Technical Debt and the Effect of Agile Software Development Practices on It – An Industry Practitioner Survey

Sixth International Workshop on Managing Technical Debt
Victoria, British Columbia, Canada
30th of September, 2014

Johannes Holvitie, Sami Hyrynsalmi & Ville Leppänen
TUCS – Turku Centre for Computer Science
UTU – University of Turku
Dept. of IT & Dept. of Management an d Entrepreneurship
Turku, Finland

Outline

1)Introduction & Motivation

2)Related Work

3)Research Questions

4)Study Design & Implementation

5)Results & Discussion

6)Conclusions

Introduction

- Modern software development environments expect projects to cope with
  - frequently changing requirements
  - optimally scarce resources
- while efficiently delivering complete software products and services
- Emergent to a school of development methods we regard as Agile or Lean

Motivation

- Agile methods describe processes and practices that solve a number of development issues
  - resulted to wide adoption; but still not without their shortcomings.
- Arguably, a major reason for this is that the methods are applied by individuals
  - makes them sensitive to subjective decisions
- Individuals base decisions on available information
  - when it's not exhaustive, the results likely deviate from the optimal
  - sometimes an adequate solution is preferred over optimal
- Involvement of individuals ensures that there is technical debt
  - success of agile development methods indicates that they are capable of managing the technical debt within

Related Work

- Klinger & al.
  - decisions to incur technical debt come from non-technical stakeholders, often with competing interests
- Snipes & al.
  - majority of defect debt cost came from investigating and validating it, six components that affect paying/incurring it; combination of said components responsible for how effective a chosen TDM strategy was

Related Work

- Software development practices and instances of technical debt
- Codabux & Williams
  - developers consider own taxonomy to be the best, dedicated teams/maintenance tasks for TDM, outcomes dependent on intentionality and stakeholder involvement
- Lim & al.
  - 75% were not familiar with TD, incurred intentionally and due to competing concerns, long-term effects, difficult to manage

Research Questions

- Agile development methods are capable of managing or at least accommodating TD
  - methods are defined through the processes and practices they introduce
- Mapping of common agile process and practices sensitive to technical debt
- Two notable factors to aggregating a subjective mapping
  - individual's software development background
  - technical debt knowledge

→ Three research question groups *RQ1, 2, and 3*

Research Questions (RQ1)

- Establish individuals' software development background and how this reflects in assumed and actual TD knowledge

For an individual

RQ1.1 does work experience,

RQ1.2 do used agile development practices, or

RQ1.3 do associated project responsibilities

correlate with what the respondent perceives his/hers assumed or actual technical debt knowledge to be?

RQ1.4 in which mediums has he/she or his/hers colleagues applied the concept of technical debt?

RQ1.5 in which situations does he perceive the use of the technical debt concept as helpful?

Research Questions (RQ2)

- Map which agile agile development practices are in use and what is their perceived effect on TD
  - rely on RQ1 to establish definitions of TD for respondents

Are there certain software development practices/processes for which

RQ2.1 their effect on technical debt is seen to be significantly positive, neutral, or negative?

RQ2.2 it is seen that they (do not) cover the team's or the project's development management needs?

RQ2.3 it is seen that they (are not) able to cover the entire space of matters that require management?

Research Questions (RQ3)

- Are the respondents able to distinguish singular instances of technical debt

For a concrete instance of technical debt

RQ3.1 in which project component does it reside?

RQ3.2 what are the causes for its emergence?

RQ3.3 is it legacy?

RQ3.4 is its size dynamic?

RQ3.5 does its effects correlate with its size?

Study Design

1) General Information

- respondent's software development background
- organization details (concurrent project count, size, revenue)
- project details (deliverable, team size, dev. cycle, roles)

2) Used Development Techniques

- Limitations forbid querying with exhaustive lists
  - Practices: XP practices/strategies (Abrahamsson abstractions)
  - Processes: Scrum process components and artifacts
- 5-point verbal scale to indicate adoption level
- sufficient in covering management needs and the space of trackables

Study Design

3) Technical debt and its manifestations

- Concept of technical debt
  - Prior knowledge, in which mediums, own definition
  - Show definitions and establish closeness
    - Intentional and unintentional accumulation (McConnell)
    - Effects on development (Brown & al.)
- Concrete technical debt
  - if and how TD affects respondents work
    - record effect of previous practices and processes
    - for a single case record: project component, reasons, origin, dynamics

