



CSSI: Elements: Can Empirical SE be Adapted to Computational Science?

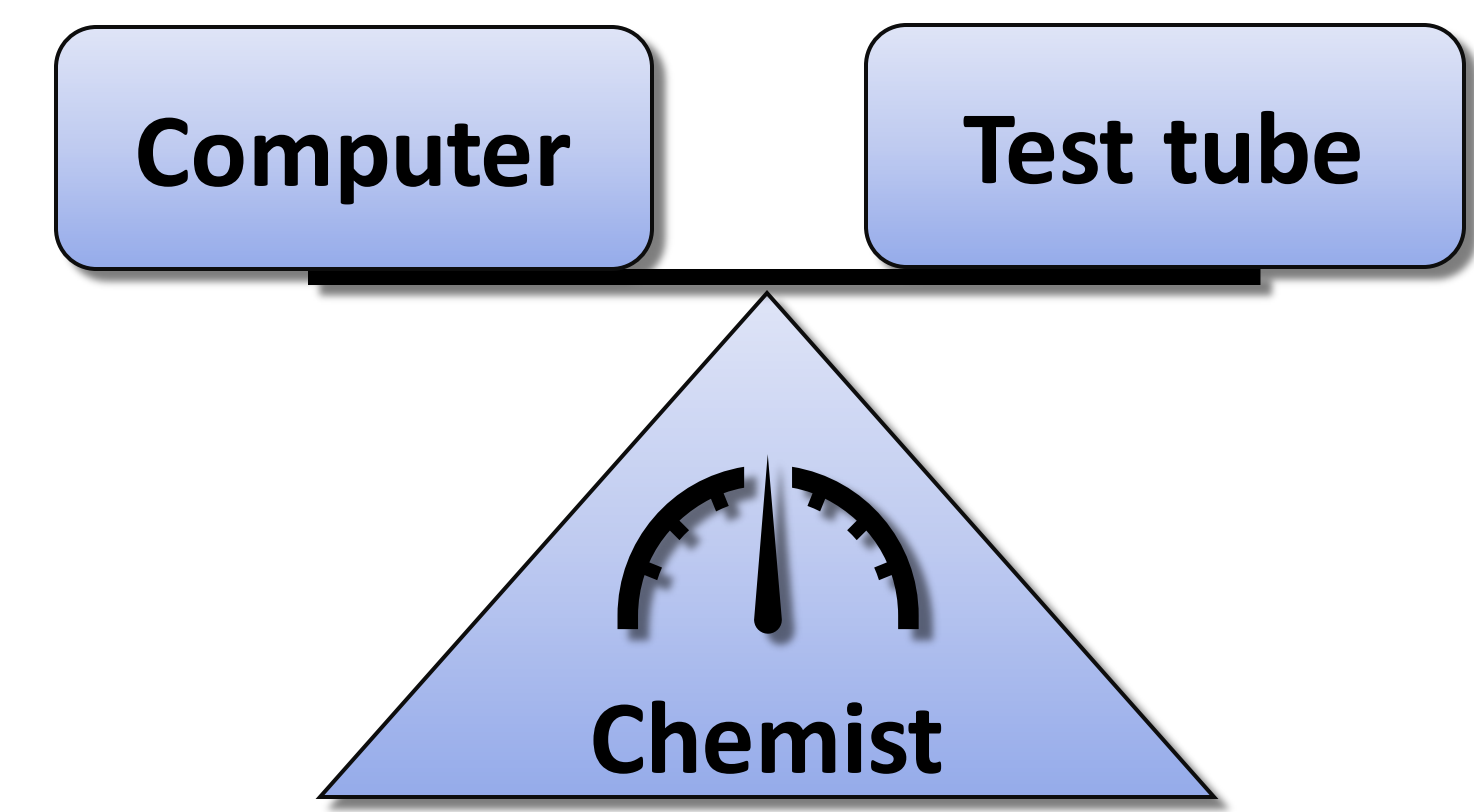
Award Number:1931425; PI: Tim Menzies <timm@ieee.org>



Authors: Huy Tu, Tim Menzies, Rishabh Agrawal, Suvodeep Majumder, Kunal Shah, Sukhad Joshi, Nagaraj Madamshetti, Shubham Dua.

