Confirmation Bias as a Human Aspect in Software Engineering

Gul Calikli, PhD

Ayse Bener, PhD

gcalikli@ryerson.ca, ayse.bener@ryerson.ca

Data Science Laboratory, Department of Mechanical and Industrial Engineering, Ryerson University
PART I : Data Science Lab
About DSL

Data to Knowledge – Multi Scale Approaches to Complex and Big Data
Transforming data, joining disciplines
Modeling, analysis, learning and knowledge extraction
Data complexity
Design of intelligent algorithms
Development of decision making models
Diverse domains- business intelligence and smart analytics
Software
Environmental
Engineering
Biology
About DSL

- People
  - 3 faculty members from industrial engineering, computer science and information systems
  - Postdoctoral research fellow
  - 3 PhD students, 4 MSc students
  - Expanding to hire
    - 6 – 8 more graduate students
About DSL

Alumni Known to Microsoft Research
About DSL

Previous Projects and Industry Collaborations
About DSL

Current Collaborations and Projects

IBM in Canada
The IBM Canada Software Labs

IBM Research

SickKids

Lunenfeld

Mount Sinai Hospital
Joseph and Wolf Lebovic Health Complex
Samuel Lunenfeld Research Institute

Ozcelik Lab

Data Science Laboratory
About DSL

Research Areas
Organization & Presentation of Information
(Dr. Ozgur Turetken)
Privacy-Preserving Data Mining & Cloud
(Dr. Ali Miri)

- **Security and Privacy in Clouds**
  - Design of secure de-duplication algorithms for data stored over distributed clouds. Currently developing efficient Proof of Retrieval (POR) and Proof of Data Possession (PDP) algorithms

- **Social Networks**
  - Developed and working on cryptographic algorithms against semi-honest or malicious cloud providers and other cloud users.

- **Privacy-Preserving Data Mining**
  - Constructed a number of privacy-preserving building blocks with applications to data mining, machine learning.

- **Sensor and RFID Systems**
  - Designed a number of efficient and side-channel resistant cryptographic tools for constrained devices, and privacy-preserving authentication protocols.

- **Intrusion and Anomaly Detection System**
  - Built novel adaptive anomaly detection systems using Boolean combinations of Hidden Markov Models
Empirical Software Engineering
(Dr. Ayse Bener)

Problems in Industry
- Decision making under uncertainty and risk management
- Organizing resources efficiently

Our Solution
- Building intelligent oracles
Research 1.0

At the intersection of AI and SWE

New Algorithms/Methods
- Cross company
- TEAK
- ENNA
- CGBR

Product/Process-Related
- Design metrics
- File Dependency Graphs
- Churn Metrics
- Static Code Metrics

Data Size
- under-sampling outperformed over-sampling.
- micro-sampling

Data Content

Algorithms
- k-NN
- Naïve Bayes
- Bayesian Networks
- Neural Networks
- SVM
- Logistic Regression
- ...........
- ...........

How can we enhance the performance of defect prediction models?
Research 2.0

Vision:
- Using Human Aspects to Build Software Prediction Models
- Tool support

Data Size
- under-sampling outperformed over-sampling.
- micro-sampling

Data Content
- People-related
  - Organizational metrics
  - # of developers
  - Developer experience
  - Social Interaction

- Product/Process-Related
  - Design metrics
  - File Dependency
  - Graphs
  - Churn Metrics
  - Static Code Metrics
  - CGBR

Tool Support

Algorithms
- k-NN
- Naïve Bayes
- Bayesian Networks
- Neural Networks
- SVM
- Logistic Regression
- ……
A tool developed by DSL (Softlab)

- Parser
  - C, Java, C++, jsp, PL/SQL
- Static Code Metric Collection
- Data Analysis
- Automated Defect Prediction
- Code.google

---


Research 2.0: Tool Support

Misirli, A.T., B. Caglayan, G. Calikli, A. Bener, T. Aytac, and B. Turhan,
“Dione: An Integrated Measurement and Defect Prediction”, FSE-20
2012, November 11-16, Cary, North Carolina, USA.
Research 2.0: Human Aspects in Defect Prediction Models