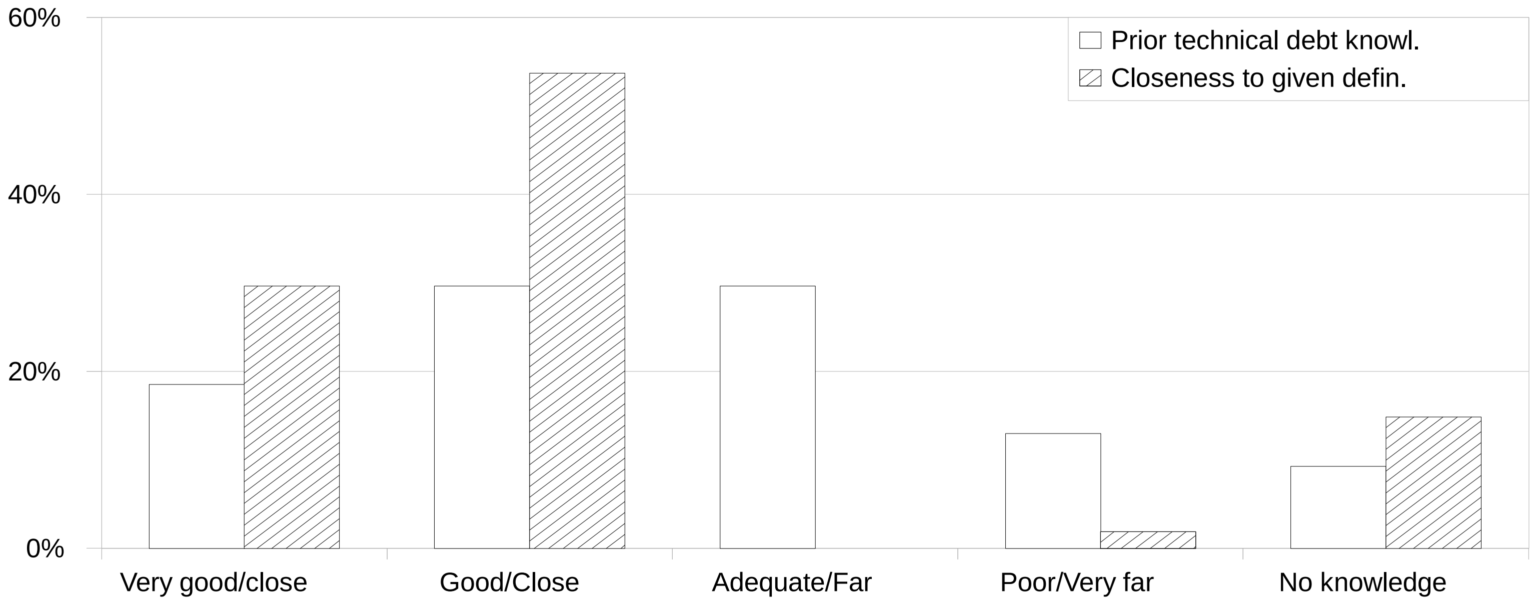
Study Implementation

- Web-questionnaire
  - 37 questions in main part
    - 34 close-ended & 3 open
    - 36 required & 1 optional
  - 6 questions in optional TD comp. definition part
- Trialed in-house and off-house
  - restructuring to increase motivation
  - answer time estimate 10 + 2 minutes

