Threats to validity in TDD research

Timm Gross
Interviews about testing practices

Small development team at a swiss university

Developing mostly integration solutions

5 developers, 3 interviews each in 1 hour

Focus on bug fixes

Method: Ethnographically informed qualitative expert interviews analyzed with the grounded theory coding technique (Flick, 2009)
Interviews: Why do you test?

Insurance of quality
Future maintainability
Dealing with complexity
Confidence in solutions
Documentation of assumptions
Passive knowledge transfer
Enjoyment

D2: Especially now or in the future when we have fresh software engineers, it will be good to influence them positively.
Interviews: Why do you test?

- Insurance of quality
- Future maintainability
- Dealing with complexity
- Confidence in solutions
- Documentation of assumptions
- Passive knowledge transfer
- Enjoyment

- Quality related
- Non-quality related
Interviews: Why do you not test?

Problem: Social desirability bias

External dependencies

Configuration

Inadequate existing testing suites

Shortcuts

D4: Testing of infrastructure makes no sense because this type of tests are hopefully done by the vendor of the product. [...] Analog: Wir testen nicht ob Java korrekt sortieren kann.
Interviews: Conclusion

Interviewees view testing as both

- A tool to achieve better results (quality related aspects)
- A tool to structure work and to collaborate (non-quality related aspects)

Developers: Testing is important (in certain cases)

Lesson learned: social desirability bias in expert interviews

Could we use TDD to better leverage the benefits of testing?
Test driven development

“No studies have categorically demonstrated the difference between TDD and any of the many alternatives in quality, productivity, or fun. However, the anecdotal evidence is overwhelming, and the secondary effects are unmistakable.”

- B. K. Beck & Date, 2002

18 years later: still true
State of research

6 meta-analyses

Quality: no difference - improvement

Productivity: inconclusive - degradation

Inconsistencies:

● Comparisons: degree of iterativeness (waterfall, iterative test last, etc.)
● Rigor (statistical methods, experiment set-up, etc.)
● Relevance (topical, realistic setting, etc.)
● Participants (skill level)
● Context (academic vs. industrial)
Application in the “Wild”

Borle et al. (2018): Analyses of 256,572 public GitHub projects

- only 16.1% of GitHub repositories contained test files
- only 0.8% strictly practiced TDD

Beller et al. (2017): Observation of 2,443 software developers over 2.5 years

- 43% of all projects contained test files
- only 1.7% of all developers followed a strict definition of TDD
Summary

1. Anecdotal evidence from “champions for TDD” is overwhelming
2. Research on the effects of TDD is inconclusive
3. The practice of TDD in real life projects is very limited
Literature analysis of threats to validity

Goal:

- Insight into the discrepancy between anecdotal evidence and literature findings
- Identify problems that hinder the application of the research in industrial contexts

Method: hermeneutic literature review

Focus: not results but threats to validity
Literature analysis: Data collection

Identification of research papers

- Web search
- Snowball approach
- Literature reviews

Inclusion criteria:

- Only TDD
- Experiments (case studies), statistical analysis, qualitative research, literature reviews
- Recent studies (2009 onwards)
- High quality & explicit threats to validity
Literature analysis: Methodology

Hermeneutical approach

- Identification of next paper
- In depth analysis of set-up, execution, conclusion and threats to validity
- Adding to and sharpening of a list of threats to validity
- Repeat until no more new categories emerge (15 papers & 7 literature reviews)
<table>
<thead>
<tr>
<th>Authors</th>
<th>Title</th>
<th>Method</th>
<th>Context</th>
<th>Subjects</th>
<th>TDD Experience of the subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thomson et al. (2009)</td>
<td>What Makes Testing Work: Nine Case Studies of Software Development Teams</td>
<td>Experiment/Qualitative Study</td>
<td>Academic</td>
<td>ca. 36 students (9 teams a 3-5 2-3 year students)</td>
<td>1 semester course</td>
</tr>
<tr>
<td>Romano et al. (2017)</td>
<td>Findings from a multi-method study on test-driven development</td>
<td>Qualitative Study</td>
<td>Academic &amp; Industrial</td>
<td>14 graduate students, 6 professionals</td>
<td>2 months course</td>
</tr>
<tr>
<td>Buchan et al. (2011)</td>
<td>Causal Factors, Benefits and Challenges of Test-Driven Development: Practitioner Perceptions</td>
<td>Qualitative Study</td>
<td>Industrial</td>
<td>5 interviews (4 team leaders, 1 business analyst)</td>
<td>3 years practice</td>
</tr>
<tr>
<td>Scanniello et al. (2016)</td>
<td>Students’ and Professionals’ Perceptions of Test-driven Development: A Focus Group Study</td>
<td>Qualitative Study</td>
<td>Academic &amp; Industrial</td>
<td>2 focus groups (13 master students, 5 professionals)</td>
<td>students: courses during education, professionals: at least 8 week course</td>
</tr>
<tr>
<td>Beller et al. (2019)</td>
<td>Developer Testing in The IDE: Patterns, Beliefs, And Behavior</td>
<td>Statistical analysis</td>
<td>Industrial</td>
<td>2,443 software engineers monitored over 2.5 years</td>
<td>unknown</td>
</tr>
<tr>
<td>Borle et al. (2018)</td>
<td>Analyzing the effects of test driven development in GitHub</td>
<td>Statistical analysis</td>
<td>Industrial</td>
<td>256572 GitHub projects</td>
<td>unknown</td>
</tr>
<tr>
<td>Bannerman and Martin (2011)</td>
<td>A multiple comparative study of test-with development product changes and their effects on team speed and product quality</td>
<td>Statistical analysis</td>
<td>Industrial</td>
<td>6 long term open source projects</td>
<td>unknown</td>
</tr>
</tbody>
</table>
Literature analysis: Findings

