The Effects of Compilation Mechanisms and Error Message Presentation on Novice Programmer Behavior

IOANNIS KARVELAS
UNIVERSITY COLLEGE DUBLIN

ANNIE LI
UNIVERSITY OF MICHIGAN

BRETT A. BECKER
UNIVERSITY COLLEGE DUBLIN

IOANNIS.KARVELAS@UCDCONNECT.IE
ANNIELI@UMICH.EDU
BRETT.BECKER@UCD.IE

Motivation

- “Computer programming could be made easier”, Journal of Occupational Psychology - 1977 paper by Sime, Arblaster and Green
- Many variables of the programming process in CS1 have been explored, such as:
  - Teaching approaches
  - Programming languages
  - Features in programming environments including level of IDE assistance, simplicity, graphic output, support for collaboration.

Motivation

- Typical university-level introductory programming courses revolve around 3 pillars:
  - Theory (fundamentals of programming) through…
  - Practice (coding assignments) through…
  - Utilizing an environment, editor or Integrated Development Environment (IDE)!
- Students spend substantial amount of time exposed to the environment. As a result, learning how to work with it becomes part of the learning process.
- Many “industry-strength” environments are armed with a wide variety of complex features that are aimed to support professional developers. **It is uncertain if these features are beneficial to novices.**

Aim & Contribution

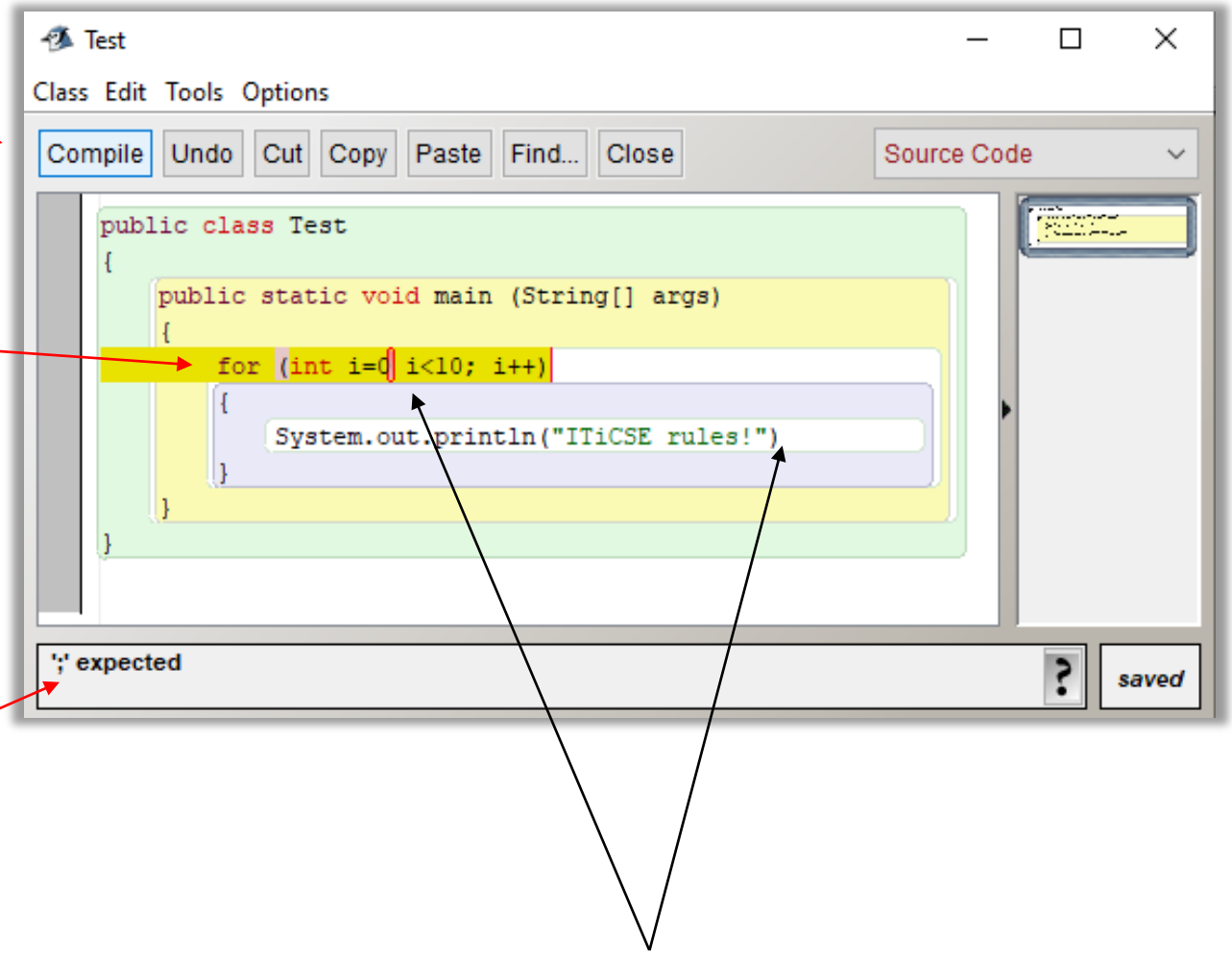
- Investigate how (1) compilation mechanism and (2) compiler error message presentation in the editor affect programming behavior.
- We used Blackbox Programming Process Data (PPD) from BlueJ users for the analysis
- Provide reliable information on the effects of changes in a programming.

