

code is poetry

A poet



poetry

Code is

pragmatique, pas dogmatique

T'essaie de comprendre tous les concepts,
technos, méthodologies qui feront de toi un ou
une meilleur.e dev ?

Ou juste parce que c'est plus fort que toi ?

Plus tu en apprends plus tu réalises que...
tu ne sais rien ?

Du coup tu te sens dépassé.e, pas à la hauteur ?

I AM NOT WORTHY.

PLEASE KILL ME
NOW.

Tout ça nourrit ton syndrome de l'imposteur
pourtant déjà gavé.

Et tu ne sais plus vers quoi te tourner.

T'en arrive même à douter d'être au bon endroit.

C'est le moment de faire un pas ou deux de côté.

C'est le moment de prendre... de la hauteur.



Il est temps d'arrêter de t'épuiser !

Qui plus est, pour rien.

Et bien entendu que OUI, t'es à ta place :)

Comme dirait Orelsan :
ralentis... réfléchis... prends ton temps...

Respire un grand coup !
Prend du recul !

PROJETS
GUIDER TES
APPRENTISSAGES
TU LAISSERAS



Laisse tes projets guider tes apprentissages.

Apprend ce dont tu as besoin,
quand tu en as besoin.

Et avance comme ça, par étape.

Chaque mission, chaque projet
te donnera l'occasion
de monter en compétences.

RIGIDITÉ = DOGME

ADAPTABILITÉ
=
CRÉATIVITÉ

Ne sois pas rigide.

Ne cherche pas à appliquer les concepts coûte
que coûte.

Ils sont là pour te guider.
Pas te diriger.
Et encore moins te figer.

Sers-toi en de base.
N'en garde que ce qui t'arrange.
Et invente le reste.



Maintenant je vais te partager
quelques concepts.

Qui vont dans le sens d'une approche
pragmatique du métier de dev.

Des fois que ça te parle.

Je n'aurais pas le temps de théoriser ni de
démontrer.

Alors je vais tenter d'imager.

$\langle = \rangle$

LESS IS MORE .

Alors, Less is more...

C'est le principe de
« moins y'en a, mieux c'est ! »

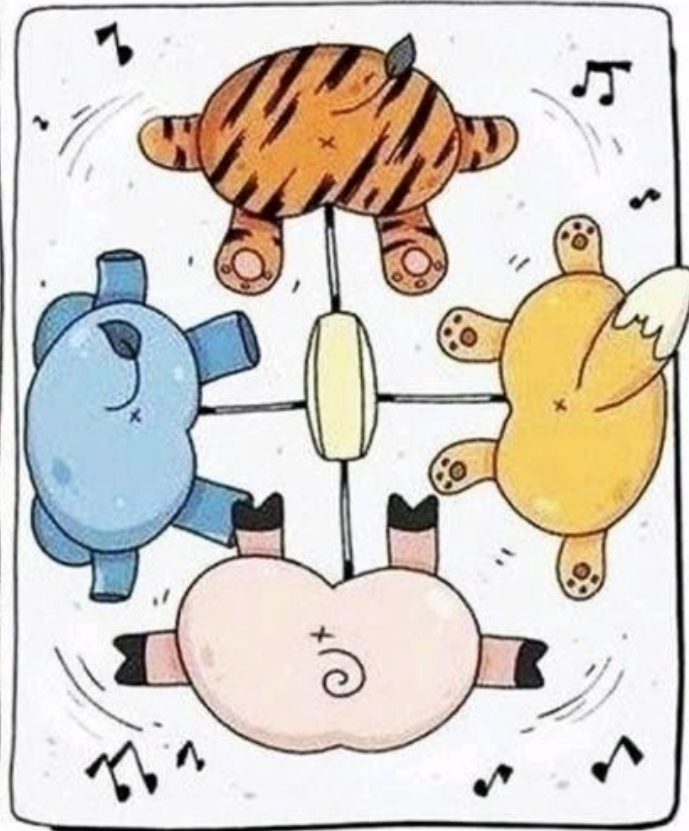
C'est valable pour beaucoup de choses d'ailleurs !

Donc si tu veux être endurant.e, performant.e,
si tu veux y arriver,
il va falloir t'alléger et rester focus !

Don't forget about the final users

Clients

Users



Tu ne sais pas de quoi
le futur sera fait.

Cherche pas à anticiper.

Avance selon les retours des utilisateurs même si
c'est toi l'utilisateur/trice !