- People as part of the team
- People as individuals

[Diagram of human aspects affecting software reliability]

Set of Human Aspects

Human Aspects affecting software reliability

Social interactions

1. Set of Human Aspects
2a. Individual human aspects (Model by Cognitive Science, Cognitive Psychology)
2b. Social interactions
3a. Quantification/Measurement of cognitive biases (Cognitive Bias Metrics)
3b. Social Network Metrics (Social Networks - Graph Theory)
Research 2.0: People as Part of a Team

**WHO** will fix this?

**WHERE** is the problem?
PART II:
Confirmation Bias Research
Why Human Aspects in Software Engineering?

- Enhance decision making under uncertainty, so that managers can take decisions about efficient allocation of resources during any phase of the SDLC.
  - Which parts of software should be prioritized for testing?
  - Who should test/develop the most critical parts of software?
  - Who should fix the bugs in the most problematic parts of the software?
  - Who should/should not develop/maintain the same source files?
  - Who should we hire as a developer/tester/analyst/designer?
Why Human Aspects in Software Engineering?

- People’s thought processes have a significant impact on software quality as software is analyzed, designed, tested, developed and managed by people.

- While solving problems in daily life people use heuristics to solve problems. When heuristics fail to produce a correct judgment, it results in a cognitive bias.

  - Heuristics employed in daily software engineering activities may also result in cognitive biases, leading to defects.

- Some common cognitive bias types:
  - confirmation bias
  - anchoring and adjustment
  - availability
  - representativeness.

We focus on confirmation bias!
Confirmation Bias in Software Engineering

- **Confirmation bias** is defined as the tendency of people to seek evidence to verify a hypothesis rather than seeking evidence to refute that hypothesis.
Confirmation Bias in Software Engineering

Why confirmation bias?

- Empirical evidence about the existence of confirmation bias among software developers and testers (Teasley et al. 1993, 1994)
- Grounded work by Wason and extensive variations of Wason’s work in cognitive psychology literature over the last sixty years.

---

Confirmation Bias in Software Engineering

- Due to confirmation bias, developers tend to perform unit tests to make their program work rather than to break their code.
- During all levels of software testing, we must employ a testing strategy, which includes adequate attempts to **fail the code** to reduce software defect density.

“Got it!”
Methodology to Quantify Confirmation Bias

Research Question 1:

How can we identify the measures of confirmation bias in relation to software development process?
Methodology to Quantify Confirmation Bias

Challenge: Quantifying confirmation bias to perform empirical analyses.

Proposed Solution:
Our methodology is an iterative process and it mainly consists of the following steps:
1) Preparation of the confirmation bias test
2) Formation of the confirmation bias metrics set
Confirmation Bias Test

- Confirmation bias test consists of the following:
  - **Interactive Test**
    - based on “Wason’s Rule Discovery Task”
  - **Written Test**
    - based on “Wason’s Selection Task”

### Written Test Content

<table>
<thead>
<tr>
<th>Question Type</th>
<th>No. of Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract Questions</td>
<td>8</td>
</tr>
<tr>
<td>Thematic Questions</td>
<td>6</td>
</tr>
<tr>
<td>SW development/testing questions</td>
<td>8</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>22</strong></td>
</tr>
</tbody>
</table>
Confirmation Bias Test

- Wason’s Rule Discovery Task

**Goal:** Discover the correct rule

Initially, subject is given three numbers, which conform to a simple rule.

Experiment Protocol:
repeat until correct rule is announced
write down three numbers & reasons for choice;
receive feedback from tester;
if you are sure about the rule announce the rule;
end
if you want to terminate break; % terminated
end
Confirmation Bias Test

- **Wason’s Selection Task:**
- **Goal:** To find out which of the four cards should be turned over to test the validity of the statement given below:

  “If there is a D on one side of the card, then it has a 3 on its other side.”

![Image of four cards: D, K, 3, 7]
Example: Wason’s Rule Discovery Task in Relation to Unit Testing

- **Wason’s Rule Discovery Task:**
  - Subjects have a tendency to select many triples (i.e., test cases) that are consistent with their hypotheses and few tests that are inconsistent with them.

