TECHNISCHE UNIVERSITÄT
IN DER KULTURHAUPTSTADT EUROPAS
CHEMNITZ

Dissertation Defense

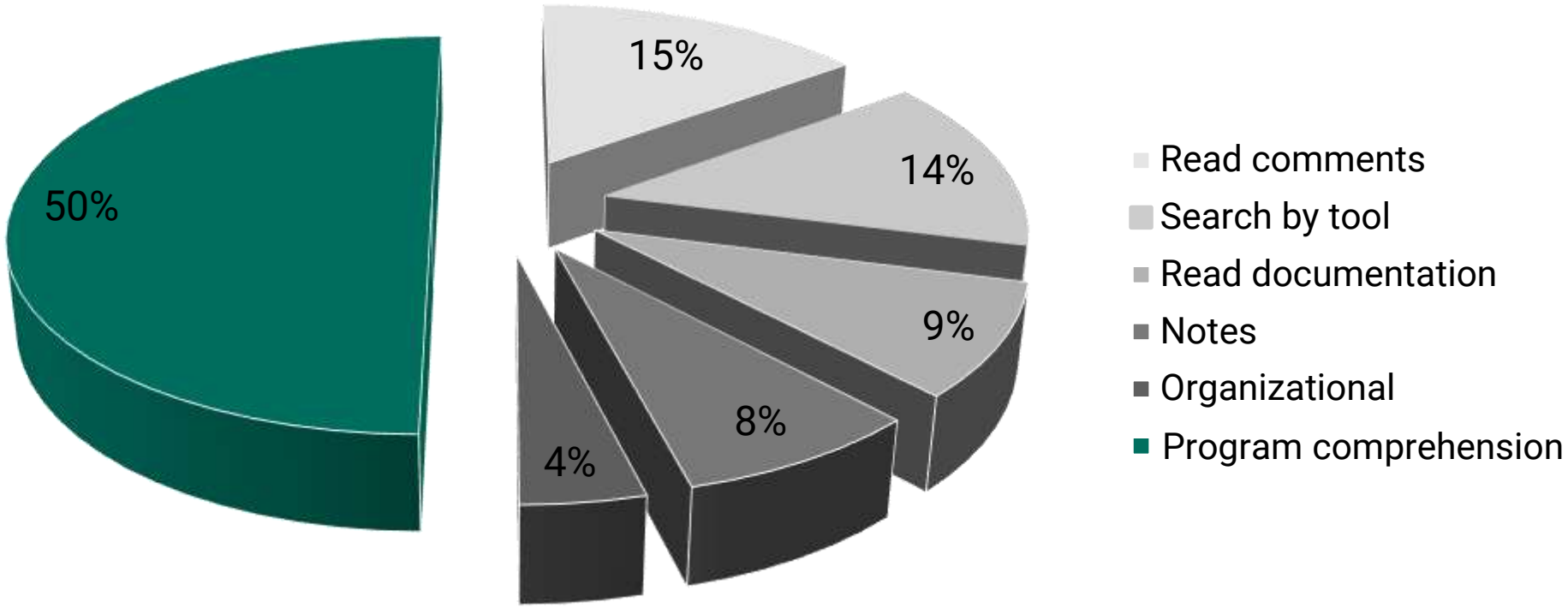
A Neuro-Cognitive Perspective of Program Comprehension

Norman Peitek

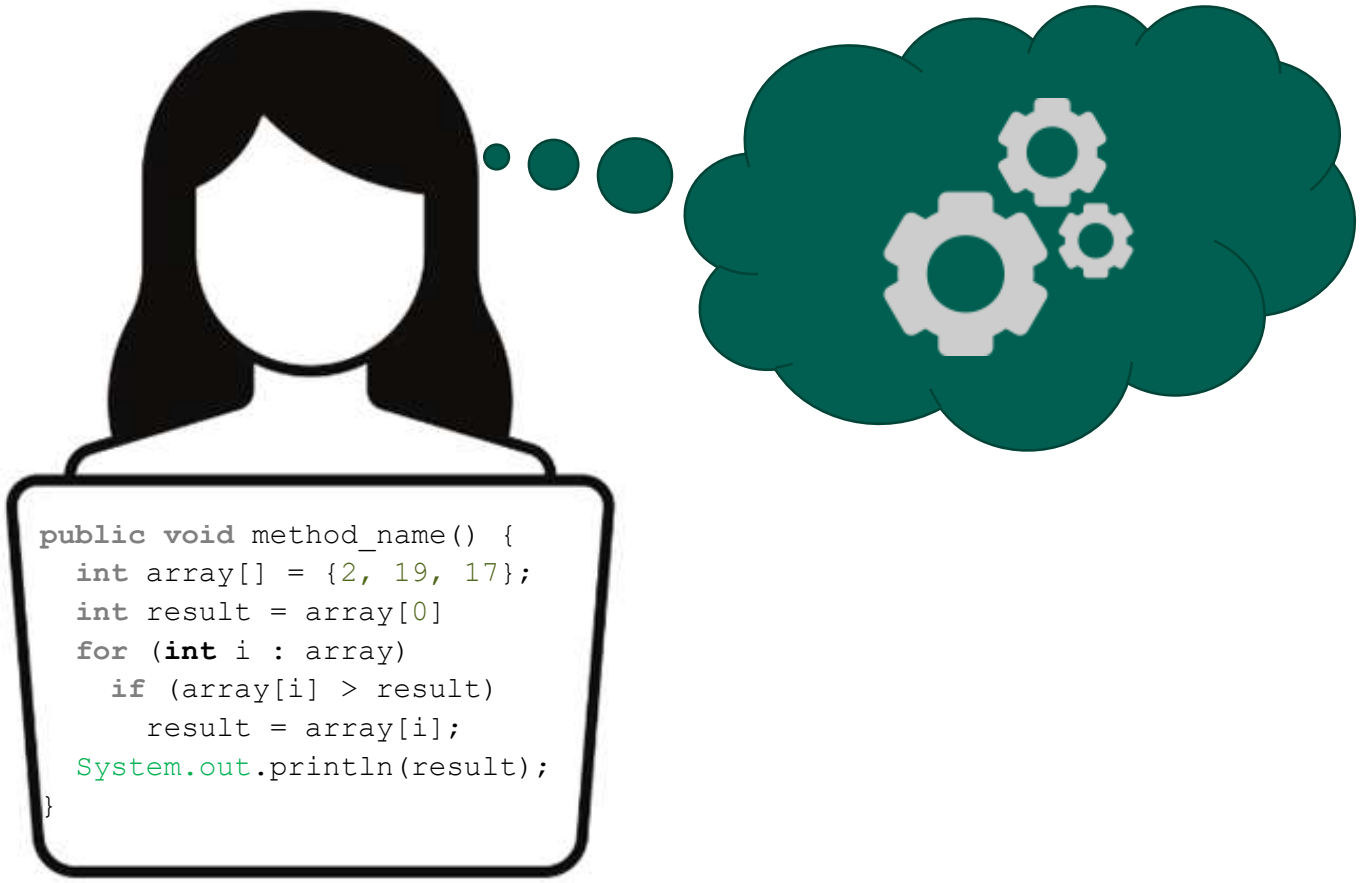
27 April 2022

Motivation: Program Comprehension

A Neuro-Cognitive Perspective of Program Comprehension



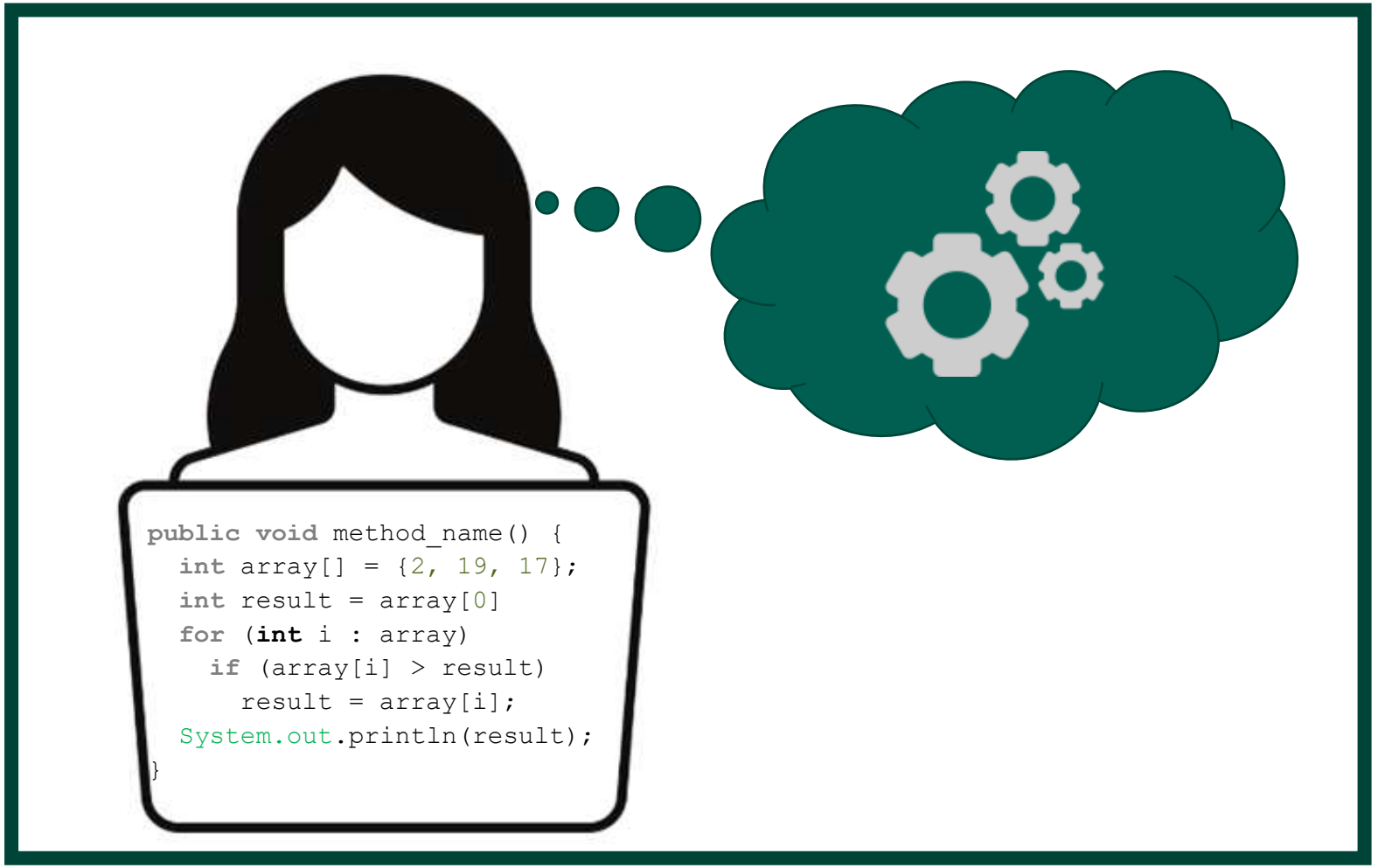
A Neuro-Cognitive Perspective of Program Comprehension



```
public void method_name() {
  int array[] = {2, 19, 17};
  int result = array[0]
  for (int i : array)
    if (array[i] > result)
      result = array[i];
  System.out.println(result);
}
```

