

## Research Interests

---

I am interested in Software Engineering and Programming Languages, especially techniques that increase developer productivity and reduce developer mistakes. I currently work on code review tools and code reviewing process. My dissertation focused on enhancing software development techniques using copies of the developers' codebase.

## Jobs

---

### Microsoft, Tools for Software Engineers

October'12–Present

I work on code review tools and code reviewing process, improving developer productivity and software quality.

## Education

---

University of Washington, PhD, Computer Science & Engineering, June 2015

- Dissertation topic: Enhancing Software Development Techniques via Copy Codebases.
- Advisors: Michael D. Ernst and Yuriy Brun.

Universtion of Washington, M.S., Computer Science & Engineering, December 2012.

Koç University, Istanbul, Turkey, B.S., Computer Science, June 2010, *valedictorian*.

## Publications

---

### Refereed Journal Articles

- [1] **Kıvanç Muşlu**, Yuriy Brun, Michael D. Ernst, and David Notkin. Reducing Feedback Delay of Software Development Tools via Continuous Analyses (in press). *IEEE Transactions on Software Engineering (TSE)*, 2015.
- [2] Serdar Tasiran, Erkan M. Keremoğlu, and **Kıvanç Muşlu**. Location Pairs: A Test Coverage Metric for Shared-memory Concurrent Programs. *Empirical Software Engineering*, 16:1–37, July 2011.

### Refereed Conference Publications

- [3] **Kıvanç Muşlu**, Luke Swart, Yuriy Brun, and Michael D. Ernst. Simplifying development history information retrieval via multi-grained views. In *the 30th International Conference on Automated Software Engineering, New Ideas Track, ASE'15*, Lincoln, NE, USA, November 2015.
- [4] **Kıvanç Muşlu**, Yuriy Brun, and Alexandra Meliou. Preventing Data Errors with Continuous Testing. In *International Symposium on Software Testing and Analysis, ISSTA'15*, Baltimore, MD, USA, July 2015.
- [5] Sai Zhang, Darioush Jalalinasab, Jochen Wuttke, **Kıvanç Muşlu**, Wing Lam, Michael D. Ernst, and David Notkin. Empirically Revisiting the Test Independence Assumption. In *International Symposium on Software Testing and Analysis, ISSTA'14*, pages 385–396, San Jose, CA, USA, July 2014.
- [6] **Kıvanç Muşlu**, Christian Bird, Nachi Nagappan, and Jacek Czerwonka. Transition from Centralized to Distributed Version Control Systems: A Case Study on Reasons, Barriers, and Outcomes. In *the 36th International Conference on Software Engineering, ICSE'14*, pages 334–344, Hyderabad, India, June 2014.
- [7] **Kıvanç Muşlu**, Yuriy Brun, Michael D. Ernst, and David Notkin. Making Offline Analyses Continuous. In *the 9th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering, ESEC/FSE'13*, pages 323–333, Saint Petersburg, Russia, August 2013.
- [8] **Kıvanç Muşlu**, Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Speculative Analysis of Integrated Development Environment Recommendations. In *the 3rd Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA'12*, pages 669–682, Tucson, AZ, USA, October 2012.

- [9] Werner Dietl, Stephanie Dietzel, Michael D. Ernst, **Kıvanç Muşlu**, and Todd W. Schiller. Building and Using Pluggable Type-Checkers. In *the 33rd International Conference on Software Engineering, Software Engineering in Practice Track*, ICSE'11, pages 681–690, Waikiki Honolulu, HI, USA, May 2011.
- [10] Ali Sezgin, Serdar Tasiran, **Kıvanç Muşlu**, and Shaz Qadeer. Run-time Verification of Optimistic Concurrency. In *the First International Conference on Runtime Verification*, RV'10, pages 384–398, St. Julians, Malta, November 2010.

