IDE Support For EXAMPLE EMBEDDING

Ohad Barzilay, Blavatnik School of Computer Science, Tel-Aviv University
Amiram Yehudai, Blavatnik School of Computer Science, Tel-Aviv University
Orit Hazzan, Department of Education in Technology and Science, Technion

May 2009

Development of Java Based Software Systems

Agenda

- Introduce **software activity** perspective
- Characterize **Example Embedding** (EE) as a software activity
- **IDE support** for Example Embedding from the software activity perspective

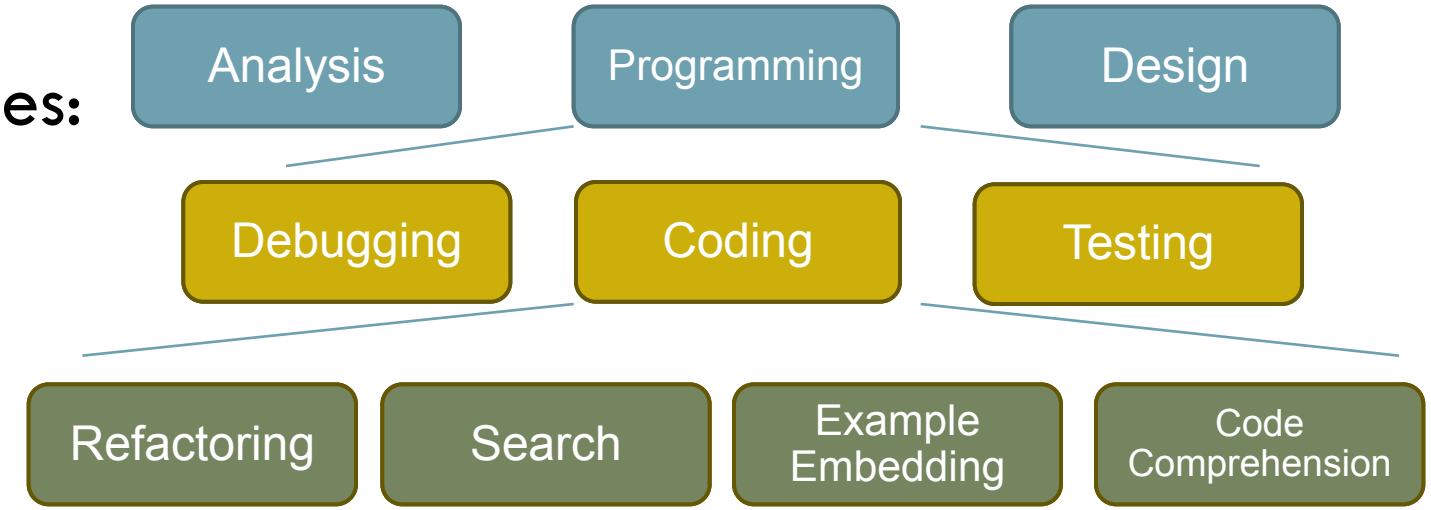
Contribution

- We define a **methodology** and a **conceptual framework** by which **empirical software engineering research** can be applied to identify and characterize software activities

- We identify and characterize a new software development activity called **Example Embedding (EE)** using the suggested methodology [work in progress]

Software Activity

- **Software Activity** is a collection of fine grained techniques which together assemble an abstract key notion in software development
- Software activity may be composed of finer grained activities
- Examples:



Motivation: Refactoring

- **Refactoring** is a disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior [www.refactoring.com/]