Motivation: Program Comprehension

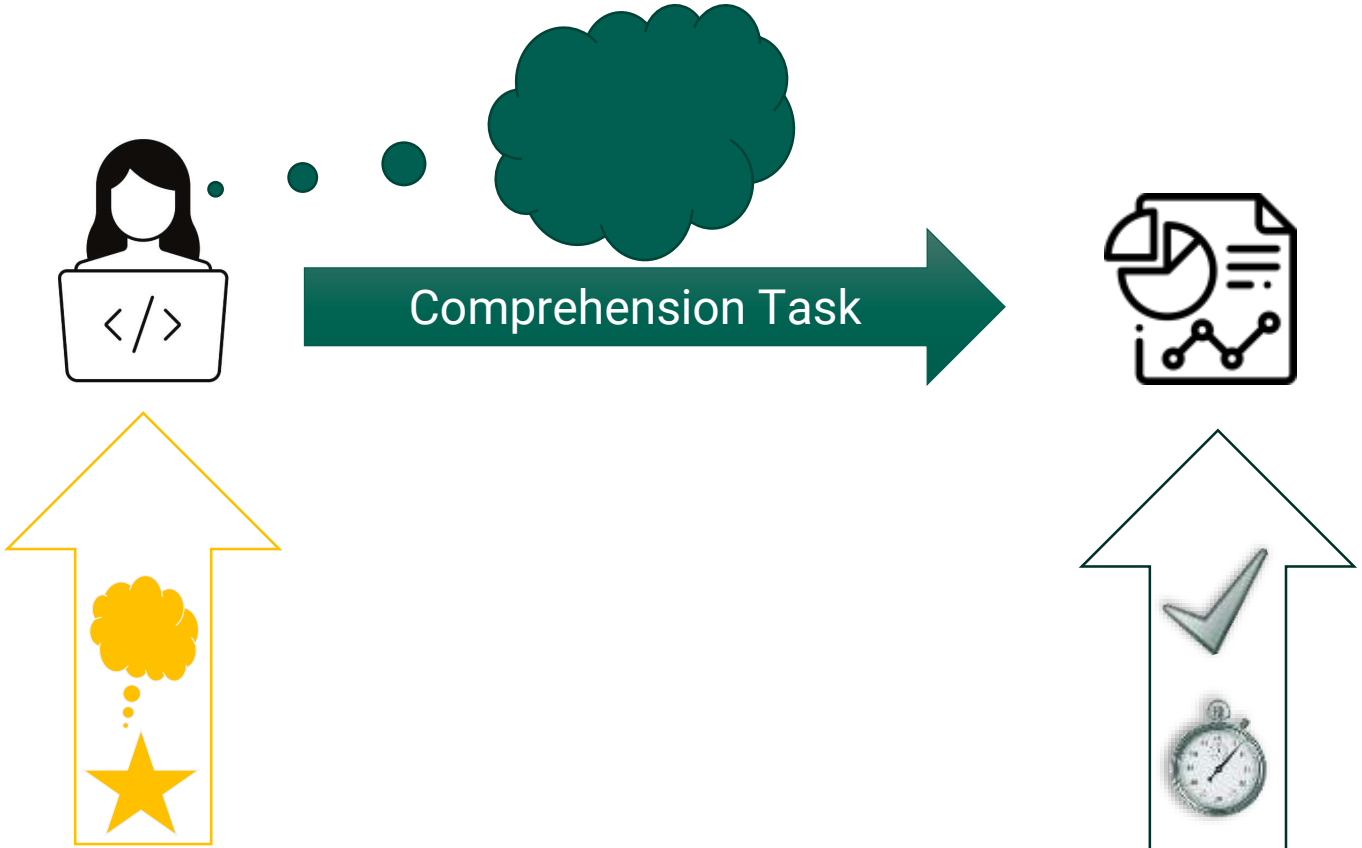
A Neuro-Cognitive Perspective of Program Comprehension



- Software Engineering
- Programming Languages
- Education & Training

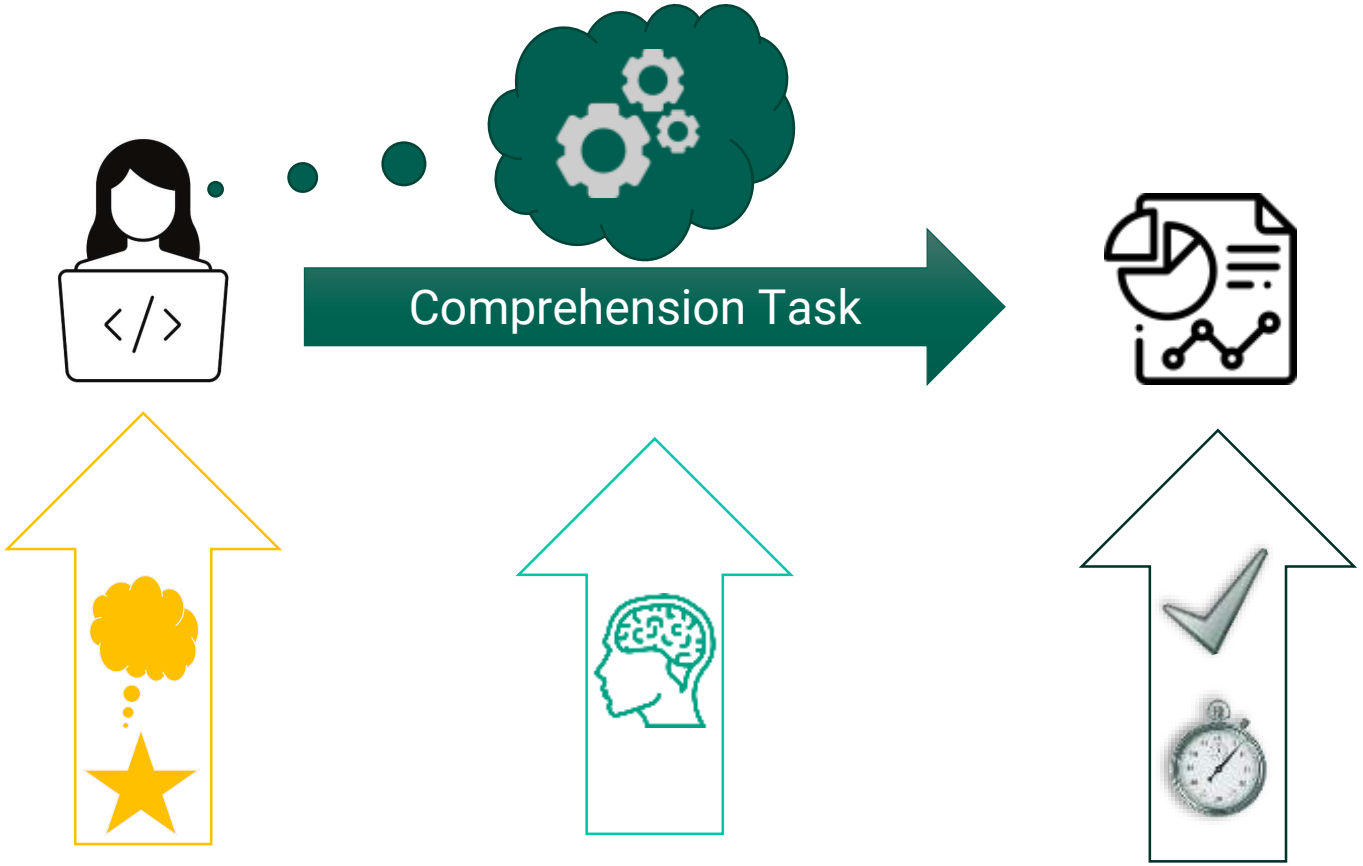
Motivation: Past Research

A Neuro-Cognitive Perspective of Program Comprehension

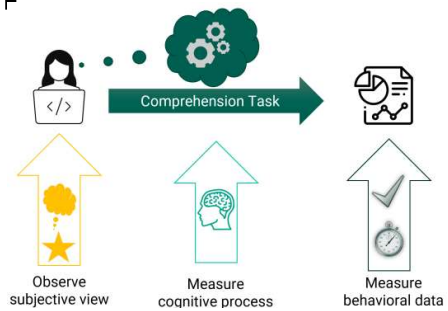


Motivation: Neuro-Cognitive Perspective

A Neuro-Cognitive Perspective of Program Comprehension



A Neuro-Cognitive Perspective of Program Comprehension



Framework

- fMRI & Eye Tracking
- Multi-Modality
- Tool Support

Cognitive Perspective

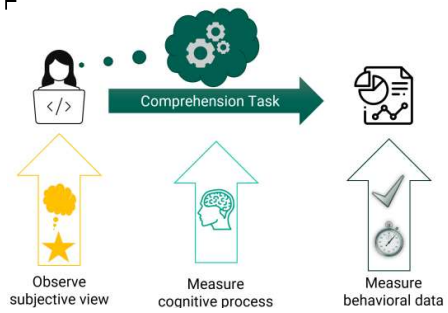
- Top-Down Comprehension
- Reading Order
- Novice & Experienced Programmers

Applications

- Code Complexity Metrics
- Role of Aggregation
- Participant-Specific Brain Parcellation

Outline

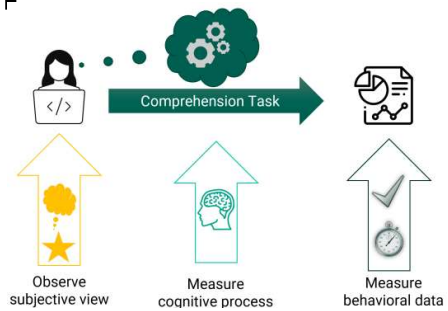
A Neuro-Cognitive Perspective of Program Comprehension



- **Framework**
  - fMRI & Eye Tracking
  - Multi-Modality
  - Tool Support
- **Cognitive Perspective**
  - Top-Down Comprehension
  - Reading Order
  - Novice & Experienced Programmers
- **Applications**
  - Code Complexity Metrics
  - Role of Aggregation
  - Participant-Specific Brain Parcellation

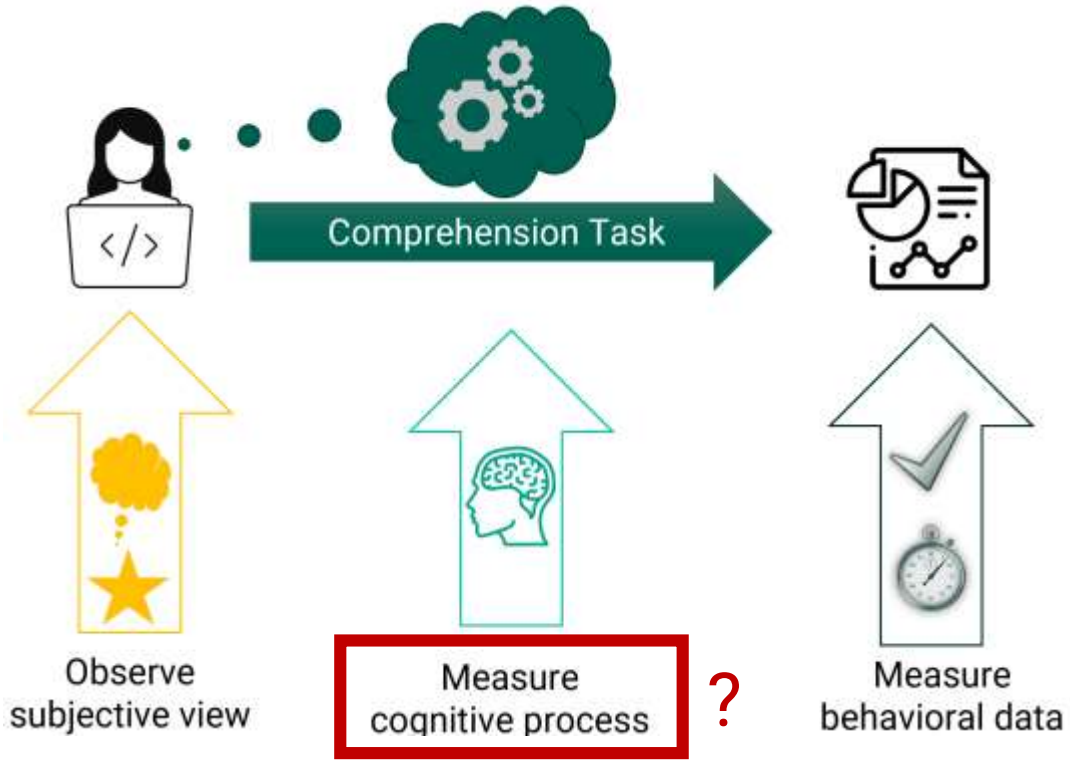
Outline

A Neuro-Cognitive Perspective of Program Comprehension



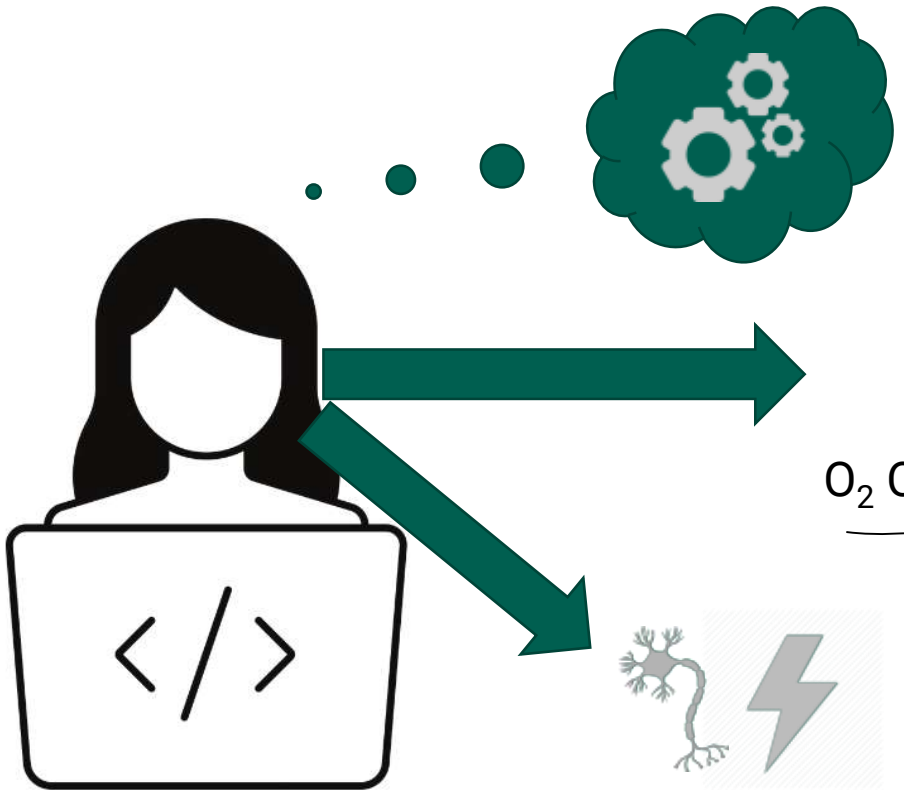
- **Framework**
  - fMRI & Eye Tracking
  - Multi-Modality
  - Tool Support
- **Cognitive Perspective**
  - Top-Down Comprehension
  - Reading Order
  - Novice & Experienced Programmers
- **Applications**
  - Code Complexity Metrics
  - Role of Aggregation
  - Participant-Specific Brain Parcellation

