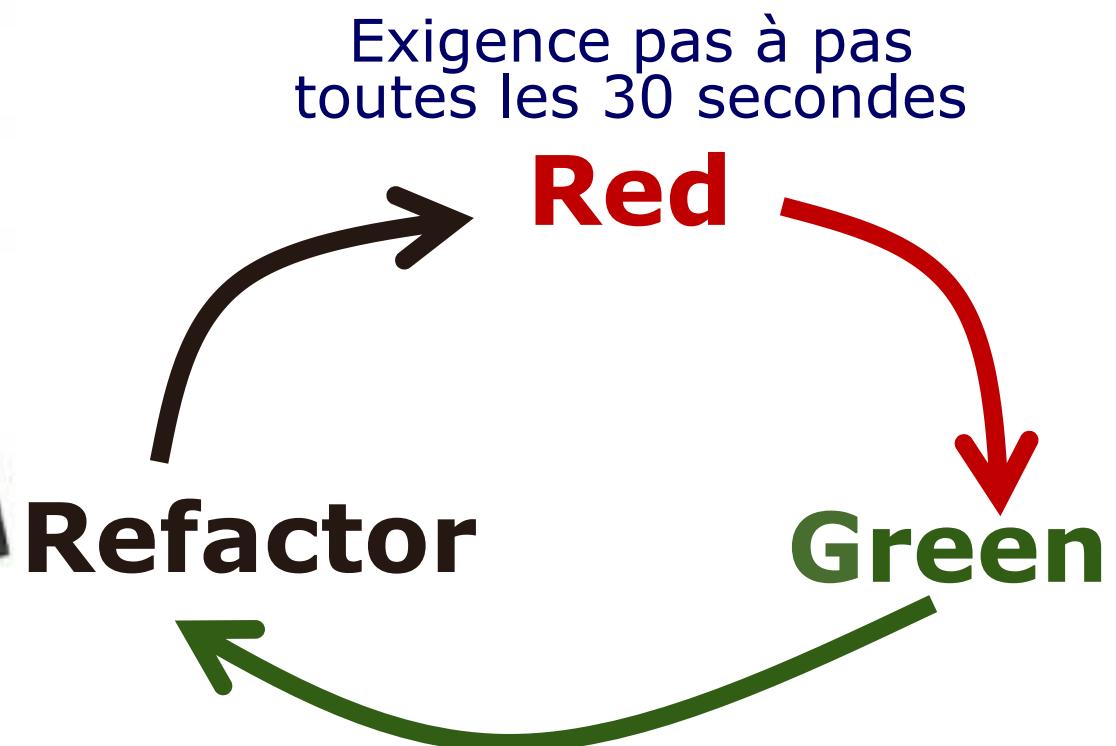
Admirez votre œuvre

La mettre de côté
(et ne pas la détruire !)

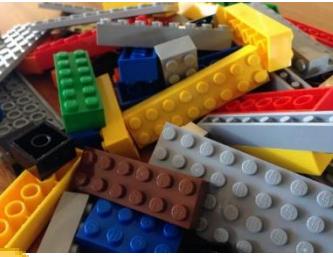


Contrainte levée :
REFACTORING AUTORISE !

c-a-d
Possibilité de séparer de séparer des briques déjà assemblées entre elles



La maison de mes rêves (refactoring autorisé)

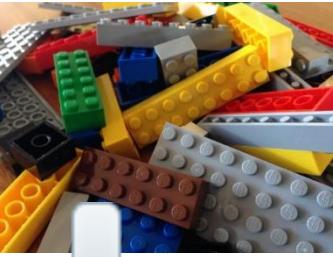


La maison
doit avoir
deux murs
connectés

La maison doit être
entièrement fermée.

Ajouter également
des murs intérieurs

La maison de mes rêves (refactoring autorisé)

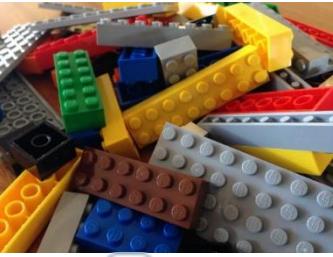


La maison
doit avoir
un toit en pente

La maison
doit avoir
une fenêtre
sur un mur

La maison
doit avoir
une fenêtre sur
un second mur

La maison de mes rêves (refactoring autorisé)



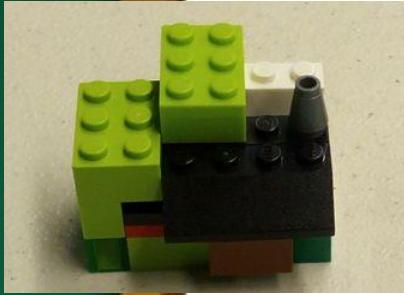
La maison
doit avoir
une cheminée

La maison
doit avoir
une porte au
rez-de-chaussée

La maison
doit avoir
maintenant
un toit plat

Débriefing:

Comparez vos constructions de la 1^{ère} itération
et de la 2nd itération.



