Towards an IDE to Support Programming as Problem-Solving

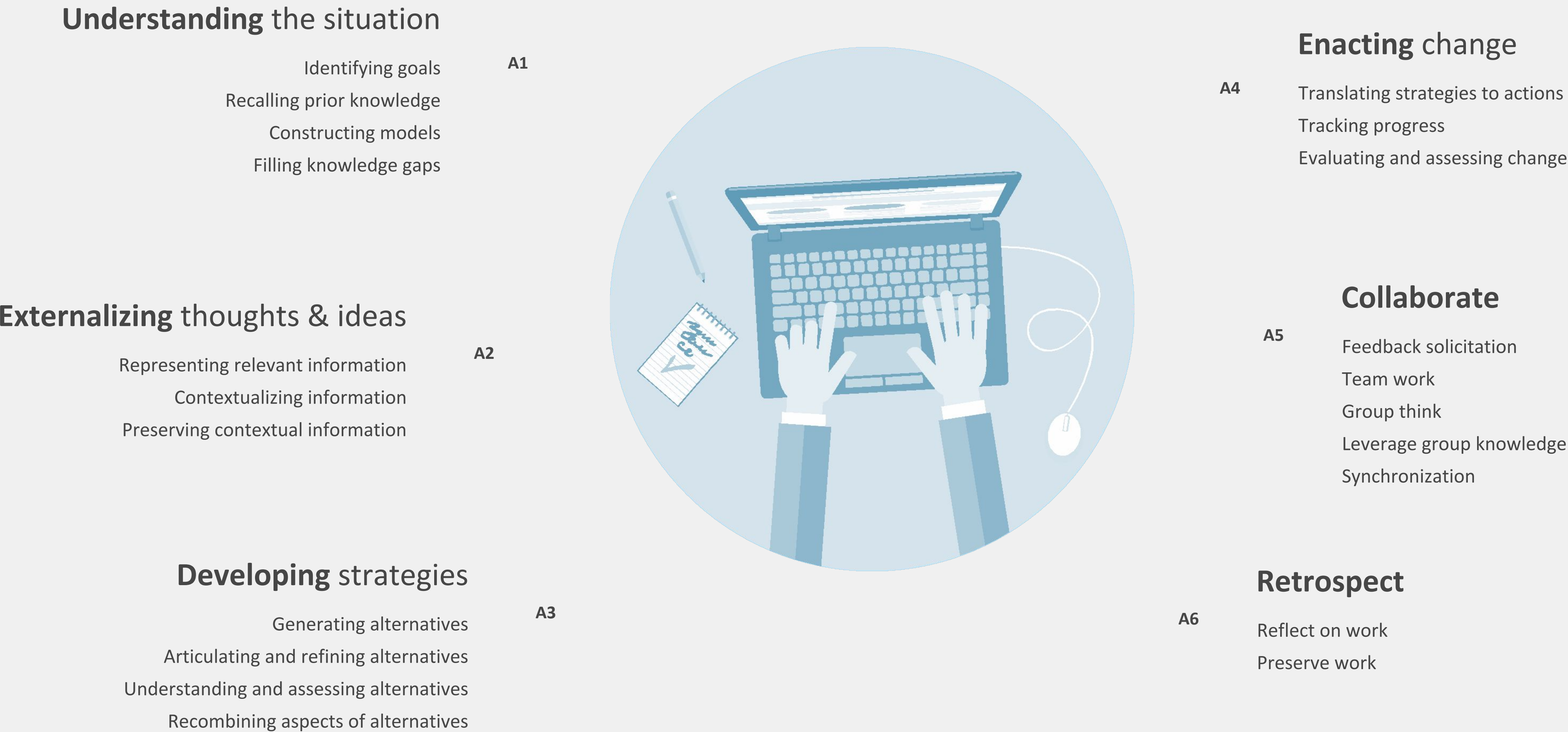Nicholas Nelson   Anita Sarma   André van der Hoek

Oregon State University   Oregon State University   THE UNIVERSITY OF CALIFORNIA · IRVINE

Programming is more than dealing with language syntax and semantics: it is inherently an exercise in problem-solving that extends beyond the act of editing code.