Part I: Framework



functional magnetic resonance imaging (fMRI)

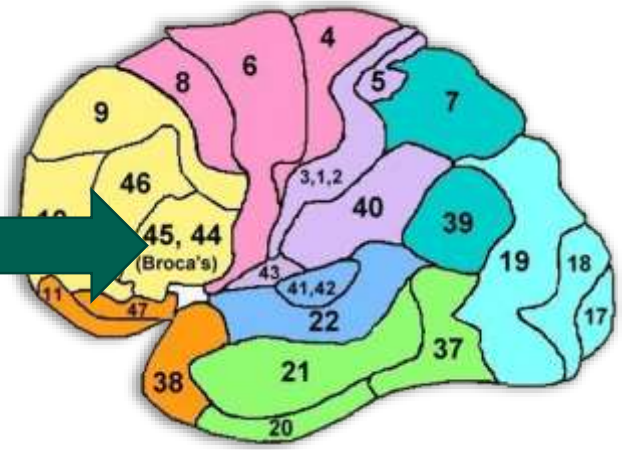
fMRI Basics



O_2 O_2
O_2 O_2

O_2 O_2

BOLD effect



First fMRI Study

Understanding Understanding Source Code with Functional Magnetic Resonance Imaging

Janet Siegmund[a], Christian Kästner[c], Sven Apel[a], Chris Parnin[d], Anja Bethmann[b], Thomas Leich[e], Gunter Saake[a], and André Brechmann[b]
[a]University of Passau, Germany [c]Carnegie Mellon University, USA
[d]Georgia Institute of Technology, USA [b]Leibniz Inst. for Neurobiology Magdeburg, Germany
[e]Metop Research Institute, Magdeburg, Germany [a]University of Magdeburg, Germany

ABSTRACT

Program comprehension is an important cognitive process that inherently eludes direct measurement. Thus, researchers are struggling with providing suitable programming languages, tools, or coding conventions to support developers in their everyday work. In this paper, we explore whether *functional magnetic resonance imaging (fMRI)*, which is well established in cognitive neuroscience, is feasible to soundly measure program comprehension. In a controlled experiment, we observed 17 participants inside an fMRI scanner while they were comprehending short source-code snippets, which we contrasted with locating syntax errors. We found a clear, distinct activation pattern of five brain regions, which are related to working memory, attention, and language processing—all processes that fit well to our understanding of program comprehension. Our results encourage us and, hopefully, other researchers to use fMRI in future studies to measure program comprehension and, in the long run, answer questions, such as: Can we predict whether someone will be an excellent programmer? How effective are new languages and tools for program understanding? How should we train programmers?

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: User/Machine Systems

General Terms

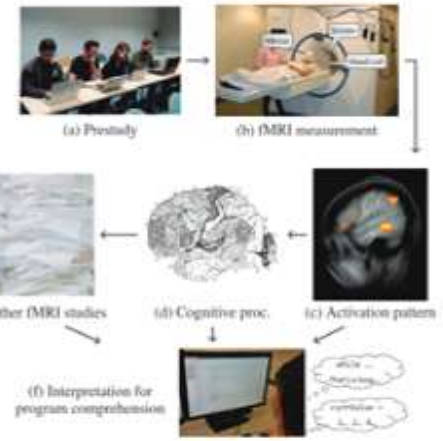
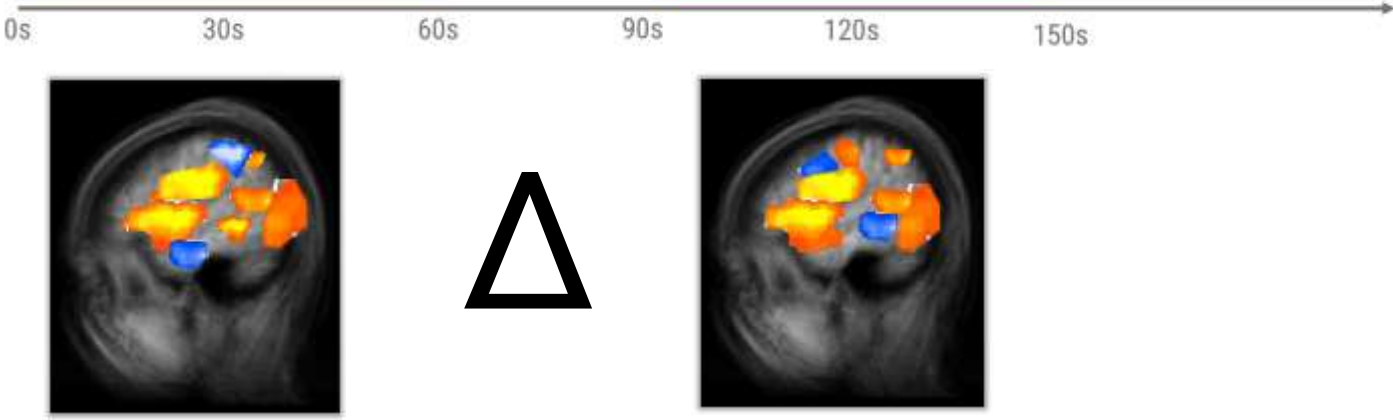


Figure 1: Workflow of our fMRI study.

Program Comprehension



First fMRI Study

Understanding Understanding Source Code with Functional Magnetic Resonance Imaging

Janet Siegmund[a], Christian Kästner[c], Sven Apel[a], Chris Parnin[d], Anja Bethmann[b], Thomas Leich[e], Gunter Saake[a], and André Brechmann[b]
[a]University of Passau, Germany [c]Carnegie Mellon University, USA
[d]Georgia Institute of Technology, USA [b]Leibniz Inst. for Neurobiology Magdeburg, Germany
[e]Metop Research Institute, Magdeburg, Germany [a]University of Magdeburg, Germany

ABSTRACT

Program comprehension is an important cognitive process that inherently eludes direct measurement. Thus, researchers are struggling with providing suitable programming languages, tools, or coding conventions to support developers in their everyday work. In this paper, we explore whether *functional magnetic resonance imaging (fMRI)*, which is well established in cognitive neuroscience, is feasible to soundly measure program comprehension. In a controlled experiment, we observed 17 participants inside an fMRI scanner while they were comprehending short source-code snippets, which we contrasted with locating syntax errors. We found a clear, distinct activation pattern of five brain regions, which are related to working memory, attention, and language processing—all processes that fit well to our understanding of program comprehension. Our results encourage us and, hopefully, other researchers to use fMRI in future studies to measure program comprehension and, in the long run, answer questions, such as: Can we predict whether someone will be an excellent programmer? How effective are new languages and tools for program understanding? How should we train programmers?

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: User/Machine Systems

General Terms

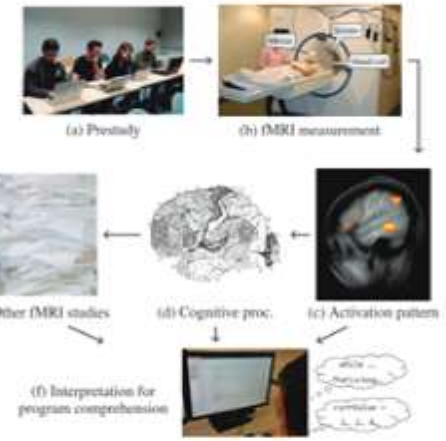
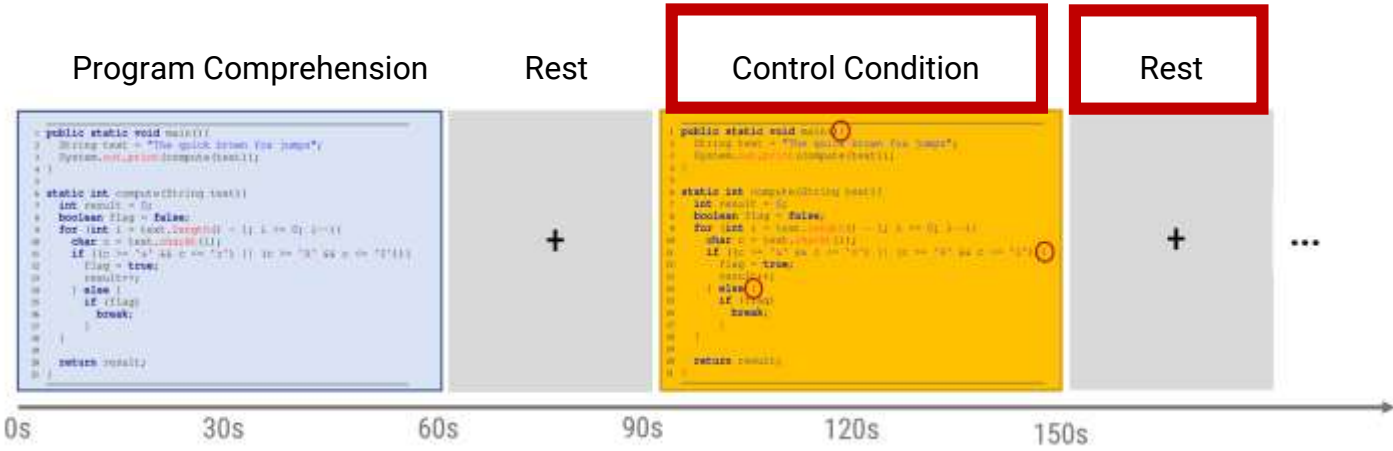
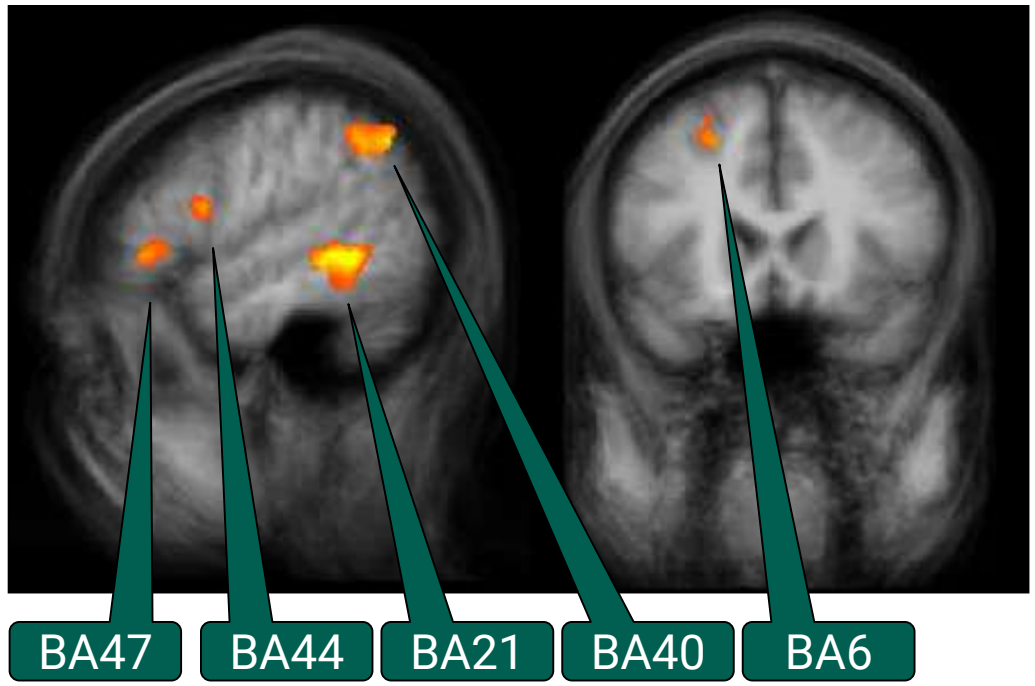
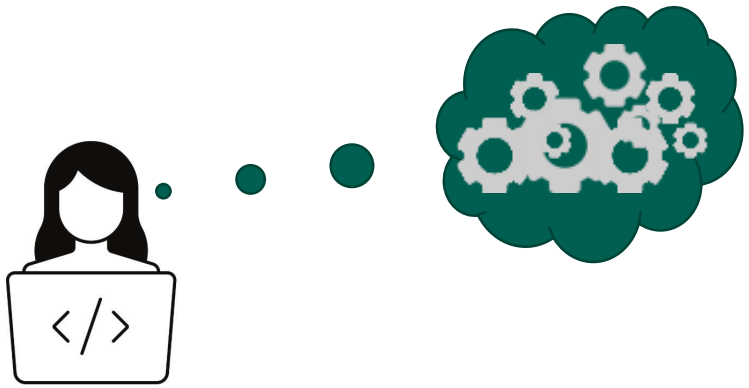


Figure 1: Workflow of our fMRI study.