Data Collection

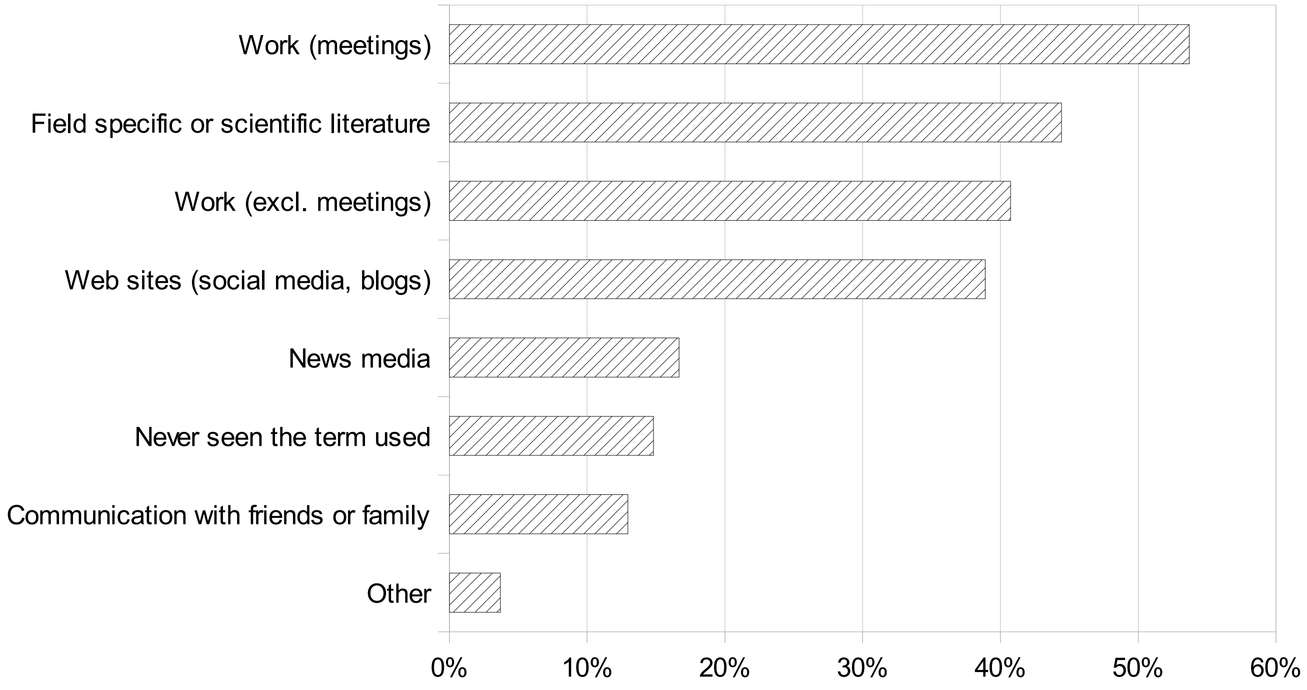
- Preset group of Finnish software organizations
  - company industrial classification and/or affiliation
  - direct contact preferred
    - cases where not possible: controlled internal distribution
  - contacts: cover letter and two follow-ups
- 507 individuals in 153 companies
  - 80 responses, 54 completed
  - India, Sweden, Greece and some unidentified

Individual Perceptions about TD (RQ1.1-3)



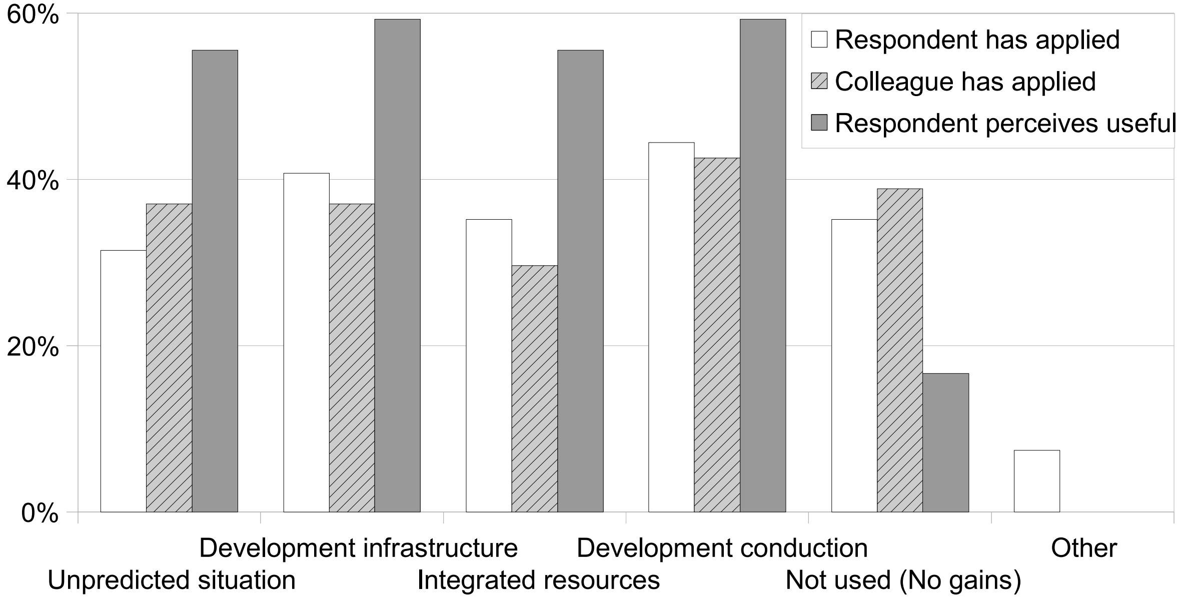
- No difference in classifying by experience, applied techniques, or roles
- Prior knowledge follows normal dist; a bit under 80% indicate at least *adequate knowledge*
- After being shown the definition 80% indicated to be *close* or *very close*
- Unfortunately *no knowledge* also increases to 15%

