







# Speculative Analysis: Exploring Future States of Software

Yuriy Brun   
Michael D. Ernst 

Reid Holmes   
David Notkin 

 University of Washington

 University of Waterloo

Have you ever made a mistake  
while programming  
and only realized it later?

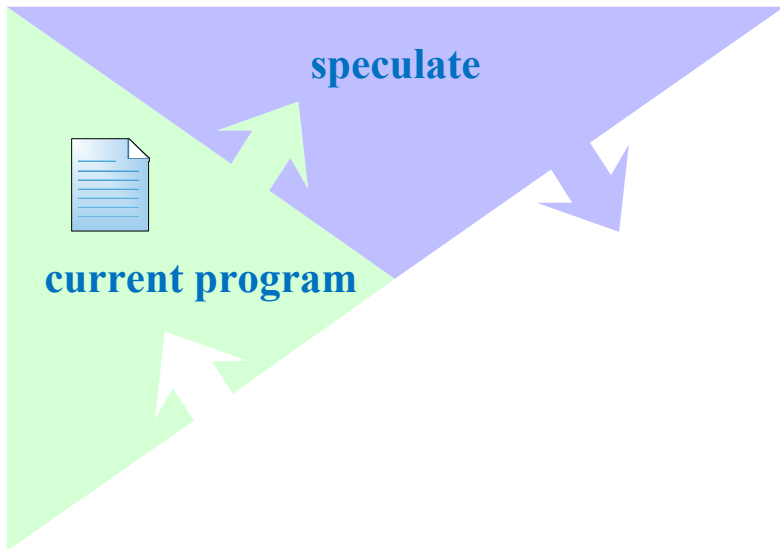
- design decision
- refactoring
- repeated someone else's work

# Speculative analysis: Predict the future and analyze it

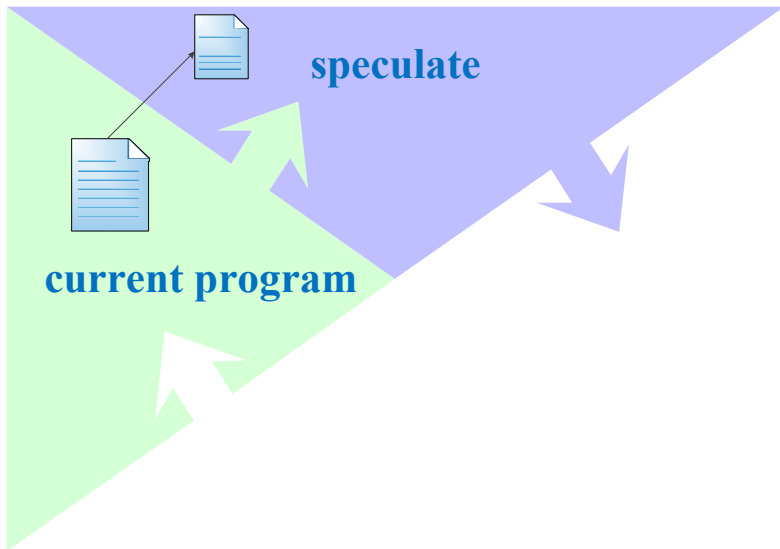


**current program**

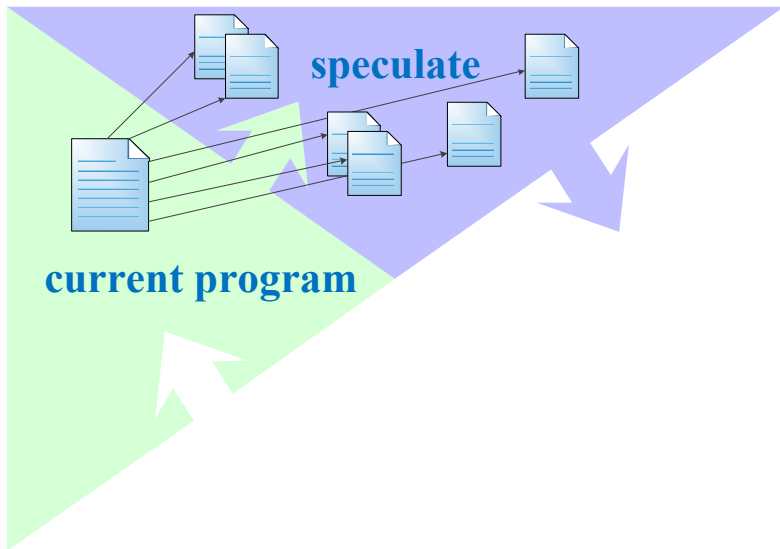
# Speculative analysis: Predict the future and analyze it