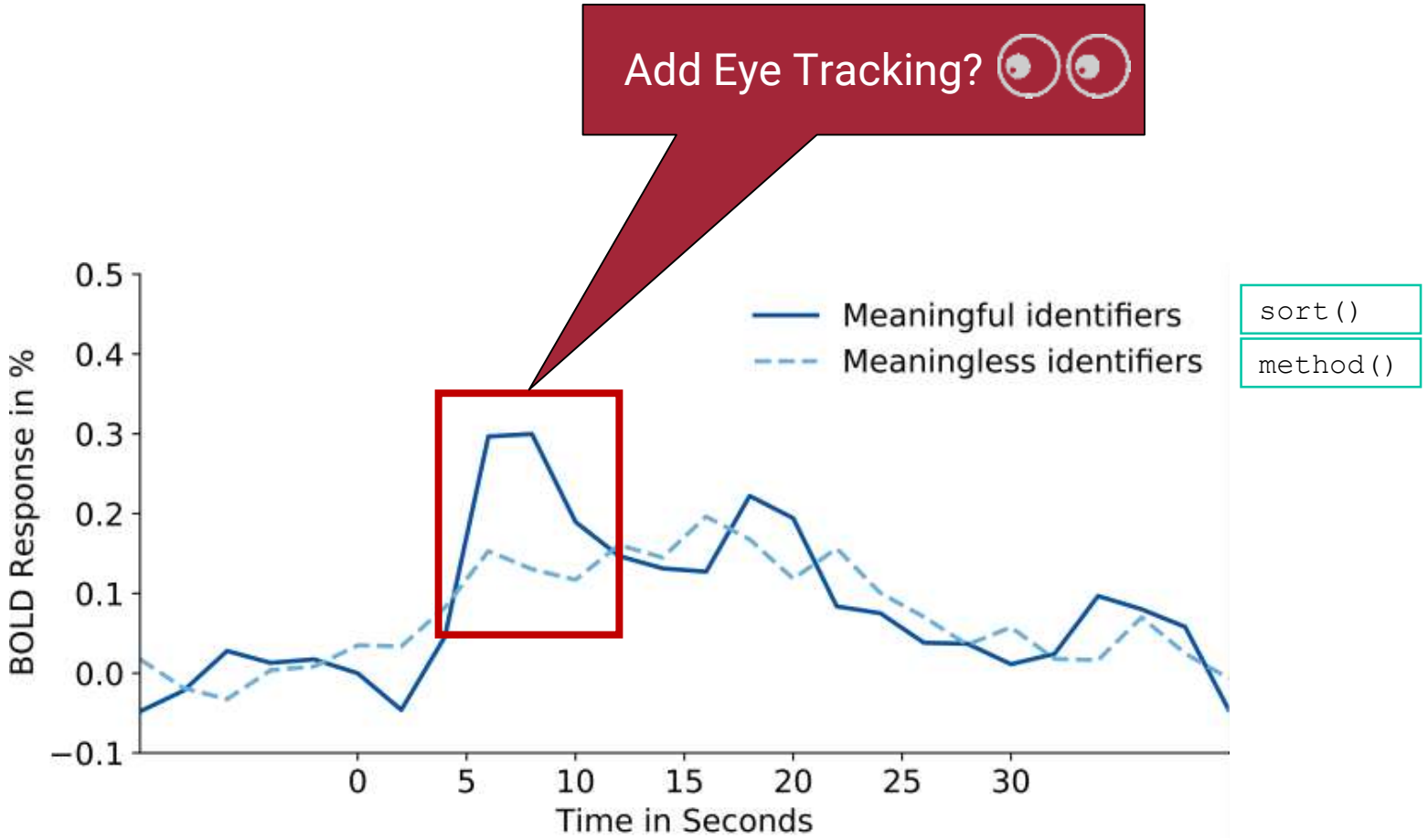




Framework: Limitations of fMRI

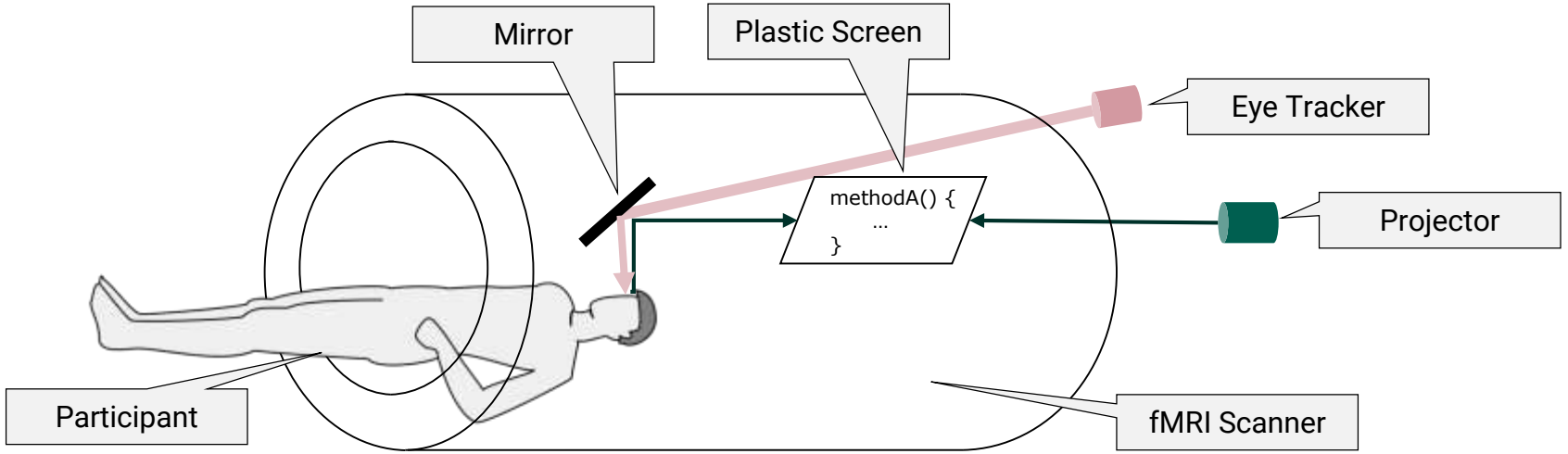


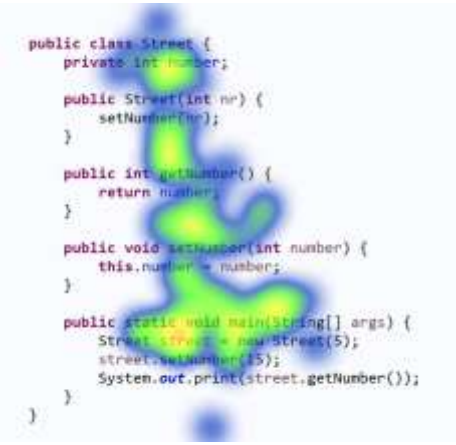
BA21




```
sort()
method()
```

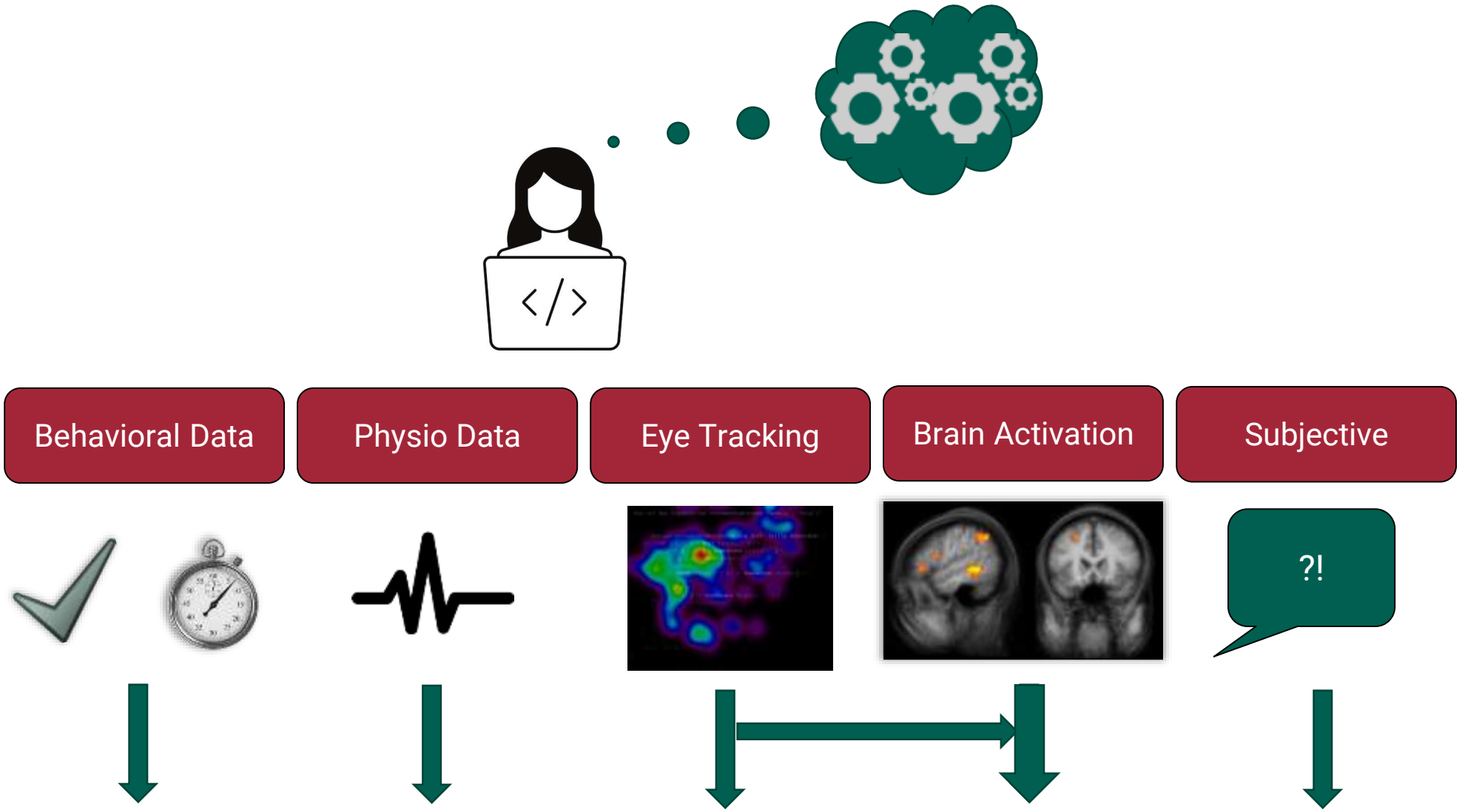
Framework: Eye Tracking in the fMRI Scanner