BlueJ 3

only manual compilation

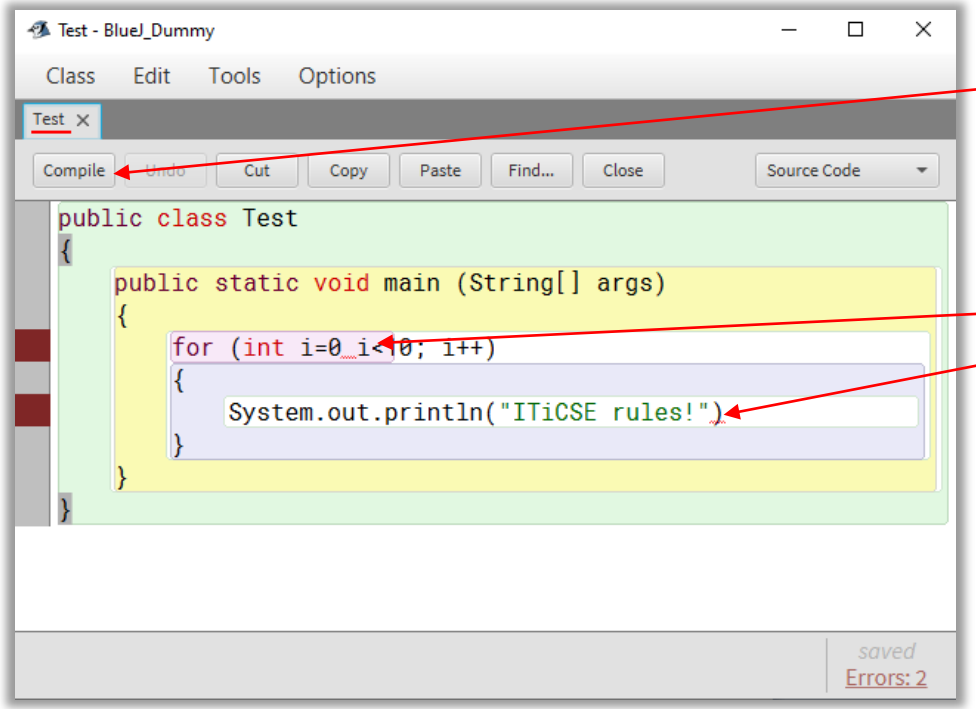
code mapped to the first highlighted error

