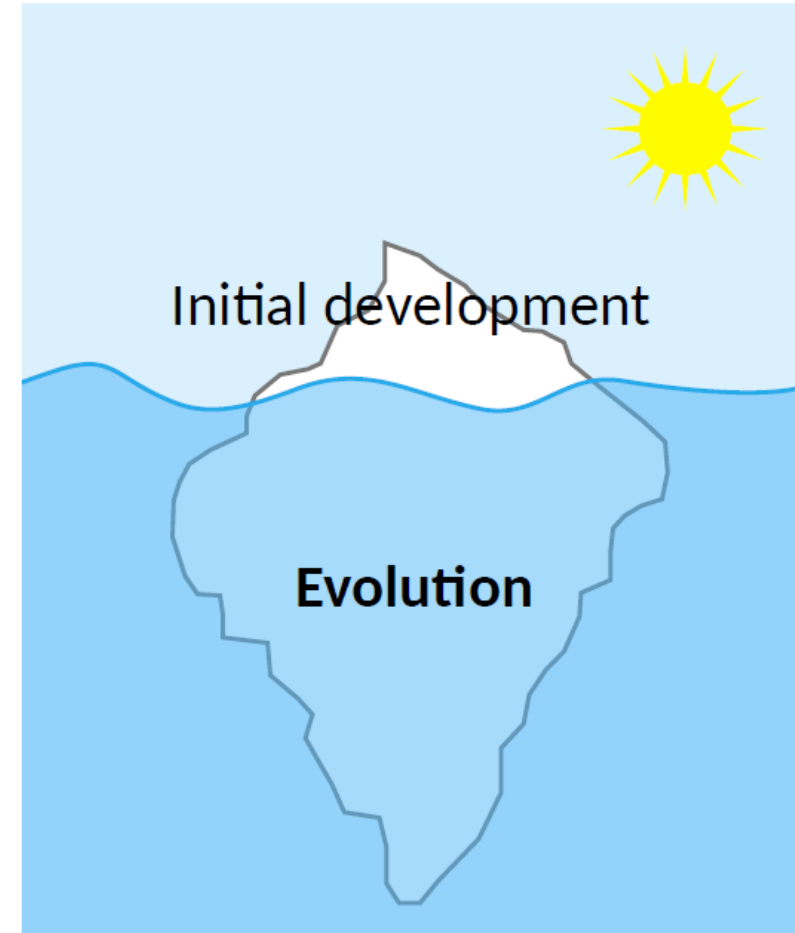
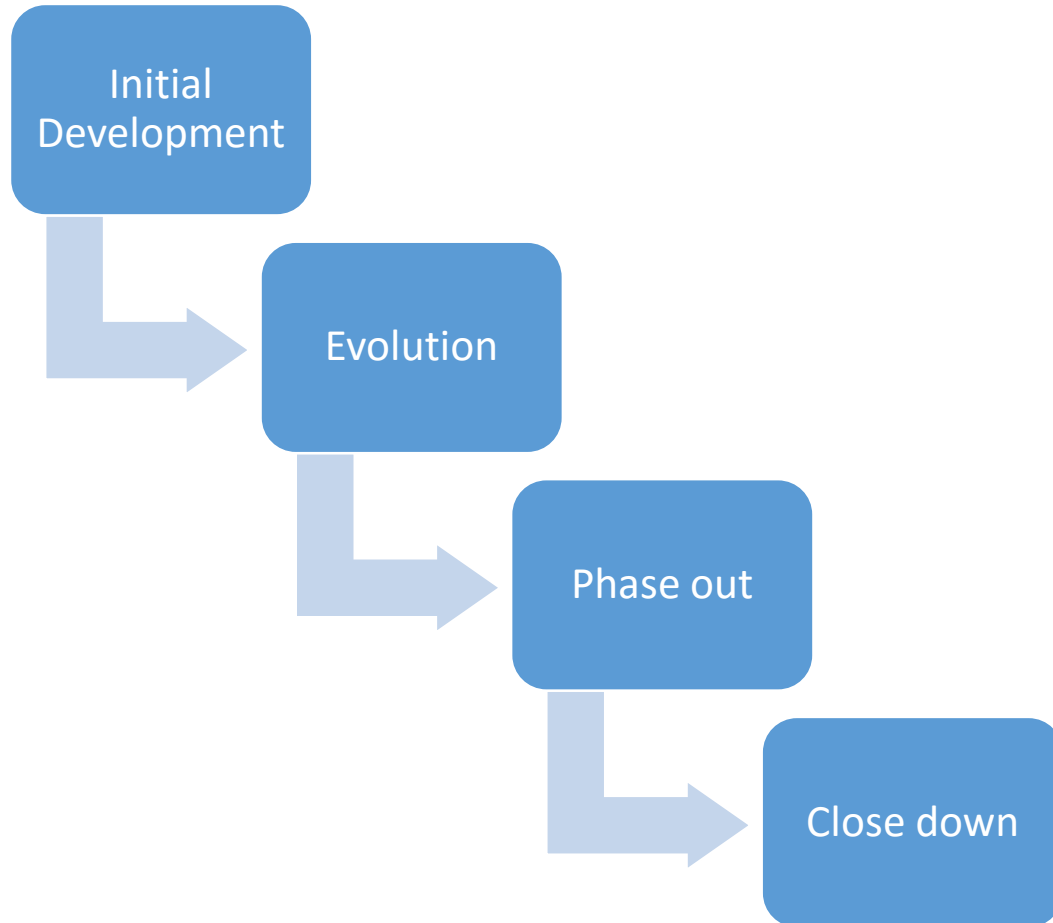


# Software Evolution Research@LaBRI

Pr. Xavier Blanc – Bordeaux Université – Institut Universitaire de France



# Software Evolution



# Research on Software Evolution

- Objectives
  - Understanding
    - Better understand what software evolution is about.
  - Support
    - Furnish tools to practitioners to face software evolution
- Approaches
  - Diff Analysis
  - MSR studies

# Change Analysis - GumTree

# Understanding Source Code Changes

- Evolution analysis
  - Source code evolution analysis, mainly to understand evolution
  - 46% of surveyed approaches use at most a textual differencing analysis

=> But textual differencing diff analysis is limited

# Textual differencing

- Source code model is a sequence of text lines
- Possible actions are
  - Delete a text line
  - Insert a text line
- Problem solved: find one shortest sequence of actions
  - Transforming the previous source code into the current source codes

# Current approaches

```
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example"
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies by 10");
        System.out.println("And displays it");
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        System.out.println(i);
    }
}
```

```
import java.util.Random;

public class Example {

    public void hello() {
        System.out.println("Hello everybody!");
        System.out.println("This code is a magnificent example"
        System.out.println("For the ASE 2014 conference");
        System.out.println("It draws a number at random");
        System.out.println("Adds 10");
        System.out.println("Multiplies by 10");
        System.out.println("And displays it");
        int i = random();
        System.out.println(i);
    }

    public int random() {
        Random r = new Random();
        int i = r.nextInt();
        i += 10;
        i *= 10;
        return i;
    }
}
```

Don't detect moves

Not aligned on the code

# A NP-Hard problem

- Three choices
  - Source code model
  - Set of possible actions
  - Problem solved
- Leads to NP-hard problems
  - Textual diff with move is NP-hard
  - Labelled graph diff is NP-hard, even with basic actions
  - Unordered tree diff is NP-hard, even with basic actions

# GumTree

- Source code model
  - A labelled ordered rooted tree (AST)
- Possible actions
  - Node insertion
  - Node deletion
  - Node relabel
  - Subtree move
- Problem solved
  - Short sequence that corresponds to a developer intent

# GumTree process

1. Parse code files to our code agnostic tree structure
- 2. Find mappings between nodes**
  - Like developers proceed
3. Deduce code edition actions
4. Output code difference

