Pomona: Continuous Code Quality Improvement via Small, Agentic Pull Requests at Bloomberg

Dave Williams
University College London

Angelos Evripiotis
Bloomberg

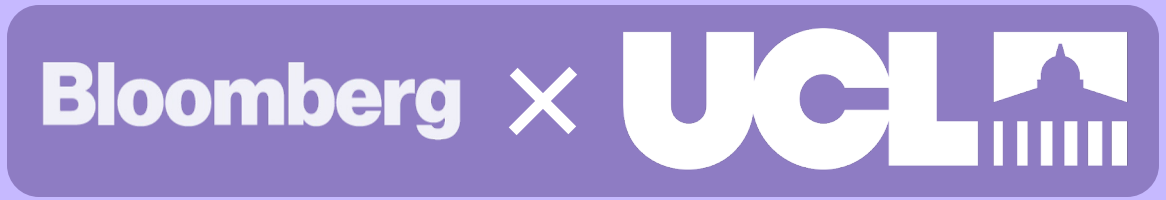
Serkan Kirbas
Bloomberg

Harry Morgan
Bloomberg

Sergey Magidovich
Bloomberg

Peter Wainwright
Bloomberg

Federica Sarro
University College London

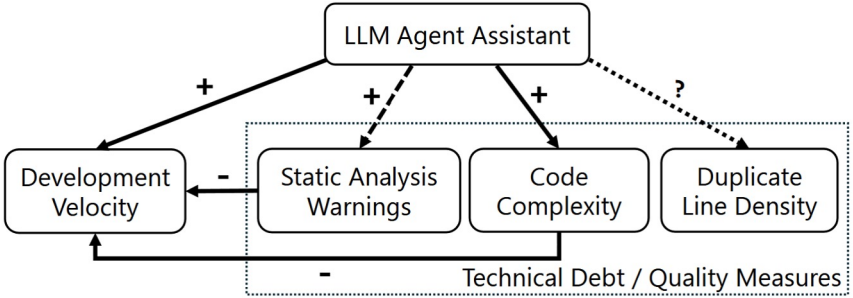


ISSRE'26 Industry Track
21/10/26 – Limassol, Cyprus

Now we've had agents for a while...

Recent work found that when open source projects adopt Cursor:

1. ↑ Work speeds up (rate of commits)
2. ↑ Code quality (Sonarqube) issues increase
3. ↓ Code quality issues slow down work



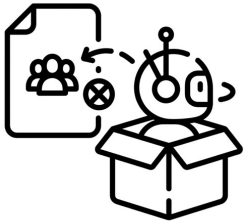
Solid lines indicate causal relationships; dashed lines indicate partial evidence; and dotted lines indicate inconclusive evidence.

[1] He et al., "Speed at the Cost of Quality: How Cursor AI Increases Short-Term Velocity and Long-Term Complexity in Open-Source Projects", MSR 2026.

Adoption Challenges (at Bloomberg)



1. Concerns about potential impacts on long-term code health



2. Out-of-the-box LLMs tend to ignore team-specific practices



3. LLM-generated code can be challenging for engineers to review

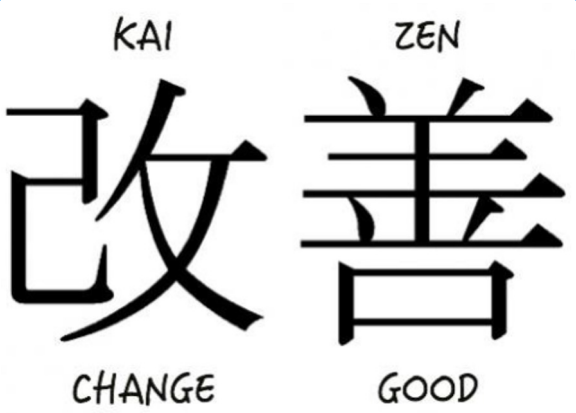
Pomona in a Nutshell

Inspired by the Kaizen™ philosophy of continuous incremental improvement.

A lightweight agentic tool for automated, small, and low-risk code quality Pull Requests (PRs).