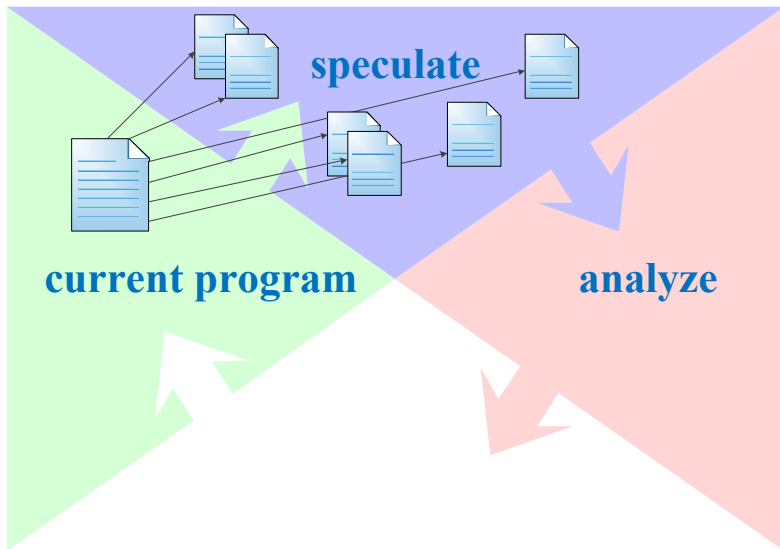
# Speculative analysis: Predict the future and analyze it



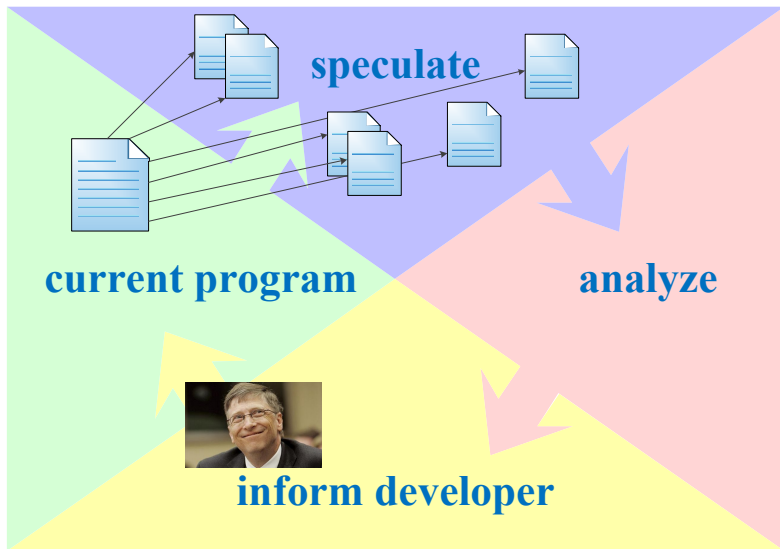
# Speculative analysis: Predict the future and analyze it



