



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



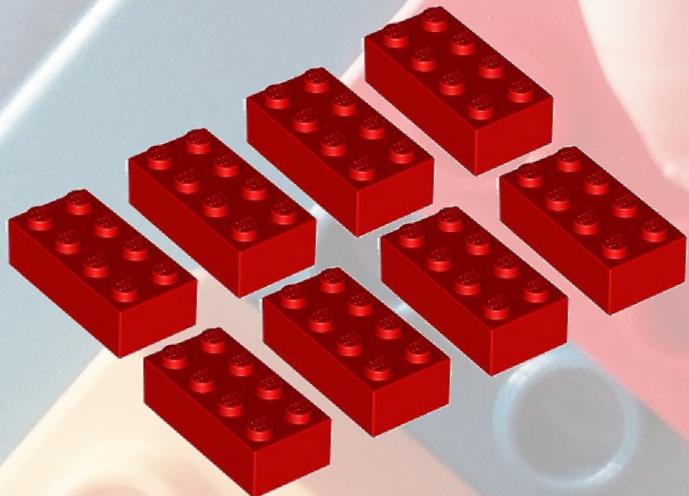
*Isabelle BLASQUEZ
@iblasquez*

Septembre 2016

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

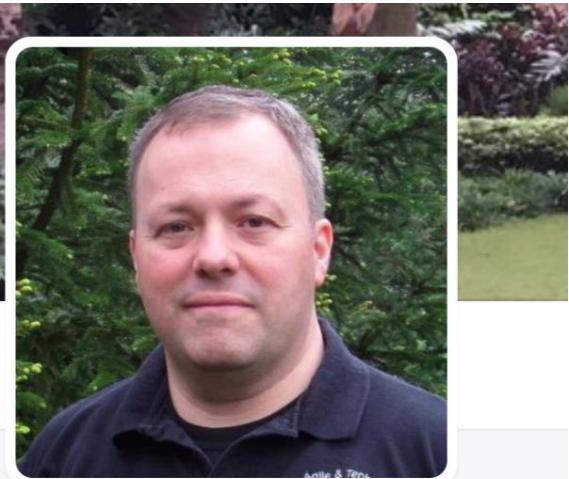
Il faut jouer pour devenir sérieux. Aristote

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.

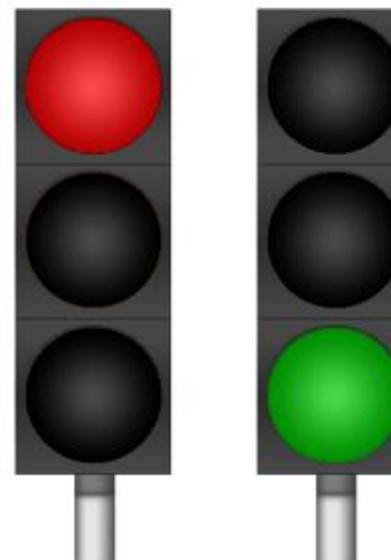
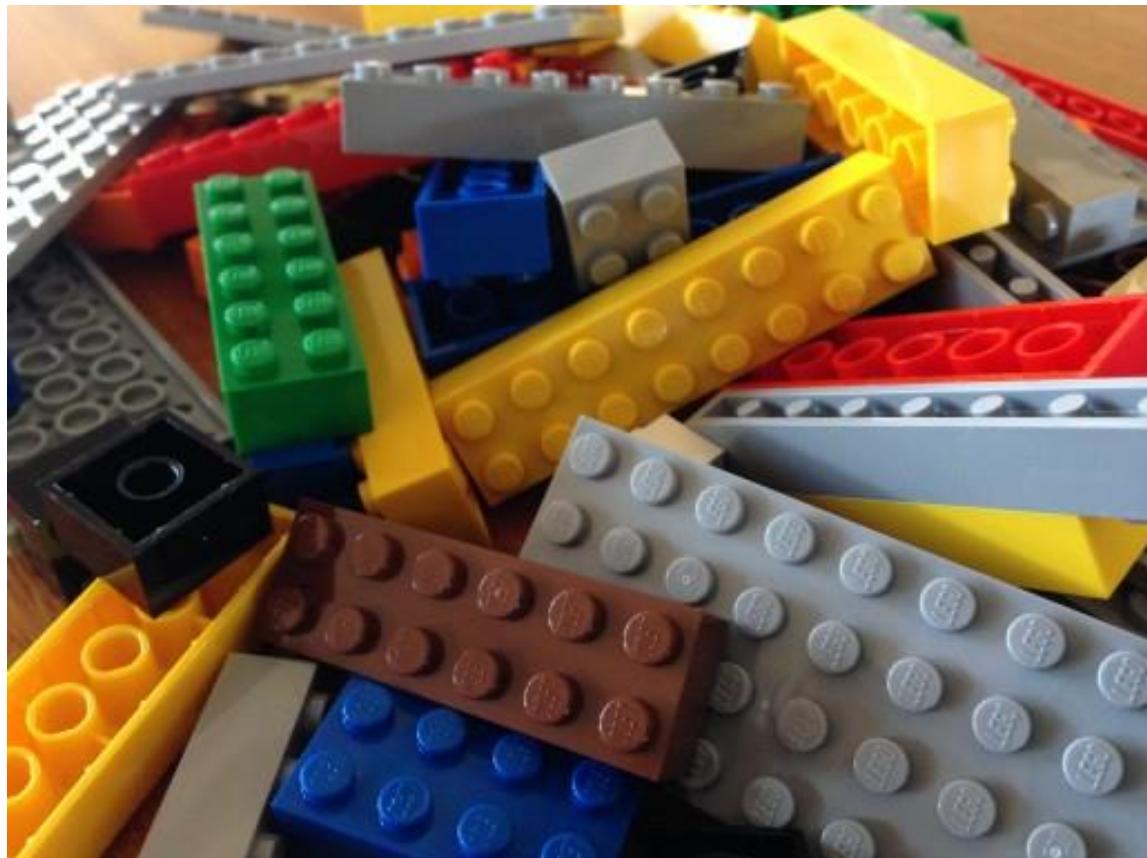
📍 Canada

🔗 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/>

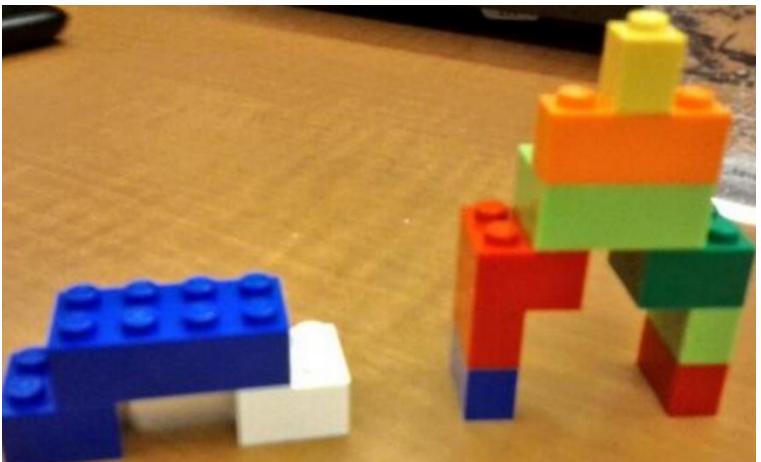
Simple, pour commencer ...



Construire une maison et une personne



Admirez et partagez !



 #TDDLego

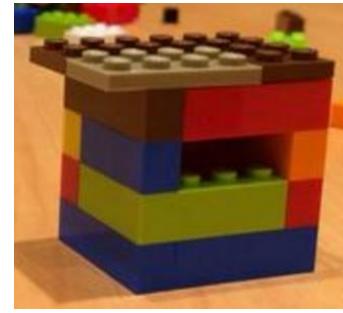


Extraits de : <https://twitter.com/sahithyaW/status/428217265667198976>
<https://twitter.com/JeffHoover/status/515550469175799808>
<https://twitter.com/iGeaux/status/515550799045607424>
<https://twitter.com/iGeaux/status/515554621629415424>

Valeur métier



ET



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

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

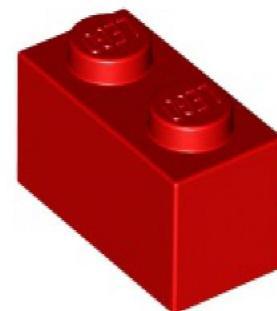
Pièces disponibles sur : <http://shop.lego.com/>

Isabelle BLASQUEZ - 2016

Bonnes Pratiques : deux principes indispensables

KISS

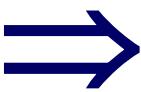
Keep It Simple Stupid



YAGNI

You Aren't Gonna Need It

Une livraison KISS pour notre problème ...



Quid du programme informatique parfait ?

Nettoyez votre espace de travail !

Espace vide = programme parfait

Un programme parfait ?