Implemented using just 3 markdown files:

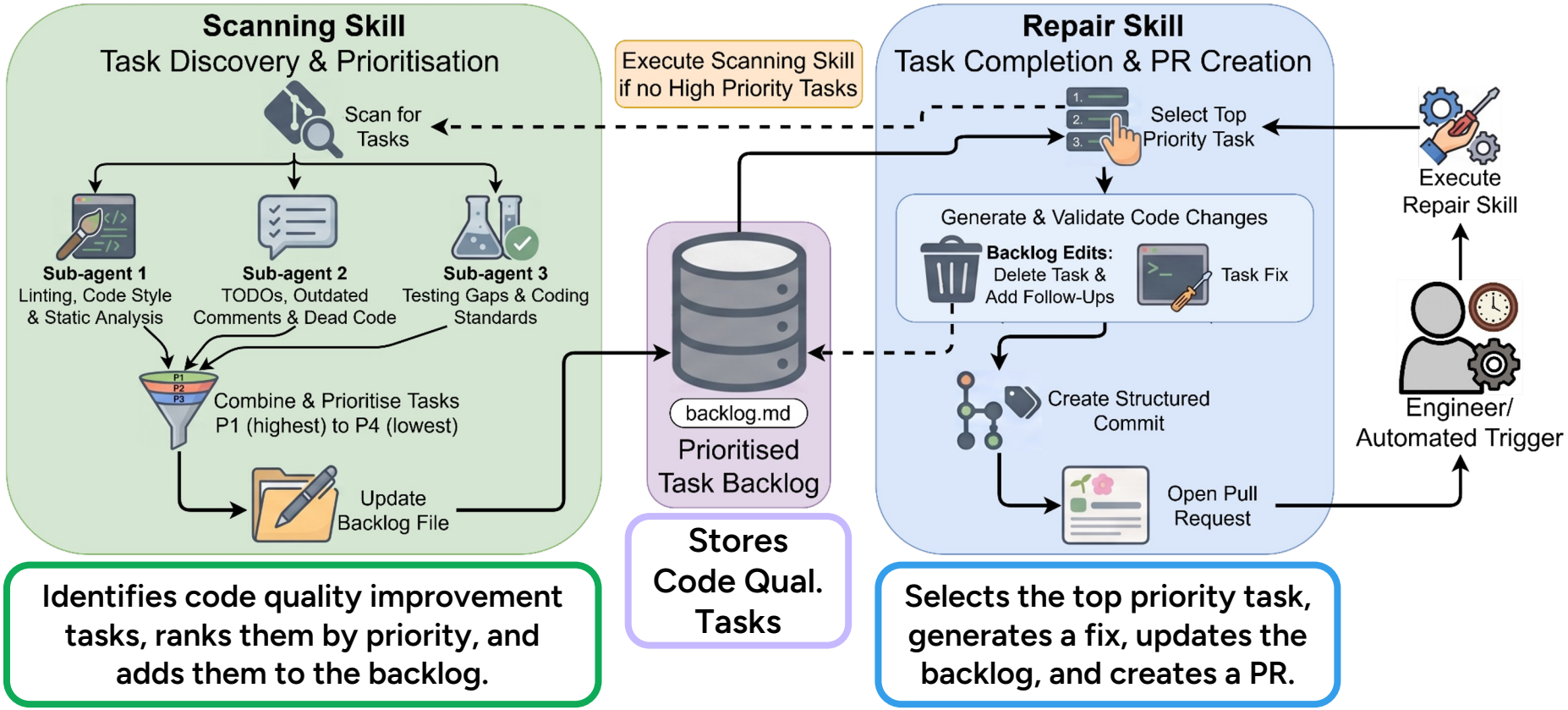
- 2 Agent Skills: *"**Scanning**"* and *"**Repair**"*
- 1 structured issue Backlog



(or gǎi shàn)

CONTINUAL IMPROVEMENT

Pomona Architecture



Structured Task Backlog

Simple Markdown list containing tasks

Tasks grouped by priority using the Benefit X Ease-of-Review matrix:

	Easy to Review	Hard to Review
High Benefit	**P1** – Top Priority	**P3** – Worth Doing
Low Benefit	**P2** – Quick Wins	**P4** – Parking Lot

