



Software Engineering Research in the Neuroage

fMRI & EEG Studies on Program Comprehension

Norman Peitek



European Research Council
Established by the European Commission

Agenda

- Part I: Motivation & Past SE Research on Programmers
- Part II: Neuroscience Basics (for SE)
- Part III: Neuroscientific Studies on Programmers
- Part IV: Perspectives & Future Research

Learning Objectives

- Understand potentials, limitations, and challenges of empirical SE research with neuroscientific methods
- Gain basic familiarity with neuroscience
- Obtain overview of insights from current SE research

Part I: Motivation & Past SE Research on Programmers



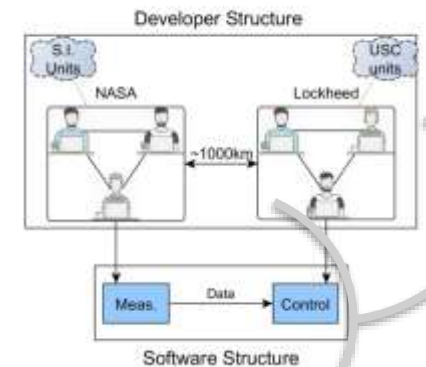
Software Engineering

Programming
languages,
Development
processes,
Testing,
Design,
...

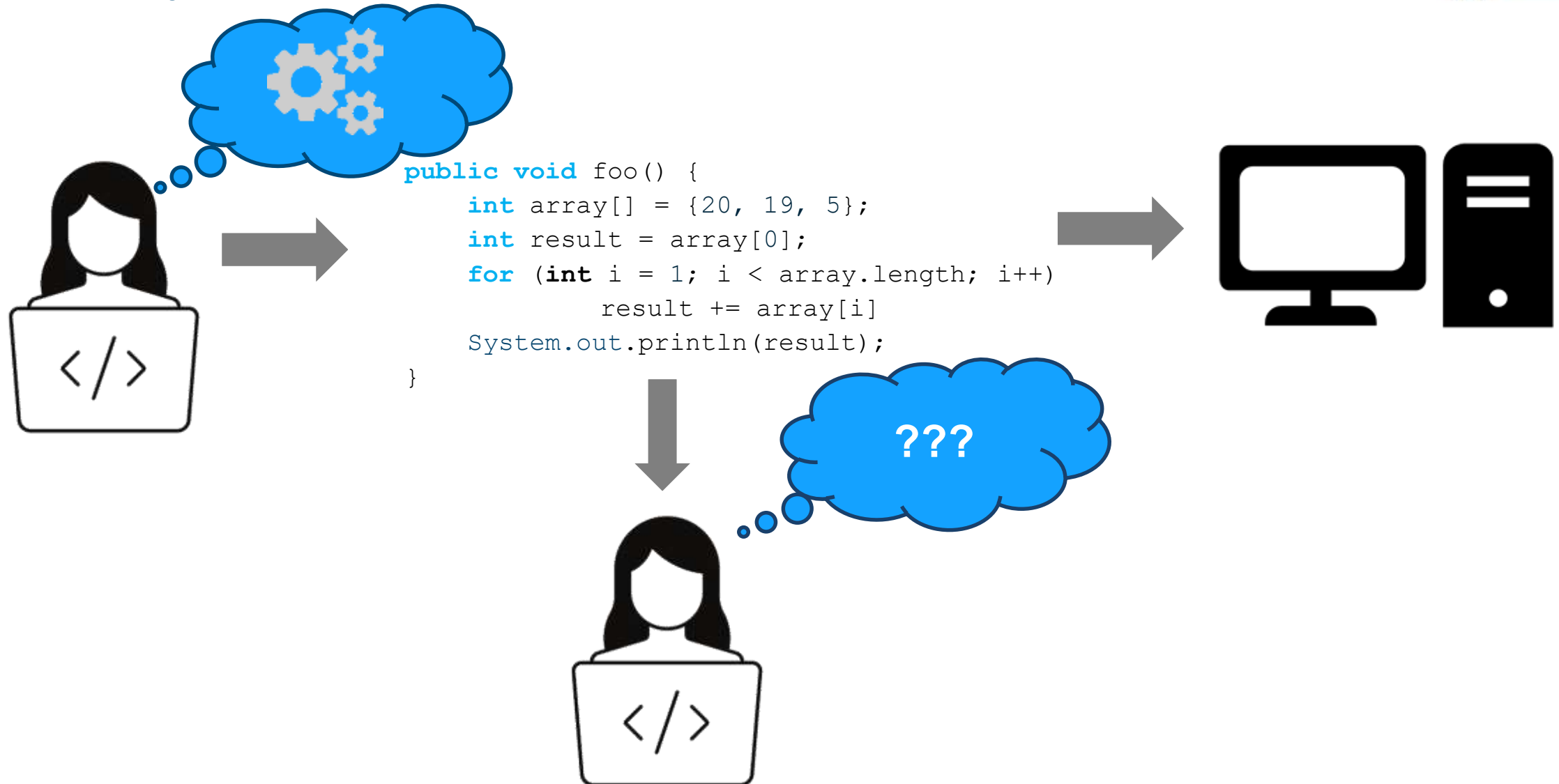
Education,
Training,
Team
collaboration,
Communication,
...