Individual Perceptions about TD (RQ1.4)



- As expected, most indicated use for occupation related areas
  - Over half have seen/heard the term being used in work meetings
- 15% still report never seeing/hearing the term being used

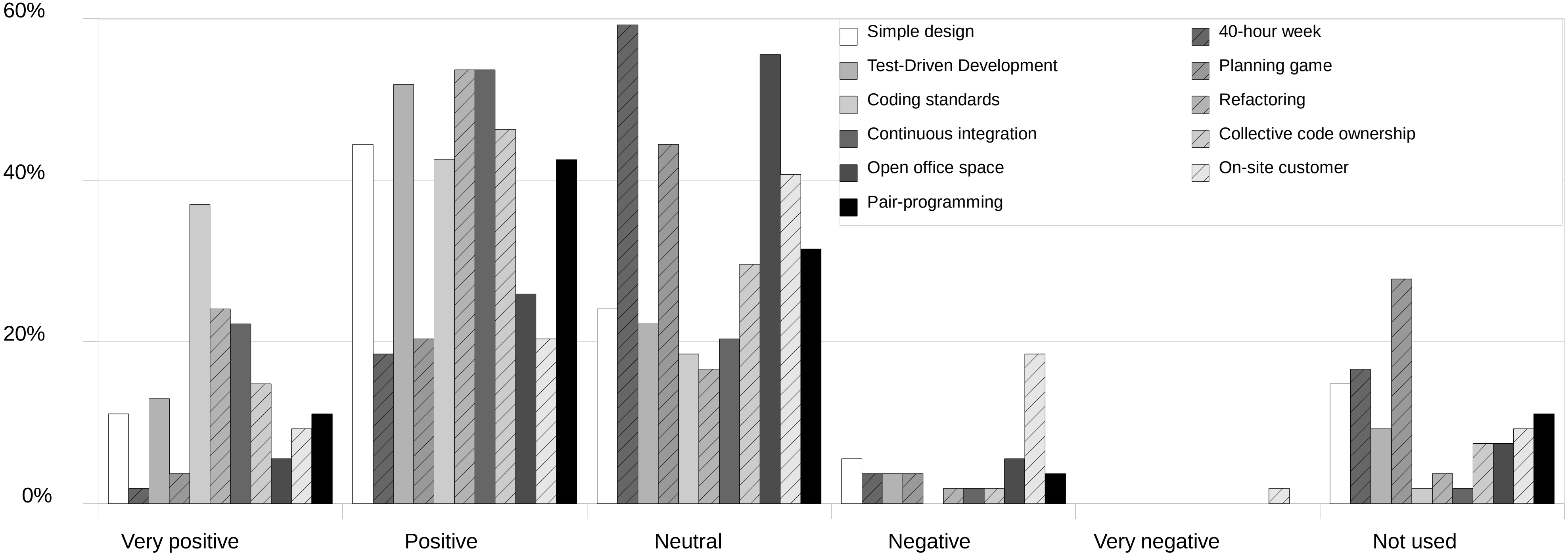
Individual Perceptions about TD (RQ1.5-6)



- Over half perceive application of the TD concept useful in all scenarios
- Current practice is different
  - applied in only circa 35% of cases
  - Almost 40% indicate no application and 15% see no gains in applying

