



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Software Development in the Cloud Era

Mira Mezini

a Symbiotic Relationship...



Programming Concepts for the Cloud

Cloud for Software Development



European Research Council

Established by the European Commission

**Supporting top researchers
from anywhere in the world**




Programming Concepts for the Cloud


The **PACE** Project

Cloud for Software Development

Sunny clouds ...



“Sensors, devices provide quantitative measurements at never-before-seen resolutions ... social media networks can reveal real-time trends in group mood and behavior”



“Buried within the flood of information are keys to solving huge societal problems and answering big questions of science”

Sunny clouds ...



Dark clouds ...



Privacy, transparency and control:
How well can we control our environment?

Reliability and dependability: what happens, if something goes wrong?
„Will BMW and Mercedes fail on high tech?“
“Will my Wifi turn on, when the neighbor turns on his espresso machine?”

Dark clouds ...

Privacy, transparency and control:
How well can we control our environment?

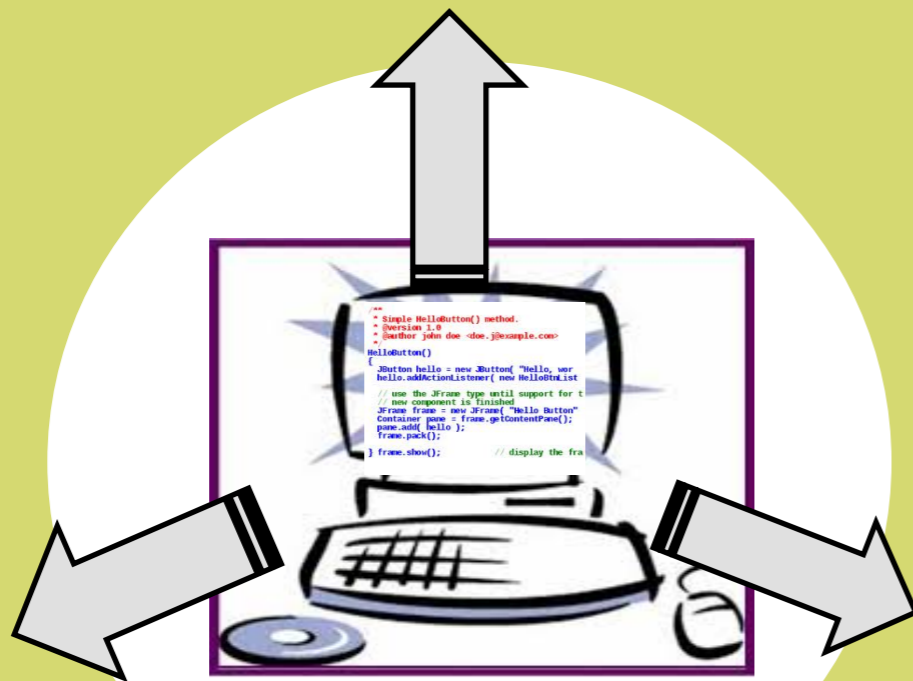
Reliability and dependability: what happens, if something goes wrong?
„Will BMW and Mercedes fail on high tech?“
“Will my Wifi turn on, when the neighbor turns on his espresso machine?”

Complexity Ceiling

Jaron Lanier: “World as a planet of the help desks in which human race will be largely engaged in maintaining very large software systems ...”



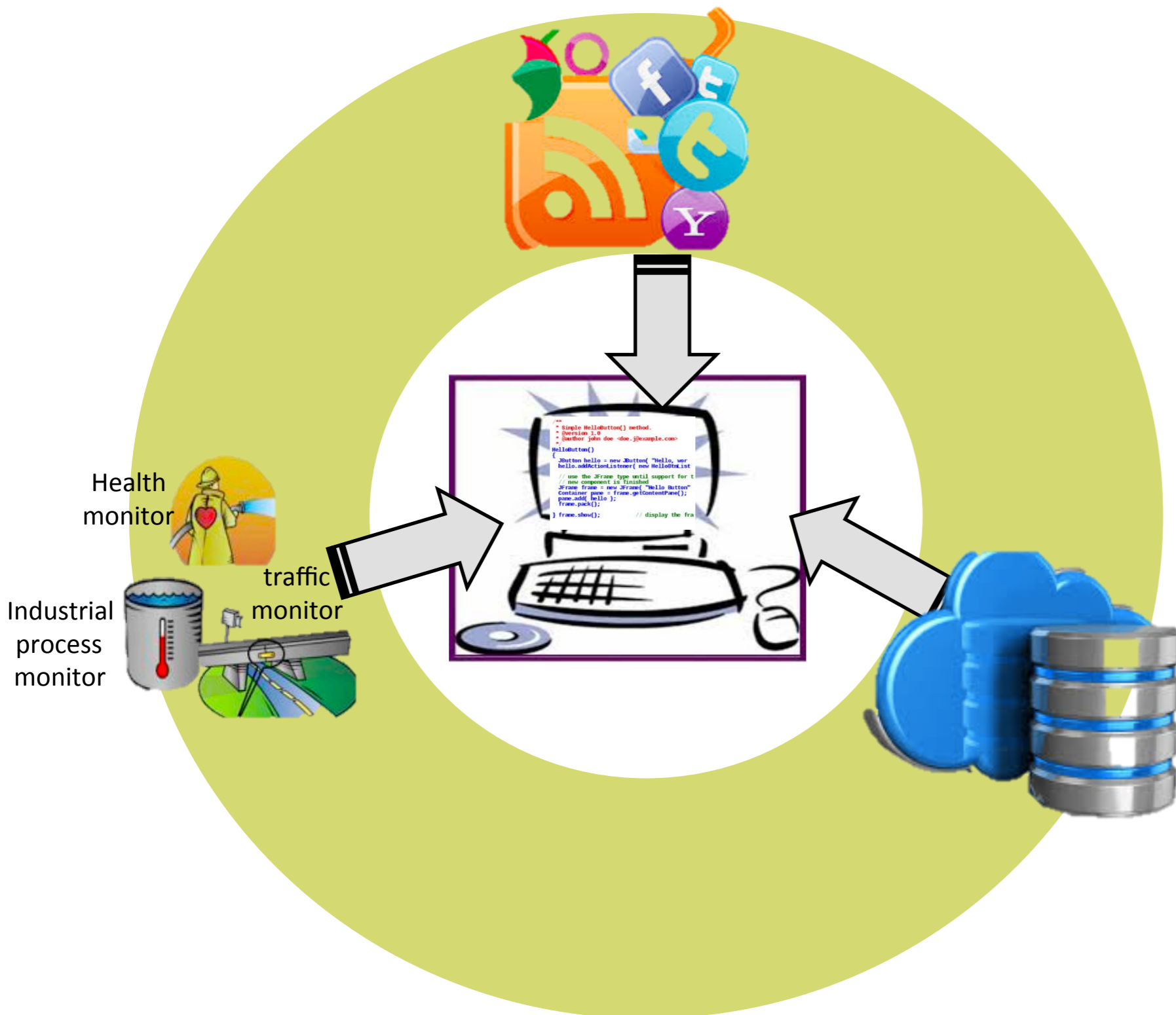
We've learned to manage this complexity...



Program controls the environment

Mechanisms for abstraction and composition geared towards single data and control units

But, the new world is different ...