P1: High benefit, easy to review

- Remove unused `min_lead_in` / `min_lead_out` params from `lead_time_stat_timeseries_repos` + 2 callers
- Remove self-documenting TODOs in `utils/agg.py` (lines 35, 45)
- Enable ruff `I` (isort) incrementally — all auto-fixable
  - `core/` (11 modules, strictest-checked — start here)
  - `utils/`
  - `pdata/`
- Enable ruff `UP` (pyupgrade) for `core/` — most rules auto-fixable

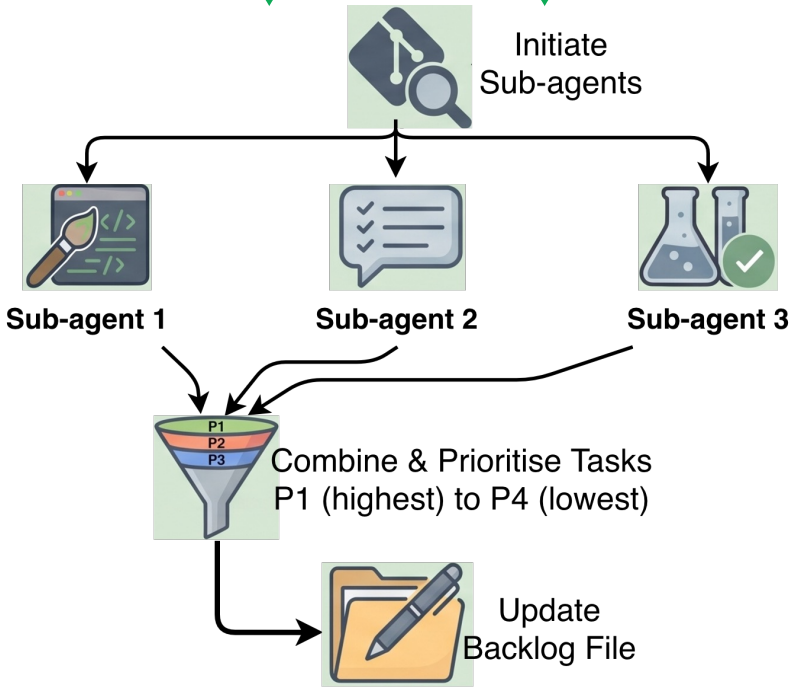
Example of P1 Tasks in Adopting Team Backlog

Scanning Skill

Goal: to fill the backlog with prioritised tasks.

In our implementation:

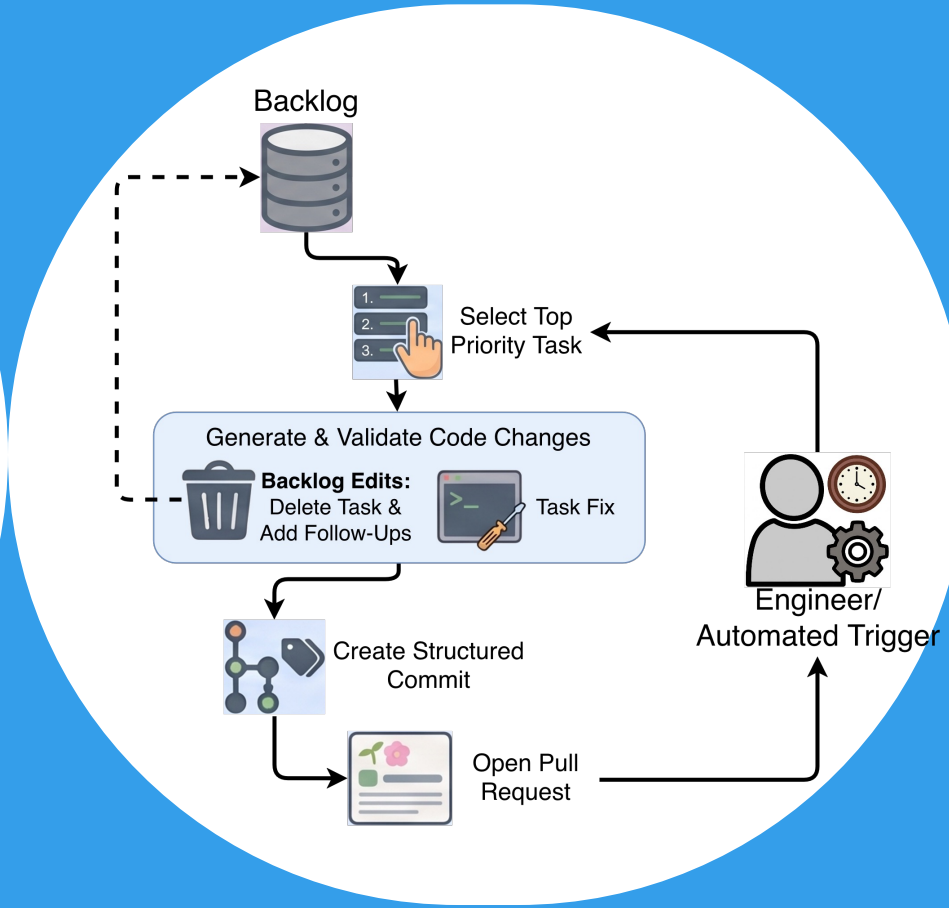
- **Subagent 1**: Mypy/Ruff rule expansions
- **Subagent 2**: TODOs, outdated comments/code
- **Subagent 3**: Repo coding standards, test gaps