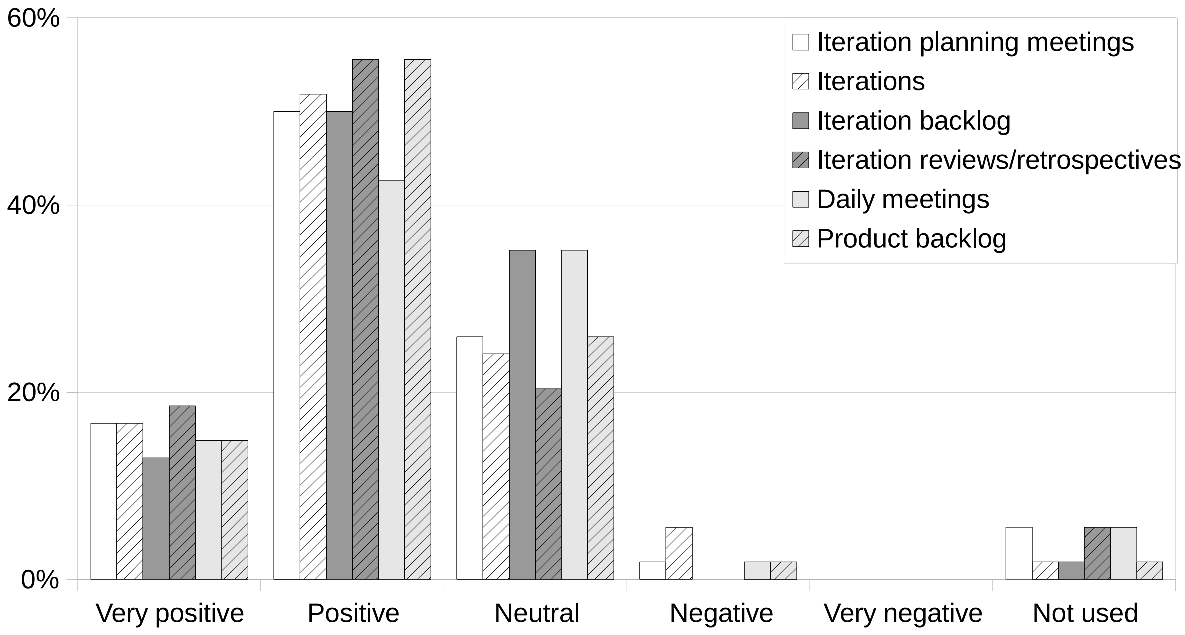
Turun yliopisto
University of Turku

Agile Software Development Practices (RQ2.1)



- Over half indicate practices immediate to implementation as being (very) positive
  - ◦ *Simple design, TDD,* ***coding standards*** etc.
- *40-hour week* and *open office space* indicated as neutral by over half
- Widest distribution recorded for the *on-site customer* 20% - 40% - 20% effects

Agile Software Development Processes (RQ2.1)



- Two thirds perceive all processes to be (very) positive
  - *Iteration reviews/retrospectives* most positive
- 6% indicated that *iterations* have a negative effect

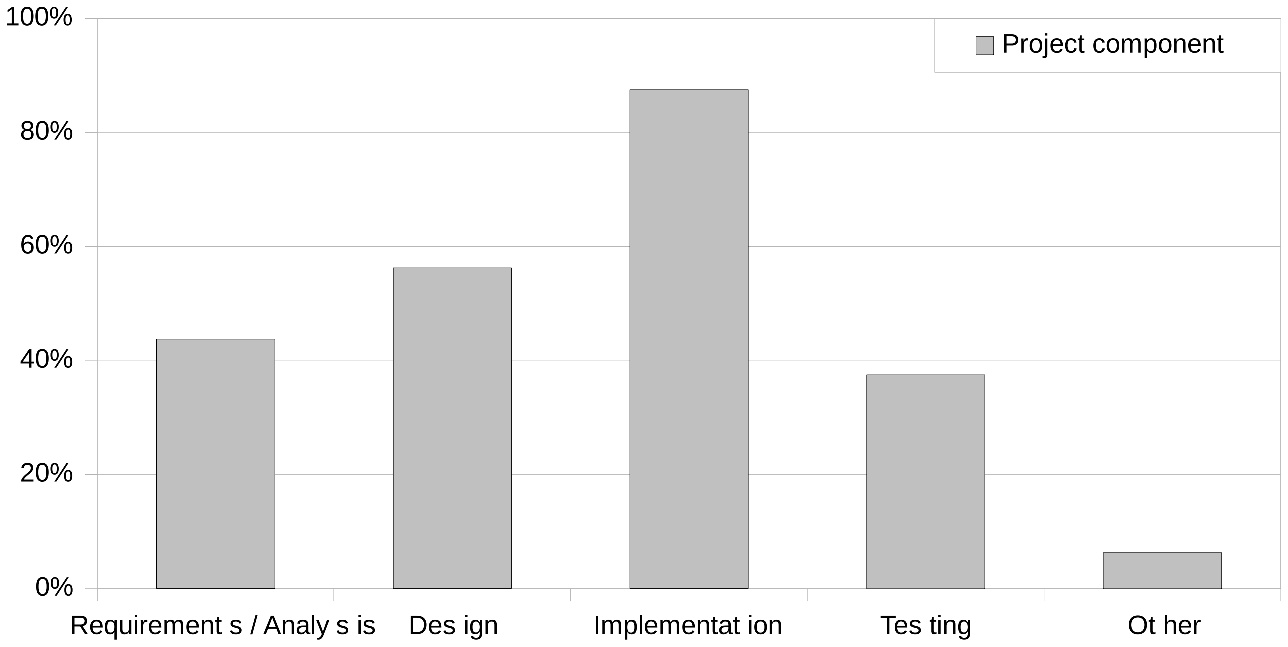
Managing Agile Software Development (RQ2.2-3)