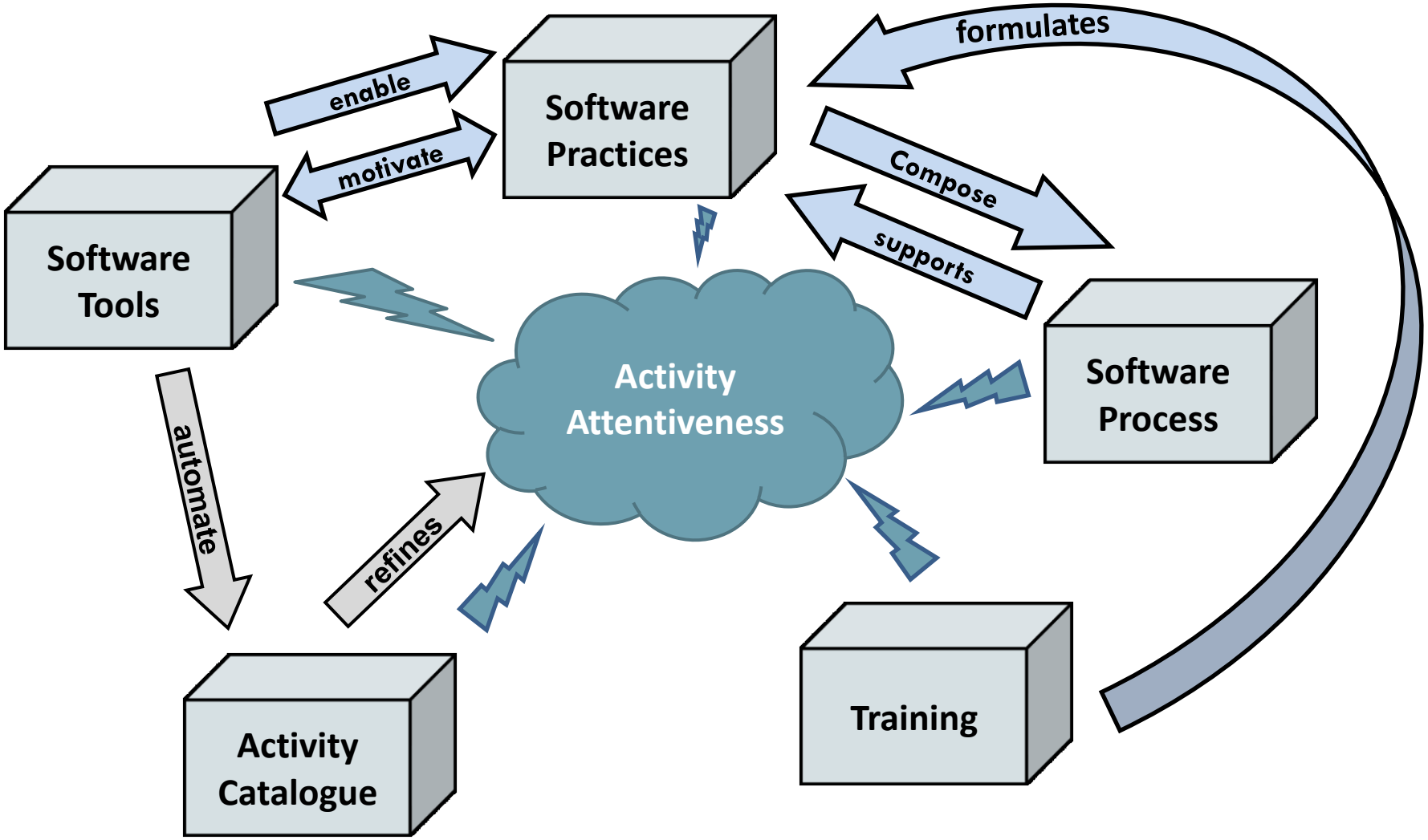
- Although refactoring code has been performed informally for years, William Opdyke's 1993 Ph.D. dissertation is the first known resource to specifically examine refactoring

Characterizing **Refactoring** as a **Software Activity**

- **Naming** the activity
- Laid the foundations for others to build a **catalogue**
- Enabled the development of **software tools**
- Promoted new **coding practices**
- Influenced the **development process**

