The Economics of Legacy Code

terry@odd-e.com

Who Am I

- I am Terry Yin
- Work for Odd-e
- Experienced in software development
- Love programming
- A father
- Live in Singapore

What is Legacy Code?

The Odd-e CSD Classes

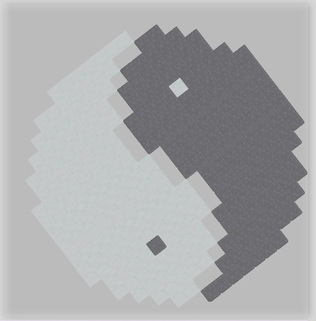
ONE WEEK, ONE SPRINT

THE WHOLE (10 PEOPLE) CLASS WORKS AS ONE TEAM.

**INSTRUCTORS ACT AS PRODUCT OWNER,
SCRUMMASTER AND TECH COACH**

UNLIKE THIS GENTLEMAN WHO ITERATES ONLY FOR FUN...

THE TEAM ITERATES TO DELIVER WORKING FEATURES CONTINUOUSLY ALONG WITH THE FUN.

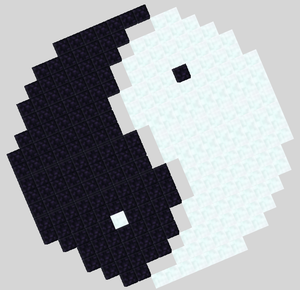


IN EACH CSD CLASS ...

IN EACH CSD CLASS ...

TEAM CONTINUES TO BUILD THE SAME PRODUCT ON THE SAME CODEBASE.