only first error message is displayed



2 errors in the code

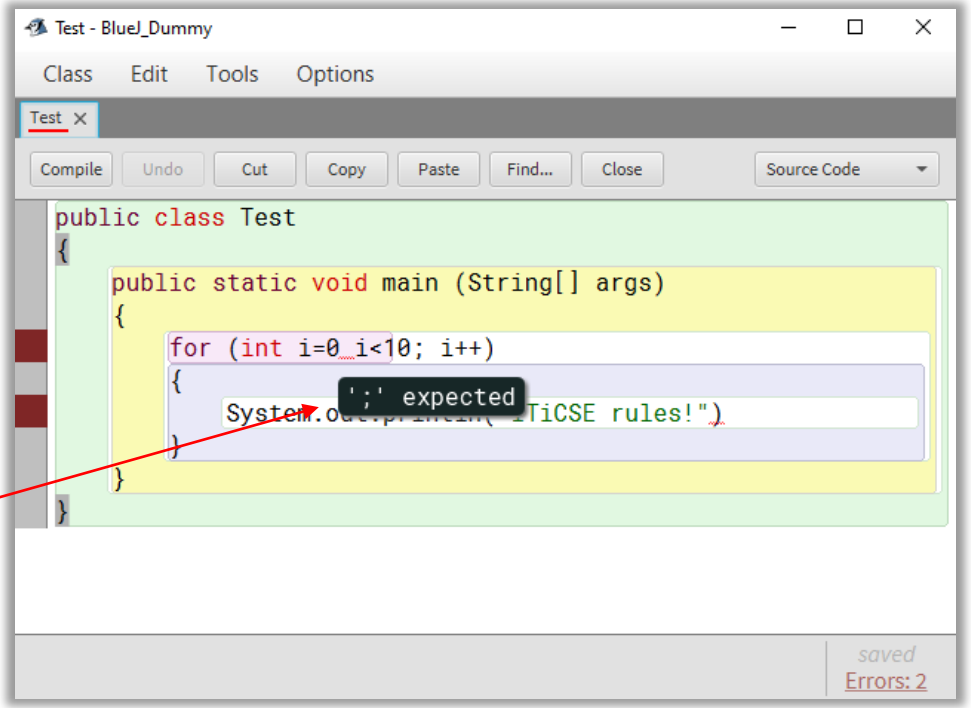
BlueJ 4



manual compilation still operational

all code lines indicated
all errors underlined

no error message displayed by default



users have to hover over the offending code by using mouse/keyboard or manually compile to cycle through all errors.

Motivating Questions

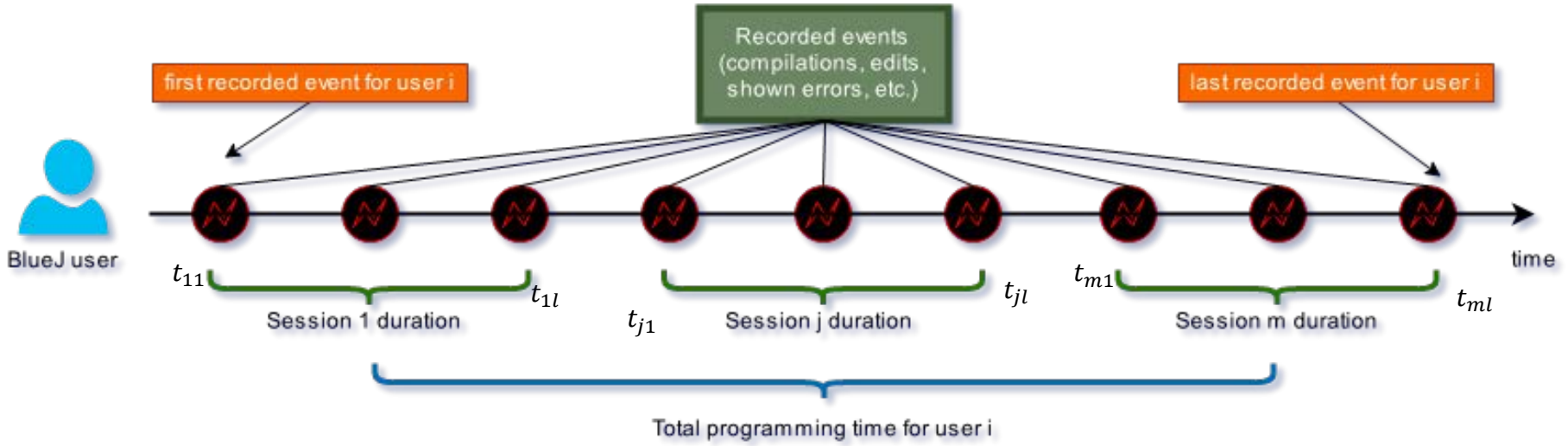
- Q1: How did changes in BlueJ affect the frequency of error messages presented to users?
- Q2: How did changes in BlueJ affect the frequency of manual compilations? *(In BlueJ 3, users have to manually click to compile, but in BlueJ 4 this is possible but not required.)*
- Q3: How did changes in BlueJ affect the percentage of successful manual compilations?
- Q4: How are these results affected by different choices of heuristics for calculating programming session time?