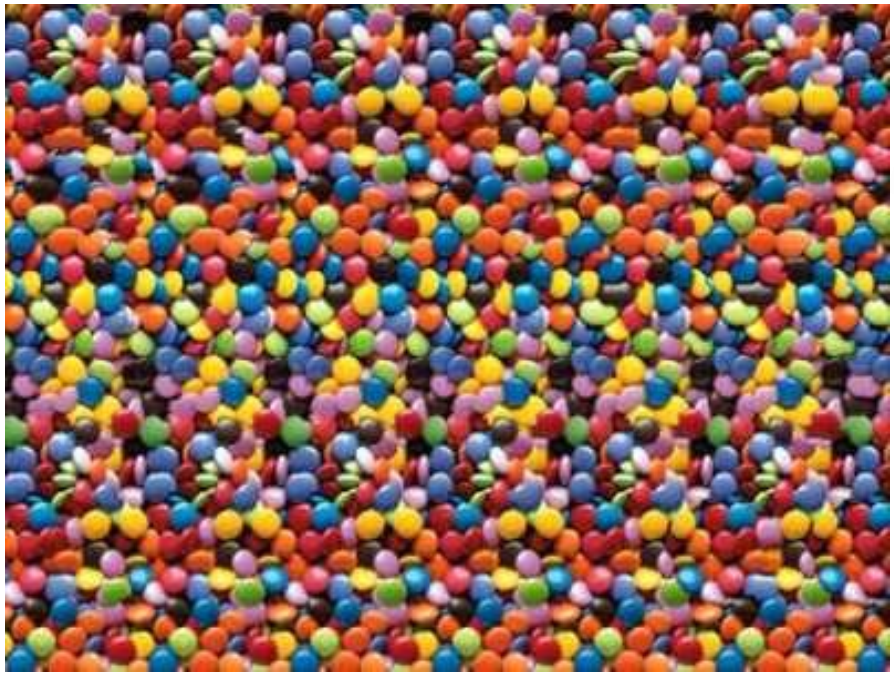
Identifying software activity as such, assists in creating an ecosystem that would exploit it and eliminate its pitfalls and deficiencies

Software Activity Ecosystem



Finding the "Next Refactoring": Methodology

- We take an **empirically driven** research methodology
  - The activities already exist – they just need to be identified
- Identifying and characterizing software activity deals with **human behavior** hence requires corresponding methodology
  - Both for **gathering the data** and **analyzing** it
- We take a **Qualitative research** approach:
  - **Holistic** and **context aware**
  - **Why** and **how** of **decision making**, not just what, where, and when
  - Smaller but focused samples rather than large random samples
  - **Categorizes** data into patterns as the primary basis for organizing and reporting results



Research outline

Goal: Extracting activities from "coding"

Observations

Guidance for reading code?

How do programmers understand code?

Secondary tools for Data gathering

ata Analysis

Literature

efining research questions

Building a field grounded theory

I'M ANALYZING SOME PIECE OF CODE I'VE FOUND
I LOVE TO DO THIS. CRAWLING THROUGH SOME VERY OLD CODE AND ASKING MYSELF:
- WHO WROTE IT?
- WHEN DID HE WRITE IT?
- WHAT DID HE THINK THEN?
- ME
- LAST WEEK
I HAVE NO IDEA
geek and poke

Example Embedding (EE)

- **Example Embedding (EE)** is the process of **finding**, **altering** and **embedding** an already existing code fragment (the example) within a new context

- To address the challenges of EE building blocks, one should address:
  - **Name** and **definition**
  - Building a **catalogue**
  - Software **tools**
  - Coding **practice**
  - The development **process**
  - **Training**

- Further, we elaborate on some these [work in progress]

Building a Catalogue

- Some programmers use examples only in certain contexts, but not on others
- **Classifying** when/where/how to use examples would raise the level of awareness for using examples and highlight its subtleties
- Should be based on both empirical data and further synthesis
- Some examples:
  - **Example for what**. programming languages, scripting languages, libraries and APIs and configurations.
  - **Example purpose**. adding functionality, fixing a syntax error, applying a design pattern, bootstrapping with a new environment ("hello world").
  - **Example size**. Examples are different in size, scope and complexity: from several characters demonstrating a language operator, through function calls, and complex operations requiring a sequence of several method invocations involving several types.
  - **Examples source.** From the organization code base, documentation, example set which is provided by the company, web tutorials, blogs, emails and more.
  - **Searching for example**. Google search, code search, code browsing, asking people, and documentation search; in several occasions, they know upfront where to look.
  - **Using the example**. Copy and paste, retyping the example, refactoring of the example code and then call it.