Quid du programme parfait ? (quelques éléments de réponse)



Il marche !!!
Pas de bugs

Satisfait toutes
les exigences

Très facile
à étendre

Très facile
à comprendre
et à maintenir

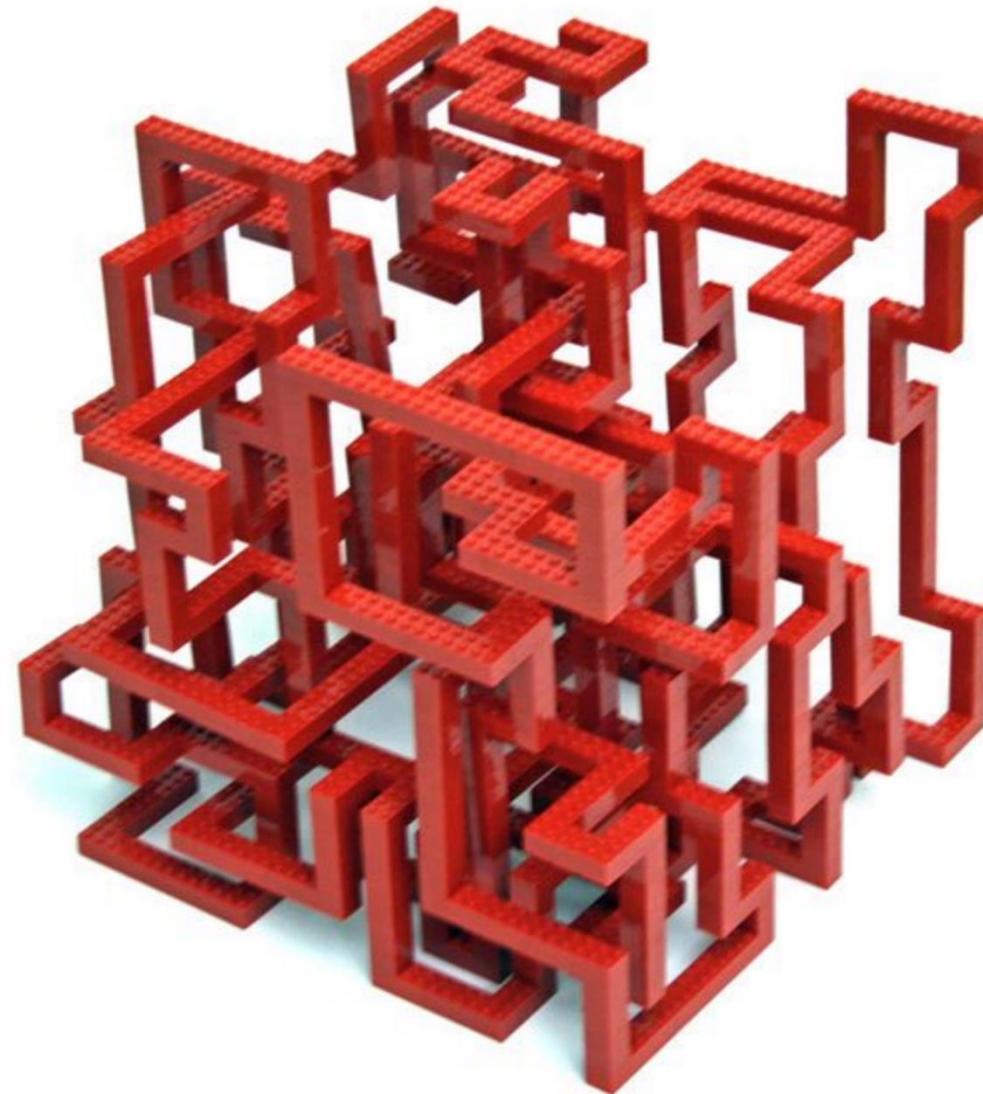
...

**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 - 2016

Bonne Pratique : Les règles de simplicité (eXtreme Programming)



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

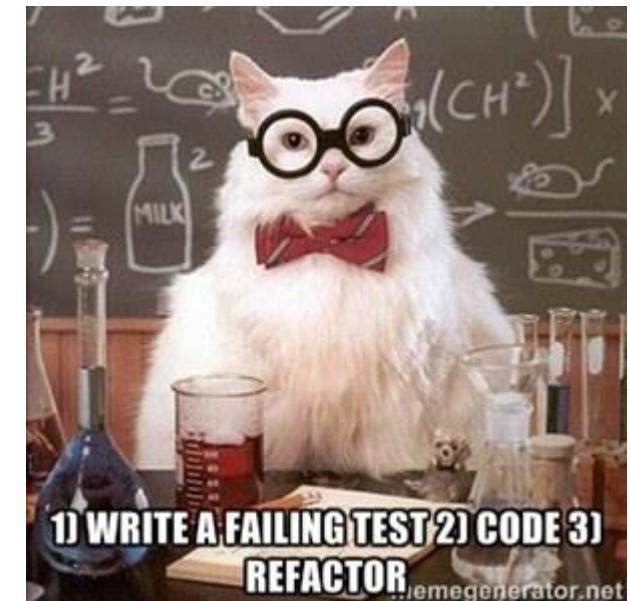
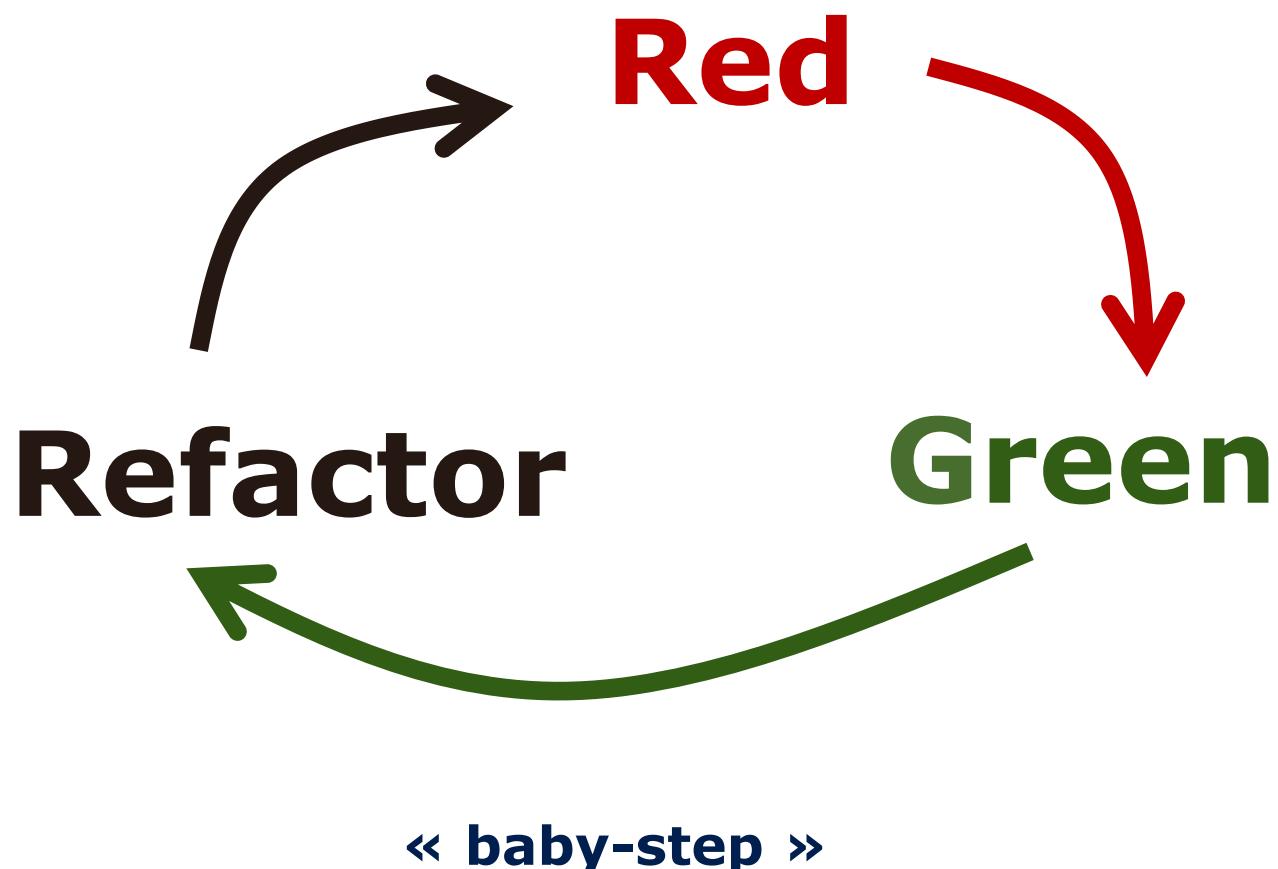
A lire également : <http://martinfowler.com/bliki/BeckDesignRules.html>
et <https://leanpub.com/4rulesof simplesdesign>



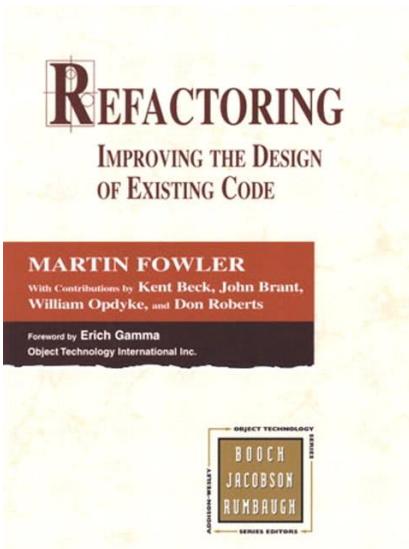
Test Driven Development (TDD) : à la Rescousse !