SOFTWARE DEVELOPMENT IS MOSTLY ABOUT KNOWLEDGE ACQUISITION AND PRESERVATION



KNOWLEDGE OF THE SOFTWARE PRODUCT

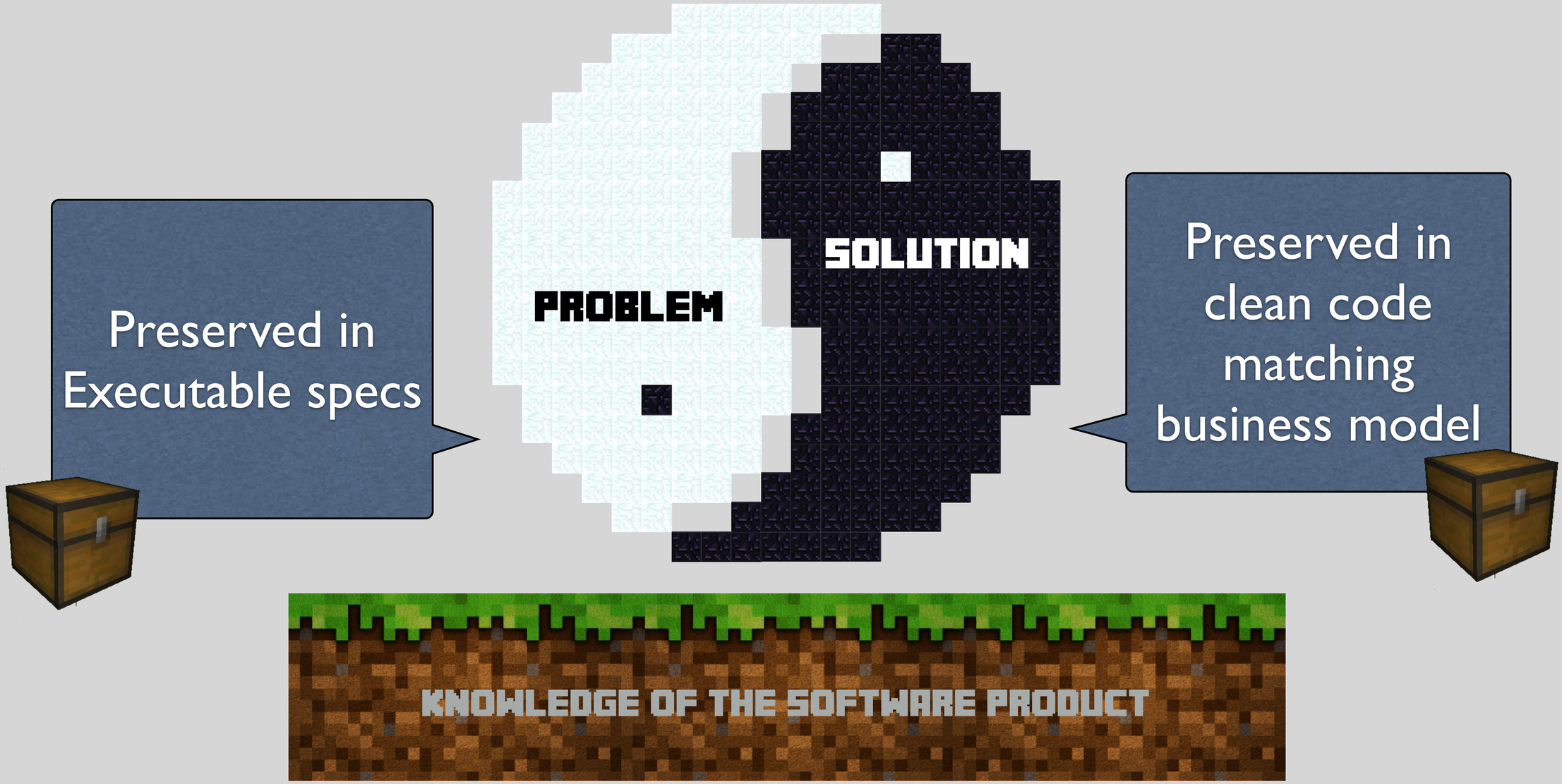
Knowledge acquired by understanding

PROBLEM

SOLUTION

Knowledge acquired by discovering

KNOWLEDGE OF THE SOFTWARE PRODUCT


Preserved in Executable specs
PROBLEM
SOLUTION
Preserved in clean code matching business model
KNOWLEDGE OF THE SOFTWARE PRODUCT