Abstract:

- This study explores factors influencing successful computational science software projects, focusing on the importance of quality in software-driven fields.
- It aims to collect and test hypotheses about the success of computational science projects.



Data Collection:

- Collected **591** prominent CSc projects.
- Pruned down to **169** projects via the prudence checks.

Prudence Checks:

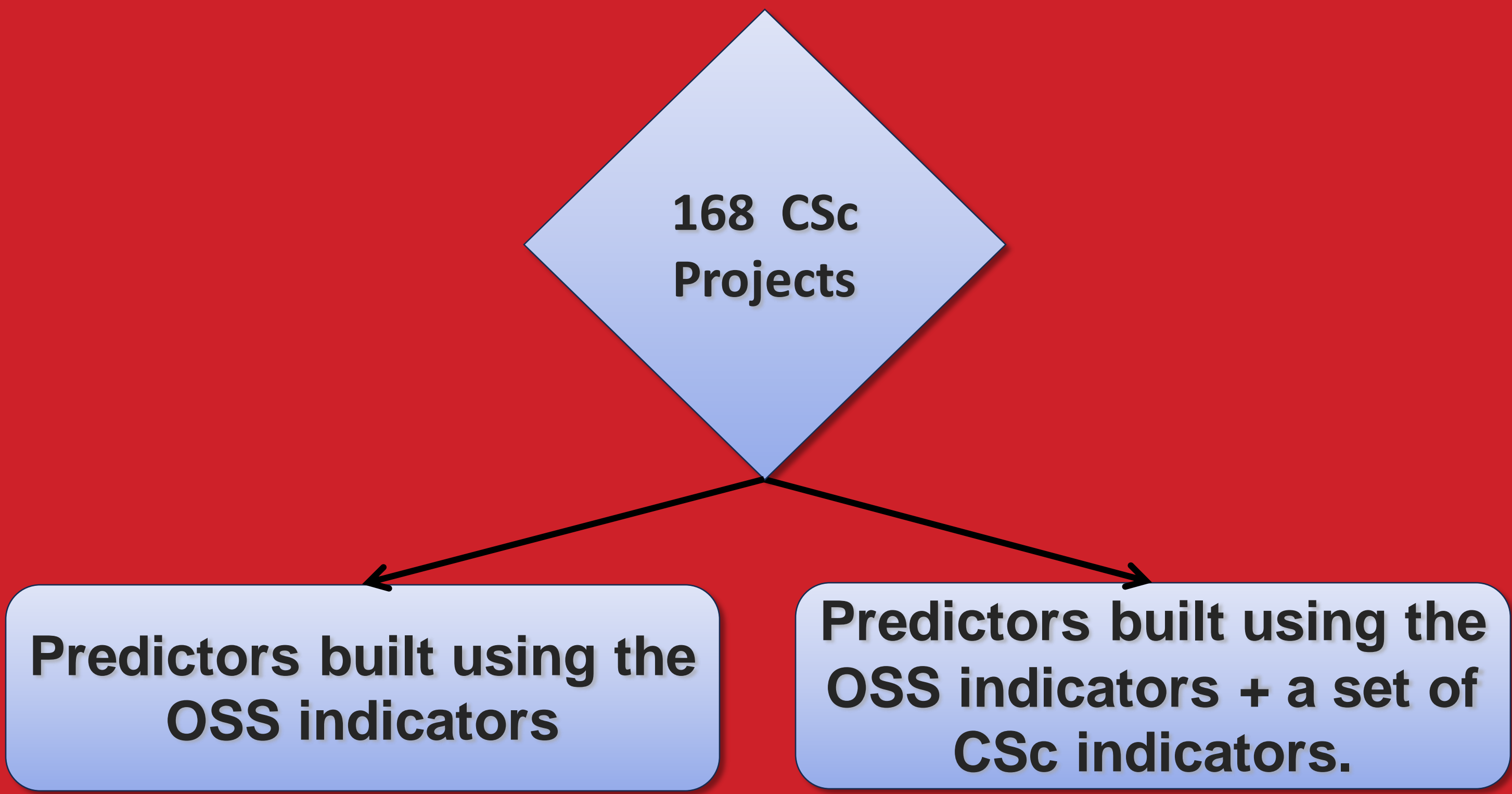
Check	Condition
# Developers	≥ 2
Pull requests	≥ 5
Issues	> 10
Releases	> 1
Commits	> 20
Duration	> 1 year