Test Driven Développement

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



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

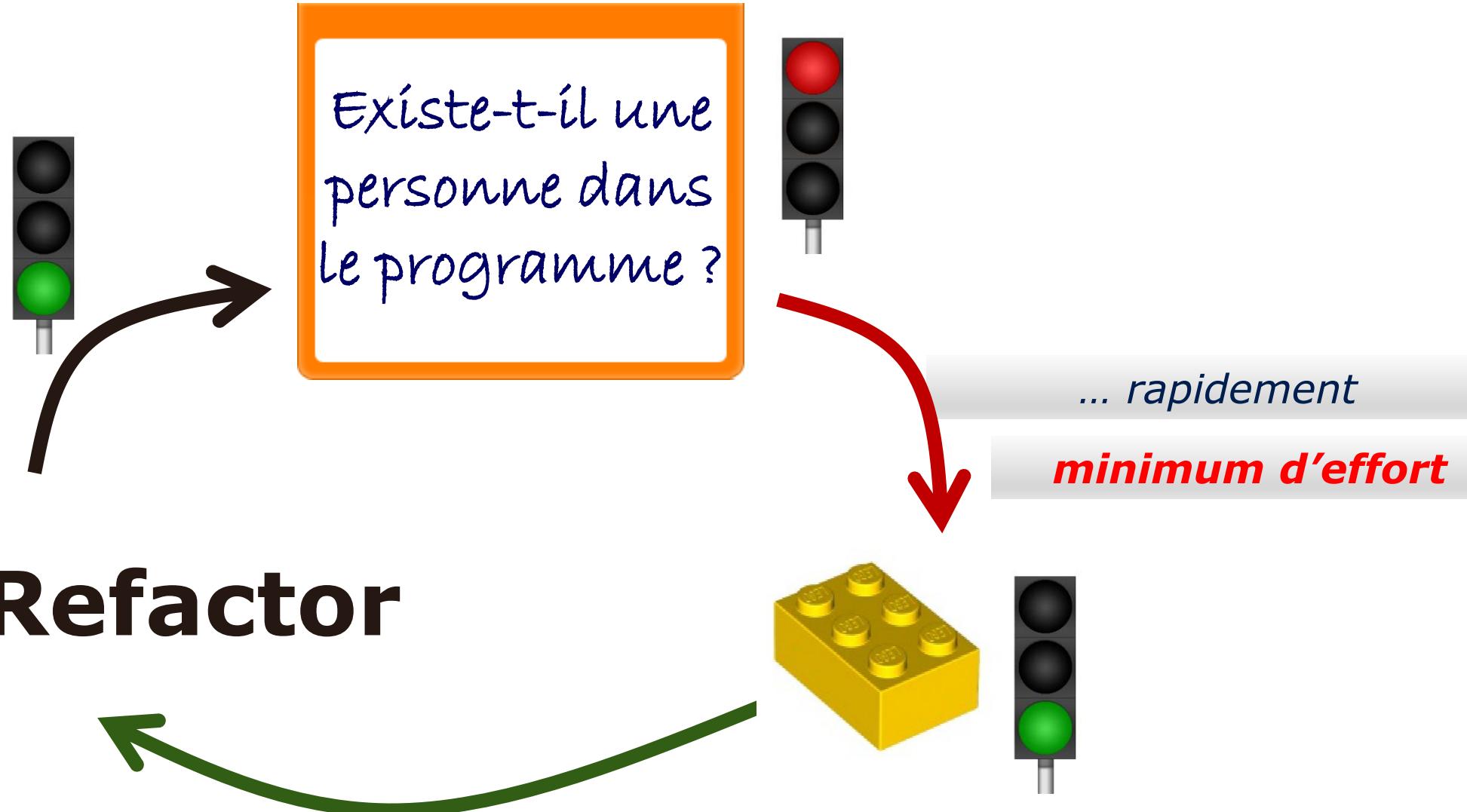
Pas de code de production sans test !

**Un nouveau test =
un nouveau comportement**

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

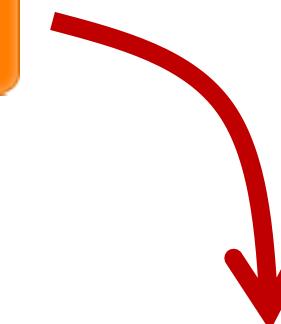
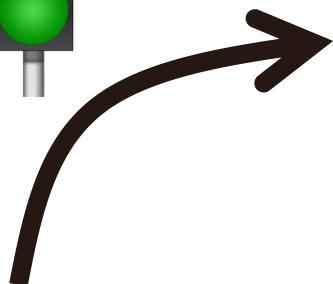
Let's go ! Itération 1

Commencer par un test en échec

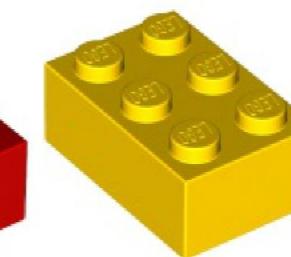
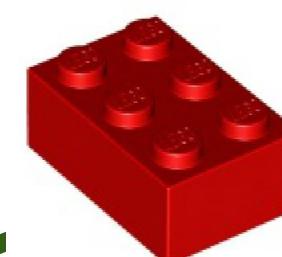


Let's go ! Itération 2

Carton vert !



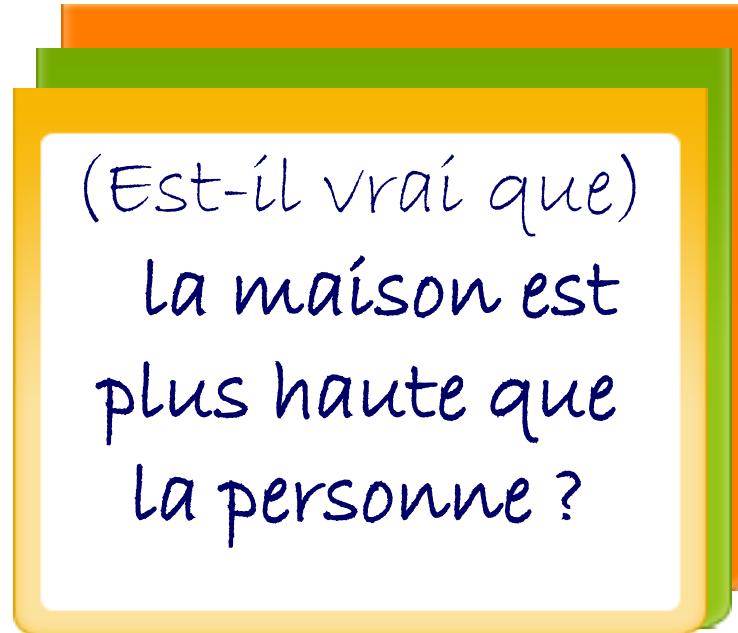
TOUS les tests



Refactor



Choix du prochain test ...



ou



?

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



Nouveau test = nouveau comportement

Let's go ! Itération 3

Carton vert !



La maison est
plus haute que
la personne ?



Refactor

Green

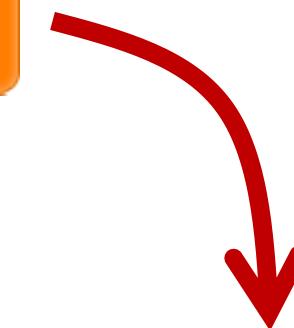
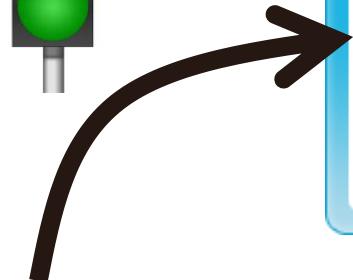


Let's go ! Itération 4

Carton vert !



La maison est
plus large que
la personne ?