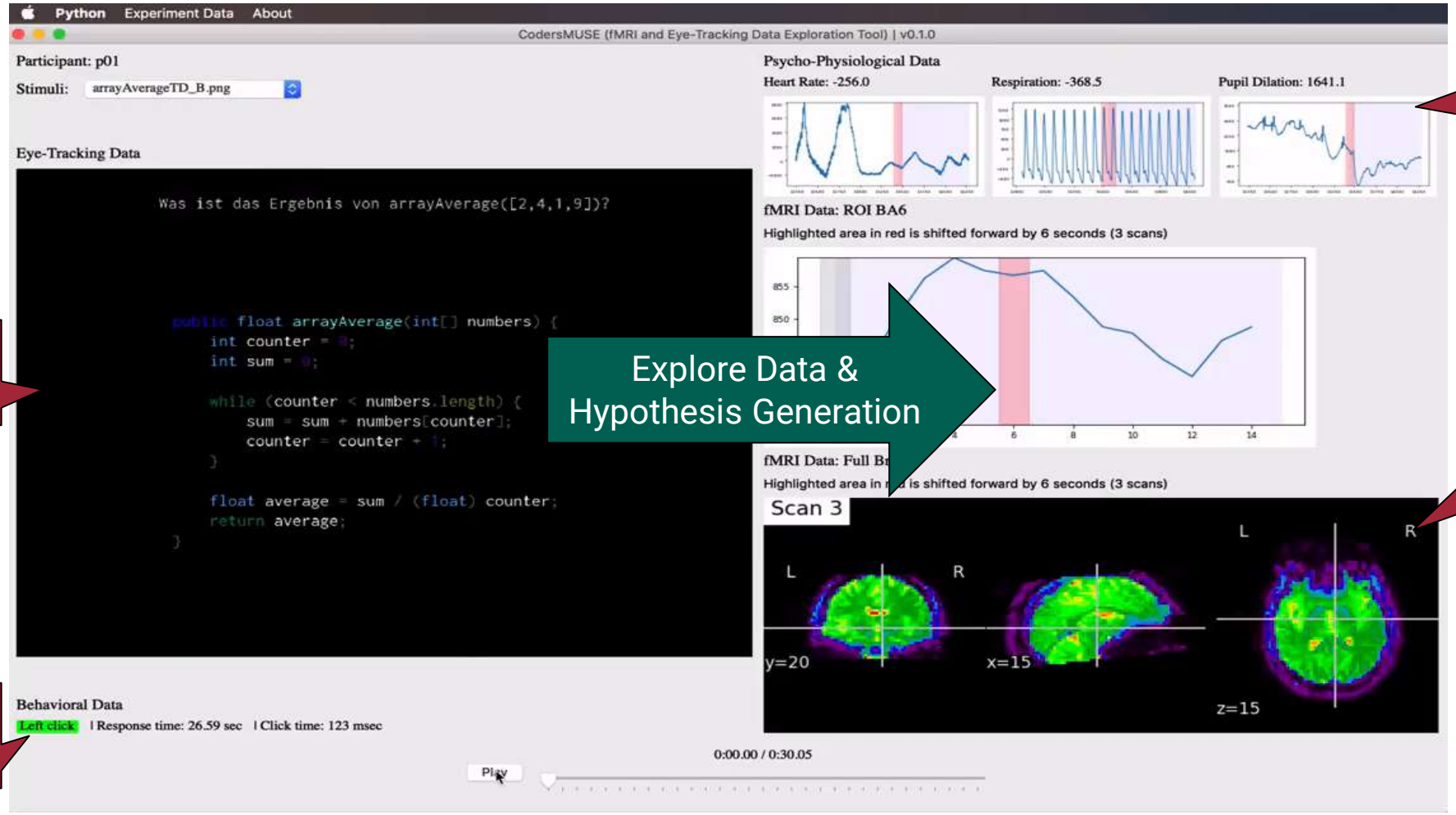
Eye tracking in the fMRI scanner is technically challenging!

Framework: Toward Multi-Modality



Framework: Tool Support

CodersMUSE



Physio Data

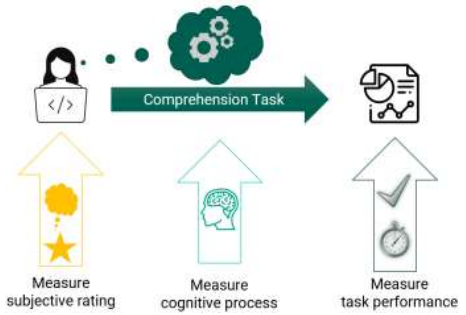
Eye Tracking

Explore Data & Hypothesis Generation

Brain Activation

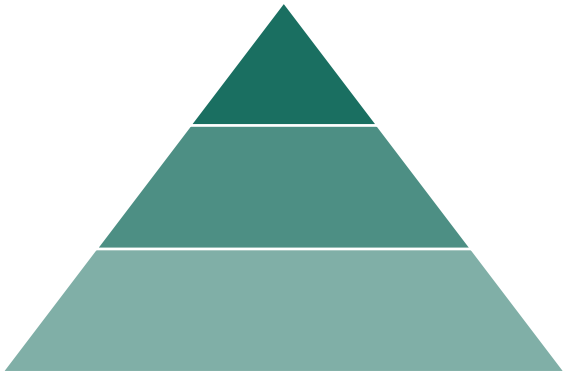
Behavioral Data

Part II: Cognitive Perspective

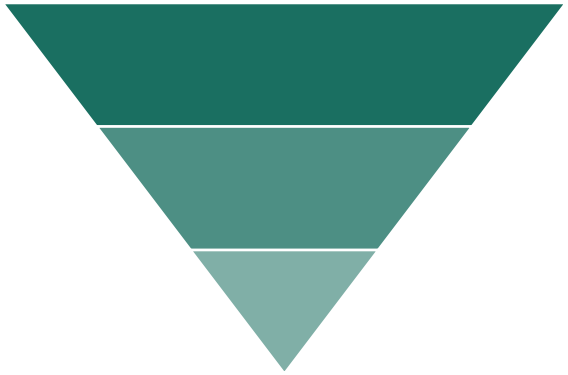


- Framework
  - fMRI & Eye Tracking
  - Multi-Modality
  - Tool Support
- **Cognitive Perspective**
  - **Top-Down Comprehension**
  - Reading Order
  - Novice & Experienced Programmers
- Applications
  - Code Complexity Metrics
  - Role of Aggregation
  - Participant-Specific Brain Parcellation

Cognitive Perspective: Top-Down Comprehension

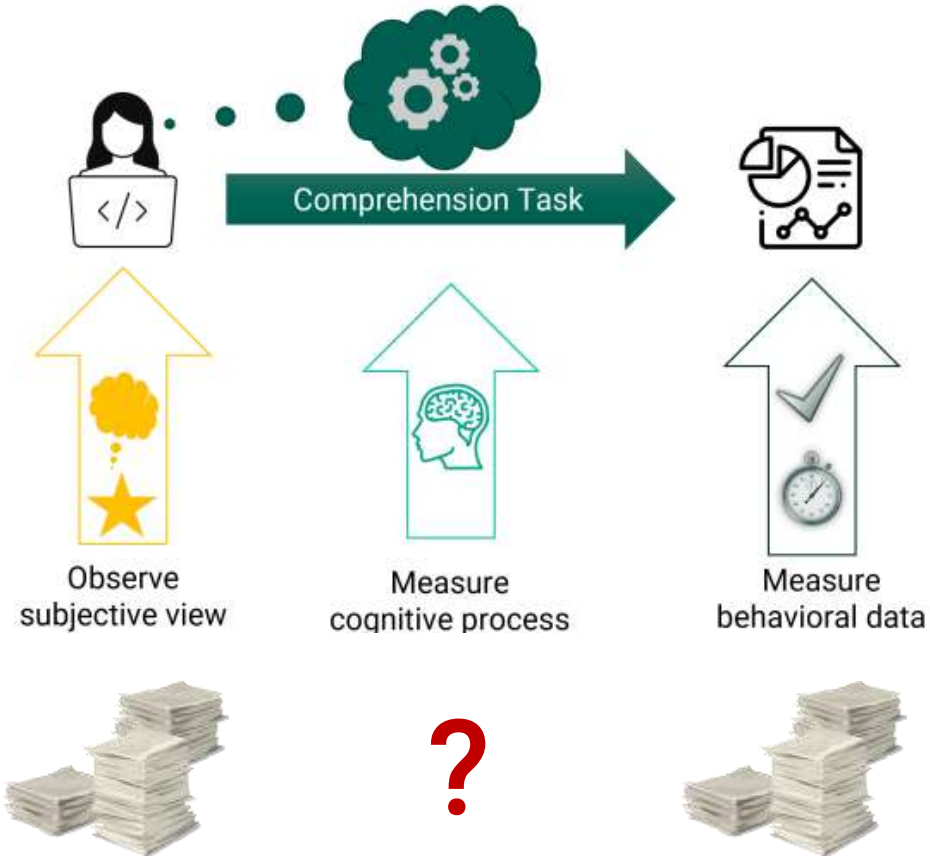


Bottom-Up Comprehension

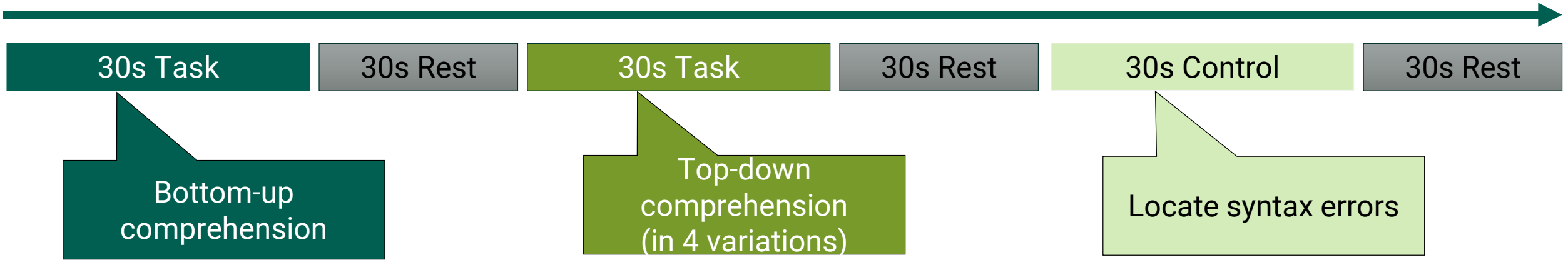


Top-Down Comprehension

```java
public static void main(){
  String text = "The quick brown fox jumps";
  System.out.print(getLengthOfLastWord(text));
}

static int getLengthOfLastWord(String text){
  int lengthOfLastWord = 0;
  boolean isLastWord = false;
  for (int i = text.length() - 1; i >= 0; i--){
    char c = text.charAt(i);
    if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')){
      isLastWord = true;
      lengthOfLastWord++;
    } else {
      if (isLastWord)
        break;
      }
  }

  return lengthOfLastWord;
}
```

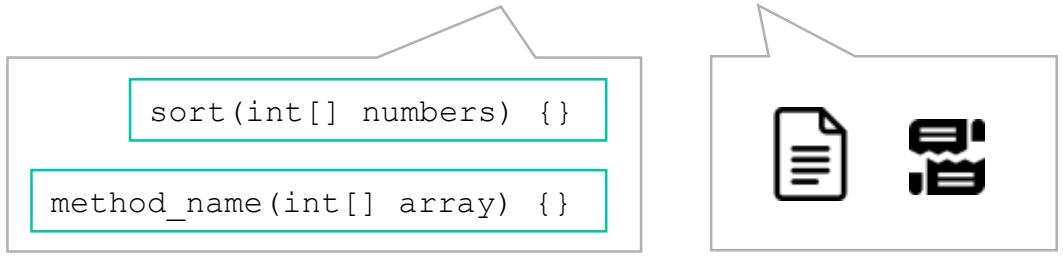


Cognitive Perspective: Experiment Design

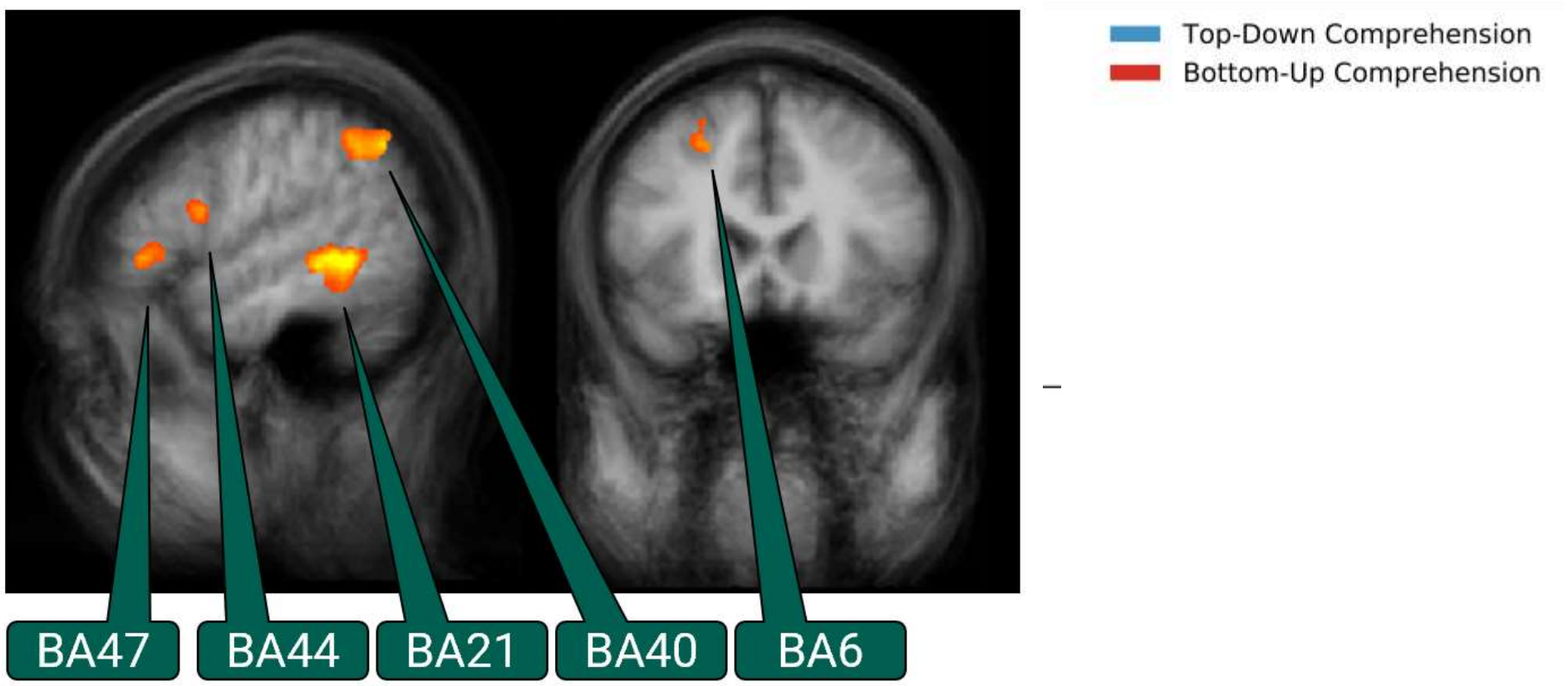


RQ1: Difference between top-down and bottom-up comprehension?