Odd-e

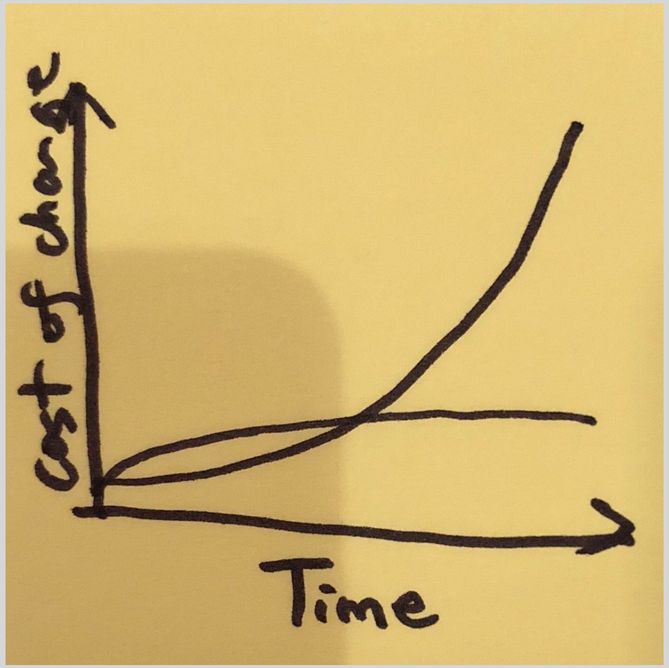
LEGACY CODE

SOFTWARE DEVELOPMENT IS MOSTLY ABOUT

KNOWLEDGE ACQUISITION AND

~~PRESERVATION~~

Cost and Interest



3 Perspectives

Automated (Unit) Test

Architecture

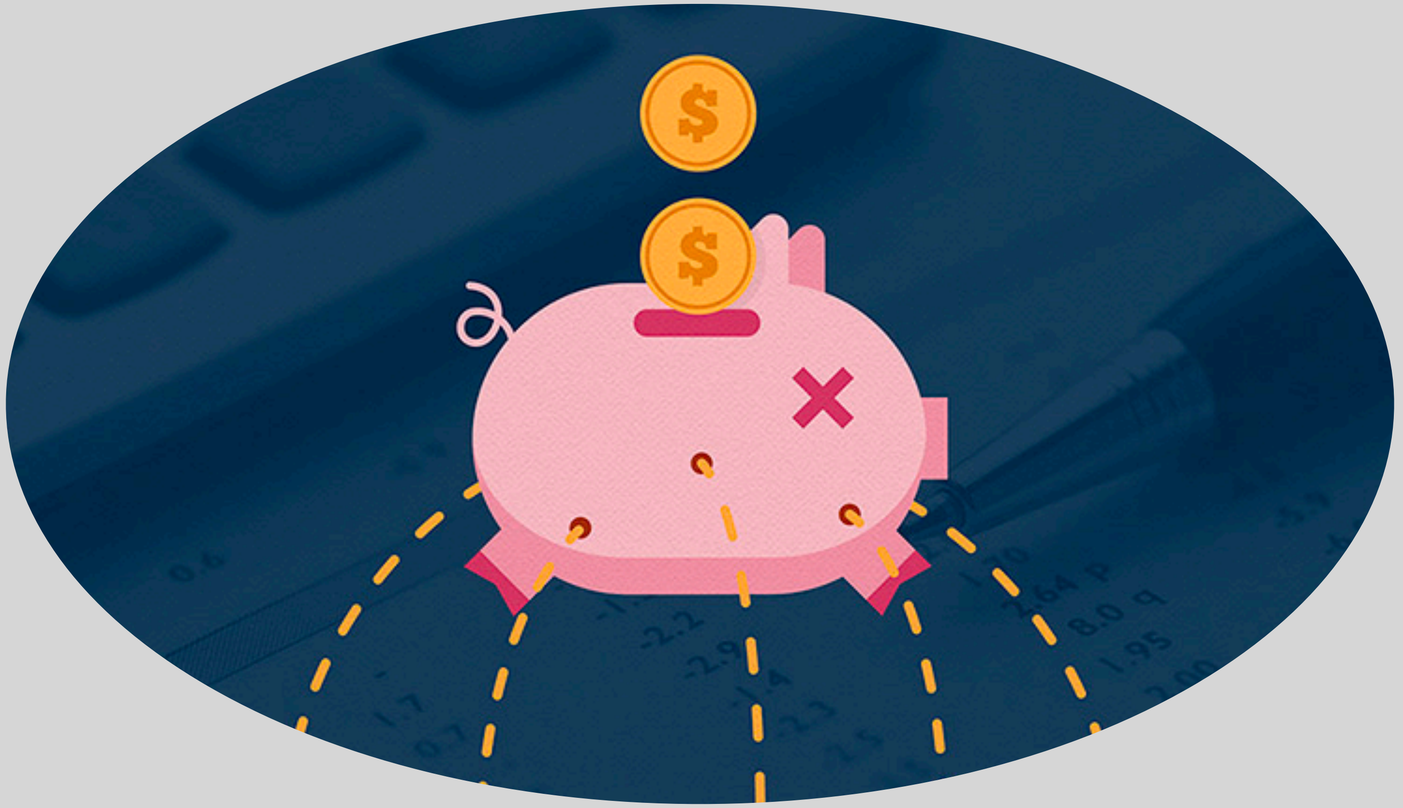