Refactor

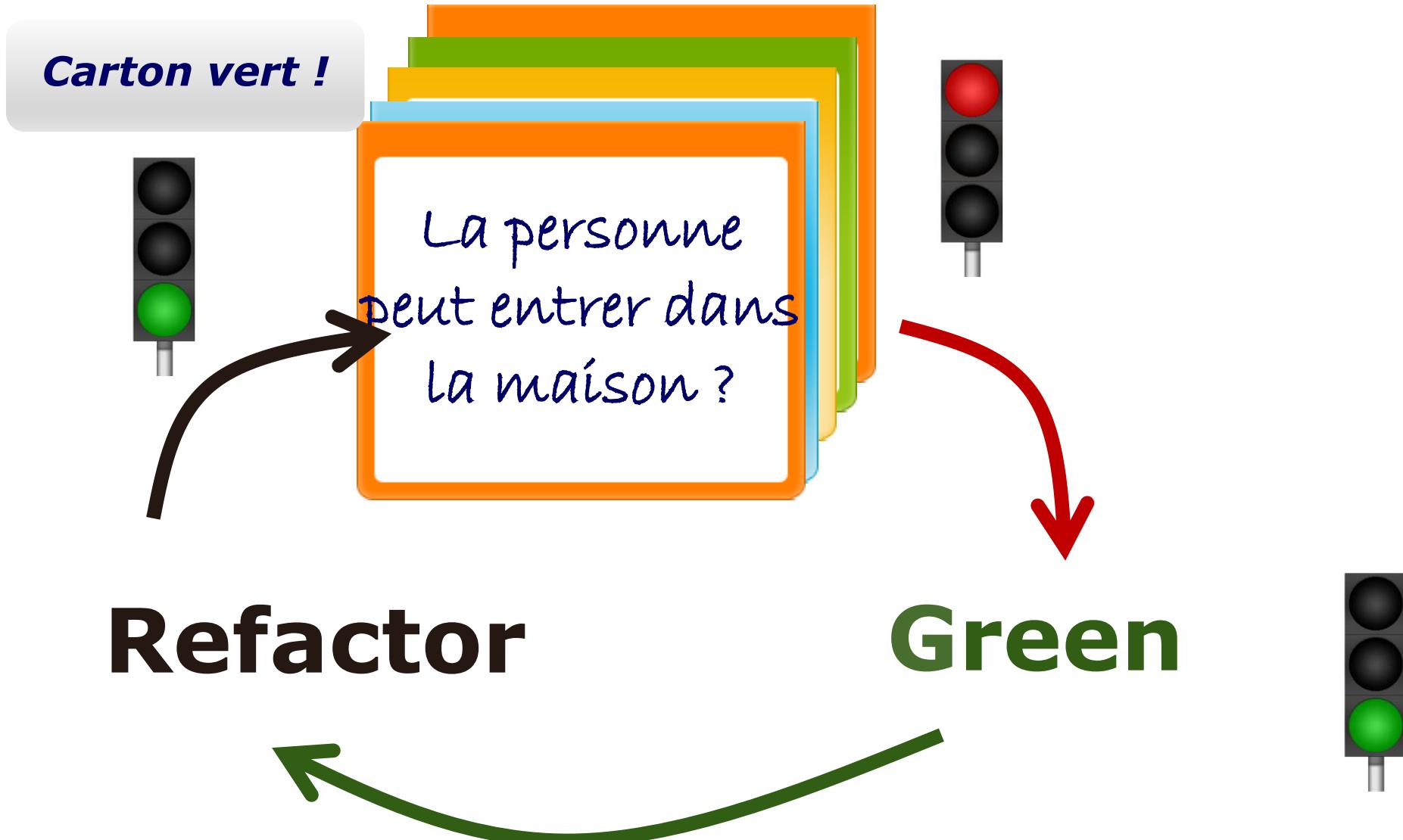
Green



A chaque nouveau test, nous apprenons de plus en plus ...

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

Let's go ! Itération 5 (Interaction des objets)



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 - 2016

Instructions



Une exigence toutes les 30 secondes



REFACTORING INTERDIT !!!
On ne touche plus aux briques utilisées

Exigence n°1



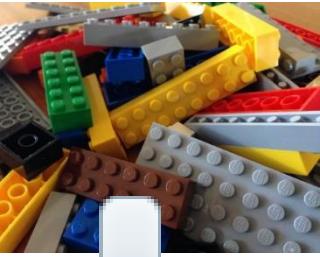
30 secondes

Exigence n°2



30 secondes

Exigence n°3



La maison doit être entièrement fermée.