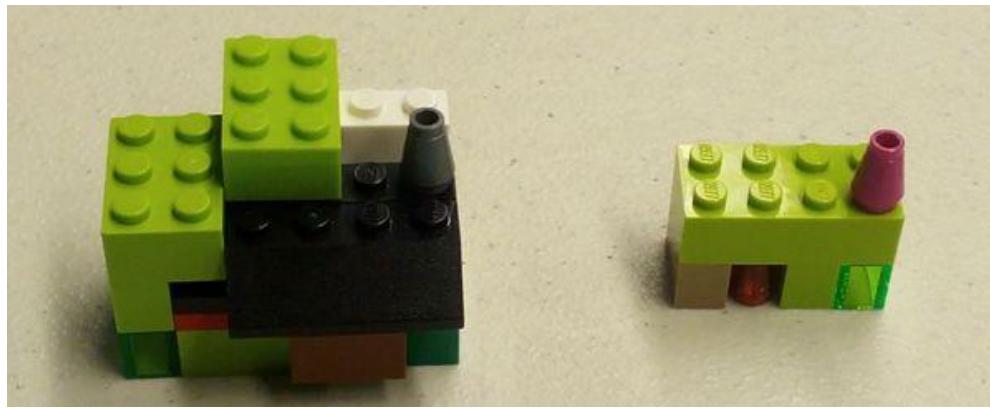
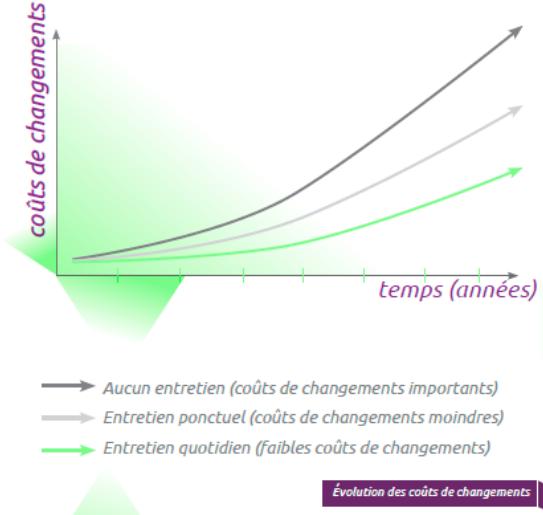
#TDDLego

Qu'avez-vous ressenti lors de la 1^{ère} itération ?
Et lors de la 2nd itération ?

Si vous deviez continuer à construire sur une de ces deux constructions, laquelle choisiriez-vous ?

Laquelle était la plus facile à construire ?
... Est-ce que refactorer vous a ralenti ?

Bonne Pratique : Refactoring & Dette technique



La dette technique est une *métaphore du développement logiciel* inventée par **Ward Cunningham**.

Just tried to explain technical debt to a customer, had to pull this out again...

Voir la traduction



Les étapes de **refactoring** permettent au développeur de travailler sur un code propre (**clean code**) et de minimiser ainsi la **dette technique**

Bonne Pratique : Refactoring & Code Legacy



Always leave the campground **cleaner** than you found it.



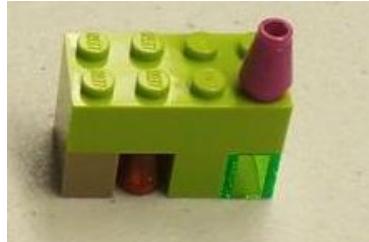
Boy scout theory

Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. *Martin Golding*

La propreté du code, une responsabilité collective



Bonne Pratique : Refactoring & TDD



Le **refactoring** contribue à travers tous les cycles du **TDD** à l'obtention d'une **conception simple et évolutive** appelée **conception émergente**



lisibilité
duplication
complexité



TDD : une méthode dirigée par les tests pour **bien concevoir** un produit (**product right**)

Good Habits Manifesto : Les bonnes pratiques du TDD sous forme de Mindmap

Source : <https://github.com/neomatrix369/refactoring-developer-habits/blob/master/02-outcome-of-collation/tdd-manifesto/tdd-good-habits-manifesto.md>

