

# The Stress of Improvisation: Instructors' Perspectives on Live Coding in Programming Classes

Xiaotian Su, April Yi Wang

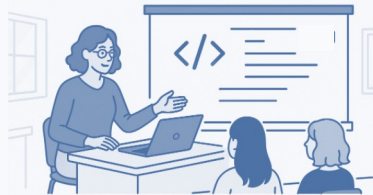
xiaotian.su@inf.ethz.ch, april.wang@inf.ethz.ch

Department of Computer Science, ETH Zurich



## 1 Introduction

Live coding is the process of designing and implementing a coding project in front of the class during lecture period. While live coding offers numerous benefits to students, there is little concrete evidence for how **instructors** perceive the benefits of live coding and the cognitive load it imposes on them.



## 2 Research Questions

**RQ1:** What **motivates** instructors to use live coding in their teaching?

**RQ2:** How do instructors **approach and conduct** live coding in the classroom?

**RQ3:** What **obstacles or barriers** do instructors encounter when teaching programming through live coding?

## 3 Methodology



### Semi-Structured Interviews

Interviews with five teaching assistants, focusing on their experiences of live coding in small exercise session



### Contextual Inquiry

Observations of four lecturers conducting live coding during large-scale computer science lectures, spanning a variety of topics.

PID	Age Range	Gender	Profession	Teaching (# years)	Class size (# students)	Students		
						H	U	C
P1	20-25	F	BSc Student	One	20-30	✓	✓	
P2	20-25	M	MSc Student, Part-Time Lecturer	Three	20-30	✓	✓	
P3	26-30	M	PhD Student	Five	20-30		✓	
P4	20-25	M	PhD Student	Five	30-50		✓	✓
P5	26-30	M	PhD Student	Six	20-30	✓	✓	
P6	30-35	M	Full-Time Lecturer	Five	300-400		✓	
P7	40-45	M	Full-Time Lecturer	Five	50-150		✓	✓
P8	45-50	M	Full-Time Lecturer	Eight	100-150		✓	✓

## 4 Results

### RQ1: Motivations

- **Improve Comprehension and Pacing:** Live coding helps instructors slow down the lecture speed.
- **Adaptivity:** Live coding allows instructors to address student queries and adapt teaching to student needs in real-time.
- **Engagement:** Live coding enables students to engage more actively than traditional classrooms.
- **Instilling Good Programming Practices.** Live coding improved students' adherence to coding conventions and best practices.

*When I show students that I transform very unreadable code into readable code, I notice that in the next submission, they follow these practices.*

-- P2



### RQ2: Implementations

- **Preparation:** Instructors for different levels have different strategies.
- **Procedure:** Live coding in lectures are restricted while in exercises are more flexible.
- **Students' Activities:** Students found it difficult to call out the next line of code when instructors sought input.

*I wouldn't have a fixed set of commands ready; I would put myself in the situation of the students that they would be starting from a blank state.*

-- P5



### RQ3: Obstacles

- **Decline of Engagement:** Students lose interests if the session is too long.
- **Unpredictability:** Live demonstrations may deviate from the plan, occasionally resulting in errors or unexpected outcomes.
- **Mental Stress.** Novice instructors reported more mental stress than experienced instructors.
- **Time Pressure.** Instructors often struggle to keep time due to the improvisational and unpredictable nature of live coding.

*When the code gets complex, I tend to focus too much on the debugging, and I lost the audience.*

-- P2