Methodology – Data Filtering

- Cohort of 3176 users who used BlueJ 3 and BlueJ 4 (switched versions between 1st of October 2017 and 31st of January 2018)
- ≈1/3 of users removed because of atypical behavior (writing code, but never compiling etc.)
- Remaining user number: 2062
- Reduced data range to 14th of January 2016 – 24th of May 2019 (the first date BlueJ 4 is used is equidistant to start and end dates of range)
- Gathered information for every user:
  - BlueJ and Java versions (only Java 8 events included for consistency in error numbers)
  - compilation events
  - reason of compilation
  - state of success
  - error messages generated by the compiler and if they were presented to users
  - timing information on sessions and events

Calculating Programming Time

-Blackbox captures a 'begin_session' and an 'end_session' event
-A large number of sessions had no end session
-This is due to technical issues such as internet connection drop-outs



Total programming time of all users: $T = \sum_{i=1}^{n} t_i = \sum_{i=1}^{n} \sum_{j=1}^{m} (t_{jl} - t_{j1})$

Metrics for Measuring Interactions

- Compilations* per Hour (CpH)
- Percentage of Successful Compilations* (PSC)
- Displayed Compiler Error Messages per Hour (DCEMpH)

Profiling of every user for each BlueJ version

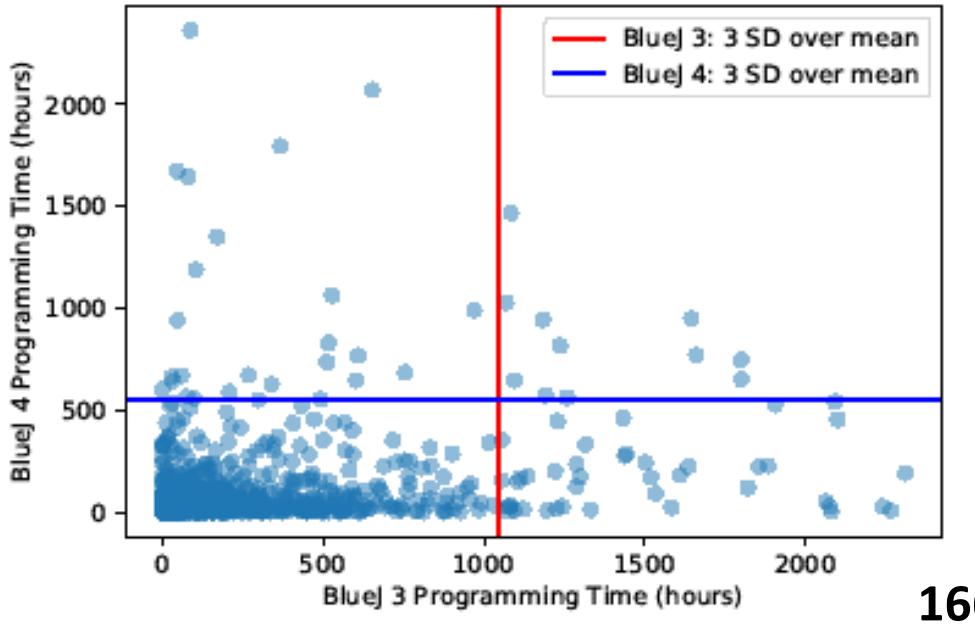
(+ Global calculation, see paper for more)