# Speculative analysis: Predict the future and analyze it

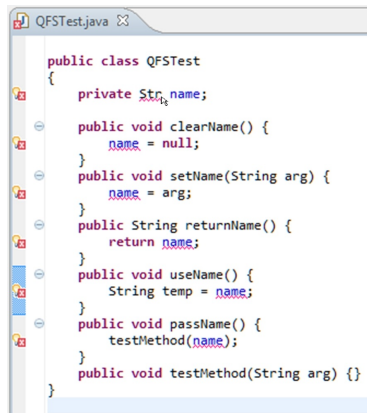


# Speculative analysis: Predict the future and analyze it





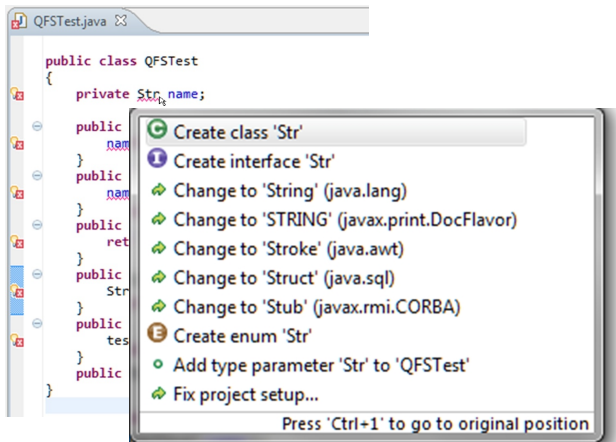
# Speculative Quick Fix



```
public class QFSTest
{
    private String name;

    public void clearName() {
        name = null;
    }
    public void setName(String arg) {
        name = arg;
    }
    public String returnName() {
        return name;
    }
    public void userName() {
        String temp = name;
    }
    public void passName() {
        testMethod(name);
    }
    public void testMethod(String arg) {}
}
```

# Speculative Quick Fix



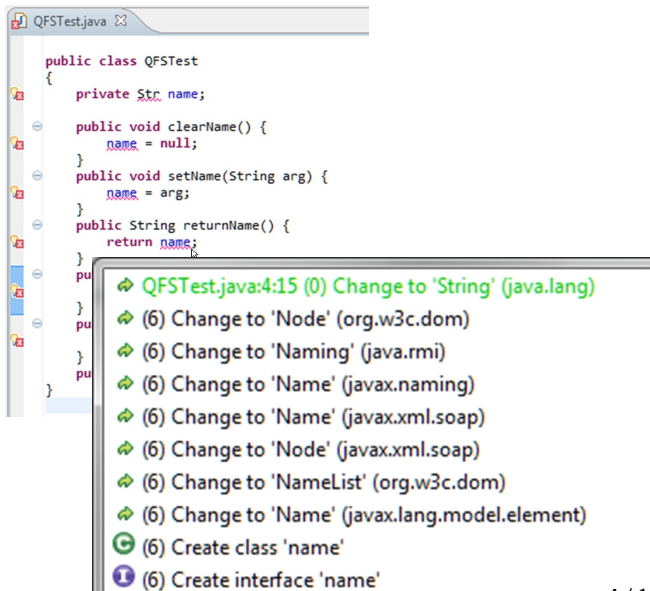
# Speculative Quick Fix

The screenshot shows a code editor window titled 'QFSTest.java'. The code defines a public class 'QFSTest' with several methods. A red squiggly line is under the 'Str' type in the 'private Str name;' declaration. A context menu is open over this line, listing several quick fixes:

- (0) Change to 'String' (java.lang)
- (4) Change to 'Stub' (javax.rmi.CORBA)
- (4) Change to 'Struct' (java.sql)
- (4) Change to 'Stroke' (java.awt)
- (4) Add type parameter 'Str' to 'QFSTest'
- (6) Create class 'Str'
- (6) Create interface 'Str'
- (6) Create enum 'Str'
- (6) Fix project setup...
- (N/A) Change to 'STRING' (javax.print.DocFlavor)

At the bottom of the menu, it says: Press 'Ctrl+1' to go to original position

# Speculative Quick Fix



```
QFSTest.java X
public class QFSTest
{
    private Str name;

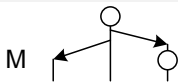
    public void clearName() {
        name = null;
    }
    public void setName(String arg) {
        name = arg;
    }
    public String returnName() {
        return name;
    }
}
pu
}
pu
}
pu
```

- ➡ QFSTest.java:4:15 (0) Change to 'String' (java.lang)
- ➡ (6) Change to 'Node' (org.w3c.dom)
- ➡ (6) Change to 'Naming' (java.rmi)
- ➡ (6) Change to 'Name' (javax.naming)
- ➡ (6) Change to 'Name' (javax.xml.soap)
- ➡ (6) Change to 'Node' (javax.xml.soap)
- ➡ (6) Change to 'NameList' (org.w3c.dom)
- ➡ (6) Change to 'Name' (javax.lang.model.element)
- Ⓞ (6) Create class 'name'
- ⓘ (6) Create interface 'name'

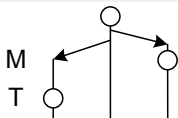
## Contributions

- Speculative analysis
- Speculative analysis for collaborative development  
Crystal: prototype tool
- Problem space ideal for search-based solutions