- Respondents were queried if the selection of processes and practices was
  - adequate for their management needs (RQ2.2)
    Avg. 2.83; SD 0.84
  - capable of covering the entire space of matters/aspects that require management (RQ2.3) **Avg. 2.56; SD 0.88**
- No technique and adoption level combination with a statistically significant difference

Manifestations of Technical Debt (RQ3)

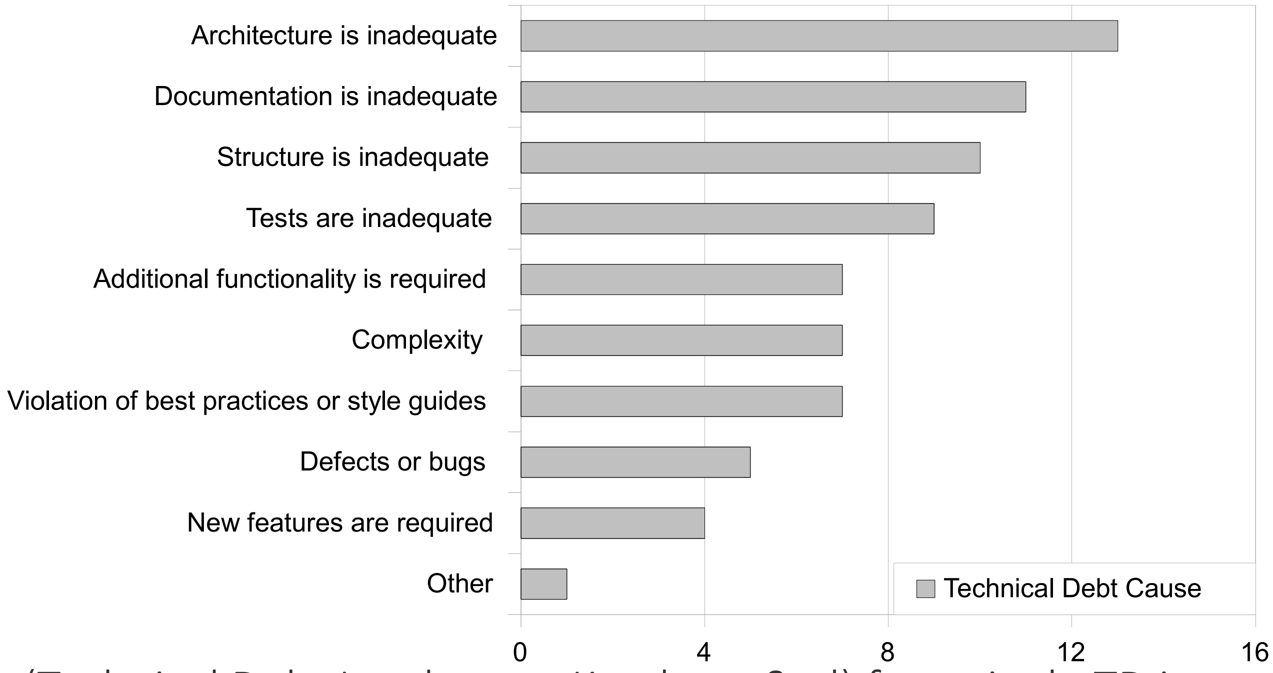
- 93% of respondents indicated that there are components in their work currently being affected by technical debt
- 30% opted to provide a detailed description

Manifestations of Technical Debt (RQ3.1)