# Finding mappings

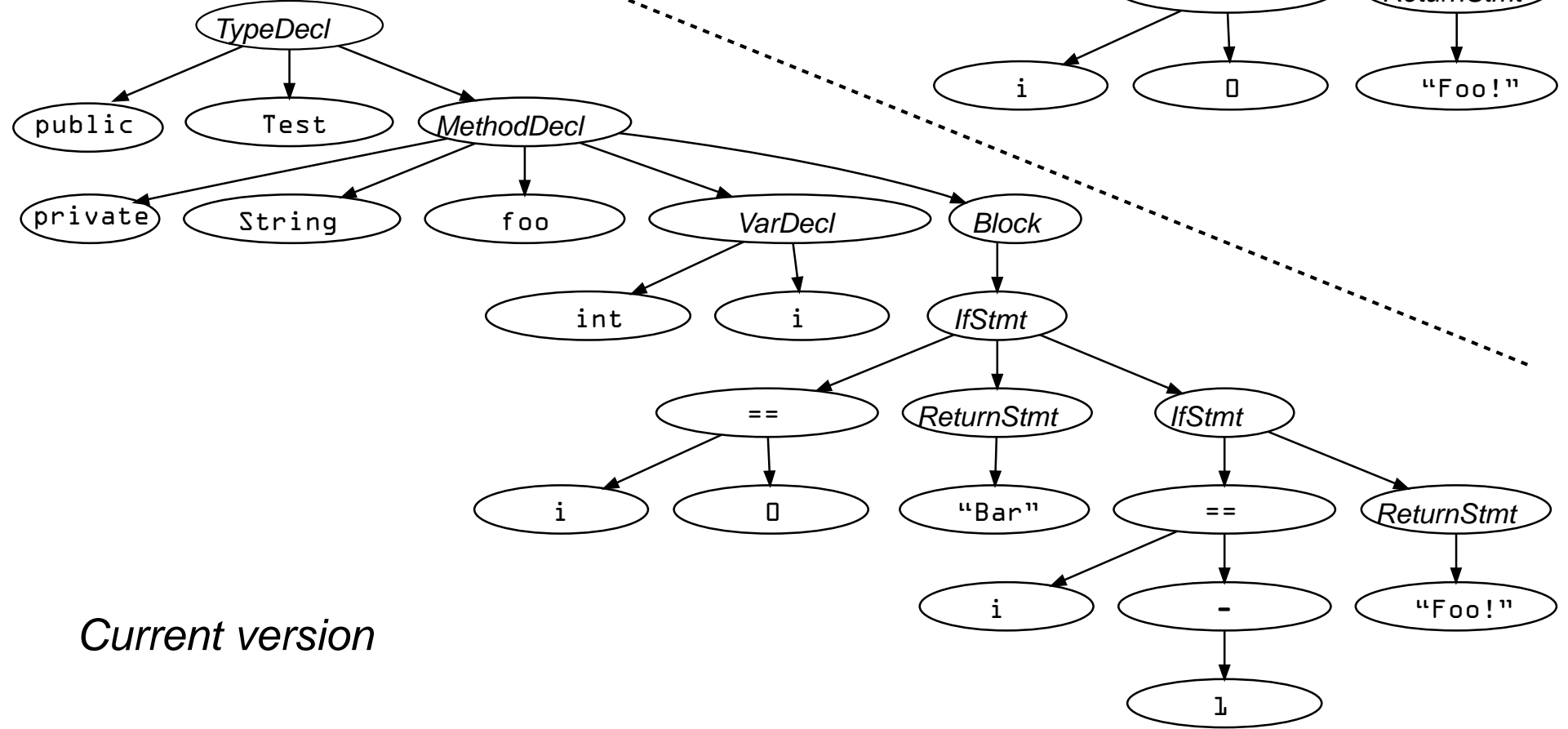
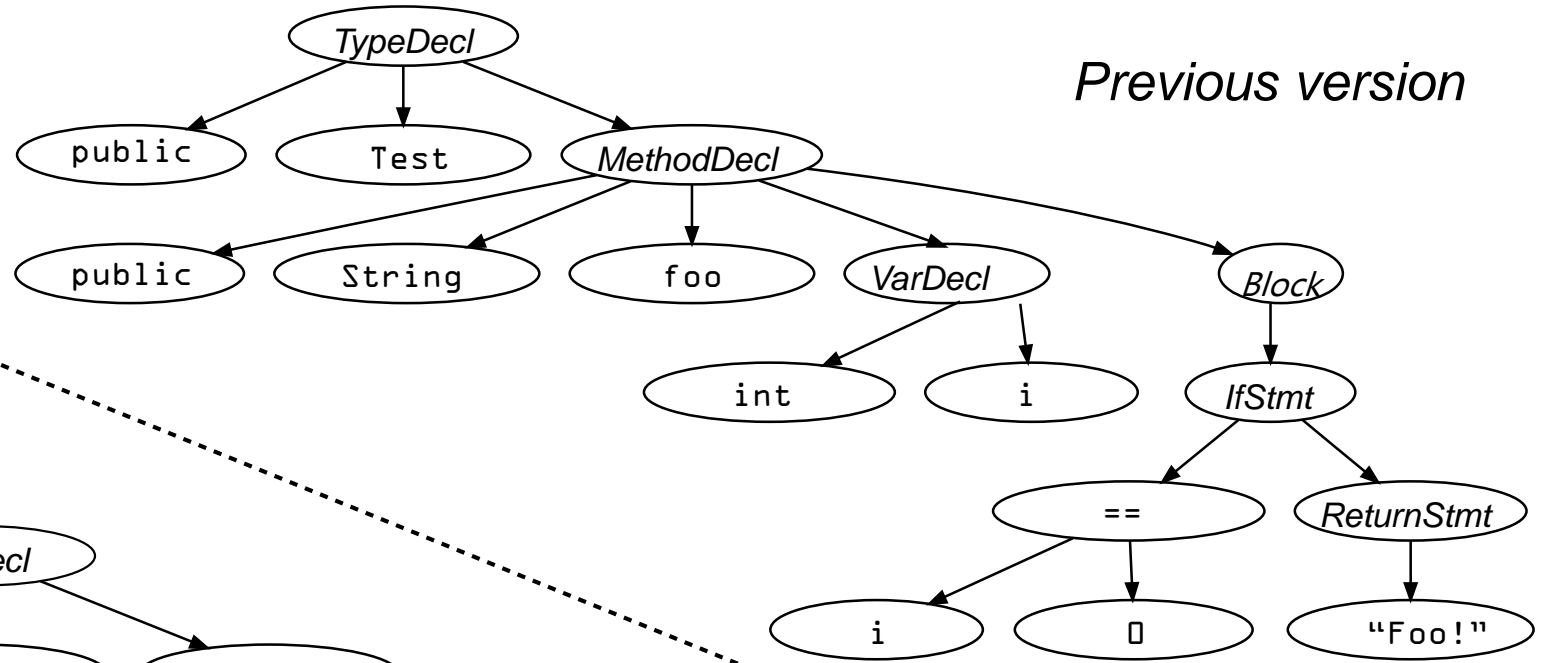
1. Top-down phase
  - Find biggest chunks of unmodified code
2. Bottom-up phase
  - Propagate mappings to the containers
  - Extend mappings in the left-over code

# Example

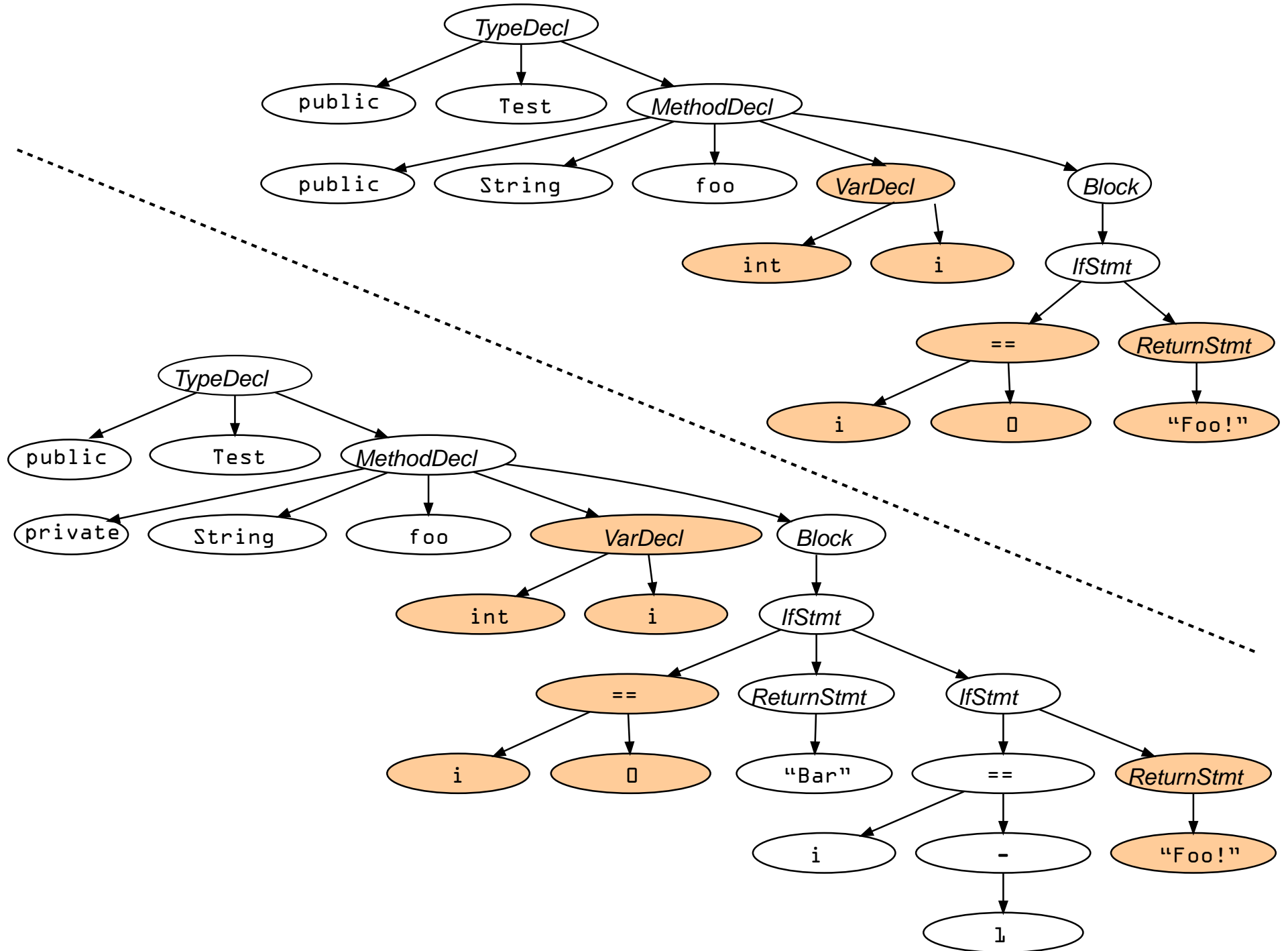
```
public class Test {  
    public String foo(int i) {  
        if (i == 0)  
            return "Foo!";  
    }  
}
```