Technical aspects

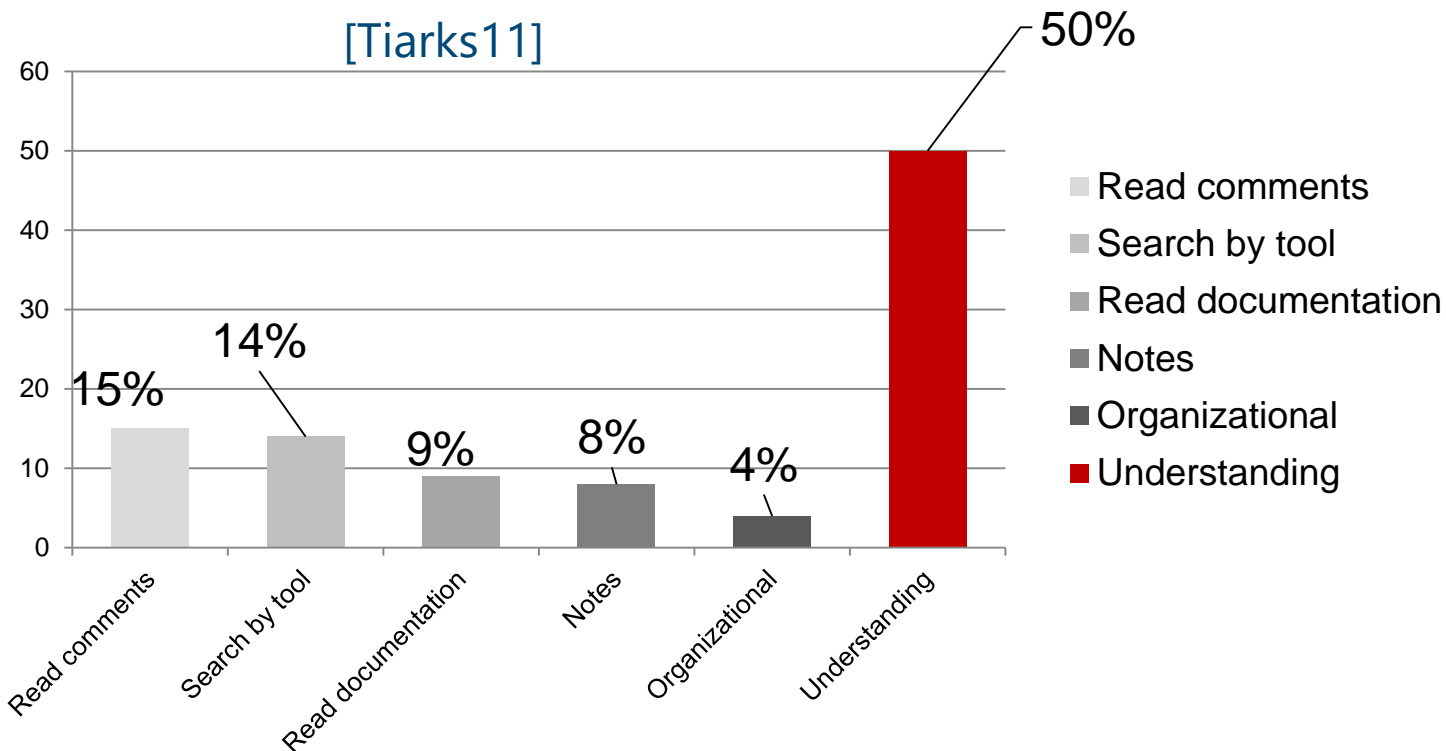
Human aspects




Bimodality of Code



Focus on Program Comprehension




Software Engineering



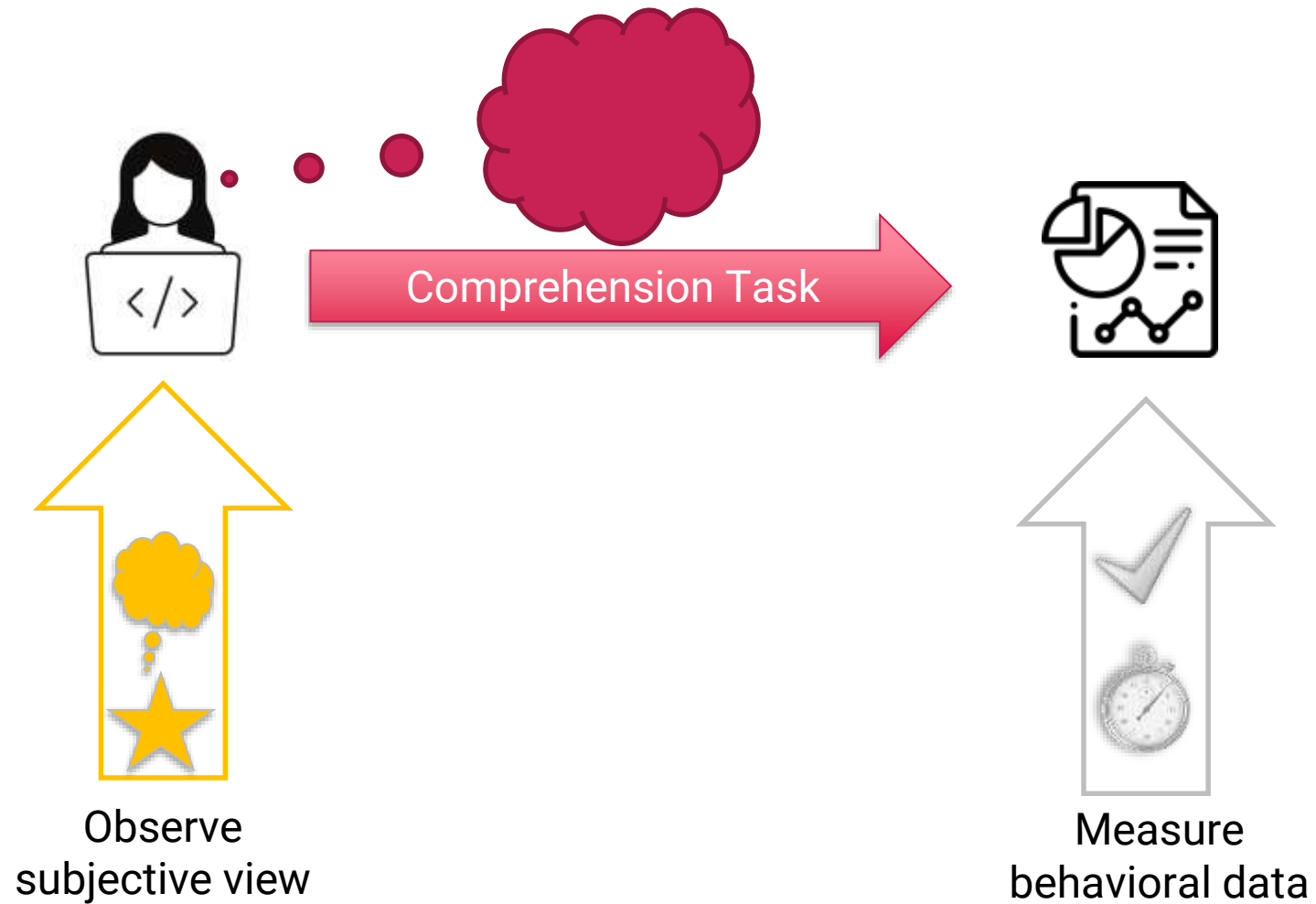
Programming Languages



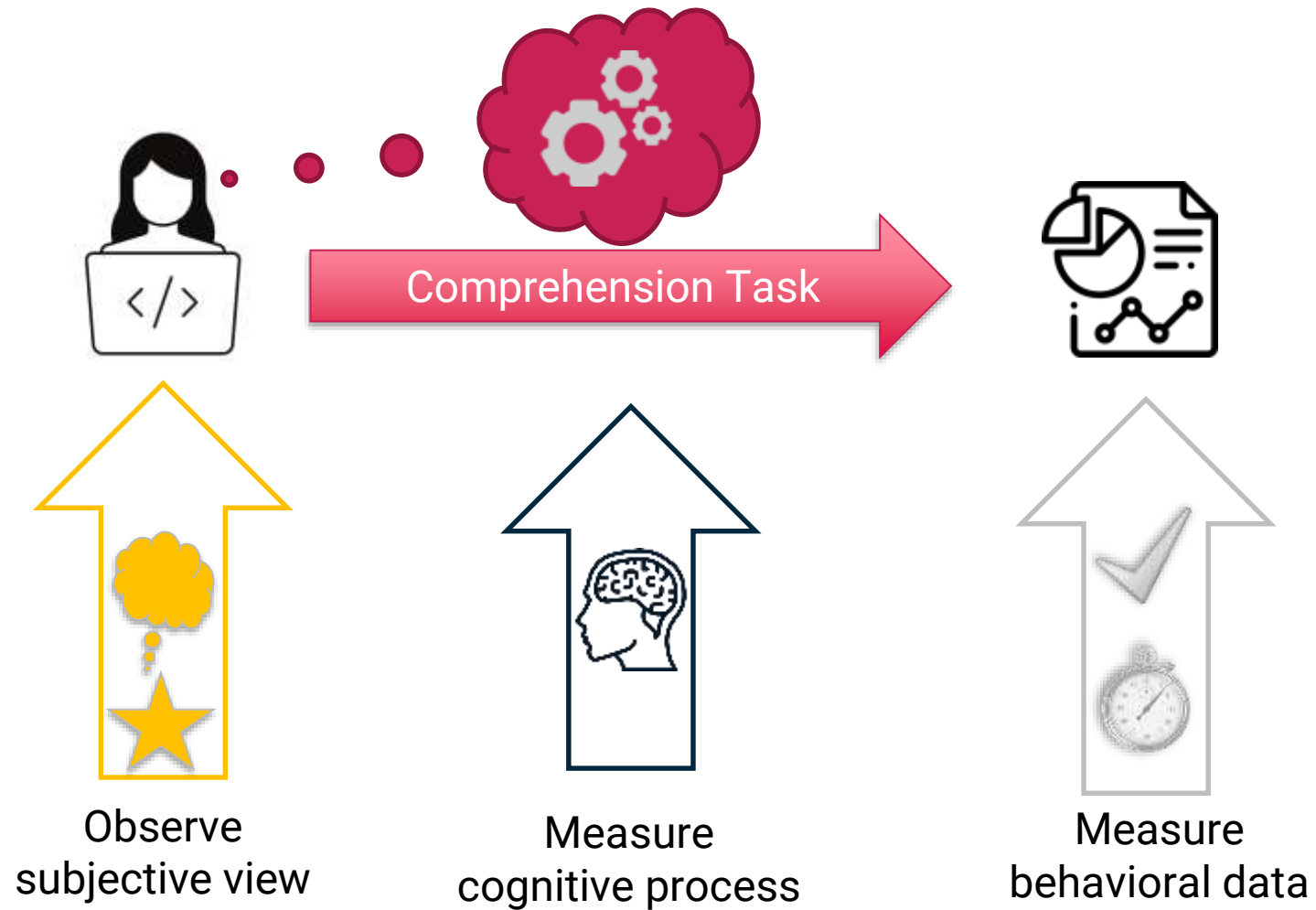
Education & Training



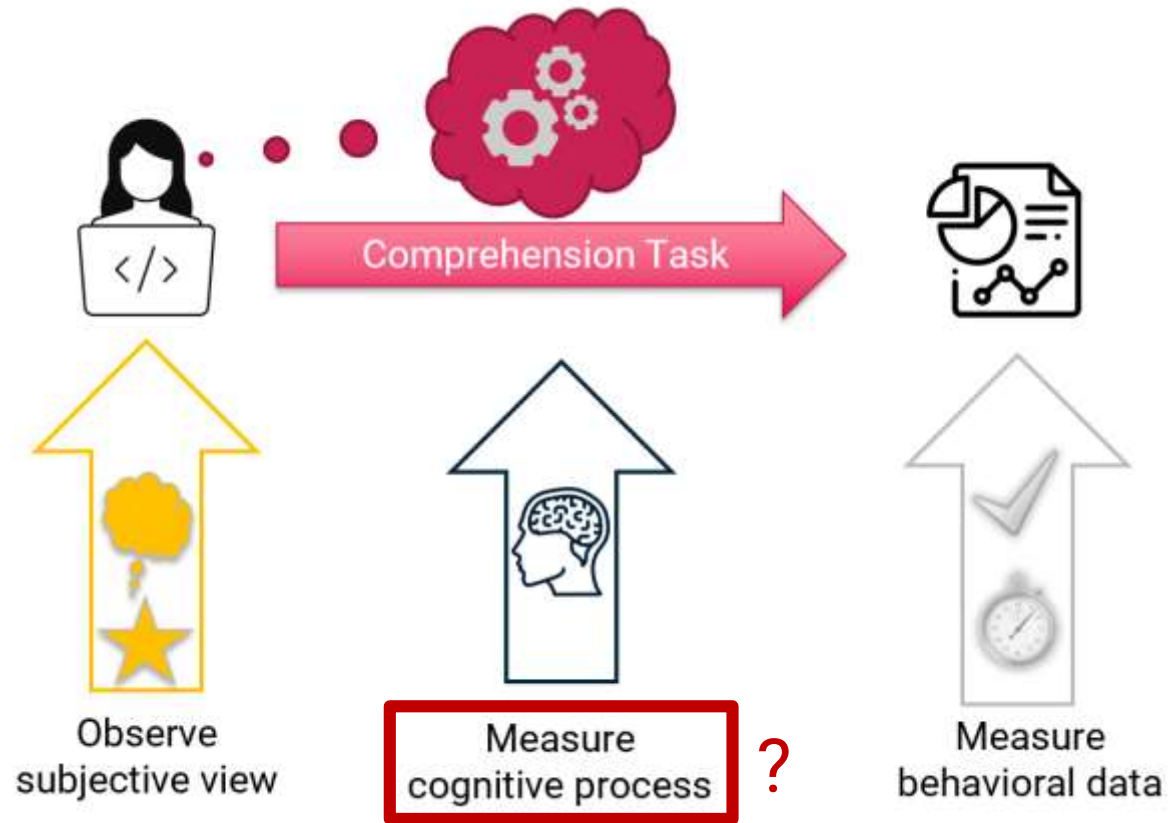
Past Research on Programmers



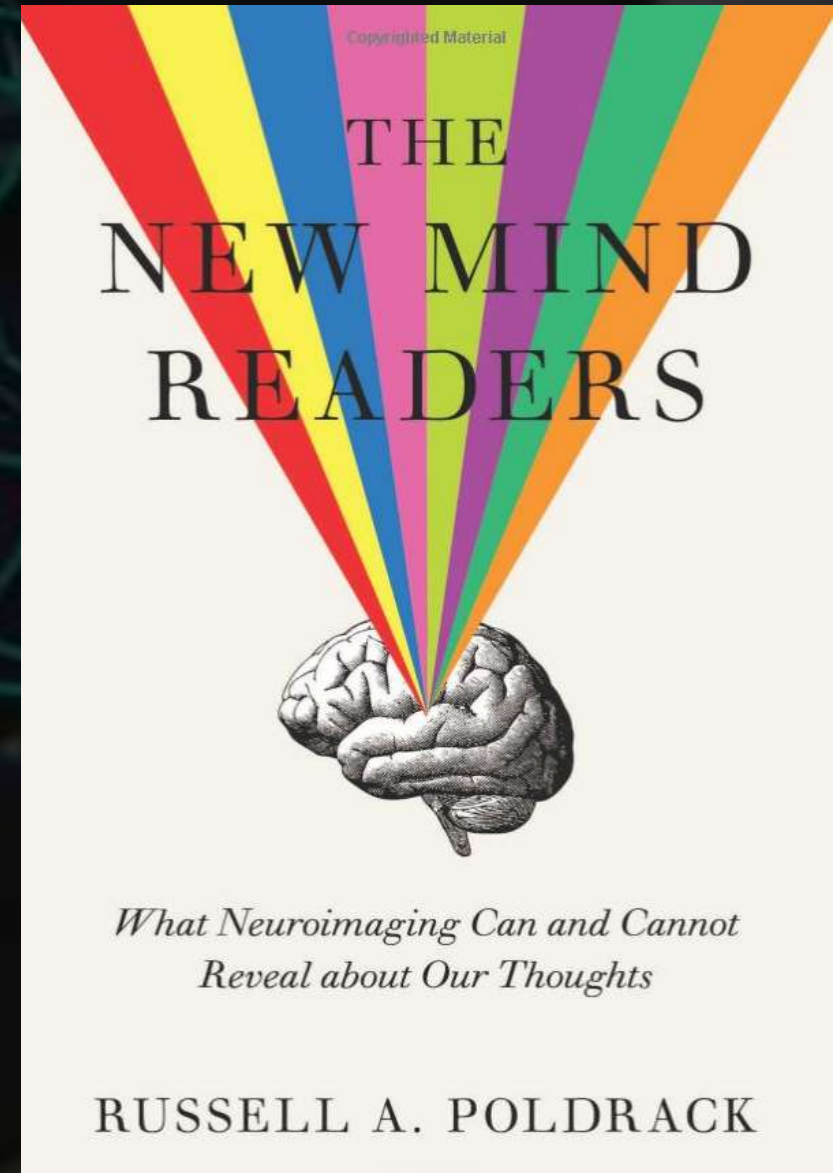
Past Research on Programmers



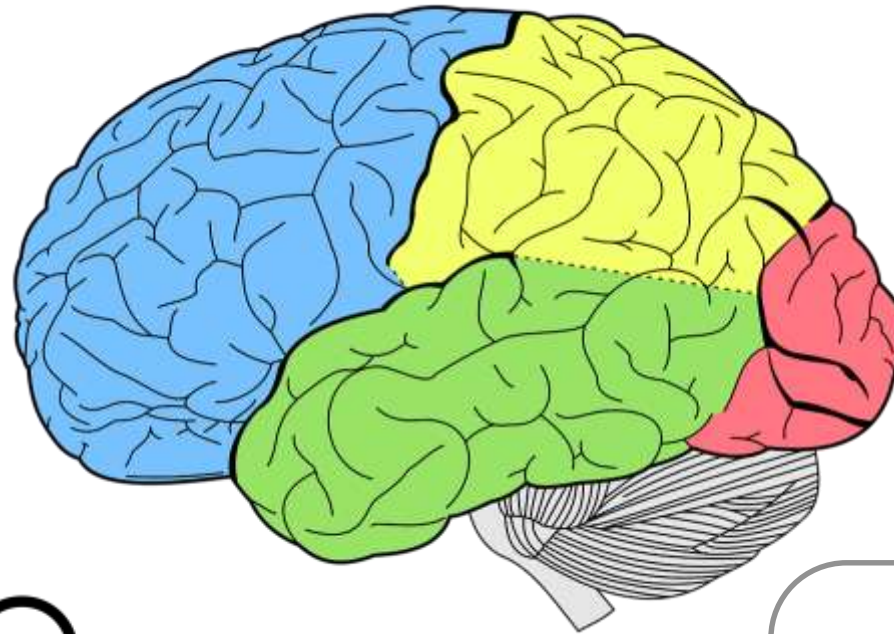
functional Magnetic Resonance Imaging (fMRI)



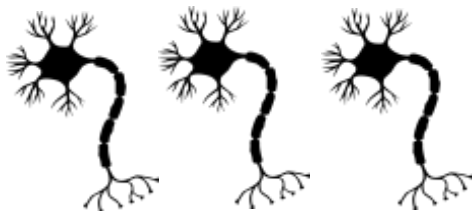
Part II: Neuroscience Basics (for SE)



Basics of Neuroscience



Micro Imaging



Macro Imaging

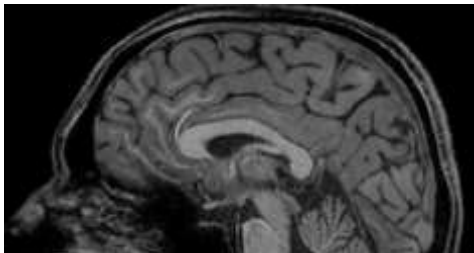
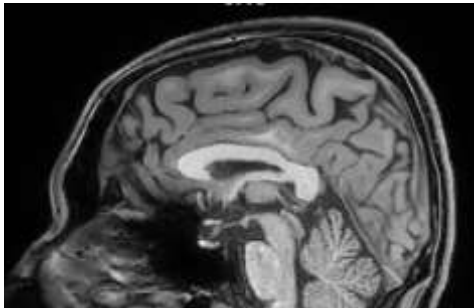


Overall Cognitive Load

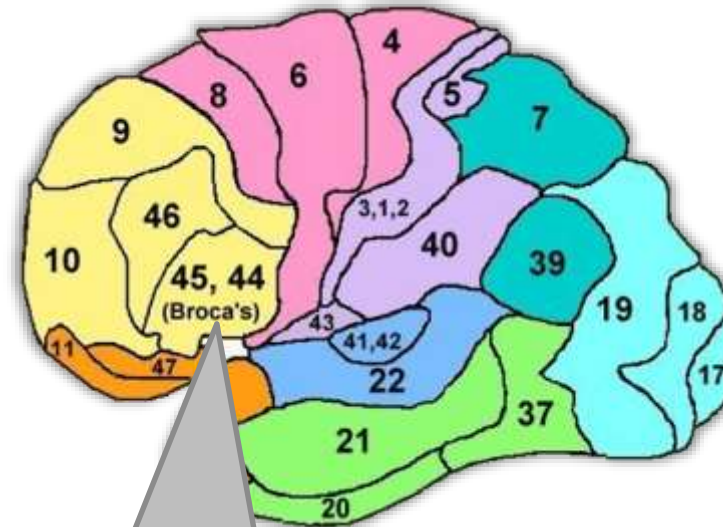
Brain Area Activation