*manual for BlueJ 4

Removing Outliers

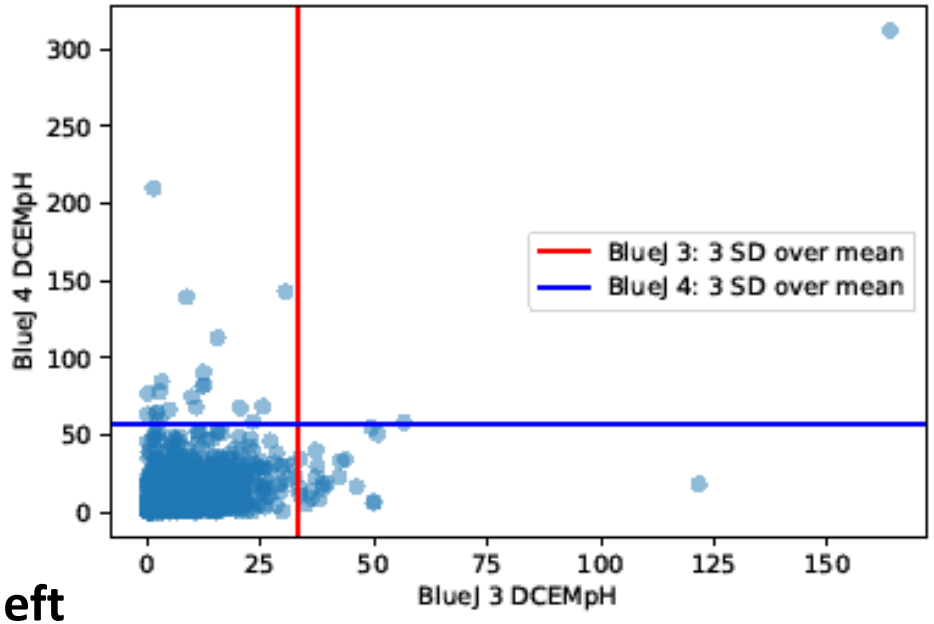
Removed all users who had prog. time> M + 3*SD

BlueJ 3: M=144.8, SD=301
BlueJ 4: M=58.7, SD=163.1



Removed all users who had DCEMpH> M + 3*SD

BlueJ 3: M=7.3, SD=8.7
BlueJ 4: M=11.1, SD=15.5



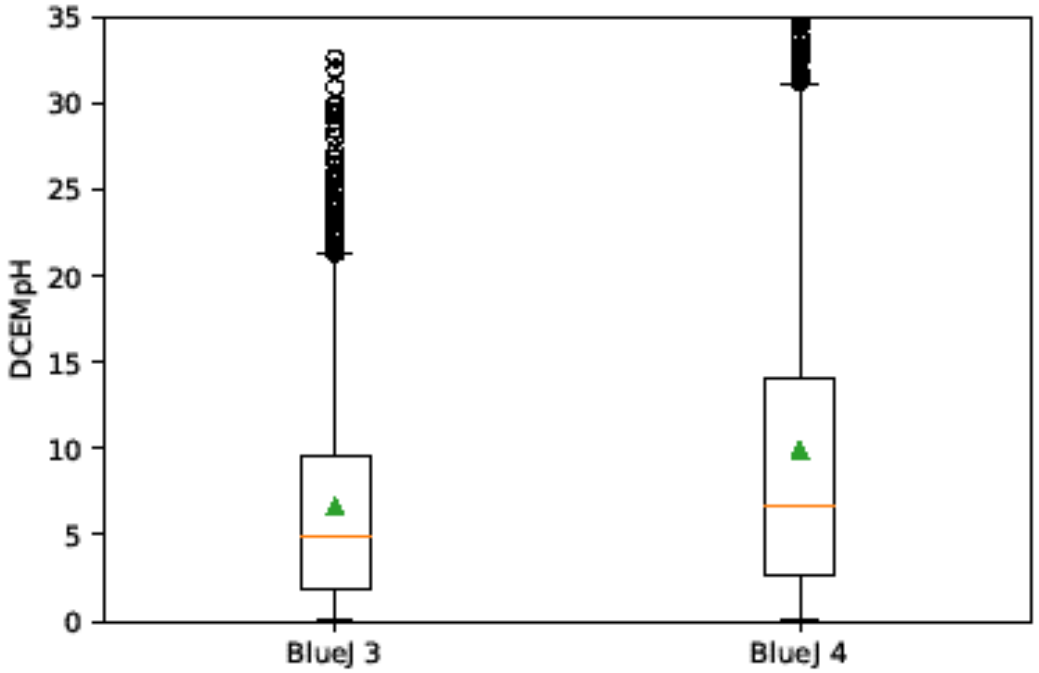
1661 users left

Trimming Inactive Intervals

- Some users tend to leave BlueJ open while away from PC.
- We recalculated DCEMpH by treating intervals between events as inactive at thresholds of: 1, 5, 10, 15, 30, 60, 90 minutes.