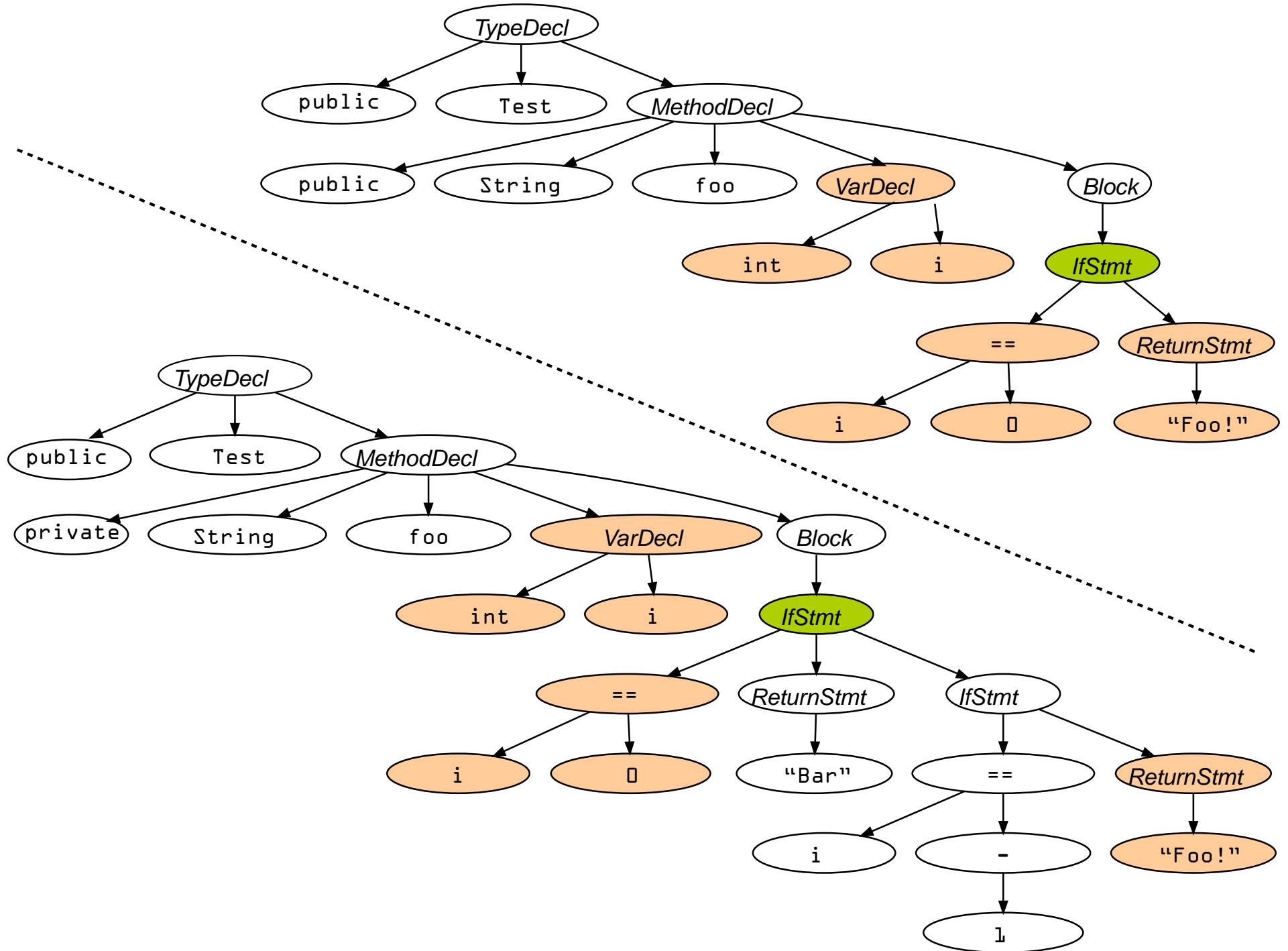
```
public class Test {  
    private String foo(int i) {  
        if (i == 0)  
            return "Bar";  
        else if (i == -1)  
            return "Foo!";  
    }  
}
```