Activities and Actions of Programming as Problem-Solving

Understanding the situation

Identifying goals **A1**
Recalling prior knowledge
Constructing models
Filling knowledge gaps

Externalizing thoughts & ideas

Representing relevant information **A2**
Contextualizing information
Preserving contextual information

Developing strategies

Generating alternatives **A3**
Articulating and refining alternatives
Understanding and assessing alternatives
Recombining aspects of alternatives

Enacting change

A4
Translating strategies to actions
Tracking progress
Evaluating and assessing change

Collaborate

A5
Feedback solicitation
Team work
Group think
Leverage group knowledge
Synchronization

Retrospect

A6
Reflect on work
Preserve work

Understanding the situation

A1

Challenges:
How to support programmers' formulation of problems and reflection on potential solutions?
How to support programmers in relying upon past experience?

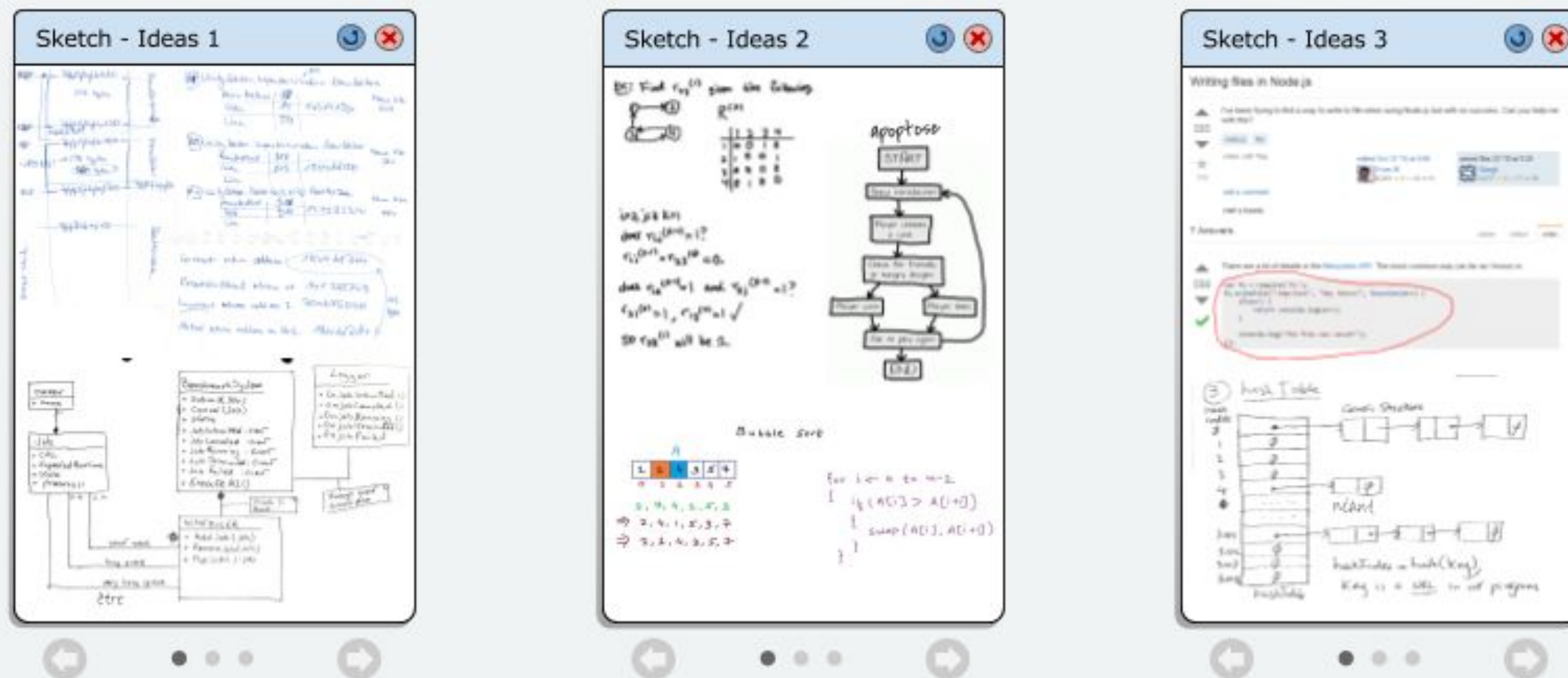**Externaling** thoughts & ideas

A2

Challenges:
How to support programmers in relying upon past experience?
How to utilize different pieces of information and context to support the act of coding?
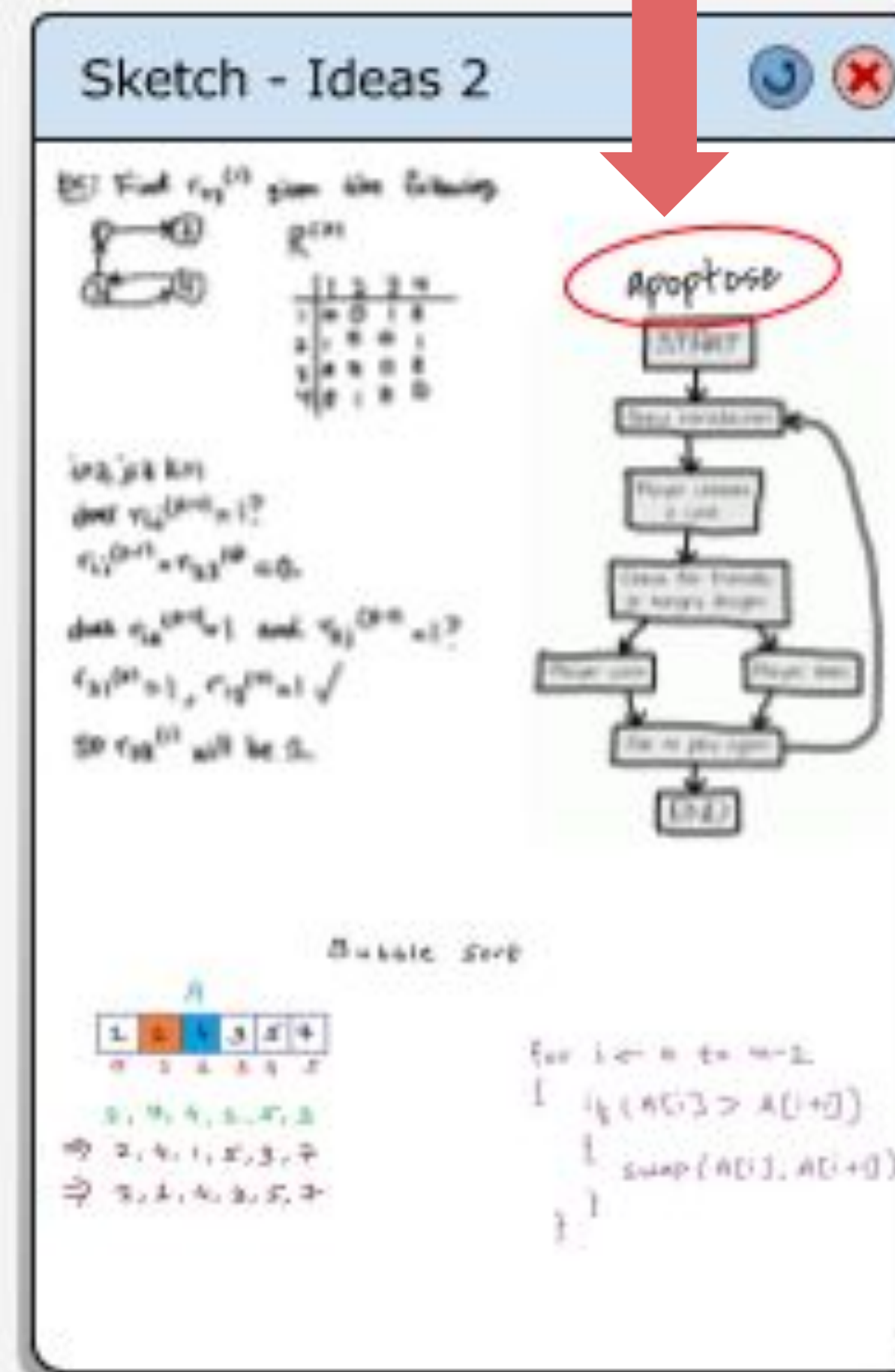
Developing strategies