Brain Standardization

We all have different brains!



Standardization:
Talairach/MNI Space



Brodman Areas (BA)

Sample Experiment: n-back

$n=2$

Position	0	1	2	3	4	5	6
Number							

Sample Experiment: n-back

$n=2$

Position	0	1	2	3	4	5	6
Number	4						

Sample Experiment: n-back

$n=2$

Position	0	1	2	3	4	5	6
Number		1					

Sample Experiment: n-back

$n=2$

Position	0	1	2	3	4	5	6
Number			9				



Sample Experiment: n-back

$n=2$

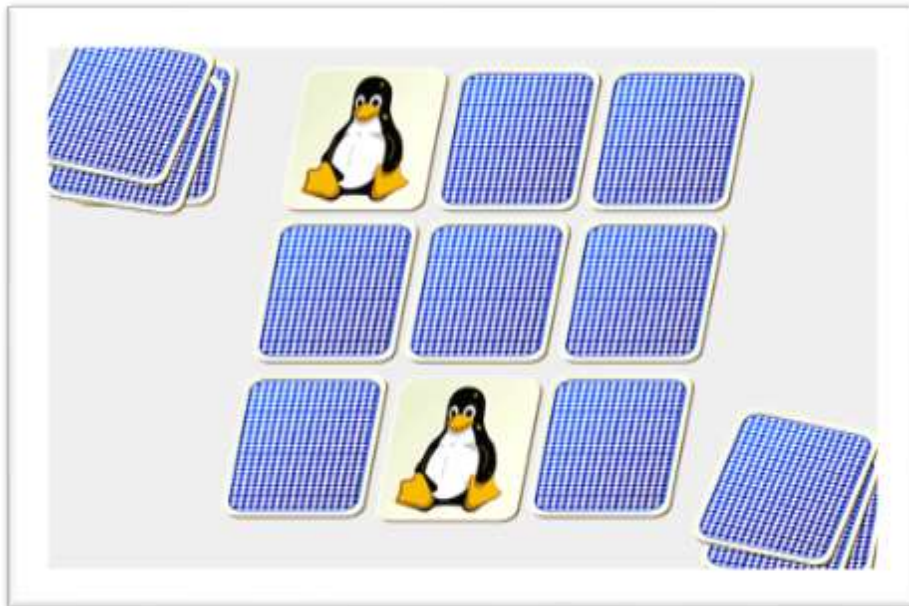
Position	0	1	2	3	4	5	6
Number				8			



Sample Experiment: n-back

$n=2$

Position	0	1	2	3	4	5	6
Number					0		



Sample Experiment: n-back

$n=3$

Position	0	1	2	3	4	5	6
Number					0		



$n=4$

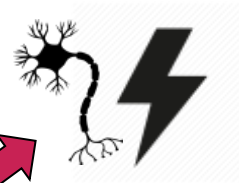
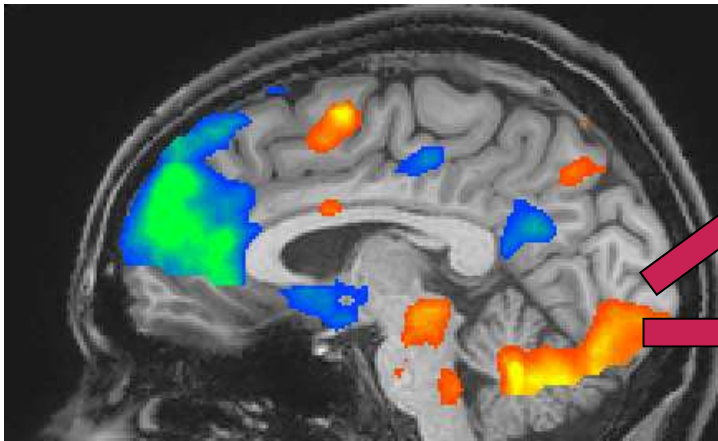
Position	0	1	2	3	4	5	6
Number					0		



How Do We Measure Brain Activity?



