Estimating software tasks: Human behaviour + Data

Derek M. Jones
<derek@knosof.co.uk>

Introduction

Pre-2019 public effort estimation data

- *Software effort estimation is mostly fake research*
  https://shape-of-code.com/2021/01/17/software-effort-estimation-is-mostly-fake-research/
- mostly bids for work, a few internal estimates, e.g., Hatton
- datasets mostly less than 100 rows, a few have several hundred rows

Post-2019 public effort estimation data

- Data discussed in this talk
- task estimated at few hours/days
- datasets containing thousands of rows

Post-2019 data

"A conversation around the analysis of the SiP effort estimation dataset"

- 10,100 unique tasks
- 8,252 completed unique tasks
- 22 developers
- 20 internal projects

"The CESAW dataset: a conversation"

- 61,817 tasks
- 45 external projects

Blog

- http://shape-of-code.coding-guidelines.com

Estimator incentives

Bidding for work

- competing against others
- bid low to win the contract, recoup by charging for unplanned work
- "I believe planners and consultants in general deliberately underestimate project costs because their political bosses or clients want the projects. Sometimes, to tell the truth is to risk your job or your contracts or the next contract…"
  Bent Flyvbjerg "How planners deal with uncomfortable knowledge: The dubious ethics of the American Planning Association"

Asked, by manager, to do the work

- not usually a competitive environment, internal project
- if estimate not accepted, other work available
- impress management, e.g., estimate high, deliver under budget

Recurring behaviors

Use of round numbers

Consistent over/under estimation

Accuracy factors

Waiting times

Round numbers

- communicate an approximate value and level of accuracy
- divisible by two or five, powers of ten
- estimate in larger units and back calculate