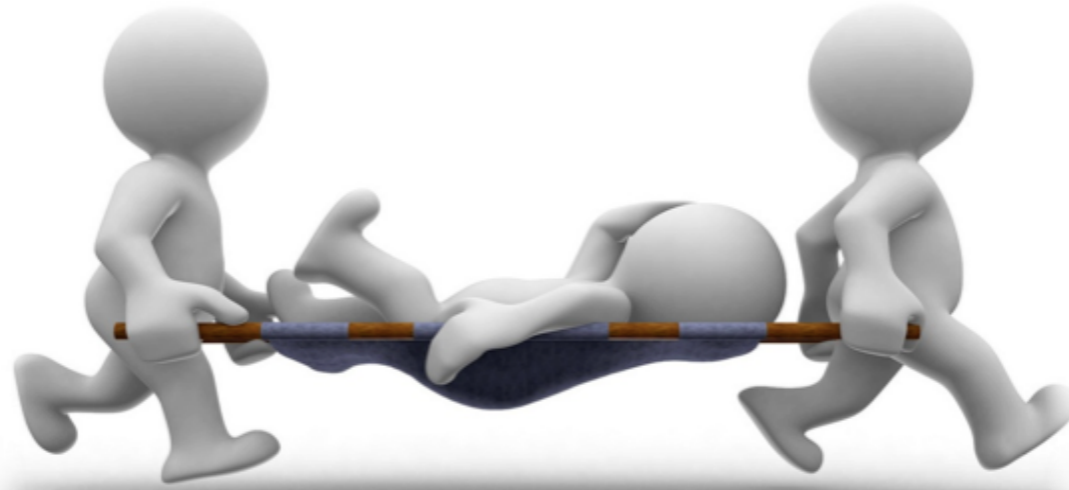


Environment has the control

Different kinds of inputs: Events, data

Aggregations of data/ events are interesting units of abstraction and composition

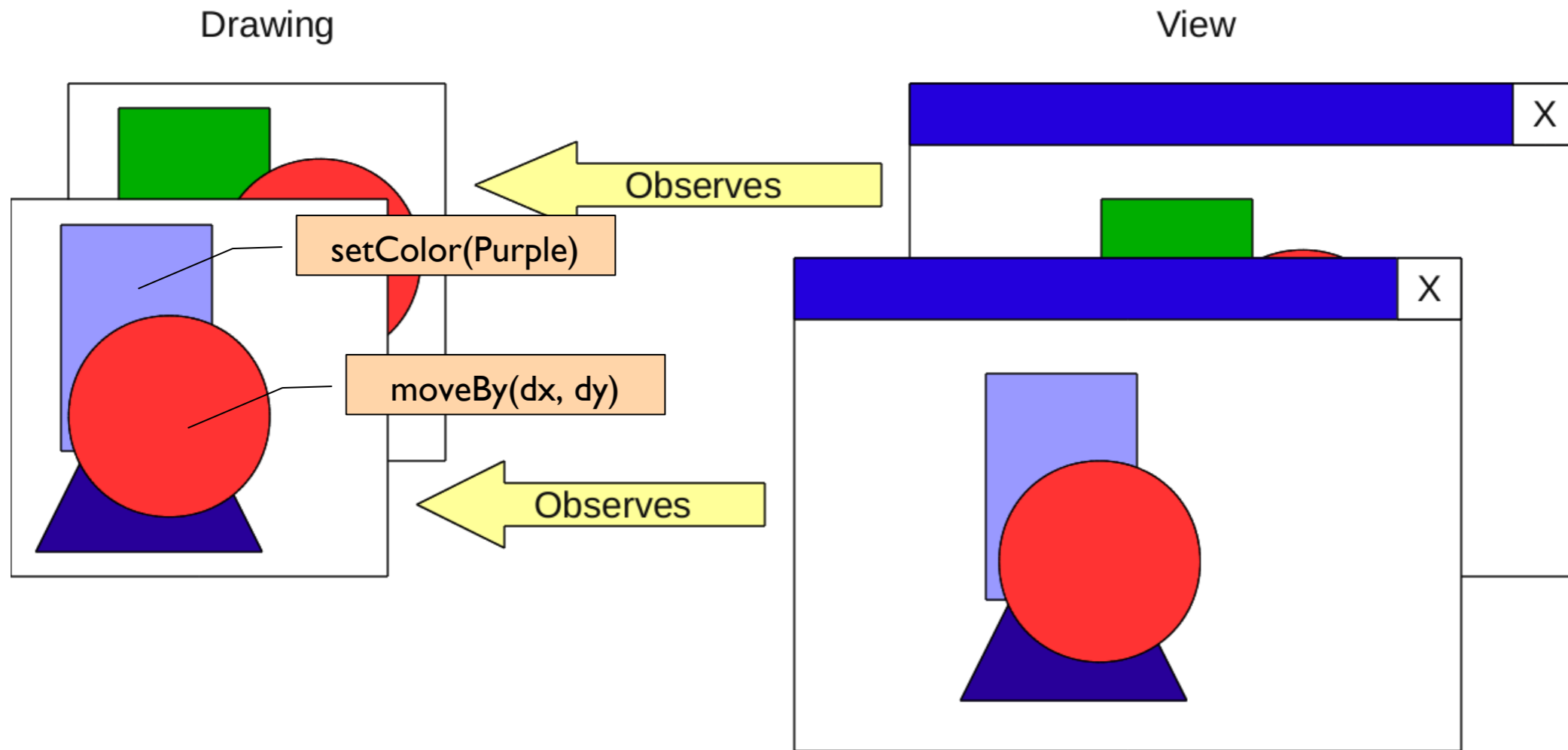
PACE is coming to rescue...



Module concepts
for SaaS

Programming
abstractions for big
data/event processing

Reactive Computations



Event-driven programming in EScala

```
abstract class Figure {  
  evt changed[Unit] = afterExec(moveBy) || afterExec(setColor)  
  evt invalidated[Rectangle] = changed.map(() => getBounds())  
  
  def moveBy(dx: Int, dy: Int)  
  def setColor(c: Color) { ... }  
  def getBounds(): Rectangle  
}
```

```
class Rectangle extends Figure {  
  override evt changed[Unit] = super.changed || afterExec(setSize)  
  
  def setSize
```

```
class Circle extends Figure {  
  override evt changed[Unit] = super.changed || afterExec(setRadius)  
  def setRadius(r: Int) { ... }  
}
```

Event-driven programming in EScala

```
abstract class Figure {  
  evt changed[Unit] = afterExec(moveBy) || afterExec(setColor)  
  evt invalidated[Rectangle] = changed.map(() => getBounds())  
  
  def moveBy(dx: Int, dy: Int)  
  def setColor(c: Color) { ... }  
  def getBounds(): Rectangle  
}  
  
class Rectangle extends Figure {  
  override evt changed[Unit] = super.changed || afterExec(setSize)  
  
  def setSize(width: Int, height: Int) { ... }  
}  
  
class Circle extends Figure {  
  override evt changed[Unit] = super.changed || afterExec(setRadius)  
  def setRadius(r: Int) { ... }  
}
```

ec(setColor)
etBounds()

terExec(setSize)

```
def setSize(width: Int, height: Int) { ... }
```

```
class Drawing(val figures: List[Figure]) {  
  evt invalidated[Rectangle] = figures.any(_.invalidated)  
  ...  
}
```

setRadius)

```
class Connector(val start: Figure, val end: Figure) {  
  start.changed += ... handler code ...  
  end.changed += ...  
  ...  
}
```

Event-driven programming in EScala

- 😊 Reduced accidental complexity related to observer infrastructures
- 😊 Yet, efficiency comparable to observer pattern
- 😊 Type-safe, modular compilation

BUT...