Position	0	1	2	3	4	5	6
Number				8			



O_2 O_2
 O_2 O_2

BOLD effect

O_2 O_2

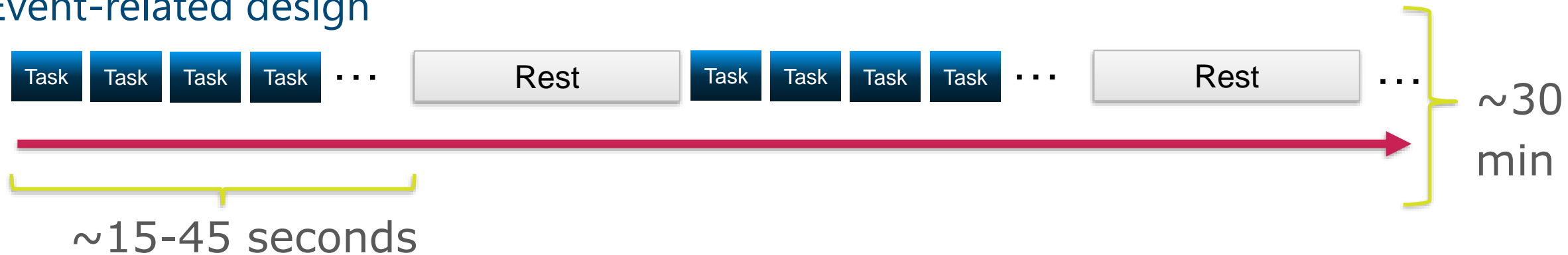
O_2 O_2

Experiment Designs

- Block design



- Event-related design



Modalities

functional magnetic resonance imaging

fMRI



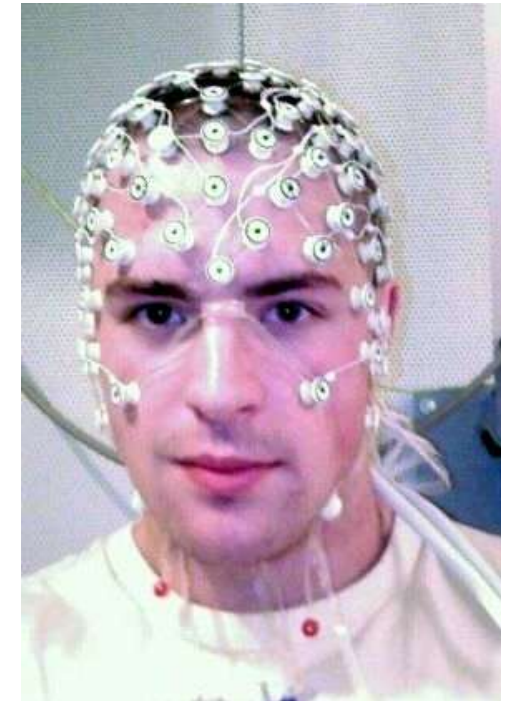
functional near-infrared
spectroscopy

fNIRS



Electroencephalogram

EEG



Comparison of Modalities

	fMRI	fNIRS	EEG
Based on/Measures	BOLD	BOLD	Electrical activity
Temporal resolution	Slow (1-2 sec), delayed	Slow (1-2 sec), delayed	Very fast (<100ms)
Spatial resolution	Good (full brain, 3D)	Weak	Very weak
Costs	~200 Euro/hour	~75 Euro/hour	~35 Euro/hour
Limitations	Many	Some	Some

Part III: Neuroscientific Studies on Programmers



fMRI Study 1 on Program Comprehension

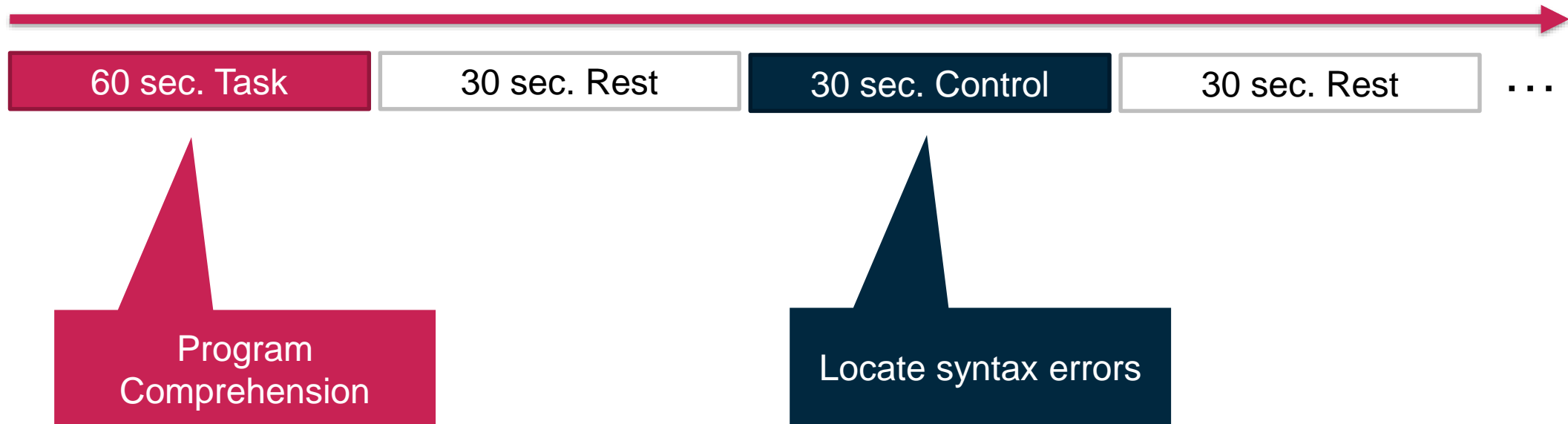


Sample Experiment II: Program Comprehension

```
def method_name():  
    array = [2, 19, 5, 17]  
    result = array[0]  
    for i in array:  
        if i > result:  
            result = i  
    print(result)
```

Experiment Design & Study Requirements

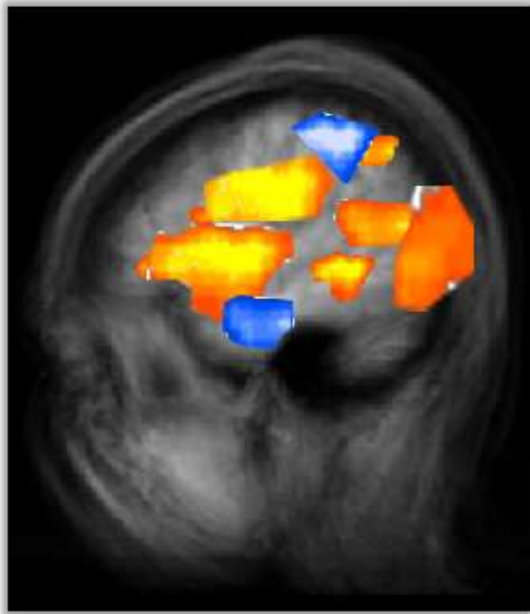
- Block design



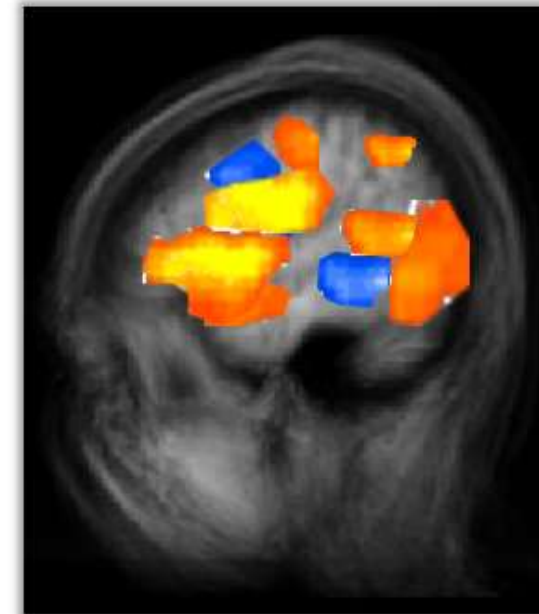
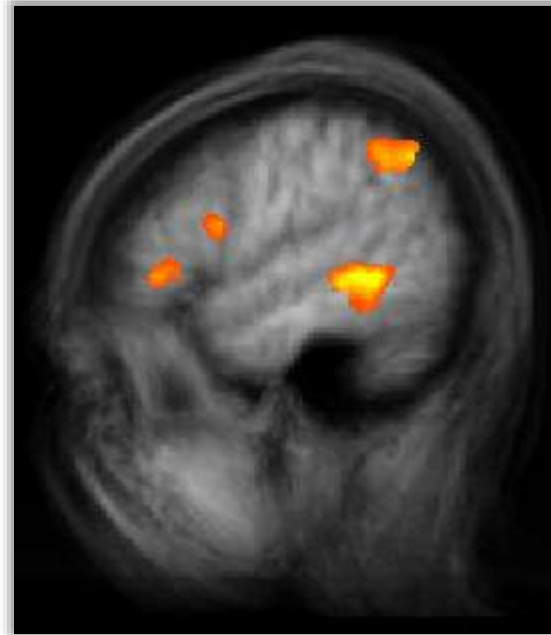
Sample Experiment III: Syntax Errors

```
def method_name():  
    array [2, 19, 5, 17]  
    result = array[0]  
    for i in array  
        if i > result:  
            result = i  
    print result)
```

