NC STATE UNIVERSITY

Exploring Tools and Strategies Used During Regular Expression Composition Tasks

Gina R. Bai∗ Brian Clee∗ Nischal Shrestha∗
Carl Chapman† Cimone Wright† **Kathryn T. Stolee∗**

∗ North Carolina State University, USA
† Iowa State University, USA

May 26, 2019

Application of Regular Expression

Search
ERROR-PRONE!
Text Editor Symbols | 40 icons
FLATICON
NETWORK SECURITY
Mobile Networks | Web Applications | Cloud | Servers

Prior Work on Regular Expression

- ❖ Regex Usage and Language Features [Chapman, 2016]
- ❖ Regex Comprehension [Chapman, 2017]
- ❖ Regex Testing [Wang, 2018]
- ❖ Regex Evolution [Wang, 2019]

- ❖ Web tools
  - ➢ Dynamic testing
  - ➢ Visualization
- ❖ Educational games

In this study:
1) Users
2) Screen-captured Videos

What We Wish to Learn…

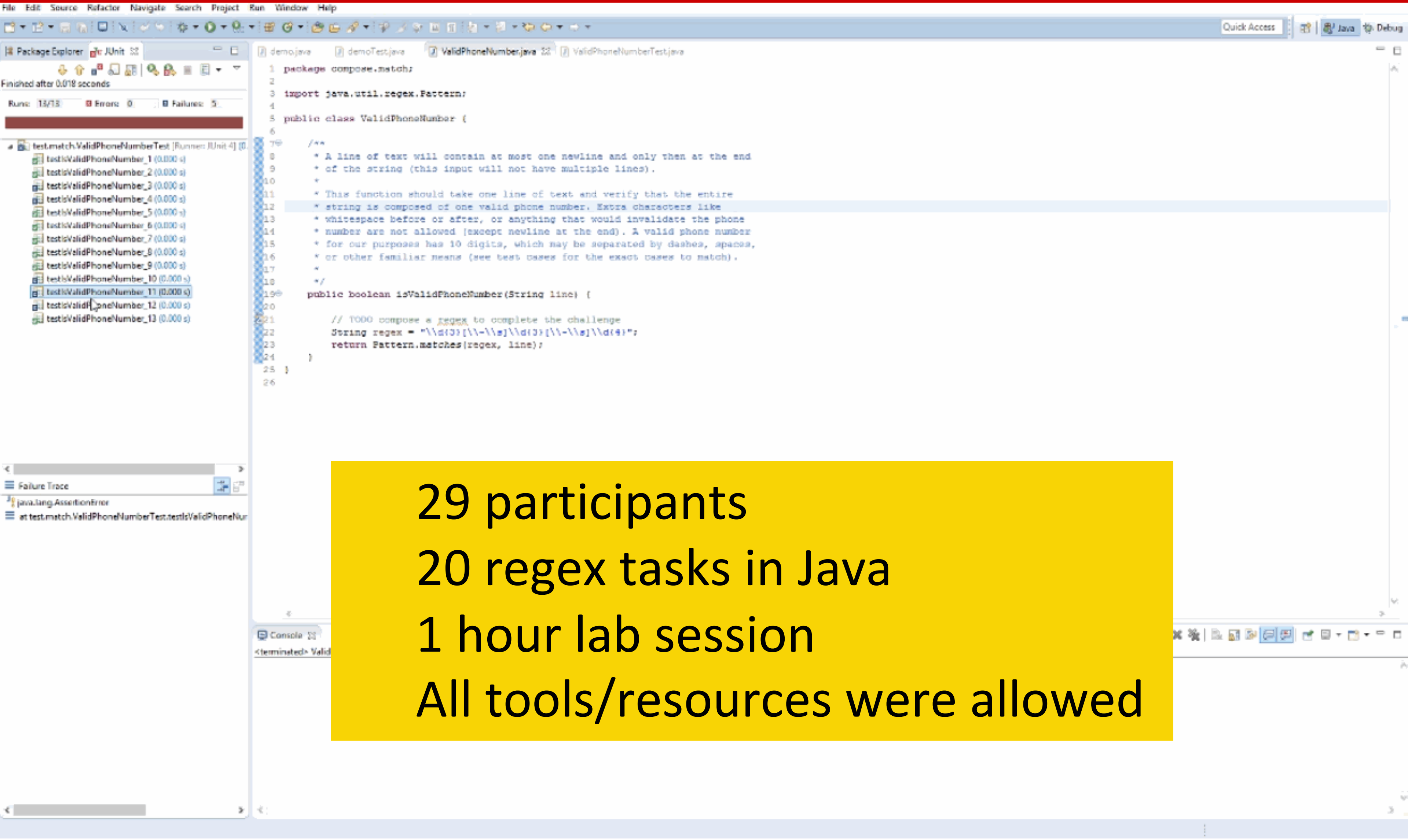
Tools and strategies

- Visualization on regex [Beck, 2014]
- Search [Singer, 1997], [Brandt, 2010]

Behavioral routine

- Persona [Stylos, 2007]

*Images from Google.com



Finished after 0.018 seconds

Runs: 13/13 Errors: 0 Failures: 5

```
test.match.ValidPhoneNumberTest [Runner: JUnit 4]
    testIsValidPhoneNumber_1 (0.000 s)
    testIsValidPhoneNumber_2 (0.000 s)
    testIsValidPhoneNumber_3 (0.000 s)
    testIsValidPhoneNumber_4 (0.000 s)
    testIsValidPhoneNumber_5 (0.000 s)
    testIsValidPhoneNumber_6 (0.000 s)
    testIsValidPhoneNumber_7 (0.000 s)
    testIsValidPhoneNumber_8 (0.000 s)
    testIsValidPhoneNumber_9 (0.000 s)
    testIsValidPhoneNumber_10 (0.000 s)
    testIsValidPhoneNumber_11 (0.000 s)
    testIsValidPhoneNumber_12 (0.000 s)
    testIsValidPhoneNumber_13 (0.000 s)
```

Failure Trace

```
java.lang.AssertionError
at test.match.ValidPhoneNumberTest.testIsValidPhoneNur
```

```java
package compose.match;

import java.util.regex.Pattern;

public class ValidPhoneNumber {

    /**
     * A line of text will contain at most one newline and only then at the end
     * of the string (this input will not have multiple lines).
     *
     * This function should take one line of text and verify that the entire
     * string is composed of one valid phone number. Extra characters like
     * whitespace before or after, or anything that would invalidate the phone