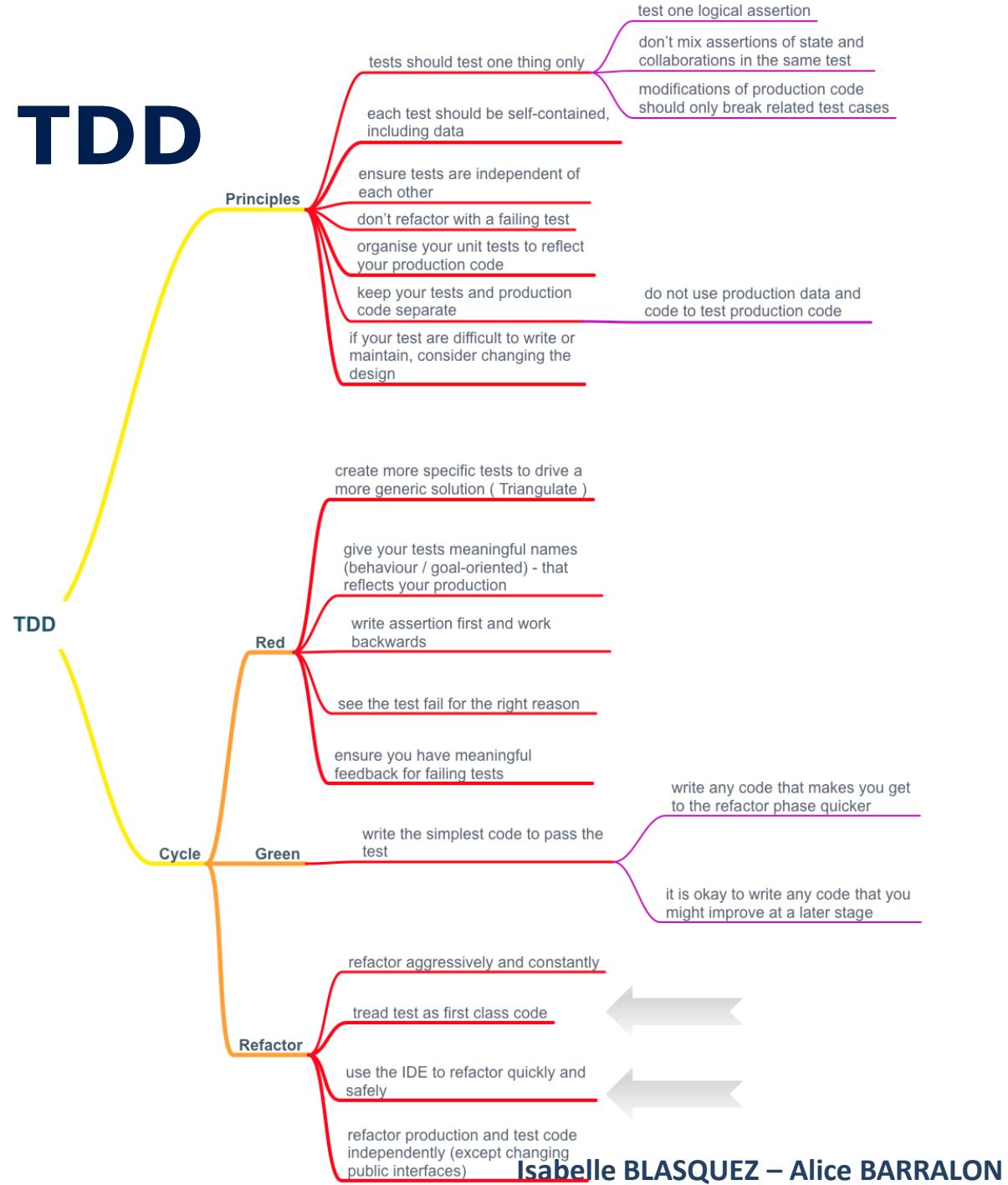
Voir aussi : <http://fr.slideshare.net/neomatrix369/refactoring-developer-habits-62785350>
<https://github.com/neomatrix369/refactoring-developer-habits>



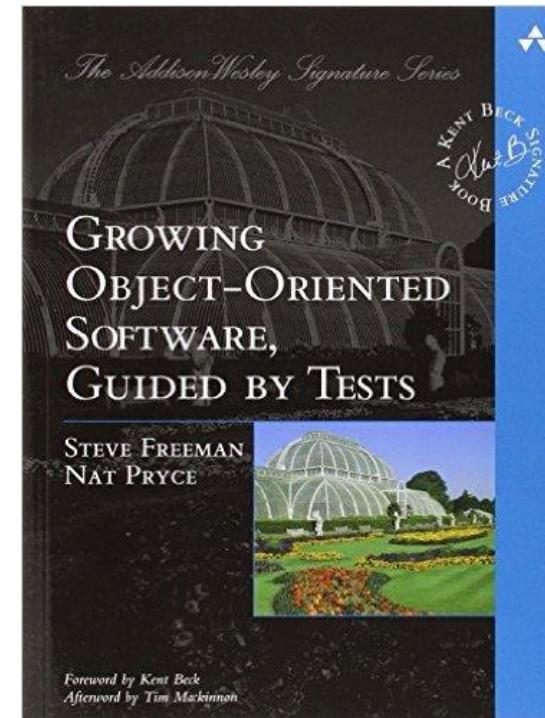
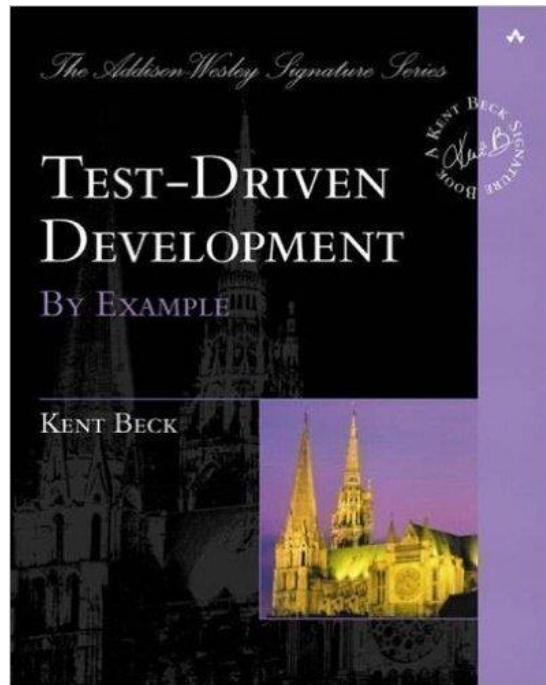
mani

neomatrix369

Passionate Java/JVM developer,
advocate to F/OSS projects. A strong
supporter of software craftsmanship
principles. Developer communities,
blogger & speaker.