Software Practice

- Software practice is a set of activities, applied systematically following some rationale to perform a software related task
- As opposed to software activity, which is declarative in nature, software practice is a meaningful imperative unit on the abstraction level of a development process

- A manifestation of **EE enabled practice** requires further investigation, but one could presume that it would comprise a **testing** phase prior to the embedding and **refactoring** of both the example and the system

Example-Aware Development Process

- Promoting EE enabled practices and taking organizational means to support them
- Adding **example writing** to the development process
- Building and using **example repositories:**
  - Synthetic examples vs. production code
  - Public repositories vs. proprietary
  - Browsable vs. searchable vs. structured
  - General purpose vs. ad-hoc example collections
  - Documentation and examples

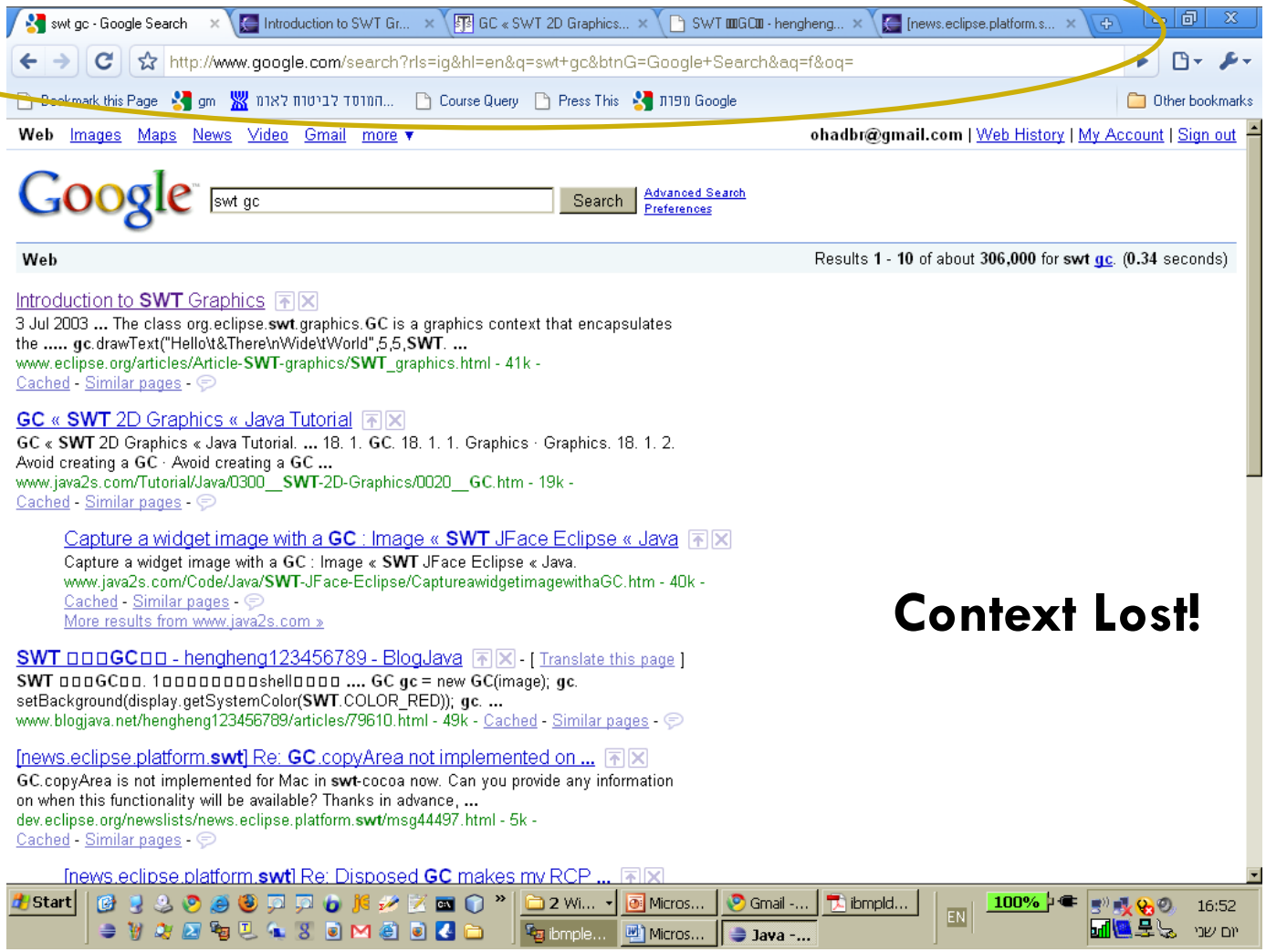
Human Perspective of EE Tool Support

- Already existing IDE plug-ins use examples in various ways:
  - **EG** [Edwards, 2004]
  - **Code Conjurer** [Hummel *et al*, 2008]

