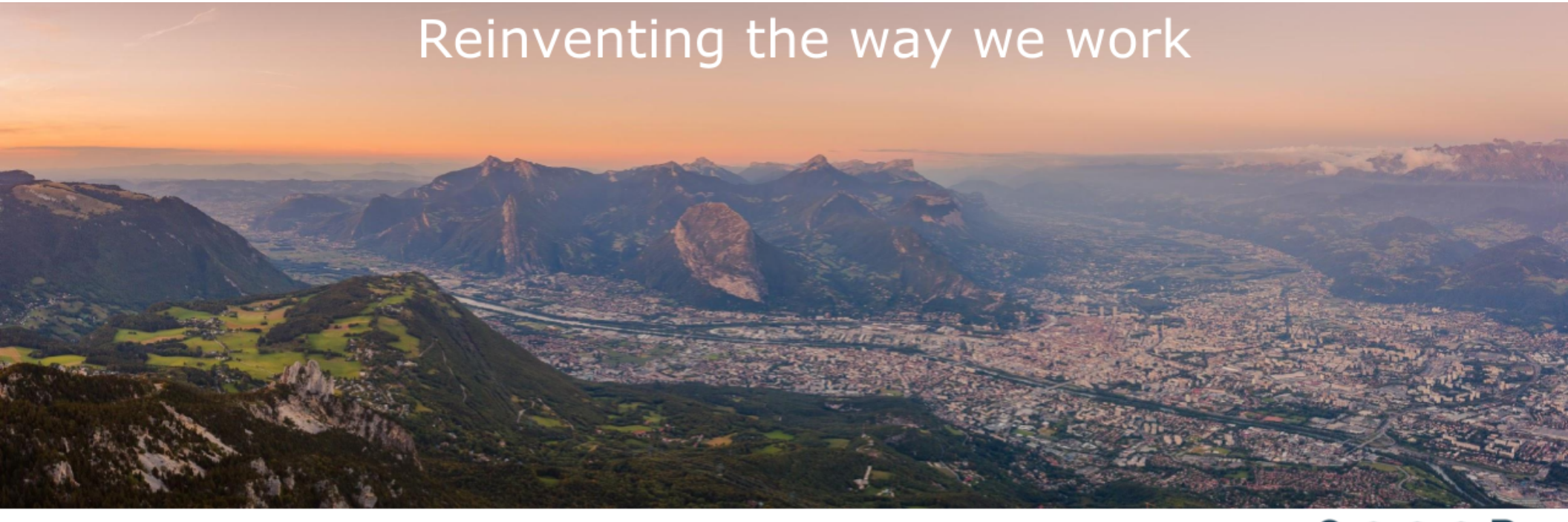


Merci pour votre confiance

VIVERIS
Innover. Simplifier. Partager.