RQ2: How do beacons and layout affect top-down comprehension?

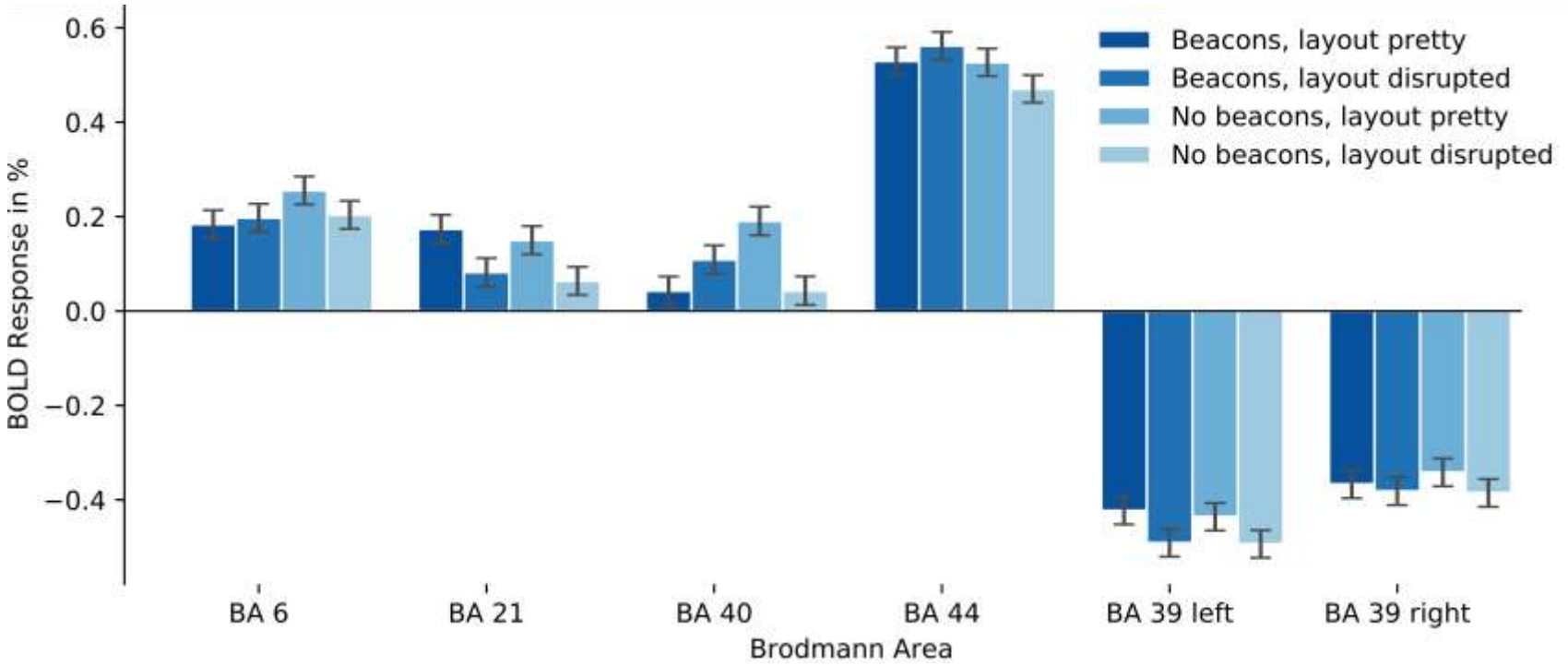


Cognitive Perspective: Results (RQ1)



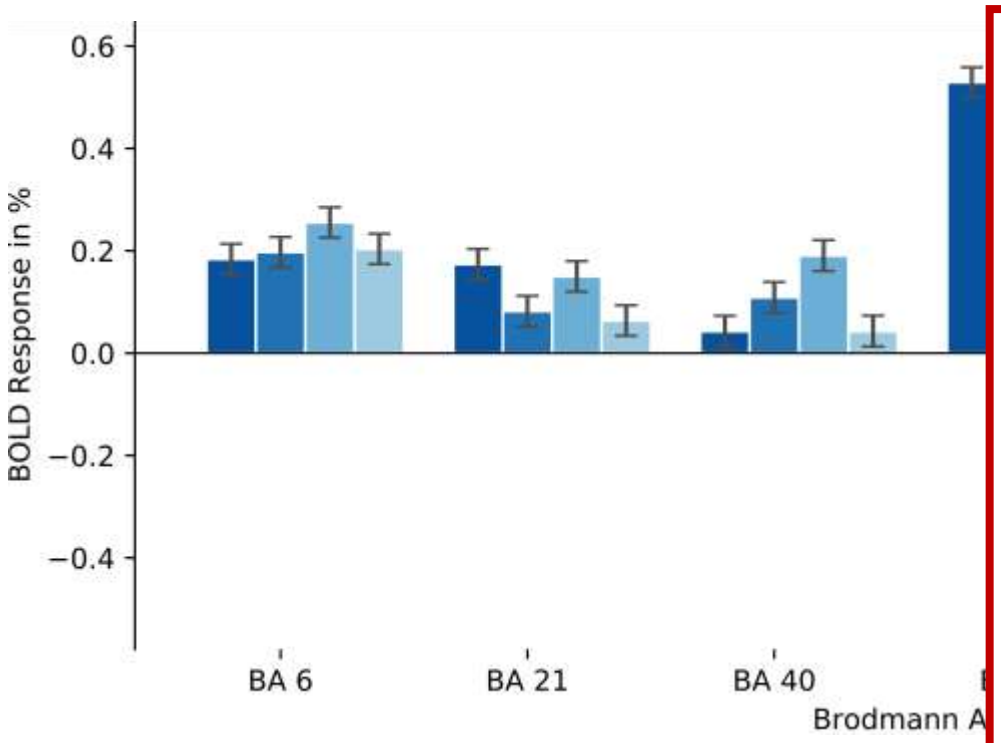
- Neural Efficiency
  - Increased neural efficiency for top-down comprehension
- Semantic Chunking
  - Evidence for semantic chunking during bottom-up comprehension

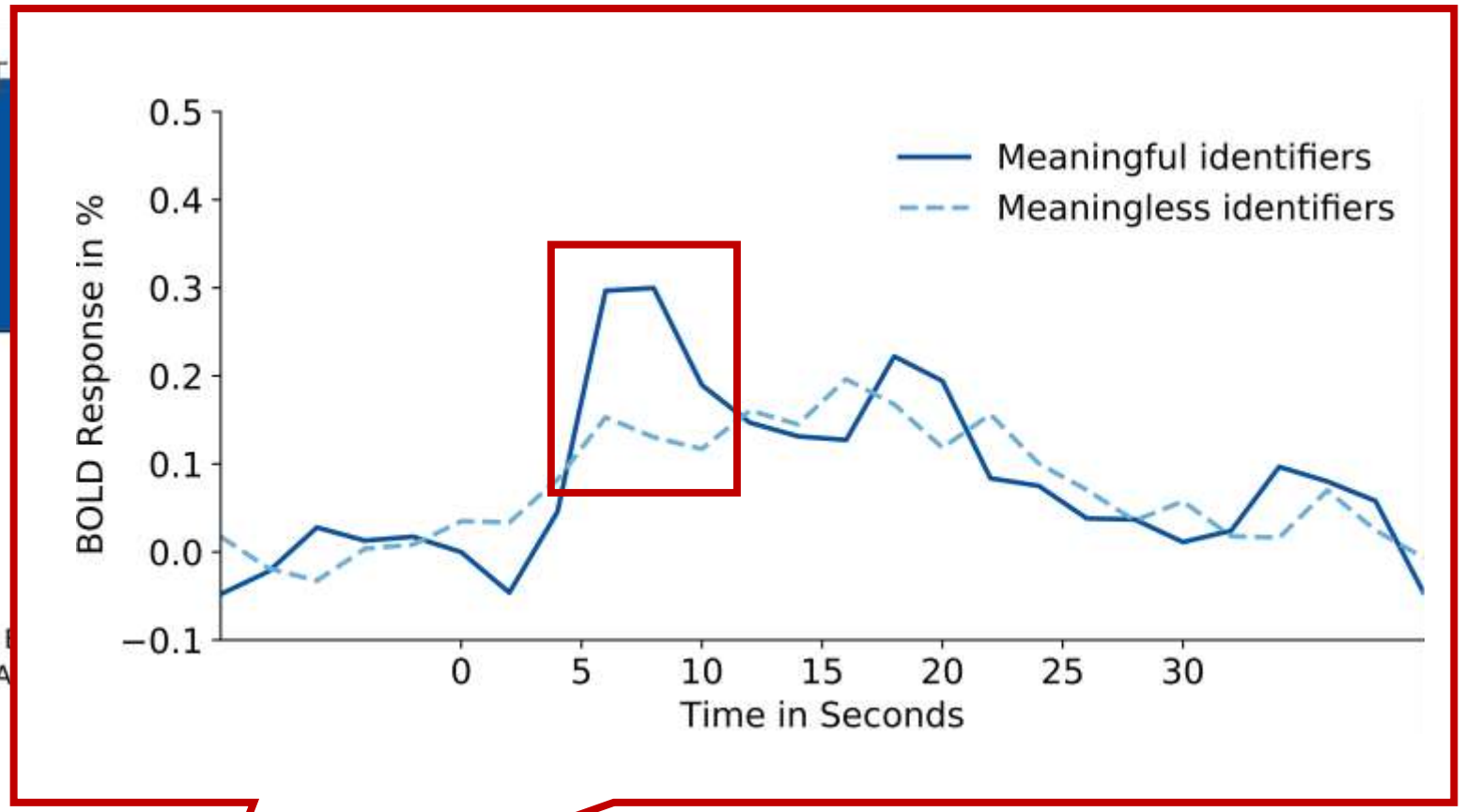
Cognitive Perspective: Results (RQ2)



→ No significant differences?

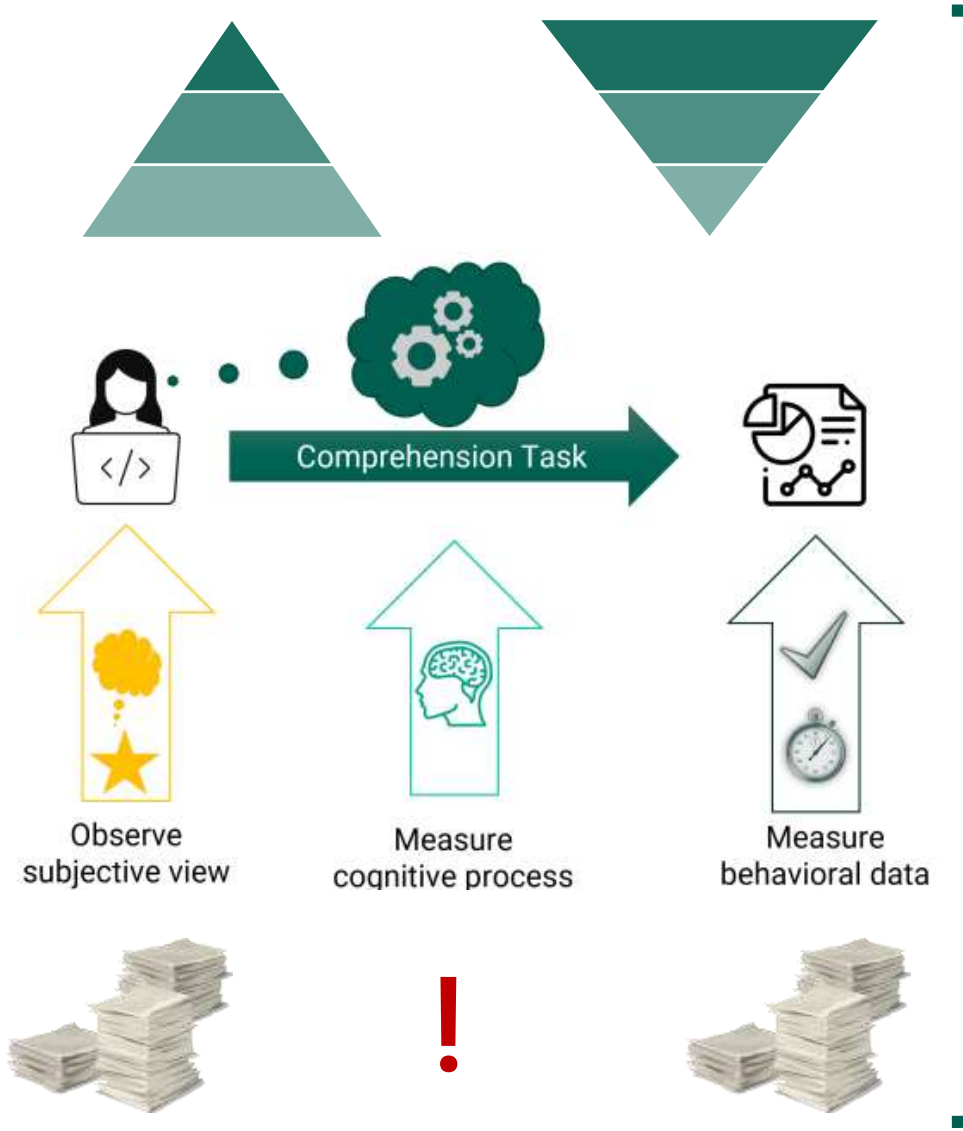
Cognitive Perspective: Results (RQ2)





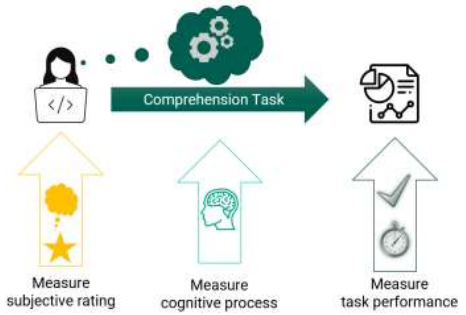
→ No significant differences?