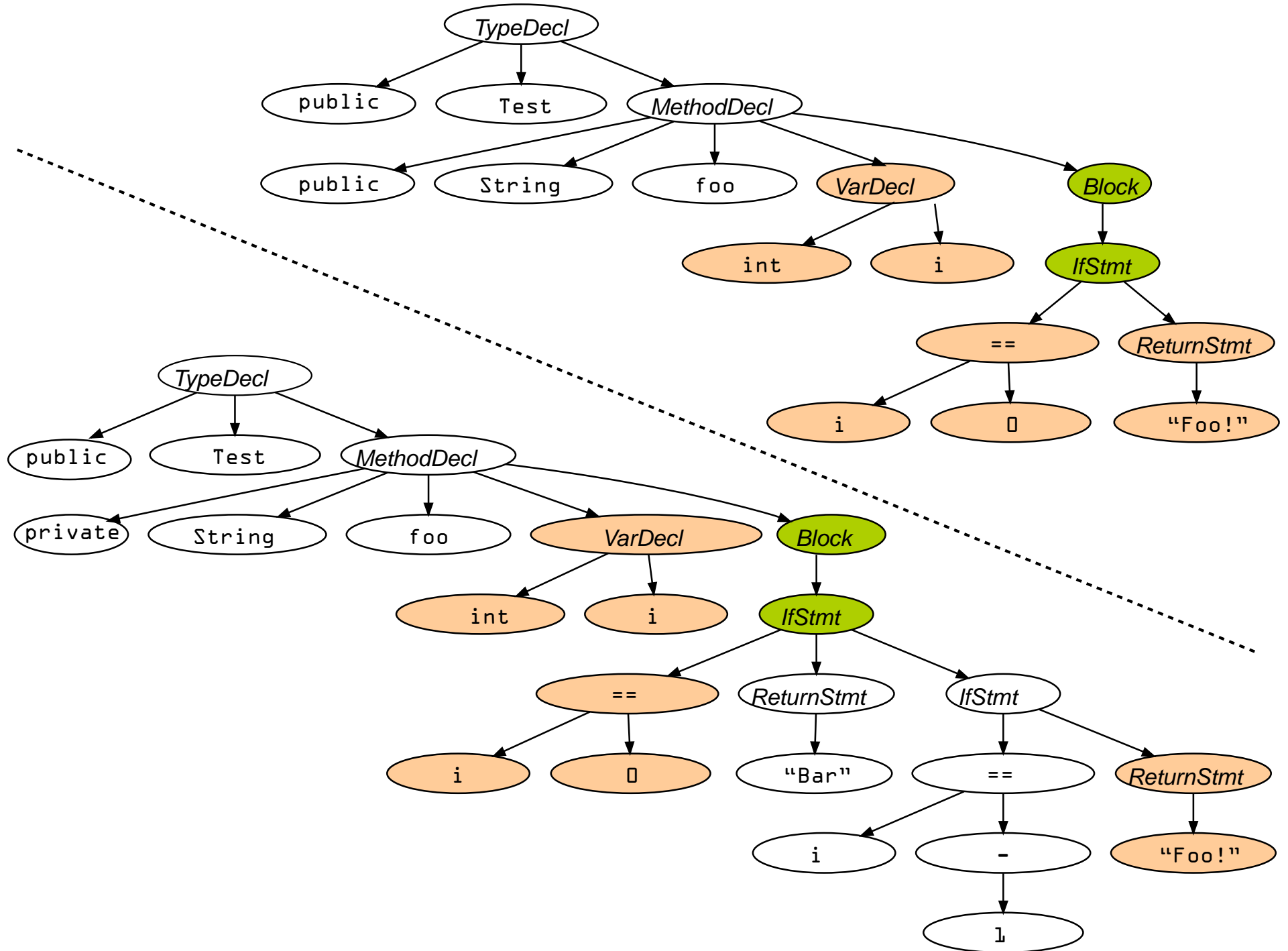
*Previous version*

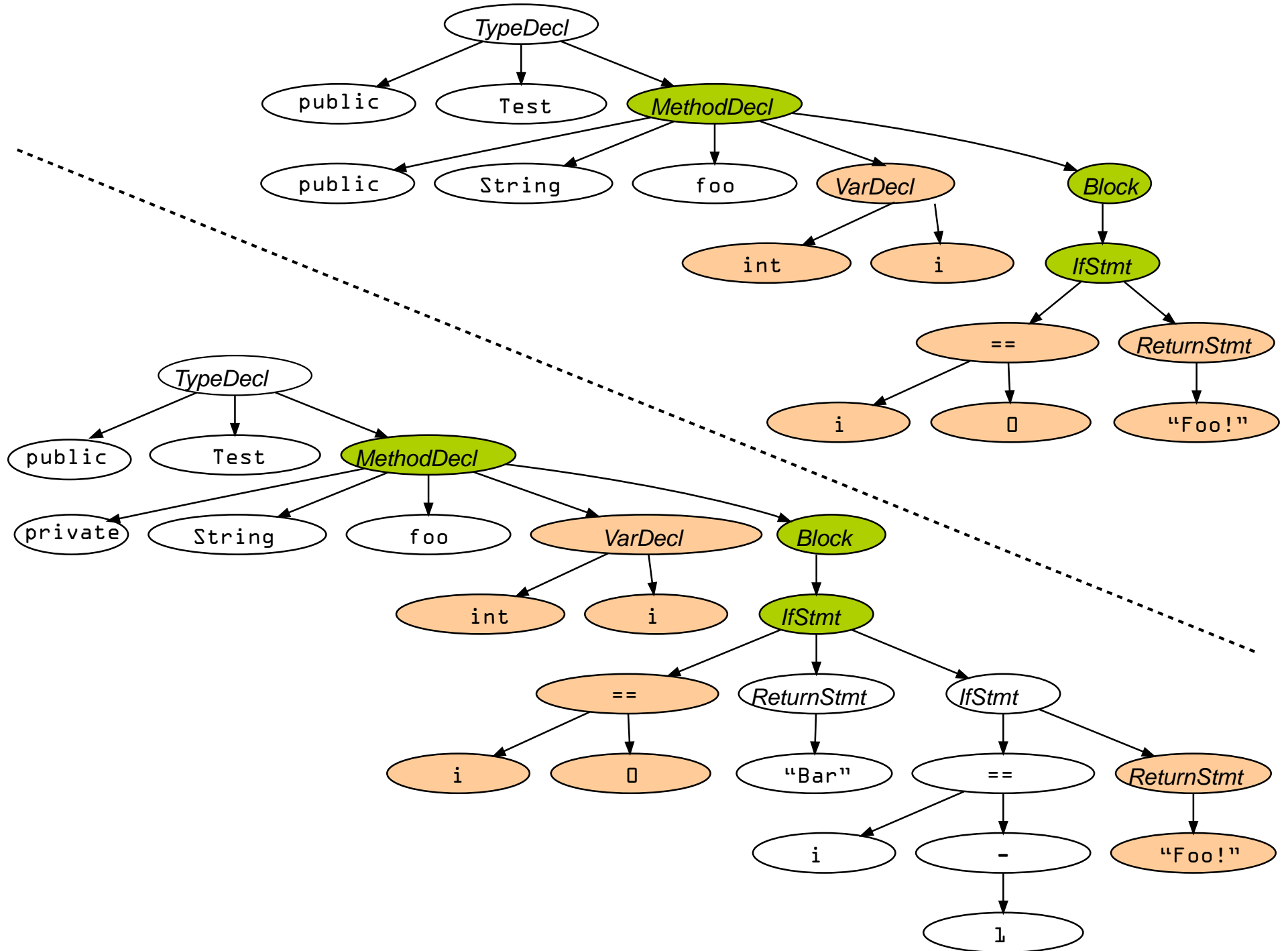


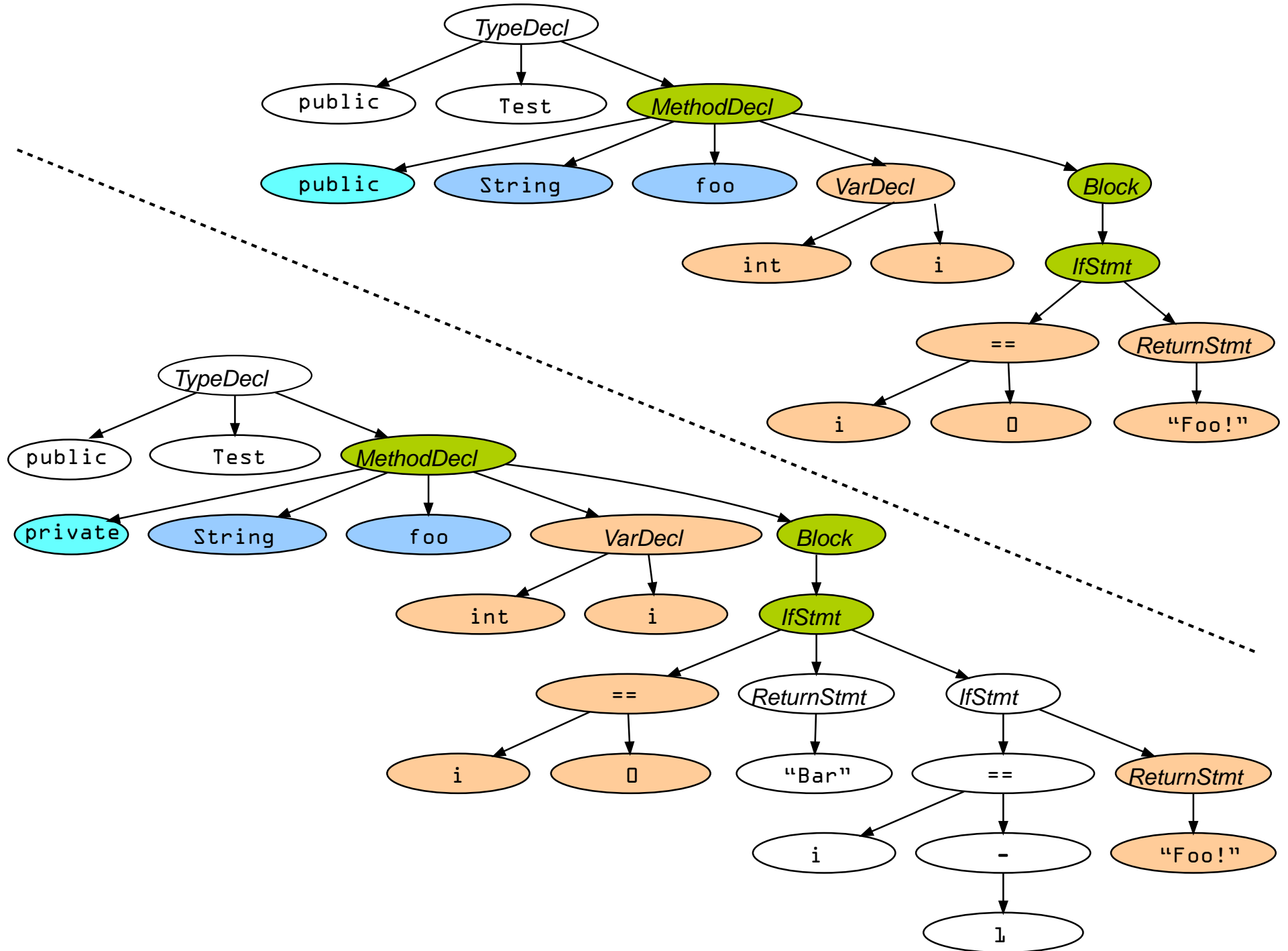
*Current version*

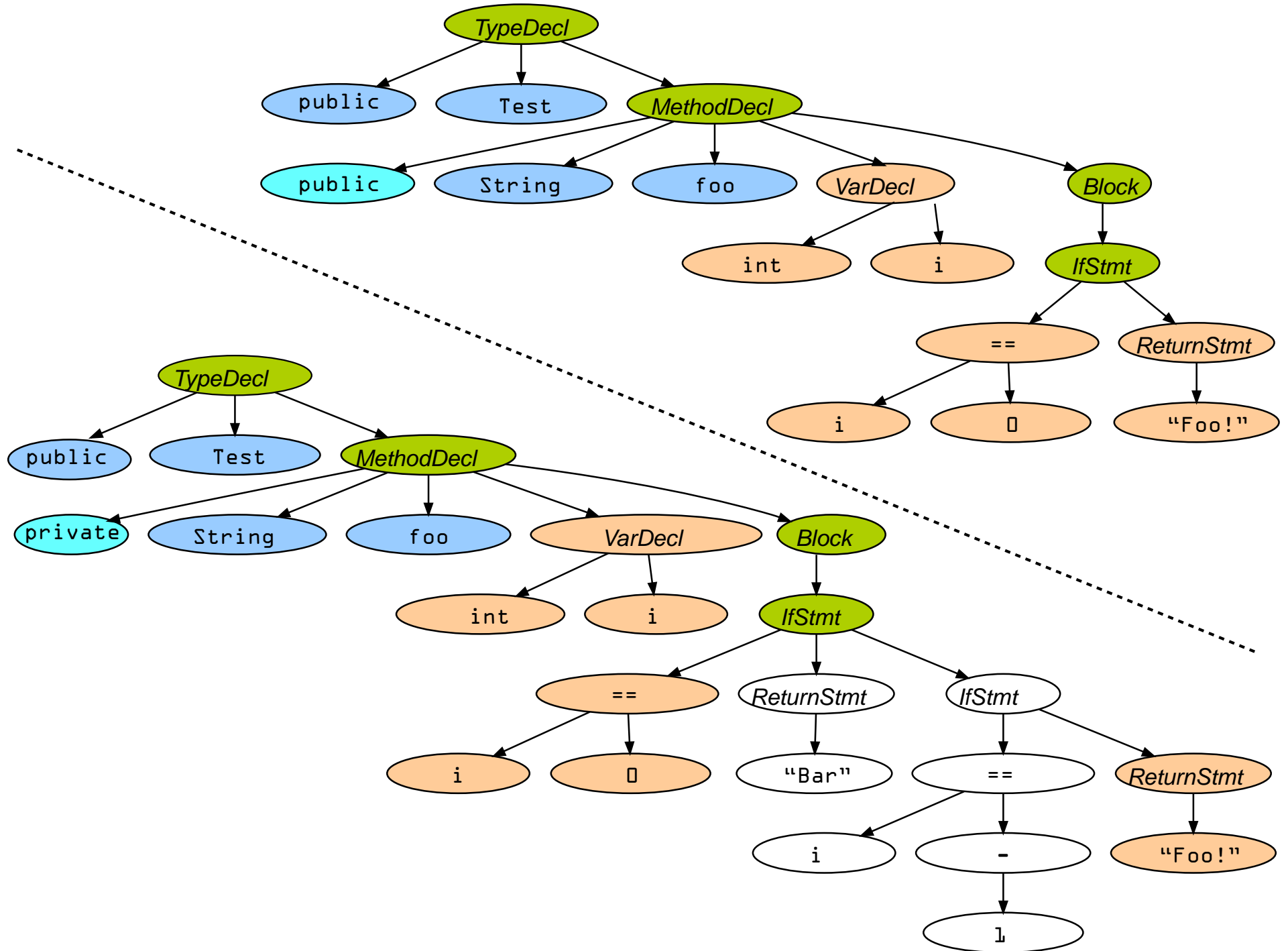












# Example

## Previous version

```
public class Test {  
    public String foo(int i) {  
        if (i == 0)  
            return "Foo!";  
    }  
}
```

## Current version

```
public class Test {  
    private String foo(int i) {  
        if (i == 0)  
            return "Bar";  
        else if (i == -1)  
            return "Foo!";  
    }  
}
```

# Change analysis Summary

- From textual diff to tree diff
- Understanding source code changes
- Falleri et al. ASE 2014
- <https://github.com/GumTreeDiff/gumtree>

# OSS Experiments - Turnover

# Software Quality

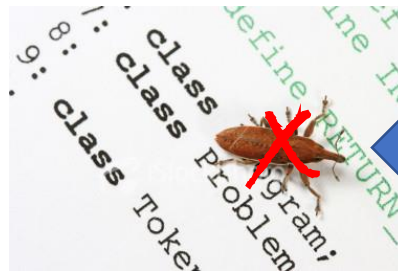
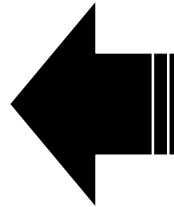


Usability  
Performance  
Security  
...