     * number are not allowed (except newline at the end). A valid phone number
     * for our purposes has 10 digits, which may be separated by dashes, spaces,
     * or other familiar means (see test cases for the exact cases to match).
     *
     */
    public boolean isValidPhoneNumber(String line) {

        // TODO compose a regex to complete the challenge
        String regex = "\\d{3}[\\-\\s]\\d{3}[\\-\\s]\\d{4}";
        return Pattern.matches(regex, line);
    }
}

```

Console

<terminated> Valid

29 participants

20 regex tasks in Java

1 hour lab session

All tools/resources were allowed

Running Example – Task

Task Description

```java
package compose.match;

import java.util.regex.Pattern;

public class ValidEmail {

        /**
         * A line of text will contain at most one newline and only then at the end
         * of the string (this input will not have multiple lines).
         *
         * This function should take one line of text and verify that the entire
         * string is composed of one valid email. Extra characters like whitespace
         * before or after, or anything that would invalidate the email are not
         * allowed (except newline at the end).
         *
         */
        public boolean isValidEmail(String line) {

                // TODO compose a regex to complete the challenge
                String regex = "                              ";
                return Pattern.matches(regex, line);
        }
}
```

Blank to fill in

Running Exam

```
public class ValidEmailTest
    private static ValidEma
    @BeforeClass
    public static void setu
        validEmail = new Va
    }
    @Test
    public void testIsValic
                                                   #
                                                   w
                                                   r
                                                   a
                                                   tion
    }
    // ... More tests, eight test cases in total
```

Possible Solutions:

1) .+@.+
2) \\S+@\\S+.\\w
3) …

Valid:

- name@domain.com
- 1.2.3.4@crazy.domain.axes
- !@B.gone

Invalid:

- @tweetybirdHandle
- www.website.com
- oneWord
- Look at that lightning storm - it's getting closer!