Organization

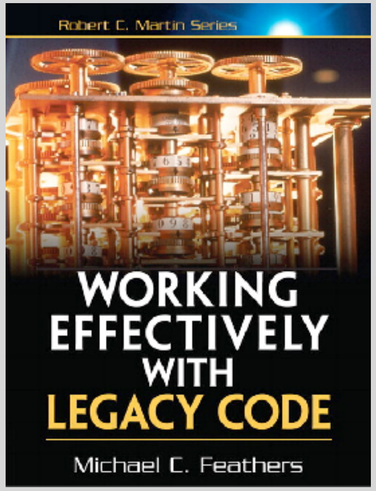
#1

Legacy Code
and

Automated (Unit) Test

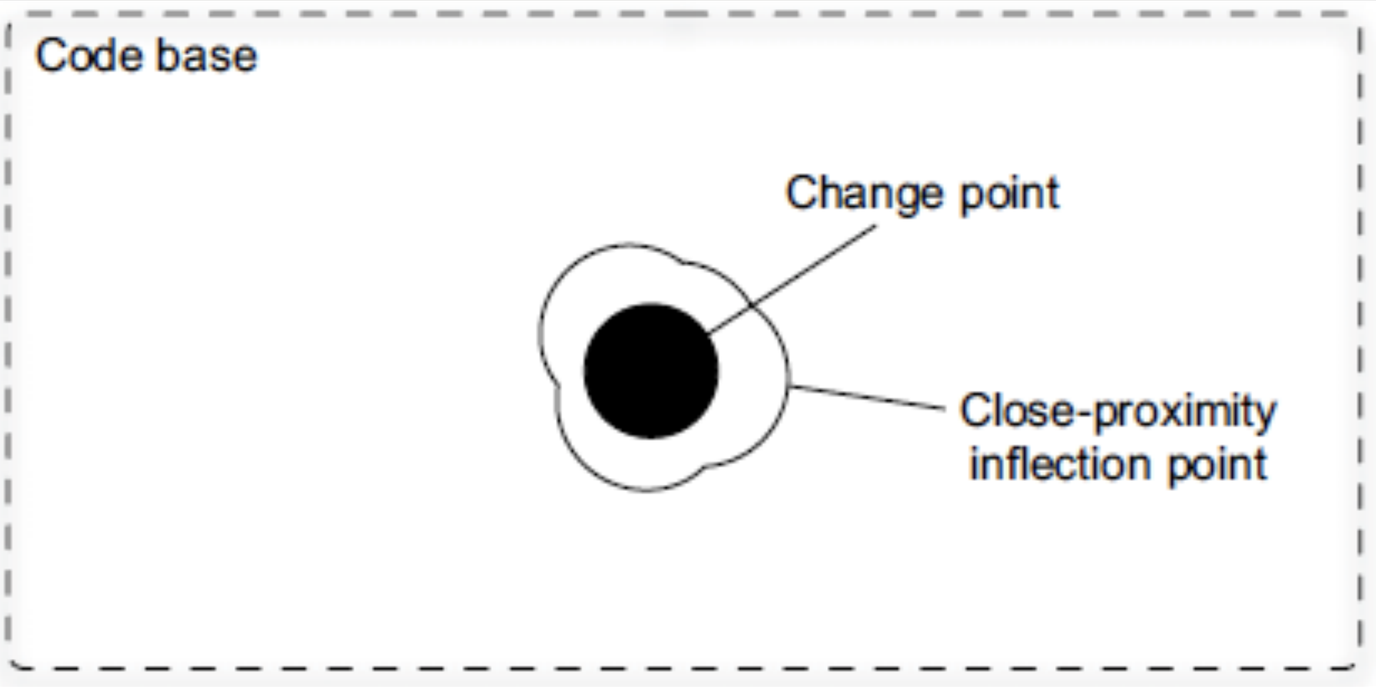
Code With No Test





What is the first thing to do,
when you find yourself in a hole?

