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

Quality related

Confidence in solutions

Documentation of assumptions

Passive knowledge transfer

Enjoyment

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 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)

Authors	Title	Method	Context	Subjects	TDD Experience of the subjects
Thomson et al. (2009)	What Makes Testing Work: Nine Case Studies of Software Development Teams	Experiment/ Qualitative Study	Academic	ca. 36 students (9 teams a 3-5 2-3 year students)	1 semester course
Romano et al. (2017)	Findings from a multi-method study on test-driven development	Qualitative Study	Academic & Industrial	14 graduate students, 6 professionals	2 months course
Buchan et al. (2011)	Causal Factors, Benefits and Challenges of Test-Driven Development: Practitioner Perceptions	Qualitative Study	Industrial	5 interviews (4 team leaders, 1 business analyst)	3 years practice
Scanniello et al. (2016)	Students' and Professionals' Perceptions of Test-driven Development: A Focus Group Study	Qualitative Study	Academic & Industrial	2 focus groups (13 master students, 5 professionals)	students: courses during education, professionals: at least 8 week course
Beller et al. (2019)	Developer Testing in The IDE: Patterns, Beliefs, And Behavior	Statistical analysis	Industrial	2,443 software engineers monitored over 2.5 years	unknown
Borle et al. (2018)	Analyzing the effects of test driven development in GitHub	Statistical analysis	Industrial	256572 GitHub projects	unknown
Bannerman and Martin (2011)	A multiple comparative study of test-with development product changes and their effects on team speed and product quality	Statistical analysis	Industrial	6 long term open source projects	unknown

Literature analysis: Findings

- Participant choice

Participants by context

	<20 participants	21-50 participants	>50 participants
Industrial	Romano et al. (2017), Buchan et al. (2011), Scanniello et al. (2016), Santos et al. (2018)	Tosun et al. (2018), Dogša and Batic (2011), Fucci et al. (2017)	
Academic	Romano et al. (2017), Scanniello et al. (2016)	Thomson et al. (2009)	Pančur and Ciglaric (2011), Kazerouni et al. (2019), Fucci and Turhan (2013)

TDD experience

<1 week	Tosun et al. (2018), Fucci et al. (2017), Thomson et al. (2009), Santos et al. (2018)
1 week - 0.5 years	Fucci et al. (2018), Kazerouni et al. (2019), Romano et al. (2017), Scanniello et al. (2016), Dogša and Batic (2011), Fucci and Turhan (2013)
0.5 years - 1 year	Pančur and Ciglaric (2011)
more	Buchan et al. (2011)

Literature analysis: Findings

- Participant choice
- Task selection

Task selection

1 synthetic task	Romano et al. (2017), Fucci and Turhan (2013)
2 synthetic tasks	Tosun et al. (2018), Pančur and Ciglaric (2011)
3 synthetic tasks	Fucci et al. (2017), Santos et al. (2018)
4 synthetic tasks	Fucci et al. (2018), Kazerouni et al. (2019)
Real projects	Thomson et al. (2009), Dogša and Batic (2011)
Not applicable	Buchan et al. (2011), Scanniello et al. (2016)

Greenfield vs. brownfield projects

Greenfield	Tosun et al. (2018), Pančur and Ciglaric (2011), Fucci et al. (2017), Fucci et al. (2018), Kazerouni et al. (2019), Romano et al. (2017), Thomson et al. (2009), Dogša and Batic (2011), Fucci and Turhan (2013), Santos et al. (2018)
Brownfield	Buchan et al. (2011), Scanniello et al. (2016)

Literature analysis: Findings

- Participant choice
- Task selection
- Context
- Quality
  - Lack of attention to internal code quality

Internal code quality metrics

Code coverage	Tosun et al. (2018), Pančur and Ciglaric (2011), Kazerouni et al. (2019), Thomson et al. (2009), Borle et al. (2018), Bannerman and Martin (2011)
Complexity	Pančur and Ciglaric (2011), Dogša and Batic (2011), Bannerman and Martin (2011)
Mutation score	Tosun et al. (2018), Pančur and Ciglaric (2011)
None	Fucci et al. (2017), Fucci et al. (2018), Fucci and Turhan (2013), Santos et al. (2018), Beller et al. (2019)
Not applicable	Romano et al. (2017), Buchan et al. (2011), Scanniello et al. (2016)

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.

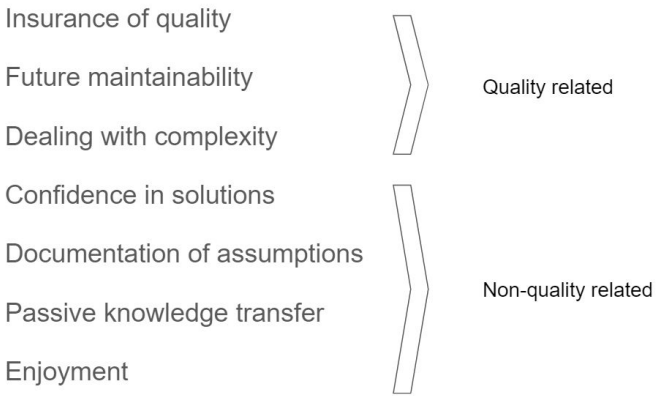
List of threats to validity to account for

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

Interviews: Why do you test?



Lesson learned: social desirability bias in expert interviews

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