Still accidental complexity when events used for propagating changes of data to computations depending on them

Time in the living environment in EScala

```
var hour : Int = 0
var day : Int = 0
var week : Int = 0

imperative evt tick[Unit]
evt newDay [Unit] = tick && (() => hour == 0)
evt newWeek [Unit] = ...

tick += () => { hour = (hour + 1) % 24 }
newDay += () => { day = (day + 1) % 7 }
newWeek += () => { ... }
```



☹ Not declarative style

☹ Observer infrastructure → **ACCIDENTAL COMPLEXITY**

Reactive Values

```
val tick = new Var(0) // incremented at every 60 min ...  
val hour = Signal { tick() % 24 }  
val day = Signal { (tick() / 24) % 7 + 1 }  
val week = Signal { ... }
```

- 😊 Declarative style
- 😊 Direct expression of domain semantics
→ **NO ACCIDENTAL COMPLEXITY**

BUT...

Reactive values and mutability

```
val source: Var[ListBuffer[Item]] = Var(new ListBuffer[Item])  
// source is mutable  
...  
val filtered: Signal[ListBuffer[Item]] =  
    Signal{ source.filter(pred) }  
...  
source() += new Item() // change not detected by sink
```

☹ fine grained imperative style changes not supported

☹ do not integrate with OO

Events as workaround...

```
class List[T] {  
  evt Event[T] elementAdded = ...  
  def addElement(x: T) { ... elementAdded() ... }  
  ...  
}  
  
class FilteredElemList[T](pred: T => Boolean, base: List[T]) {  
  evt filteredAdded[T] = base.elementAdded(elem) && pred(elem)  
  
  filteredAdded += addElement  
  def addElement(x: T) { ... }  
  ...  
}
```

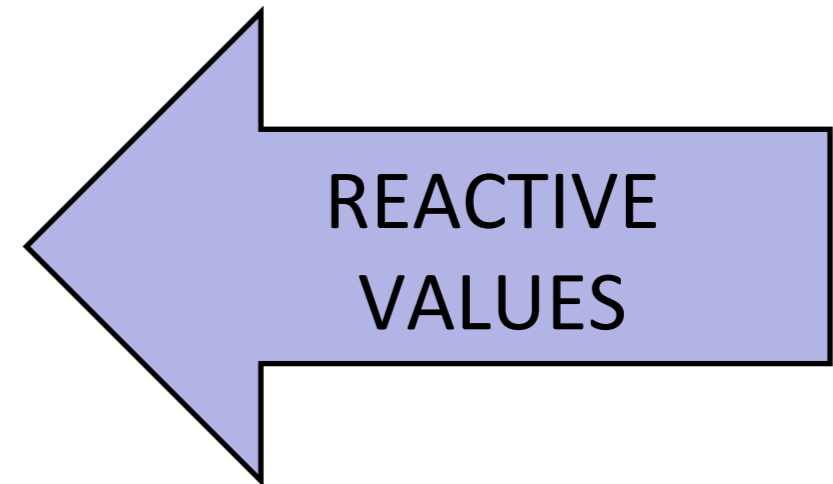
😊 fits into the OO design

😊 mutability, fine-grained changes well supported

RESCALA = R(eactive)E(vent)Scala



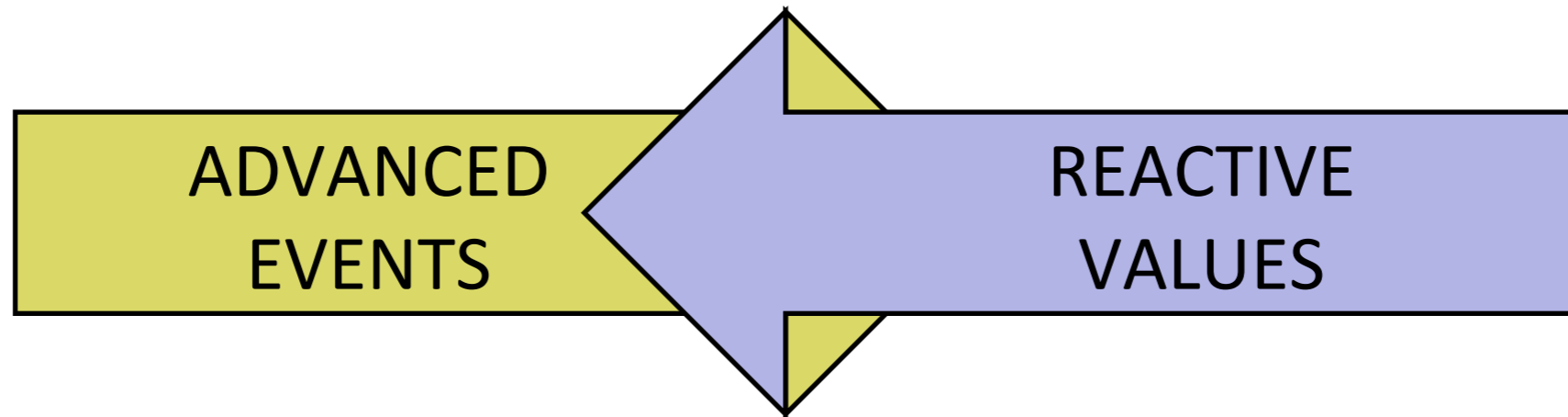
- ☹ Not declarative style.
- ☹ Accidental complexity
- 😊 Fit into the OO design
- 😊 Fine-grained changes



- 😊 Declarative style
- 😊 No accidental complexity
- ☹ Imperative changes not supported
- ☹ Do not integrate with OO

Philosophy:
foster a declarative and functional style without
sacrificing the power of OO design

... as unified abstractions



[Signal → Event]	Signal.changed
[Event → Signal]	Event.hold
[(S, E) → Signal]	snapshot
[(S, S, E) → Signal]	toggle
...	

So far, so good ...

BUT...

... we're still using events as a workaround for lack of proper support for reactive computations over mutable objects ...