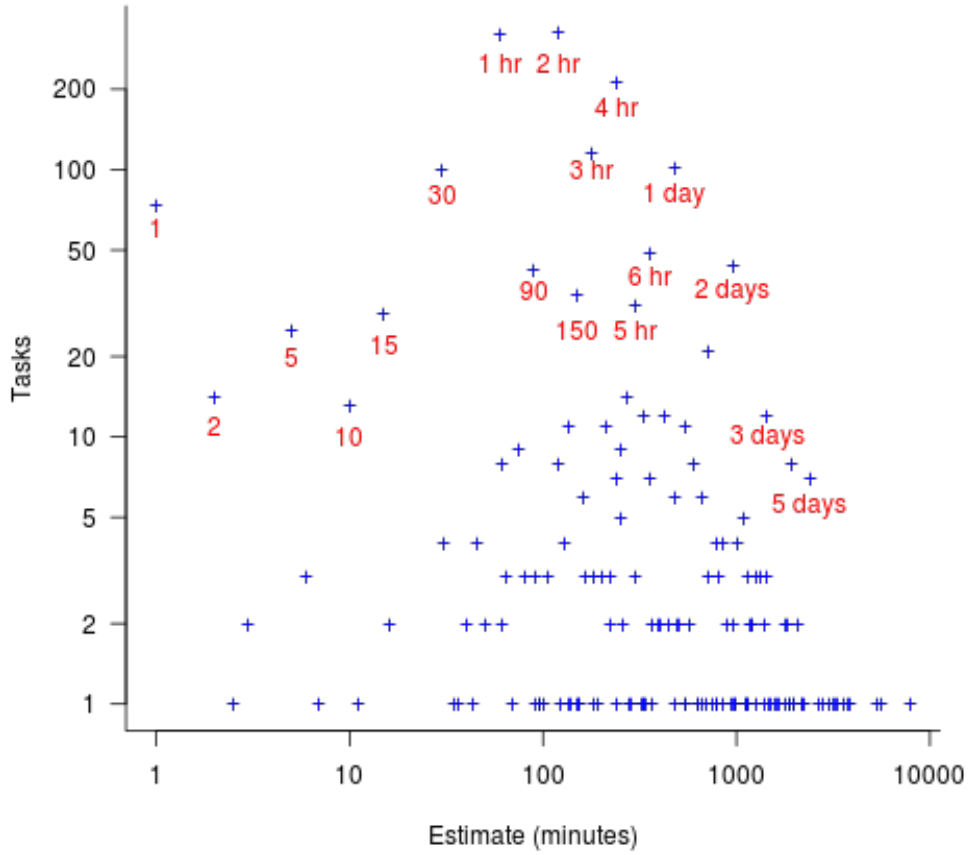

Figure 1. Number of tasks estimated to take a given time; total 1,945 tasks

Developer risk preference

- consistent under/over estimation

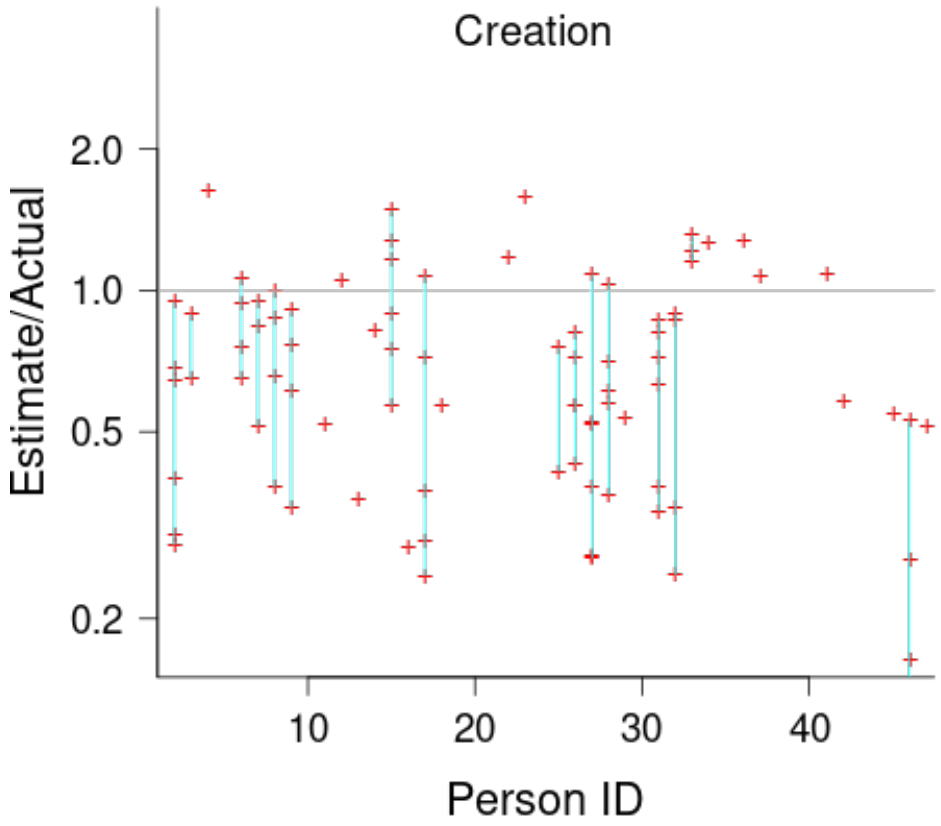

Figure 2. Individual relative estimate/actual ratio for 7 creation tasks (CESAW project 615)