Qui ont toutes les chances de t'amener de partout
sauf là où tu l'avais... anticipé !

T'y vois qqch ?

Moi non plus !

Sans visibilité on ne va nulle part.

La clé ce sont des itérations courtes
et les feedbacks utilisateurs.

Plus vite tu mets en production,
plus vite tu as des retours,
plus vite tu peux réajuster ta trajectoire.

UX



Pour t'aider dans cette quête, je te présente
les 3U.

Utile, Utilisable et Utilisé !
Ils représentent l'UX !

Si t'arrives à les atteindre,
ils te donneront :
une lampe, une carte et une boussole !

Avec ça, t'es sûr.e de ne plus jamais te retrouver
à errer dans le noir !

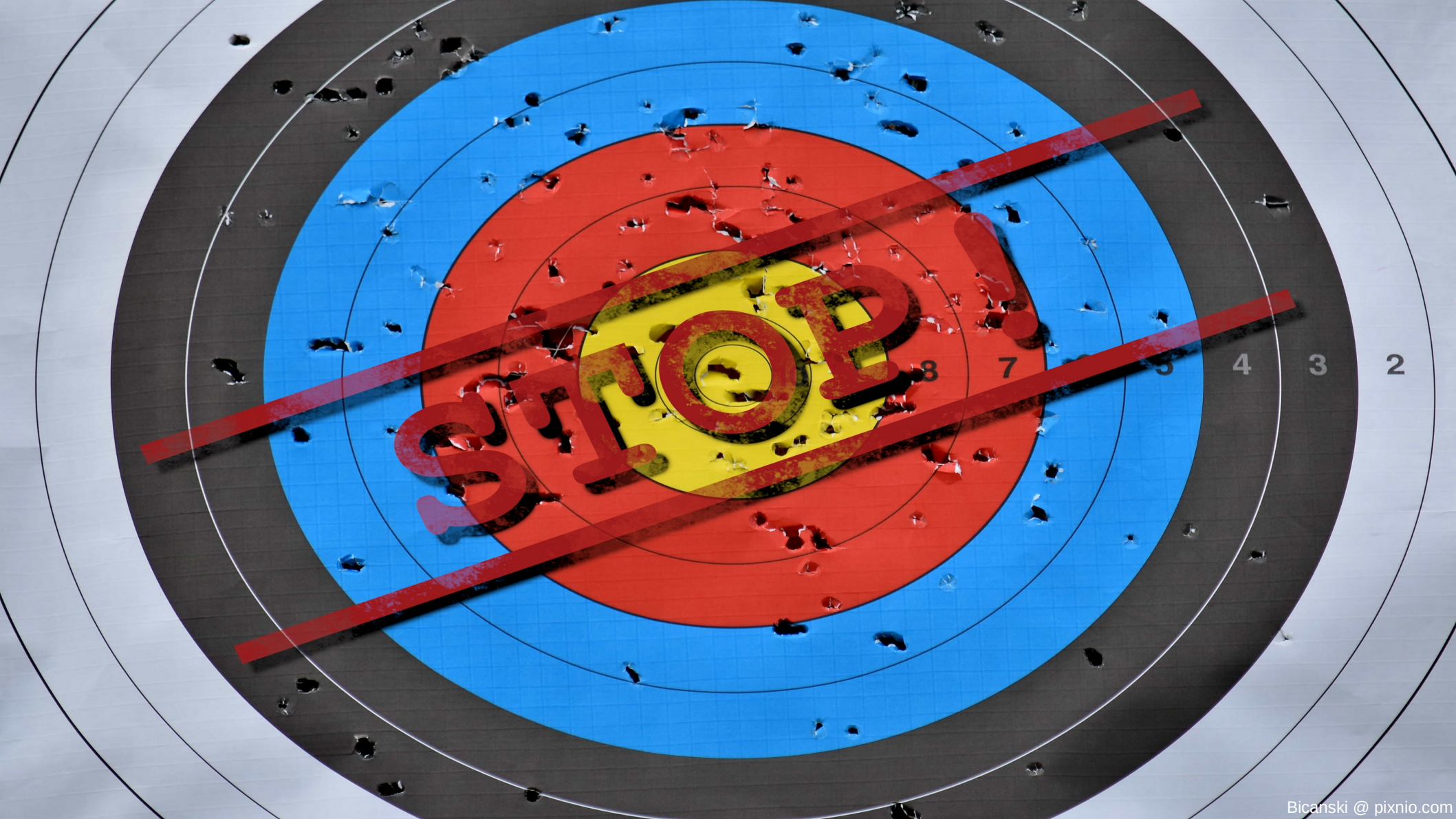


DX

Mais n'oublie pas :
ton xp en tant que dev est tout aussi importante
que celle des utilisateurs.