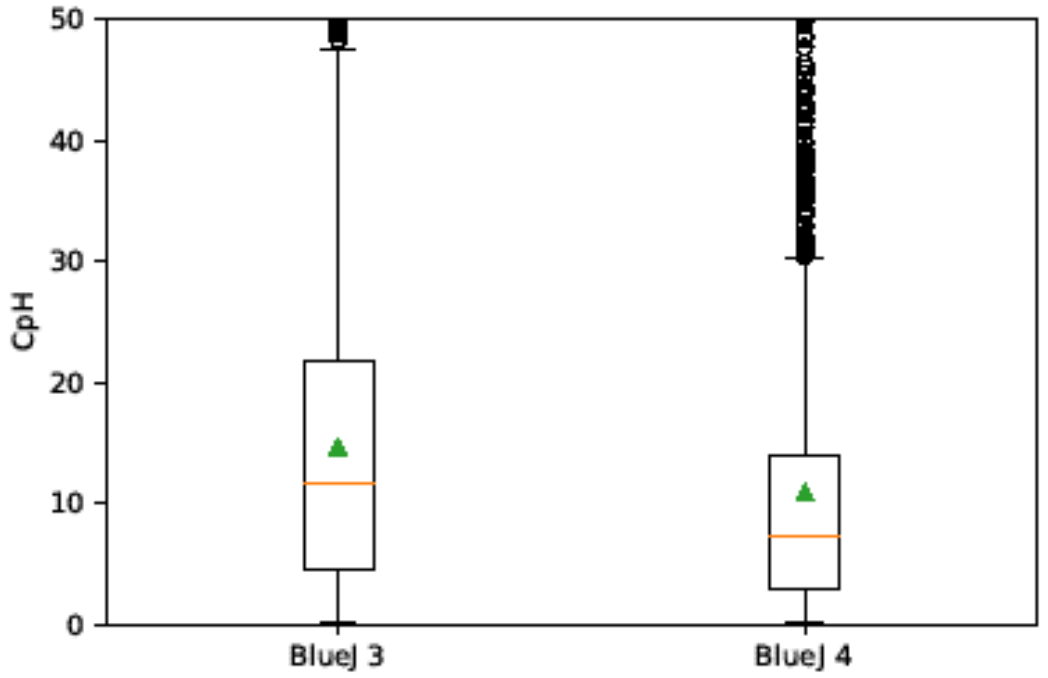
Results Q1: How did changes in BlueJ affect the frequency of error messages presented to users?

- Increase in BlueJ 4 (M=9.7) compared to BlueJ3 (M=6.6)



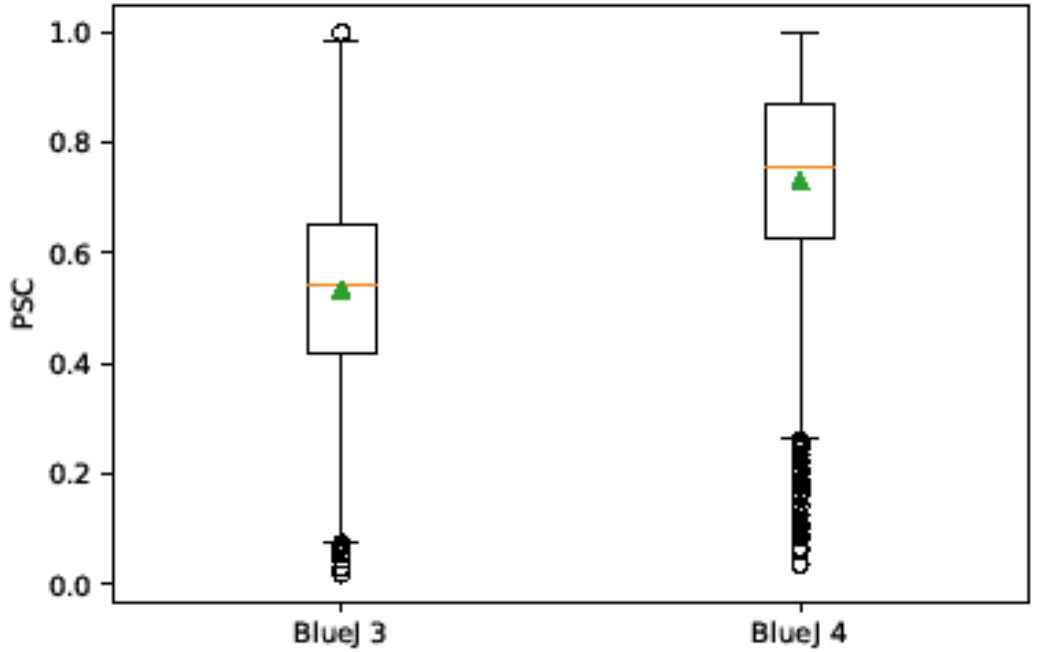
Results Q2: How did changes in BlueJ affect the frequency of manual compilations?