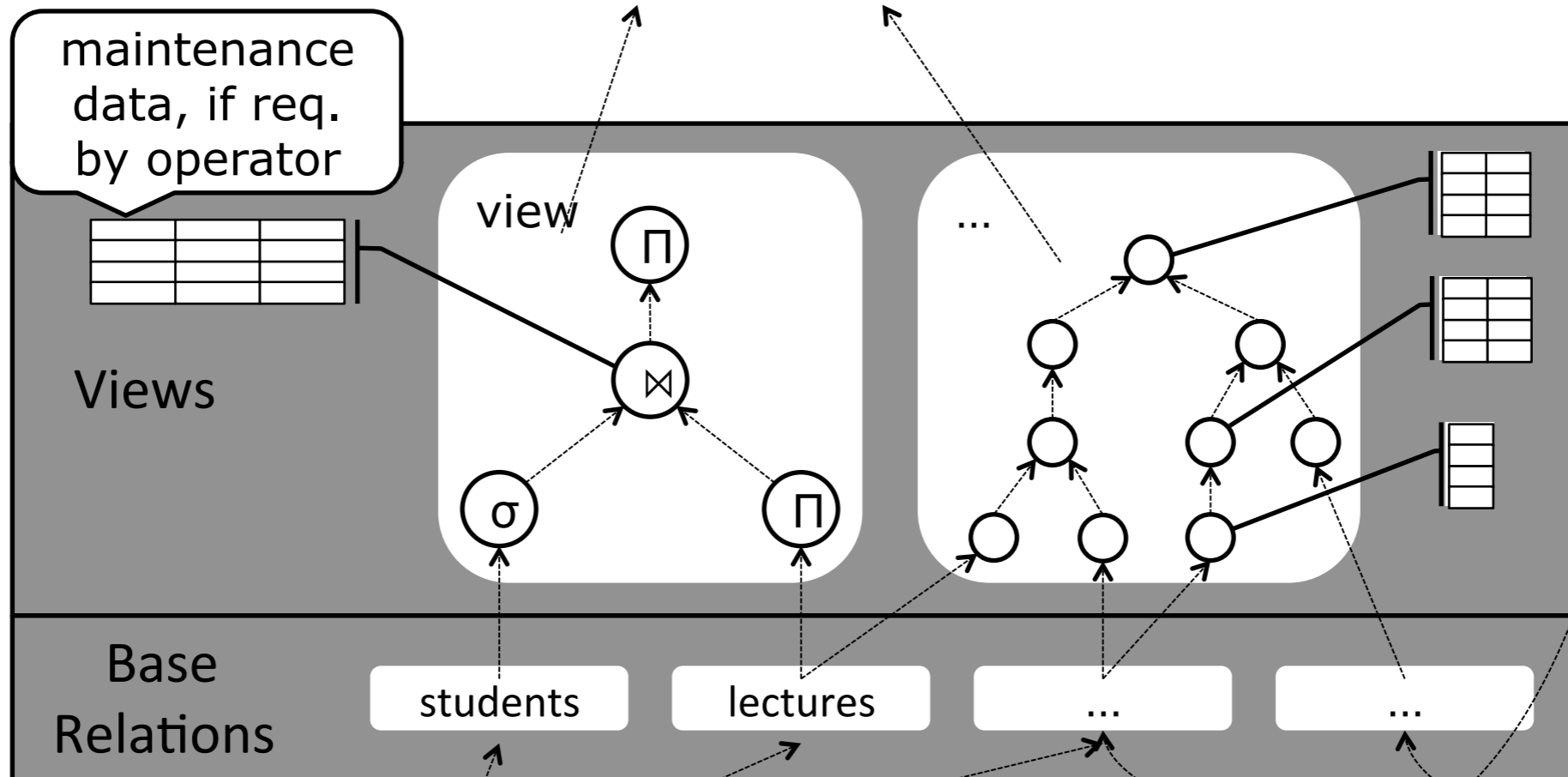


Ongoing: Support for dependable views

Application

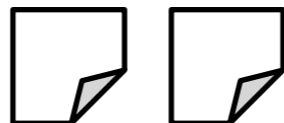
```
val view = SELECT (*) FROM students WHERE (_.firstName== ...)
...
students.add(new Student("Sally", "Fields"))
```

Query Engine



Data

e.g., file based



or application driven

Ongoing ...



Seamless integration of incremental data views with the rest of REScala?

Generalization beyond relational algebra for incrementally maintainable computations?

Support for asynchronicity, CEP?

Support for distribution and parallelism?

...

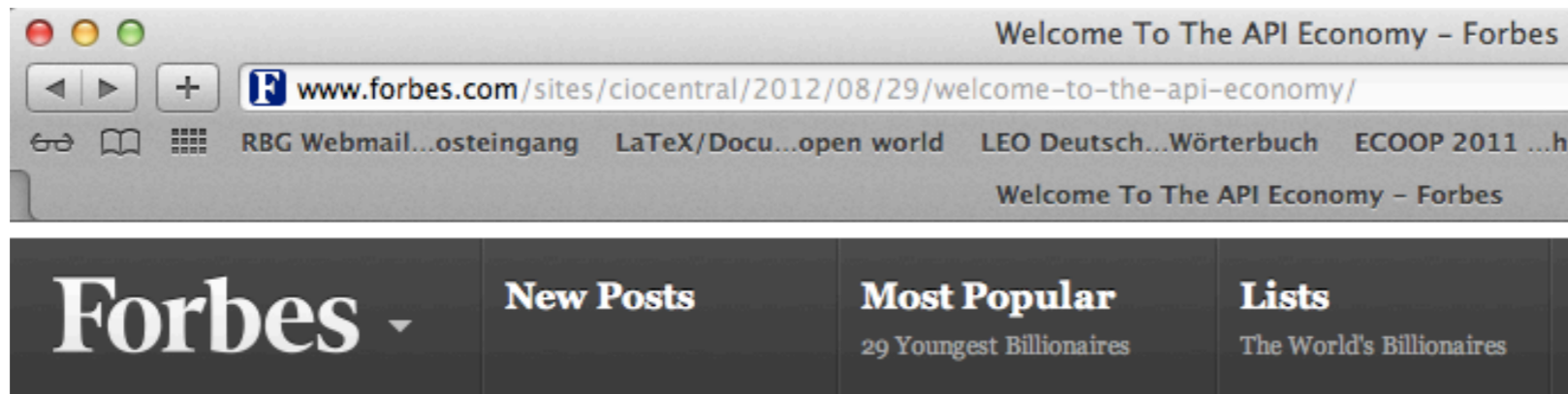


Programming Concepts for the Cloud

Cloud for Software Development

Code Recommenders Project

ubiquitous APIs ...




CIO Network

INSIGHTS AND IDEAS FOR TECHNOLOGY LEADERS.

+ Follow (483)

TECH | 8/29/2012 @ 3:14PM | 5,159 views

Welcome To The API Economy

 Eric Savitz, Forbes Staff

... Salesforce.com generates more than half of its \$2.3 billion in revenue through its APIs, not its user interfaces. Twitter is said to process 13 billion transactions a day through its APIs. Google is around 5 billion transactions a day. For its part, Amazon is rapidly closing in on a trillion transactions ...

About 33,500,000 results (0.14 seconds)

[10 reasons APIs and development platforms are difficult to use ...](#)

communicationcloud.wordpress.com/2013/01/30/926/ ▾

Jan 30, 2013 – In 2012 I worked for several customers on a range of projects which were essentially about making it easier for people to **use APIs**, SDKS, ...

[Rails API difficult to use - Ruby on Rails: Documentation | Google ...](#)

groups.google.com/group/rubyonrails-docs/.../82acf547947f814e ▾

May 25, 2012 – I find the Rails **API difficult to use**, in terms of finding the classes and methods I want with a minimum number of mouse clicks. I don't care if it ...

[\[PDF\] What Makes APIs Hard to Learn? - School of Computer Science ...](#)

www.cs.mcgill.ca/~martin/papers/software2009a.pdf ▾

by MP Robillard - 2009 - Cited by 62 - Related articles

make an **API hard** to learn? To answer this question, I investigated the obstacles professional developers at Microsoft faced when learning how to **use APIs**.

[\[PDF\] What Makes APIs Difficult to Use? - IJCSNS - International Jour...](#)

paper.ijcsns.org/07_book/200804/20080436.pdf ▾

by MF Zibrán - 2008 - Cited by 6 - Related articles

Apr 5, 2008 – Therefore, APIs should be implemented to have high usability, and to this extent a good understanding of what makes **APIs difficult to use** ...

[\[PDF\] Why are APIs Difficult to Learn and Use?](#)

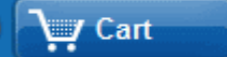
web.engr.oregonstate.edu/~cscaffid/papers/pr_20060601_api.pdf ▾

by C Scaffidi - Cited by 10 - Related articles

Shop All Departments

Search Books

GO



Wish List

Books

[Advanced Search](#)

[Browse Subjects](#)

[New Releases](#)

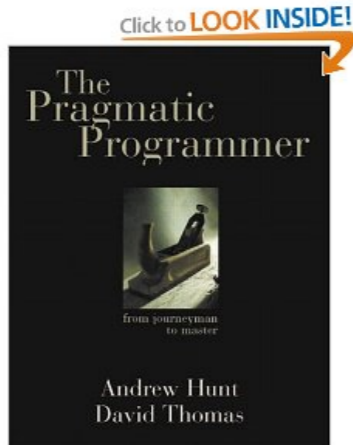
[Bestsellers](#)

[The New York Times® Bestsellers](#)

[Libros En Español](#)

[Bargain Books](#)

[Textbooks](#)



The Pragmatic Programmer: From Journeyman to Master [Paperback]

[Andrew Hunt](#) (Author), [David Thomas](#) (Author)

★★★★☆ (150 customer reviews)

List Price: ~~\$49.99~~

Price: **\$38.99** & this item ships for **FREE with Super Saver Shipping.** [Details](#)

You Save: **\$11.00 (22%)**

In Stock.