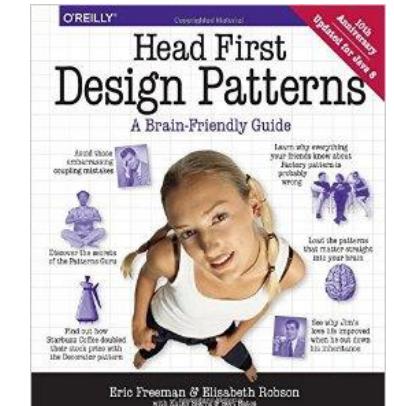
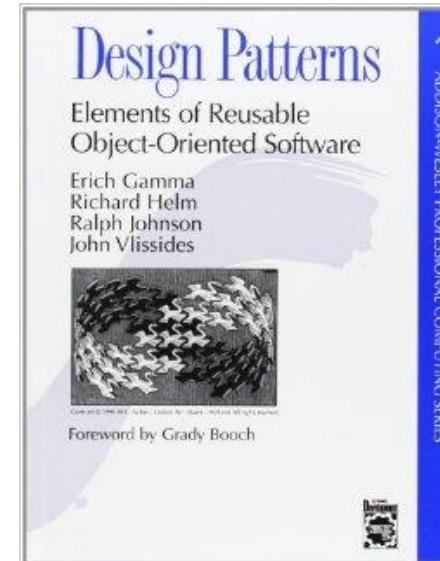
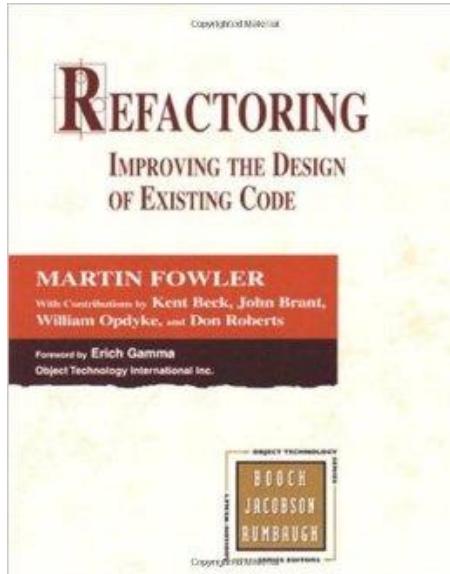
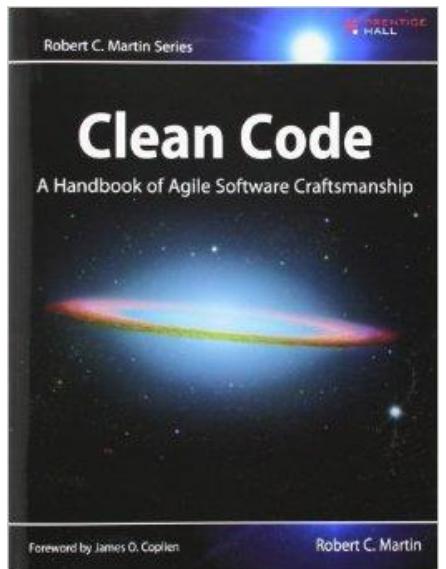
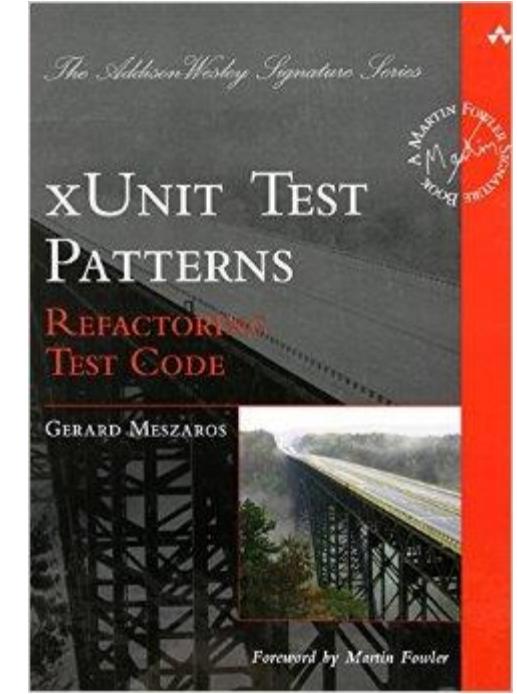
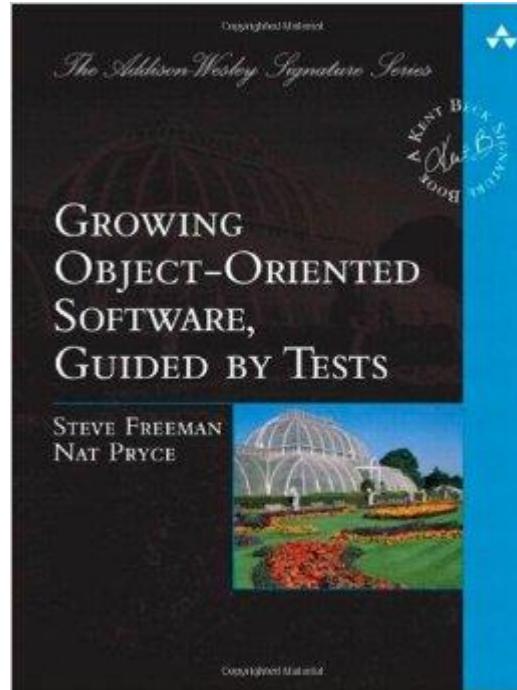
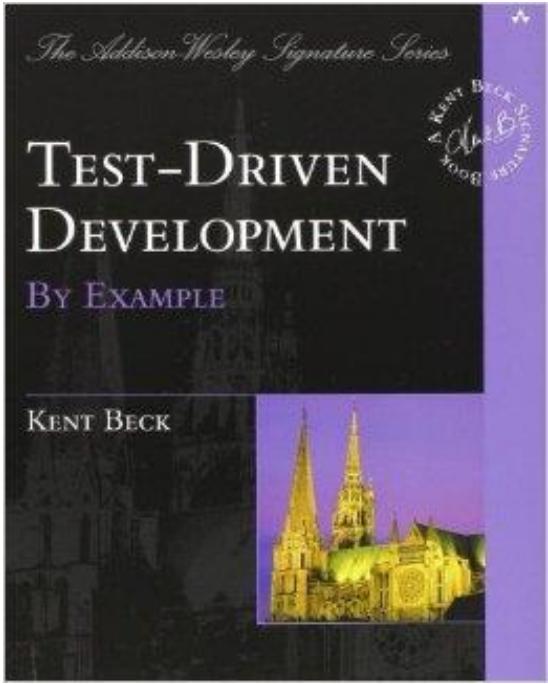