- Decrease in BlueJ 4 (M=10.9) compared to BlueJ3 (M=14.6)



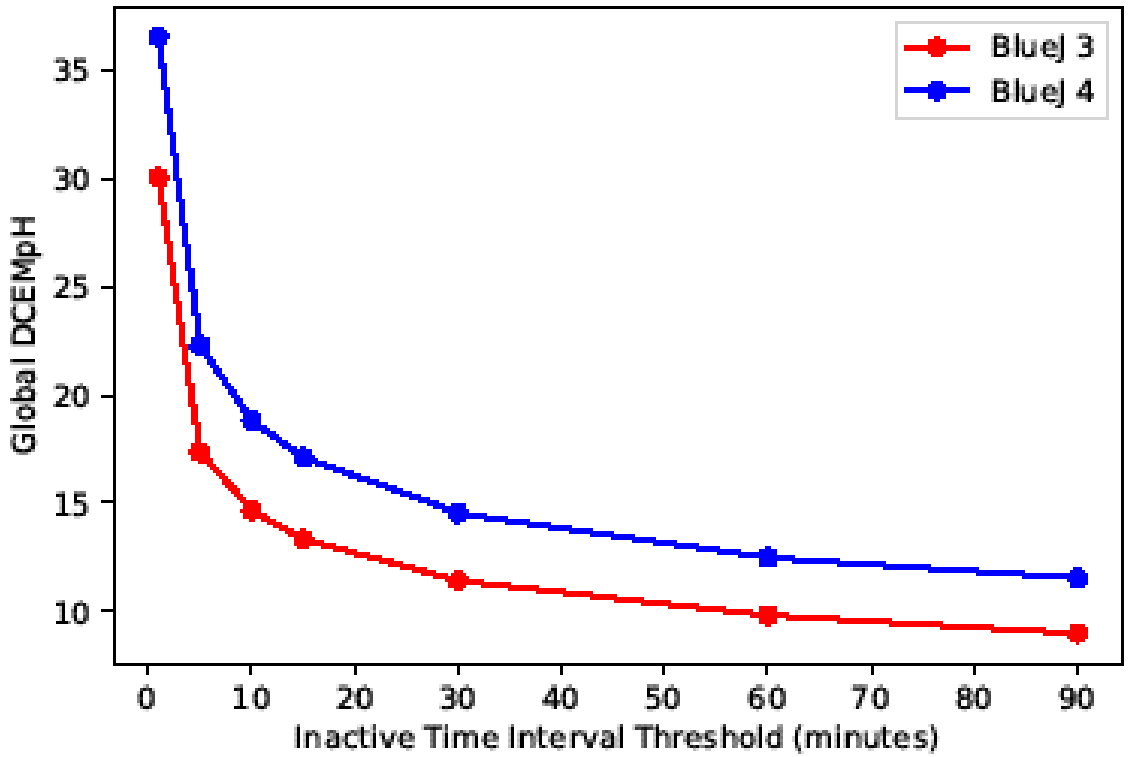
Results Q3: How did changes in BlueJ affect the percentage of successful manual compilations?

- Increase in BlueJ 4 (M=0.7) compared to BlueJ3 (M=0.5)



Results Q4: How are these results affected by different choices of heuristics for calculating programming session time?

- Filtering algorithm for treating different time intervals as inactive and global calculation method (more on paper) align with the results and reinforce the findings



Threats to Validity

- No contextual information for users (all Blackbox studies share that)
- Inadvertently triggered displayed compiler error messages
- Experience due to exposure to another BlueJ version may affect behaviour
- No ‘end session’ – missing data

Conclusions

- Novices see more error messages in BlueJ 4, likely because they feel they need to.
  - Further qualitative study would be needed for evaluation…
- Users compile manually less frequently in BlueJ 4, since automatic compilation is also present.
  - This is interesting, because technically they don't have to manually compile anymore
- The majority of manual compilations in BlueJ 4 are successful.
  - This could mean that manual compilation acts as reassurance. If no errors are underlined, they click compile to be sure

Future Work

- Repeat experiments with cohorts exposed to a single BlueJ version
- Further exploration on high percentages of success in manual compilations