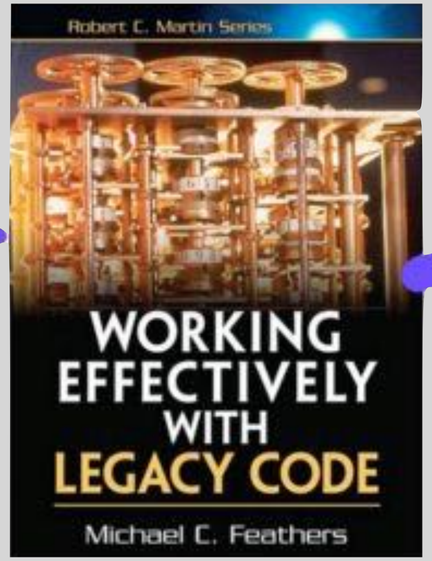
Legacy code algorithm



1. Identify change point
2. Find test points
3. Break dependencies
4. Write tests
5. Make changes and refactor

Challenge from working with legacy code: Breaking Dependencies

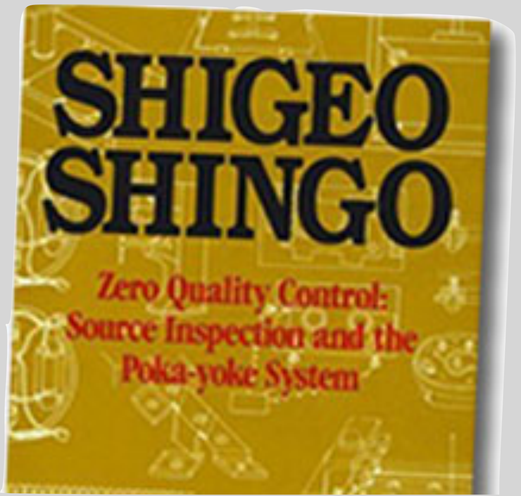
Sensing



Separation

Autonomation vs Automation

自働化 ≠ 自動化



#2

Legacy Code
and

Architecture

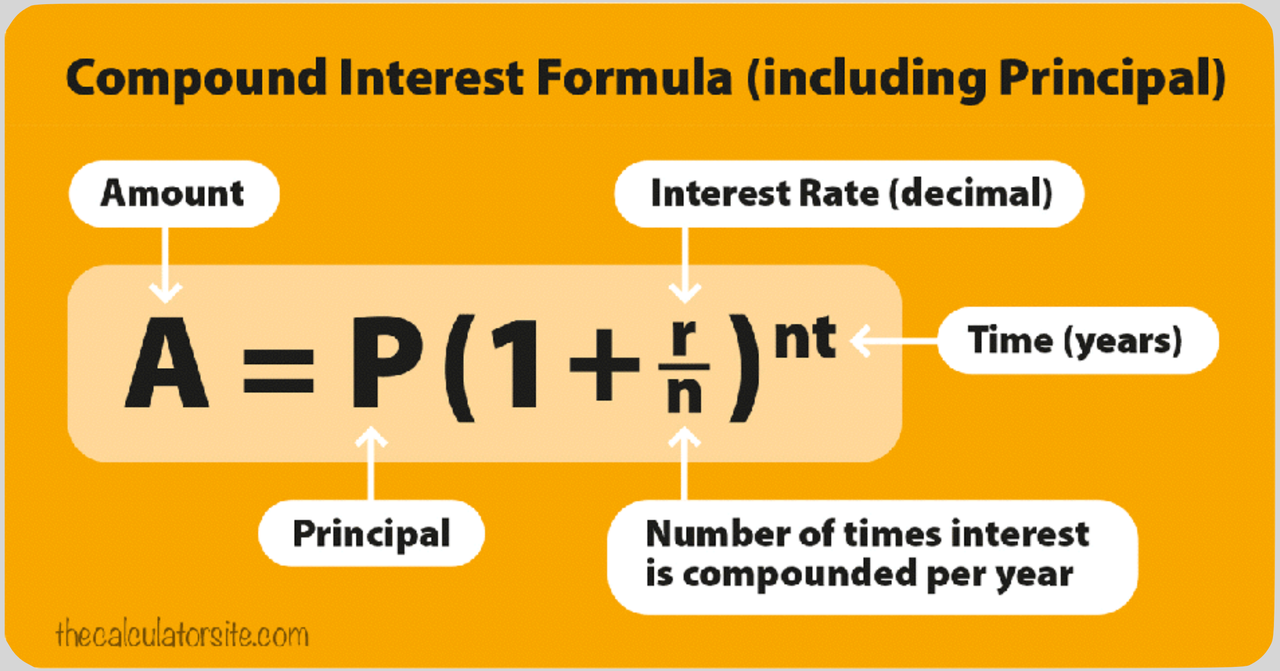
High Cohesion, Low Coupling

But high cohesion has higher priority

Craig Larman

High Cohesion

Cost of Not Having High Cohesion

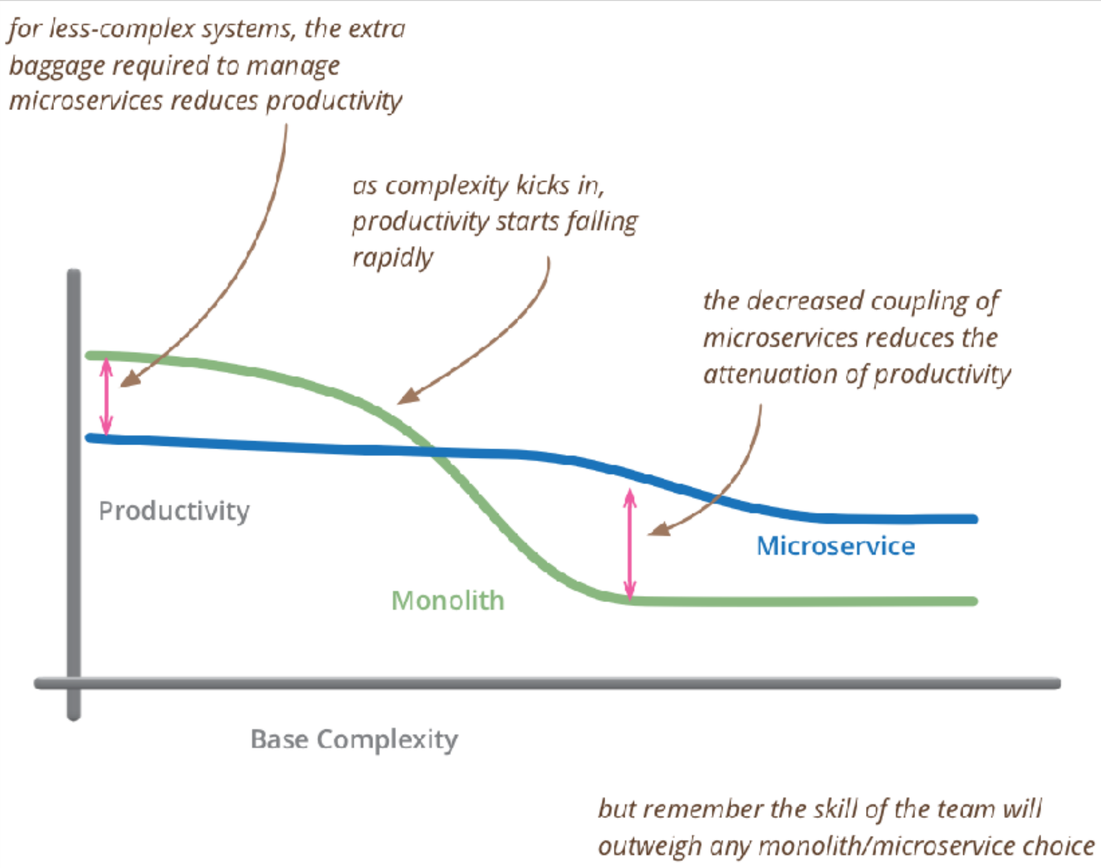


Compound interest

Low Coupling

Cost of Not Having Low Coupling?

