When will it be done?

The what, who and how of timely software delivery

Elvan Kula

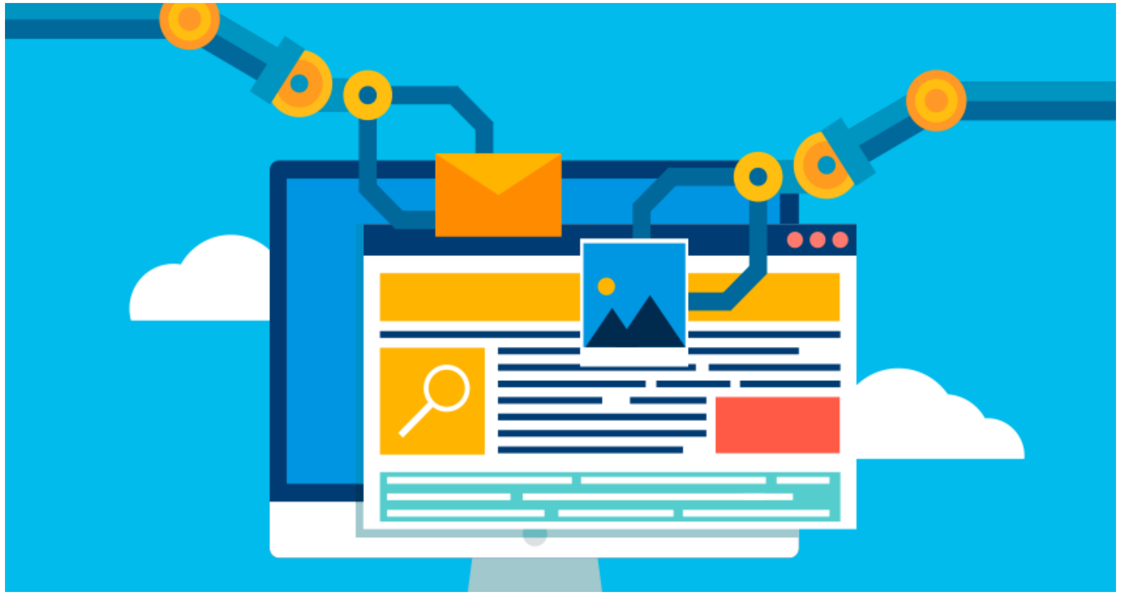
February 3, 2020

The Case for On-Time Software Delivery

Being agile as well as on-time is essential to modern software development:

- ✓ Improved customer satisfaction
- ✓ Delivery speed-up

On-time delivery is a *complex* problem, dependent on previous performance and factors affecting the timeliness of deliveries.



Related Work

- Focus on different estimation methods: models and process (agile versus traditional)
- Few studies on factors affecting on-time delivery (other than estimation model & process)

Questionnaire-based studies

1. Changing requirements
2. Unplanned work
3. Underestimated complexity

↳ Project managers!

Regression-based studies

1. Project factors (size, domain)
2. Participation of estimator
3. Personnel factors

↳ Pre-defined factors

Research Goal

This study attempts to identify and quantify the factors affecting the predictability of software deliveries.

Such insights can help us to better understand what data and techniques are needed to become more predictable.

Research Questions

RQ1: What factors affect predictability?

RQ2: Who (in terms of role and experience) are more accurate at predicting software delivery dates?

RQ3: How can teams improve their predictability?

RESEARCH CONTEXT

Research Context: ING

Since 2018, ING is actively trying to improve their on-time delivery in terms of epic deliveries from 66% (2016) to 80% (2021).

- Agile transformation
- DevOps teams
- Continuous delivery pipeline

Epics, features, user stories

Epics planned by tribe leads, area leads and product owners in QBR

User stories planned by squad members together with product owners (planning poker or analogy)

RESEARCH METHOD

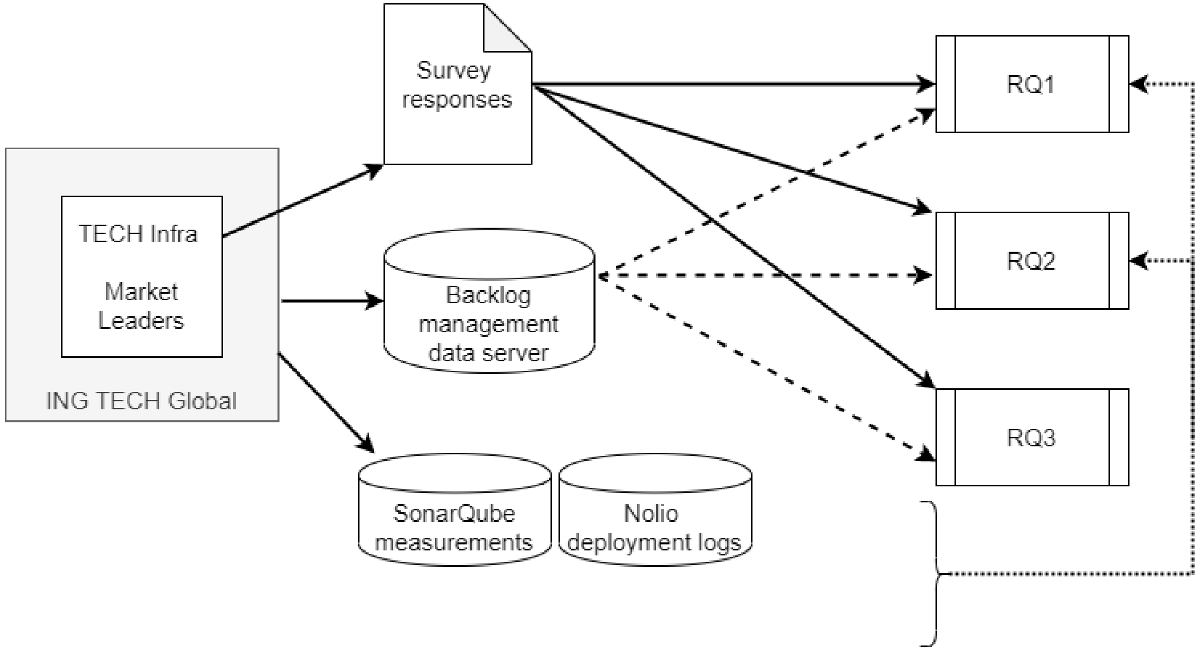
A Case Study at ING

Mixed-methods Approach

Two surveys answered by 631 participants

An analysis of 3 years of backlog data (*ServiceNow*)

An analysis of 3 years of CDaaS data (*SonarQube, Nolio*)



Survey Data

Survey Contents

The surveys consisted of open-ended questions mixed with multiple choice and Likert-scale questions.