Contrasts



Program
Comprehension

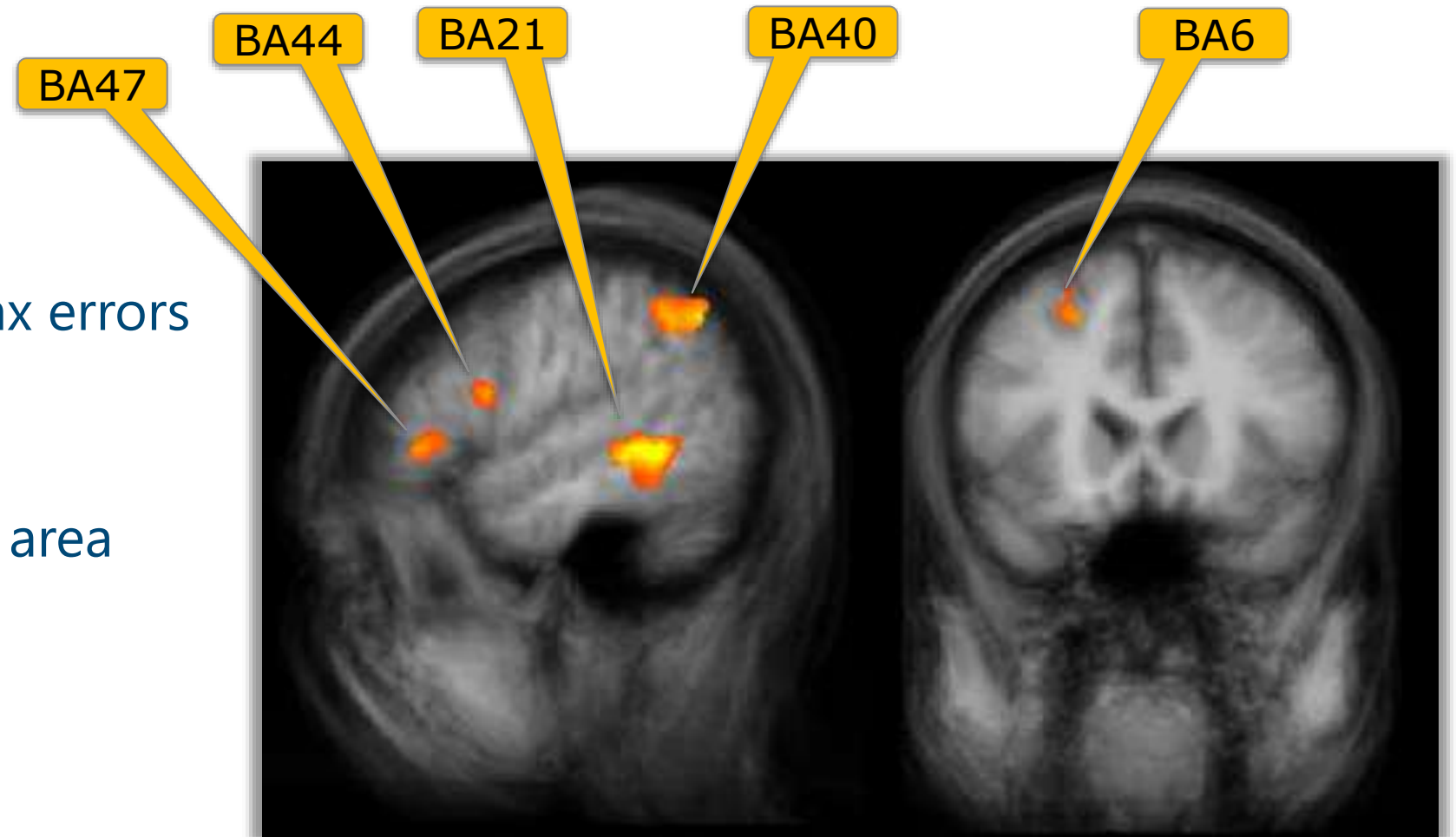


Control

Program Comprehension - Control

How Do We Understand Code? [Siegmund14]

- Contrast between comprehension and syntax errors
- Network of brain areas
- No “classic” logic brain area
- All left lateralized



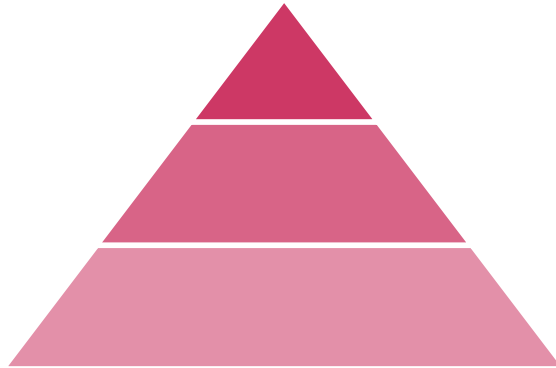
fMRI Study 2 on (Top-Down) Program Comprehension



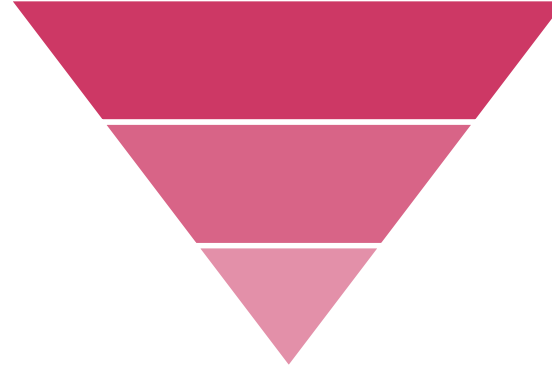
Sample Experiment IV: Program Comprehension

```
def array_average(numbers):  
    counter = 0  
    sum = 0  
  
    for number in numbers:  
        sum = sum + number  
        counter = counter + 1  
  
    average = sum / counter  
    return average
```

Bottom Up versus Top-Down Comprehension



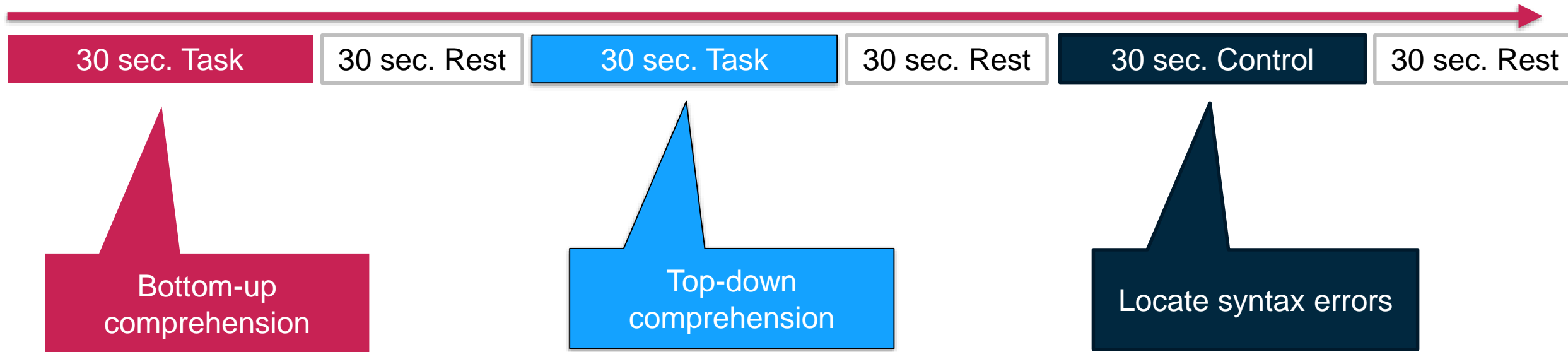
Bottom-Up Comprehension



Top-Down Comprehension

```
1 public static void main() {  
2     String text = "The quick brown fox jumps";  
3     System.out.print(getLengthOfLastWord(text));  
4 }  
5  
6 static int getLengthOfLastWord(String text) {  
7     int lengthOfLastWord = 0;  
8     boolean isLastWord = false;  
9     for (int i = text.length() - 1; i >= 0; i--) {  
10        char c = text.charAt(i);  
11        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {  
12            isLastWord = true;  
13            lengthOfLastWord++;  
14        } else {  
15            if (isLastWord)  
16                break;  
17        }  
18    }  
19  
20    return lengthOfLastWord;  
21 }
```

Top-Down vs. Bottom-Up Comprehension



Example Experiment V: Chunking



492 412 202 327 042 023










No problem, right?






Example Experiment V: Chunking

+49 24122023 27042023
Christmas Today

“Chunking”

Example Experiment VI: Numbers

1 
2 
3 
4 
5 
6 
7 
8 
9 

2 7 4 2 3
    

7	8	9
4	5	6
1	2	3

Chunking in Programming

```
public class A {  
    String b;  
    String c;  
}
```

Chunking in Programming



```
public class DataContainer {  
    String name;  
    String title;  
}
```


Chunking in Programming



```
public class Speaker {  
    String name;  
    String title;  
}
```