- **Observed Similarity with Functional (Black-box) Testing:**
  - Program testers may select many test cases consistent with the program specifications (positive tests) and a few that are inconsistent with them (negative tests).
Example: Wason’s Rule Selection Task in Relation to Unit Testing

Example 1: Suppose you want to make sure that a program avoids dereferencing a null pointer by always checking before dereferencing.

Someone tells you there are only four sections to be tested, and they have determined the following things about those sections:

- **Section A** checks whether the pointer is null. The pointer may or may not be dereferenced there.
- **Section B** does not check whether the pointer is null. The pointer may or may not be dereferenced there.
- **Section C** dereferences the pointer. The pointer may or may not have been checked for nullity.
- **Section D** does not dereference the pointer. The pointer may or may not have been checked for nullity.

Which sections need to be investigated further?

## Confirmation Bias Metrics Set

### Interactive Test Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N_A)</td>
<td>Number of rule announcements</td>
</tr>
<tr>
<td>(T_i)</td>
<td>Duration of interactive question session (in minutes)</td>
</tr>
<tr>
<td>(Ind_{elim/enum})</td>
<td>Eliminative/enumerative index by Wason</td>
</tr>
<tr>
<td>(F_{negative})</td>
<td>Frequency of negative instances</td>
</tr>
<tr>
<td>(F_{IR})</td>
<td>Immediate rule announcement frequency</td>
</tr>
<tr>
<td>(avgL_{IR})</td>
<td>Average length of immediate rule announcements</td>
</tr>
<tr>
<td>(Instances/Time)</td>
<td>Number of instances given per unit time</td>
</tr>
<tr>
<td>(UnqReasons/Time)</td>
<td>Number of unique reasons given per unit time</td>
</tr>
<tr>
<td>(Rules/Time)</td>
<td>Number of rules announced per unit time</td>
</tr>
<tr>
<td>(UnqRules/Time)</td>
<td>Number of unique rules announced per unit time</td>
</tr>
</tbody>
</table>

### Written Test Metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_{Abs})</td>
<td>Score in abstract questions</td>
</tr>
<tr>
<td>(S_{Th})</td>
<td>Score in thematic questions</td>
</tr>
<tr>
<td>(S_{SW})</td>
<td>Score in the second part of the written question set</td>
</tr>
<tr>
<td>(T_{Th+A})</td>
<td>Time it takes to answer the first part of the written question set</td>
</tr>
<tr>
<td>(T_{SW})</td>
<td>Time it takes to answer the second part of the written question set</td>
</tr>
<tr>
<td>(ABS_{CompleteInsight})</td>
<td>Number of abstract questions answered with complete insight</td>
</tr>
<tr>
<td>(ABS_{PartialInsight})</td>
<td>Number of abstract questions answered with partial insight</td>
</tr>
<tr>
<td>(ABS_{NoInsight})</td>
<td>Number of abstract questions answered with no insight</td>
</tr>
<tr>
<td>(Th_{CompleteInsight})</td>
<td>Number of thematic questions answered with complete insight</td>
</tr>
<tr>
<td>(Th_{PartialInsight})</td>
<td>Number of thematic questions answered with partial insight</td>
</tr>
<tr>
<td>(Th_{NoInsight})</td>
<td>Number of thematic questions answered with no insight</td>
</tr>
<tr>
<td>(N_{Falsifier})</td>
<td>Total number of answers with only falsifying tendency</td>
</tr>
<tr>
<td>(N_{Verifier})</td>
<td>Total number of answers with only verifying tendency</td>
</tr>
<tr>
<td>(N_{Matcher})</td>
<td>Total number of answers with only verifying tendency</td>
</tr>
<tr>
<td>(N_{None})</td>
<td>Total number of answers with no defined tendency</td>
</tr>
</tbody>
</table>
Confirmation Bias Metrics Set: Some Practical Results