GOOOD
DÉVELOPPONS NOTRE AGILITÉ

KAIZEN

sopra steria

WORLD TRADE CENTER® GRENOBLE
Centre de congrès

Remo

zenika
<animés par la passion>

Schneider Electric

orange™

Persistent
See Beyond, Rise Above

open
WE EMPOWER YOUR DIGITAL WORLD

CGI

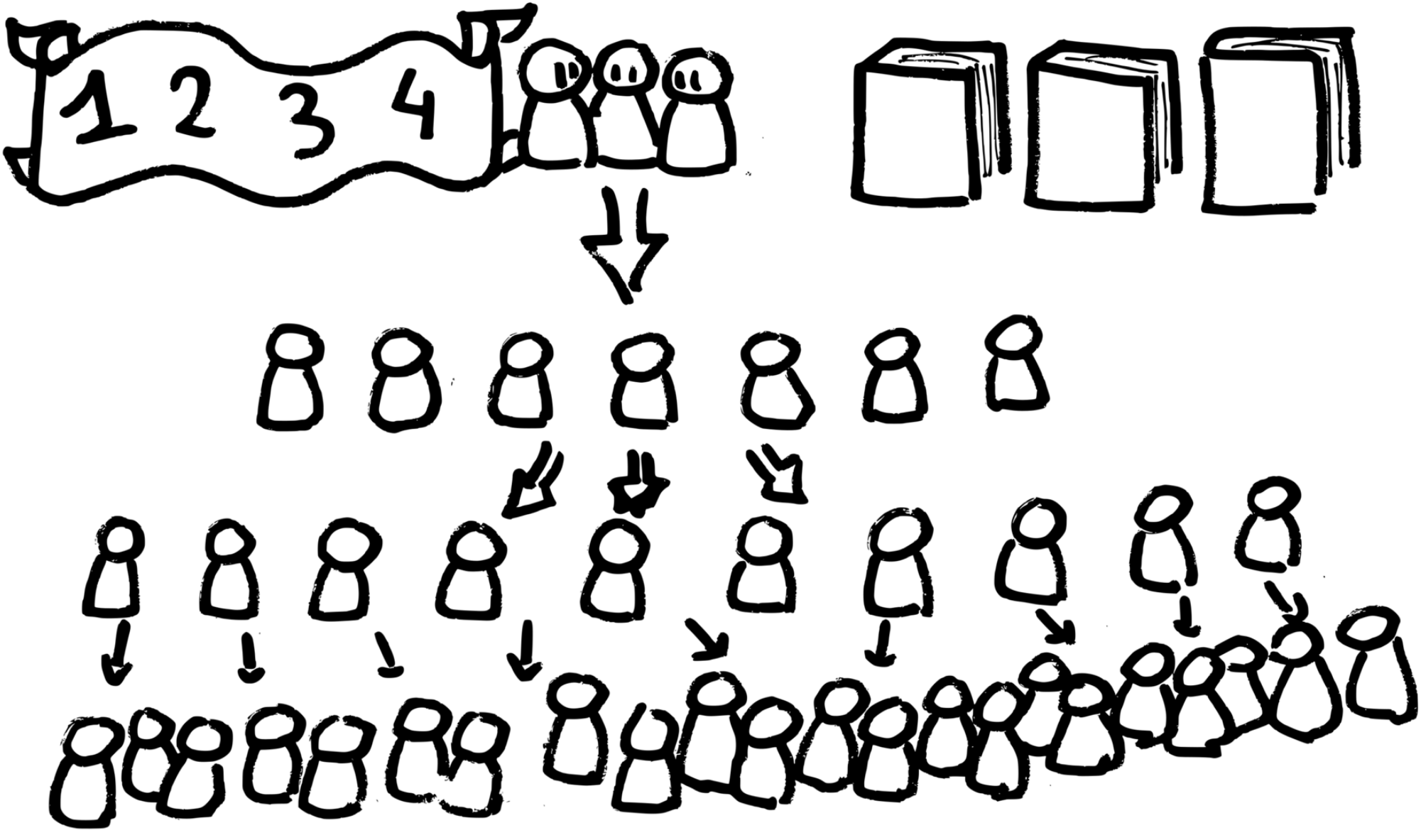
klaxoon

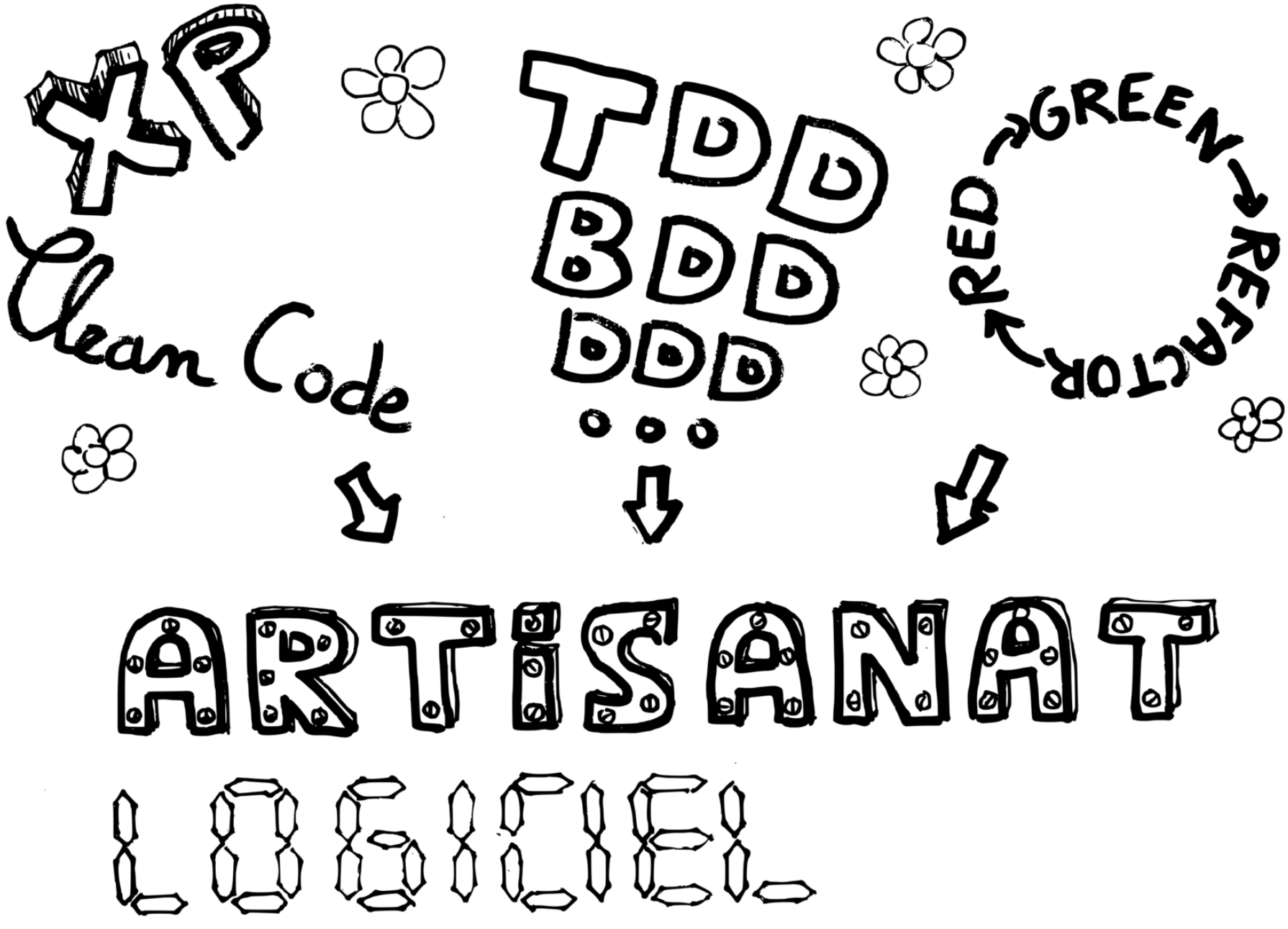
Roti.express

ARTISANAT LOGICIEL

UN  ZÉTÉTIQUE

ARTISANAT LOGICIEL

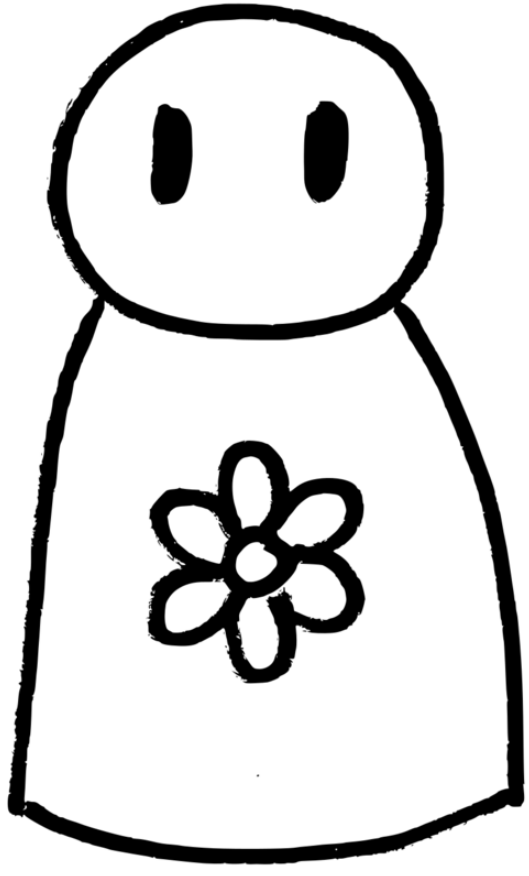

1 2 3 4

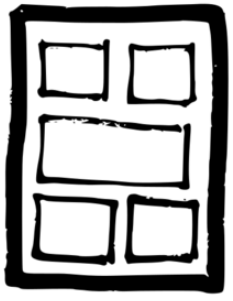

XP
Clean Code
TDD
BDD
DDD
...
RED → GREEN → REFACTOR
ARTISANAT
LOGICIEL

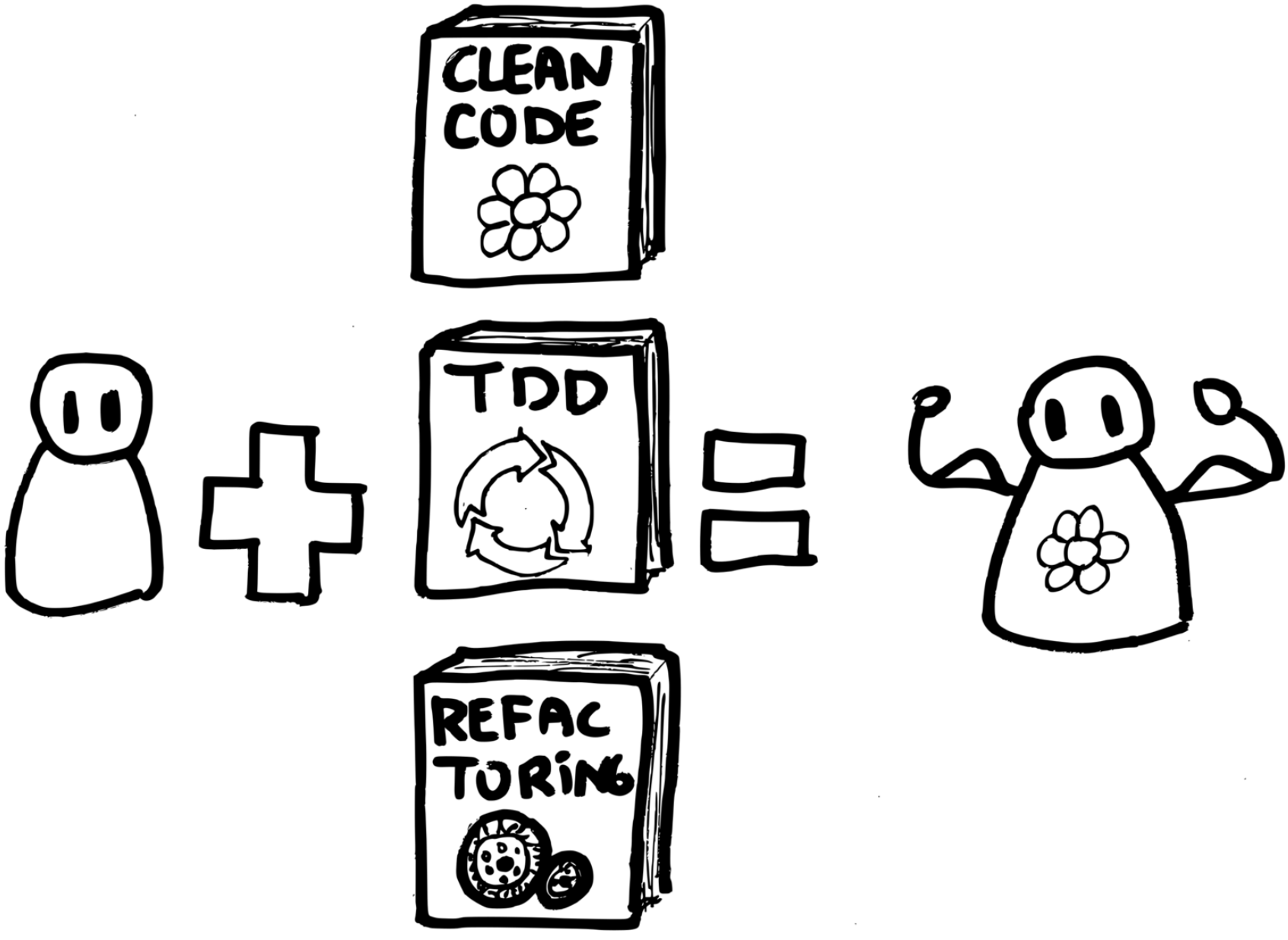
MON CONTEXTE

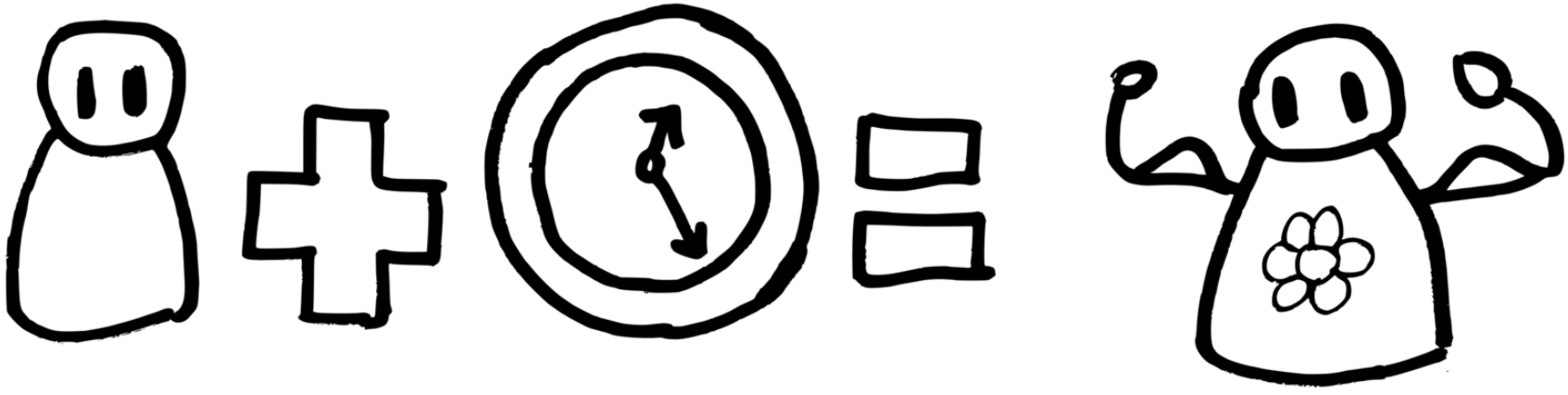
VICTOR LAMBRET

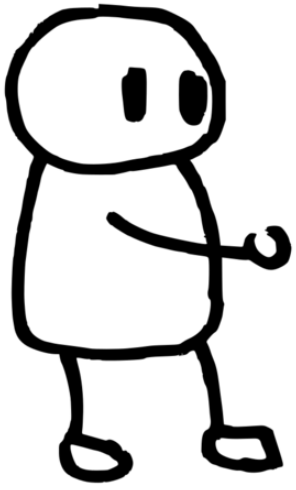
DÉVELOPPEUR

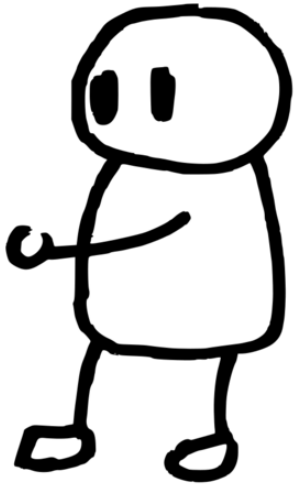


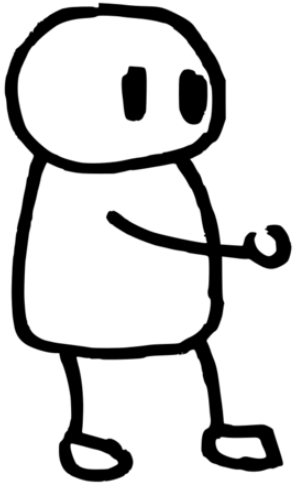


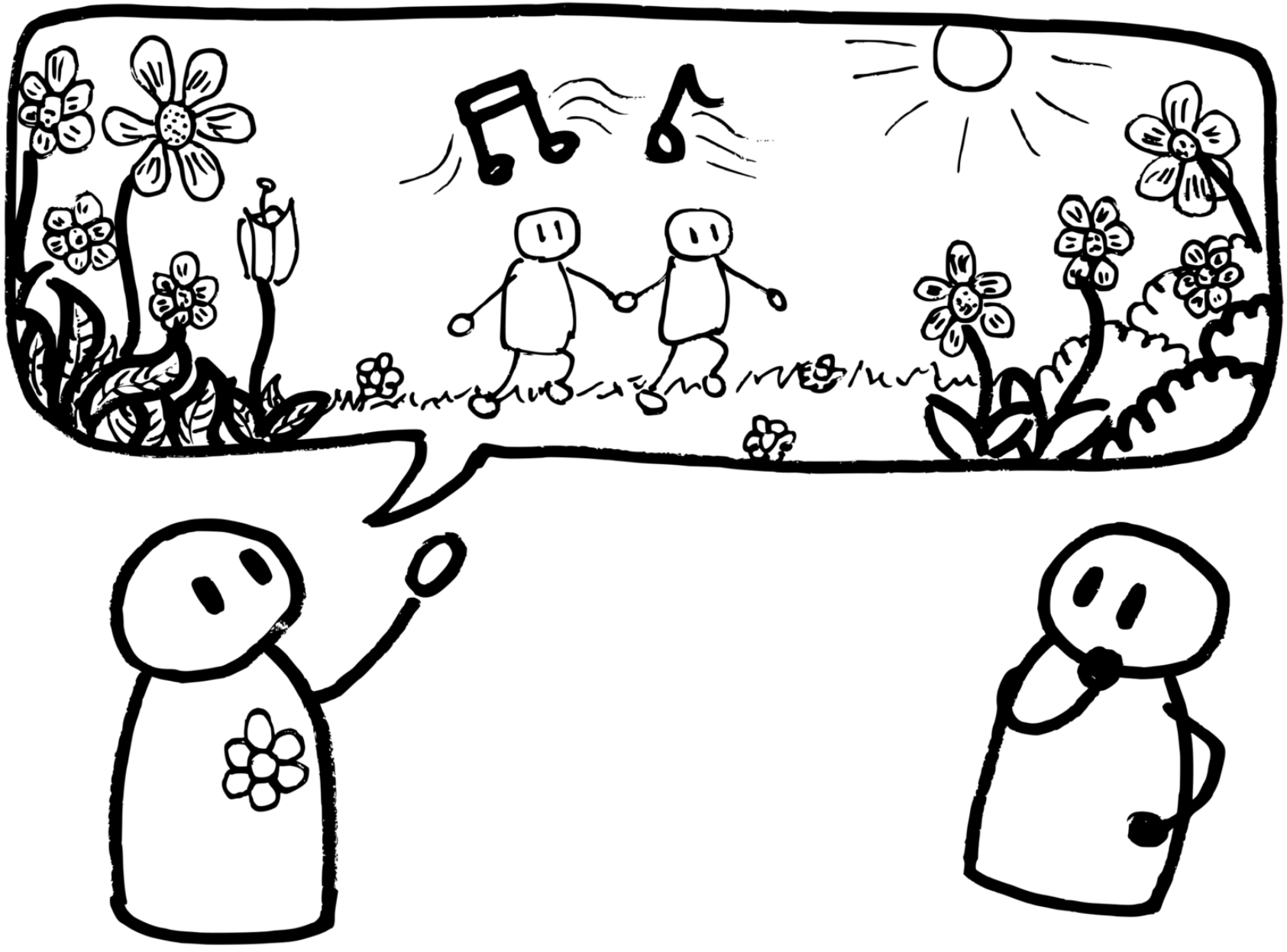

CLEAN
CODE
TDD
REFAC
TORING

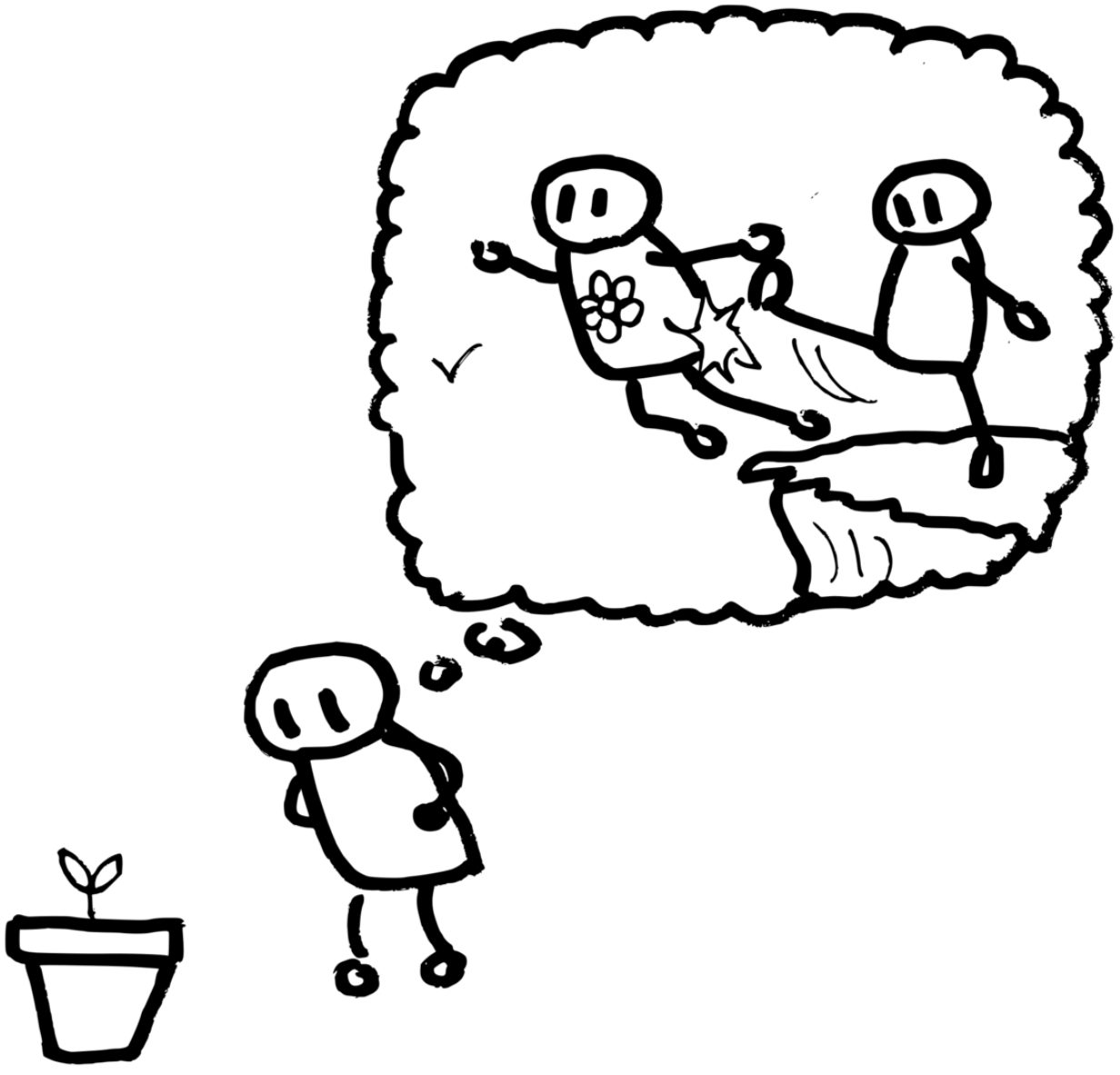


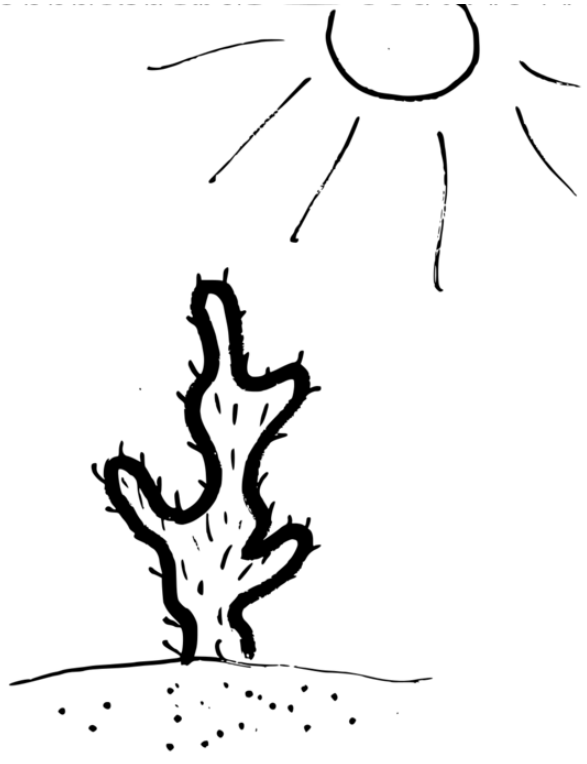








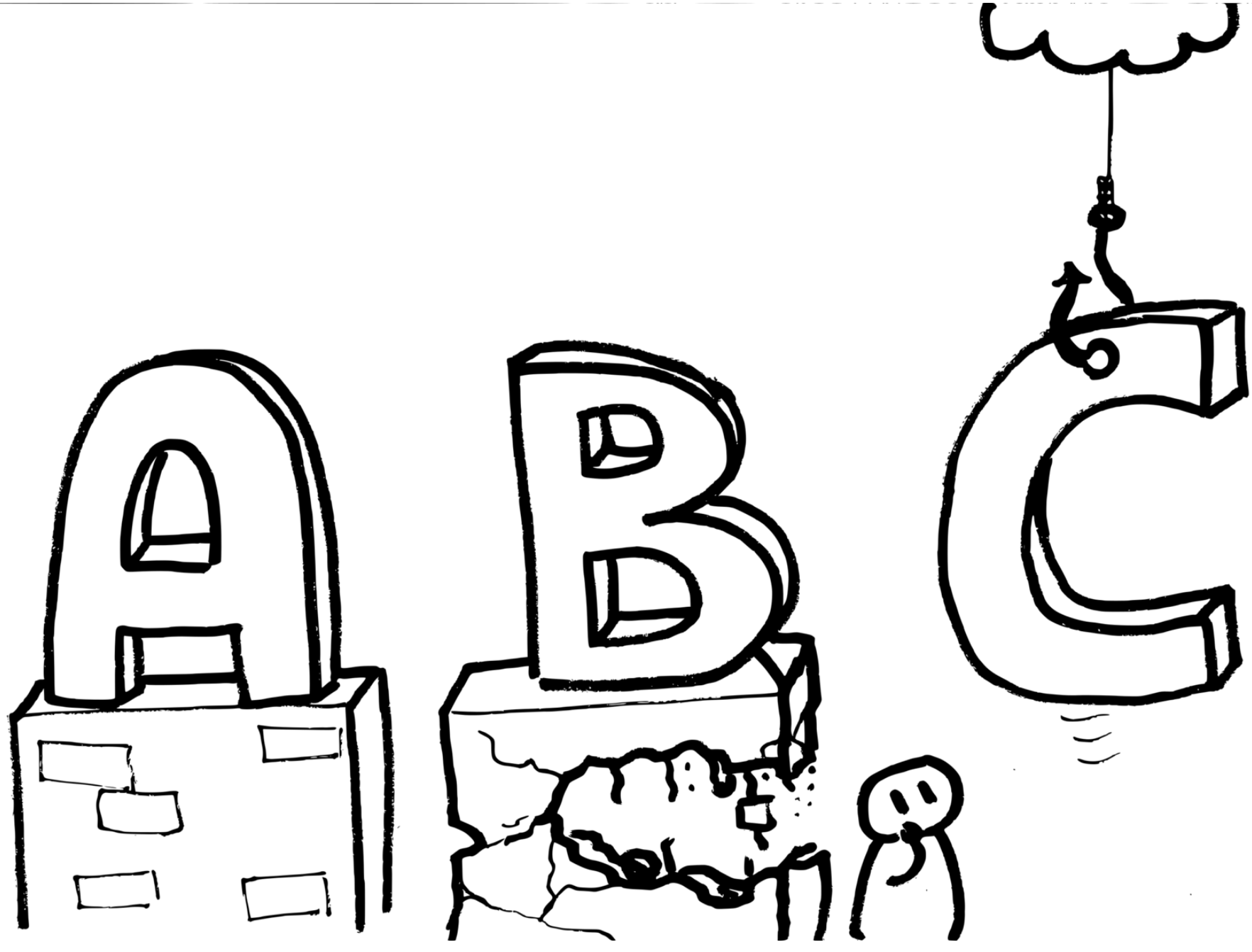




LA ZÉTÉTIQUE

DÉFINITIONS MULTIPLES

- L’art du doute
- Hygiène préventive du jugement
- L’art de faire la différence entre ce qui relève de la science et ce qui relève de la croyance
- Scepticisme scientifique


A
B
C

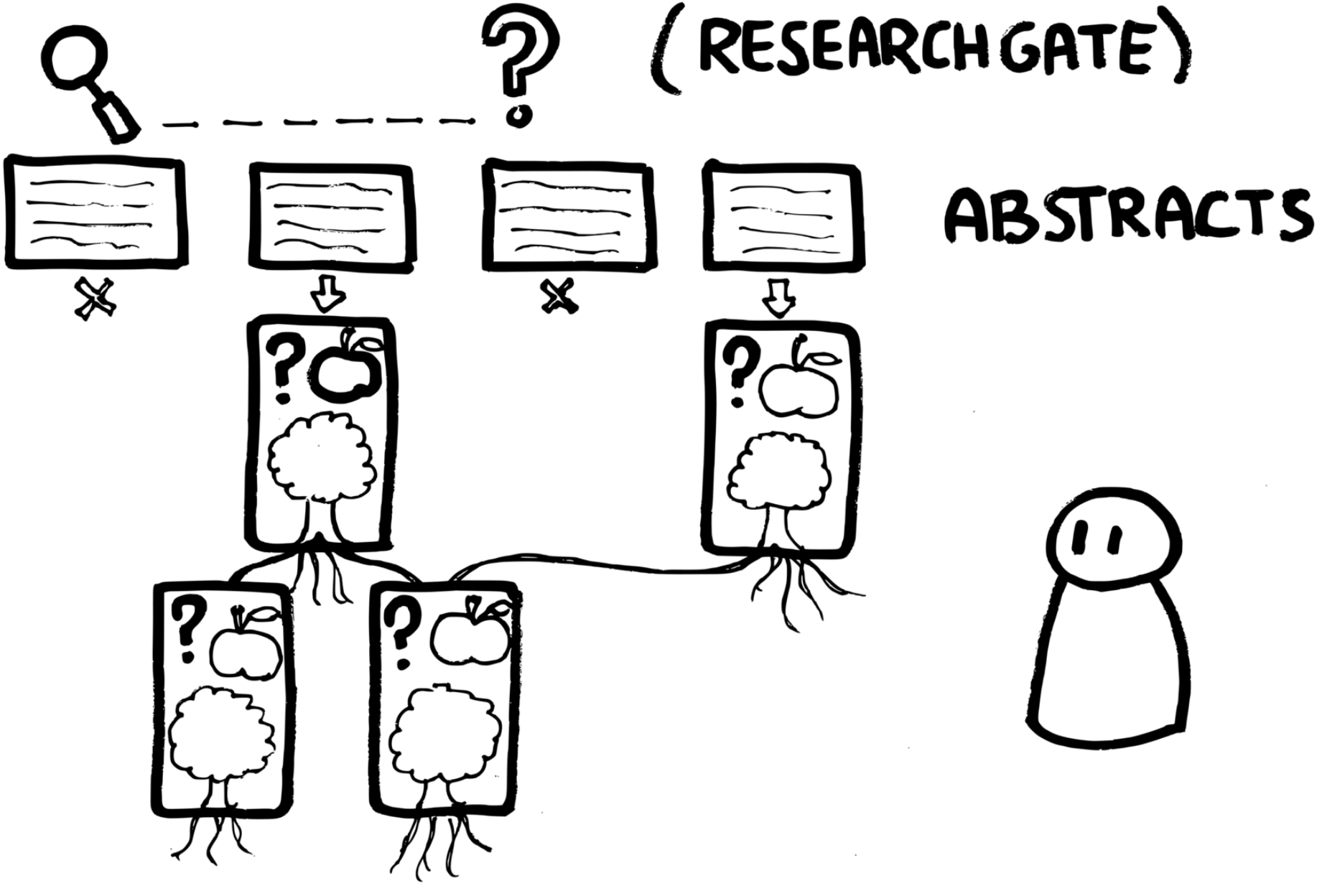
A
?
B
?
C

MÉTHODE SCIENTIFIQUE

- Basée sur les faits
- Réfutable
- Revue par les pairs

STRUCTURE PUBLICATION SCIENTIFIQUE

- Introduction
- Description méthode
- Résultats
- Discussion
- Conclusion
- Bibliographie


?
(RESEARCHGATE)
ABSTRACTS

APPLICATION SUR DEUX SUJETS

- Un sujet trivial Clean Code
- Le Test Driven Development

CLEAN CODE : TAILLE DES FONCTIONS

CLEAN CODE : TAILLE DES FONCTIONS

The first rule of functions is that they should be small. The second rule of functions is that they should be smaller than that.