OSS Indicators:

1. # of contributors
2. # of commits
3. # of open issues
4. #of closed issues
5. # of open PR
6. # of closed PR
7. # of merged PR
8. # of forks
9. # of watchers
10. # of stargazers
11. # of issue comments
12. # of PR comments

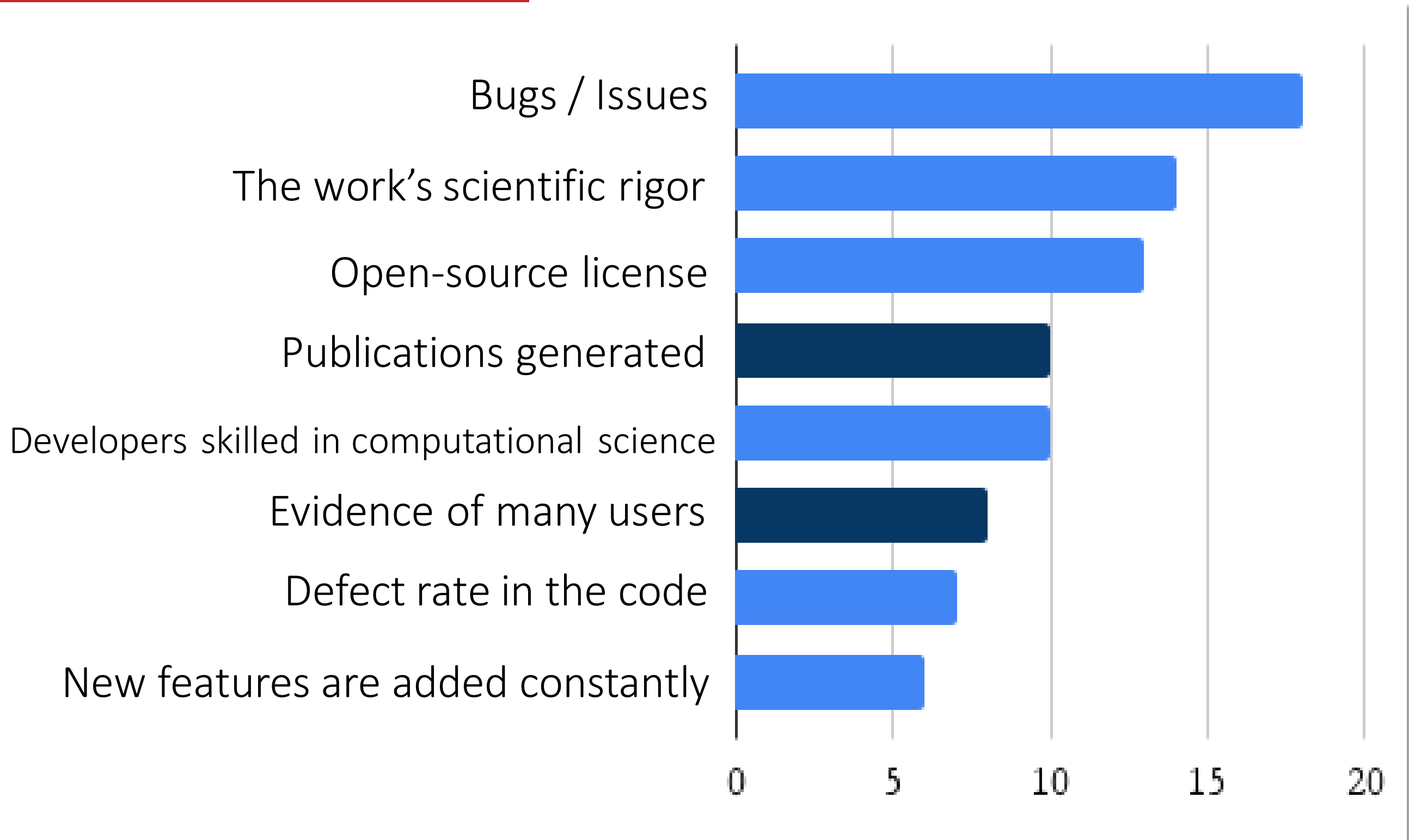
Research Questions

1. Are *Science Indicators* important for CSc software projects?
2. Is “*Goodness*” of CSc software projects measured using project *Health Indicators*?



- What are the factors that lead to a successful CSc software project?
- What makes CSc software work better?

When asked, we found **13 common assumptions** on what is "*good software*". But when we tested these on 168 GitHub projects, most of these assumptions could not predict for "good project" in 12 months time.



Findings:

- In CSc projects, control is limited; some metrics (e.g., open pull requests) had weak predictors, while others (e.g., monthly contributions) showed high accuracy (median MREs of 32% in 20 experiments). Focus on manageable issues in management actions, not all.

month_12_level_0_bellwether_prediction_for_monthly_open_PRs_SA			
rank	treatment	median	IQR
1	baseline	81	67
1	baseline.plus_expert_score	83	63
1	baseline.plus_developer_skill	83	64
1	baseline.plus_monthly_buggy_commits	84	63
1	baseline.plus_license	85	56
1	baseline.plus_monthly_features	87	59
1	baseline.plus_all_features	90	54

month_12_level_0_bellwether_prediction_for_monthly_open_PRs_MRE			
rank	treatment	median	IQR
1	baseline.plus_all_features	0	94
1	baseline	0	92
1	baseline.plus_developer_skill	0	100
1	baseline.plus_license	0	92
1	baseline.plus_expert_score	0	93
1	baseline.plus_monthly_features	0	100
1	baseline.plus_monthly_buggy_commits	0	94

Open Pull requests Stats

month_12_level_0_bellwether_prediction_for_monthly_contributors_SA			
rank	treatment	median	IQR