Iterative Survey Design

Survey 1: Collect factors
Survey 2: Rank impact of factors

Participants

2200 candidates at ING Market Leaders and 600 at TECH Infra. Including: Tribe leads, Area leads, Product Owners, Agile Coaches, Squads.

- Survey 1: 296 responses (21%),
- Survey 2: 335 responses (24%)

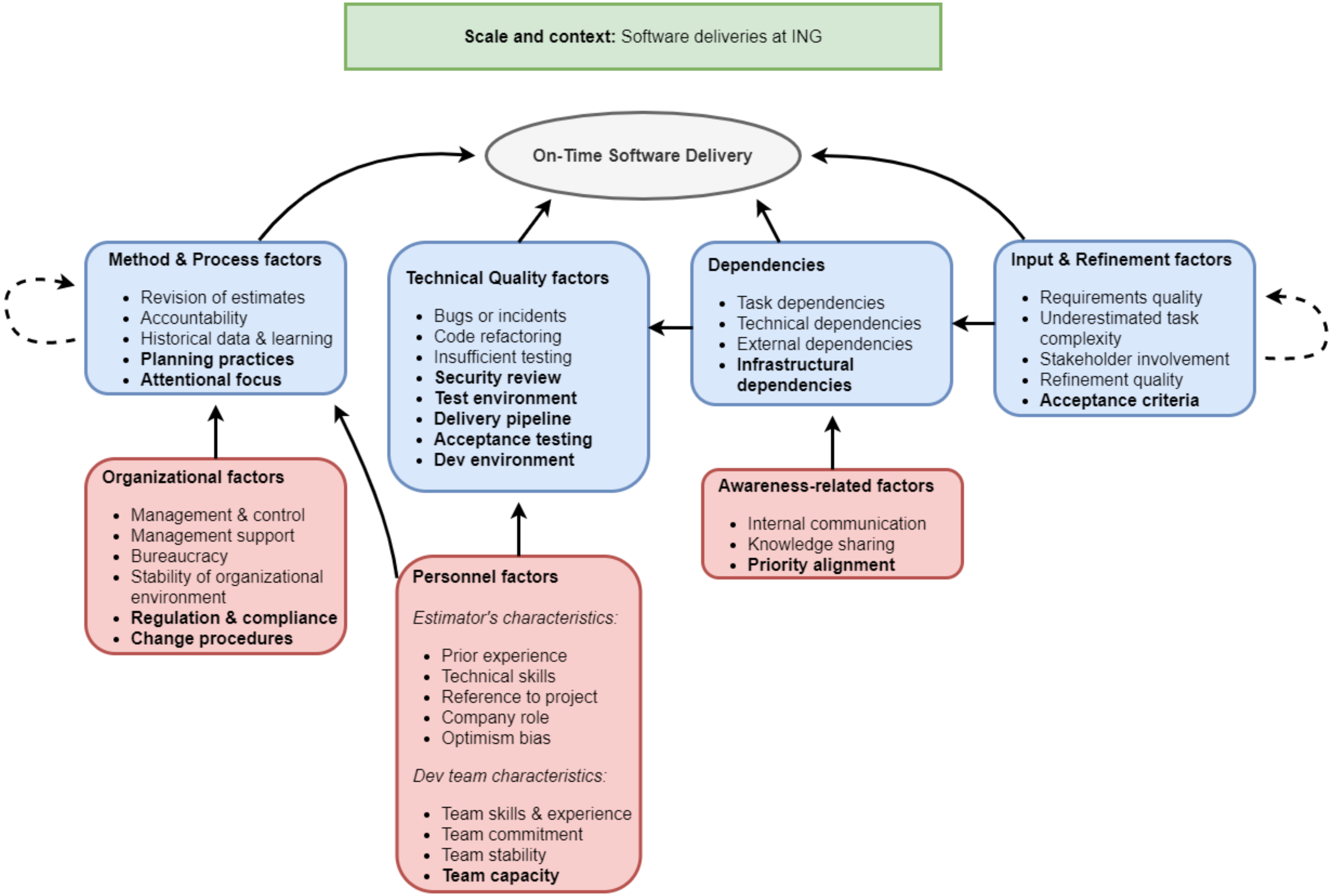
Analysis

We performed **manual coding** to summarize the results of the open-ended questions during two integration rounds.

TU Delft ING

RESULTS RQ1

RQ1: What factors affect predictability?



Scale and context: Software deliveries at ING

On-Time Software Delivery

Method & Process factors

- Revision of estimates
- Accountability
- Historical data & learning
- **Planning practices**
- **Attentional focus**

Technical Quality factors

- Bugs or incidents
- Code refactoring
- Insufficient testing
- **Security review**
- **Test environment**
- **Delivery pipeline**
- **Acceptance testing**
- **Dev environment**

Dependencies

- Task dependencies
- Technical dependencies
- External dependencies
- **Infrastructural dependencies**

Input & Refinement factors

- Requirements quality
- Underestimated task complexity
- Stakeholder involvement
- Refinement quality
- **Acceptance criteria**

Organizational factors

- Management & control
- Management support
- Bureaucracy
- Stability of organizational environment
- **Regulation & compliance**
- **Change procedures**

Personnel factors

Estimator's characteristics:

- Prior experience
- Technical skills
- Reference to project
- Company role
- Optimism bias

Dev team characteristics:

- Team skills & experience
- Team commitment
- Team stability
- **Team capacity**