CLEAN CODE : TAILLE DES FONCTIONS

Every function in this program was just two, or three, or four lines long. Each was transparently obvious. Each told a story. And each led you to the next in a compelling order. That's how short your functions should be !

CLEAN CODE : TAILLE DES FONCTIONS

> *This is not an assertion that I can justify. I can't provide any references to research that shows that very small functions are better*

Pas de références ? Aucun soucis, c'est une telle évidence qu'on va trouver facilement :-)

Software errors and complexity: an empirical investigation

Victor R. Basili and Barry T. Perricone

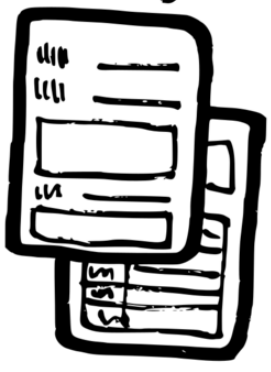
Communications of the ACM, January 1984 Volume 27 Number 1 (24 ans avant CC)

http://www.lsmod.de/~bernhard/cvs/text/dipl/papers/p-basili.pdf

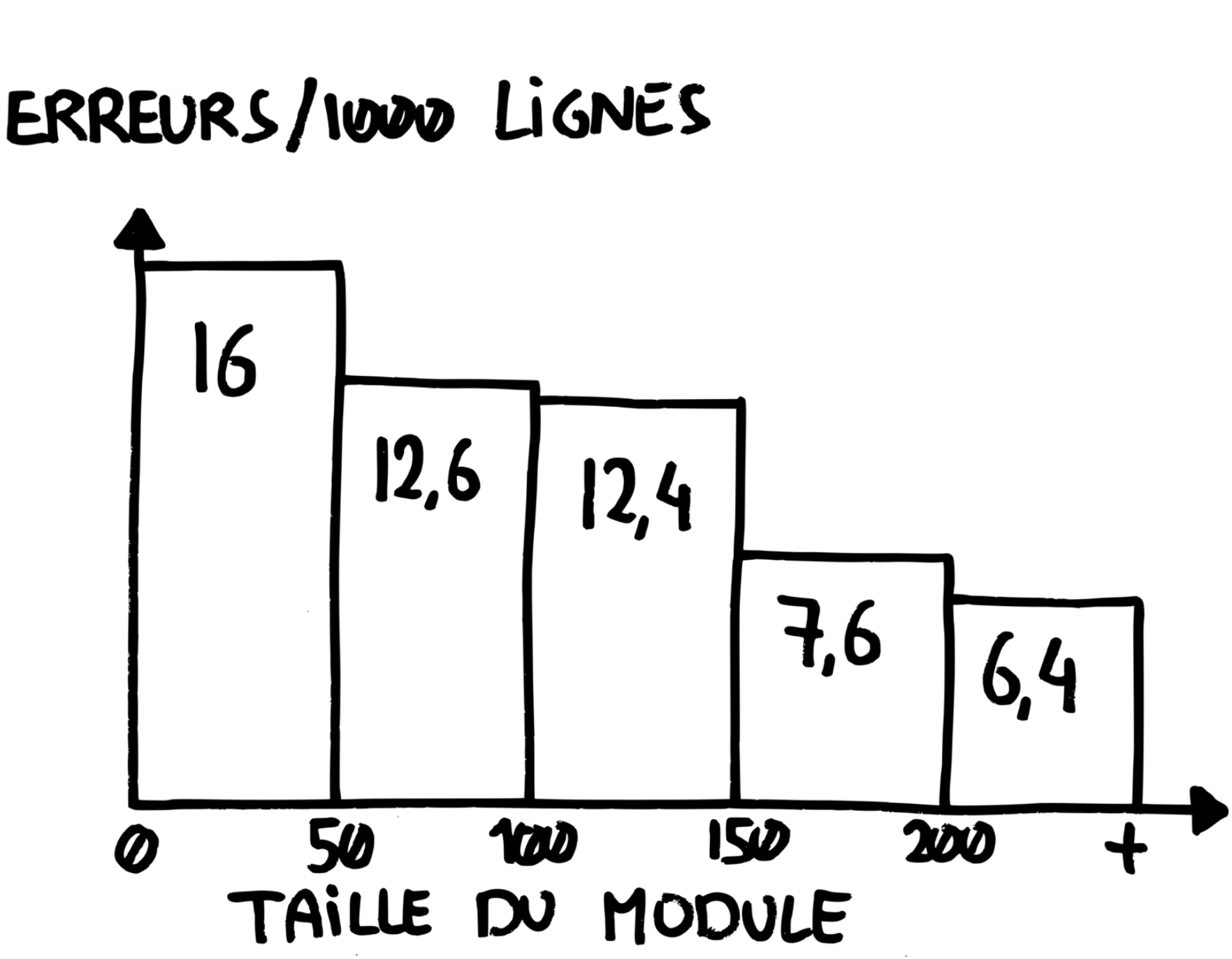
CHANGEMENT

M M M
M M M

FORTRAN
90000 LIGNES
370 MODULES



FORMULAIRE


ERREURS/1000 LIGNES
16
12,6
12,4
7,6
6,4
0
50
100
150
200
+
TAILLE DU MODULE

CONCLUSION

One surprising result was that module size did not account for error proneness. In fact, it was quite the contrary […]. This result implies we are not yet ready to put artificial limits on module size and complexity.

MINCE ALORS !

Les résultats ne sont pas ceux attendus !

Mais c’est une vielle étude. Maintenant, dans le contexte de Clean Code ça marche forcément !

C’est une telle évidence qu’on va trouver facilement :-)

Effects of Clean Code on Understandability

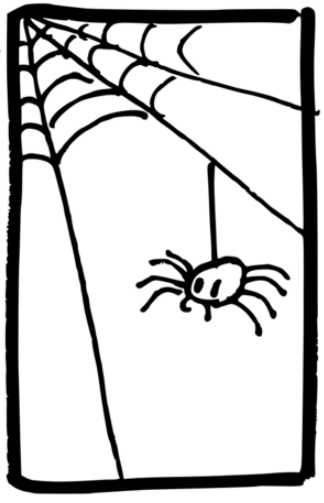
Henning Grimeland Koller

Master's Thesis Spring 2016

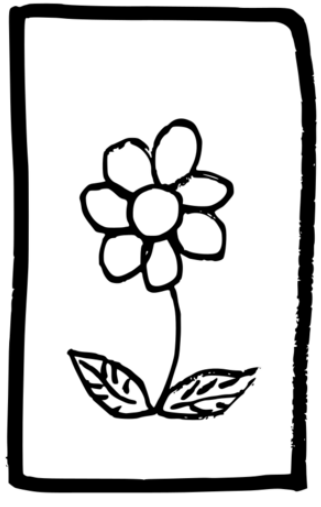
https://www.duo.uio.no/bitstream/handle/10852/51127/

(Pas de revue par les pairs mais plus détaillé.
Publication scientifique similaire disponible ensuite)