TDD : Les références



ET Confiance

TDD : une discipline de programmation ...





**Alors #TDDLego :
Entraînement ! Let's Go !**

Bonne Pratique : Pair Programming



A partir de maintenant,
vous travaillerez
en **binôme**



Extrait :
<https://twitter.com/LEGOideas/status/760814604003999744>

Pair-building pour cet exercice ...

Rôle pilote-copilote échangé à chaque itération

Le copilote écrit un nouveau test, sous forme de question sur un post-it et la place sur la table.



Refactor jusqu'à ce que la personne qui a écrit le test (copilote) soit satisfaite de l'implémentation

Le pilote construit le minimum en LEGO® pour faire passer les tests.

Itérations « baby-step » rapides
(de quelques secondes à quelques minutes)

Rappel de quelques tests déjà écrits ...

Existe-t-il une personne dans le programme ?

Existe-t-il une maison dans le programme ?

La maison est plus haute que la personne ?

La maison est plus large que la personne ?

La maison a au moins 4 briques de haut ?

La personne peut entrer dans la maison ?

La maison a une cheminée ?

...

... 1 seul test à la fois & « baby step »...

Bonne pratique !

Kata du jour :

TDD et Pair-programming

1 personne

1 animal

1 plante

1 véhicule

1 bâtiment



Débriefing:

Qu'avez-vous observé ? Qu'avez-vous ressentí ?

Qui a rempli les cinq exigences de base ?

Est-ce que tous vos tests passent ?

Revue : un petit coup d'œil sur les autres œuvres ...

Prêt pour une nouvelle mission?

Détruire vos constructions

Nettoyer votre espace de travail



Prêt pour le grand plongeon en équipe ?

Descriptif original de l'atelier disponible sur : http://www.gargoylesoftware.com/ex/lego_tdd (integrated builds)
Image https://twitter.com/LEGO_Group/status/767013378371952640

Isabelle BLASQUEZ – Alice BARRALON

Construire en Equipe ...



Une équipe de 6 personnes
en pair-building
sur une même table

Chaque table devra procéder à
une **grande construction** un peu complexe
composée de **petites constructions**
indépendantes et différentes
pouvant se connecter entre elles
(à l'image d'un zoo, un aéroport, un parc d'attraction,
un vaisseau spatial ... mais pas d'un immeuble traditionnel !)

Une équipe
=
une entreprise
avec une super idée !

Choisir votre idée (originale ?)



TOP SECRET

TO DO ...



Backlog Produit (liste des choses à faire)
Prenez quelques minutes pour identifier
quelques composants à construire



Implémentation



Chaque paire construit un composant en TDD ...



... Puis intègre sa construction au sein
du grand projet (centre de la table) et
dépose le test sur la table ...

... en vérifiant bien que tous les tests de
l'Equipe continuent à passer !