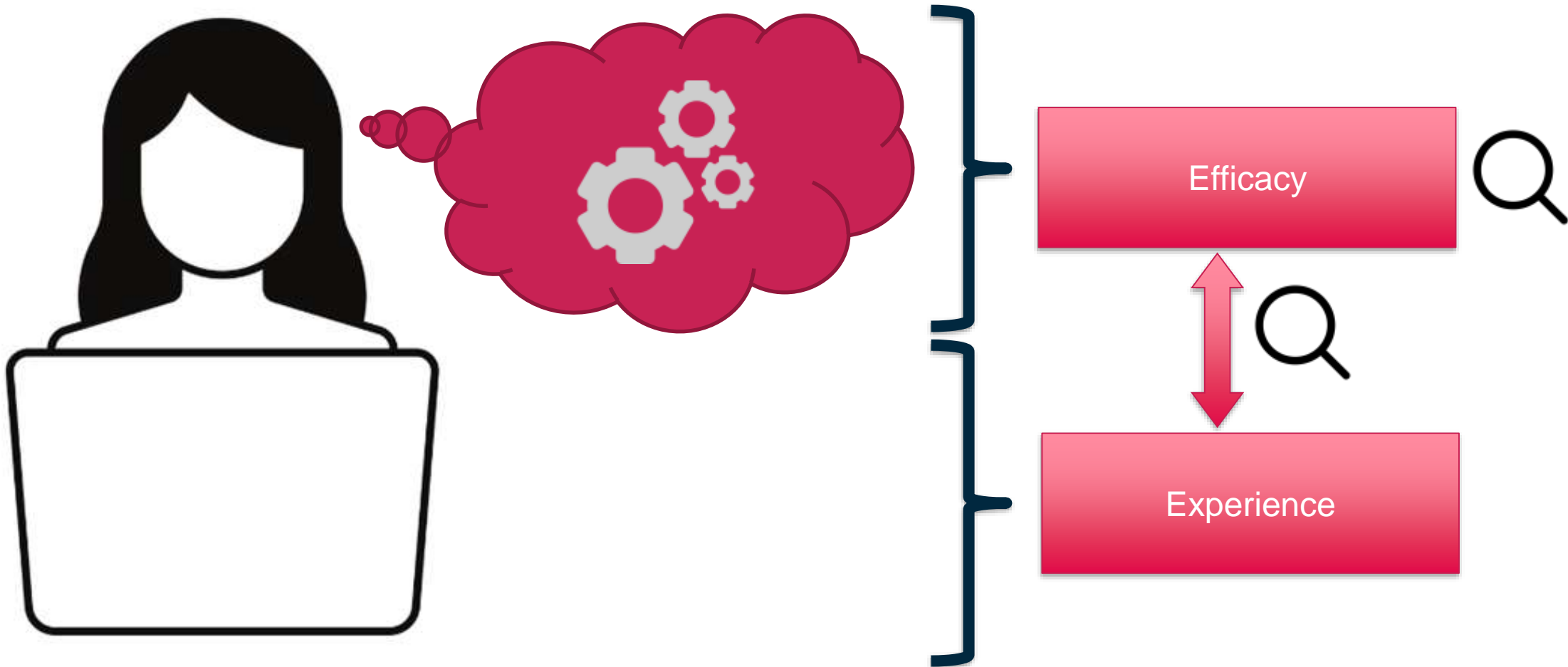
Chunking in Programming

```
public class ConferenceSpeaker {  
    String speakerName;  
    String talkTitle;  
}
```

EEG Study: Programmer Efficacy



Motivation



Participants

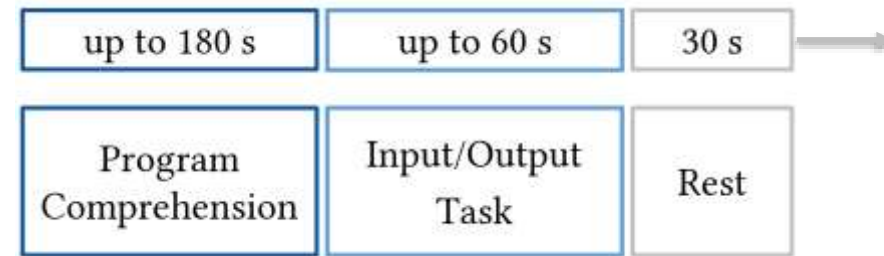
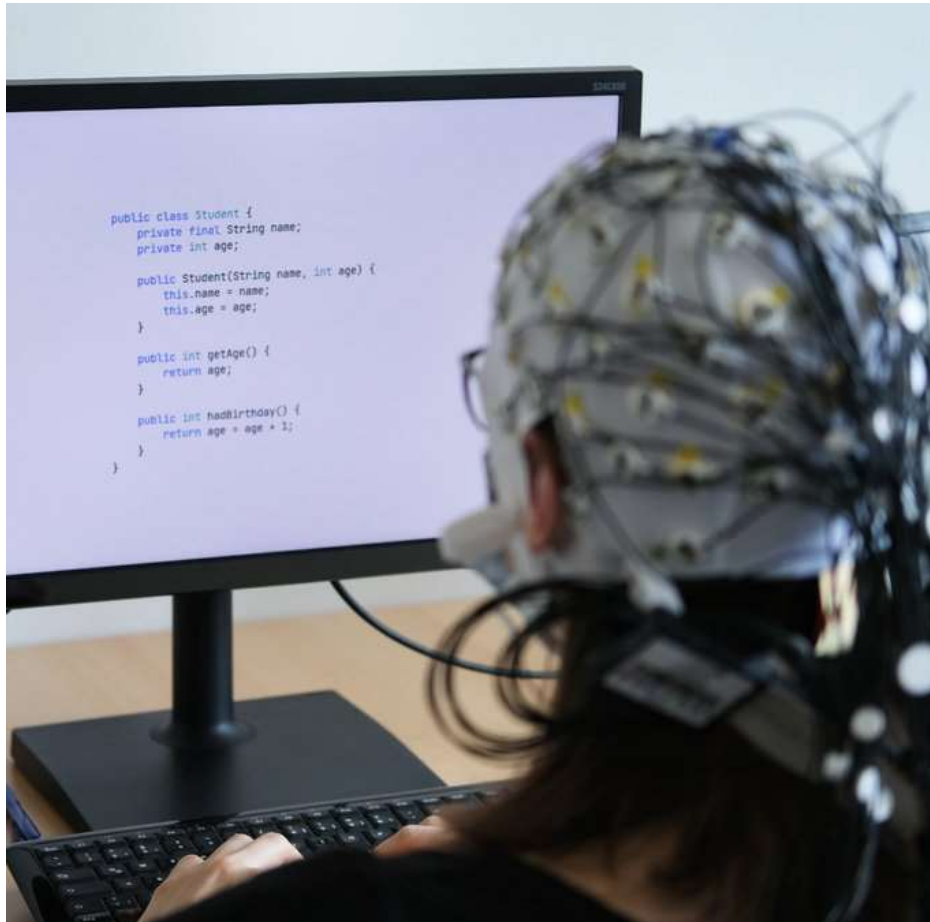
Demographics

Invited Participants	37
Female	5
Male	31
Non-Binary	1
Age (in Years)	25.95 ± 6.76

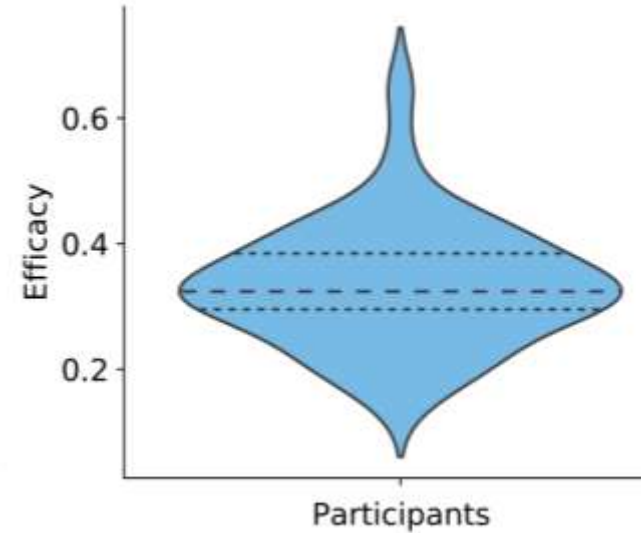
Programming Experience Measures

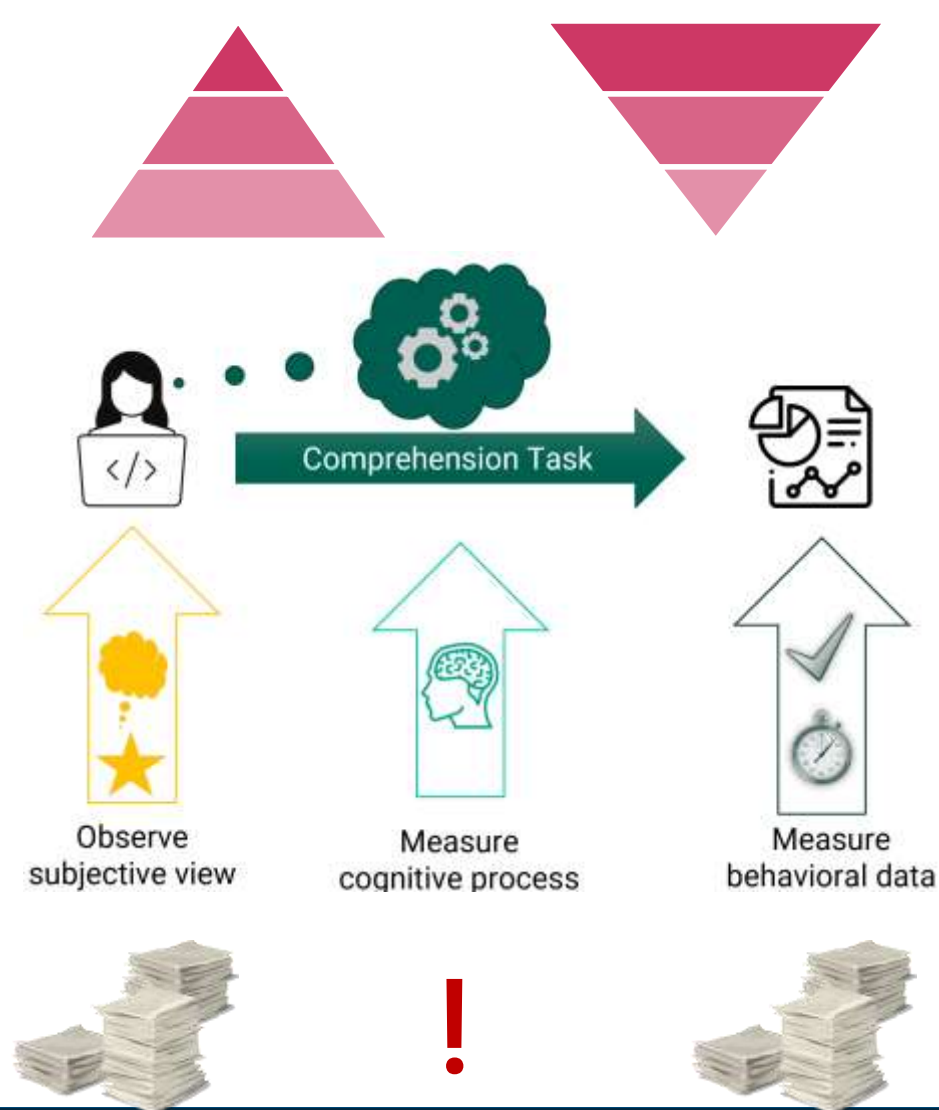
Employment	Undergraduate/graduate student	27 of 37
	... of which work (at least part time)	14 of 27
	Full-time professionals	10 of 37
Years of Programming	Years of Learning Programming	7.93 ± 6.14
	Years of Professional Programming	3.55 ± 4.30
	Number of Known Programming Languages	5.11 ± 2.02
Self-Estimation	Comparison to Peers	3.67 ± 0.76
	Comparison to 10-Year Professional	2.25 ± 0.94
Work Stats	Hours per Week Spent in Software Engineering	24.76 ± 21.08
	Hours per Week Spent Programming	10.78 ± 11.36