Interactive Test Outcome: Hypothesis Testing Strategy

Written Test Outcome: Reich and Ruth’s Falsifier/Verifier/Matcher Classification

Group 1*: Developers of a GSM/Telecommunications company (29 subjects)
Group 8*: Computer Engineering PhD candidates with minimum 2 years of development experience (36 subjects)
Research Question 2:

How do confirmation biases of developers affect software quality?
Influence of Developers’ Confirmation Bias on Software Quality – Part 1

**Dataset:**

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of active files</th>
<th>Defect rate</th>
<th># of developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP</td>
<td>3199</td>
<td>0.07</td>
<td>6</td>
</tr>
<tr>
<td>Telecom1</td>
<td>826</td>
<td>0.11</td>
<td>9</td>
</tr>
<tr>
<td>Telecom2</td>
<td>1481</td>
<td>0.03</td>
<td>4</td>
</tr>
<tr>
<td>Telecom3</td>
<td>284</td>
<td>0.02</td>
<td>7</td>
</tr>
<tr>
<td>Telecom4</td>
<td>63</td>
<td>0.05</td>
<td>17</td>
</tr>
</tbody>
</table>

**Steps of the Analysis:**

- Formation of developer groups
- Estimation of developer groups confirmation bias metric values from individual values:
  \[
  S_{ji}^{\max} = \max(A_{di} | \forall d \in G_j) \quad S_{ji}^{\min} = \min(A_{di} | \forall d \in G_j)
  \]
- Measurement of defect rate for each developer group
  \[
  d\tau^i = \frac{N^i_{defectiveFiles}}{N^i_{allFiles}}
  \]

Analysis of the Pearson correlation between developer groups’ confirmation bias metrics and defect rates
Influence of Developers’ Confirmation Bias on Software Quality - Part 1

- Estimation of the correlation between developer groups’ confirmation bias metrics (interactive test) and defect rates.

**Results:**

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>min/max operator</th>
<th>$\rho$</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_A$</td>
<td>max</td>
<td>+0.2092</td>
<td>0.0003</td>
</tr>
<tr>
<td>$Ind_{elim/enum}$</td>
<td>min</td>
<td>-0.2547</td>
<td>1.1E-05</td>
</tr>
<tr>
<td>$T_I$</td>
<td>max</td>
<td>+0.0396</td>
<td>0.5014</td>
</tr>
<tr>
<td>$F_{negative}$</td>
<td>min</td>
<td>-0.3546</td>
<td>4.8E-10</td>
</tr>
<tr>
<td>$F_{IR}$</td>
<td>max</td>
<td>+0.1252</td>
<td>0.0327</td>
</tr>
<tr>
<td>$avg_L_{IR}$</td>
<td>max</td>
<td>+0.5297</td>
<td>1.9E-22</td>
</tr>
<tr>
<td>$Instances/Time$</td>
<td>min</td>
<td>-0.2389</td>
<td>3.8E-05</td>
</tr>
<tr>
<td>$UnqReasons/Time$</td>
<td>min</td>
<td>-0.2355</td>
<td>5.0E-03</td>
</tr>
<tr>
<td>$Rules/Time$</td>
<td>max</td>
<td>-0.4493</td>
<td>7.2E-16</td>
</tr>
<tr>
<td>$UniqueRules/Time$</td>
<td>max</td>
<td>-0.4510</td>
<td>5.5E-16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>TELCO</th>
<th>CMPE</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ind_{Elim/Enum}$</td>
<td>1.16</td>
<td>0.99</td>
<td>0.1875</td>
</tr>
<tr>
<td>$T_I$</td>
<td>18.14</td>
<td>7.40</td>
<td>0.0015</td>
</tr>
<tr>
<td>$F_{IR}$</td>
<td>0.88</td>
<td>0.29</td>
<td>0.4600</td>
</tr>
<tr>
<td>$avg_L_{IR}$</td>
<td>0.65</td>
<td>0.25</td>
<td>0.2735</td>
</tr>
<tr>
<td>$avg_{F_{IR}}$</td>
<td>1.11</td>
<td>1.04</td>
<td>0.8075</td>
</tr>
<tr>
<td>$N_A$</td>
<td>3.07</td>
<td>2.12</td>
<td>0.3380</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>IBM</th>
<th>TELCO</th>
<th>$p$-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ind_{Elim/Enum}$</td>
<td>1.55</td>
<td>1.16</td>
<td>0.0220</td>
</tr>
<tr>
<td>$T_I$</td>
<td>11.56</td>
<td>18.14</td>
<td>0.0385</td>
</tr>
<tr>
<td>$F_{IR}$</td>
<td>0.19</td>
<td>0.85</td>
<td>0.0263</td>
</tr>
<tr>
<td>$avg_L_{IR}$</td>
<td>0.11</td>
<td>0.54</td>
<td>0.0280</td>
</tr>
<tr>
<td>$avg_{F_{IR}}$</td>
<td>1.61</td>
<td>1.01</td>
<td>0.2295</td>
</tr>
<tr>
<td>$N_A$</td>
<td>1.78</td>
<td>3.07</td>
<td>0.0060</td>
</tr>
</tbody>
</table>