Démonstration

pair
demonstrating

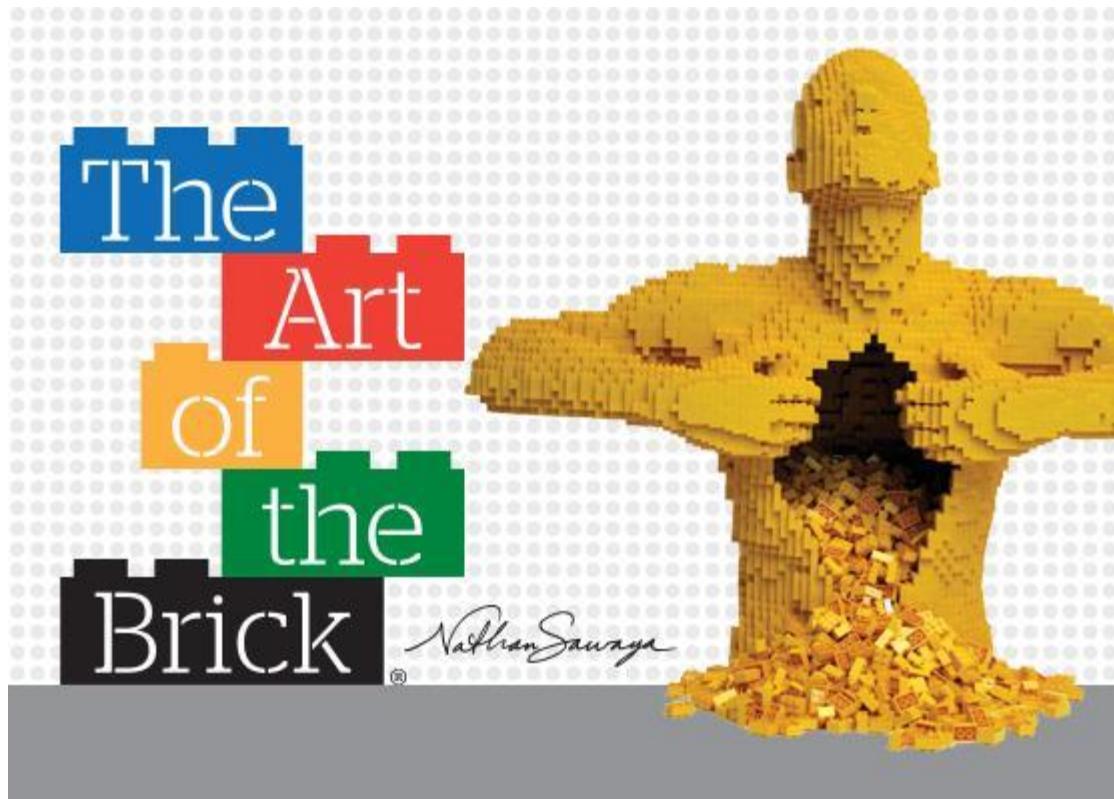


Présentation des constructions ...

... et des tests qui ont permis
d'arriver à ces
magnifiques créations !



Combien d'itérations pour arriver ces œuvres d'art ?



En savoir plus sur l'exposition **Art of the Brick** :
<http://www.parentsatoutprix.fr/the-art-of-the-brick-avis/>

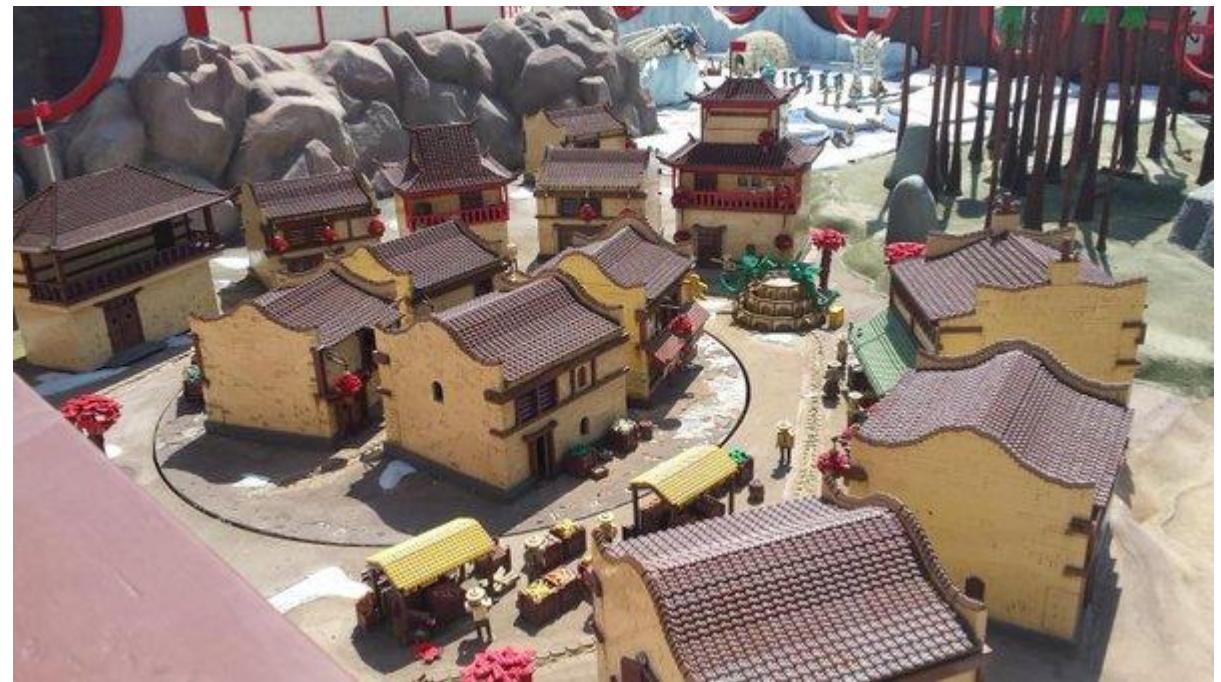


Photo prise à



Bilan & Rétrospective

Qu'avez-vous ressenti ?