A3

Challenges:
How to support different information processing styles and workflows of programmers?
How to utilize different pieces of information and context to support the act of coding?
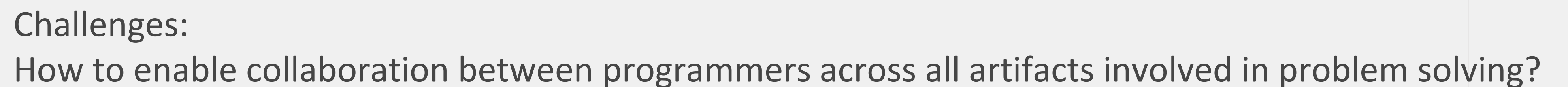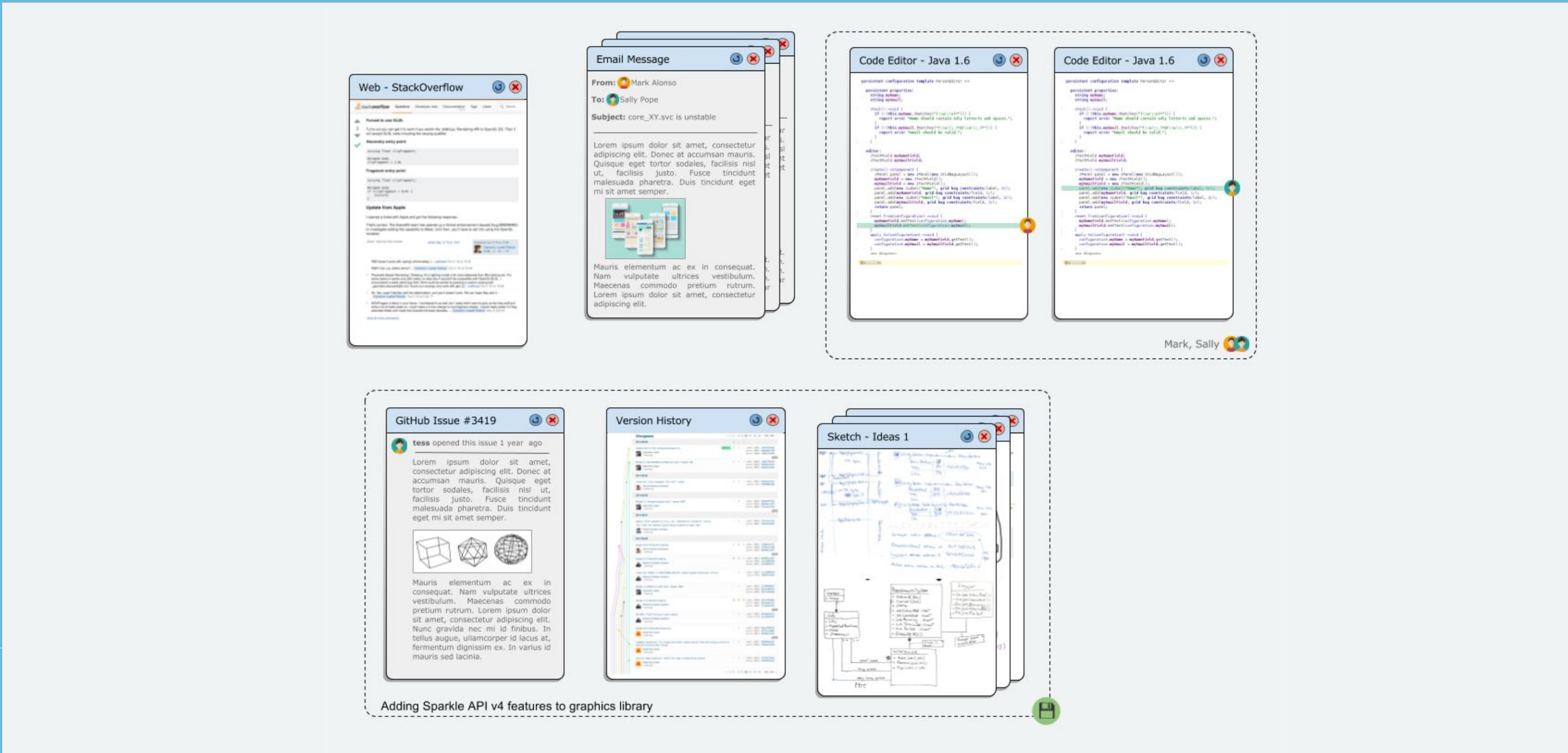
Enacting change

A4

Challenges:
How to utilize different pieces of information and context to support the act of coding?

Collaborate

A5

Challenges:
How to enable collaboration between programmers across all artifacts involved in problem solving?