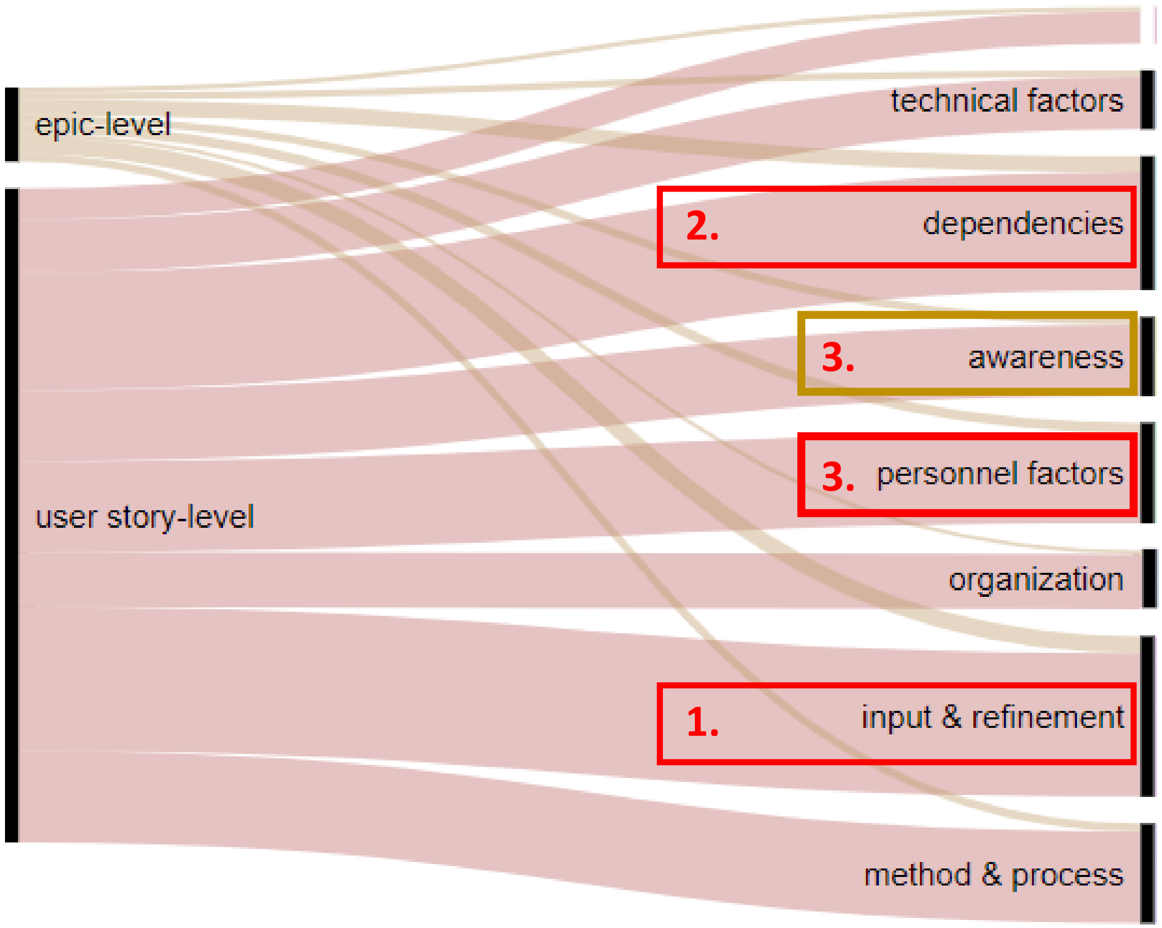
Awareness-related factors

- Internal communication
- Knowledge sharing
- **Priority alignment**

Respondents identified **35** factors in **7** categories

Estimation method was reported by only 3%

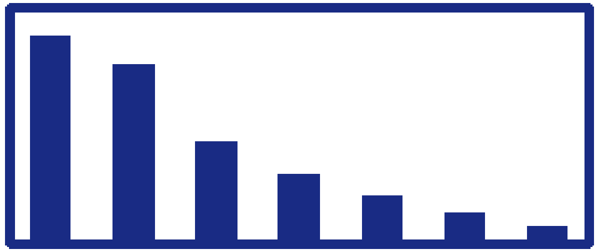
RQ1: What factors affect predictability? At estimation level



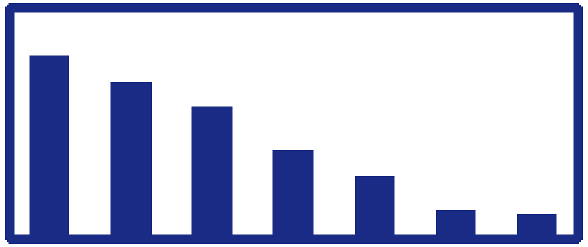
Team factors are reported by a significantly larger number of respondents estimating at the story-level

Chi-squared test,
p = 0.029,
Cramer’s V = 0.138

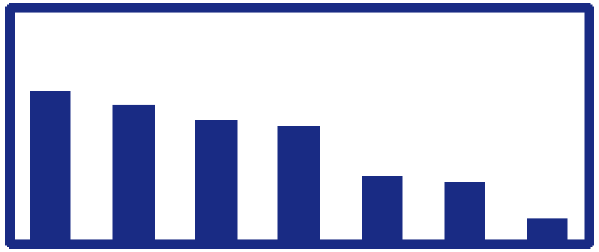
RQ1: What factors affect predictability?