Qu'avez-vous appris ?

Que mettrez-vous en application dès lundi ?

Que vous ayez

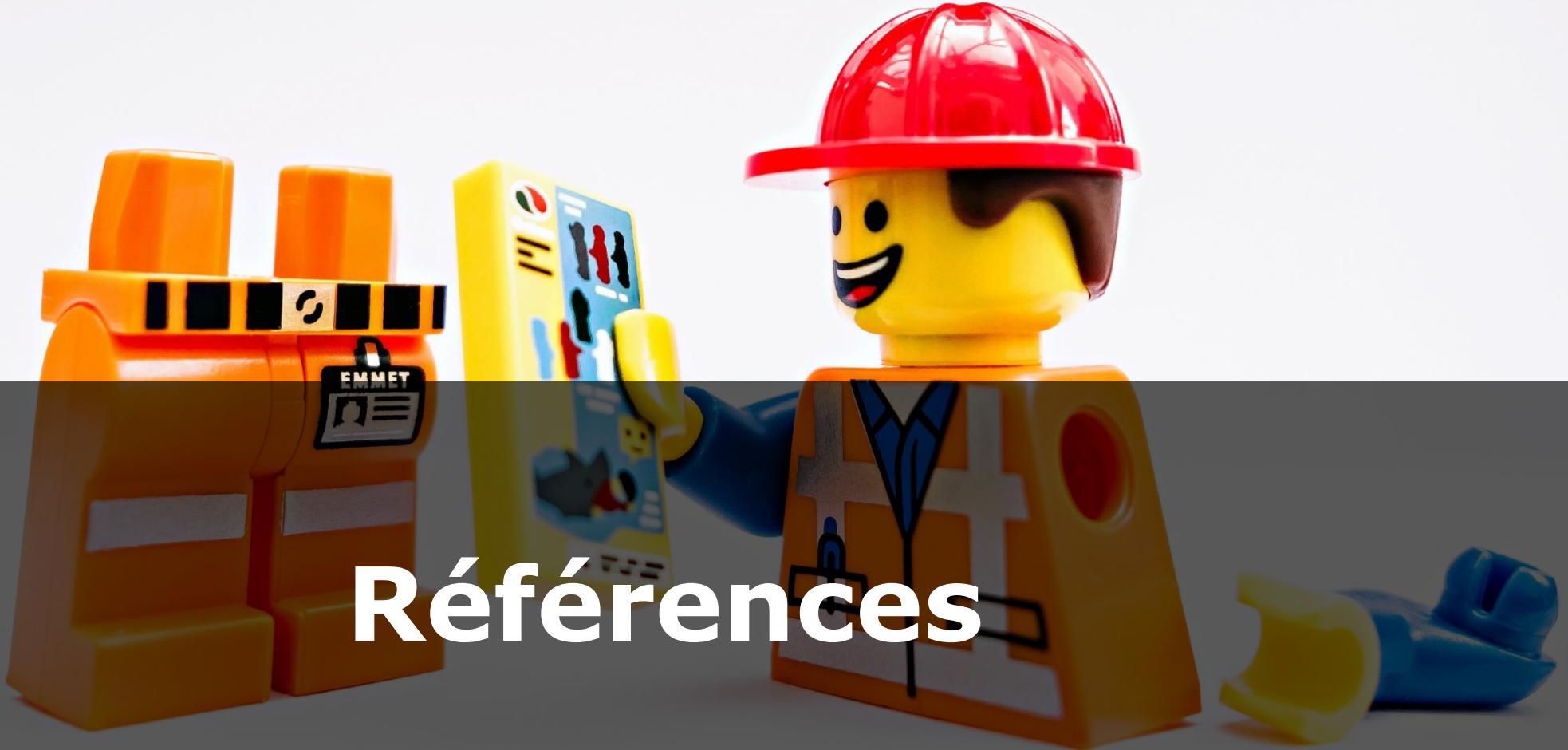


ou pas ...

N'oubliez pas de remplir le sondage. Merci !



Références



WebOGRAPHIE (1/2)

Descriptif original des ateliers “About the LEGO® technical practices exercises” :

<http://www.gargoylesoftware.com/ex>

Transparents : TDD and Refactoring with LEGO at Agile2013 :

<http://fr.slideshare.net/BillyGarnet/tdd-and-refactoring-with-lego-agile2013-25143447>

<http://fr.slideshare.net/BillyGarnet/tdd-and-refactoring-with-lego>

Vidéo : Enregistrement d'un atelier : TDD and Refactoring with LEGO by [Bryan Beecham](#) (2014)

<https://www.infoq.com/presentations/tdd-lego>

Video teasing Lego TDD & Refactoring ATT :

https://www.youtube.com/watch?v=w7oupm_2KH0

Photos: <https://flic.kr/s/aHsjPgKJAk>

sur le compte de <https://www.flickr.com/photos/livingjuicy> (album NYC Scrum Users Group Dec. 2013)

WebOGraphie (2/2)



Articles :

Using Lego to Teach Technical Practices (Article original) :