Ajouter également des murs intérieurs

30 secondes

Exigence n°4



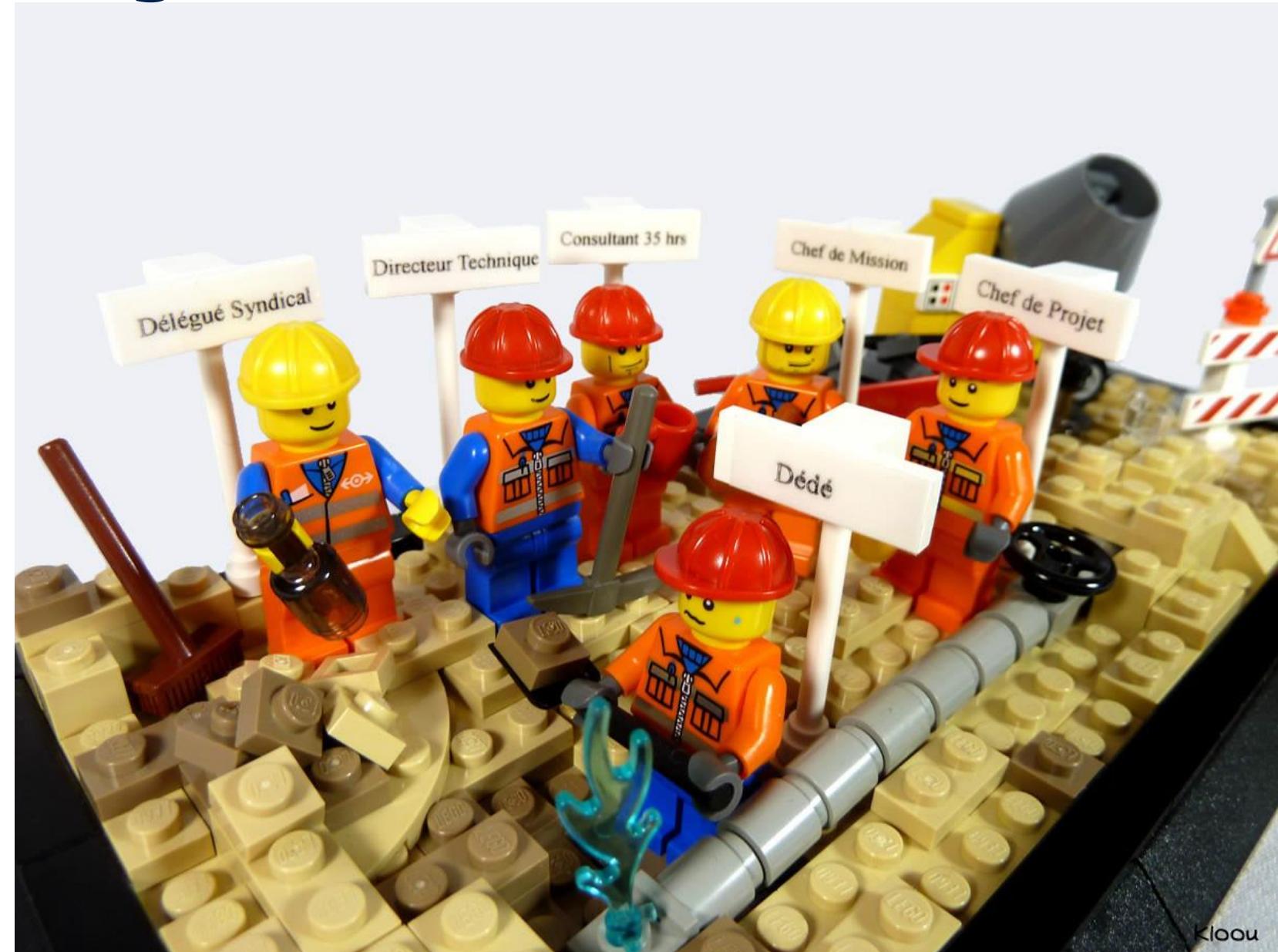
30 secondes

Exigence n°5



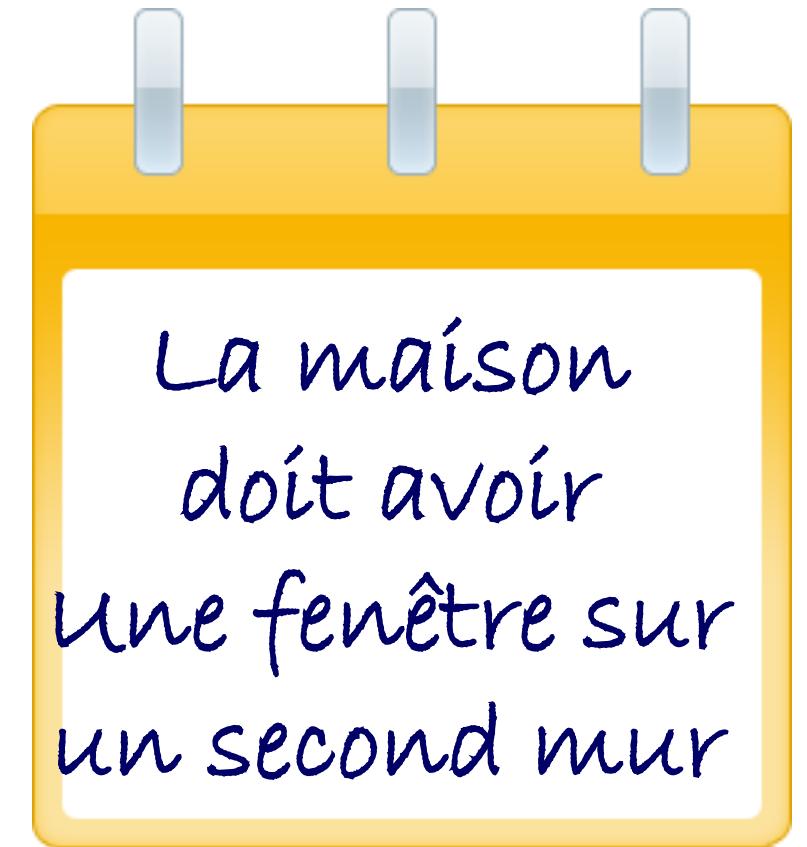
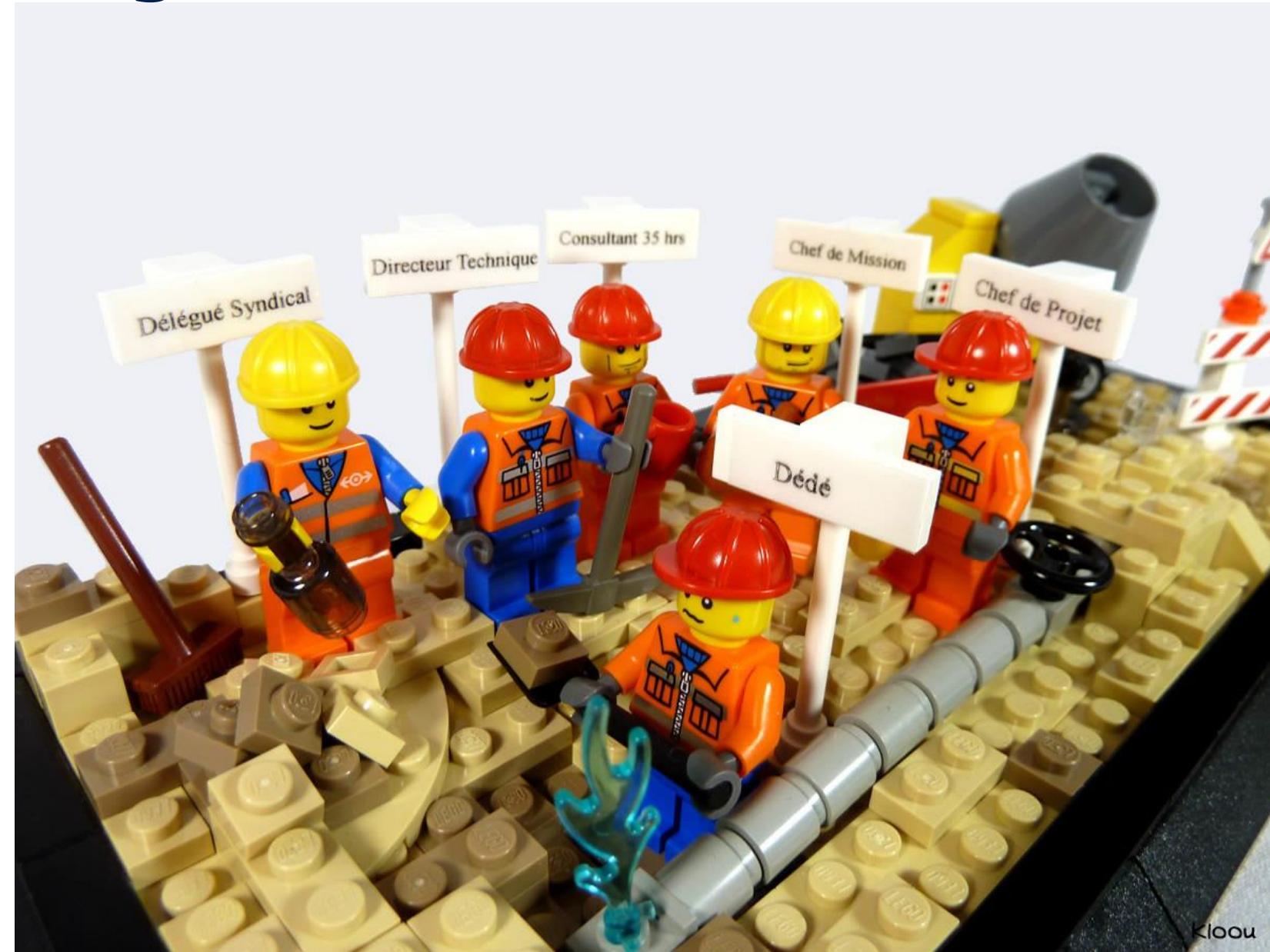
30 secondes

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 !)

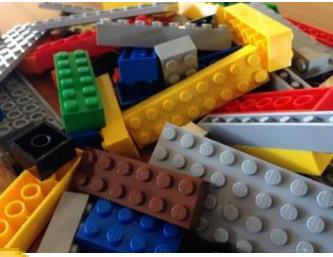


*La maison de
mes rêves :
C'est reparti !*

Contrainte levée :
REFACTORING AUTORISE !

⇒ **Le plus simplement possible**
(KISS)

La maison de mes rêves (refactoring autorisé)

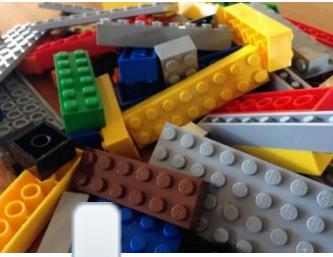


La maison
doit avoir
deux murs
connectés

La maison doit être
entièrement fermée.

Ajouter également
des murs intérieurs

La maison de mes rêves (refactoring autorisé)

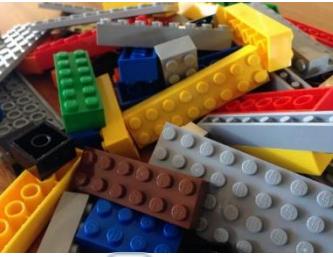


La maison
doit avoir
un toit en pente

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

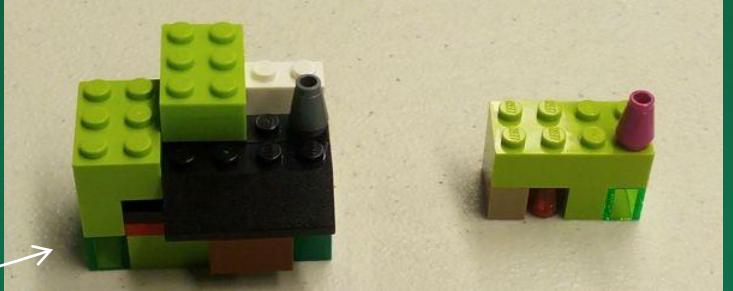
La maison
doit avoir
maintenant
un toit plat



Débriefing:

Comparez

Dette accumulée



La plus facile à construire ?

Refactorer ralentit-il ?

Bonne Pratique : Refactoring & Dette technique



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 la **dette technique**

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



La propreté du code, une responsabilité collective





**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>

TDD en pair-building pour cet exercice ...

Rôle pilote-
copilote

Le copilote écrit un nouveau test,



Refactor

**Le pilote construit
en LEGO®**

Bonne pratique !

Kata du jour :

TDD et Pair-programming



1 personne

1 véhicule

1 pont

Rien de doit être construit
sans test écrit au préalable !!!



#TDDLego

Débriefing:

Qu'avez-vous ressentí ?

Quid des trois exigences ?

Tous les 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

Construire en Equipe ...



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

Une équipe
=
une entreprise
avec une super idée !

Une **grande construction** composée de
petites constructions indépendantes
(zoo, un aéroport, un parc d'attraction...)

Choisir votre idée (originale ?)



TOP SECRET

TO DO ...



Backlog Produit



Implémentation



Construction des composants : TDD pair-building



Rien de doit être construit
Sans test écrit au préalable !!!

et intégration au fur et à mesure des
composants au centre de la table avec les tests !

Démonstration

pair
demonstrating



#TDDLego

Présentation des constructions ...

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

TDD : Une pratique de conception ?



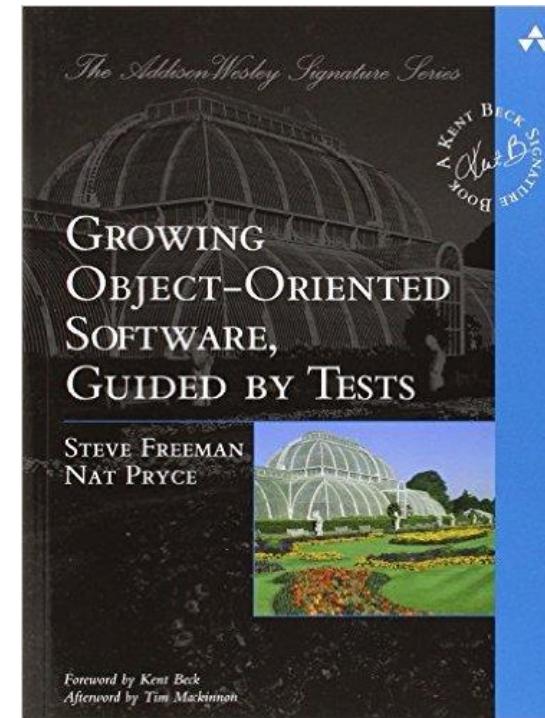
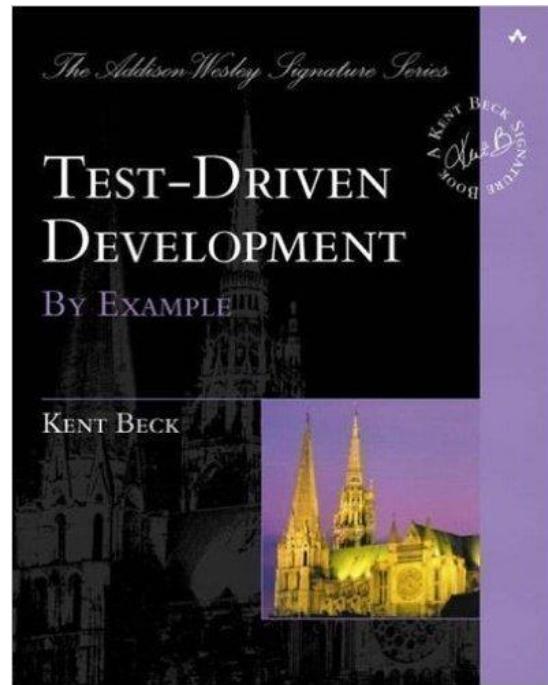
*The act of writing a unit test
is more an act of design than of verification.
It is also more an act of documentation than of verification
(Robert. C. Martin)*