- Would programmer use these?
  - A programmer doesn't want to **teach**/**help**/**assist** the computer
  - She knows Java, hence she would like to **write Java code** and not some query language
  - She could spot a solution when she sees one
  - She could use assistance in the activity she is doing no matter what

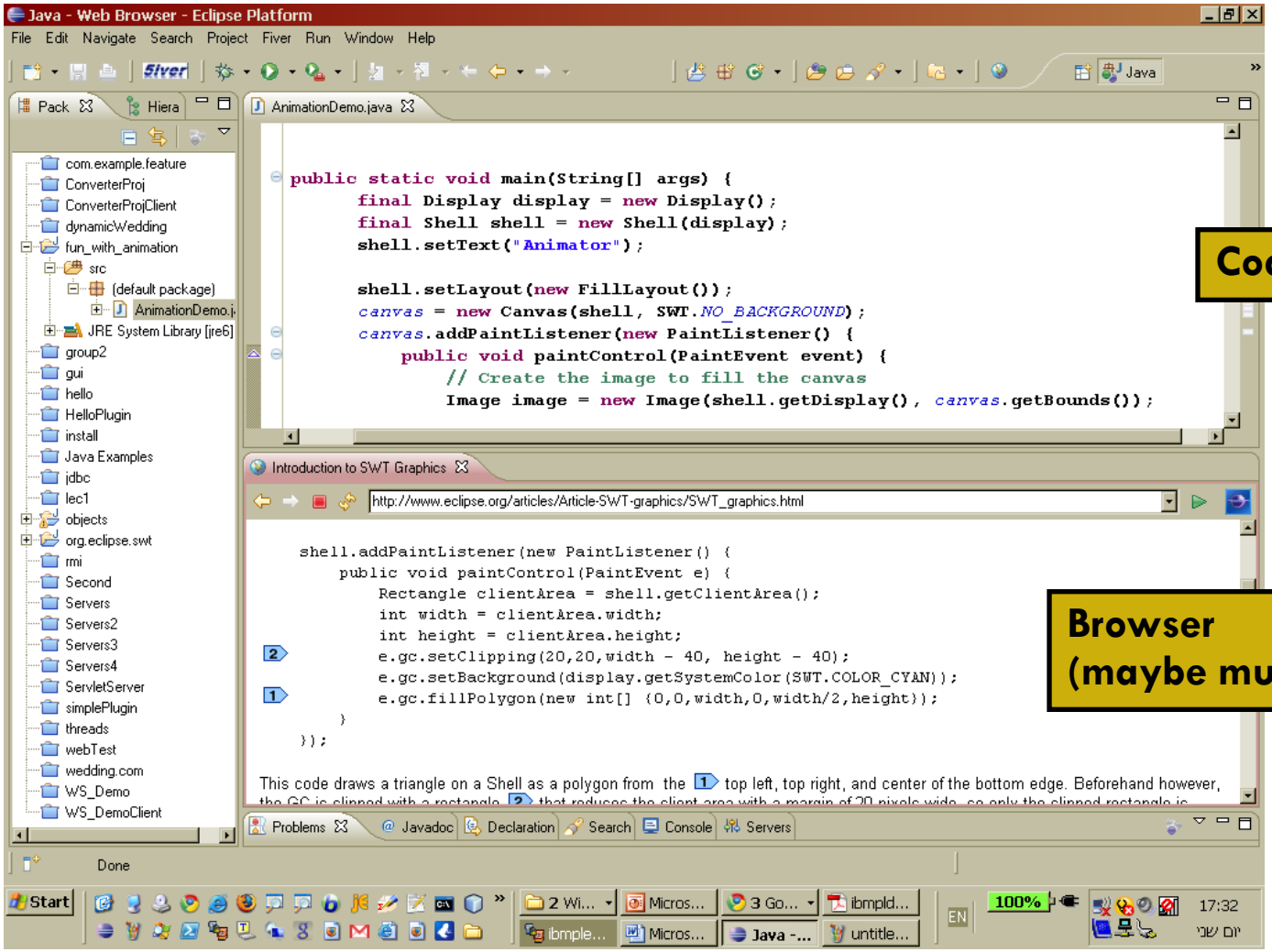
- A human perspective of EE tools
  - Favoring good example repositories combined with good but simple indexing/searching tools
  - **Automate repetitive tasks** in finding, altering and embedding examples
  - **Framework to streamline** EE building blocks (find, alter and embed)