Repair Skill

Run by a user or on a schedule

- Includes validation commands
- If no high-priority items exist, it runs the Scanning skill.
- Aim: PRs with ~10 lines of diff

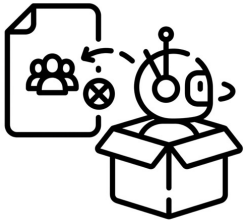


Pomona vs. Our Challenges



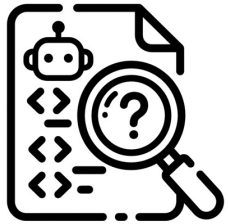
1. Concerns about potential impacts on long-term code health

Solution: Pomona's primary focus is on improving code quality/maintainability.



2. Out-of-the-box LLMs tend to ignore team-specific practices

Solution: Pomona integrates team practices for both Scanning and Repair skills.



3. LLM-generated code can be challenging for engineers to review

Solution: Pomona generates small, predictable PRs using familiar tools.

Evaluating Pomona in Practice

3-Month Team Deployment

Aiming for "about one Pomona PR per day"
Manual audit of Pomona PRs:

- Merge outcome
- Commits, comments, reviews
- Task types

Non-User Questionnaire

Distributed across DevX
Presents Pomona and elicits:

- Usage frequency
- Desired task types
- Concerns

Results

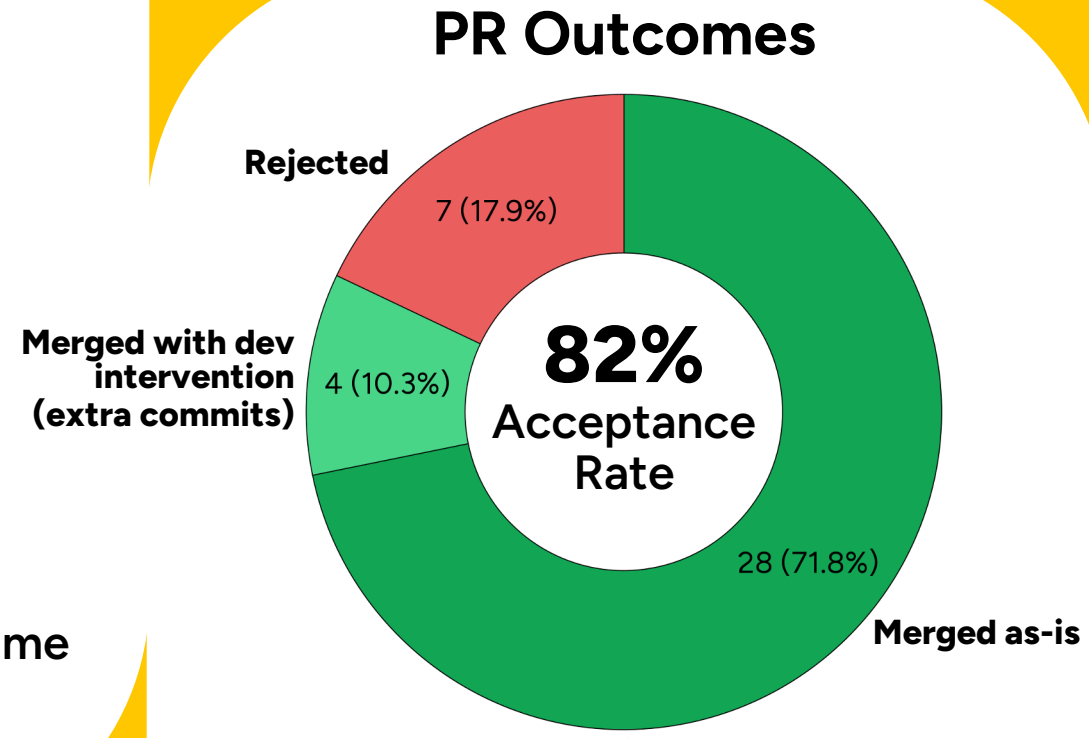
Team Deployment

39 PRs Generated

55 Workdays

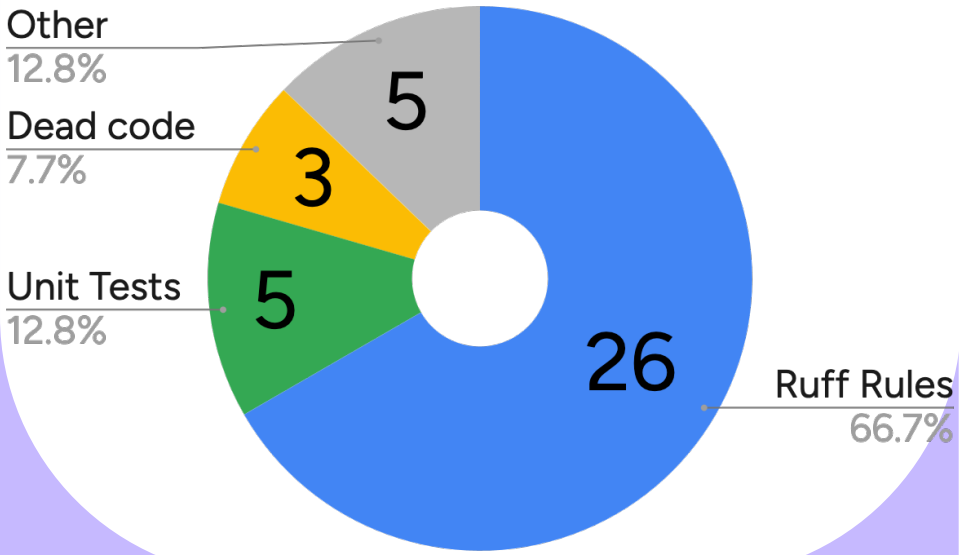
~3-4 PRs/Week

2h 14m Med. Decision Time



Results
Team Deployment

Task Types



PR Sizes

	Num. Files	Num. Lines
Range	1–35	5–168
Median	4	37
Mean	6.5	58

PR size metrics across 39 Pomona PRs

Failure Reasons (7 PRs)

- Duplicated PRs (4)
- Undesirable formatting (2)
- Unknown (1)

