

Identifying concept inventories in agile programming

Christopher Fuhrman (christopher.fuhrman@etsmtl.ca)
École de technologie supérieure (Montréal, Canada)



Stéphane Ducasse (stephane.ducasse@inria.fr)
Institut national de recherche en sciences et
technologies du numérique (Lille, France)



Introduction

Programme : Samuel de Champlain (volet *formation*)

« Identification d'inventaire de concepts dans le développement agile »

**Fonds de recherche
Nature et
technologies**

Québec 



Objectives

- Strengthen the collaboration between INRIA/Lille (France) and ÉTS/UQAM (Québec).
- Create a concept inventory for learning and teaching agile development with OO languages.
- Improve our courses

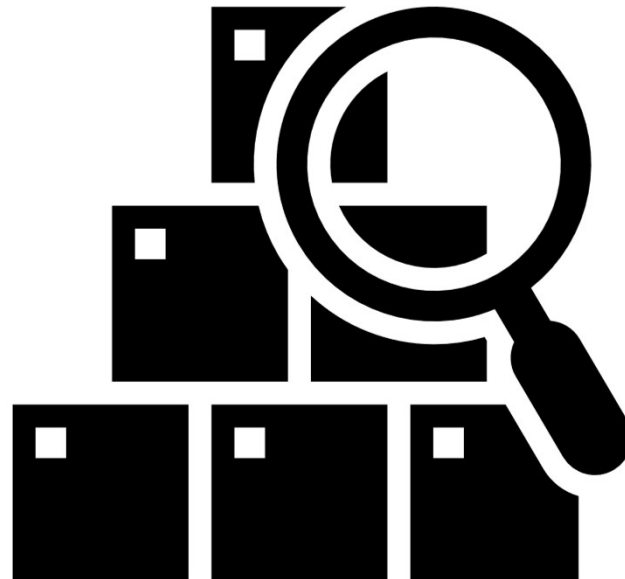
Team

- ÉTS (software engineering)
 - **Prof. Christopher Fuhrman**
 - Prof. Roberto Erick Lopez-Herrejon
- UQAM (neuroscience, education → computer science)
 - Prof. Julien Mercier
- INRIA/Lille (développement agile, orienté objet, Pharo.org)
 - **Prof. Stéphane Ducasse**
 - Dr. Guillermo Polito