Cognitive Perspective: Conclusion



Theory of Program Comprehension

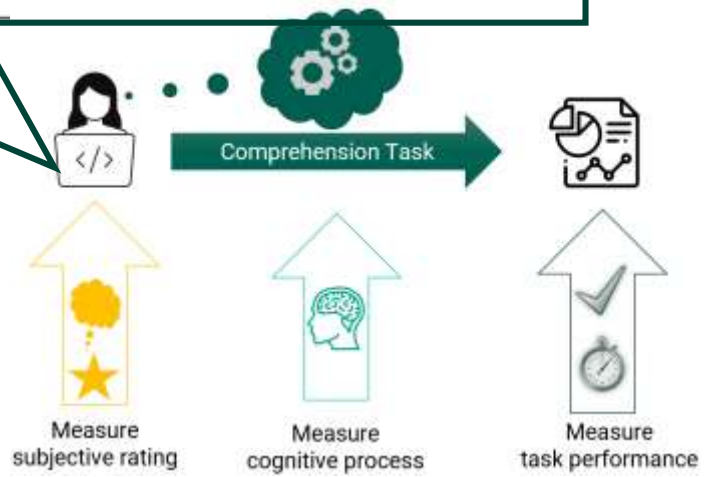
Part III: Applications



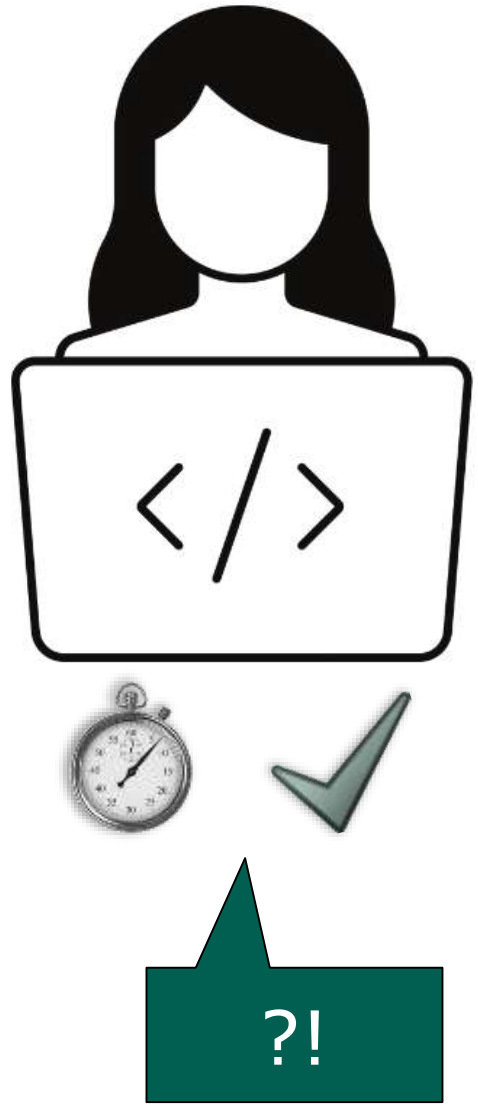
- Framework
  - fMRI & Eye Tracking
  - Multi-Modality
  - Tool Support
- Cognitive Perspective
  - Top-Down Comprehension
  - Reading Order
  - Novice & Experienced Programmers
- **Applications**
  - **Code Complexity Metrics**
  - Role of Aggregation
  - Participant-Specific Brain Parcellation

Code Complexity Metrics: Motivation

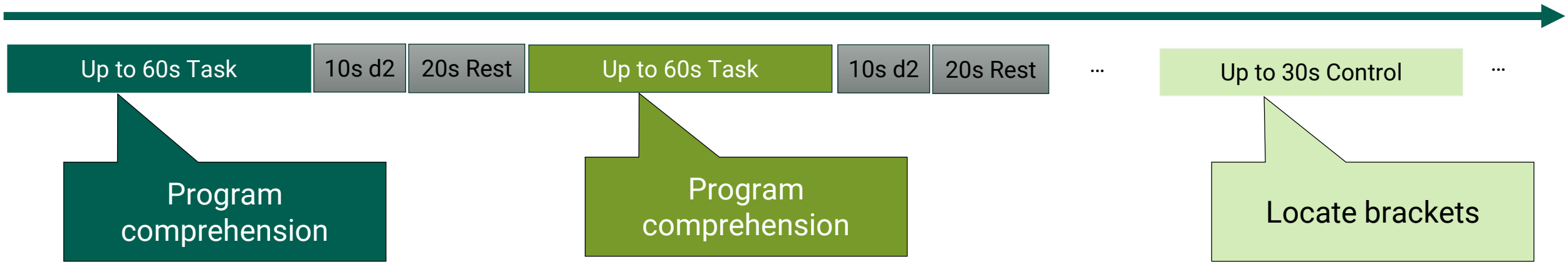
```java
public static void main(){
  String text = "The quick brown fox jumps";
  System.out.print(compute(text));
}

static int compute(String text){
  int result = 0;
  boolean flag = false;
  for (int i = text.length() - 1; i >= 0; i--){
    char c = text.charAt(i);
    if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')){
      flag = true;
      result++;
    } else {
      if (flag)
        break;
      }
  }

  return result;
}
```



- Size (LOC)
- Vocabulary (Halstead)
- Control Flow (McCabe)
- Data Flow (DepDegree)



Code Complexity Metrics: Study Design



Code Complexity Metrics: Study Design

Up to 60s Task | 10s d2 | 20s Re

Program comprehension

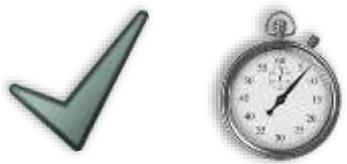
	Snippet	Code Size (LOC)[1]	Vocabulary (Halstead)[1]	Control Flow (McCabe)[1]	Data Flow (DepDegree)[2]
Loop	Average of array	17	12.64	3	17
	Contains substring	26	25.50	7	29
	Count vowels in string	19	13.00	5	22
	Greatest common divisor	24	26.63	5	33
	h index	21	16.25	4	20
	Length of last word	22	18.58	8	17
	Palindrome check	17	16.72	4	17
	Square root of array	23	39.83	5	27
Recursion	Binary to decimal	17	16.75	4	10
	Cross sum	12	14.66	3	4
	Factorial	12	16.50	3	4
	Fibonacci variation	12	10.88	3	4
	Power	17	15.38	4	8
If/Else	Contains yes or no	22	6.13	6	7
	Hurricane check	17	9.00	7	13
	Sort four elements	17	30.30	7	61

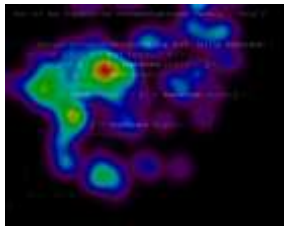
...

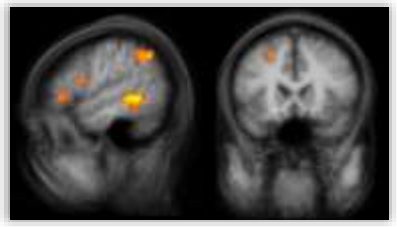
s

Code Complexity Metrics: Modalities

Behavioral Data | Physio Data | Eye Tracking | Brain Activation | Subjective

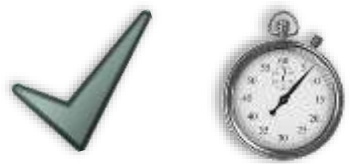


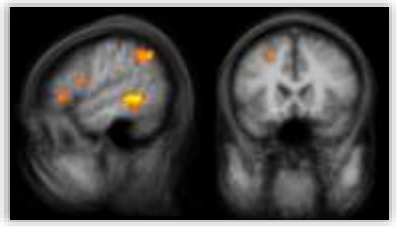




Code Complexity Metrics: Modalities

Behavioral Data | Physio Data | Eye Tracking | Brain Activation | Subjective

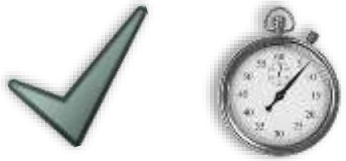




RQ1 RQ2 RQ3

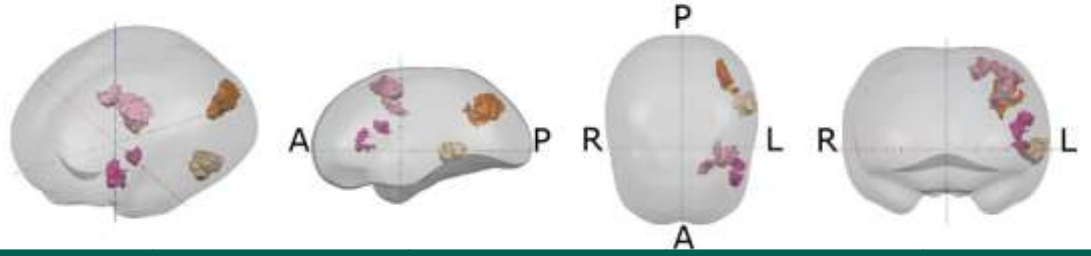
Code Complexity Metrics: Results (1)

Behavioral Data



τ correlation	Correctness	Time
Size	−.46	.22
Vocabulary	−.45	.24
Control Flow	−.09	.06
Data Flow	−.41	.26

Brain Activation



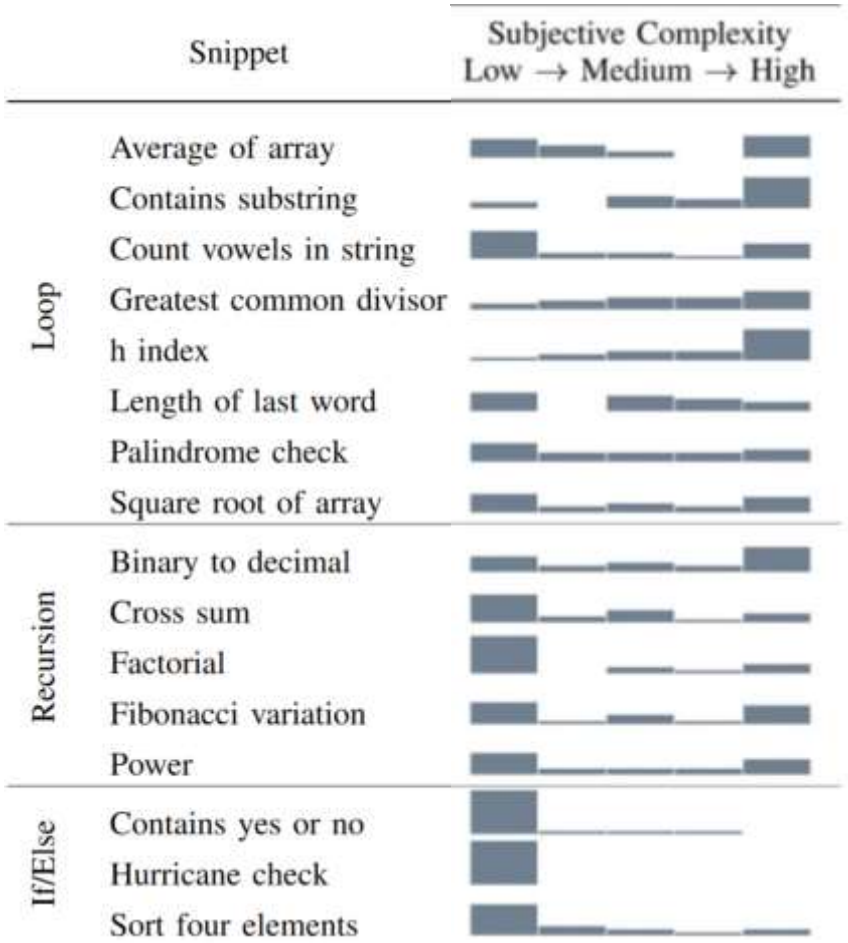
τ correlation	BA 6	BA 21	BA 39	BA 44/45
Size	.26	.43	.17	.15
Vocabulary	.38	.32	.40	.17
Control Flow	.04	.09	.07	−.04
Data Flow	.32	.41	.36	.22