JAVA

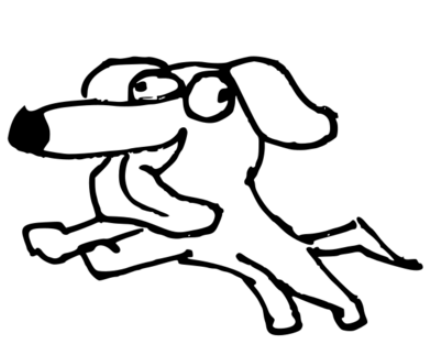


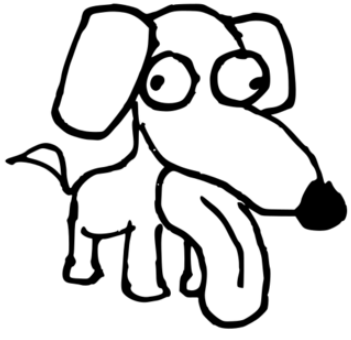
PAS D'BOL



WINNERS


KILL ME
PLIZ





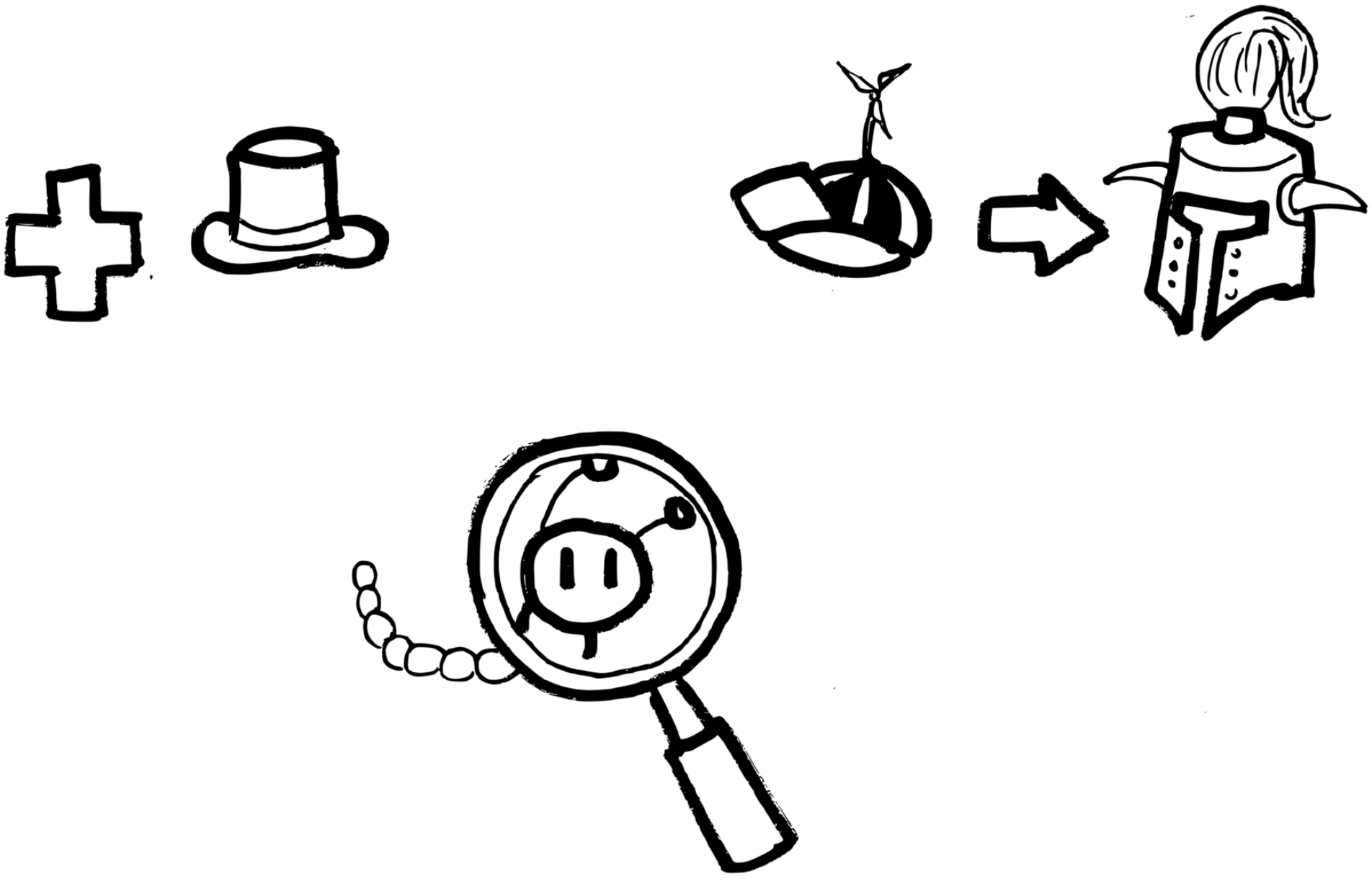


Figure 5.11: Second Run of the Experiment: Time used by participants on assignment 1

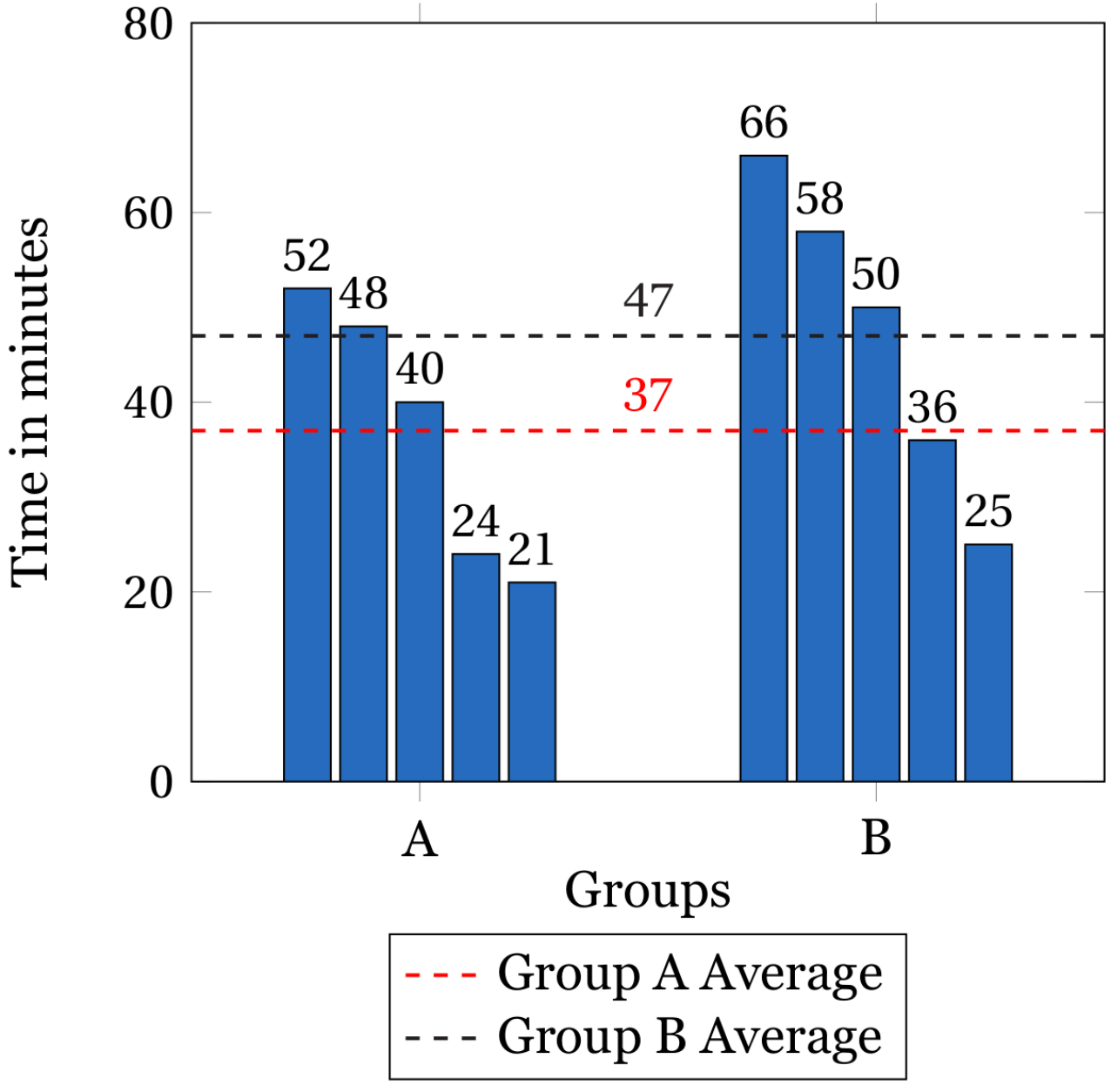


Figure 5.12: Second Run of the Experiment: Time used by participants on assignment 2

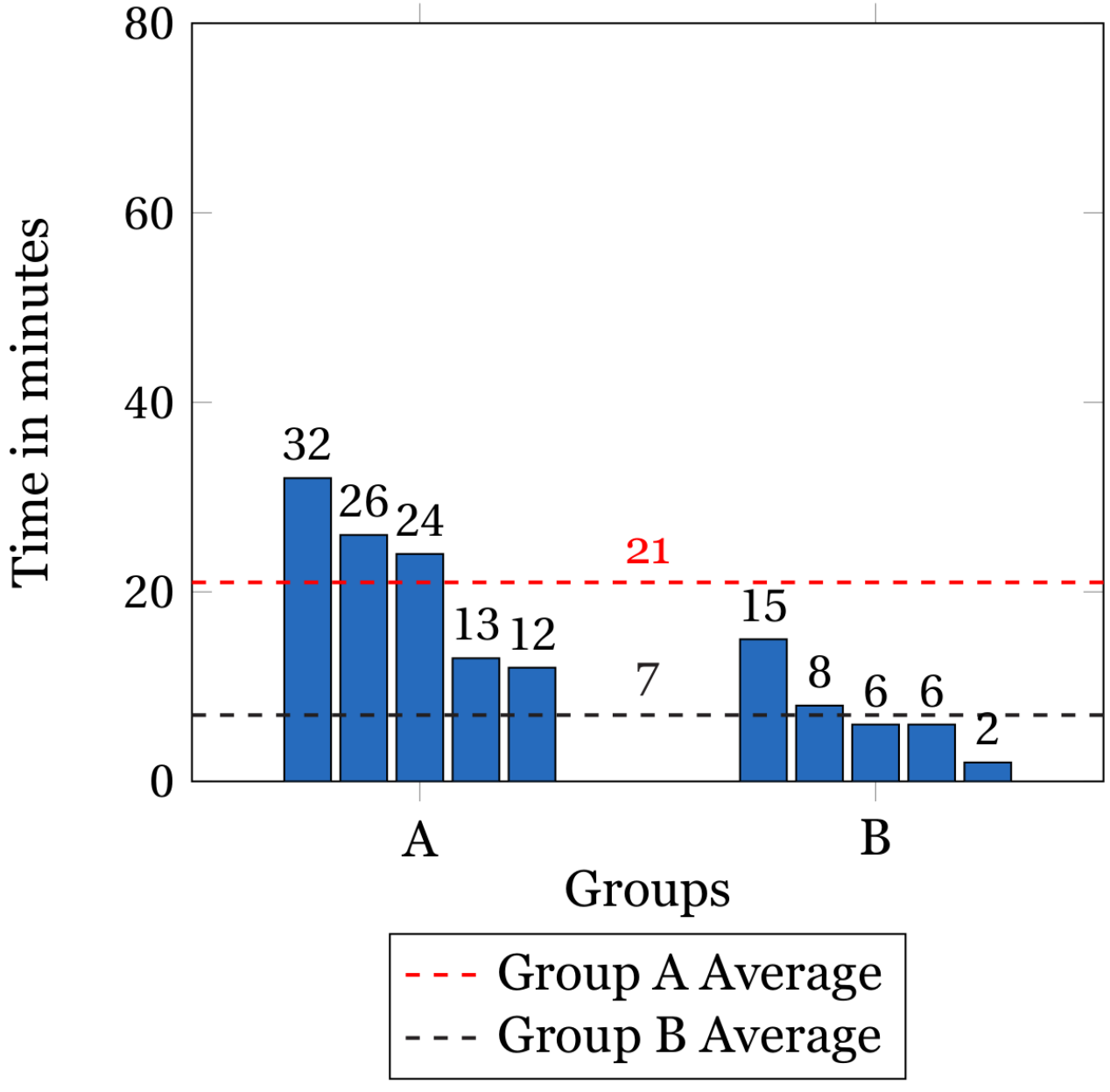
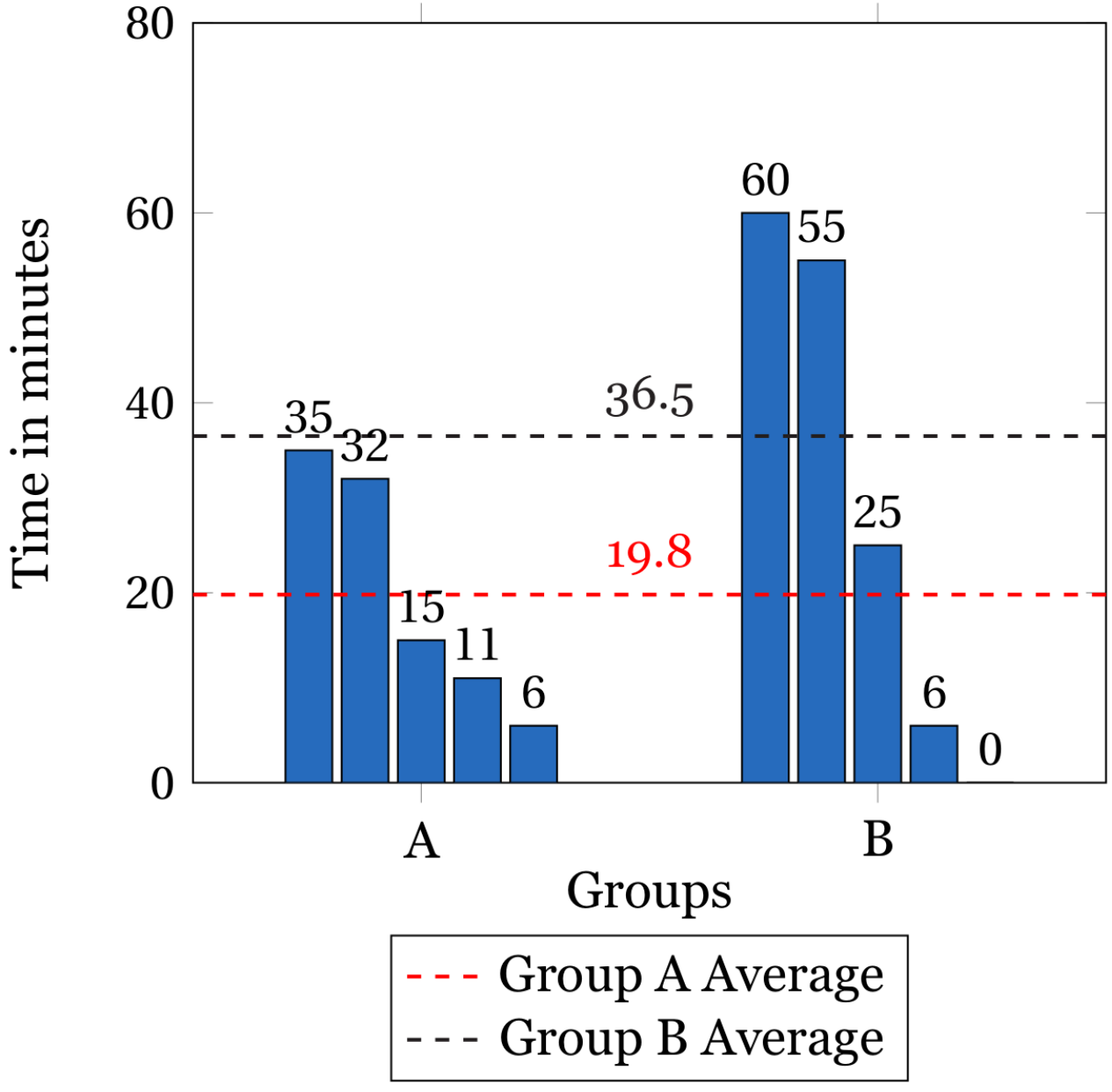


Figure 5.13: Second Run of the Experiment: Time used by participants on assignment 3



CONCLUSION

Despite our expectations, the results from the experiment show that we were wrong […] there seems to be no immediate benefit of Clean Code in form of understandability.