Project

Create a concept inventory (CI) for teaching OO and agile development methods

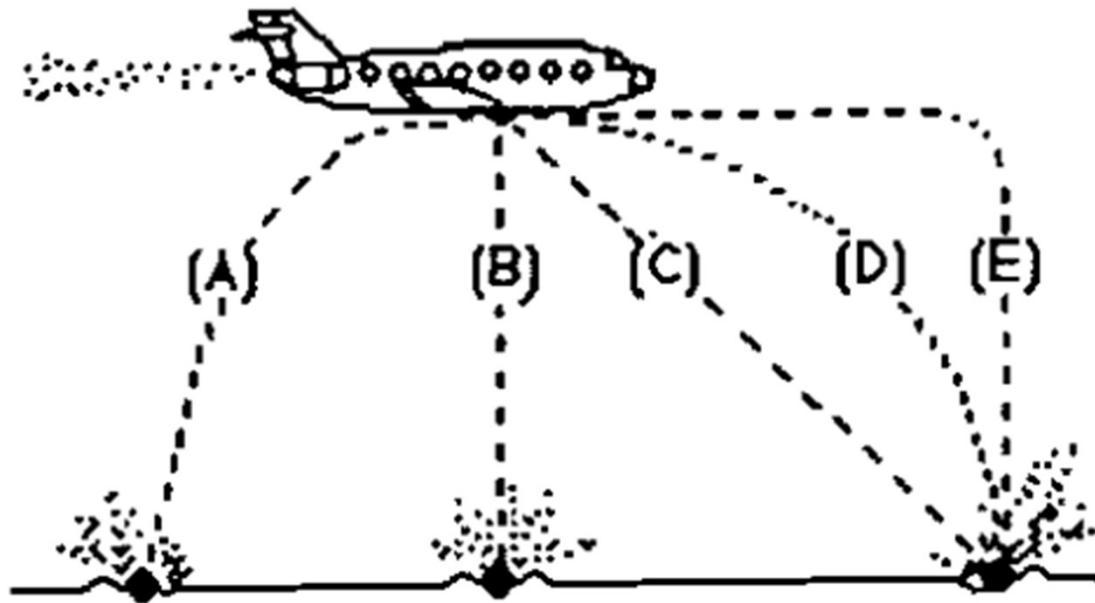


Concept inventory

- Initially proposed by physics professors, e.g., Hestenes, Wells, et Swackhamer (1992) : *Force concept inventory (FCI)*
- Standardized test
- The FCI measures the understanding of fundamental concepts in physics, such as **force**, **movement** and **energy**.

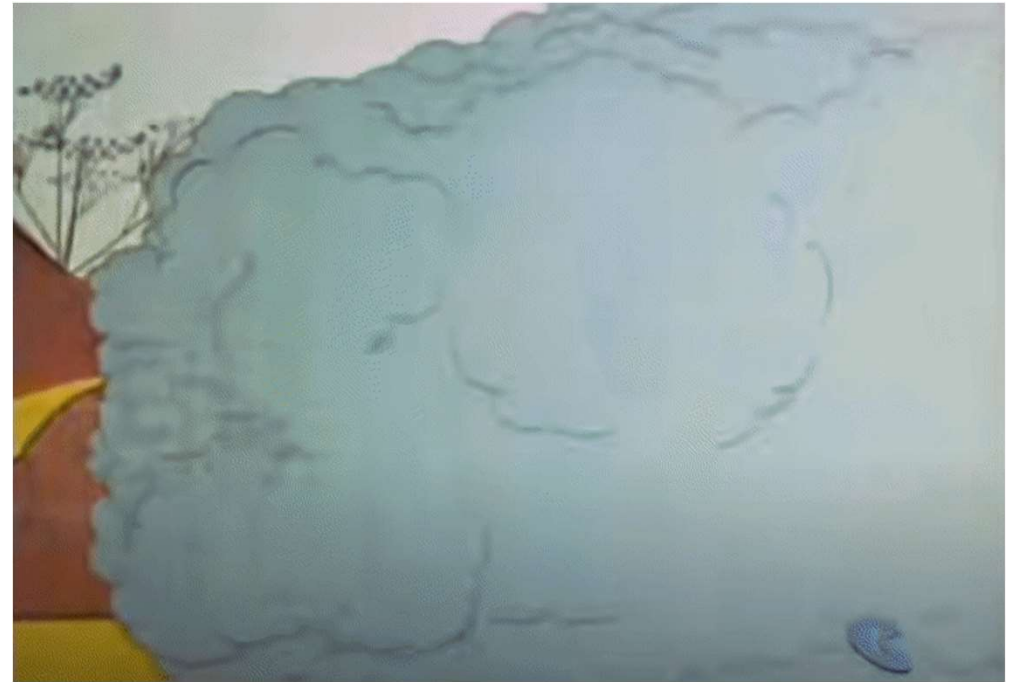
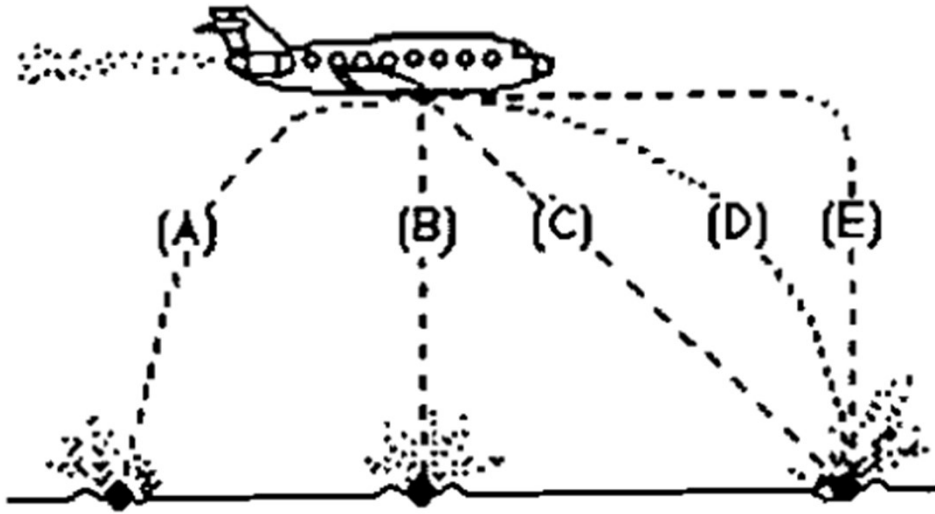