(1) Dependencies: 75%

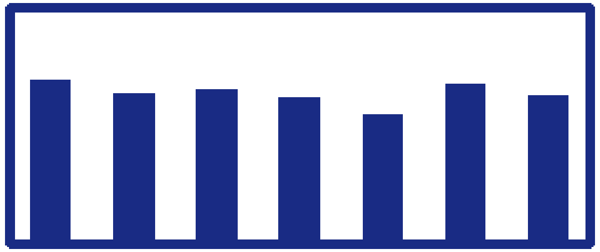


(2) Technical quality: 70%

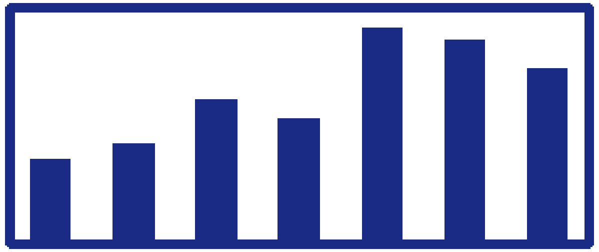


(3) Refinement: 60%

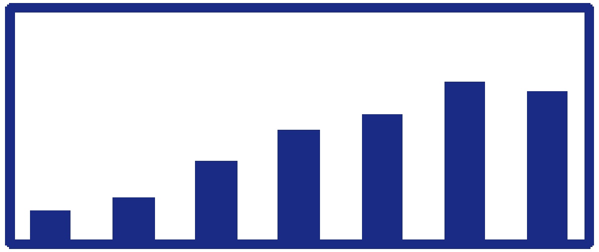
RQ1: What factors affect predictability?