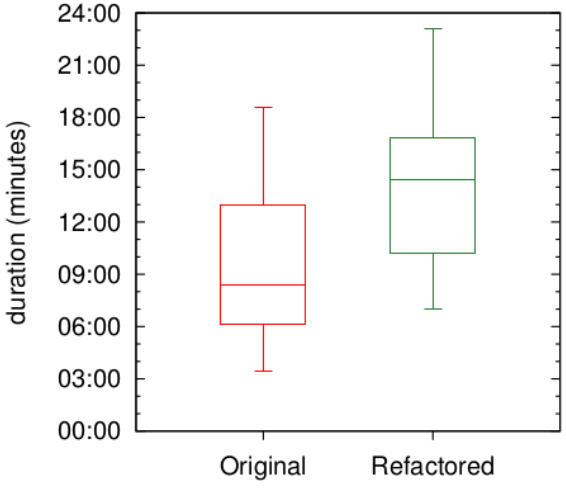
CONCLUSION

Our research points to other benefits than understandability being achieved, such as maintainability and code quality.

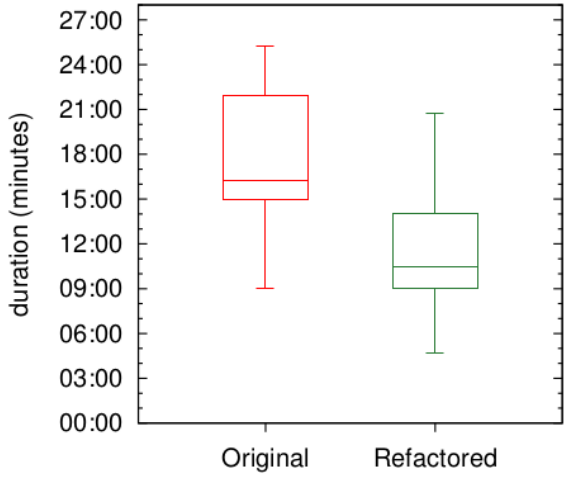
Old Habits Die Hard: Why Refactoring for Understandab Not Give Immediate Benefits

Erik Ammerlaan, Wim Veninga, Andy Zaidman

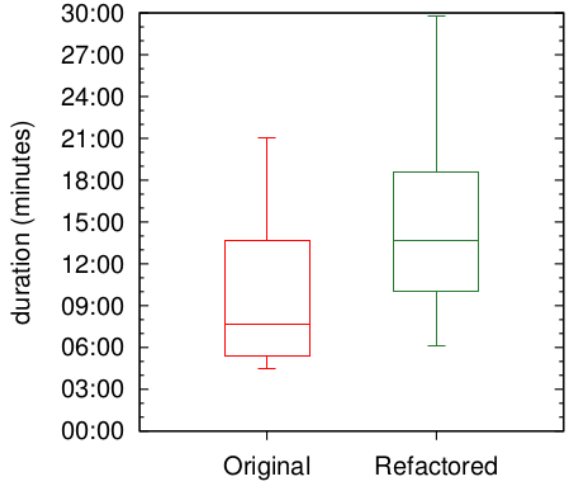
Conference: 22nd IEEE International Conference on Softw Evolution, and Reengineering, SANER 2015, Montreal, QC March 2-6, 2015



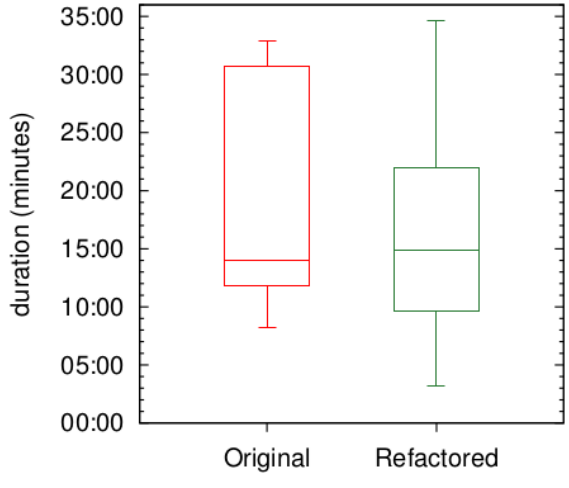
(a) RejectionNotifier (S)



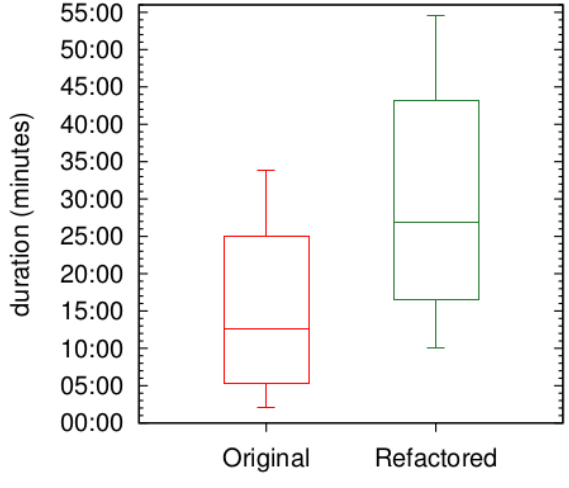
(b) PurchaseOrderTools (S)



(c) ContractProlongation (S)



(d) FinancialEntryTools (M)



(e) ERDomain (L)

CONCLUSION

we observed that the productivity of developers can be influenced both positively and negatively by code that is refactored for understandability when performing small coding tasks

MINCE ALORS !

Les résultats ne sont pas ceux attendus !

Il faut se faire une raison : cette "évidence" n’en est pas une

La réalité est plus compliquée que ça :)

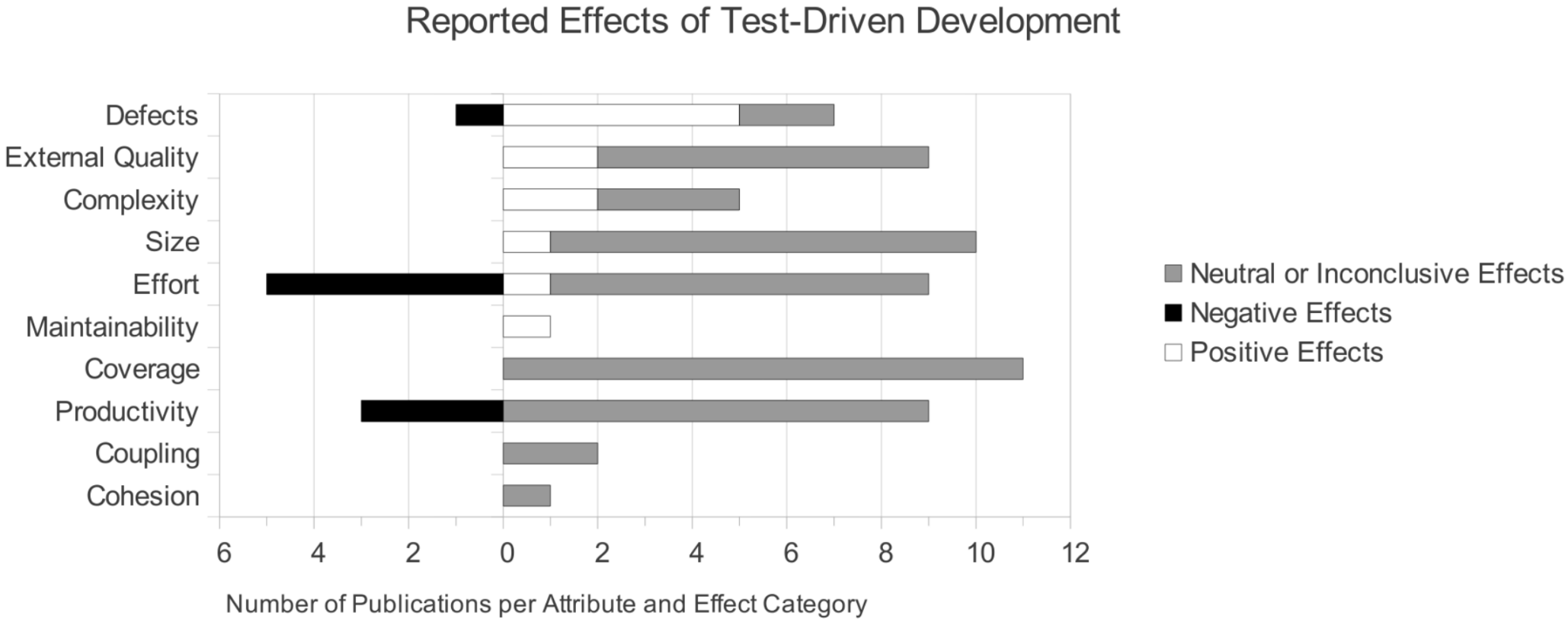
ÉVALUATION DU TEST DRIVEN DEVELOPMENT

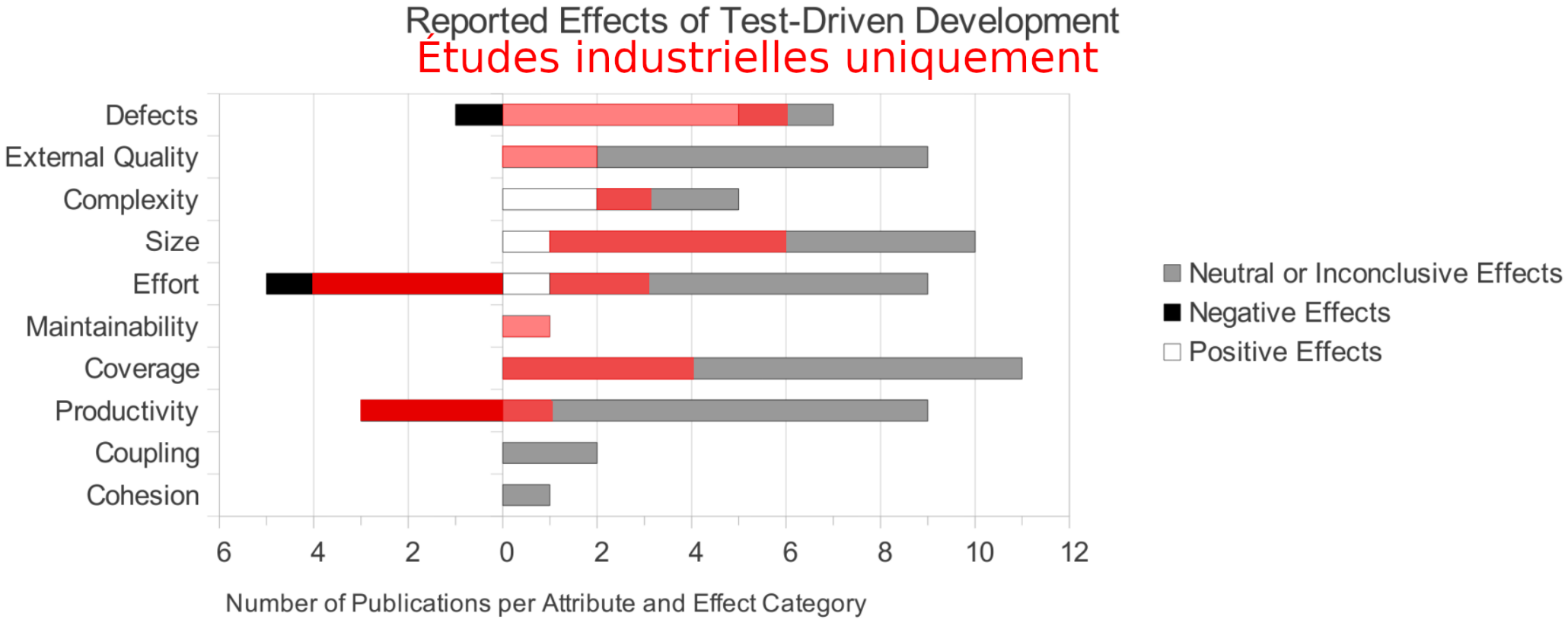
Vu les résultats précédents, que vais-je découvrir sur un sujet complexe ?

Effects of Test-Driven Development: A Comparative Ana Empirical Studies

Simo Mäkinen and Jürgen Münch

Conference PaperinLecture Notes in Business Informatio January 2014


Reported Effects of Test-Driven Development
Defects
External Quality
Complexity
Size
Effort
Maintainability
Coverage
Productivity
Coupling
Cohesion
6
4
2
0
2
4
6
8
10
12
Number of Publications per Attribute and Effect Category
Neutral or Inconclusive Effects
Negative Effects
Positive Effects

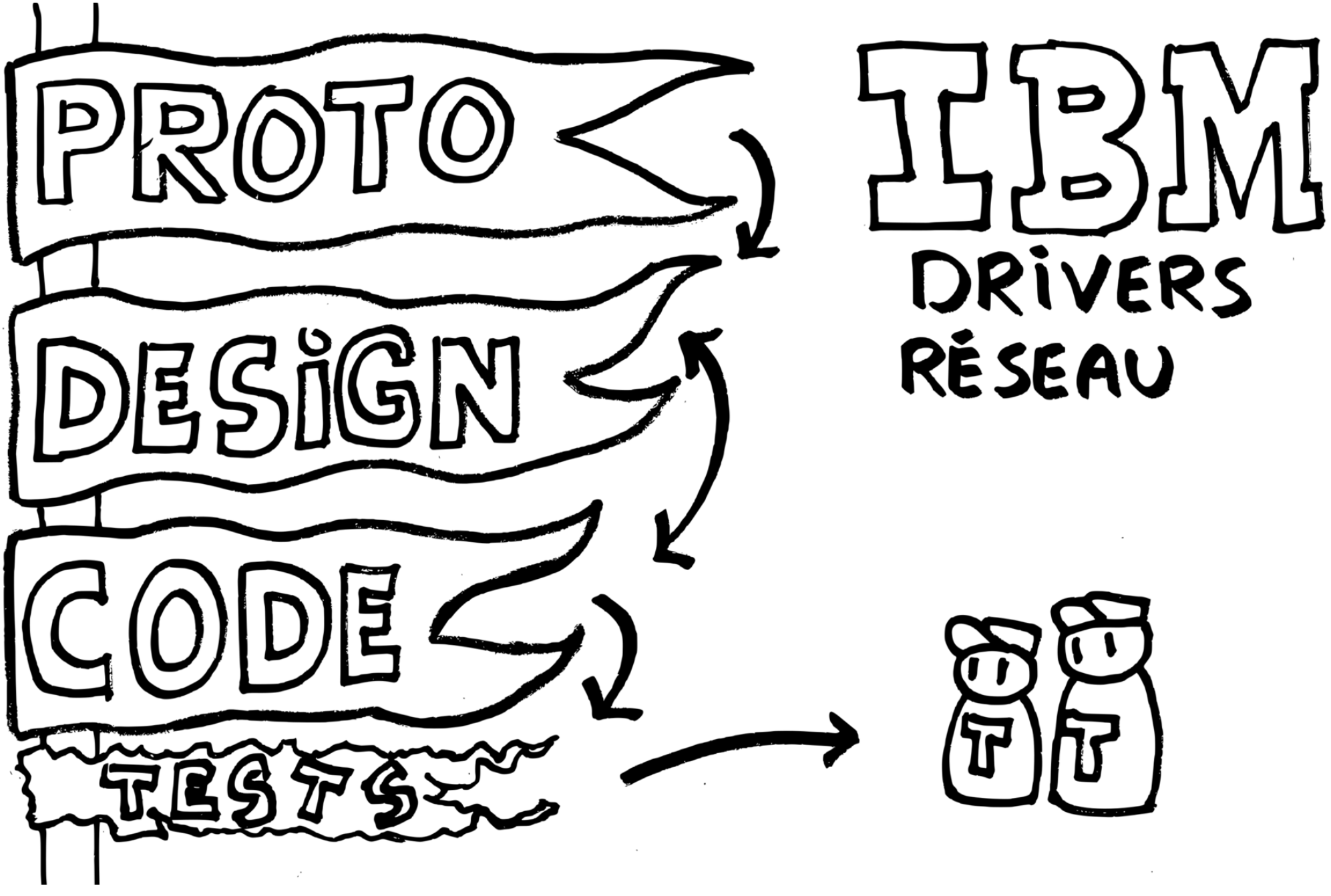

Reported Effects of Test-Driven Development
Études industrielles uniquement
Defects
External Quality
Complexity
Size
Effort
Maintainability
Coverage
Productivity
Coupling
Cohesion
6
4
2
0
2
4
6
8
10
12
Number of Publications per Attribute and Effect Category
Neutral or Inconclusive Effects
Negative Effects
Positive Effects