Question 23 from the FCI (1992)



Question 23. A bowling ball accidentally falls from the cargo hold of a jetliner in flight. From the perspective of an observer on the ground, which path will the bowling ball take after falling from the plane? (Hestenes, Wells, et Swackhamer 1992)

Misconception



Example of a misconception in Java

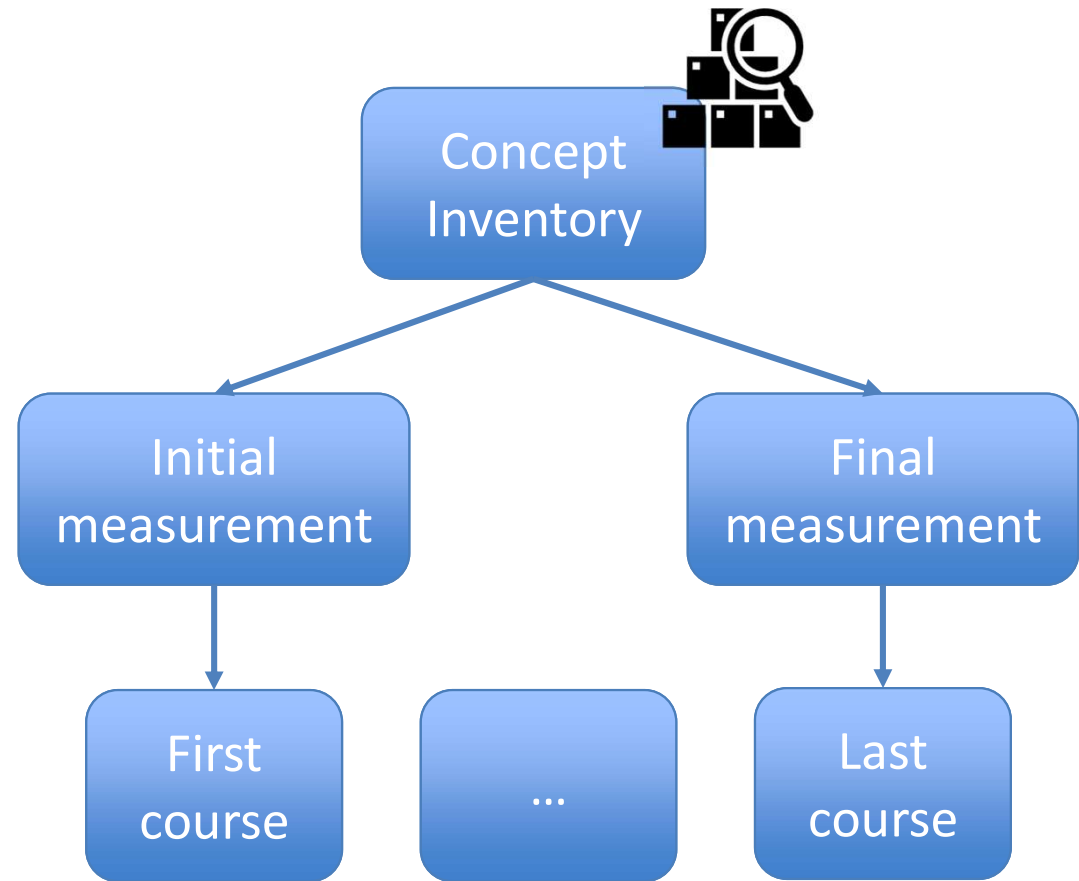
Example from Caceffo et al. (2019) :

```
int i = 0;
int sum = 0;
while (i < 10) {
    sum = sum + i;
    i = 1;           // problem
}
System.out.println(sum);
```