Si t'arrives à rendre ton code facile à comprendre,
ce sera déjà un cadeau
pour ton futur toi ou les autres.

Consigne toutes les raisons de tes décisions
afin de ne pas revenir dessus
ou d'y revenir en conscience.



En 1 mot : sois efficient.e !
Ne t'éparpille pas.

Fais comme l'OM, va droit au but !

Code le minimum nécessaire
pour franchir chaque étape.

Optimise et automatise
tous les process récurrents.

$$1 > 0$$

1 VAUT MIEUX QUE 0.

Comme dirait Fabien Olicard :
1 vaut mieux que 0

C'est la clé du progrès
et c'est devenu mon mantra.

Fais. Même si c'est pas parfait. Fais.

Même si tu n'as pas assez de temps,
avance ce que tu peux.

Un peu, c'est toujours mieux que rien.




Tu ne seras jamais prêt.e,
alors autant commencer !

C'est comme le sport.
Mieux vaut en faire 5-10 minutes
tous les jours...

...plutôt qu'attendre d'avoir la « possibilité »,
ou plutôt... la motivation,
d'en faire 1h... un jour !

L'enthousiasme vient de petites victoires.

- ~~1 SEULE CHOSE À LA FOIS~~
- ~~CHECK ET/OU BARRE~~
- CONSTATE L'EFFET BOOST 
- ~~RECOMMENCE~~

En 2010 je cherchais un moyen de pérenniser ma liberté géographique.

Je n'avais jamais pensé au dev avant ça et comme ça cochait pas mal de cases...

...j'ai testé,
j'ai appris en codant.

Et toutes ces micro réussites très concrètes, m'ont encouragée à continuer.

COMMIT
PUSH
MERGE
MEP
AGAIN.

Depuis, j'ai réalisé plusieurs projets.

Et j'ai compris comment
m'organiser efficacement.

À chaque fois que j'ouvre un projet,
je sais exactement où je dois aller, comment et pourquoi.

Je n'ai plus qu'à suivre les étapes
que j'ai planifiées en amont
et les adapter au fur et à mesure de l'évolution des
besoins.



GitLab

LA PLATEFORME **DEVSECOPS** ULTIME

DE LA PLANIFICATION
À LA PRODUCTION

D'ailleurs, si je ne devais te citer qu'un outil, ce serait... Gitlab !

À lui seul, il te fournit pratiquement tout ce dont tu peux avoir besoin pour découper cette montagne qui peut paraître infranchissable...

...en petits cailloux colorés tous mignons !

Que tu pourras organiser selon l'importance et l'urgence !

FRAMAGIT



VOUS ÊTES ENCORE SUR GITHUB?!



J'utilise...

- ses **milestones** pour **planifier** comment aborder la montagne qui se dresse devant moi.
- ses **issues** pour en **définir** les étapes.
- ses **tasks** pour **découper** les étapes en micro-étapes facilement franchissables.

PLANIFIER
DÉFINIR LES ÉTAPES
DÉCOUPER EN MICRO-ÉTAPES
DOCUMENTER
VISUALISER
PRIORISER
RESTER FOCUS
GARDER LA MOTIVATION

• • •

Mais aussi...

- son **wiki** notamment pour **documenter** mes choix et la vie de mon projet
- son **kanban** pour **visualiser et prioriser** les étapes
- sa **todo** pour rester **focus** et garder la **motivation**

et tout ce qu'un outil de versionning peut apporter de meilleur :)

MERCI 🙌



Nadia Meyer
Développeuse PHP / Symfony ♥
lanewbee.fr
in/nadia-meyer/



<https://lanewbee.fr/pragmatique-pas-dogmatique/>

Voilà c'est déjà fini !
Je n'ai pu qu'effleurer le sujet.

Il y a beaucoup d'autres choses qu'il aurait fallu
évoquer ou nuancer.

Mais si tu souhaites aller plus loin,
retrouve tous les concepts évoqués dans cette
présentation
en scannant le QR code ou suivant le lien...

Merci !