Microservice



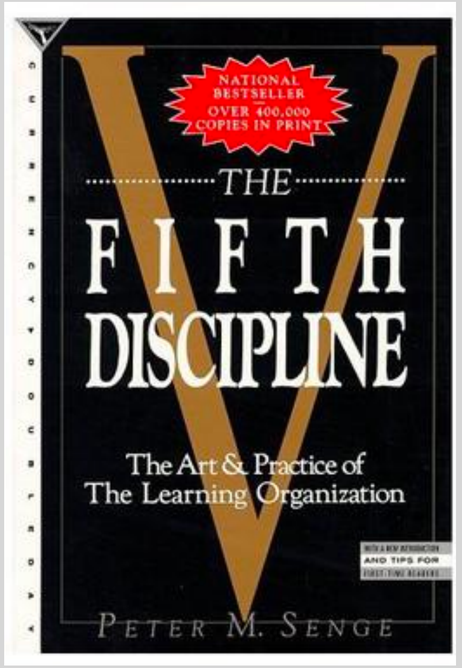
Picture from: https://www.martinfowler.com/

Martin Folwer

#3

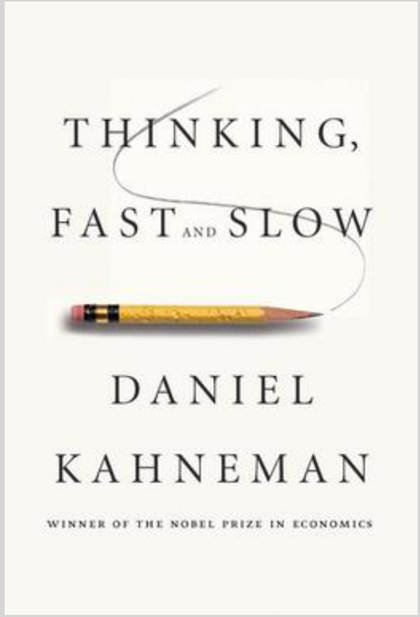
Legacy Code
and

Organization

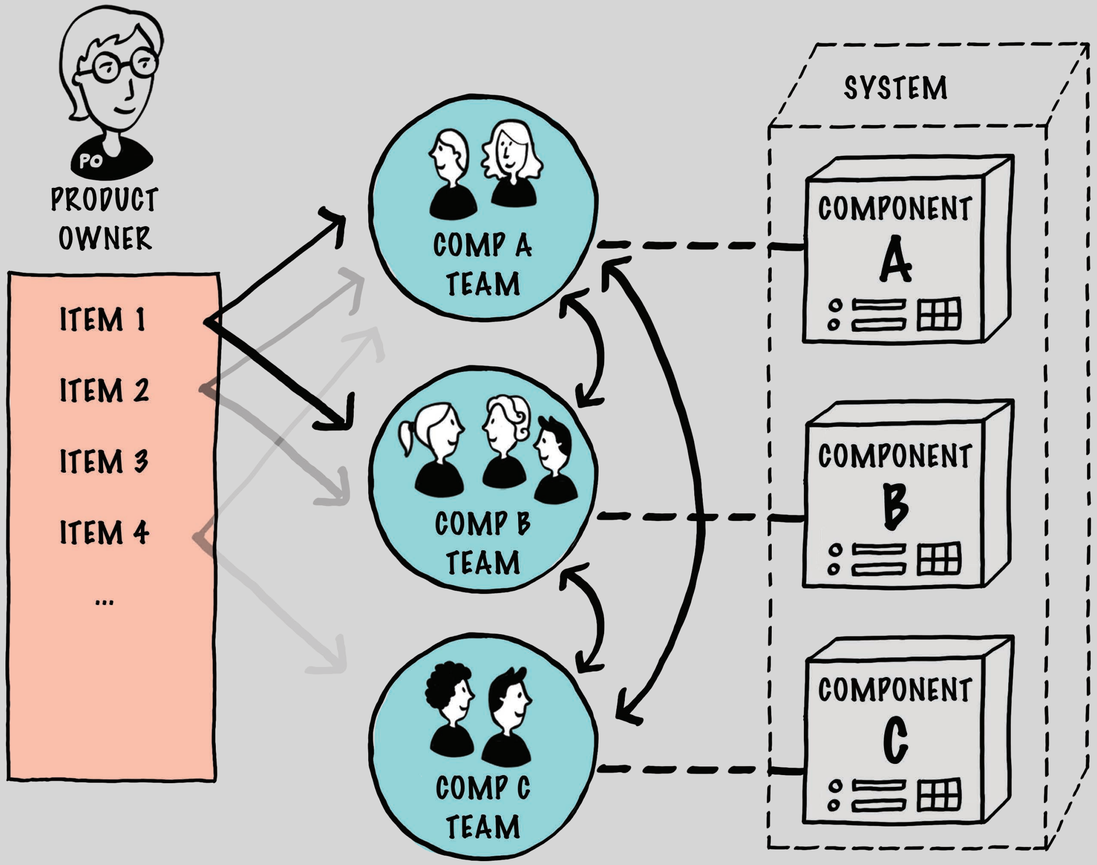


Global optimisation vs.