Method: Measures



Comprehend up to 32 Java source-code snippets

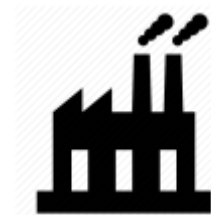




Theory of Program Comprehension



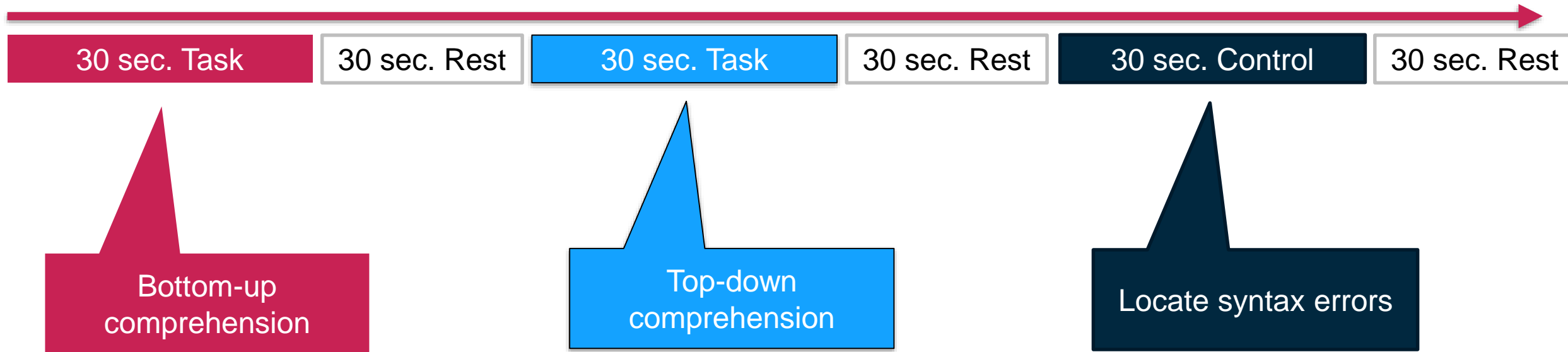
Impact on SE Practice



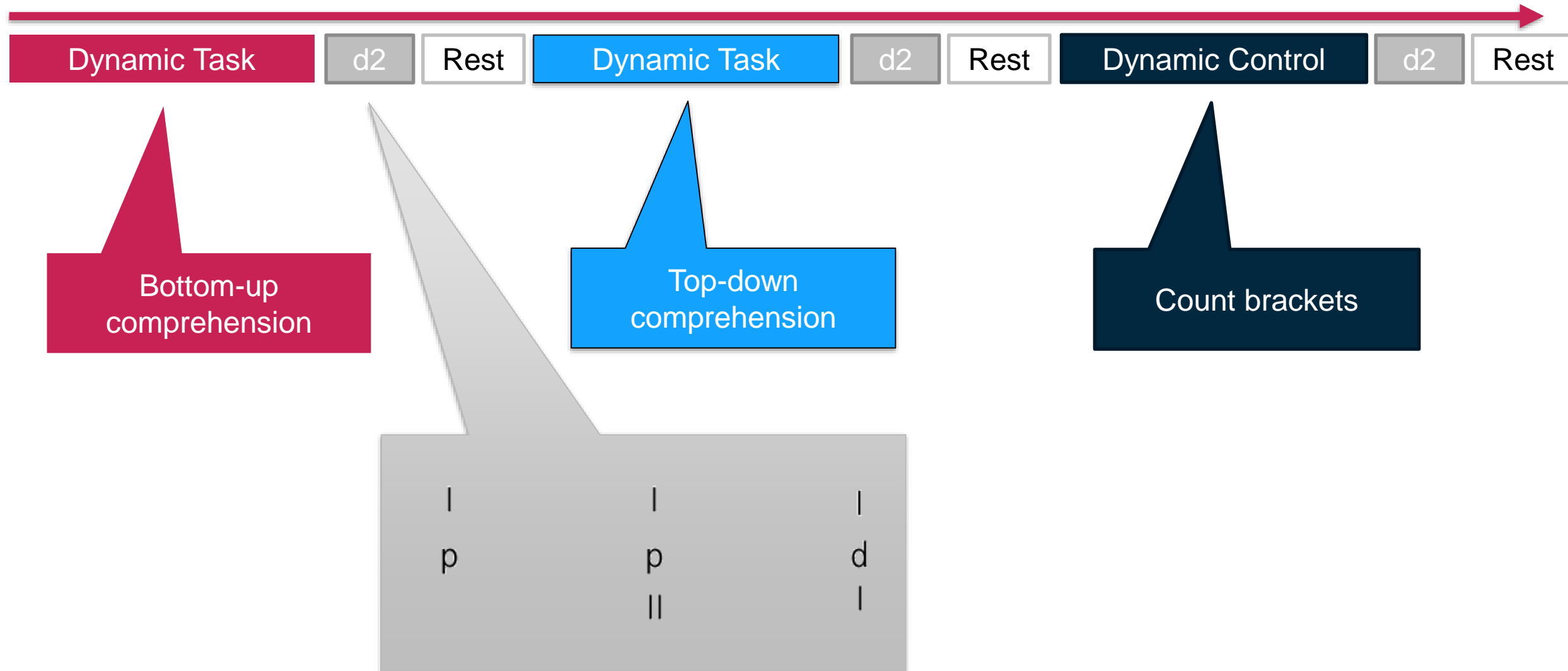
Part IV: Perspectives & Future Research



Problems with Experiment Design



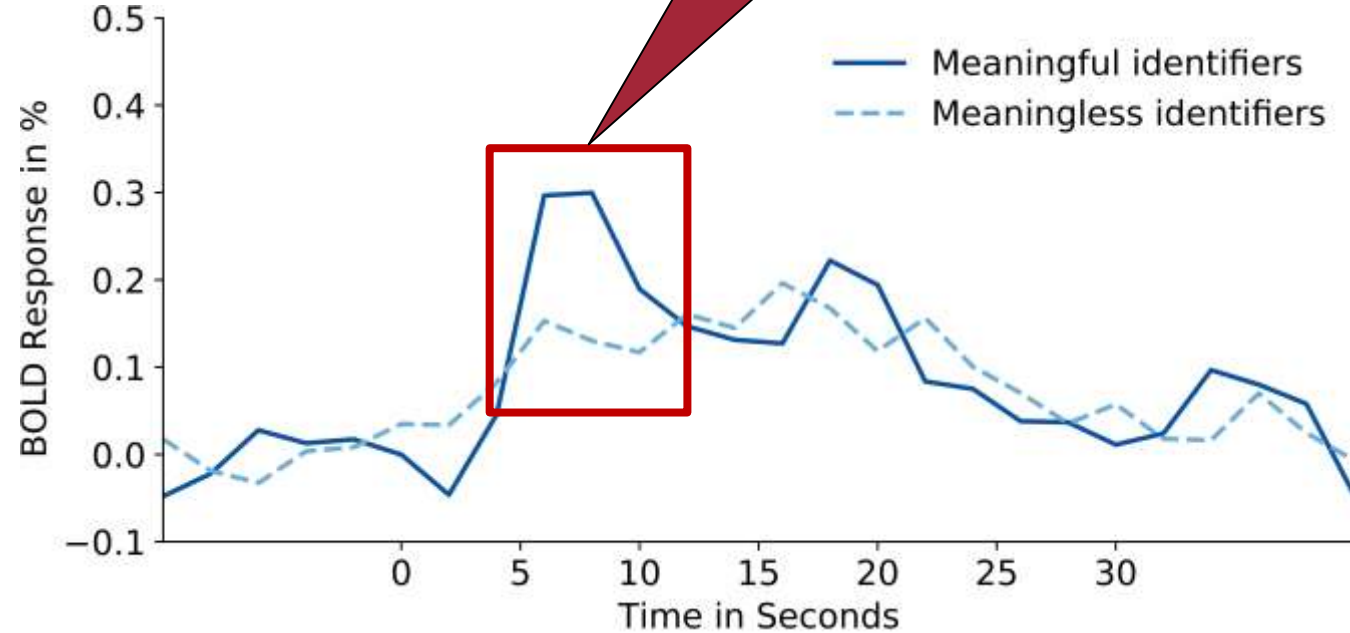
Problems with Experiment Design



Dynamic Brain Activation



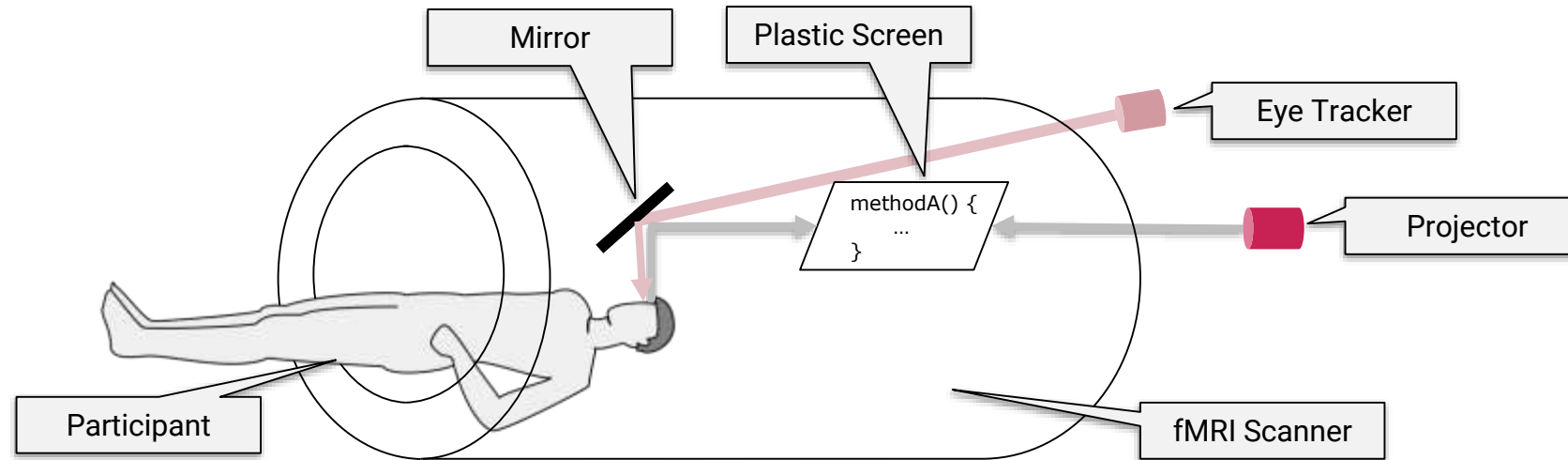
BA21



`sort()`

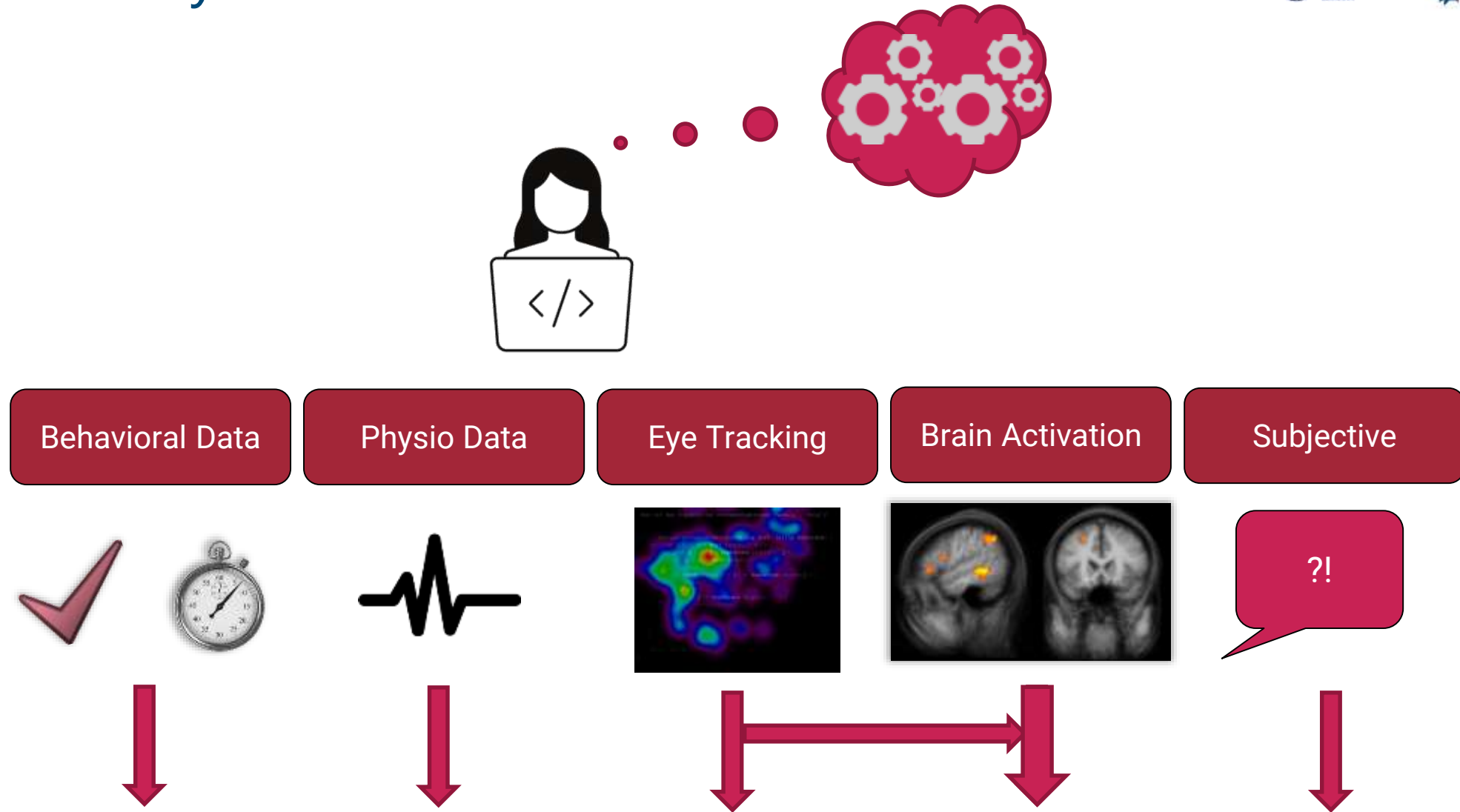
`method()`

Dynamic Brain Activation



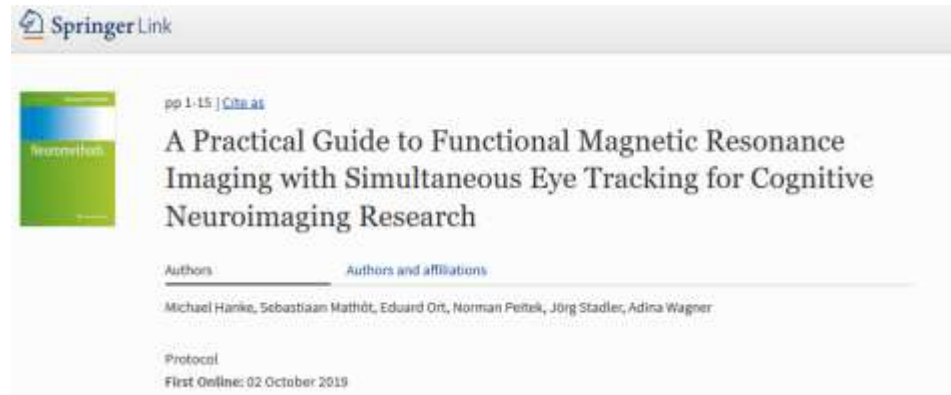
```
public class Street {  
    private int number;  
  
    public Street(int nr) {  
        setNumber(nr);  
    }  
  
    public int getNumber() {  
        return number;  
    }  
  
    public void setNumber(int number) {  
        this.number = number;  
    }  
  