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

Test Driven

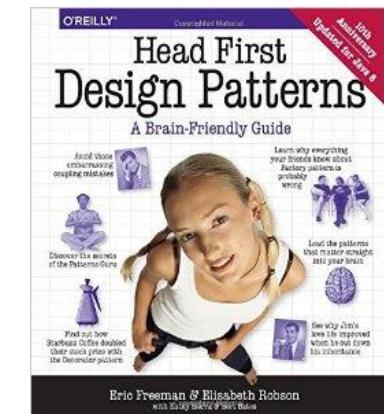
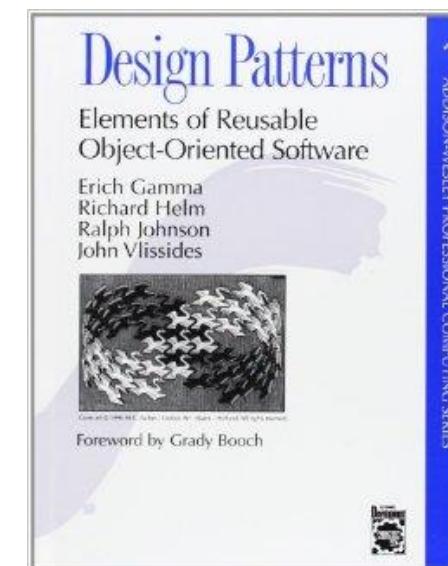
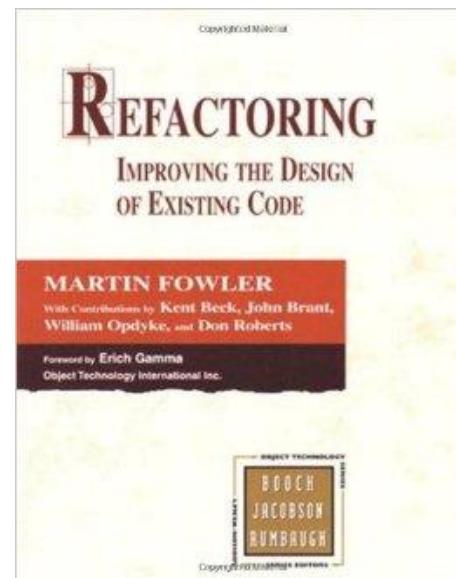
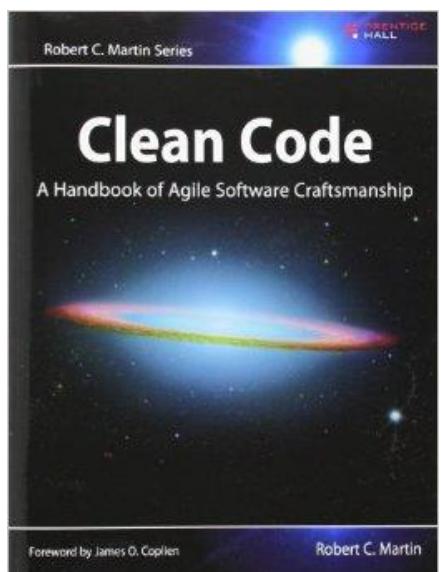
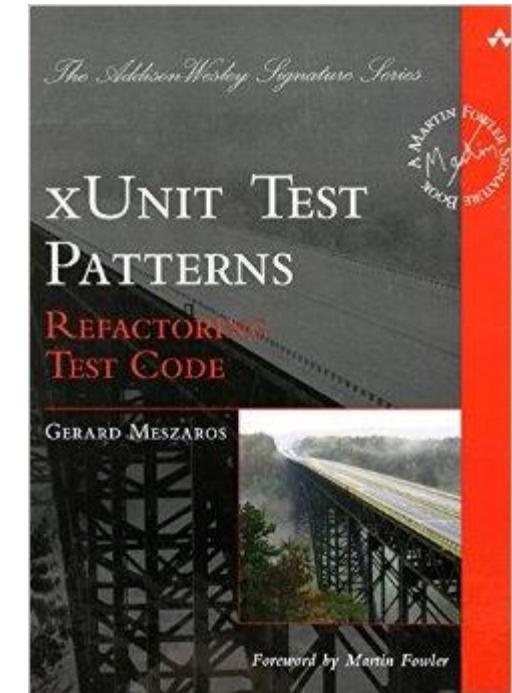
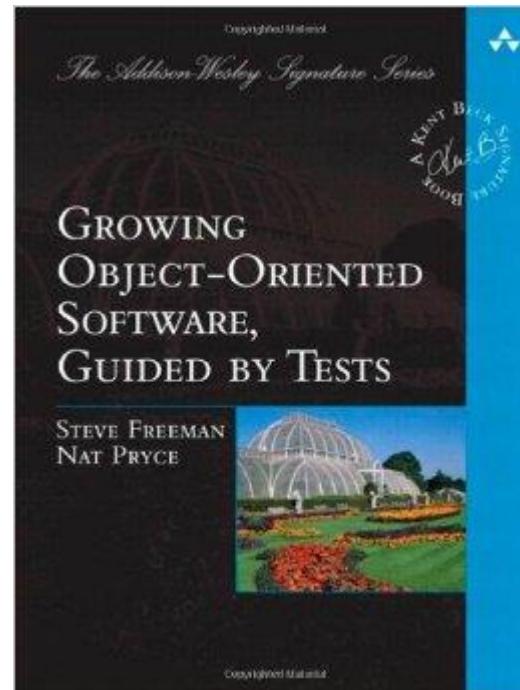
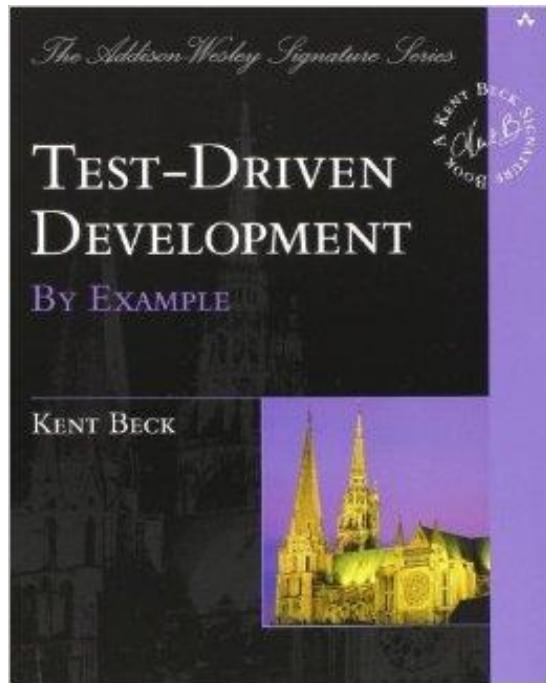
**Developpement
Design**

TDD : Les références



ET Confiance

TDD : une discipline de programmation ...