Running Example – Attempt & Logs

[Ko, 2006] & [Snipes, 2015]

1. 0:00:24, Opened ValidEmail task in Eclipse IDE, and started to compose
2. 0:00:36, Switched to browser and visited google.com
3. 0:00:40, Searched “valid email in regex Java”
4. 0:00:44, Accessed the StackOverflow result

Time	Search	WebsiteVisited	Eclipse	Task
0:00:24			T	ValidEmail
0:00:36		Google		
0:00:40	Valid email in regex Java			
0:00:44		StackOverflow		

Running Example – Attempt & Logs

1. …
2. 0:02:12, C expression X
3. 0:02:14, S
4. 0:02:18, F
5. 0:02:25, F passed #1 and #4 JUnit tests (among 8)

Attempt:

Opens a task

Runs tests at least once

Leaves this task

Time	Search	Website Visited	Eclipse	Copy Paste	Regex String	Task	Test Passed	Pass Rate
0:02:12		StackOverflow		regex				
0:02:14			T					
0:02:18			T	regex	X			
0:02:25			T				1,4	2/8

Running Example – Metrics

Time	Task	Test Passed	Pass Rate
Time 1	ValidEmail		
Time 2		1,4	2/8 = 25%
Time 3		1,2,4,5,7,8	6/8 = 75%
Time 4		1,2,3,5	4/8 = 50%
Time 5	NoVowelsWord		
Time 6		none	0/5 = 0%
Time 7		1,3	2/5 = 40%
Time 8		1,2,3,4,5	5/5 = 100%

#AvgTestRun = 3

Avg First-time PR
= (25% + 0%) / 2
= 12.5%

Avg Pass Rate
= (75% + 100%) / 2
= 87.5%

Avg Improved PR
= (50% + 100%) / 2
= 75%

Overview of Transcribed Data

- ❑ 11 trigger events, 12 columns & 11,644 rows logged
- ❑ 121 tasks viewed
- ❑ **94 of them tested** (attempts used for analysis)
  - ❑ **28 attempts achieved 100%**
  - ❑ **avgPassRate: 56%**
- ❑ 1,097 total web searches
- ❑ 3,401 websites visited
- ❑ 230 copy-and-paste

Research Question 1

What **tools** and **strategies** do developers employ while solving regular expression tasks in the Eclipse IDE?

RQ1: Tools – Debugger vs. Web Tools

Eclipse built-in debugger

- AvgPassRate: 48%
- No improvement

Web tools

- Higher AvgPassRate than Non web tool attempts (68% vs 54.6%)
- Involved in 10/28 successful attempts
- 7/9 participants passed more test cases

RQ1: Strategies – Search Online

[Ragkhitwetsagul, 2019]

Online Sources & Average Improvement in Pass Rate & Average Pass Rate

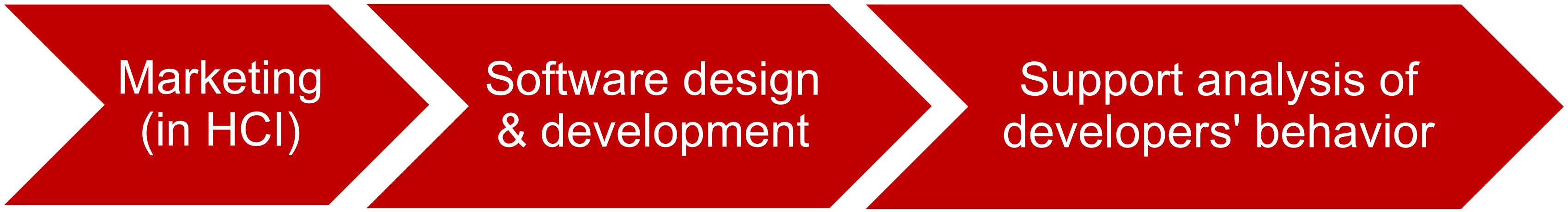
Online Sources	avgImp	avgPassRate	# Attempt
Q&A sites only	24%	50%	7
Documentations & Tutorials sites only	35%	62%	13
Both Q&A and D&T sites	31%	51%	57
None	29%	70%	17
Total			**94**

RQ-1: Strategies – Reusing

- 33/94 attempts involved Copy&Paste
  - Avg pass rate: 45%, (vs non-C&P: 62%)
  - Avg improvement: 27%
- 36.3% C&P from web to IDE tested directly
- 57.7% C&P from web to IDE modified before being tested
  - 29/80 → Correct compile error
  - e.g. modify \w+ to \\w+

Research Question 2

Which **personas** emerge as representative of the problem solving strategies exhibited by the developers?