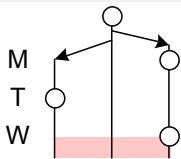
# The Gates conflict



# The Gates conflict

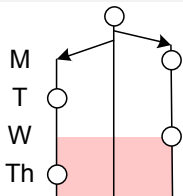


# The Gates conflict

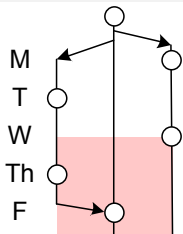




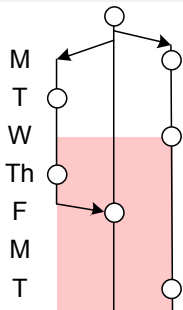
# The Gates conflict



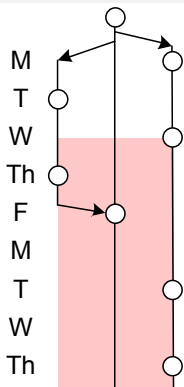
# The Gates conflict



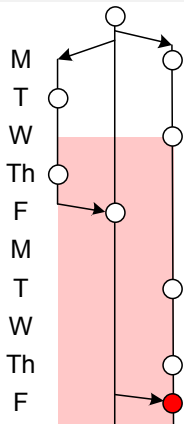
# The Gates conflict



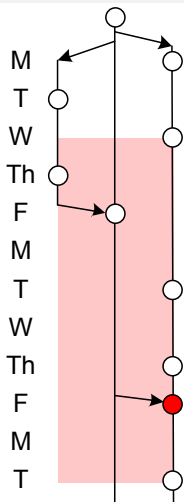
# The Gates conflict



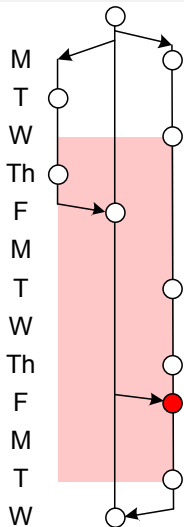
# The Gates conflict



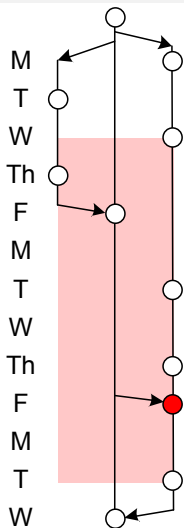
# The Gates conflict



# The Gates conflict



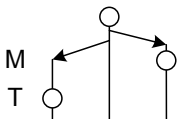
# The Gates conflict



The information was all there, but the developers didn't know it.

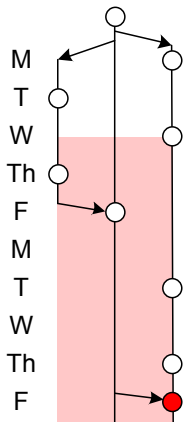


## What could well-informed developers do?



- Avoid conflicts

# What could well-informed developers do?



- Avoid conflicts

- Reduce conflict severity

# Introducing Crystal: A proactive conflict detector

## DEMO

# Introducing Crystal: A proactive conflict detector

## DEMO

The screenshot shows a window titled "Crystal - George" with a menu bar containing "File" and "About". The main content area is divided into two sections:

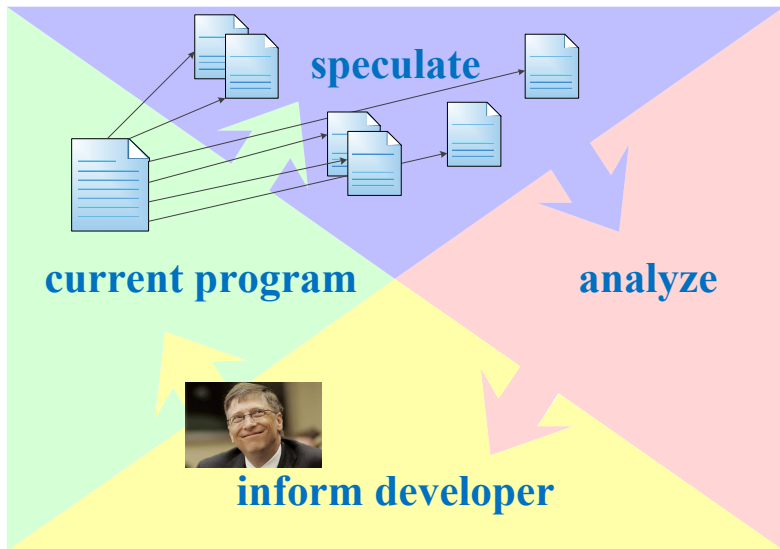
- Let it Be** (hg commit):
  - master: green checkmark
  - Paul: green up arrow
  - Ringo: yellow crossed arrows
  - John: green down arrow
- Handle with Care**:
  - master: red up arrow
  - Jeff: red up arrow
  - Roy: green up arrow
  - Bob: red up arrow with a small 'B' next to it
  - Tom: red up arrow