Local optimisation

Human instinct are often prone to local optimisation



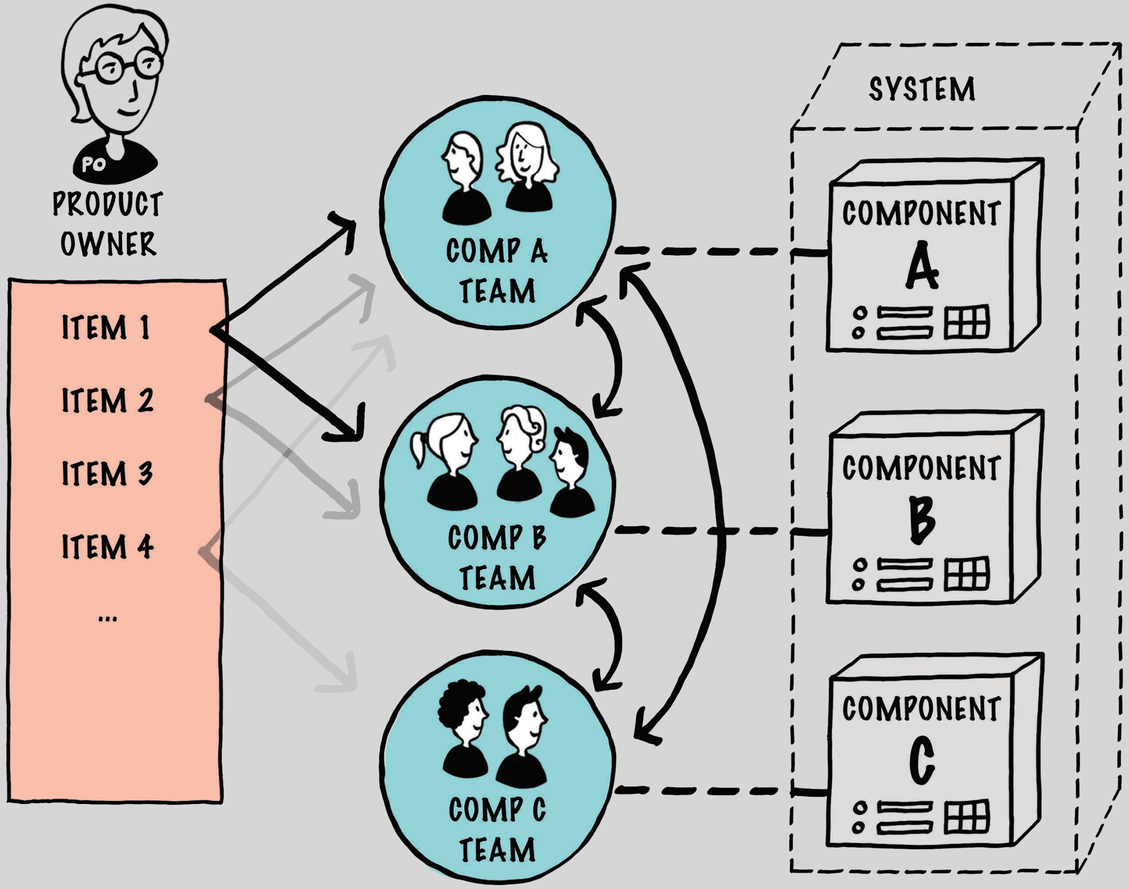
Conway's Law



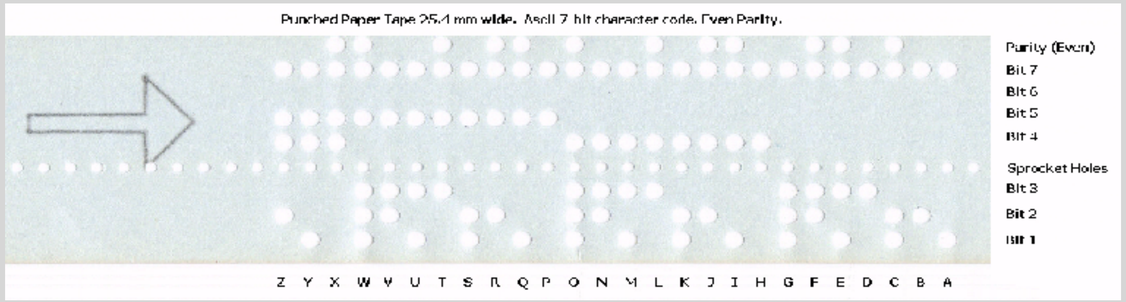
"organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations."