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>

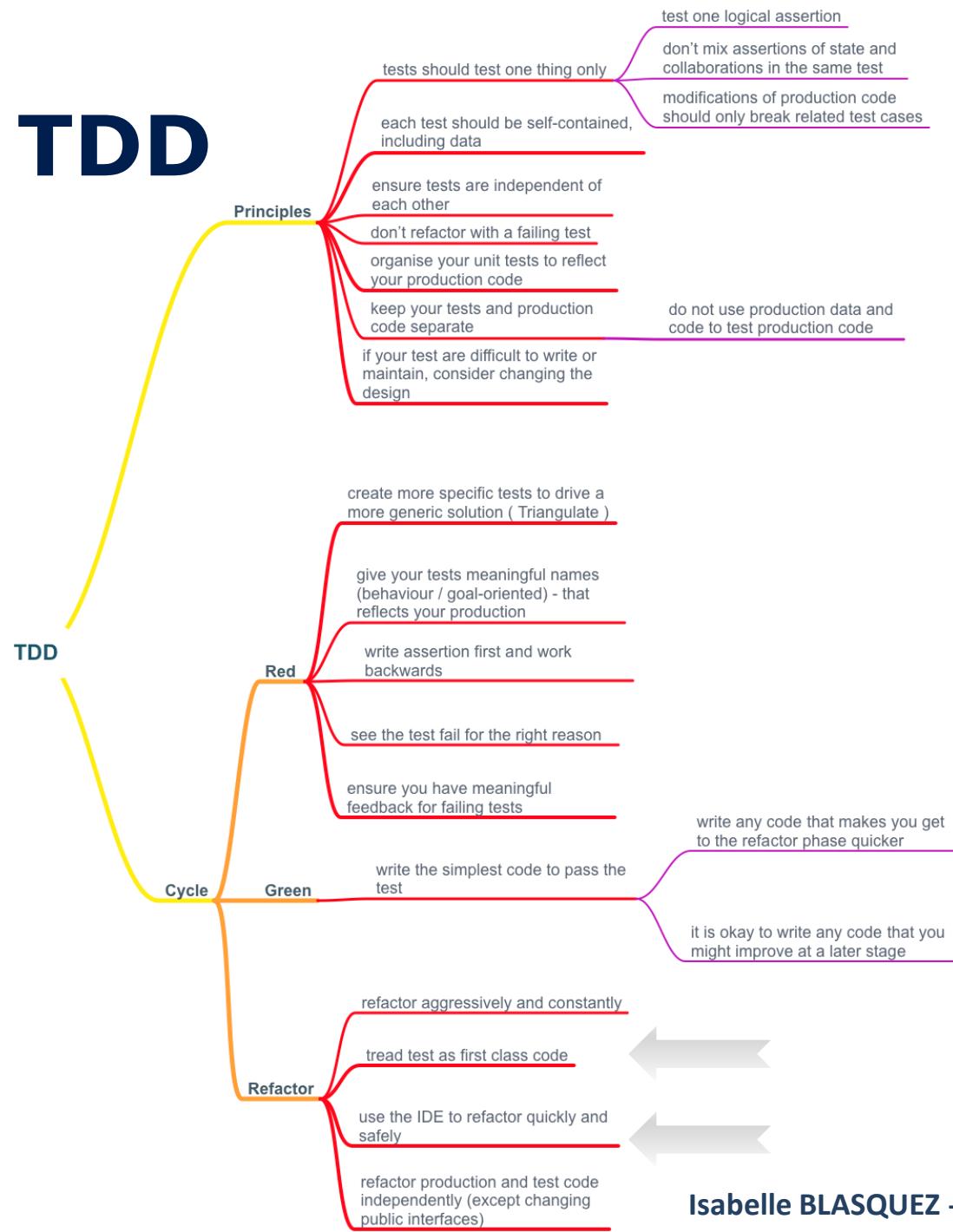
Voir aussi : <http://fr.slideshare.net/neomatrix369/refactoring-developer-habits-62785350>
<https://github.com/neomatrix369/refactoring-developer-habits>



mani

neomatrix369

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



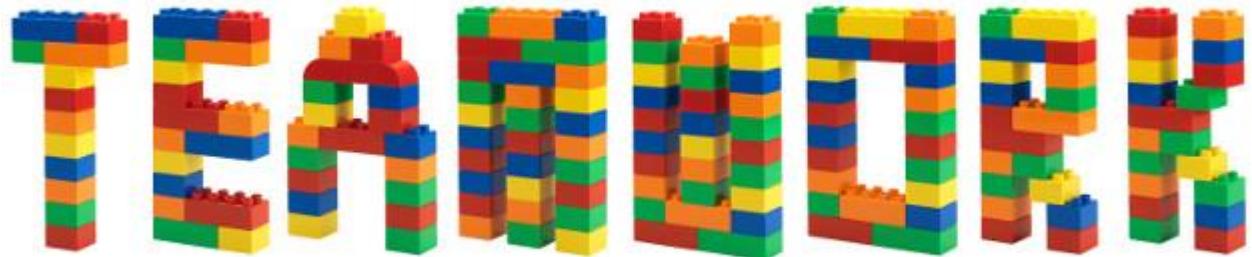


Intégration continue en LEGO®

Descriptif original de l'atelier disponible sur : http://www.gargoylesoftware.com/ex/lego_continuous_integration
Image <http://www.cloveretl.com/blog/new-features-cloveretl-4-0/>

Isabelle BLASQUEZ - 2016

1 projet commun pour le groupe ...



5 équipes
1 équipe/table



En vacances dans une destination exotique,
nous souhaitons envoyer des cadeaux
à nos proches par voie maritime ...

Votre mission : Construire un système d'acheminement de cadeaux disposant des composants suivants :

Equipe 1

construire
les **cadeaux** à ramener

Equipe 2

construire
le conteneur
dans lequel seront
déposés les cadeaux

Equipe 3

construire
le camion qui contiendra
le conteneur

Equipe 4

construire la **grue**
qui permettra de soulever
le conteneur du camion
pour le déposer
à bord du navire.

Equipe 5

construire
le navire
qui acheminera
le conteneur

Itération 1



Intégration Time : Debriefing ...



Est-ce que les cadeaux rentrent dans le conteneur ?

Est-ce que le conteneur est adapté à la taille du camion ?



Est-ce que la grue est adaptée à la taille du conteneur ?

Que dire sur le navire ?

Rétrospective

Découvrir ce qu'on a fait de bien

Comprendre ce qui n'a pas marché

Trouver des moyens pour s'améliorer

Échanger, communiquer

Qu'est-ce qui a bien fonctionné et qu'il faut garder ?



Qu'est-ce qui doit être amélioré ?

Itération 2



Mission Identique :

Construction d'un système d'acheminement de cadeaux

Equipe 1

construire
les **cadeaux** à ramener

Equipe 2

construire
le conteneur
dans lequel seront
déposés les cadeaux

Equipe 3

construire
le camion qui contiendra
le conteneur

Equipe 4

construire la **grue**
qui permettra de soulever
le conteneur du camion
pour le déposer
à bord du navire.

Equipe 5

construire
le navire
qui acheminera
le conteneur

Intégration & Rétrospective

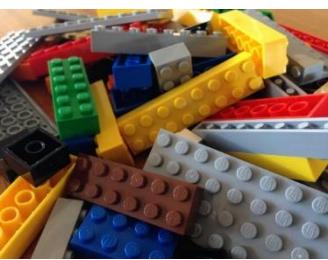


**Quels composants
ont été livrés ?**

**Attention !!!
Si les cadeaux n'arrivent pas jusque dans le navire,
votre livraison n'apporte aucune valeur métier ...**



Itération 3



Dernière chance
de livrer !!!

Si itération précédente réussie, essayez d'optimiser le process !



#TDDLego

Débriefing:

Qu'est-ce qui vous a surpris ?

Comment s'organiser pour offrir des cadeaux au plus tôt ? c-à-d livrer au plus tôt de bout en bout tous les composants.

Qu'avez-vous mis en place pour vous améliorer ?

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.**

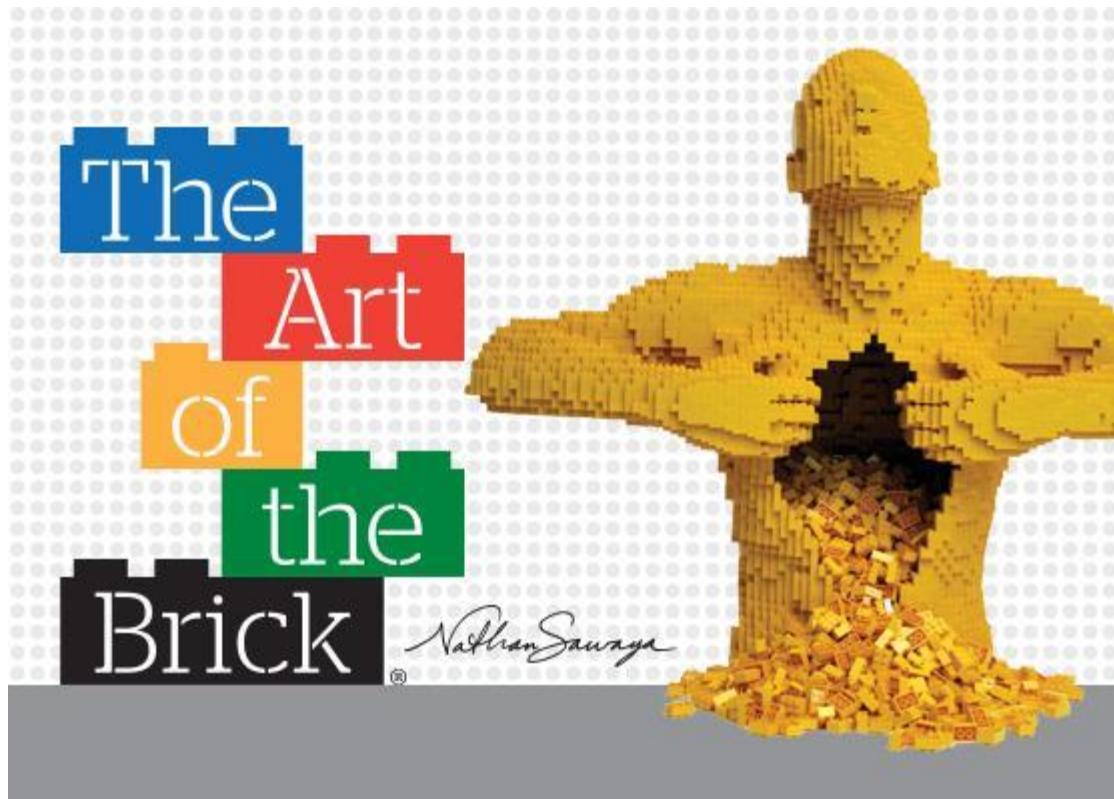
Pas seulement les individus et leurs interactions,
mais aussi **une communauté de professionnels.**

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.