Ligne 5 : bad assignment of looping variable

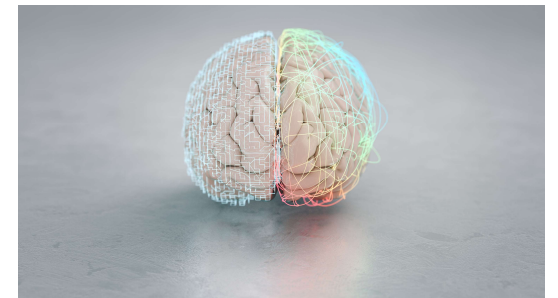
Concept inventory

- Multiple-choice Quiz with questions about concepts
- Measurement of understanding of fundamental concepts by learners



Our project (Inria and ETS/UQAM)

- Agile development
- OO languages
 - TypeScript/JavaScript
 - Pharo
- Scientific validation
 - Data from the MOOC Pharo (INRIA/Lille)
 - Data from programming courses (ÉTS)
 - Neuroscientific (cognitive) measures (UQAM)



Methodology

- **O1: Identify misconceptions**
- O2: Propose a concept inventory (CI).
- O3: Deploy and validate the CI in courses.



Workshop goals

- Identify misconceptions in SmallTalk/Pharo
 - Capture in a collaborative tool
 - <https://notes.inria.fr/s/DIIUXSVXI>

How to find misconceptions?

- Instructors will know the “common pitfalls” from seeing them with students.
- Personal troubles you had initially in SmallTalk.
- A source of misconceptions comes from “interference” from other languages ([Learning strategies and interference themes](#))

Language Interference

| THEME | DESCRIPTION | REPRESENTATIVE EXAMPLES |
|---|--|--|
| <i>Old habits die hard</i> (Section 3.3.1) | Programmers had to constantly suppress old habits from previous languages. | “I’m typing a[1] thinking that it’s a[0].” “I still type the type first before the variable.” “I’m gonna make it an object for this, no don’t do that!” |
| <i>Mindshifts when switching paradigms</i> (Section 3.3.2) | Sometimes programmers wrestled with larger differences that required fundamental shifts in mindsets, or “mindshifts.” | “All my assumptions were thrown out the window.” “I had to rethink the problem and re-implement it.” “There are lots of events and promises all these things makes it really hard to debug.” |
| <i>Little to no mapping with previous languages</i> (Section 3.3.3) | Programmers had a harder time learning the new language when there was little to no mapping of features to previous languages. | “There’s a very alien concept in Rust that is the borrow checker.” “I’ve never had a language with traits before.” “I did not work with concepts like virtual DOM, shadow DOM before.” |
| <i>Searching for terms and documentation is hard</i> (Section 3.3.4) | Programmers found it difficult to search for information about the language and its associated technologies. | “You don’t even know what exists, what to even look for.” “Scala is not that common. Some of it required a little deeper digging.” “They have their own convention, TypeScript has its own convention, JavaScript has its own convention.” |
| <i>Retooling is a challenging first step</i> (Section 3.3.5) | Programmers faced difficulty retooling themselves in the environment of the new language. | “I was using Visual Studio to debug C# code and now it’s gdb to debug C++ code.” “In Xcode, build targets aren’t ‘Universal’ in definition like .NET.” “The problem is IntelliJ is aimed at the Java developer and I’m using SBT which is from the Scala world.” |

References

Caceffo, Ricardo, Pablo Frank-Bolton, Renan Souza, et Rodolfo Azevedo. 2019. « Identifying and validating java misconceptions toward a CS1 concept inventory ». In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 23-29.

<https://dl.acm.org/doi/pdf/10.1145/3304221.3319771>.

Hestenes, David, Malcolm Wells, et Gregg Swackhamer. 1992. « Force concept inventory ». *The physics teacher* 30 (3): 141-58.

<https://pubs.aip.org/aapt/pte/article/30/3/141/270140/Force-concept-inventory>.