(4) Organization: 45%



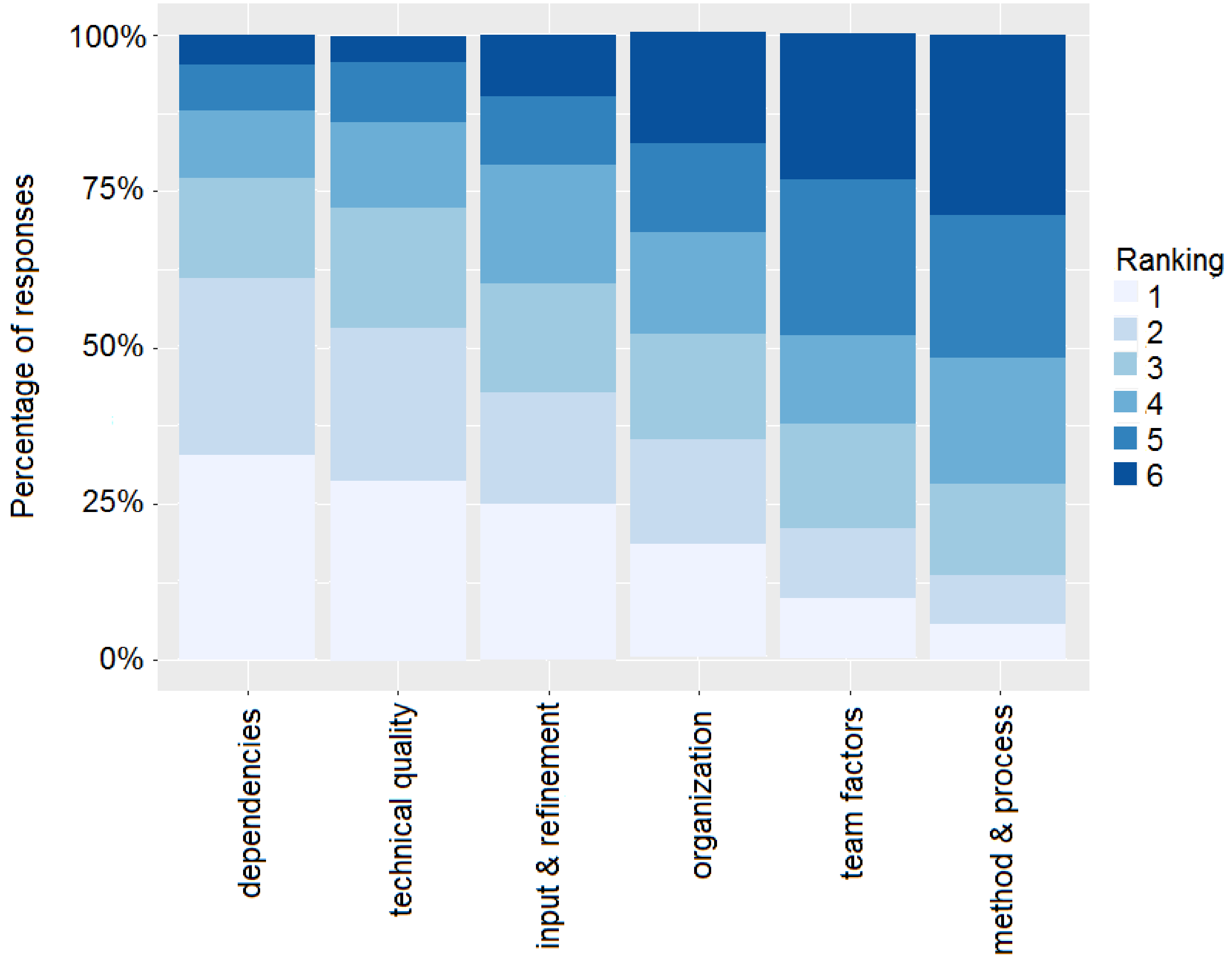
(5) Personnel factors: 31%



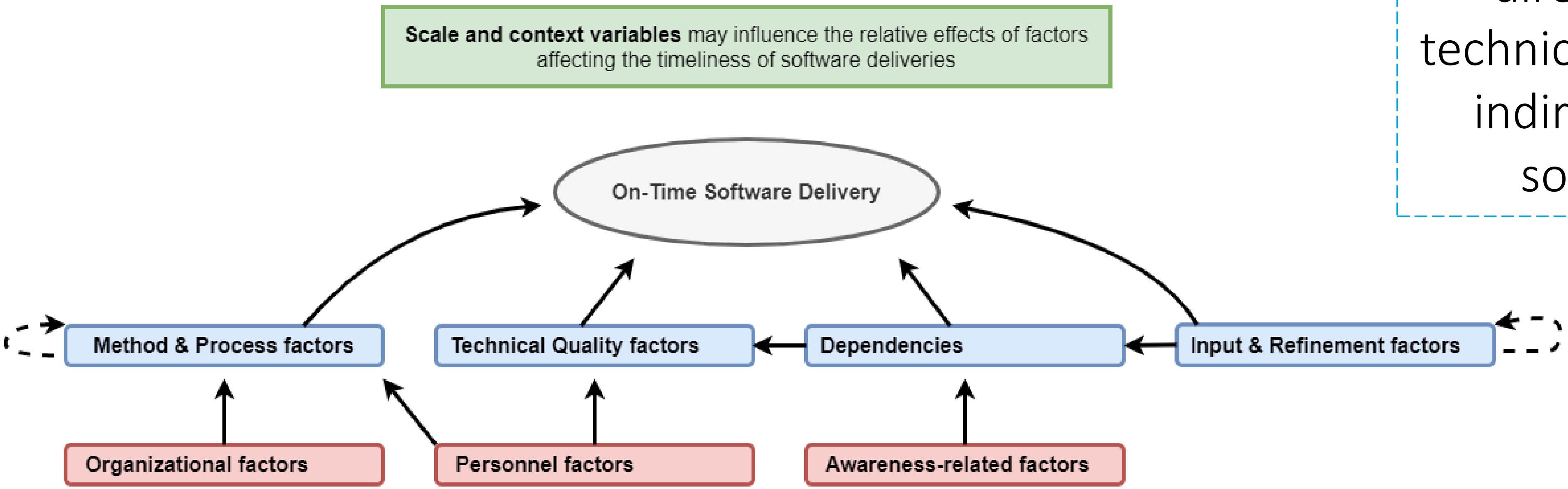
(6) Method: 22.5%

RQ1: What factors affect predictability?

- Dependencies, technical quality and refinement-related factors are perceived to be the **most influential** factors.
- Rankings of **input & refinement** and **technical quality** are significantly different based on **role**: architects, area leads and chapter leads ranked these factors the highest.



RQ1: What factors affect predictability?


Scale and context variables may influence the relative effects of factors affecting the timeliness of software deliveries
On-Time Software Delivery
Method & Process factors
Technical Quality factors
Dependencies
Input & Refinement factors
Organizational factors
Personnel factors
Awareness-related factors

Conceptual model with 4 directly influencing technical categories and 3 indirectly influencing social categories

RQ1: What factors affect predictability?

765,000 user stories and 10,000 epics over 2017 - 2019

- **Domain**
  Tribe, Theme, Type (business or architecture)
- **Input & Refinement**
  # Updates of epic/ story description, Template story, Acceptance criteria
- **Method & Process**
  Planned Duration, Planned Effort, Unplanned Effort, SprintLength, Avg Story Size
- **Technical Quality**
  # deployments (TST/ACC/PRD), Failed Test Ratio, # reported bugs, quality metrics
- **Team Factors**
  Team size, Avg ING Experience, Duration of existence, Team Stability Ratio, Historic Predictability
- **Dependencies**
  # Epic relations, # Squads working on an epic
- **Organizational factors**
  # Operational incidents

RQ1: What factors affect predictability?

Stepwise linear regression with backward elimination

Relative Error (RE) =

(Actual Effort – Estimated Effort)/ Actual Effort

Positive value corresponds to an under-estimate,

Negative value to an over-estimate.

Magnitude of Relative Error (MRE) = abs[RE]

Results coming soon!

RESULTS RQ2

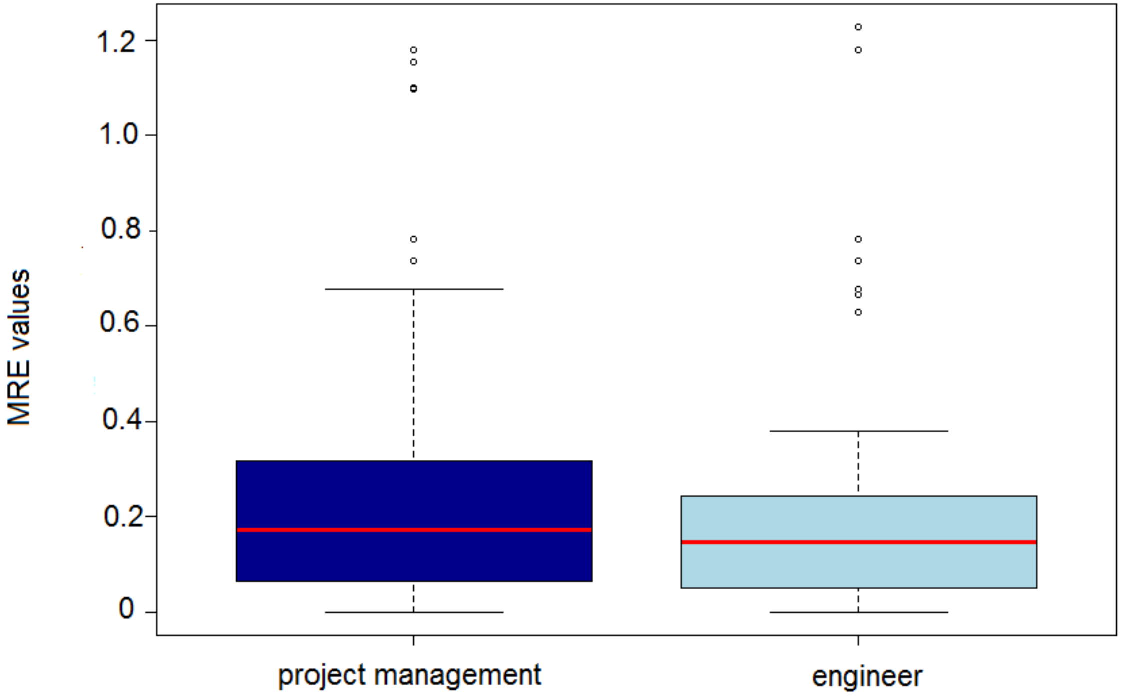
RQ2: Who are better at predicting delivery dates?