Conventional effect sizes as offered by Cohen:

- $\rho = 0.10 - 0.23$ is small effect size,
- $\rho = 0.24 - 0.36$ is medium effect size, and
- $\rho = 0.37 - larger$ is large effect size.
Influence of Developers’ Confirmation Bias on Software Quality - Part 1

Estimation of the correlation between developer groups’ confirmation bias metrics (written test) and defect rates.

Results:

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>min/max operator</th>
<th>ρ</th>
<th>pval</th>
<th>(Group1*)</th>
<th>(Group8*)</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_ABS</td>
<td>min</td>
<td>−0.2371</td>
<td>4.4E-5</td>
<td>0.13</td>
<td>0.61</td>
<td>0.0050</td>
</tr>
<tr>
<td>S_Th</td>
<td>min</td>
<td>0.0536</td>
<td>0.3622</td>
<td>0.53</td>
<td>0.88</td>
<td>0.0110</td>
</tr>
<tr>
<td>S_SW</td>
<td>min</td>
<td>0.0367</td>
<td>0.5332</td>
<td>0.40</td>
<td>0.80</td>
<td>0.0005</td>
</tr>
<tr>
<td>TThABS</td>
<td>max</td>
<td>−0.1396</td>
<td>0.0172</td>
<td>16.41</td>
<td>13.65</td>
<td>0.1405</td>
</tr>
<tr>
<td>TSW</td>
<td>max</td>
<td>+0.1553</td>
<td>0.0172</td>
<td>15.41</td>
<td>12.10</td>
<td>0.0820</td>
</tr>
<tr>
<td>AbsCompleteInsight</td>
<td>min</td>
<td>−0.1280</td>
<td>0.0290</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AbsPartialInsight</td>
<td>max</td>
<td>+0.3572</td>
<td>3.5E-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AbsNoInsight</td>
<td>max</td>
<td>+0.5364</td>
<td>4.4E-23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThCompleteInsight</td>
<td>min</td>
<td>−0.2851</td>
<td>7.7E-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThPartialInsight</td>
<td>max</td>
<td>+0.1128</td>
<td>0.0545</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThNoInsight</td>
<td>max</td>
<td>+0.1342</td>
<td>0.0220</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NFalsifier</td>
<td>min</td>
<td>−0.0935</td>
<td>0.8738</td>
<td>0.43</td>
<td>0.27</td>
<td>0.4180</td>
</tr>
<tr>
<td>NVerifier</td>
<td>max</td>
<td>+0.3742</td>
<td>4.2E-11</td>
<td>0.71</td>
<td>0.73</td>
<td>0.7180</td>
</tr>
<tr>
<td>NMatcher</td>
<td>max</td>
<td>+0.1749</td>
<td>0.0027</td>
<td>16.54</td>
<td>16.36</td>
<td>0.6730</td>
</tr>
<tr>
<td>NNone</td>
<td>max</td>
<td>+0.1852</td>
<td>0.0015</td>
<td>0.79</td>
<td>0.51</td>
<td>0.0000</td>
</tr>
<tr>
<td>TSW</td>
<td></td>
<td></td>
<td></td>
<td>11.54</td>
<td>15.97</td>
<td>0.0075</td>
</tr>
</tbody>
</table>