Ships from and sold by **Amazon.com**. Gift-wrap available.

Customers Who Bought This Item Also Bought

Page 1 of 20



[Code Complete: A Practical Handbook of Software...](#) by Steve McConnell

★★★★☆ (119)

\$31.49



[Clean Code: A Handbook of Agile Software Craf...](#) by Robert C. Martin

★★★★☆ (67)

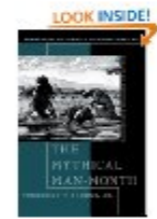
\$35.99



[Refactoring: Improving the Design of Existing Code](#) by Martin Fowler

★★★★☆ (141)

\$50.69



[The Mythical Man-Month: Essays on Software...](#) by Frederick P. Brooks

★★★★☆ (145)

\$32.94



[The Passionate Programmer: Creating a Remarkable C...](#) by Chad Fowler

★★★★☆ (32)

\$16.29

What Do Customers Ultimately Buy After Viewing This Item?



75% buy the item featured on this page:

[The Pragmatic Programmer: From Journeyman to Master](#) ★★★★★ (150)

\$38.99



9% buy

[Code Complete: A Practical Handbook of Software Construction](#) ★★★★★ (119)

\$31.49



6% buy

[Design Patterns: Elements of Reusable Object-Oriented Software](#) ★★★★★ (271)