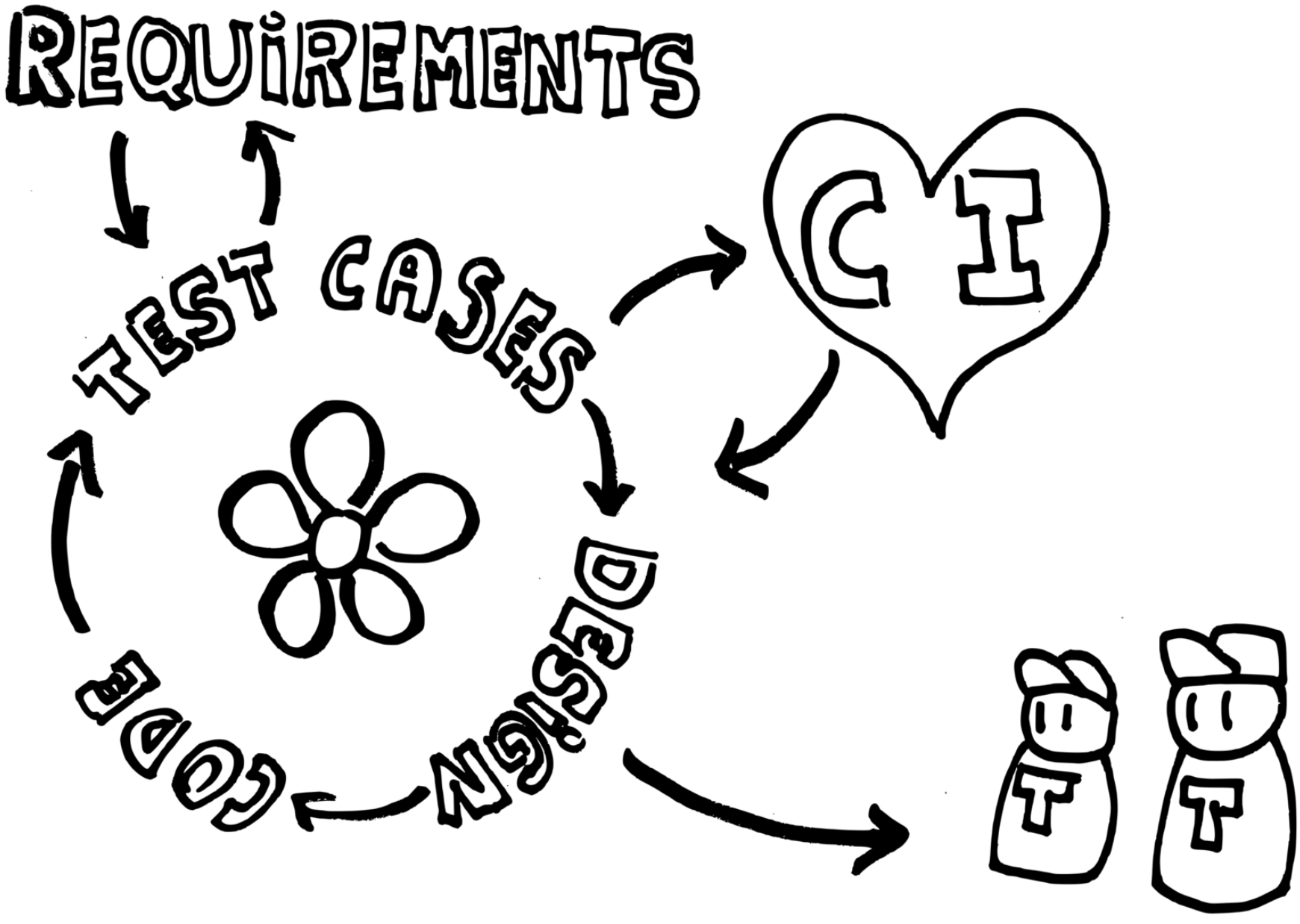
Realizing quality improvement through test driven development: results and experiences of four industria

Nachiappan Nagappan, E. Michael Maximilien, Thirumale
Laurie Williams

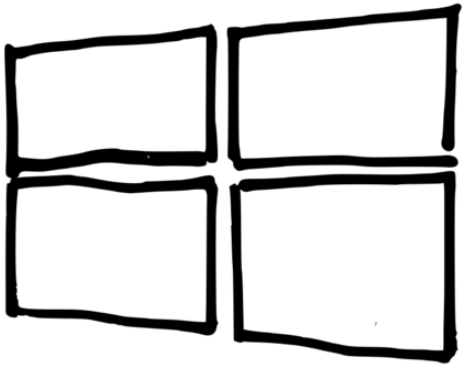
June 2008 Empirical Software Engineering 13(3):289-302