Conventional effect sizes as offered by Cohen

<table>
<thead>
<tr>
<th>ρ</th>
<th>Power Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.29</td>
</tr>
<tr>
<td>0.20</td>
<td>0.71</td>
</tr>
<tr>
<td>0.30</td>
<td>0.96</td>
</tr>
<tr>
<td>0.40</td>
<td>≥ 0.995</td>
</tr>
<tr>
<td>0.50</td>
<td>≥ 0.995</td>
</tr>
</tbody>
</table>

− ρ = 0.10 – 0.23 is small effect size,
− ρ = 0.24 – 0.36 is medium effect size, and
− ρ = 0.37 – larger is large effect size.
Research Question 3:

How do measures of confirmation bias perform in predicting defect prone parts of software?
Influence of Developers’ Confirmation Bias on Software Quality – Part 2

Construction of the Prediction Model

- **Algorithm**: Naive Bayes
- **Input data**: static code, churn, confirmation bias metrics (models are constructed for each combination of these metrics)
- **Preprocessing**: undersampling
- **10x10 cross validation
- **Performance measures**:

<table>
<thead>
<tr>
<th>Actual Case</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defected</td>
</tr>
<tr>
<td>Defected</td>
<td>TP</td>
</tr>
<tr>
<td>Not-defected</td>
<td>FP</td>
</tr>
</tbody>
</table>

\[ pd = \frac{TP}{TP + FN} \]

\[ pf = \frac{FP}{FP + TN} \]

\[ bal = 1 - \frac{\sqrt{(1 - pd)^2 + (0 - pf)^2}}{\sqrt{2}} \]
Influence of Developers’ Confirmation Bias on Software Quality – Part 2

Results

<table>
<thead>
<tr>
<th>Metric Types</th>
<th>Confirmation Bias</th>
<th>Static Code</th>
<th>Churn</th>
<th>pd</th>
<th>pf</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset: ERP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>0.72</td>
<td>0.29</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>0.91</td>
<td>0.31</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.81</td>
<td>0.38</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>0.93</td>
<td>0.30</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0.71</td>
<td>0.15</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.77</td>
<td>0.27</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.93</td>
<td>0.32</td>
<td>0.74</td>
</tr>
<tr>
<td>Dataset: Telecom1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>0.63</td>
<td>0.33</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>0.60</td>
<td>0.35</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.70</td>
<td>0.32</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>0.69</td>
<td>0.29</td>
<td>0.69</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0.68</td>
<td>0.26</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.64</td>
<td>0.35</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.70</td>
<td>0.32</td>
<td>0.67</td>
</tr>
<tr>
<td>Dataset: Telecom2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>0.83</td>
<td>0.04</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.94</td>
<td>0.11</td>
<td>0.88</td>
</tr>
<tr>
<td>Dataset: Telecom3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>0.83</td>
<td>0.12</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>0.90</td>
<td>0.11</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>0.93</td>
<td>0.21</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.85</td>
<td>0.14</td>
<td>0.81</td>
</tr>
<tr>
<td>Data Science Laboratory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results Summary:

- Confirmation Bias is a single human aspect.
- Yet, using confirmation bias metrics led to comparable performance results in predicting defect prone parts of software.
  - The performance of defect prediction models built by using only confirmation bias metrics is comparable with the performance of the defect prediction models that use static code metrics and churn metrics.
- Therefore, we should further investigate other human aspects...
Factors Affecting Confirmation Bias

Research Question 4:

What are the factors affecting confirmation bias levels of software engineers?
Factors Affecting Confirmation Bias?