\$35.57

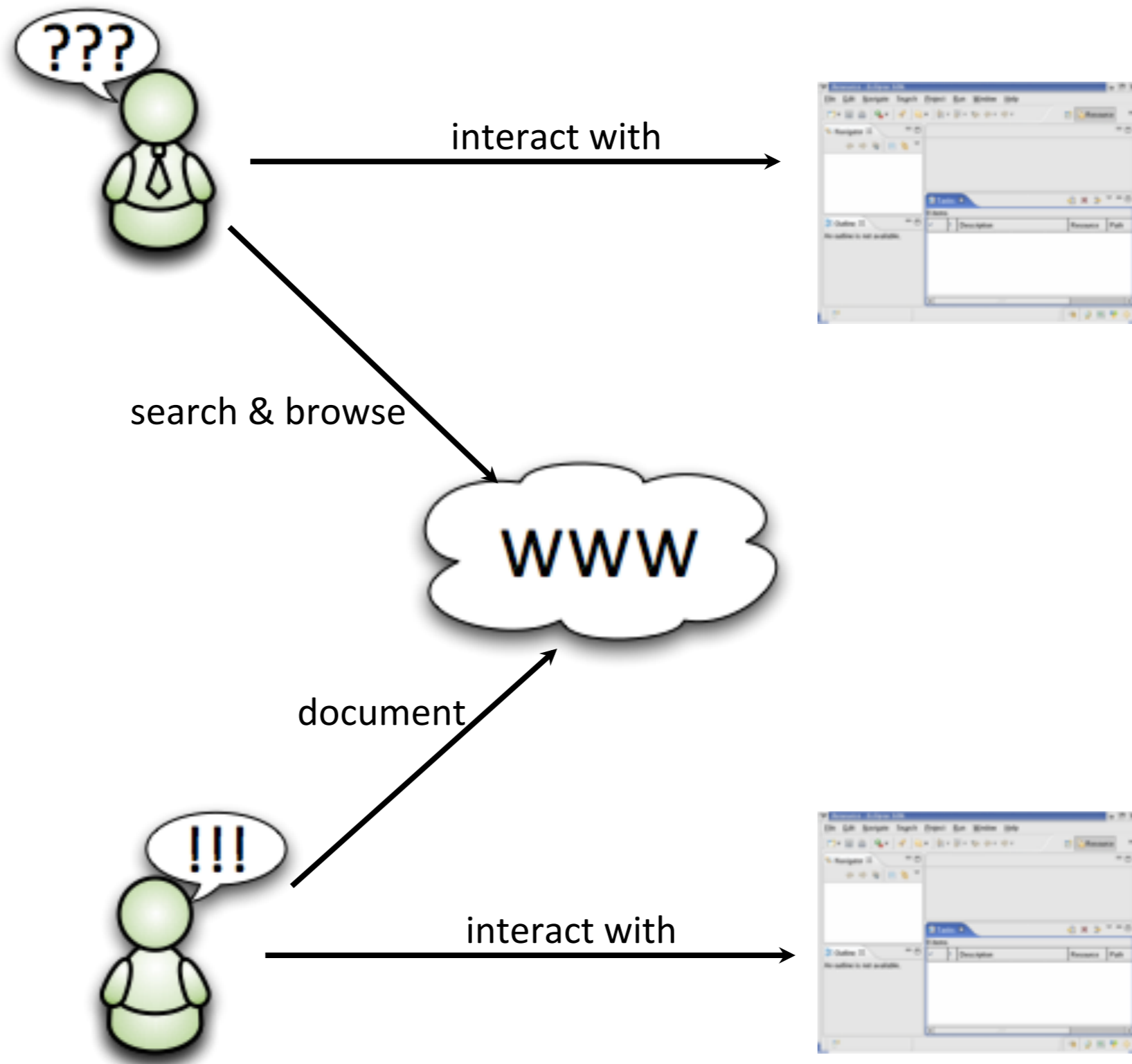


5% buy

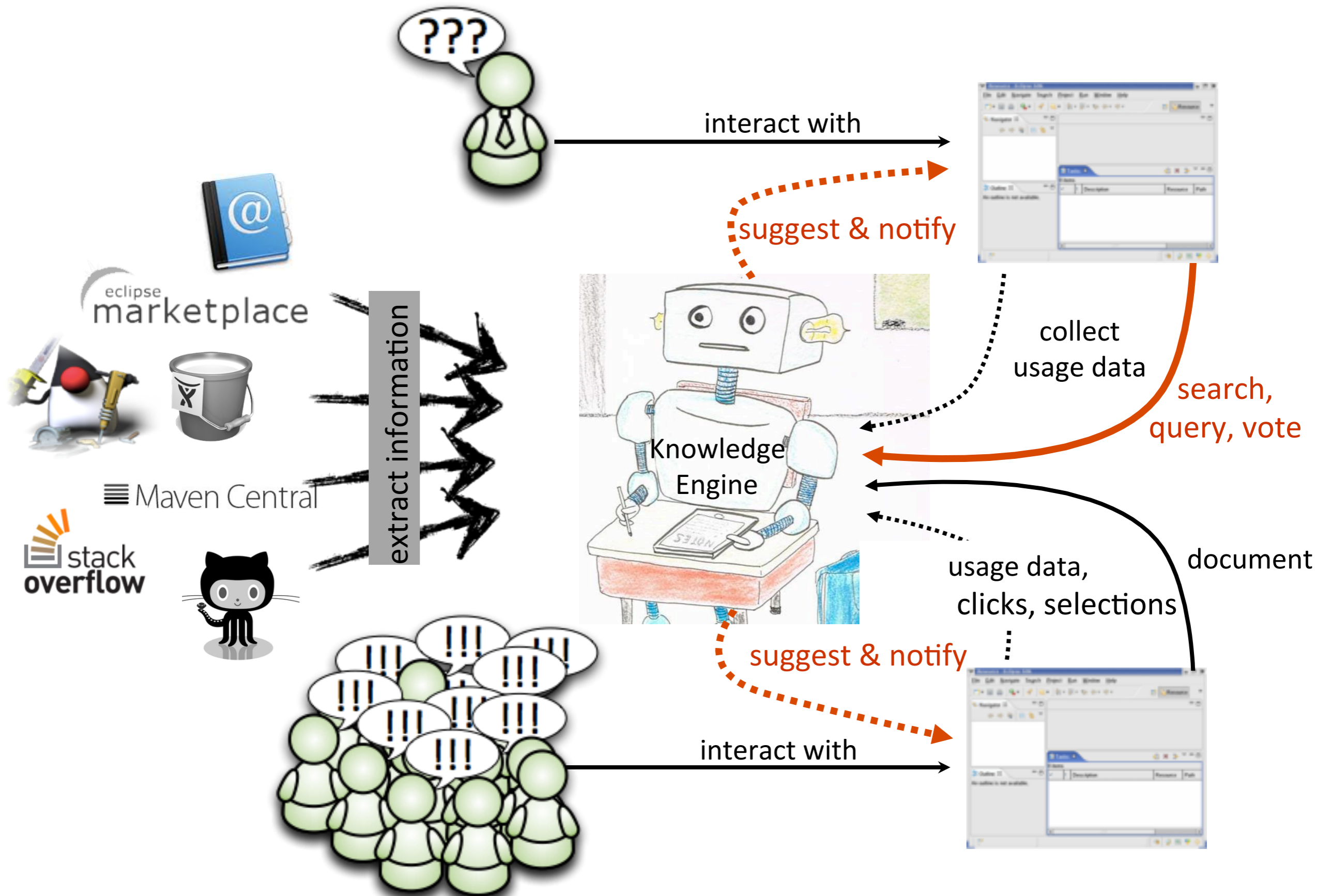
[Clean Code: A Handbook of Agile Software Craftsmanship](#) ★★★★★ (67)

\$35.99

Software development 1.0 ...



Software development 2.0



Completion assistance today

The screenshot shows the Eclipse IDE with a Java file named `Parser.java` open. The code is as follows:

```
1 package example;
2
3 import java.util.regex.Matcher;
4
5
6 public class Parser {
7
8     public void parse(String code) {
9         Pattern p = Pattern.compile("[eHllo]{5} [dlorW]{5}!$");
10        Matcher m = p.matcher(code);
11        m.
12    }
13 }
14
```

A completion popup is visible over the `m.` prefix on line 11. The popup lists several methods of the `Matcher` class, with `matches() : boolean - Matcher` selected. To the right of the popup, a yellow tooltip provides details for the `matches()` method:

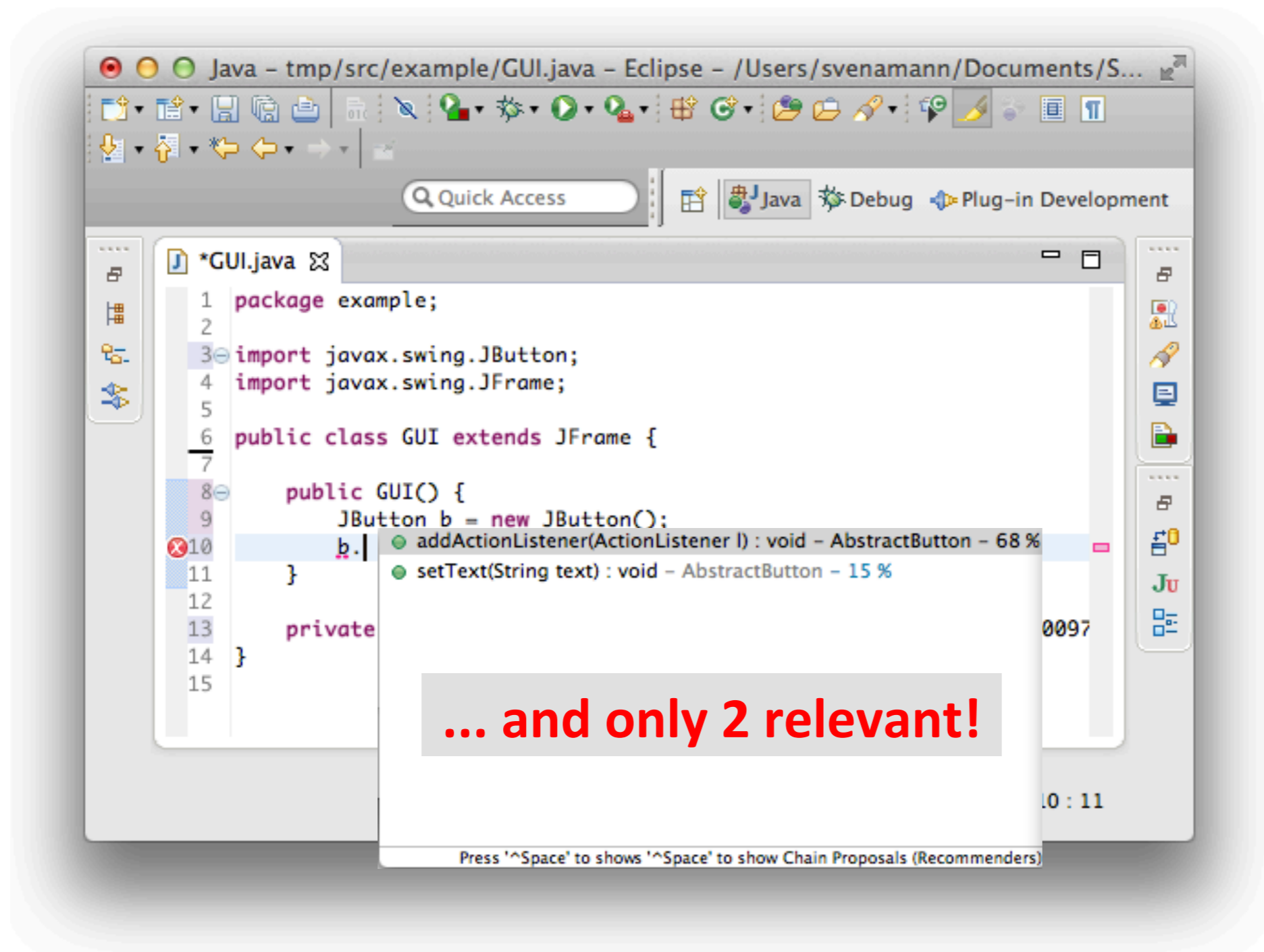
Attempts to match the entire region against the pattern.

If the match succeeds then more information can be obtained via the `start`, `end`, and `group` methods.