https://people.engr.ncsu.edu/gjin2/Classes/591/Spring2
tdd-b.pdf

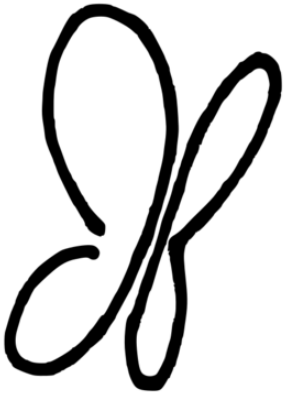

PROTO
DESIGN
CODE
TESTS
IBM
DRIVERS
RÉSEAU
T
T


REQUIREMENTS
TEST CASES
DESIGN
CODE
CI
FT
FT

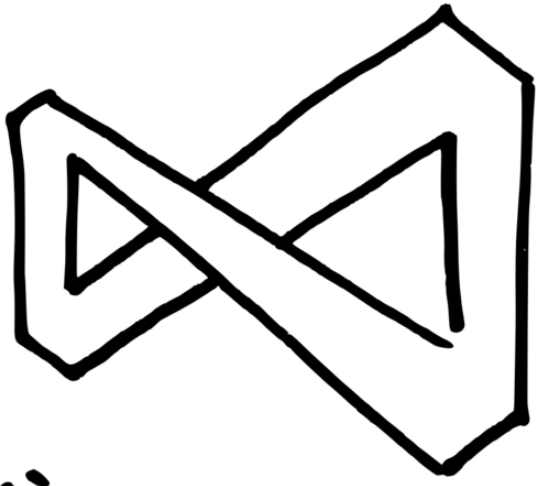
MICROSOFT



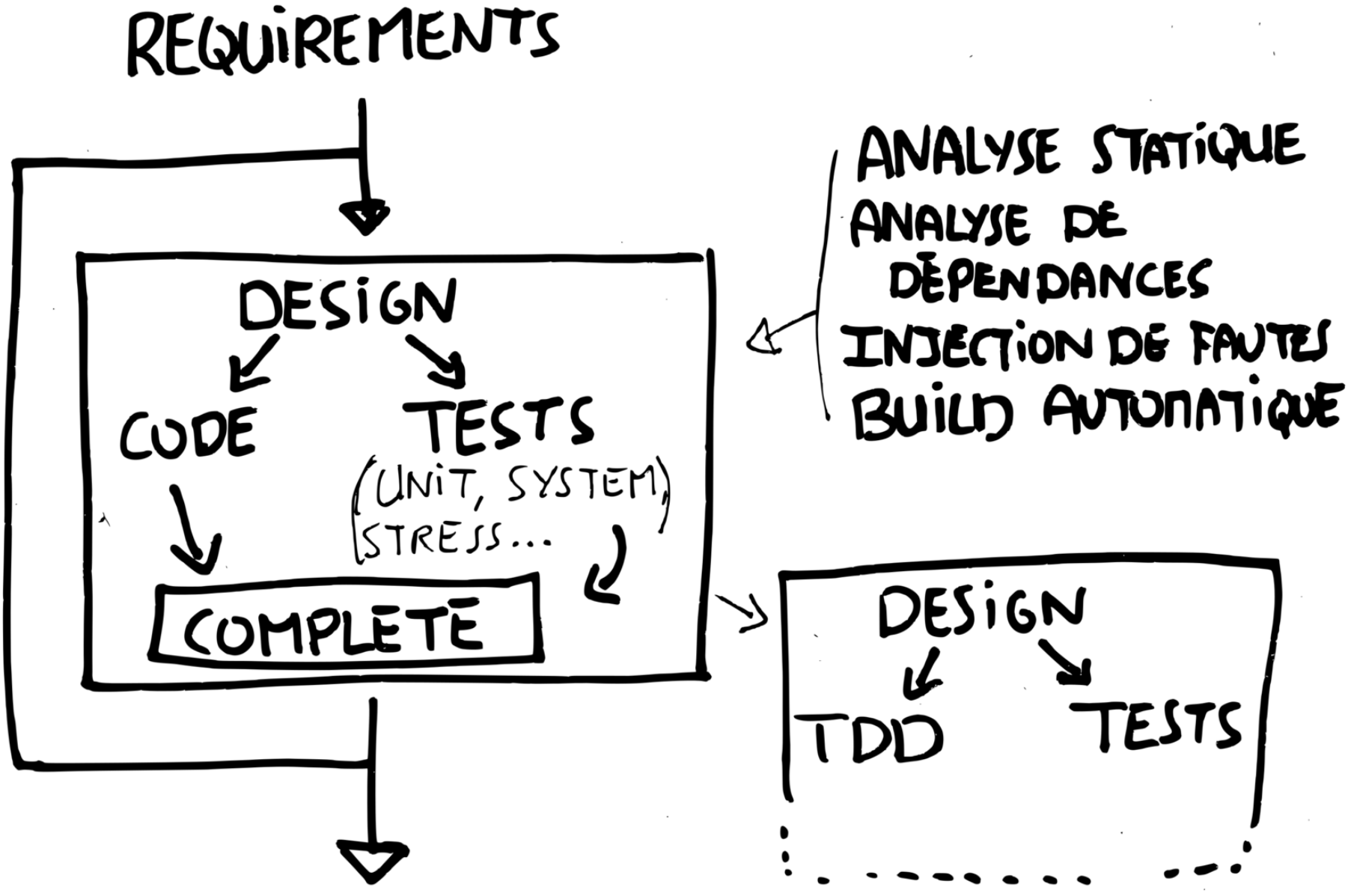
WINDOWS

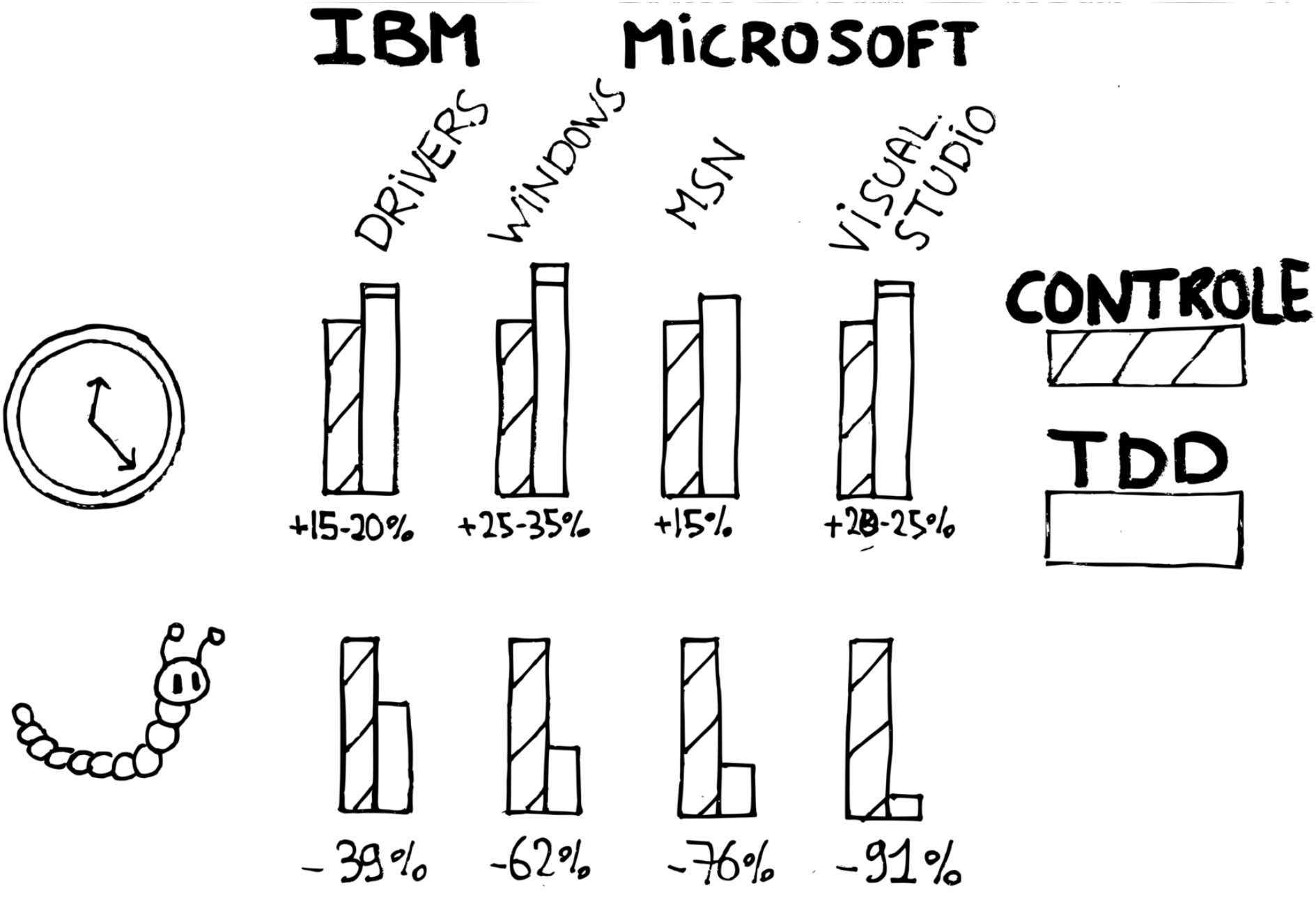


MSN



VISUAL STUDIO

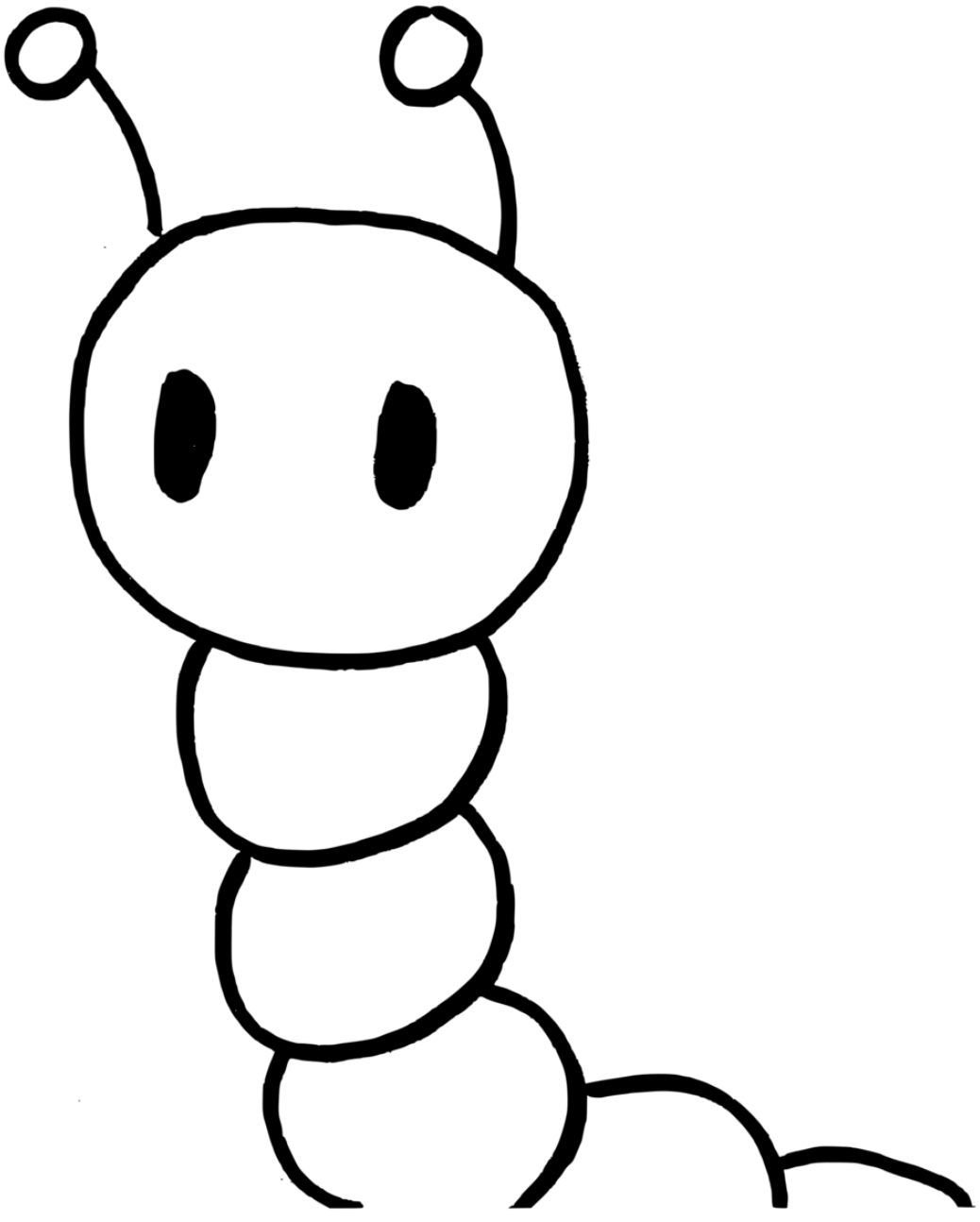

REQUIREMENTS
DESIGN
CODE
TESTS
(UNIT, SYSTEM, STRESS...
COMPLETE
ANALYSE STATIQUE
ANALYSE DE DÉPENDANCES
INJECTION DE FAUTES
BUILD AUTOMATIQUE
DESIGN
TDD
TESTS

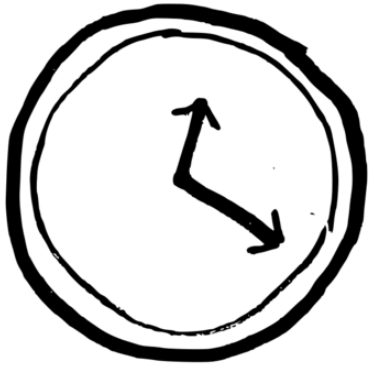

IBM
MICROSOFT
DRIVERS
WINDOWS
MSN
VISUAL STUDIO
CONTROLE
TDD
+15-20%
+25-35%
+15%
+20-25%
-39%
-62%
-76%
-91%

CONCLUSION

L’étude en question ne porte pas de conclusion sur l’efficacité de la pratique du TDD. Elle ouvre de nouvelles pistes pour continuer l’évaluation.

Pourquoi ?





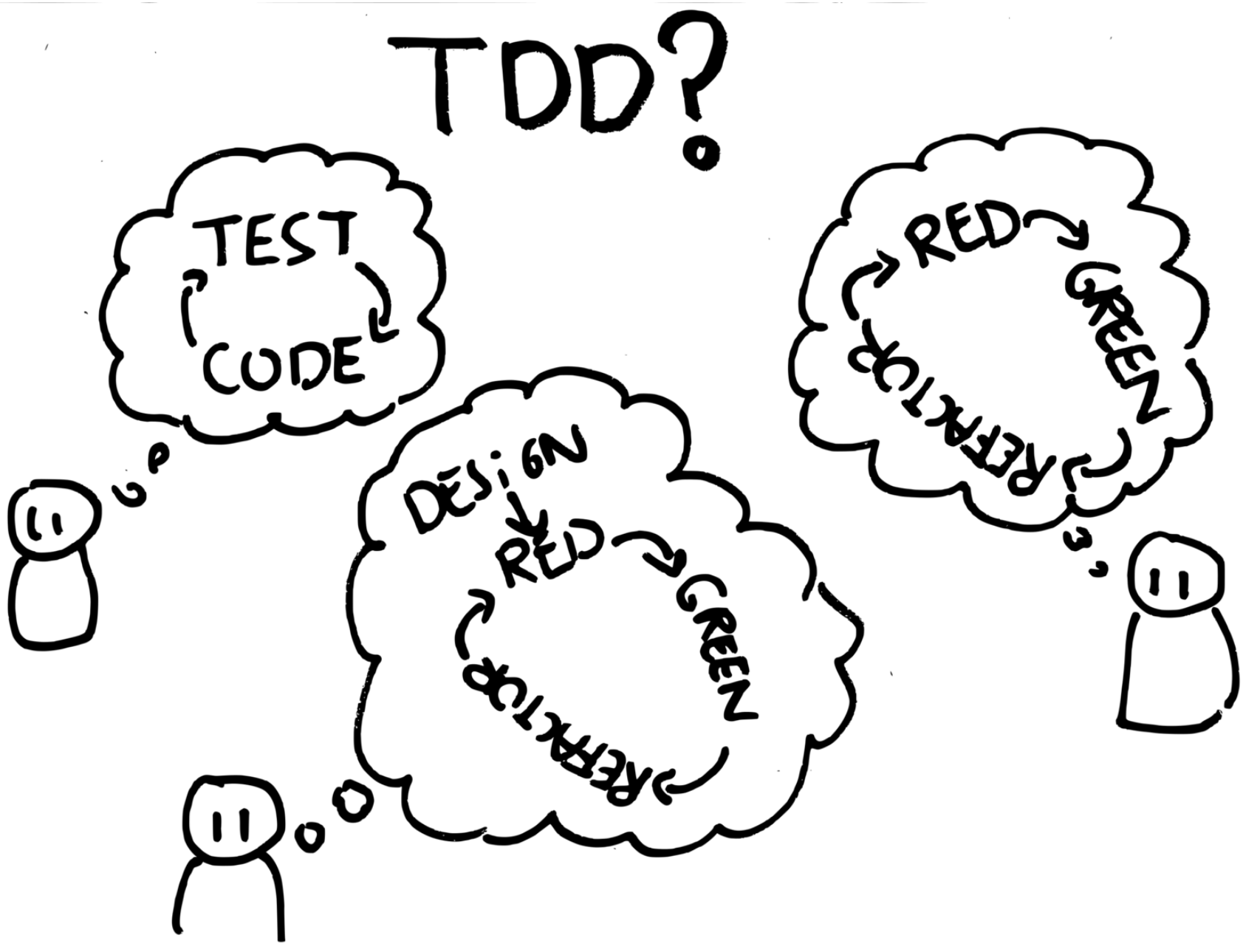
DEV
DEV
PRO
DUIT
CON
TROL

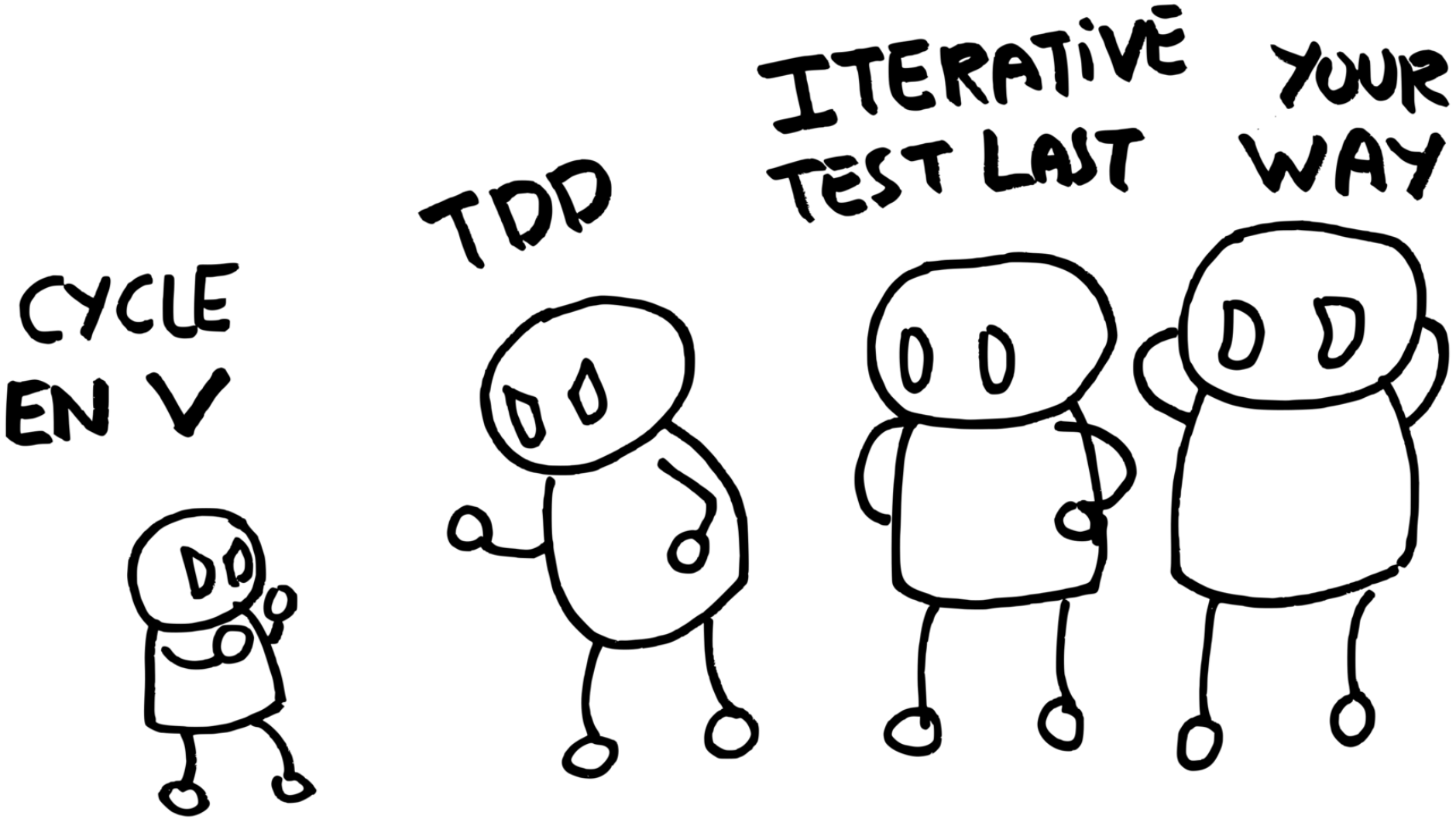
Why Research on Test-Driven Development is Inconclusive?

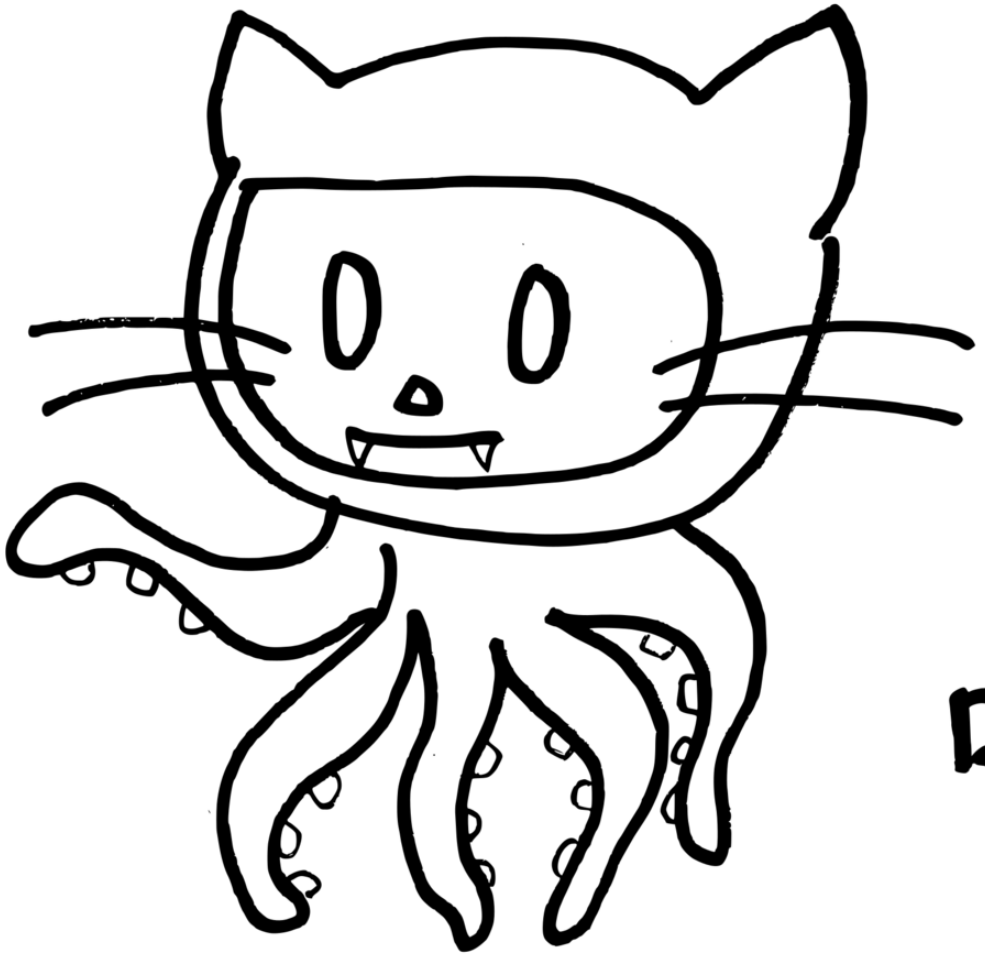
Mohammad Ghafari, Timm Gross, Davide Fucci, Michael Felderer