*Does the software system behave like it should ?*

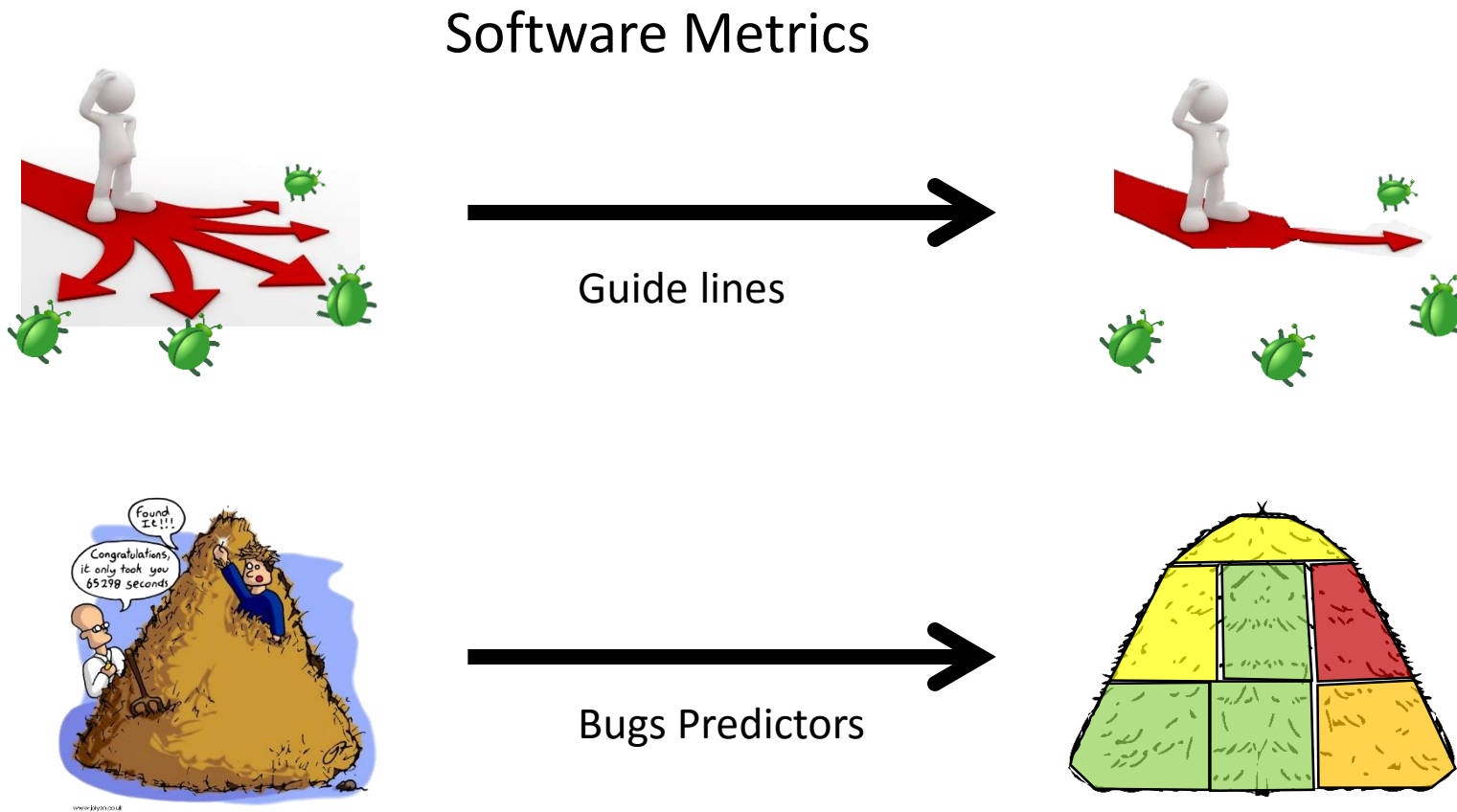


~~X~~ Quality



Objective

# Metrics to support Software Maintenance



# Two kinds of metrics

## **Source code metrics**

Complexity

Size

Coupling

## **Process metrics**

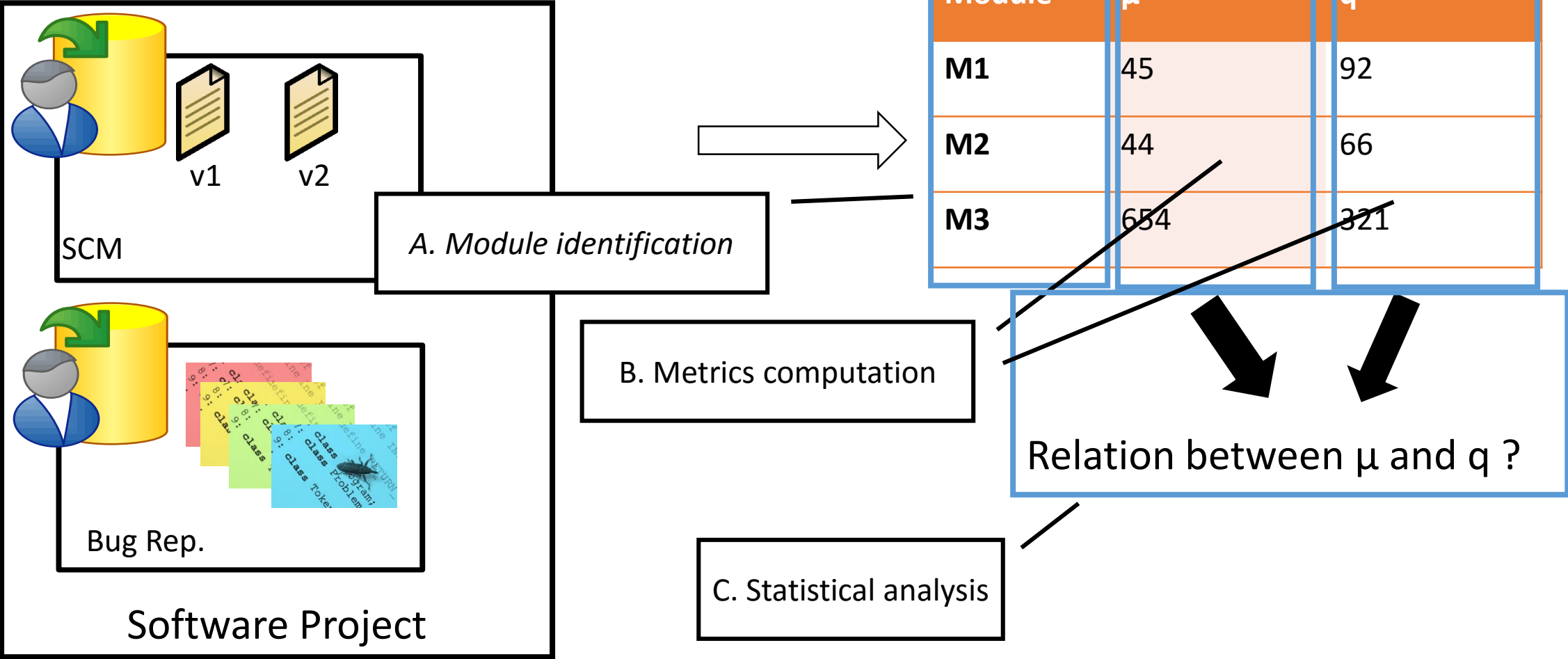
Churn

Frequency

Developers organisation

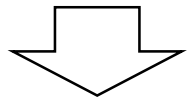
F. Rahman and P. Devanbu, *“How, and why, process metrics are better,”* ICSE 2013  
Source code metrics are probably too static !