Marketing (in HCI)
Software design & development
Support analysis of developers' behavior

[Mikkelson, 2000]

[Cooper, 2004]
[Schneidewind, 2012]
[Anvari, 2015]

[Stylos, 2007]
[Dubey, 2017]
[Ford, 2017]

RQ2: Personas – Metrics

Learning Progress

- RegexExperience
- JavaExperience
- PassRate (on each)
- AvgPassRate (among all)
- FirstTimePassRate
- **AvgFirstTimePassRate**
- PassRate-EachTestRun
- ImprovedPassRate

Prior knowledge

- **AvgImpPassRate**
- #TaskAttempted
- #TestRun
- **#AvgTestRun**
- #Search
- #C&P
- #WebsitesVisit
- #StackOverflowVisit
- #DocumentationVisit

Testing behavior

Persona Vector

<AFPR, AIPR, ATR>

RQ2: Personas – Identification

Novice Tester (7/29)

Intermediate (9/29)

Knowledgeable Tester (5/29)

Knowledgeable (8/29)

RQ2: Personas – Stats Summary (Partial)

Knowledgeable

- **Highest average Pass Rate**
  - 63.5% vs 56% for all
- **Lowest average Q&A site visits**
  - 1.8 vs 8.9 for all
- **Lowest average Docs site visits**
  - 1.1 vs 4.5 for all
- Search with **specific** keywords

Main Findings

- **Web tools** with visualization of regexes are helpful
  - Testers & Novices!
- **Consulting official documentation and tutorials** is more beneficial than Q&A sites
- 4 personas
  - The most frequent persona was *Intermediate*

We Suggest Tool Developers…

- ❖ Visualization + Documentation search tool in IDE
- ❖ Support more languages and language migration
    - ➢ **compile errors**

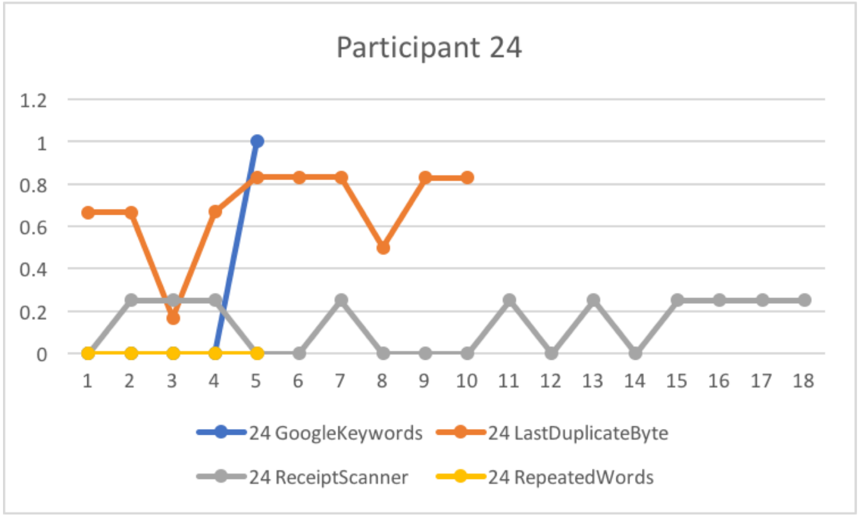
Future work

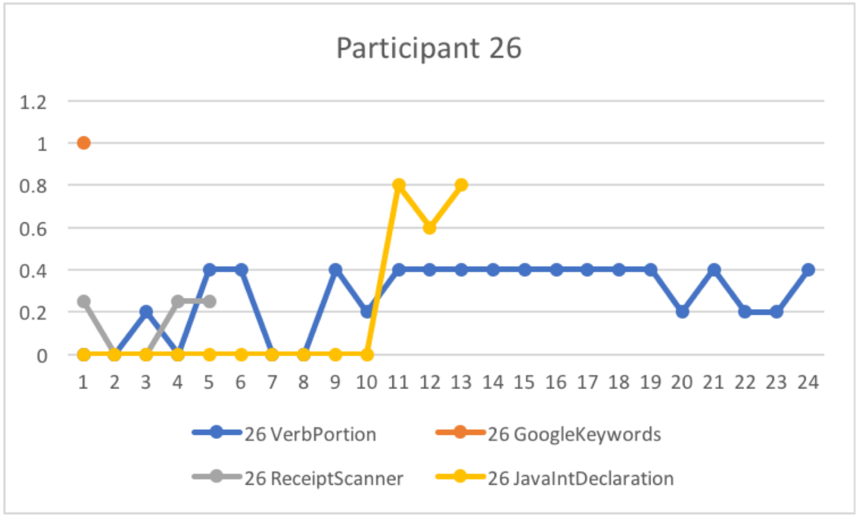
❖ Replication studies:
  ➢ **Think-aloud**
  ➢ More diverse set of **professional developers**