Code Complexity Metrics: Results (2)

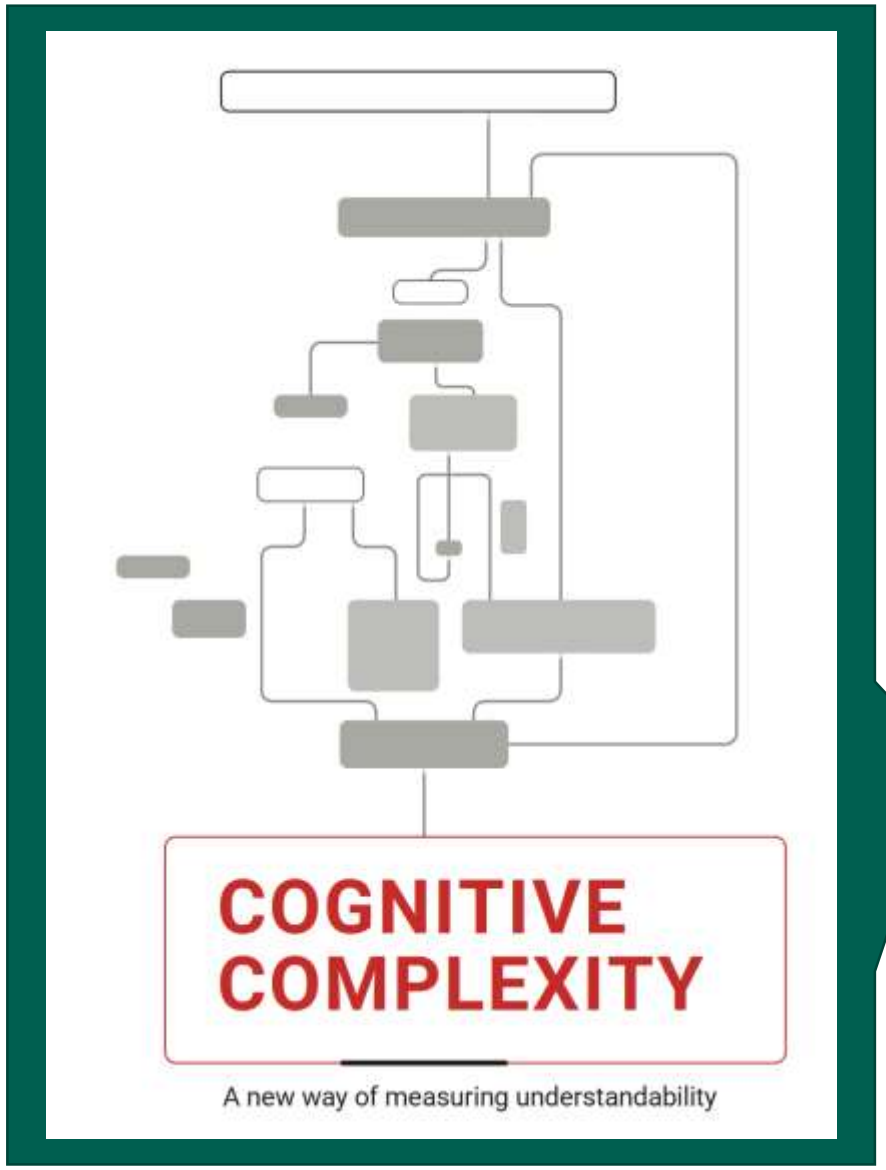
Subjective

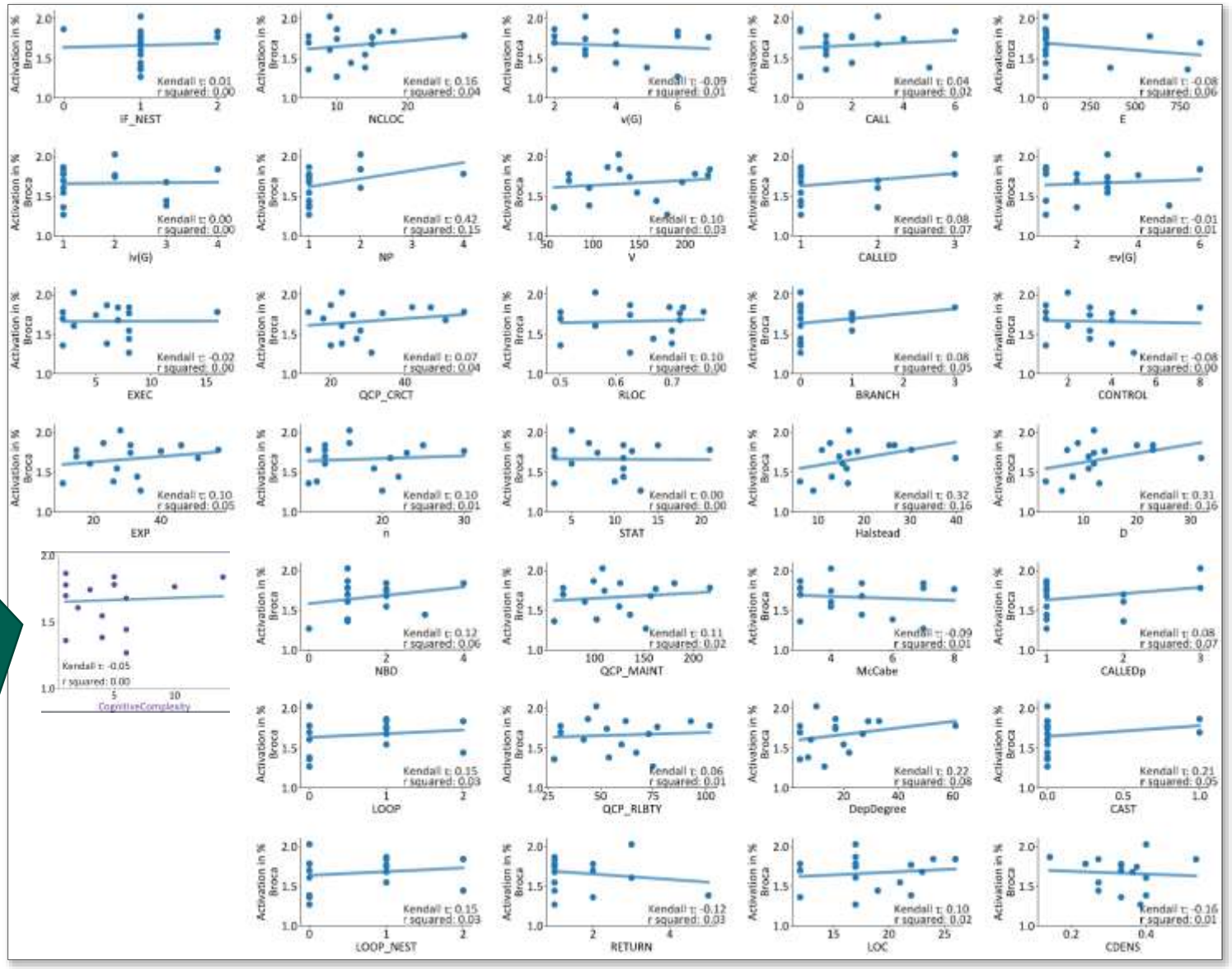
?!

τ correlation	Subjective
Size	.16
Vocabulary	.20
Control Flow	−.07
Data Flow	.16



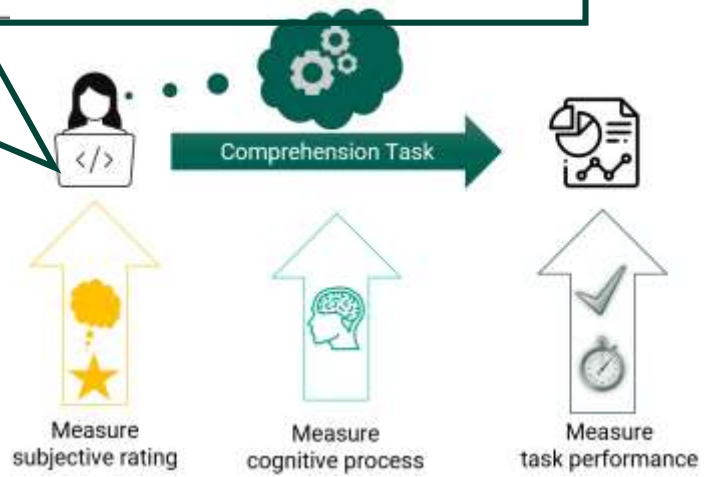
Code Complexity Metrics: But what about *my* metric…?



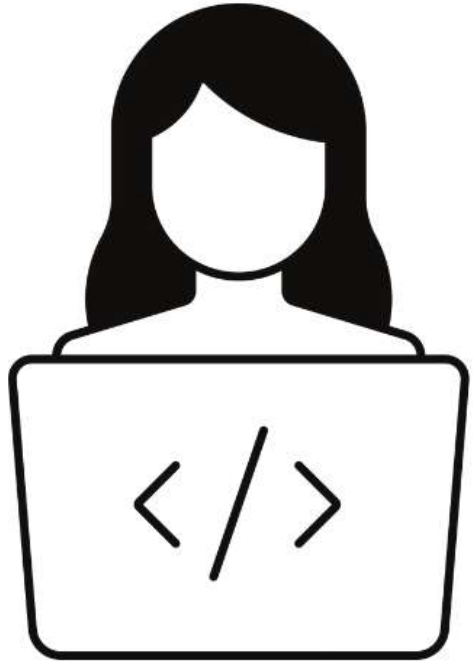


Code Complexity Metrics: Conclusion

```java
public static void main(){
  String text = "The quick brown fox jumps";
  System.out.print(compute(text));
}

static int compute(String text){
  int result = 0;
  boolean flag = false;
  for (int i = text.length() - 1; i >= 0; i--){
    char c = text.charAt(i);
    if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')){
      flag = true;
      result++;
    } else {
      if (flag)
        break;
      }
  }

  return result;
}
```



- Size (LOC)
- Vocabulary (Halstead)
- Control Flow (McCabe)
- Data Flow (DepDegree)

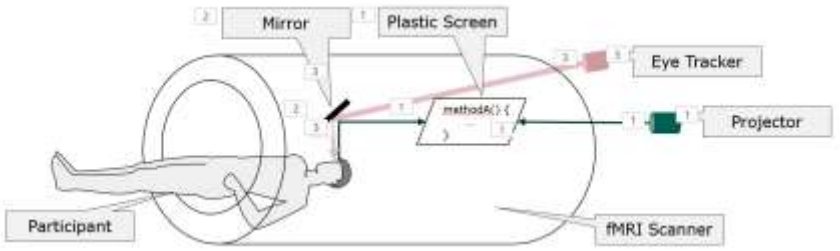


- Only to a limited degree
- No all-encompassing metric
- Individuality of programmers

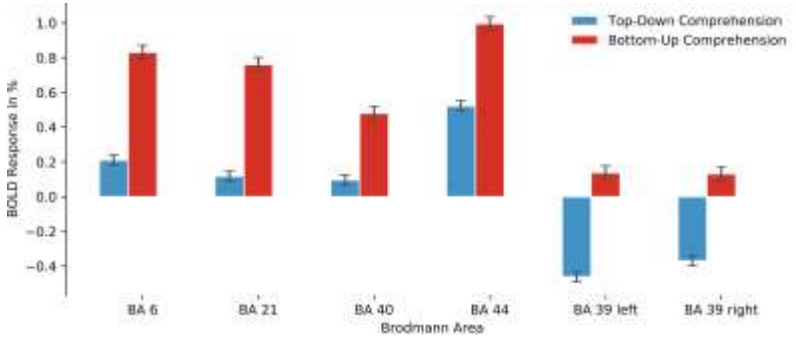
→ Starting point for a more empirical-driven search of code complexity metrics

Dissertation Conclusion

Framework



Cognitive Perspective



Applications