IDE – Browser Integration

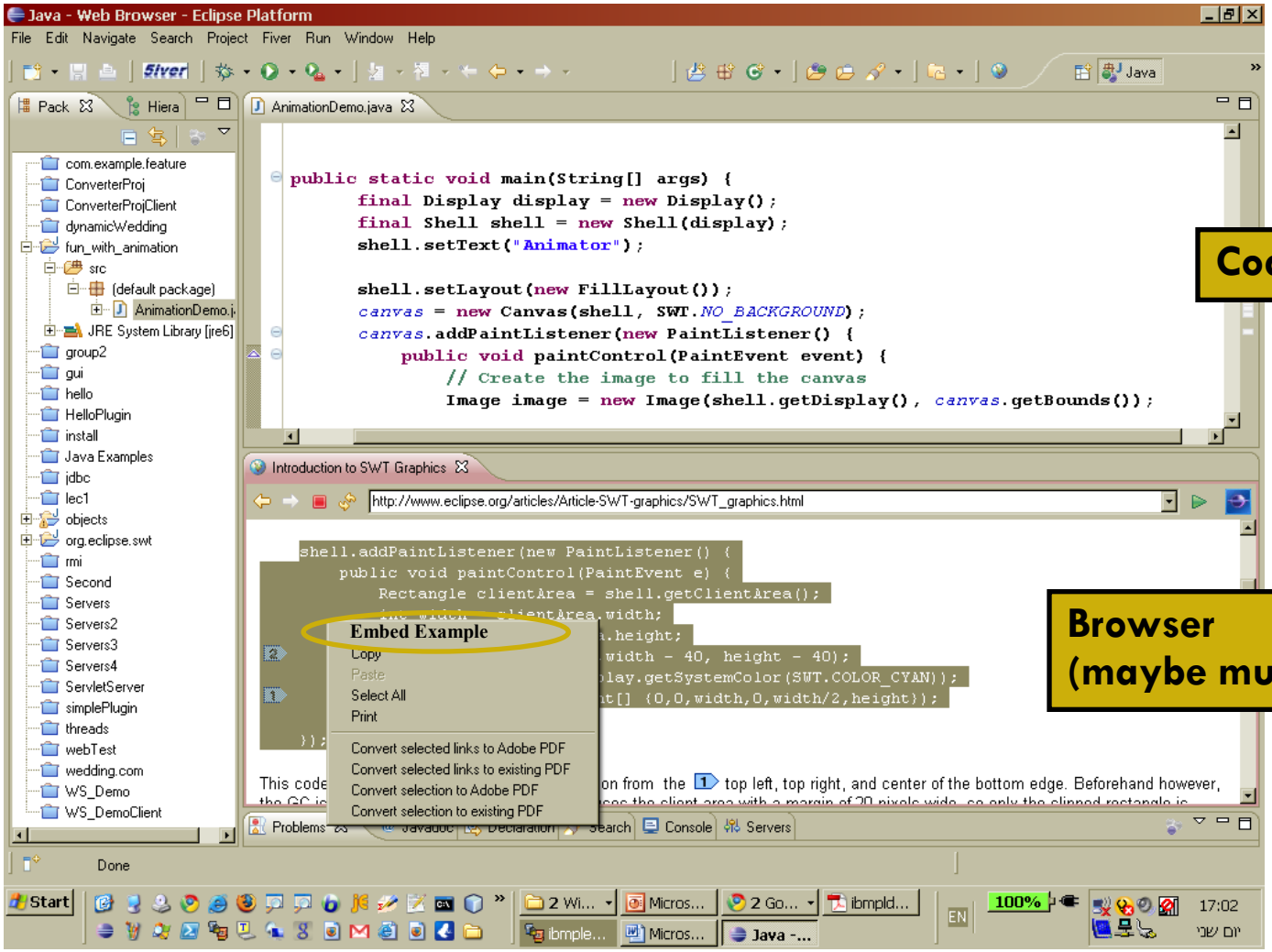


Context Lost!

We wish we had **IDE – Browser** Integration with **Extract Example** support



We wish we had **IDE – Browser** Integration with **Extract Example** support



Summary

- We introduced the **software activity** perspective and explained its benefits using **Refactoring** and **Example Embedding**

- We motivated considering **human aspects** in software engineering research in general and **empirical** and **qualitative research methodology** in particular