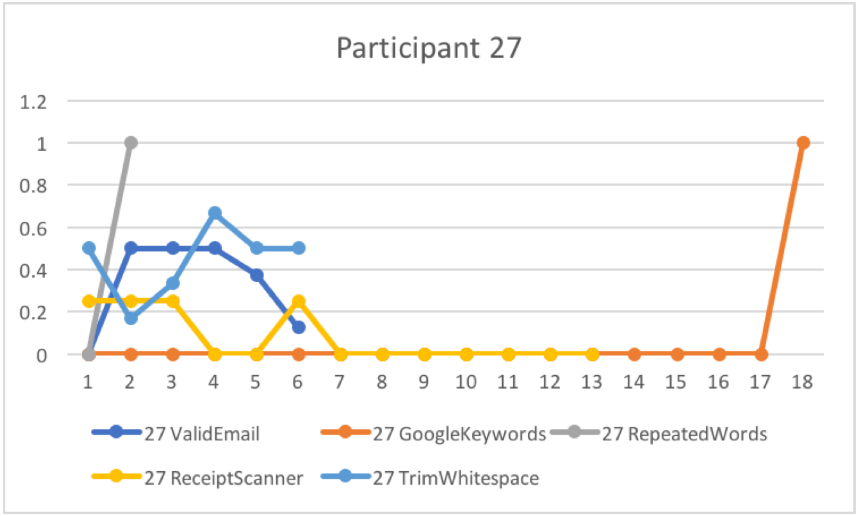
❖ Explore the **technical mistakes** made during regular expression composition

