Towards an IDE to Support Programming as Problem-Solving

Nicholas Nelson
Anita Sarma
André van der Hoek
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
- Recalling prior knowledge
- Constructing models
- Filling knowledge gaps

**Externalizing thoughts & ideas**
- Representing relevant information
- Contextualizing information
- Preserving contextual information

**Developing strategies**
- Generating alternatives
- Articulating and refining alternatives
- Understanding and assessing alternatives
- Recombining aspects of alternatives

**Enacting change**
- Translating strategies to actions
- Tracking progress
- Evaluating and assessing change

**Collaborate**
- Feedback solicitation
- Team work
- Group think
- Leverage group knowledge
- Synchronization

**Retrospect**
- Reflect on work
- Preserve work

Section 1
Programming as Problem-Solving
Understanding the situation

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

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

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

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

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?
Questions for the audience.

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?