A tooltip is visible over the Jeff commit, containing the following text:

- Action: hg fetch
- Consequences: new relationship will be AHEAD
- Committers: George and Tom

<http://crystalvc.googlecode.com>

# Speculative analysis in collaborative development



# Reducing false positives in conflict prediction

## Collaborative awareness

- Palantír [Sarma et al. 2003]
- FASTDash [Biehl et al. 2007]
- Syde [Hattori and Lanza 2010]
- CollabVS [Dewan and Hegde 2007]
- Safe-commit [Wloka et al. 2009]
- SourceTree [Streeting 2010]

## Reducing false positives in conflict prediction

### Collaborative awareness

- Palantír [Sarma et al. 2003]
- FASTDash [Biehl et al. 2007]
- Syde [Hattori and Lanza 2010]
- CollabVS [Dewan and Hegde 2007]
- Safe-commit [Wloka et al. 2009]
- SourceTree [Streeting 2010]

Crystal analyzes **concrete artifacts**,  
eliminating false positives and false negatives.

## Utility of proactive collaborative conflict detection

- Are textual collaborative conflicts a real problem?
  - 16% of the merges have textual conflicts.
  - Conflicts live a mean of 9.8 days.
- How dangerous are safe merges?
  - 93% of textual conflicts developed from safe merges.
  - 20% of textually-safe merges developed into conflicts.
- Do higher-order collaborative conflicts exist?
  - One in three conflicts are of higher-order.

[Brun et al. 2011]



## Crystal is in the wild

“Crystal handles several projects and users effortlessly and presents the necessary information in a simple and understandable way.”

– a user

### Microsoft Beacon

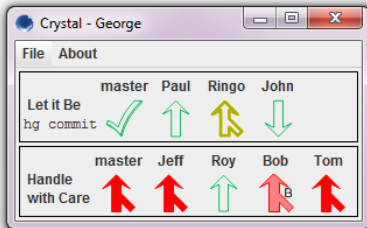
- A centralized version control-based tool.
- Microsoft product groups will use Beacon to help identify conflicts earlier in the development process.
- We will conduct user studies to measure effects on developers.

Open problem:

How to search speculative analysis'  
many possible futures?

## Contributions

- Introduced **speculative analysis** to guide future actions.
- Developed Crystal to **precisely** detect conflicts and **unobtrusively** inform developers.
- Identified a problem space ideal for search-based solutions.



<http://crystalvc.googlecode.com>

- Jacob T. Biehl, Mary Czerwinski, Greg Smith, and George G. Robertson. FASTDash: A visual dashboard for fostering awareness in software teams. pages 1313–1322, San Jose, CA, USA, 2007. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240823.
- Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Proactive detection of collaboration conflicts. In *ESEC FSE*, Szeged, Hungary, Sep. 2011.
- Prasun Dewan and Rajesh Hegde. Semi-synchronous conflict detection and resolution in asynchronous software development. In *ECSCW*, pages 159–178, Limerick, Ireland, 2007.
- Lile Hattori and Michele Lanza. Syde: A tool for collaborative software development. In *ICSE Tool Demo*, pages 235–238, Cape Town, South Africa, May 2010. ISBN 978-1-60558-719-6. doi: 10.1145/1810295.1810339.
- Anita Sarma, Zahra Noroozi, and André van der Hoek. Palantír: Raising awareness among configuration management workspaces. In *ICSE*, pages 444–454, Portland, OR, May 2003. ISBN 0-7695-1877-X.
- Steve Streeting. Sourcetree. <http://www.sourcetreeapp.com>, 2010.
- Jan Wloka, Barbara Ryder, Frank Tip, and Xiaoxia Ren. Safe-commit analysis to facilitate team software development. In *ICSE*, pages 507–517, Vancouver, BC, Canada, May 2009. ISBN 978-1-4244-3453-4. doi: 10.1109/ICSE.2009.5070549.