# Metrics validation

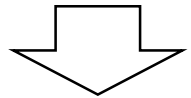


# External Turnover

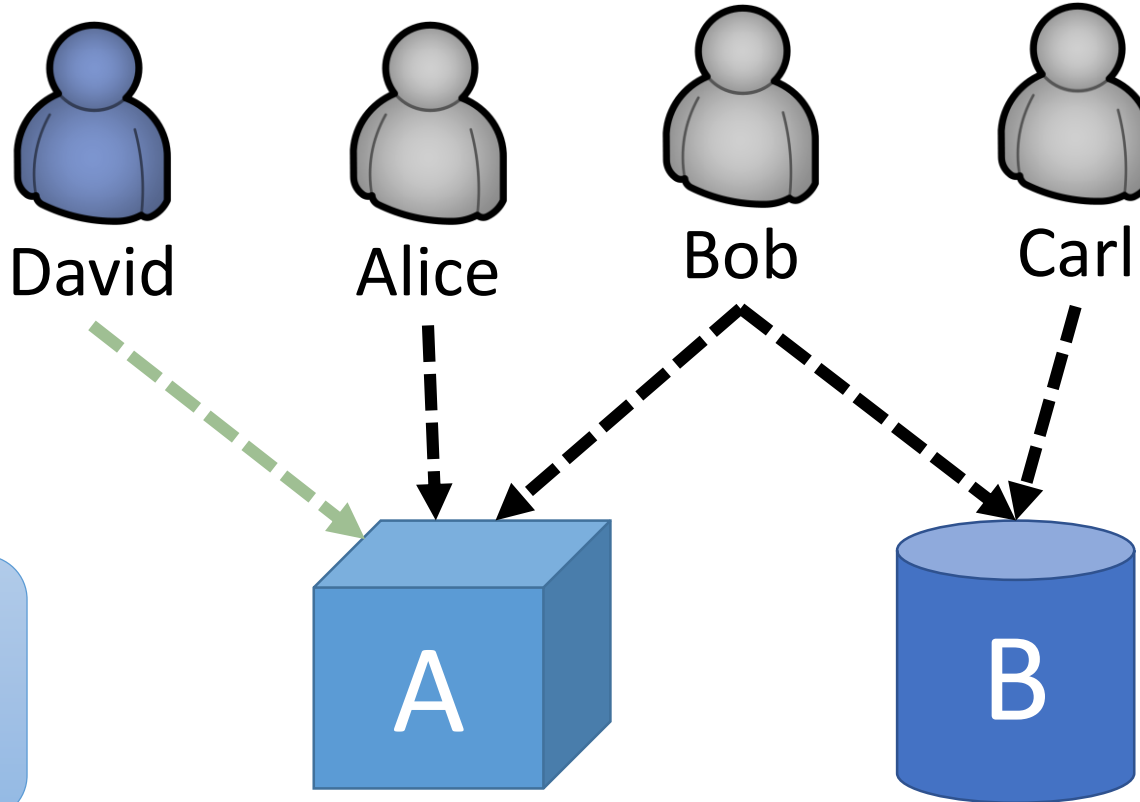
External Newcomer



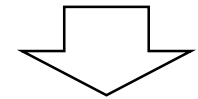
No or poor knowledge



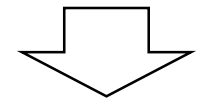
More bugs  
[Rahman & Devanbu,  
ICSE'11]



External Leaver



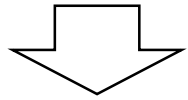
Loss of knowledge



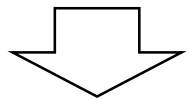
More bugs  
[Mockus, FSE'10]

# Internal turnover

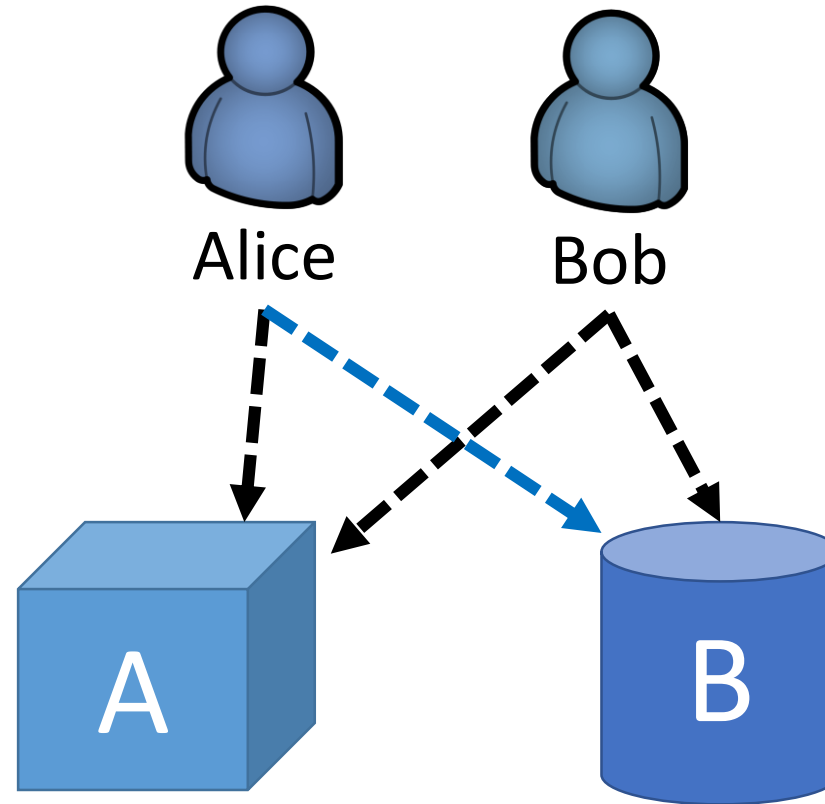
Internal Newcomer



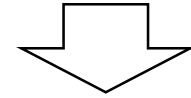
Alice is increasing her knowledge of the project



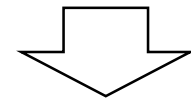
Less bug  
[Mockus & Weiss, Bell  
Labs Tech J., 2000]



Internal Leaver



Changing responsibility is good for the motivation

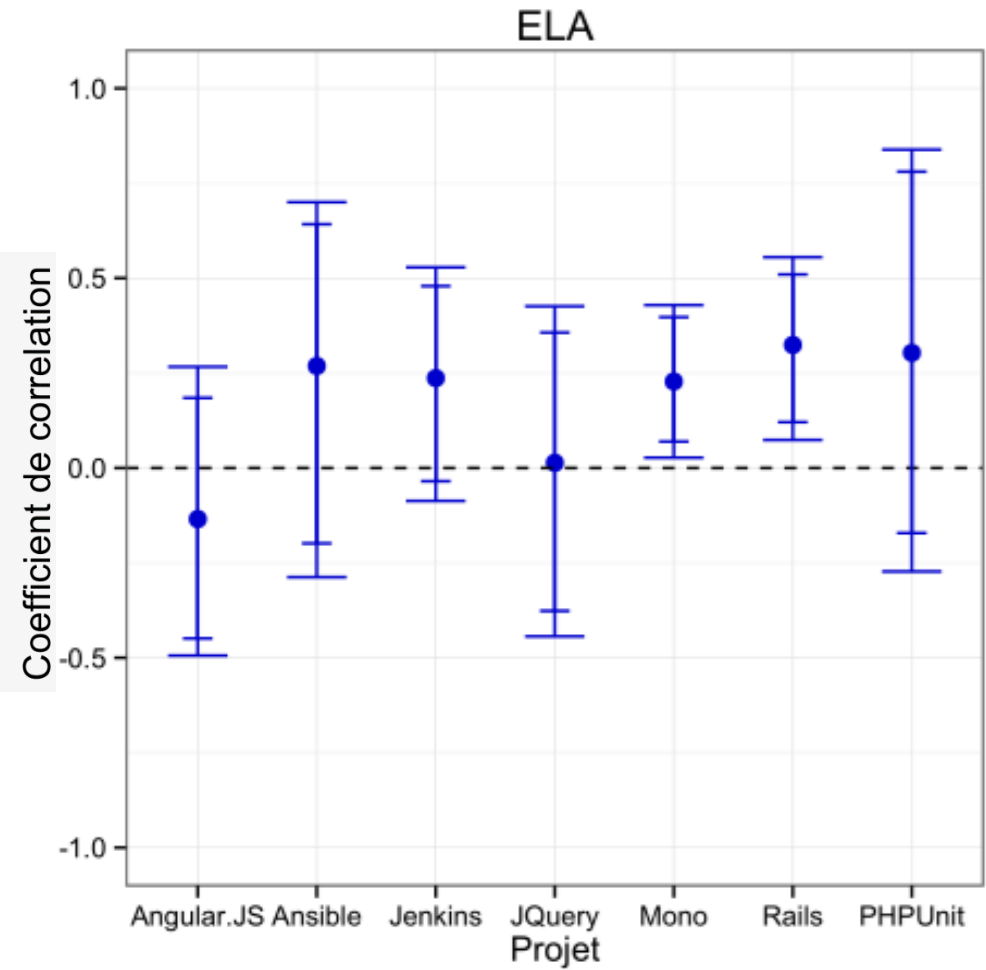
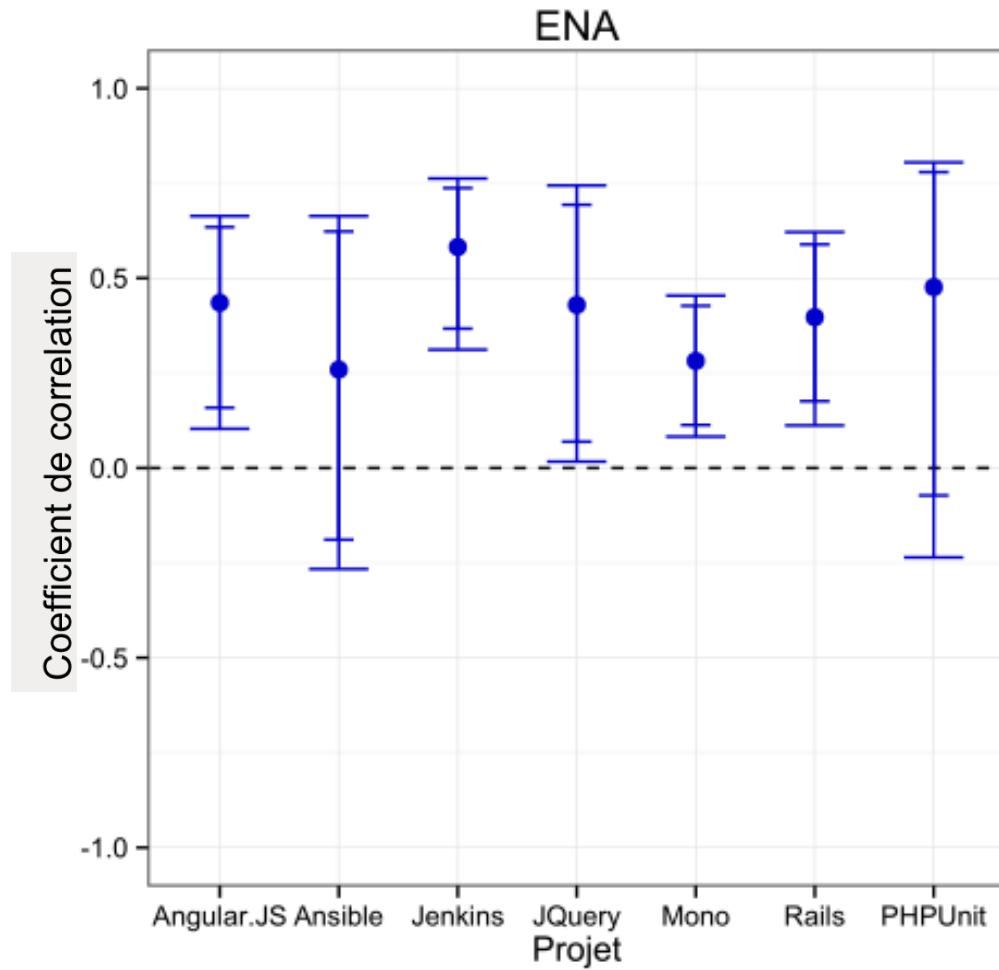