Retrospect

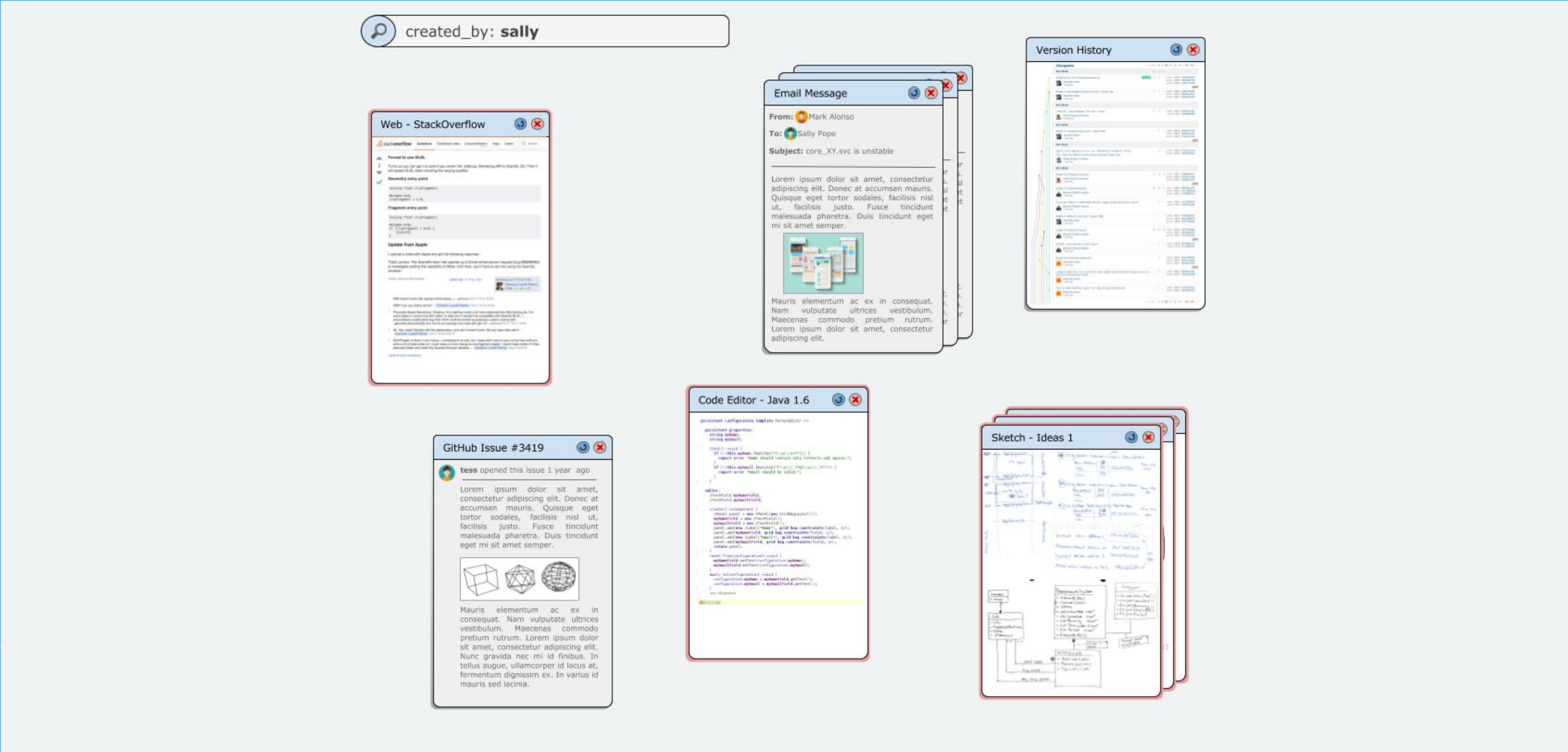**A6**

Challenges:
How to support programmers in relying upon past experience?
How to provide programmers access to the relevant context in a problem space?
How to support different information processing styles and workflows of programmers?

Retrospect

A6

Challenges:
How to support programmers in relying upon past experience?
How to provide programmers access to the relevant context in a problem space?

Challenges to introducing problem solving in programming to IDE design

Challenges

1   How to support programmers' formulation of problems and reflection on potential solutions?

2   How to provide programmers access to the relevant context in a problem space?

3   How to support different information processing styles and workflows of programmers?

4   How to support programmers in relying on past experience?

5   How to enable collaboration between programmers across all artifacts involved in problem solving?

6   How to utilize different pieces of information and context to support the act of coding?

Q

Programming as Problem Solving

Q1 — Does the open-ended and free-form interaction paradigm make sense?

Q2 — Are cards the best metaphor for problem solving?

Q3 — Do cards with information on many faces help or hinder?

Q4 — Do cards with different media types make sense?