    public static void main(String[] args) {  
        Street street = new Street(5);  
        street.setNumber(15);  
        System.out.print(street.getNumber());  
    }  
}
```

Multi-Modality

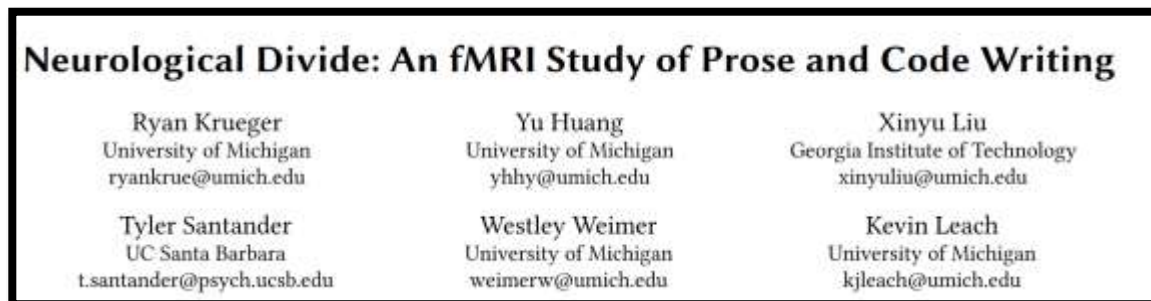


SE Pushes Neuroscience Forward

- Integration of Eye Tracking



- fMRI-compatible Keyboard

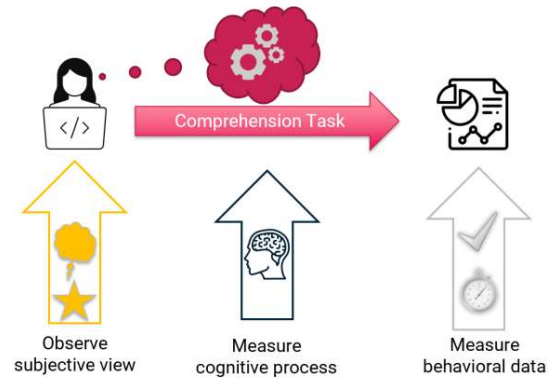




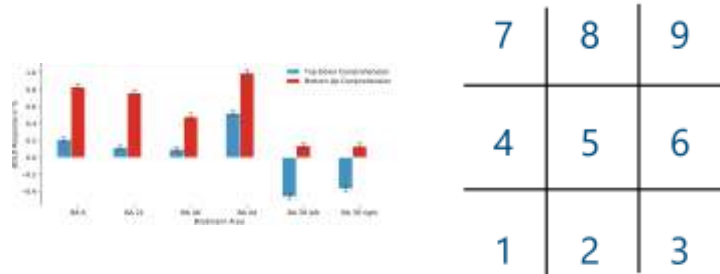
- Pay attention to experiment methods in your paper and related work
- Develop your own idea
 - Could target cognitive processes, different languages, groups of programmers, ...
 - Participant selection
 - What do you want to observe?
 - Consistency between goals and (detailed) methods is important

Conclusion

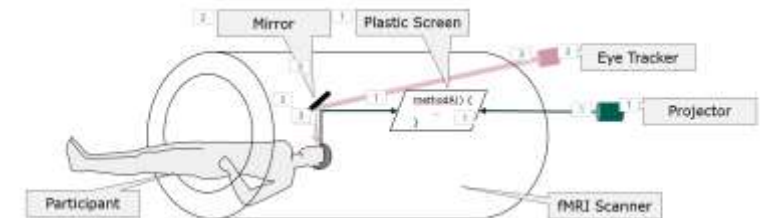
Motivation for Neuroscience & SE



Impact of fMRI Studies in Research and Practice



Future



Questions?