Less bug  
[Kanter, 1976]

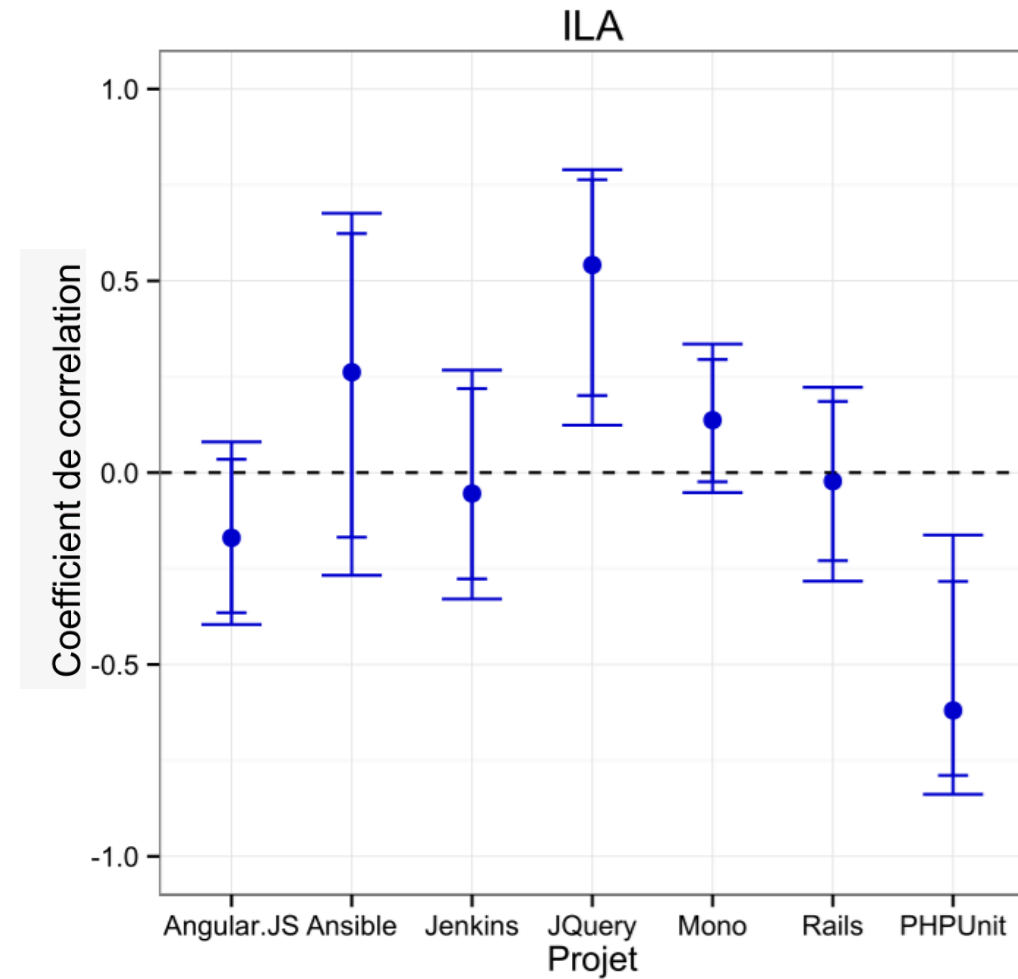
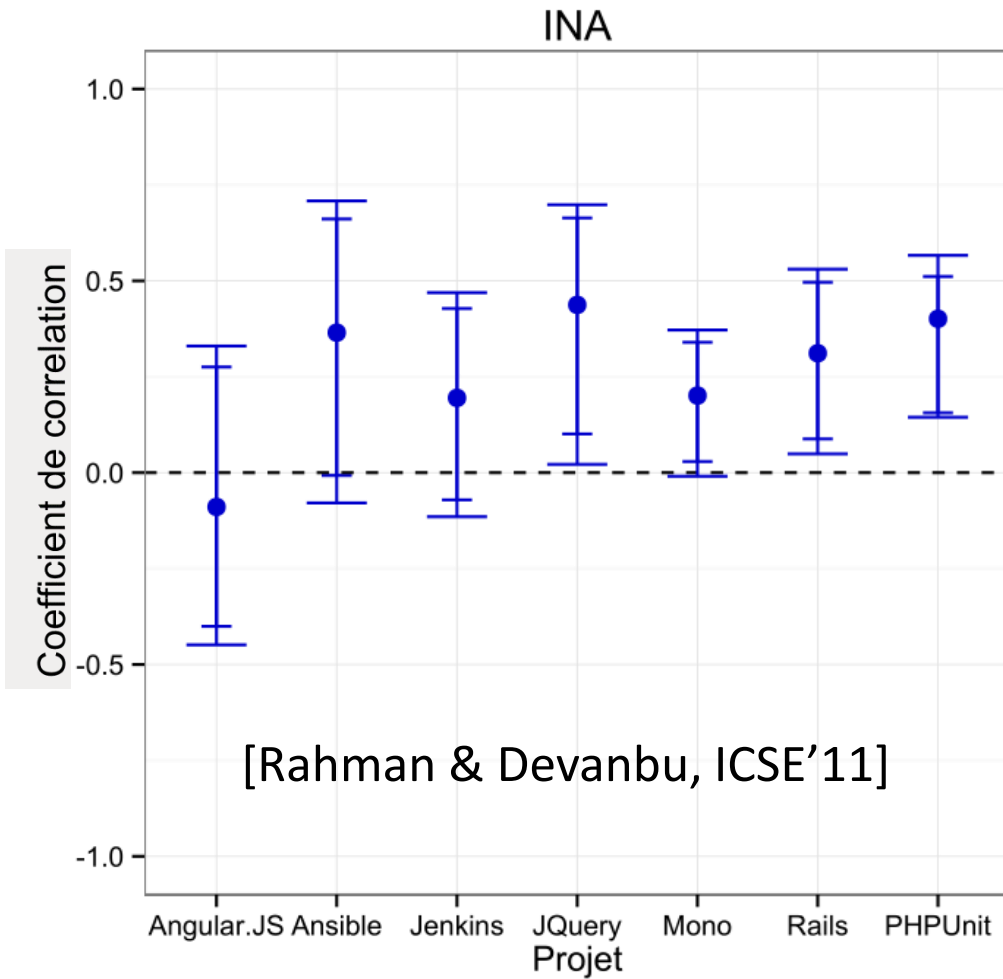
# Research Questions

1. *Is turnover an important phenomena?*
2. *Is there any schema for turnover?*
3. Is there any relationship between turnover and quality?