## Refereed Short Publications

- [11] **Kıvanç Muşlu**, Yuriy Brun, and Alexandra Meliou. Data Debugging with Continuous Testing. In *the 9th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering , New Ideas Track*, ESEC/FSE'13, pages 631–634, Saint Petersburg, Russia, August 2013.
- [12] **Kıvanç Muşlu**. Integrating Systematic Exploration, Analysis, and Maintenance in Software Development. In *the 35th International Conference on Software Engineering , Doctoral Symposium Track*, ICSE'13, pages 1389–1392, San Francisco, CA, USA, May 2013.
- [13] **Kıvanç Muşlu**, Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Improving IDE Recommendations by Considering Global Implications of Existing Recommendations. In *the 34th International Conference on Software Engineering , New Ideas and Emerging Results Track*, ICSE'12, pages 1349–1352, Zurich, Switzerland, June 2012.
- [14] Yuriy Brun, **Kıvanç Muşlu**, Reid Holmes, Michael D. Ernst, and David Notkin. Predicting Development Trajectories to Prevent Collaboration Conflicts. In *the Working Conference on the Future of Collaborative Software Development*, FCSO'12, Seattle, WA, USA, February 2012.
- [15] **Kıvanç Muşlu**, Bilge Soran, and Jochen Wuttke. Finding Bugs by Isolating Unit Tests. In *the 8th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering , New Ideas and Emerging Results Track*, ESEC/FSE'11, pages 496–499, Szeged, Hungary, September 2011.

## Research Experience

---

### Automating Version Control and Improving Historical Analyses Spring'14–Present

Development history is useful for various software engineering tasks, such as finding the cause of regression bugs. Unfortunately, manually managed development histories are incomplete and coarse-grained. I designed a technique that automatically records a complete and fine-grained development history. I showed that this fine-grained history improves the output of the existing historical analyses — analyses that use the development history — and makes it possible to implement new historical analyses.

**Collaborators:** Luke Swart, Yuriy Brun, and Michael D. Ernst

**Publications:** [3]

### Investigating the transition from CVCS to DVCS Summer'13

Why do developers transition from centralized to decentralized version control systems? I found that incremental development via offline commits and context switching via lightweight branches are the major motivations. The study also explains the transition barriers and outcomes.

**Collaborators:** Christian Bird, Nachi Nagappan, and Jacek Czerwonka

**Publications:** [6]

### Continuous Data Testing Spring'13–Present

Data entry errors cost the world economy billions of dollars. I found that extending testing to data in databases could eliminate 60% of data entry errors.

**Collaborators:** Yuriy Brun and Alexandra Meliou

**Publications:** [4] and [11]

### **Realtime Code Analysis through IDE-Managed Code Copies**

**Winter'12–Present**

Continuous analyses automatically compute an up-to-date result without requiring the developer to execute an acting other than editing the source code. These analyses can greatly improve developer productivity, but they are hard to build. I identified two major challenges to building such analyses: currency — how fast an analysis gets access to developer edits — and isolation — ensuring that an ongoing analysis and concurrent developer edits do not conflict with each other. I built Solstice, a framework that makes the implementation of continuous analyses easier by letting them exclusively own a copy codebase and run on it isolated from concurrent developer edits.

**Collaborators:** Yuriy Brun, Michael D. Ernst, and David Notkin

**Publications:** [1] and [7]

### **Speculative Analysis of IDE Recommendations**

**Autumn'10–Autumn'12**

IDEs recommend multiple fixes for each compilation error, but it is difficult for programmers to mentally evaluate them and choose the best one. I showed that computing the effects of these fixes on a program's compilation error can guide the developer to more quickly fix such errors. I built Quick Fix Scout, an Eclipse plug-in to speculatively evaluate the effects of Eclipse quick fixes and recommend the best fixes for compilation errors.

**Collaborators:** Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin

**Publications:** [8] and [13]

### **Beacon, a Proactive Conflict Detector**

**Summer'11**

Collaboration conflicts, such as merge conflicts in a version control system, can be very difficult to resolve and can delay software development. I showed that these conflicts can be detected proactively, before they happen. I implemented BEACON, which warns developers the moment a conflict between two development branches occurs.