- Experiment with 305 engineers for their epics in the next quarter

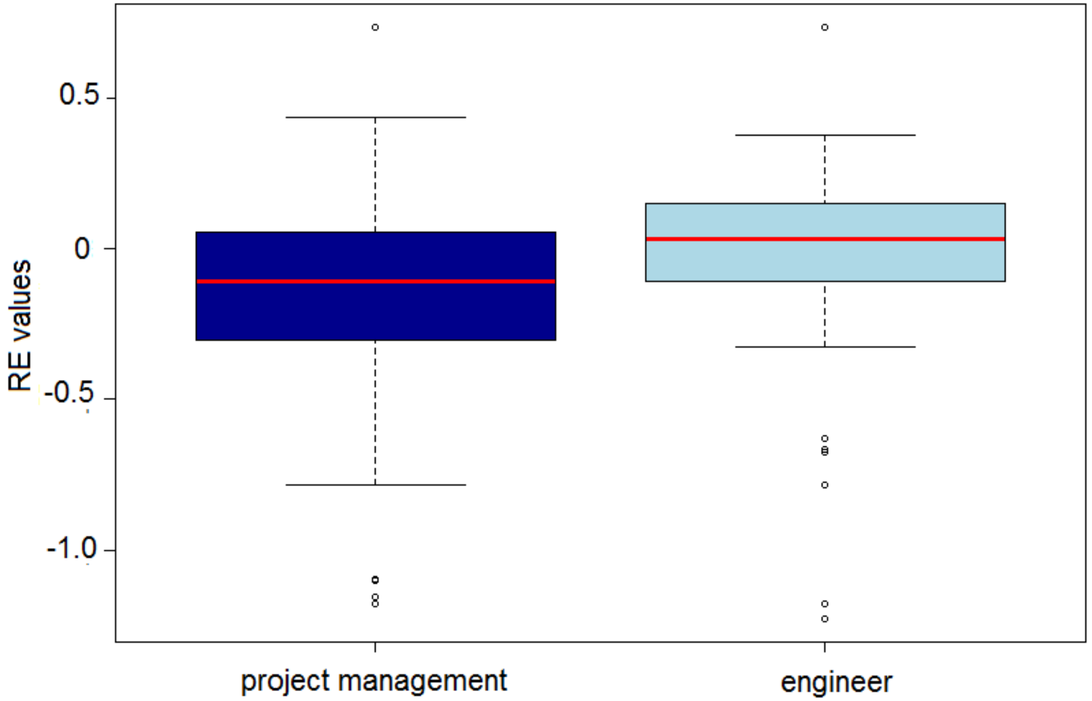
- Epics in 2019 Q3 + Q4

- Statistical analysis of the accuracy of dates estimated by *project management* versus *engineers* at different experience levels

RQ2: Who are better at predicting delivery dates?