ESEM ’20, October 8–9, 2020, Bari, Italy

https://arxiv.org/pdf/2007.09863.pdf


TDD?
TEST
CODE
DESIGN
RED
GREEN
REFACTOR
RED
GREEN
REFACTOR


CYCLE EN V
TDD
ITERATIVE TEST LAST
YOUR WAY



~250 000

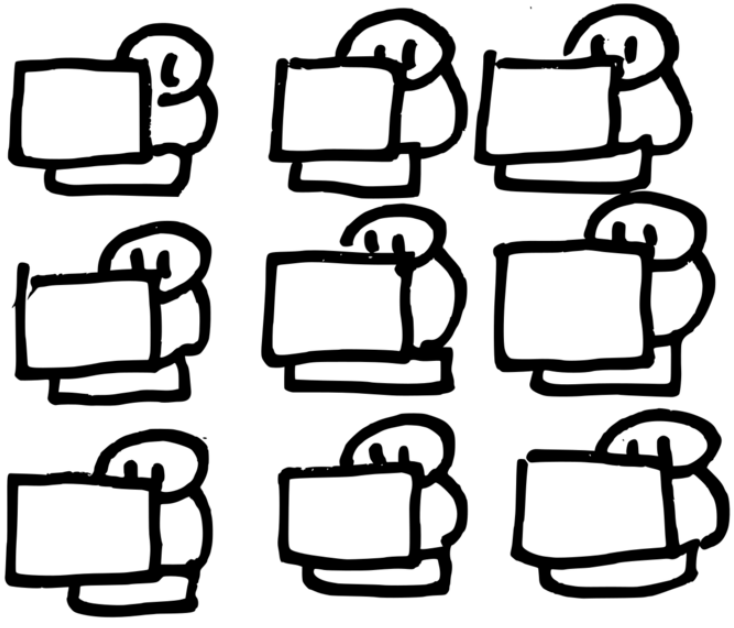
PROJETS GITHUB

0,8 %

CONTIENNENT

DES TESTS

~2500



2,2%
TDD
STRICT

CONCLUSION

The promise of TDD is that it should lead to more testable and easier to modify code. This makes it appealing from an indus- trial perspective, as developers spend half of their time dealing with technical debt, debugging, and refactoring with an associated opportunity cost of 85$ billion.

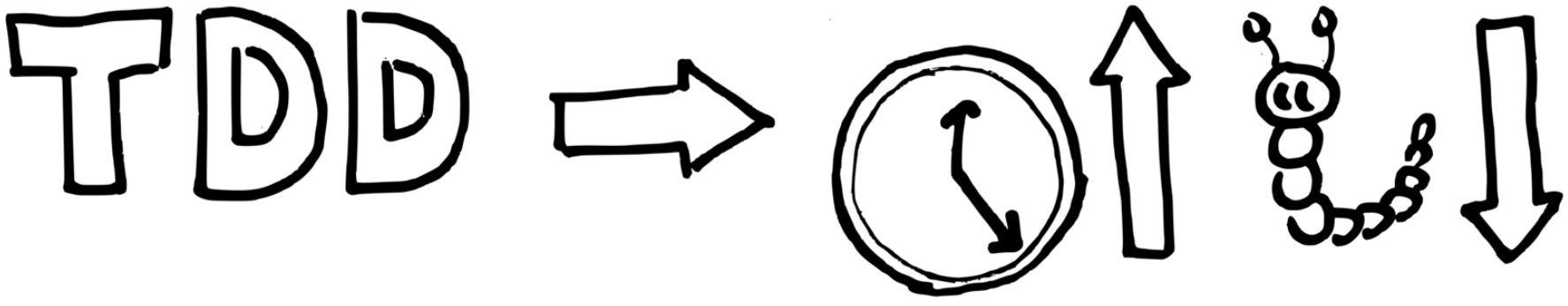
CONCLUSION

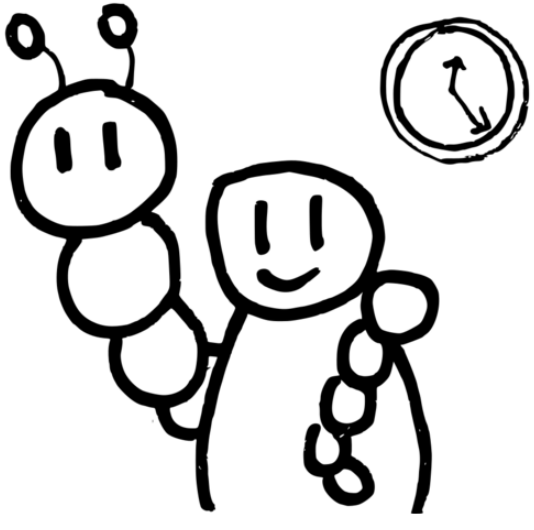
Nevertheless, the empirical evidence on TDD is contradictory, which hinders the adoption of this technique in practice.

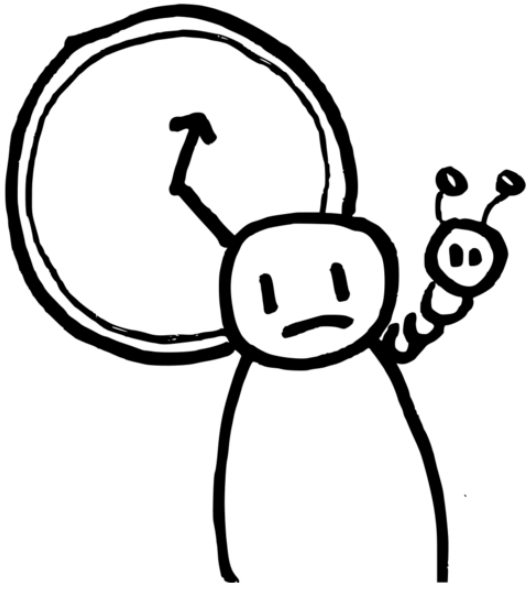
CONSÉQUENCES

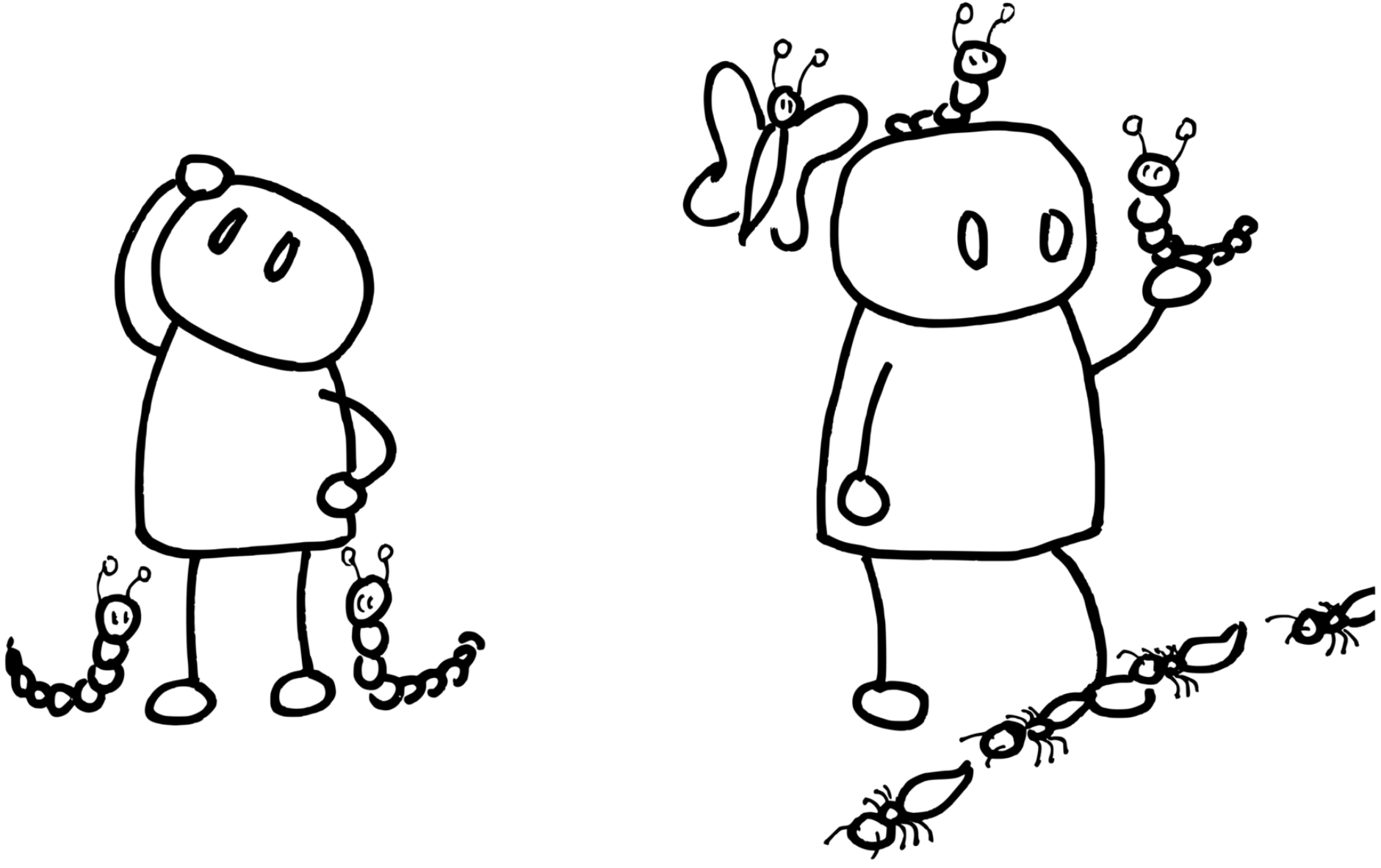
DIFFICULTÉ DE CONCLURE

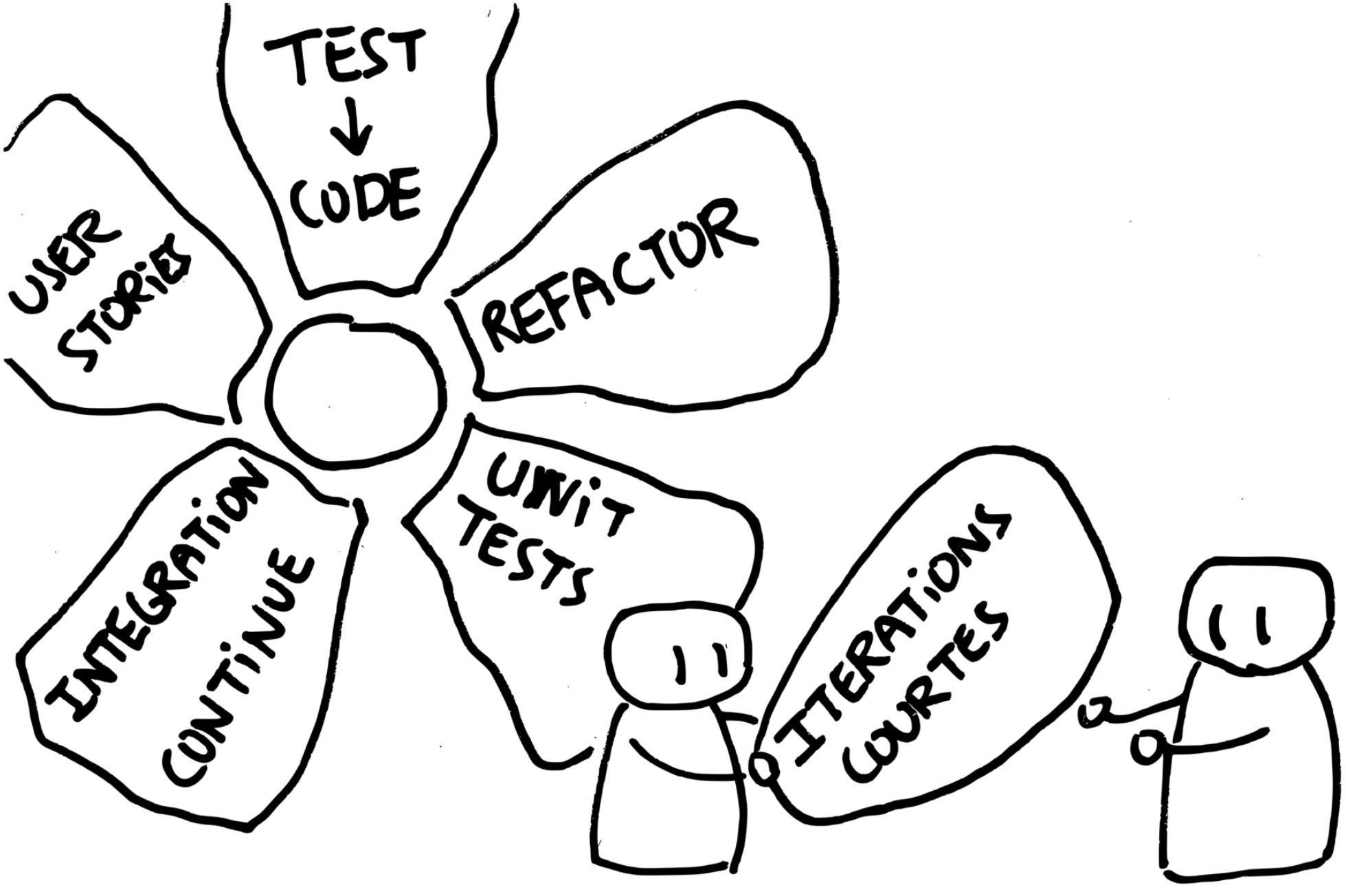
- Les résultats des recherches chamboulent un peu ma vision des choses
- Les résultats sont peu tranchés
- Par contre ils ouvrent à de nouvelles questions plus précises
- Et on obtient quand même des résultats utilisables…


TDD








TEST
↓
CODE
USER STORIES
REFACTOR
INTEGRATION CONTINUE
UNIT TESTS
ITERATIONS COURTES

ARTISANAT IS BACK !

- A mon niveau, je peux être subjectif !
- Observer mon contexte
- Facile d’identifier dans la communauté les personnes qui rejoignent mes intérêts
- Je peux écouter leur propos, complétés par le regard critique des résultats de recherche

LE MOT DE LA FIN

+ prudent

+ curieux

toujours aussi motivé par ces pratiques ! :-)

LE MOT DE LA FIN APRÈS LE MOT DE LA FIN

Merci de m'avoir écouté ! :-)

Merci à toute l'orga d'Agile Grenoble ! :-D

Des questions ? :-)

https://roti.express/r/ag20-077