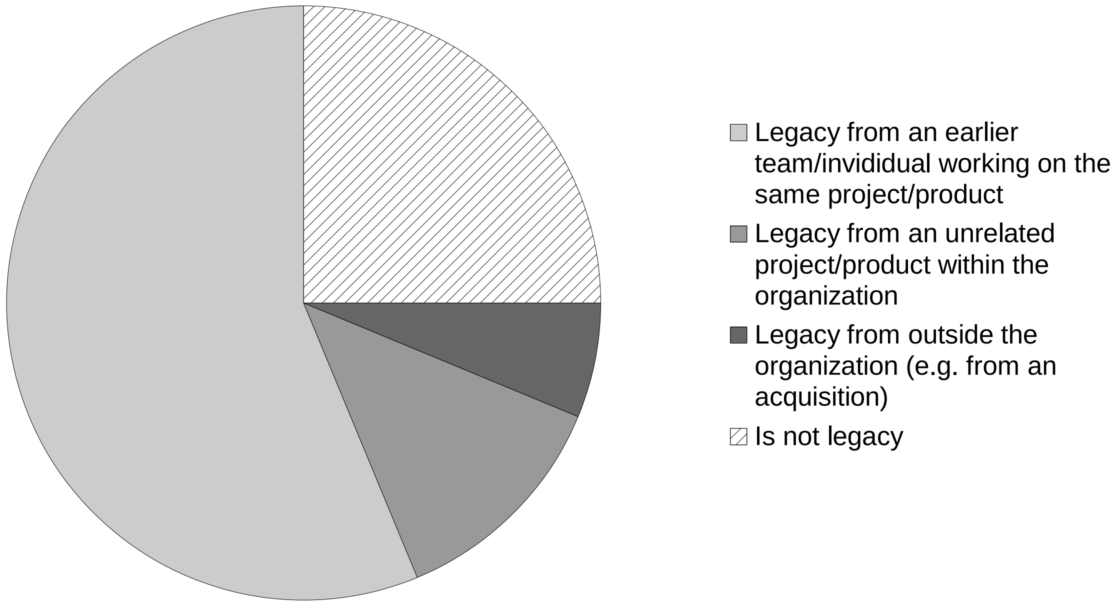
- A single TD instance affected a bit over two project components
  - *Implementation* is most affected
  - *RA & D* more affected than *testing*

Manifestations of Technical Debt (RQ3.2)



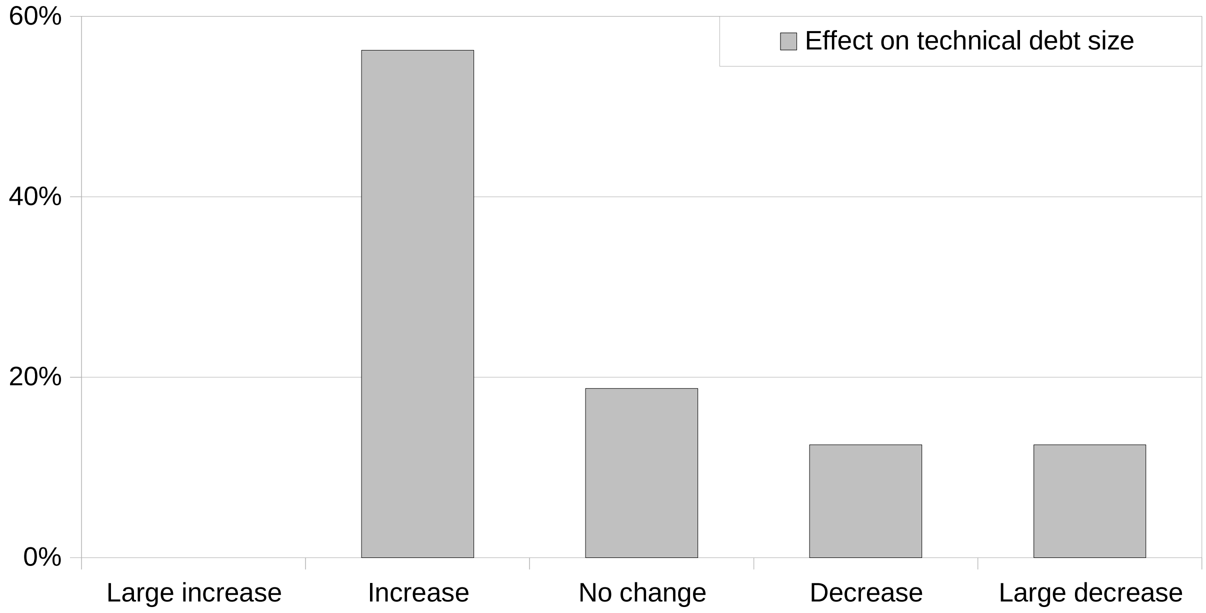
- ~4.5 causes (Technical Debt Landscape, Kruchten & al) for a single TD instance
  - *Inadequate architecture & documentation* most common
  - *New features are required* and *defects or bugs* least frequent
- Obfuscation of implementation structures seen to cause more TD than incomplete functionality

Manifestations of Technical Debt (RQ3.3)