Returns:
`true` if, and only if, the entire region sequence matches this matcher's pattern

Press '^Space' to show Code Recommenders Proposals (feedback) Press 'Tab' from proposal table or click for focus

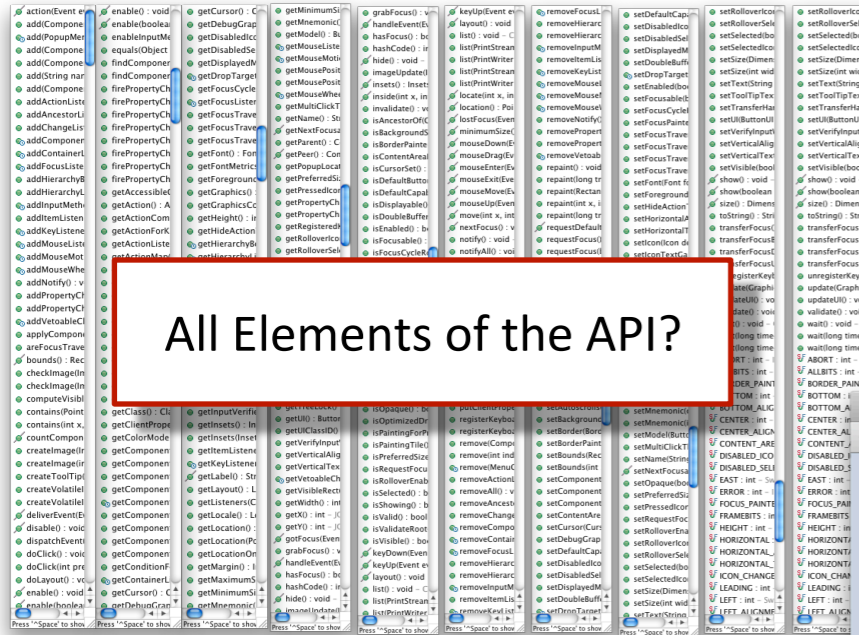
Intelligent completion assistance



Code Recommenders is...

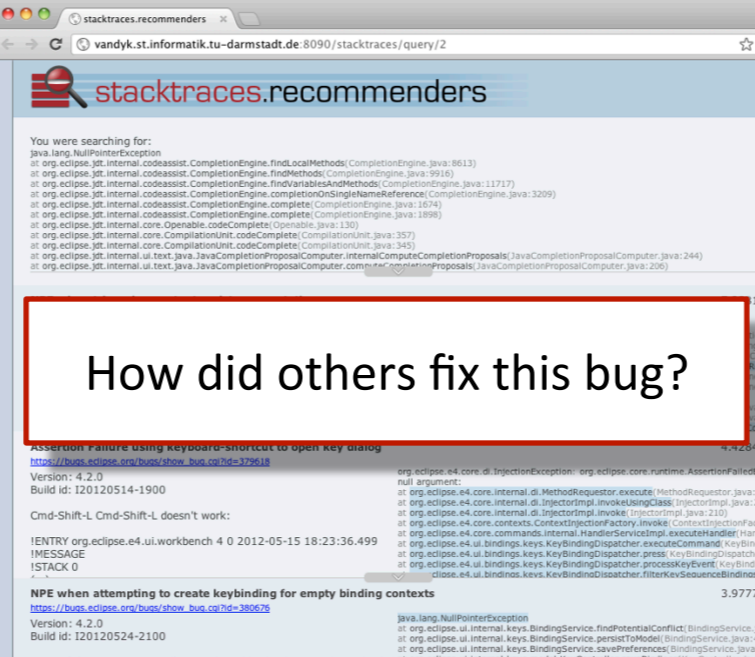
Usage-Driven Docs

Intelligent Code Completion



All Elements of the API?

Debugging 2.0

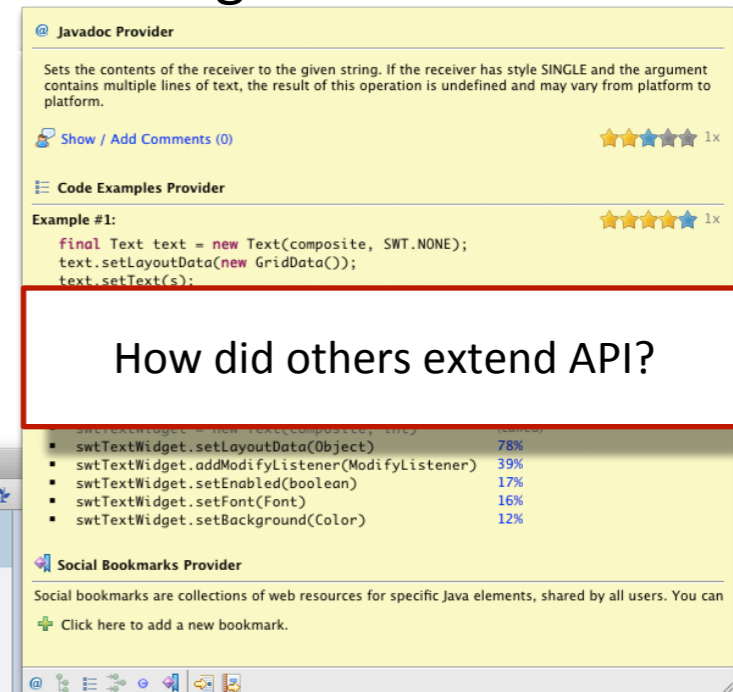


How did others fix this bug?

Smart Bug Detection



What have I missed?



How did others extend API?

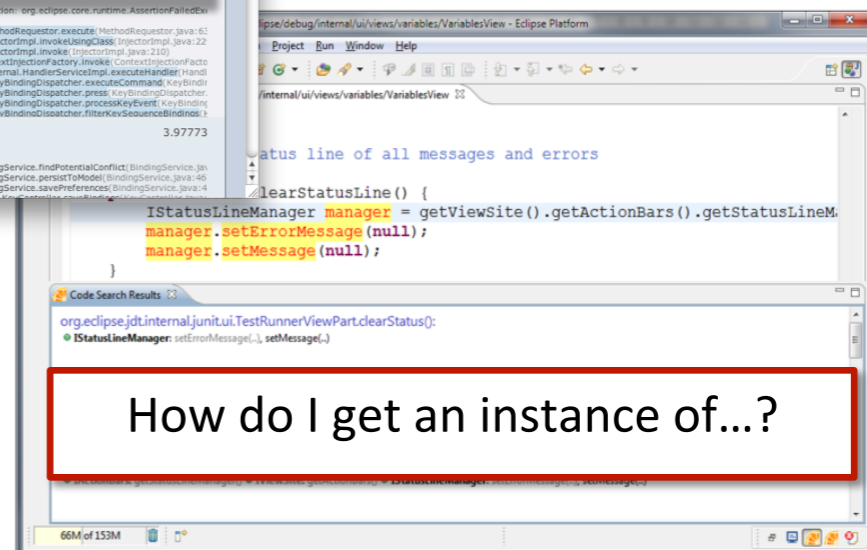
- swt.Widget.setLayoutData(Object) 78%
- swt.Widget.addModifyListener(ModifyListener) 39%
- swt.Widget.setEnabled(boolean) 17%
- swt.Widget.setFont(Font) 16%
- swt.Widget.setBackground(Color) 12%

Social Bookmarks Provider

Social bookmarks are collections of web resources for specific Java elements, shared by all users. You can