1	baseline.plus_monthly_buggy_commits	41	63
1	baseline.plus_all_features	42	60
1	baseline.plus_expert_score	44	63
1	baseline	46	58
1	baseline.plus_license	47	57
1	baseline.plus_developer_skill	48	54
1	baseline.plus_monthly_features	48	64

month_12_level_0_bellwether_prediction_for_monthly_contributors_MRE			
rank	treatment	median	IQR
1	baseline.plus_license	33	57
1	baseline.plus_monthly_features	32	57
1	baseline.plus_developer_skill	33	61
1	baseline.plus_expert_score	33	57
1	baseline.plus_all_features	33	61
1	baseline.plus_monthly_buggy_commits	36	63
1	baseline	38	62

Monthly Contributors Stats

- When segregating model variables into "SE measures" (e.g., developer count) and "CSc measures" (e.g., software-generated publications), no extra insights were gained from CSc-specific metrics. In essence, CSc software is just software, suggesting that CSc professionals could gain from enhancing their software engineering knowledge.

CSc Indicators:

These indicators are determined based on a survey that was conducted for CSc reserachers associated with successful CSc software projects through a designed questionnaire.
<https://forms.gle/GBbWkb1QSQ8gHLhu6>

Statistics:

- **MRE** is magnitude of relative error.
 $MRE = |actual - predicted| / actual$
- **SA** is Standardized accuracy.
 $SA = (1 - (MAE / MAE_{guess})) \times 100$
where MAE is Mean Absolute Error

We seek to:

- **Minimize MRE** since they means our predictions are closer to actuals.
- **Maximize SA** since that means we are doing better that our unsophisticated method.

Conclusion:

- It's concerning that there's significant effort in CSc without a clear understanding of success predictors.
- More research is urgently needed on CSc software.
- CSc professionals should study standard SE.
- Encouragingly, CSc software follows general software success principles.