**Collaborators:** Judith Bishop, Christian Bird, and Thomas Zimmermann

### **Dependent Tests**

**Spring'11–Present**

Data dependencies between unit tests can mask bugs in the program and in the test suite, and lead to inconsistent documentation. I implemented a framework that runs tests in isolation so that dependencies between tests will not affect test suite results. This approach could have detected a bug in Apache Commons CLI 120 revisions earlier than the developers fixed it.

**Collaborators:** Wing Lam, Bilge Soran, Darioush Jalali, Sai Zhang, Jochen Wuttke, Michael D. Ernst, and David Notkin

**Publications:** [5] and [15]

### **Pluggable Type Checking for Java Naming Conventions**

**Autumn'10–Spring'11**

Java uses different naming conventions to represent the same class in the source code, class files, and byte code. For software that translates between naming conventions, using the correct convention is error-prone. I built a type system and a type-checker that detects misuse of naming conventions. Using the type-checker, I found 14 bugs in JDK 7, and 14 more bugs in other open-source software.

**Collaborators:** Werner Dietl and Michael D. Ernst

**Publications:** [9]

### **Measuring the Effectiveness of a Concurrency Coverage Metric**

**Summer'08**

Sequential code coverage metrics are not correlated well with concurrency bugs. I showed that Keremoğlu's concurrency coverage metric based on conflicting read-write accesses is better correlated with detecting bugs, compared to the Definition-Use and Method Pairs metrics.

**Collaborators:** Erkan Keremoğlu and Serdar Taşiran

**Publications:** [2]

### **Tressa: Asserts in the Future**

**Spring'09**

It is difficult to detect atomicity violations in optimistic concurrent software using assert statements. I showed that Sezgin's tressa theory is effective for detecting such violations.

**Collaborators:** Ali Sezgin, Serdar Taşiran, and Shaz Qadeer

**Publications:** [10]

## Awards

---

SIGSOFT CAPS travel award to attend ICSE.	2012
SIGSOFT CAPS travel award to attend FSE.	2011
Ranked 5 <sup>th</sup> out of over 1.8 million students in the Turkish University Entrance Exam.	2005
Full merit scholarships from Vehbi Koç Foundation, İş Bank, and Turkish Government.	2005

## Teaching Experience

---

<b>Teaching Assistant:</b> Software Engineering,	University of Washington	<b>Spring'12</b>
<b>Teaching Assistant:</b> Concurrency,	University of Washington	<b>Spring'11</b>
<b>Teaching Assistant:</b> Parallel Programming,	Koç University	<b>Fall'09</b>
<b>Tutor:</b> Introduction to Programming (Java),	Koç University	<b>Spring'08</b>
<b>Lab Assistant:</b> Structure and Interpretation of Computer Programs,	Koç University	<b>2006–2007</b>

## Formal Presentations

---

### Conference Talks

- CT1 Transition from Centralized to Distributed Version Control Systems: A Case Study on Reasons, Barriers, and Outcomes. *At the 36th International Conference on Software Engineering*, Hyderabad, India, June 4, 2014.
- CT2 Which Configuration Option Should I Change? (Presented on behalf of Sai Zhang). *At the 36th International Conference on Software Engineering*, Hyderabad, India, June 4, 2014.
- CT3 Making Offline Analyses Continuous. *At the 9th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering*, Saint Petersburg, Russia, August 22, 2013.
- CT4 Data Debugging with Continuous Testing. *At the 9th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering, New Ideas Track*, Saint Petersburg, Russia, August 22, 2013.
- CT5 Enhancing Software Development with IDE-managed Codebases. *At the 35th International Conference on Software Engineering*, San Francisco, CA, USA, May 21, 2013.
- CT6 Speculative Analysis of Integrated Development Environment Recommendations. *At the 3rd Conference on Object-Oriented Programming Systems, Languages, and Applications*, Tucson, AZ, USA, October 24, 2012.
- CT7 Improving IDE Recommendations by Considering Global Implications of Existing Recommendations. *At the 34th International Conference on Software Engineering*, Zurich, Switzerland, June 8, 2012.
- CT8 Finding Bugs by Isolating Unit Tests. *At the 8th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering*, Szeged, Hungary, September 8, 2011.