Melvin Conway, 1967

Following Conway's Law — Component Team



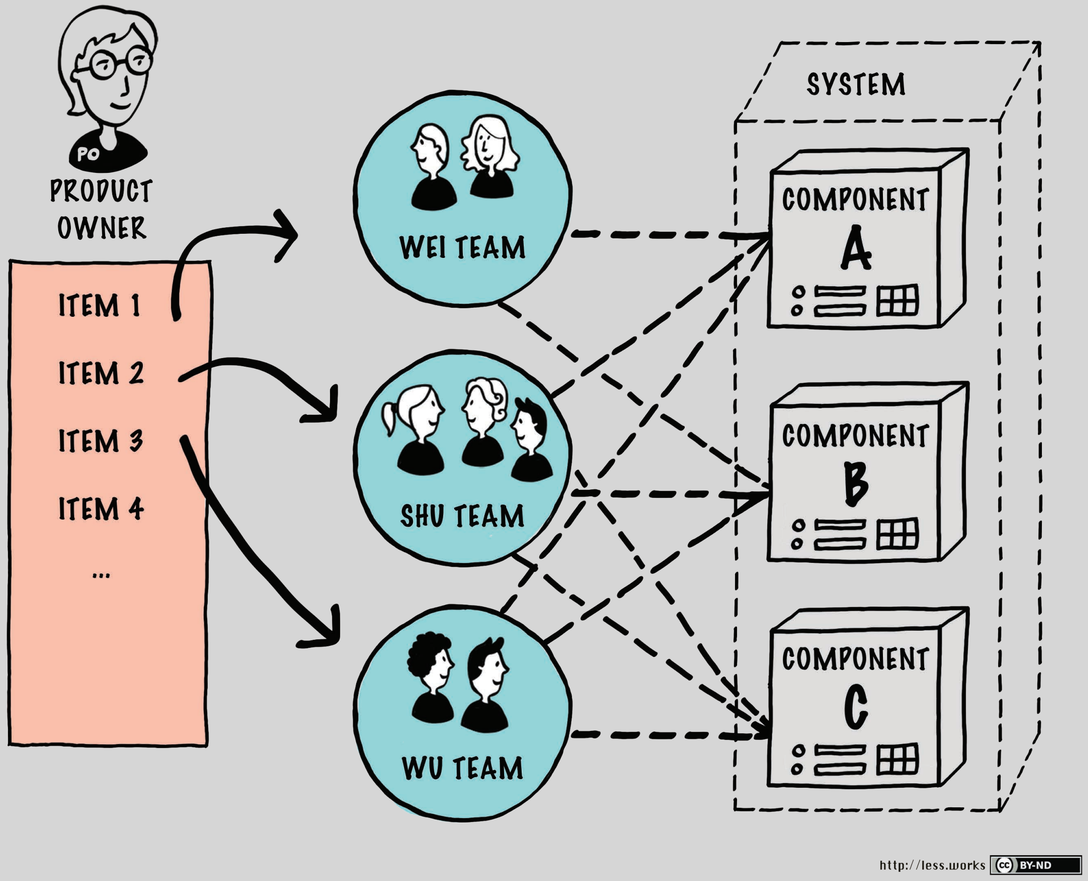
Trying to Preserve knowledge by organisational structure



vs.

Representational Gap

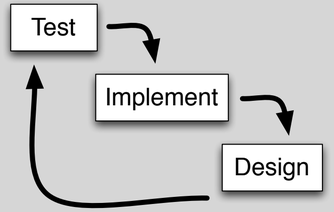
Business Domain vs. Solution Domain



Lower representational gap

Less legacy code

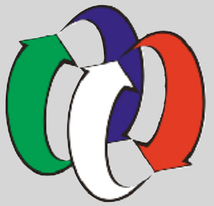
Collective Code Ownership Needs Supporting Practices



Test Driven Development

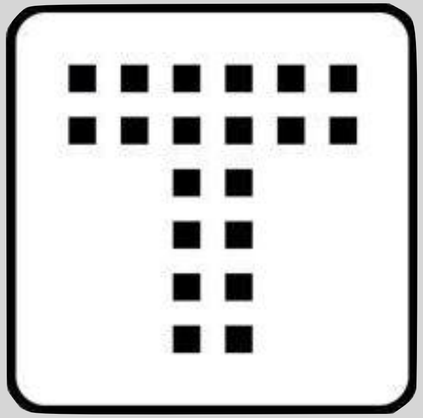
Pair Programming

Refactoring



Continuous Integration

Acknowledgement



“91” Joey Chen