- Participant choice
## Participants by context

<table>
<thead>
<tr>
<th></th>
<th>&lt;20 participants</th>
<th>21-50 participants</th>
<th>&gt;50 participants</th>
</tr>
</thead>
</table>
## TDD experience

<table>
<thead>
<tr>
<th>Experience</th>
<th>Authors and Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1 week</td>
<td>Tosun et al. (2018), Fucci et al. (2017), Thomson et al. (2009), Santos et al. (2018)</td>
</tr>
<tr>
<td>1 week - 0.5 years</td>
<td>Fucci et al. (2018), Kazerouni et al. (2019), Romano et al. (2017), Scanniello et al. (2016), Dogša and Batic (2011), Fucci and Turhan (2013)</td>
</tr>
<tr>
<td>0.5 years - 1 year</td>
<td>Pančur and Ciglaric (2011)</td>
</tr>
<tr>
<td>more</td>
<td>Buchan et al. (2011)</td>
</tr>
</tbody>
</table>
Literature analysis: Findings

- Participant choice
- Task selection
# Task selection

<table>
<thead>
<tr>
<th>Number of Synthetic Tasks</th>
<th>Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 synthetic task</td>
<td>Romano et al. (2017), Fucci and Turhan (2013)</td>
</tr>
<tr>
<td>2 synthetic tasks</td>
<td>Tosun et al. (2018), Pančur and Ciglaric (2011)</td>
</tr>
<tr>
<td>3 synthetic tasks</td>
<td>Fucci et al. (2017), Santos et al. (2018)</td>
</tr>
<tr>
<td>4 synthetic tasks</td>
<td>Fucci et al. (2018), Kazerooni et al. (2019)</td>
</tr>
<tr>
<td>Real projects</td>
<td>Thomson et al. (2009), Dogša and Batic (2011)</td>
</tr>
<tr>
<td>Not applicable</td>
<td>Buchan et al. (2011), Scanniello et al. (2016)</td>
</tr>
</tbody>
</table>
### Greenfield vs. brownfield projects

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Brownfield</td>
<td>Buchan et al. (2011), Scanniello et al. (2016)</td>
</tr>
</tbody>
</table>
Literature analysis: Findings

- Participant choice
- Task selection
- Context
- Quality
  - Lack of attention to internal code quality
<table>
<thead>
<tr>
<th></th>
<th>Tosun et al. (2018), Pančur and Ciglaric (2011), Kazerouni et al. (2019), Thomson et al. (2009), Borle et al. (2018), Bannerman and Martin (2011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>Tosun et al. (2018), Pančur and Ciglaric (2011)</td>
</tr>
<tr>
<td>Mutation score</td>
<td>Fucci et al. (2017), Fucci et al. (2018), Fucci and Turhan (2013), Santos et al. (2018), Beller et al. (2019)</td>
</tr>
<tr>
<td>None</td>
<td>Romano et al. (2017), Buchan et al. (2011), Scanniello et al. (2016)</td>
</tr>
<tr>
<td>Not applicable</td>
<td></td>
</tr>
</tbody>
</table>
Literature analysis: Findings

- Participant choice
- Task selection
- Context
- Quality
  - Lack of attention to internal code quality
  - Lack of attention to test quality
  - Productivity (short term vs long term)
- Length of observation
- Comparisons
- Lack of qualitative research
- TDD on a spectrum
- Inclusion of TDD in company policies
Literature analysis: Conclusion

Often TDD is understood through a mechanistical lens

Analogous the medical studies:

- Problem: Produce Code with high quality and high productivity
- Treatments: Application of TDD vs. Waterfall
- Analysis: Comparison between the treatments

BUT: We argue that TDD is not only a treatment to the problem.

It is also a way for developers to structure their work and their working together
Conclusion

TDD research is inconclusive

TDD advocates defend it strongly

TDD is not as widely applied as expected

Interviewed developers put equal emphasis on quality related and non-quality related factors

TDD research often has a very mechanistical lens and is in general unconcerned with non-quality related aspects
Conclusion

We argue that further study of the non-quality related aspects of TDD might be worthwhile to close the gap between research and anecdotal evidence.

List of threats to validity to account for

Analog: Computer supported collaborative work
Summary

1. Anecdotal evidence from “champions for TDD” is overwhelming
2. Research on the effects of TDD is inconclusive
3. The practice of TDD in real life projects is very limited

Method: hermeneutic literature review
Focus: not results but threats to validity

Threats to validity

- Participant choice
- Task selection
- Context
- Quality
  - Lack of attention to internal code quality
  - Lack of attention to test quality
  - Productivity (short term vs long term)
- Length of observation
- Comparisons
- Lack of qualitative research
- TDD on a spectrum
- Inclusion of TDD in company policies

Interviews: Why do you test?

Insurance of quality
Future maintainability
Dealing with complexity
Confidence in solutions
Documentation of assumptions
Passive knowledge transfer
Enjoyment

Lesson learned: social desirability bias in expert interviews