Gina Bai
< rbai2@ncsu.edu >
< https://ginabai.github.io >

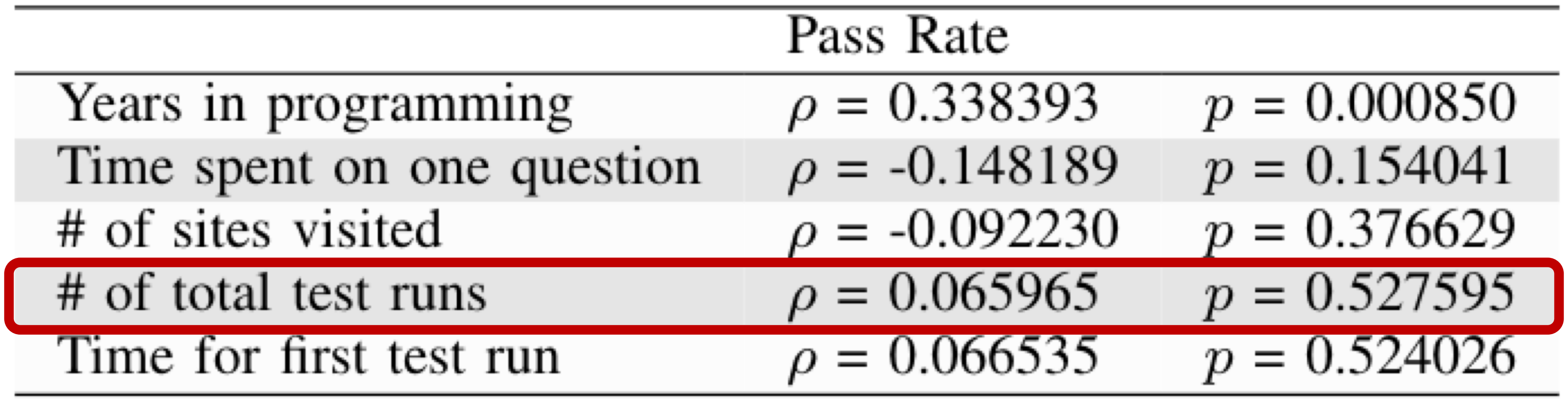
Micro-Progress Pattern
- Backup Slides





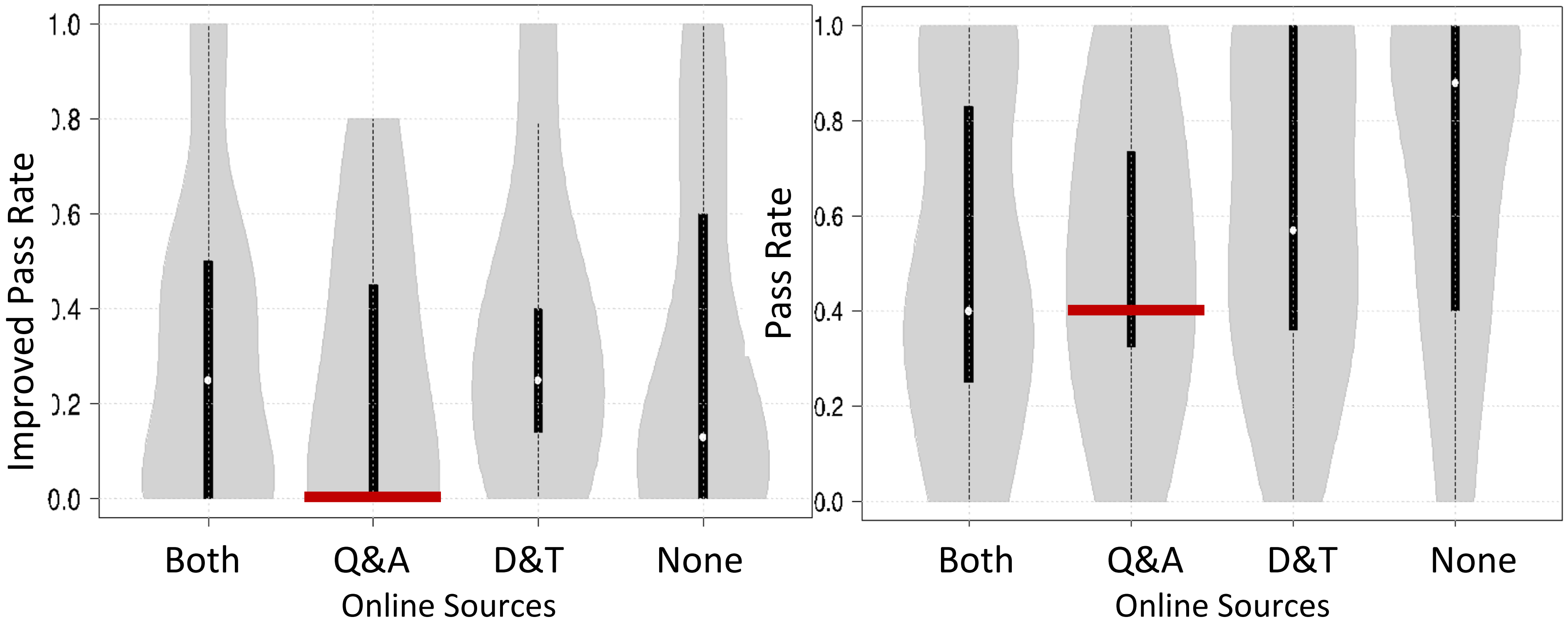


Spearman Correlation on Factors
- Backup Slides



	Pass Rate	
Years in programming	ρ = 0.338393	p = 0.000850
Time spent on one question	ρ = -0.148189	p = 0.154041
# of sites visited	ρ = -0.092230	p = 0.376629
# of total test runs	ρ = 0.065965	p = 0.527595
Time for first test run	ρ = 0.066535	p = 0.524026

Strategies – Search
- Backup Slides



Pass rates & pass rate improvements for attempts that access various online sources

RQ-2: Personas – Stats Summary (Partial) - Backup Slides

Persona Statistical Summary

Persona	AvgPassRate	AvgSearch	AvgC&P	AvgStack	AvgDocs
Novice Tester	44.1%	7.7	1.4	9.7	8.1
Knowledgeable Tester	58.8%	11.0	3.0	15.3	4.4
Knowledgeable	**63.5%**	7.0	4.8	**1.8**	**1.1**
Intermediate	37.7%	11.9	4.7	10.9	4.7