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/>

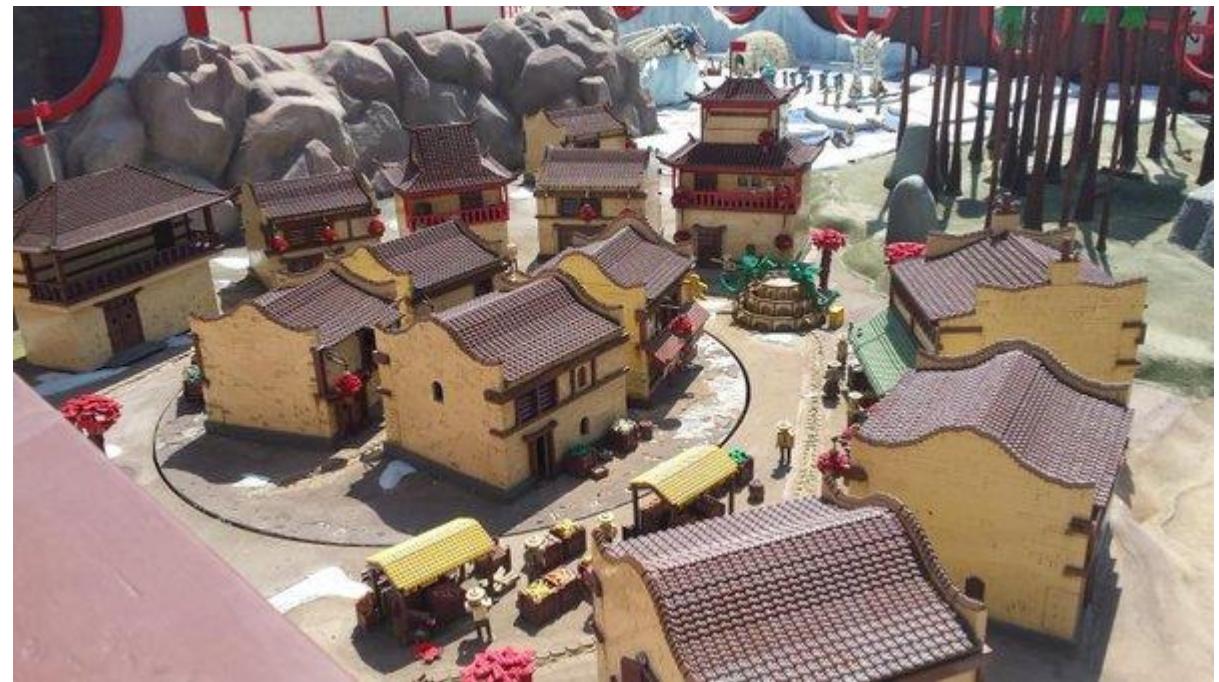


Photo prise à



Bilan & Rétrospective

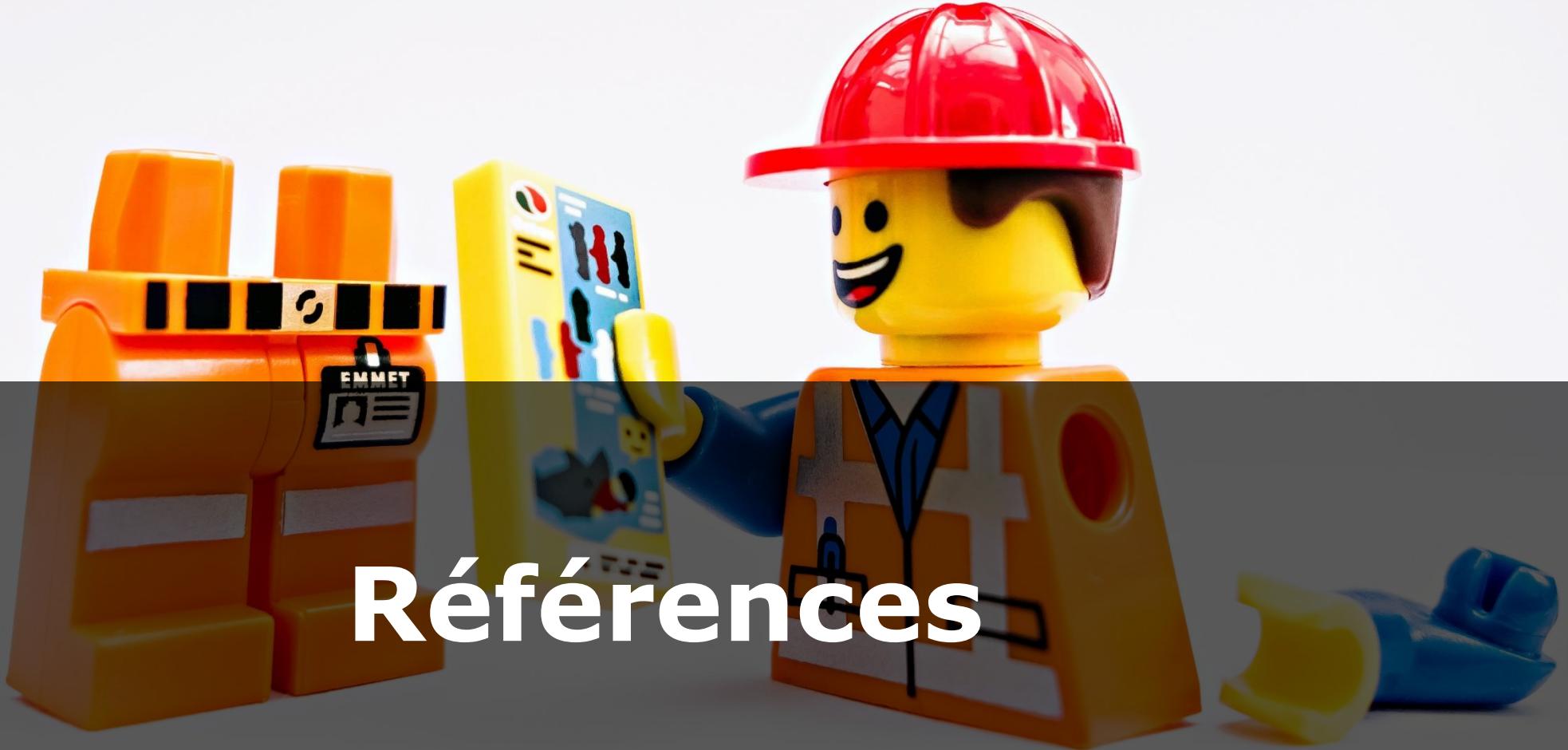
Qu'avez-vous ressenti ?

Qu'avez-vous appris ?

Que mettrez-vous en application dès lundi ?



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 :

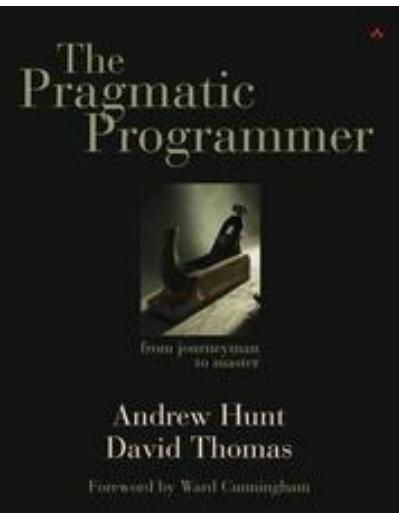
Intro to Refactoring with LEGOs : <http://agilecomplexificationinverter.blogspot.fr/2013/12/intro-to-refactoring-with-legos.html>

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

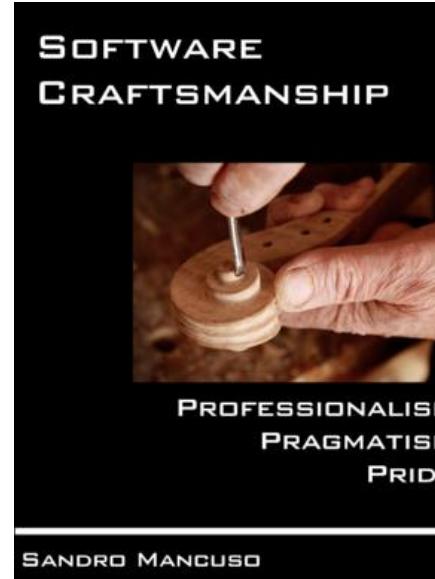
The ROI of Refactoring: Lego vs. Play-Doh :

<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 - 2016

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

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



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