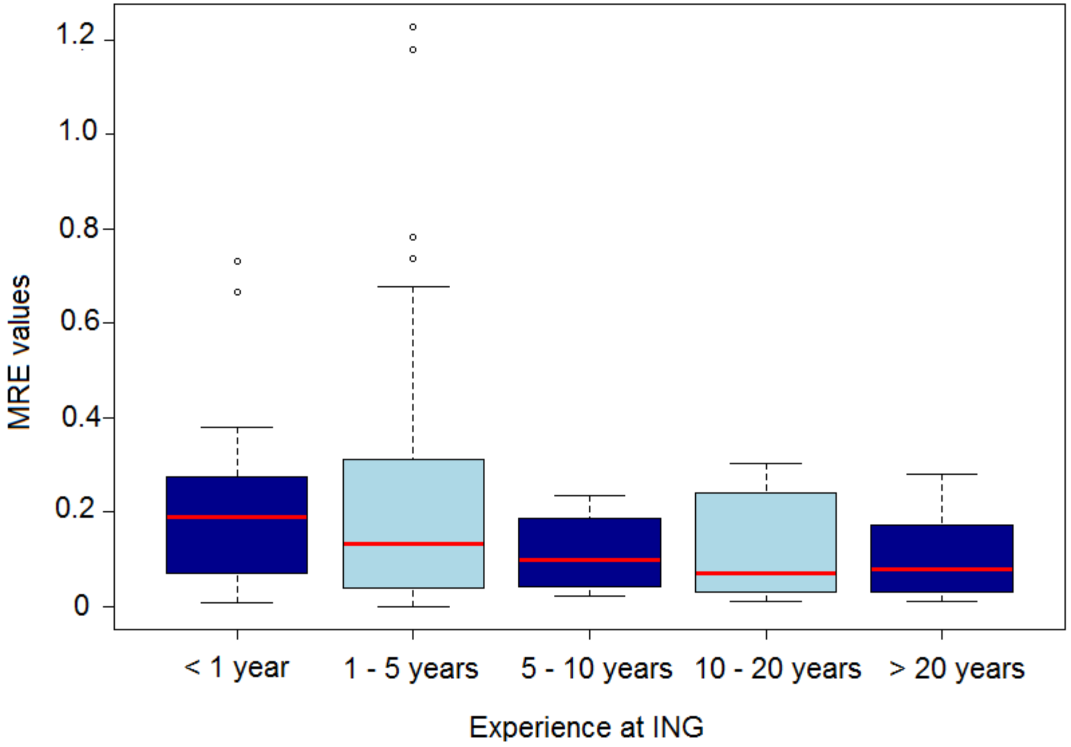


No significant difference in MRE scores

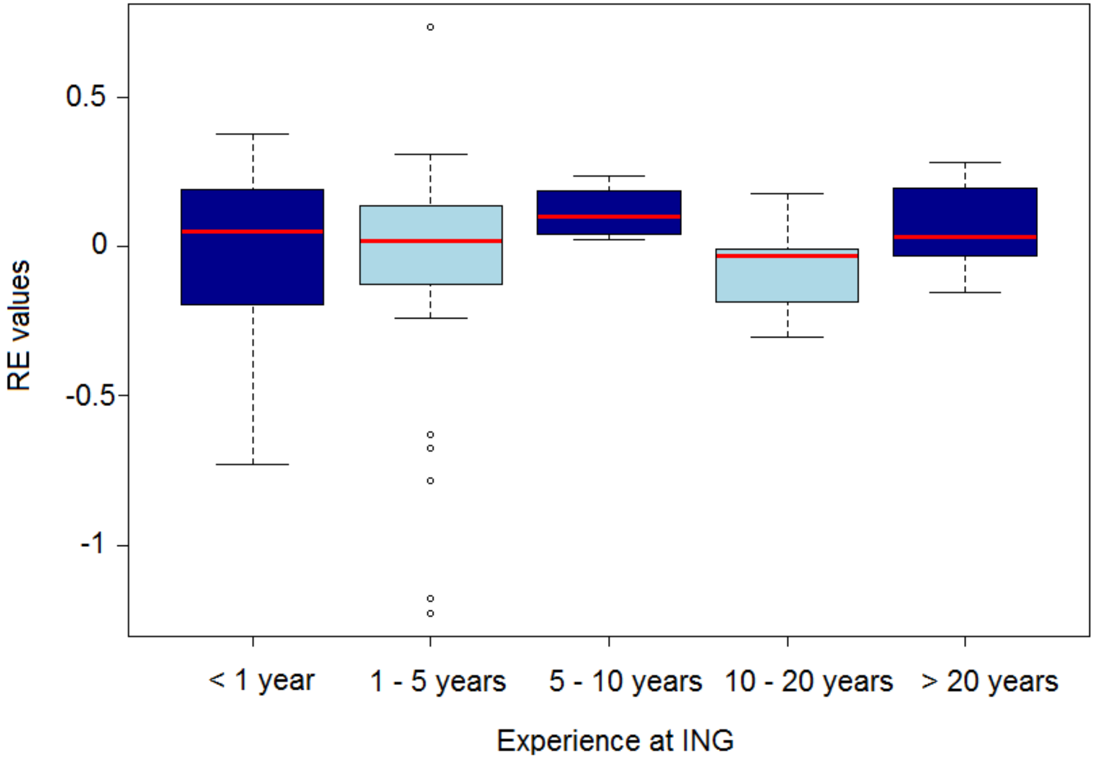


Significant difference in RE scores: engineers slightly underestimate while PMs tend to overestimate (p = 0.03, Cliff’s delta = 0.28)

RQ2: Who are better at predicting delivery dates?

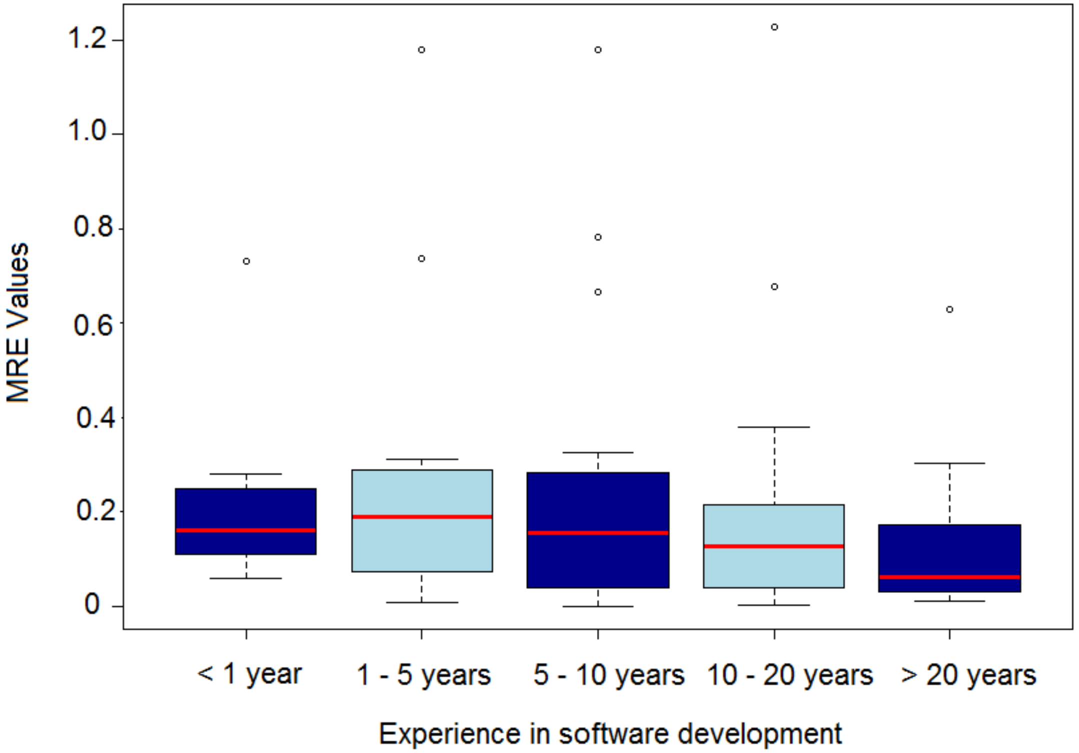


Engineers with experience at ING > 10 years estimate significantly better (p = 0.03)