**characteristics of the software engineer**
- Education (fields of graduate & undergraduate study)
- Job title
  - researcher
  - developer
  - tester
  - analyst
- Development experience (years)
- Testing Experience (years)
- Confounding factors:
  - Age
  - Gender

**characteristics of the software company**
- In-House vs ISV
- Large-scale vs SME
- Confounding factor:
  - Country (Turkey vs Canada)

**characteristics of the software product**
- Development Methodology
  - Agile
  - TDD
  - Incremental
- Confounding factors:
  - Development Language
  - Level of domain expertise
Factors Affecting Confirmation Bias

**Dataset:**

- 174 subjects (18 researchers, 100 developers, 30 analysts, 26 testers)
- 4 large scale companies and 3 SMEs (1 large scale company in Canada, the remaining companies are in Turkey)

<table>
<thead>
<tr>
<th>Group</th>
<th>Researcher</th>
<th>Analyst</th>
<th>Developer</th>
<th>Tester</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSM-CRM</td>
<td>–</td>
<td>–</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>GSM-Billing</td>
<td>–</td>
<td>–</td>
<td>14</td>
<td>–</td>
</tr>
<tr>
<td>GSM-TDD</td>
<td>–</td>
<td>–</td>
<td>3</td>
<td>–</td>
</tr>
<tr>
<td>Banking</td>
<td>–</td>
<td>29</td>
<td>27</td>
<td>–</td>
</tr>
<tr>
<td>DBMS</td>
<td>–</td>
<td>1</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>CMPE-Exp</td>
<td>18</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Finance</td>
<td>–</td>
<td>–</td>
<td>12</td>
<td>–</td>
</tr>
<tr>
<td>ERP</td>
<td>–</td>
<td>–</td>
<td>6</td>
<td>–</td>
</tr>
<tr>
<td>CRM</td>
<td>–</td>
<td>–</td>
<td>6</td>
<td>–</td>
</tr>
<tr>
<td>Mobile</td>
<td>–</td>
<td>–</td>
<td>8</td>
<td>–</td>
</tr>
<tr>
<td><strong>Subtotals</strong></td>
<td><strong>18</strong></td>
<td><strong>30</strong></td>
<td><strong>100</strong></td>
<td><strong>26</strong></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>174</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Factors Affecting Confirmation Bias

Methodology:
- Multi-way Analysis of Variance (ANOVA) using MATLAB statistics toolbox.
- In order to make the interpretation of the results much easier, we formulated a single derived metric to quantify confirmation bias level.

\[ CB^i = \frac{D^i}{maxDist} \]

\[ D^i = \sqrt{\left( \sum_{j}^{N} (cb^i_j - cb^j_{\text{ideal}})^2 \right)} \]

\[ maxDist = \sqrt{\left( \sum_{j}^{N} (cb^j_{\text{worst}} - cb^j_{\text{ideal}})^2 \right)} \]

Confirmation Bias Level: Deviation of confirmation bias metrics values from the corresponding ideal metrics values.
Factors Affecting Confirmation Bias

Some Significant Results:

- **Education**: A significant effect of education (e.g. fields of undergraduate and graduate study.)
- **Experience**: A significant effect of experience in software development/testing was not observed.
- **Development methodology**: TDD developers performed better at confirmation bias tests.
- **Researchers vs. Developers/Testers/Analysts**: Researchers performed better.
  - Individuals who have been trained in logical reasoning and hypotheses testing techniques exhibit less confirmatory behavior.
Factors Affecting Confirmation Bias

- **Some Practical Outcomes:**
  - Training in organizations is focused on tasks rather than personal skills.
  - Considering that the percentage of people with low confirmation bias is very low in the population, an organization should find ways to improve basic logical reasoning and strategic hypothesis testing skills of its software engineers.
  - There is a need to design training sessions involving the application of various de-biasing strategies. Goal should be to increase metacognitive awareness among individuals.
Future Directions: towards a Tool...

- Towards a Tool...
- A tool is required to make things less complicated for managers such as “Dione\(^2\)”. 

---

Related Publications

We are looking for collaborations

- To work in projects
- To formulate new research questions
- To understand patterns in software development in real life settings
- To replicate learnings in public data sets
- Jointly publish
THANK YOU
ANY QUESTIONS?

Ayşe Bener: ayse.bener@ryerson.ca
Gül Çalıklı: gcalikli@ryerson.ca

http://www.ryerson.ca/~abener/dsl.html