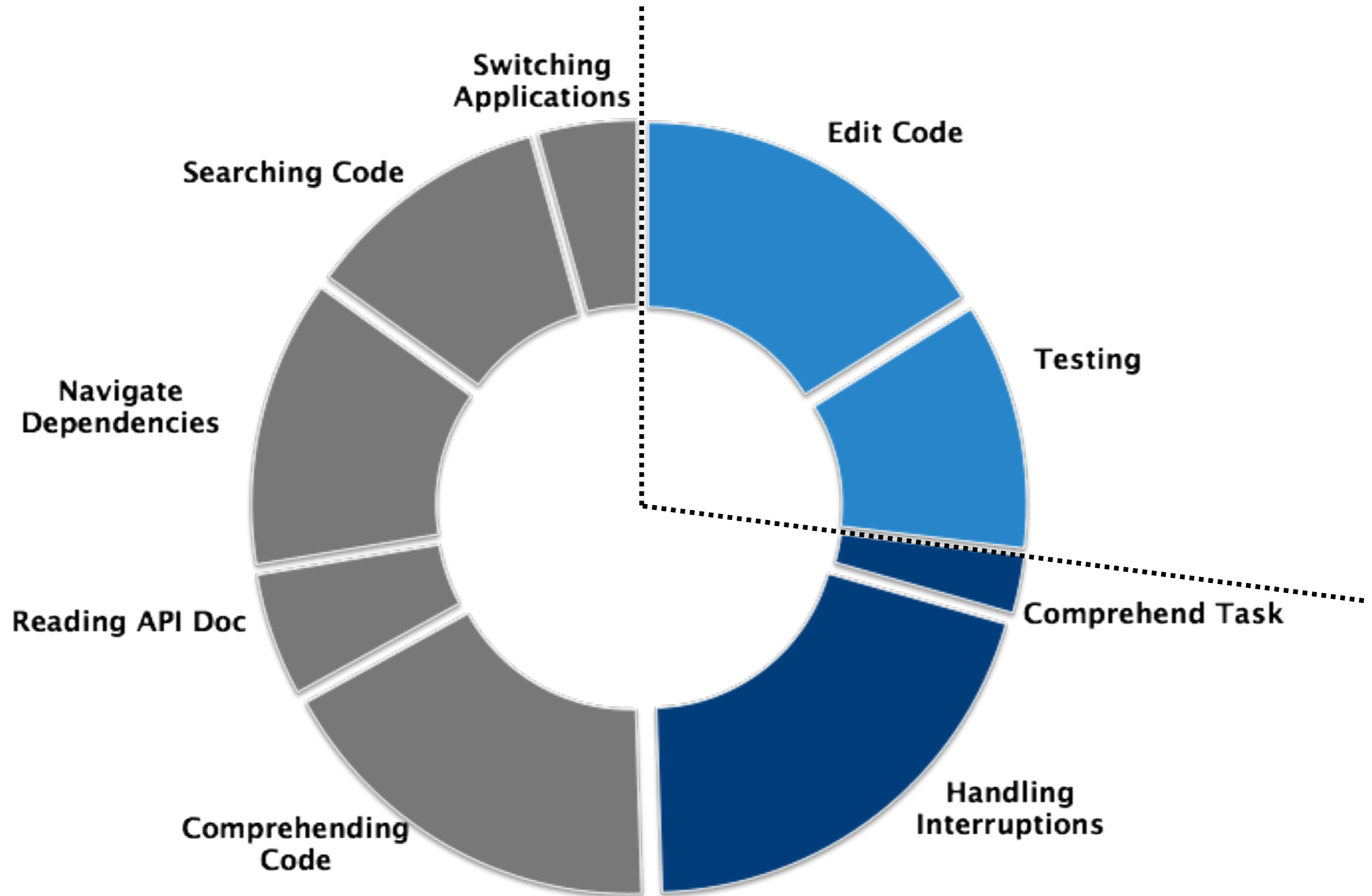
Click here to add a new bookmark.

Example Recommender

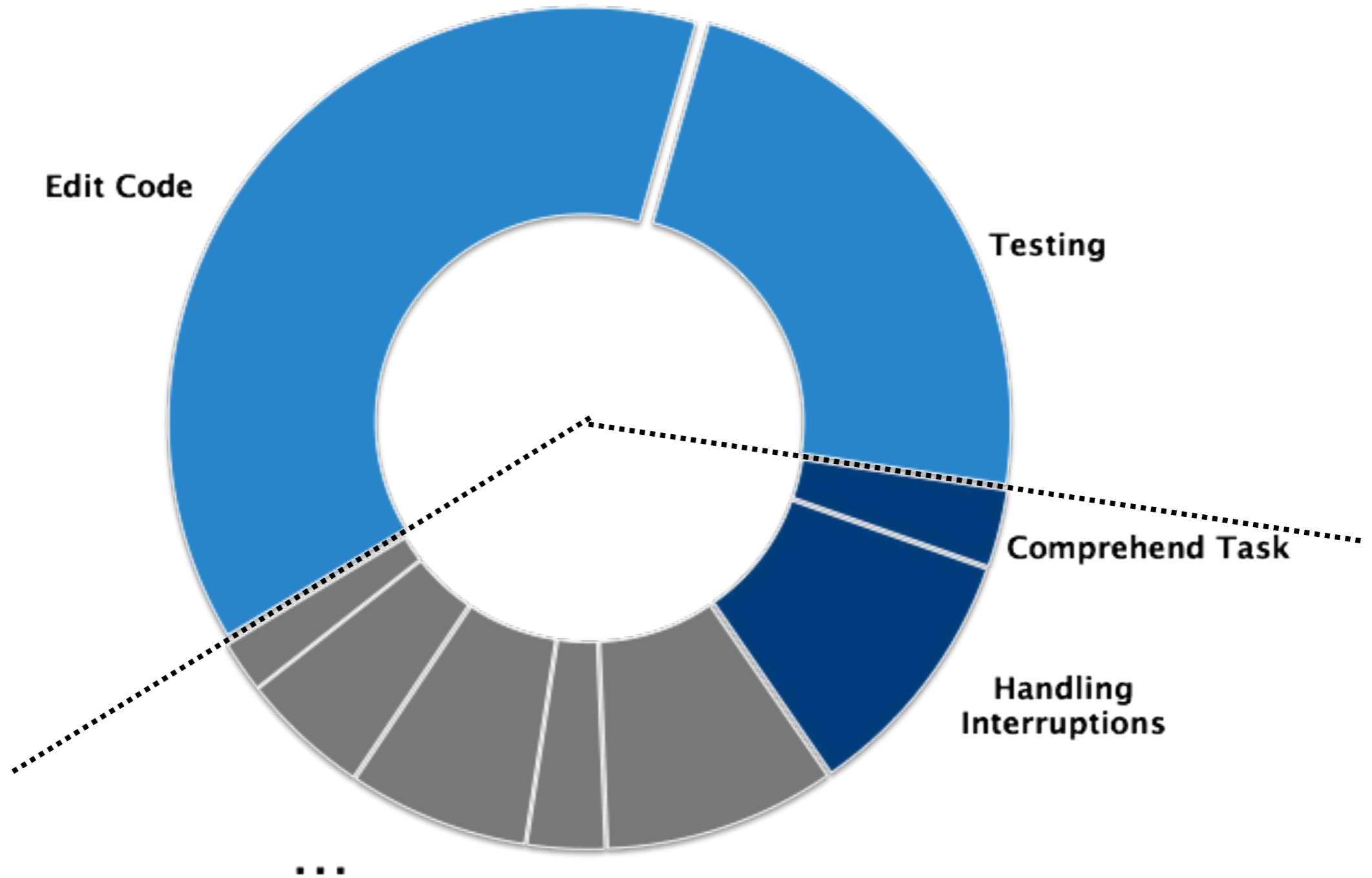


How do I get an instance of...?

A Developer's Day 1.0 ...



A Developer's Day 2.0 ...





Project History:

- Started in 2007
- Published in 2009 as Eclipse Code Recommenders
- Eclipse Incubator Project in 2011
- Part of Eclipse Juno (4.2) - Java Package in 2012

Awards:

- IBM Innovation Award 2006
- Uni-TechSpin Ideas Competition 2010
- Eclipse Community Award as Most Innovative Eclipse Project 2012
- BMWi Exist StartUp Program

Q & A