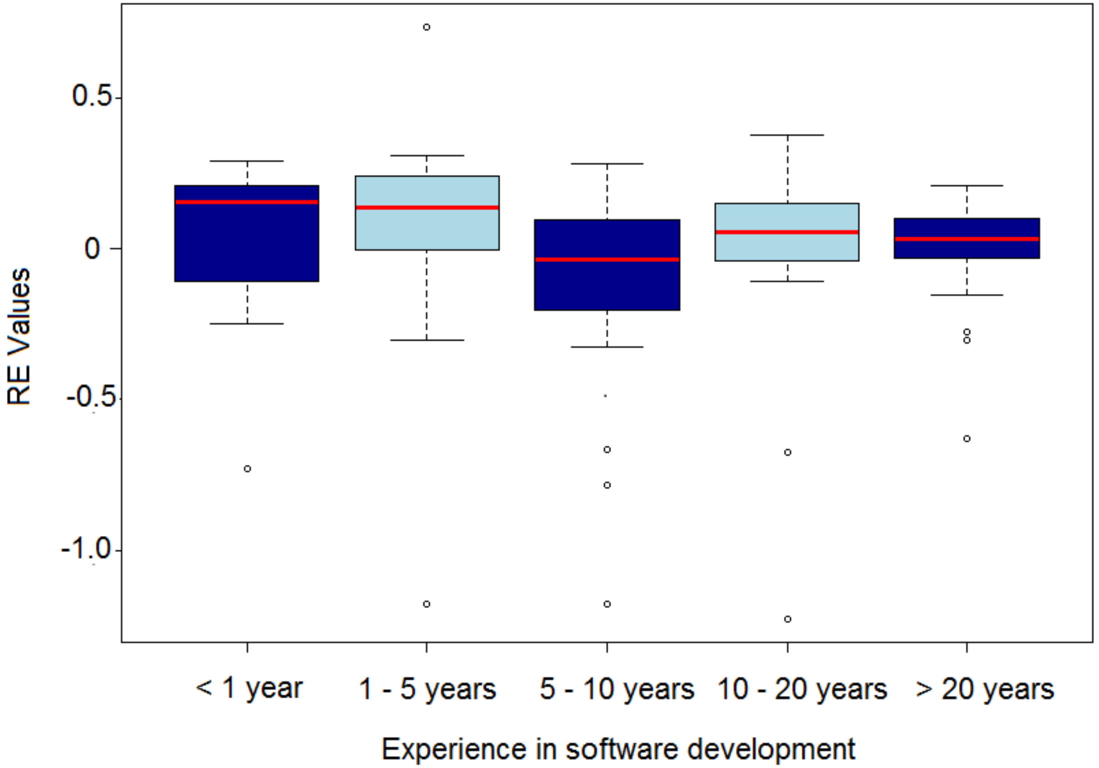


No significant difference: all engineers have a tendency to underestimate

RQ2: Who are better at predicting delivery dates?



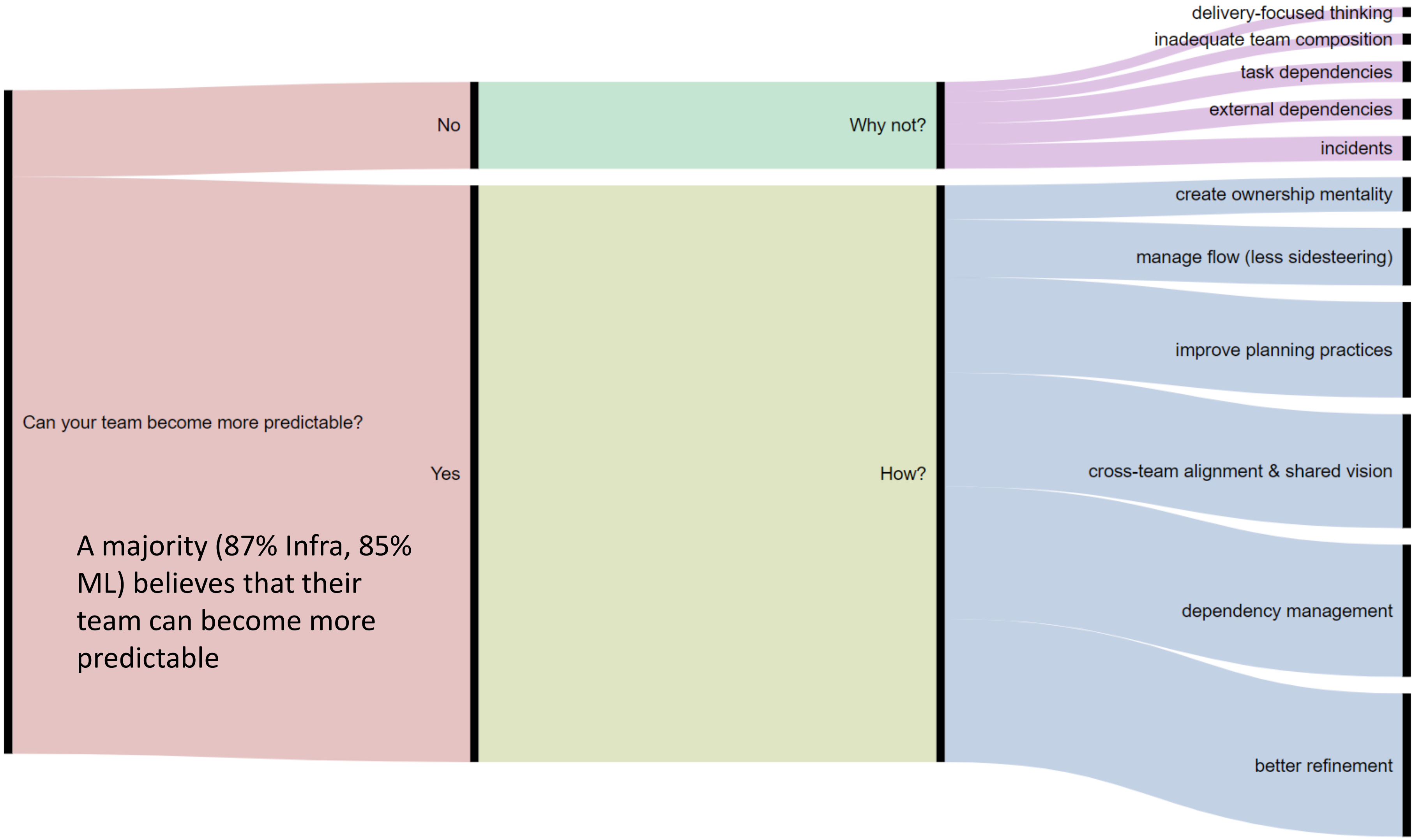
Engineers with experience in SE > 5 years estimate significantly better (p = 0.02)



Engineers with 5 -10 years have a significantly smaller tendency to underestimate

RESULTS RQ3

RQ3: How can teams become more predictable?



A majority (87% Infra, 85% ML) believes that their team can become more predictable

Stick to the committed effort, more discipline

Less pressure to meet deadlines, more focus on quality and business value

Apply WIP limit, slice large user stories, plan buffer size, have clear deployment timelines

Create shared vision and strategy, work on shared goals, better cross-team communication

Reduce dependencies through joined planning sessions and better dependency management (dependency call graphs)

Better refinement and clearer acceptance criteria, involve the stakeholder early on

FUTURE WORK

Next steps

Writing for publication!

Follow-up study on ML-based effort estimation tool and incident impact prediction