Accuracy intervals

- 30% accurate
- 66% within a factor of two
- 95% within a factor of four

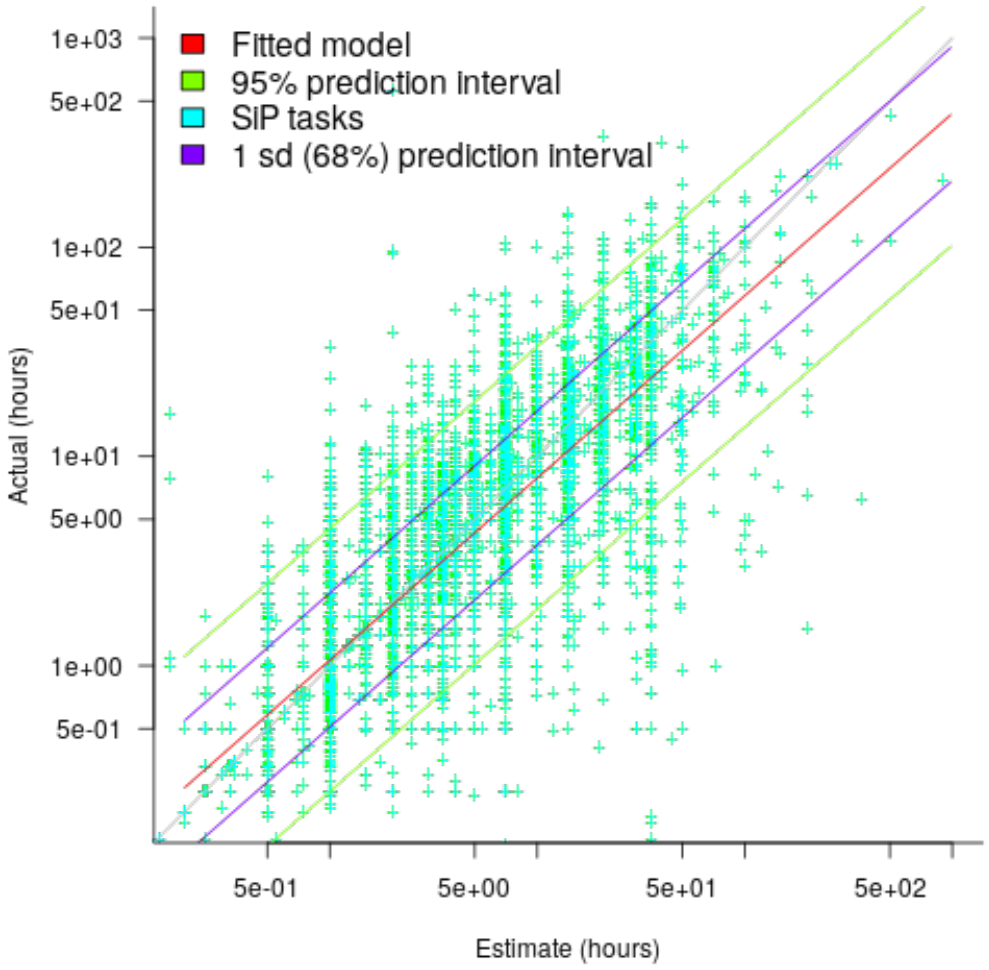

Figure 3. Estimate/Actual 9k+ software tasks, with fitted regression line+confidence bounds

Waiting time

Waiting time for tasks in a priority queue is a power law

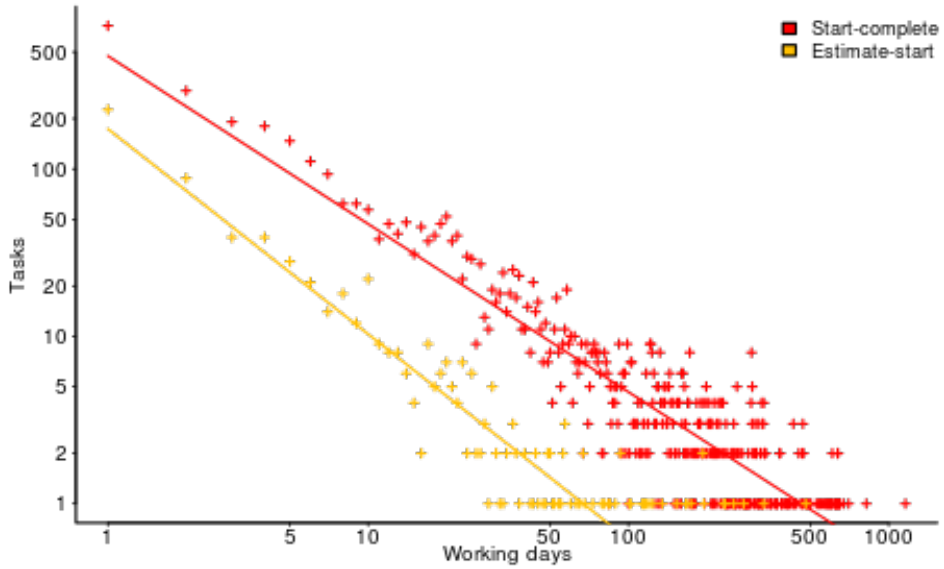

Figure 4. Power law fitted to waiting times (exponent -1)

Book

Evidence-based Software Engineering based on the publicly available data pdf+code+all data freely available *http://knosof.co.uk/ESEUR*

Analyse your data?

- Do you have any human related software engineering data?
  Jira repo, project schedules, etc
- Free analysis of your data
  Provided I can publish an anonymized version of the data
  Renzo's Pomodoro data *https://shape-of-code.com/2019/12/15/the-renzo-pomodoro-dataset/*

- derek@knosof.co.uk
- blog: https://shape-of-code.com
- Twitter: @evidenceSE