### Conference Posters

- CP1 Finding Bugs by Isolating Unit Tests. *At the 8th joint meeting of the European Software Engineering Conference and the Symposium on the Foundations of Software Engineering*, Szeged, Hungary, September 8, 2011.

### Workshop Posters

- WP1 Predicting Development Trajectories to Prevent Collaboration Conflicts. *At the Working Conference on the Future of Collaborative Software Development*, Seattle, WA, USA, February 12, 2012.

### Invited Talks

- IT1 Improving Software Development Techniques. *At Microsoft Research Seminar*, Redmond, WA, USA, March 30, 2015.
- IT2 Improving Software Development Techniques. *At North Carolina State University Computer Science Seminar*, Raleigh, NC, USA, March 17, 2015.

- IT3 Improving Software Development Techniques. At *Virginia Tech Computer Science Seminar*, Blacksburg, VA, USA, March 2, 2015.
- IT4 Improving Software Development Techniques. At *Koç University Engineering Seminar*, Istanbul, Turkey, January 6, 2015.
- IT5 Improving Software Development Techniques. At *Boğaziçi University Computer Engineering Seminar*, Istanbul, Turkey, January 5, 2015.
- IT6 Making Offline Analyses Continuous. At *Bilkent University Computer Engineering Seminar*, Ankara, Turkey, July 9, 2014.
- IT7 Making Offline Analyses Continuous. At *Koç University Engineering Seminar*, Istanbul, Turkey, July 4, 2014.
- IT8 Integrating Systematic Exploration, Analysis, and Maintenance in Software Development. At *Koç University Engineering Seminar*, Istanbul, Turkey, December 18, 2012.

---

## Service

### Program Committee Member

- M1 International Conference on Software Analysis, Evolution, and Reengineering, Industry Track Committee, 2016.
- M2 International Symposium on Software Testing and Analysis, Artifact Evaluation Committee, 2014.

### Reviewer

- R1 Transactions on Software Engineering, 2015
- R2 Journal of Object Technology, 2015
- R3 Automated Software Engineering, 2014
- R4 Transactions on Software Engineering, 2013
- R5 International Symposium on Network Computing and Applications, 2012
- R6 Object-Oriented Programming Systems, Languages and Applications, 2011
- R7 International Conference on Software Engineering, 2011

---

## Membership

ACM and ACM SIGSOFT: 2011–present  
IEEE: 2014–present

---

## Advising

### Undergraduate Advising

- U1 Christopher Chen, Autumn 2014–Present: Christopher is building infrastructure that makes it easy to properly detect compound edit operations from the IDE. His focus is undo/redo operations.
- U2 MengQiao Han, Autumn 2014–Present: MengQiao is working with Christopher, focusing on refactorings.
- U3 Alain Orbino, Winter 2013–Summer 2014: Alain worked on a test mocking framework for a project that uses localhost network for inter-process communication. He mocked the server side of the framework to make it easier to test the client code.
- U4 Luke Swart, Summer 2013–Summer 2014: Luke worked on an interactive analysis that helps the developer to rewrite part of the development history into cohesive commits representing high-level software engineering tasks. Luke also worked on the visualization of continuous analysis plug-ins through Eclipse markers and problems views.
- U5 Riley Klinger, Summer 2013: Riley worked on a continuous mutation testing analysis.
- U6 Chuong Thanh Dao, Winter 2012: Chuong worked on visualizing speculative analysis results for quick fixes.

---

## Professional Experience

**Facebook, Menlo Park, CA, USA**

**Summer'12**

It is difficult to detect the cause of such non-bisectable test failures. I implemented trace serialization and analysis

to better understand such failures.

**Byte Company, Turkey**

**August'09**

I implemented two client-side applets and one server-side daemon. These applets enable customers to see network logs in real time.

**References**

---

Michael D. Ernst, Professor, University of Washington

mernst@cs.washington.edu

Yuriy Brun, Assistant Professor, University of Massachusetts Amherst

brun@cs.umass.edu

Nachiappan Nagappan, Principal Researcher, Microsoft Research

nachin@microsoft.com

Christian Bird, Researcher, Microsoft Research

christian.bird@microsoft.com