<https://www.infoq.com/news/2016/05/lego-teach-technical>

Des Lego pour Apprendre les Pratiques Techniques (version française) :

<https://www.infoq.com/fr/news/2016/06/lego-teach-technical>

A lire également :

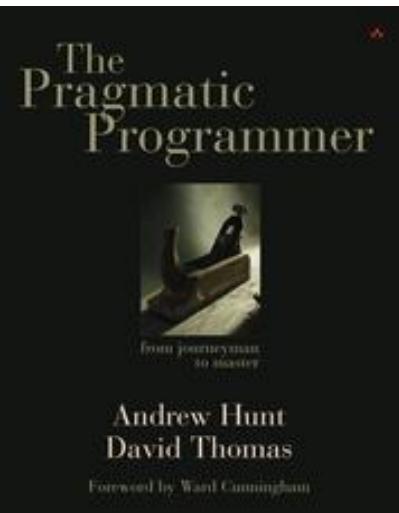
Intro to Refactoring with LEGOs : <http://agilecomplexificationinverter.blogspot.fr/2013/12/intro-to-refactoring-with-legos.html>

<http://www.boost.co.nz/blog/2013/11/what-does-lego-and-building-high-quality-software-have-in-common/>

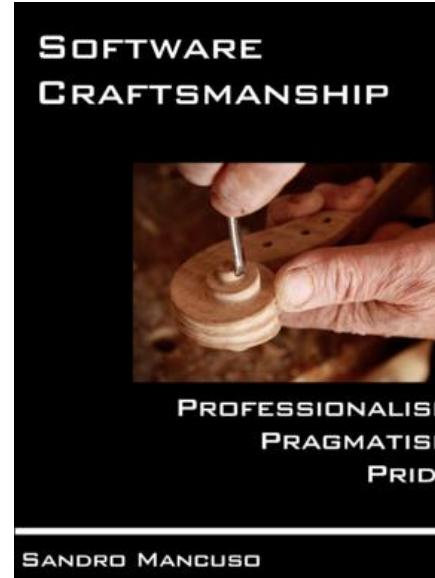
The ROI of Refactoring: Lego vs. Play-Doh :

<http://fr.slideshare.net/NeilGreen1/the-roi-of-refactoring-lego-vs-playdoh>

En savoir plus sur le Software Craftsmanship



[https://pragprog.com/book/tpp/
the-pragmatic-programmer](https://pragprog.com/book/tpp/the-pragmatic-programmer)



<https://leanpub.com/socra>



techtrends.xebia.fr



<http://www.octo.com/fr/livres-blancs>

Isabelle BLASQUEZ – Alice BARRALON

A méditer ...



Gilles Roustan

Artisan développeur Web



» Blog

» A propos de moi

Artisan développeur

12/06/2016

Avant-propos

Cet article est une retranscription de la conférence que j'ai donnée lors de l'agile tour de Montpellier le 13 octobre 2015. Pour mieux suivre, je vous conseille de regarder en parallèle les [slides de la conférence](#).

A propos de moi

Je m'appelle Gilles et je suis artisan développeur.

Dans cet article, je vais vous parler de moi, de mon métier et de ma famille.

Plus particulièrement de l'évolution de la perception de mon métier et comment le métier de mon père et de mon grand-père m'ont aidé à devenir un développeur plus heureux.

La mode du software craftsmanship

Il n'y a pas si longtemps, j'ai changé de poste et quand j'ai fait mon CV, j'ai mis que j'étais **artisan développeur**.

Il y a quelques années, j'ai vu débarquer cette *mode* lors de [conférences](#), dans des articles sur Software Craftsmanship.

J'ai trouvé ça sympa comme titre : artisan développeur ! Un contraste entre du moderne et du traditionnel. Et moi comme un mouton, j'ai trouvé ça cool et je me suis dépêché de modifier mon profil twitter.

How Can You Say You're A Software Crafts(wo)man?

Or what makes you a Software Crafts(wo)man

This is a question that came up yesterday during our latest meetup with our local French Riviera Software Craftsmanship Community (note: a very nice round table session on the beach of Juan Les Pins ☺).

This post is my answer to this question.

What Software Craftsmanship is not

It's not an elitist club of people who think to hold the truth of how to write the perfect code. This is an important aspect because it is related to how much inclusive or exclusive a group of people is (and also to how much nice people are). I've been following the international Software Craftsmanship movement since some time now and I met some of them during SoCraTes UK and I learned how much important are things like kindness, mentorship and empathy for a Software Crafts(wo)man. Software development is not only techniques and programming: it's also relationship and communication with your peers.

It's not a certification of one's capabilities to write good software (assuming it does exist really). A person yesterday was doubtful whether he could call himself a *software craftsman* or not because he is not working in TDD or applying all the time the SOLID principles.

These are technicalities, very important ones, but they are just tools that a person who cares about what he does can use or not in his daily job.

It's not in conflict with the Agile movement. Software Craftsmanship is focused a lot more on technical practices and writing quality software, but the goal is always the same: deliver values for the customer. The only difference is

Article complet disponible sur : <https://medium.com/alebaffa-blog/how-can-you-say-youre-a-software-crafts-wo-man-48ebc055ba9d#.byqyxzdjx>