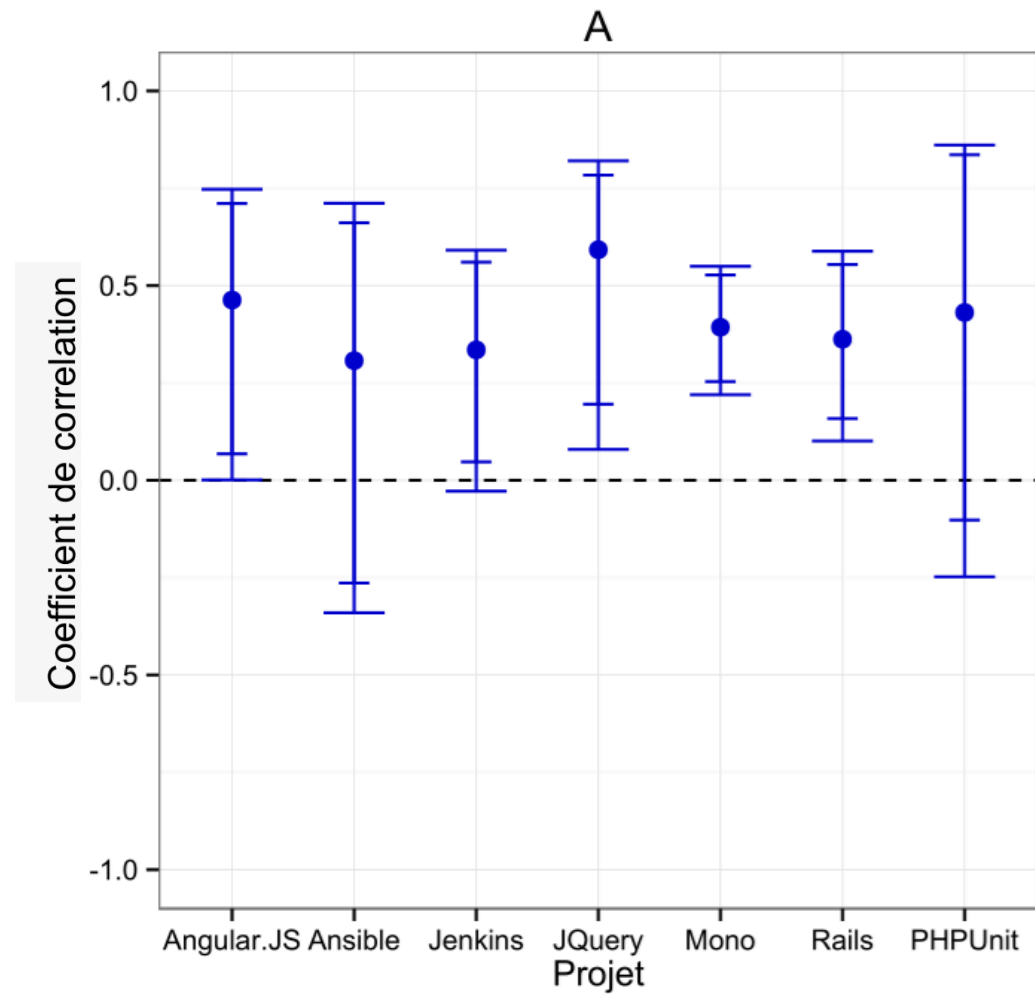
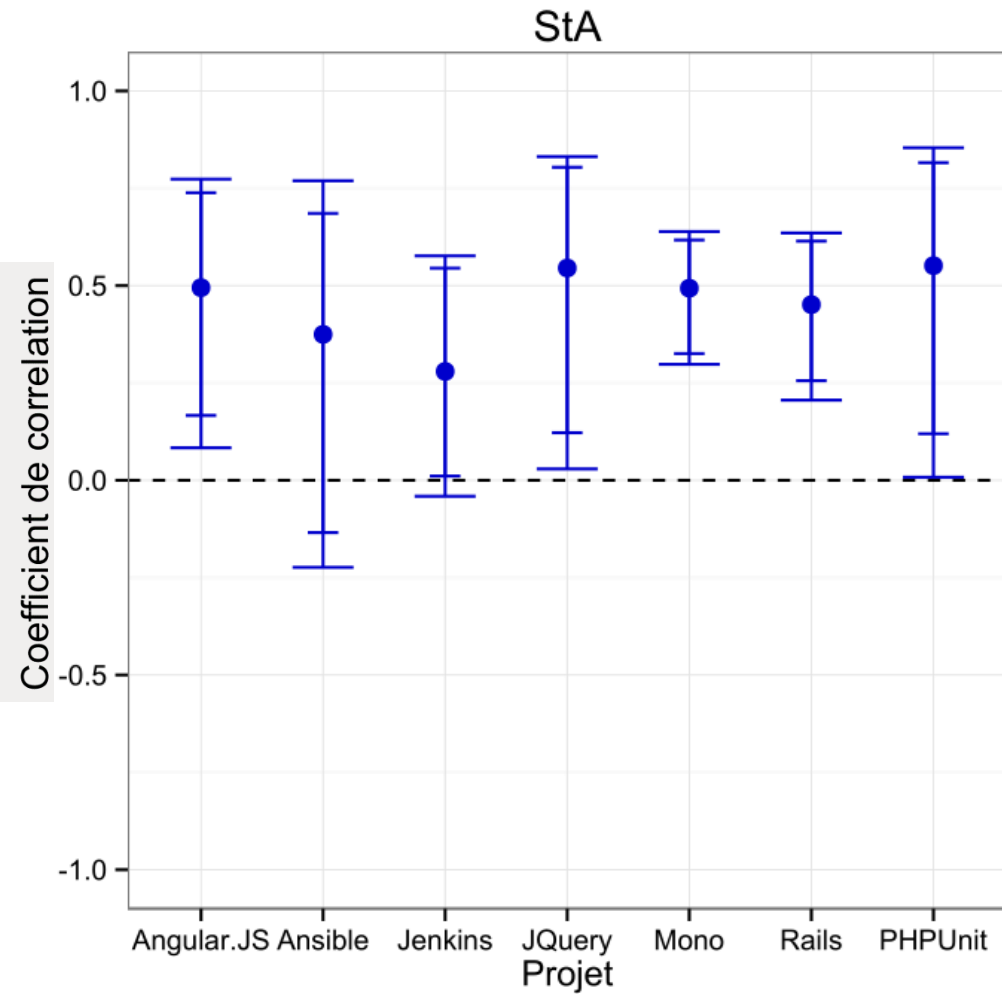
# External Turnover



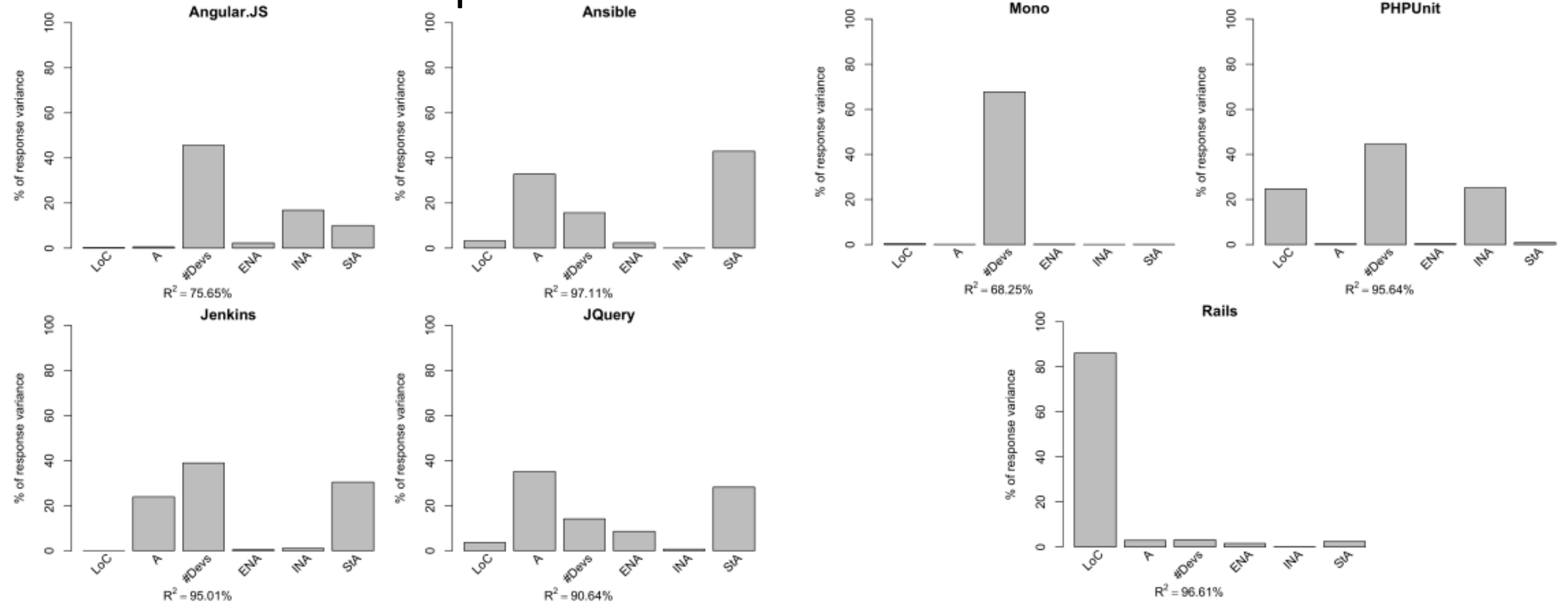
# Internal Turnover



# Stayers



# Relative Importance

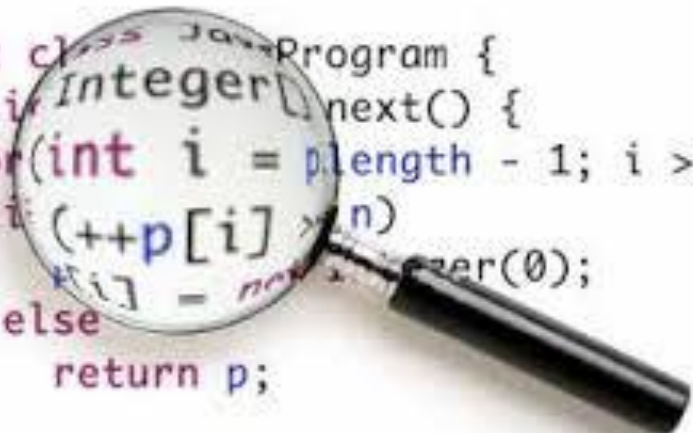


# OSS Experiments

- Process metrics to support software maintenance
- Turnover metrics, MSR study on OSS projects, Foucault et al. [FSE15]
- Tailored Predictors
- Startup

# Conclusion

# Software Evolution @LaBRI



```
public class JavaProgram {  
    public Integer[] next() {  
        for(int i = p.length - 1; i >= 0;  
            i--){  
            if(++p[i] > n)  
                p[i] = nextElement(0);  
            else  
                return p;  
        }  
        throw new NoSuchElementException();  
    }  
}
```

- Code driven
  - GumTree
- Repository Analysis
  - DigIt, Statistics, etc.
- Human in the loop
  - Log, Organisation