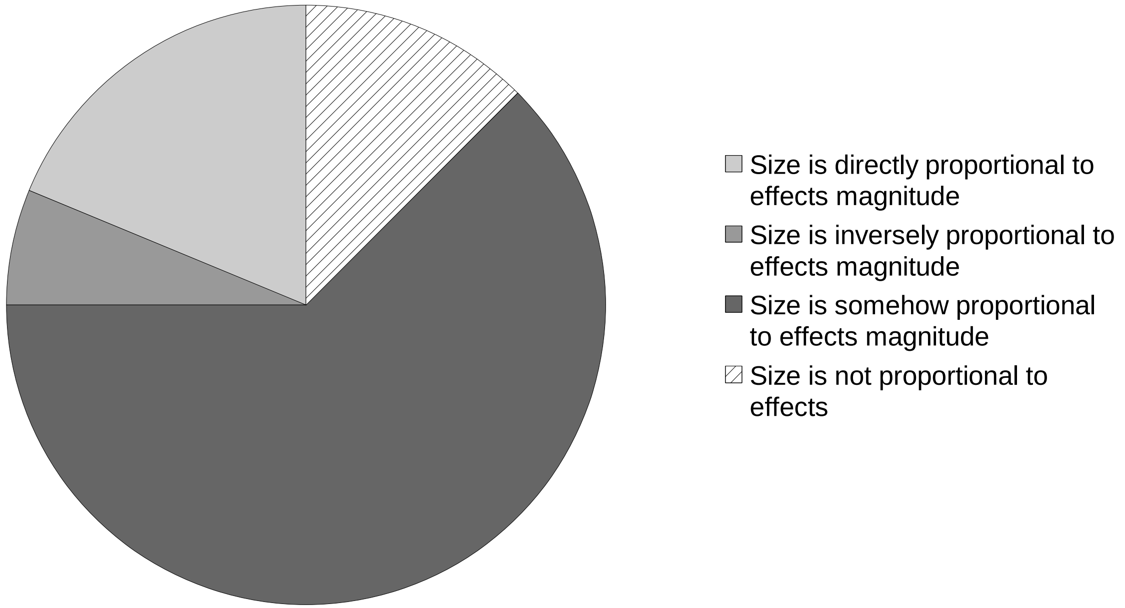
- In three cases out of four TD instance is legacy
  - 90% from this is internal legacy

Manifestations of Technical Debt (RQ3.4)



- Over half indicated continued development to increase the size of technical debt
- One fifth indicated no change
- One fourth indicated (large) decrease

Manifestations of Technical Debt (RQ3.5)



- ~90% indicated a correlation between the size of a TD instance and the effects it caused
  - 60% were however unable to indicate in detail how
- Rest didn't see a connection

Discussion - Individual Perceptions about TD (RQ1)

- Practitioners are quite familiar with the TD concept but they assume less of themselves
    - a possible reason for why TD is applied in 20% less cases than it is seen useful for
- General outlook for TD increased use is positive but there is still a 15% share who have no prior knowledge and have never encountered the term

Discussion - Agile Practices and Processes (RQ2)

- Practices which safe guard the state of the implementation are considered to have the most positive effect on TD and its management
- Issues raised by competing interests of different stakeholder groups are evident
  - widest distribution recorded for *on-site customer*
  - However, the most positive process effect was recorded for *iteration review/retrospective* which may have multi-stakeholder participation

    → limited stakeholder participation is preferred from TD PoV?

Discussion - Manifestations of TD (RQ3)

- *Implementation* is by far the most affected component

  → practices closest to implementation maintenance considered to have the most positive effect

- but the root cause in most cases is either *inadequate architecture* or *documentation*

- ¾ cases were legacy, size followed development and effects *somehow* followed size

Limitations

- Queried companies were based in Finland
  - bias towards country's culture
- Only parties already interested in TD answered
  - may bias towards a more positive outcome
  - however, over 20% respondents were not familiar
- Major part of the respondents represent the same company
  - may bias towards this opinion/culture
  - however, recorded company size distrib. ~ national distrib.

Conclusions

- Studied Finnish professional software developer perceptions about technical debt
  - Presumed TD knowledge is lower than actual and it shows as under utilization
- Effect of agile software development practices and processes on TD
  - ones closest to implementation most positive
  - widest distributions on a multi-stakeholder related activity
- Concrete technical debt
  - in implementation, cause in design, legacy, dynamic size/eff.

Thank you!

Currently working on replicating the study, especially for different software development cultures, and would like to welcome all collaboration!

Johannes Holvitie
jjholv@utu.fi