Results
Non-User Questionnaire

Demographics

12 Responses

11 with 4+ YoE, 7 of them with 10+

Most use AI/Agentic SE tools daily

Headlines

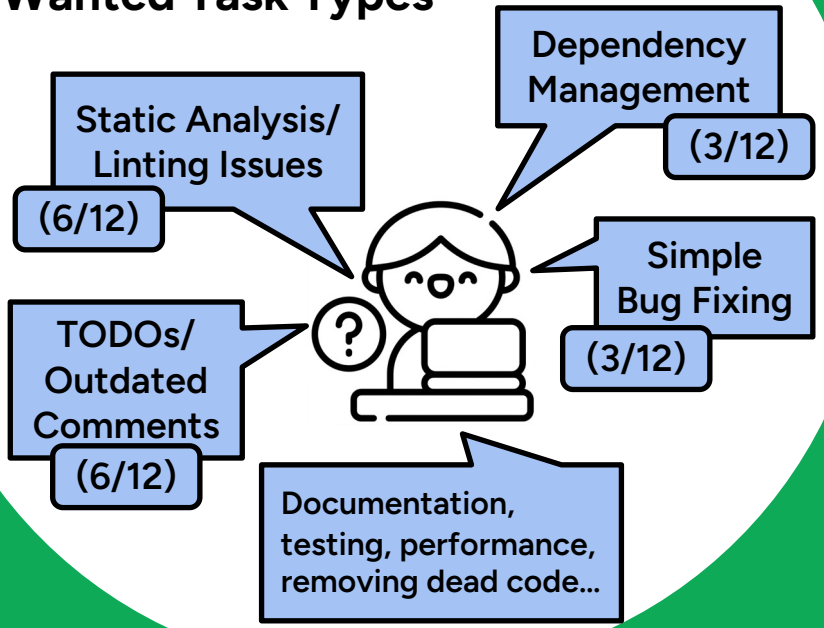
10/12 wanted to try Pomona

8/12 would review 2–3 PRs per week

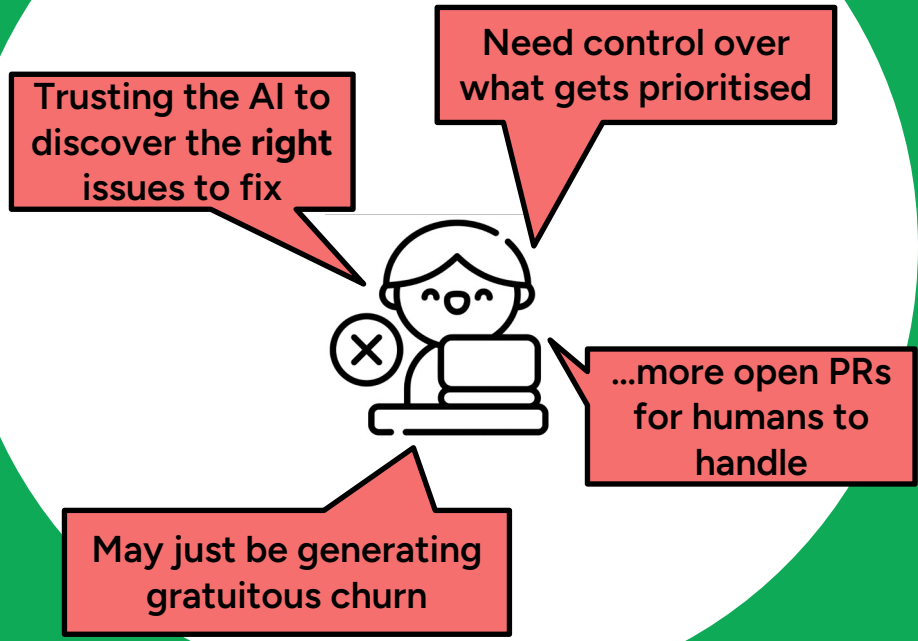
Results

Questionnaire

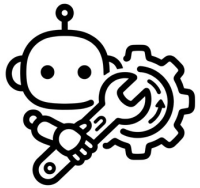
Wanted Task Types



Concerns



(A few) Lessons Learned



Agents are valuable even when they just execute other tools

Agents that run deterministic, reliable, and familiar tools produce more predictable, easier-to-review output.



Task Prioritisation is as important as identification

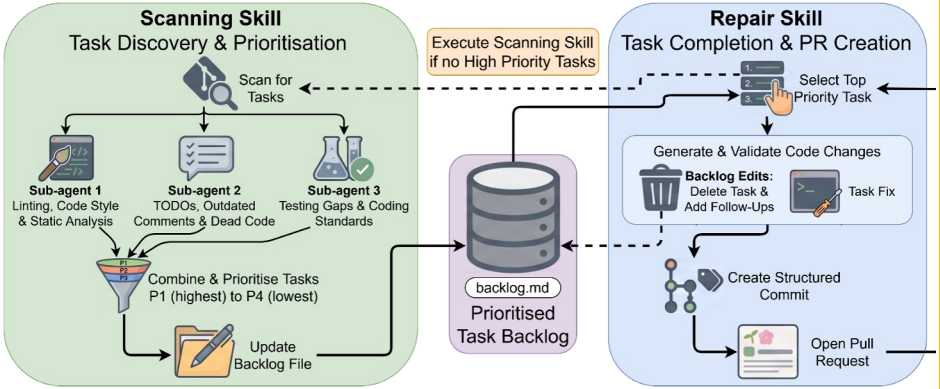
It's easy to find tasks and create work for engineers to review. Prioritising high-impact work is key.



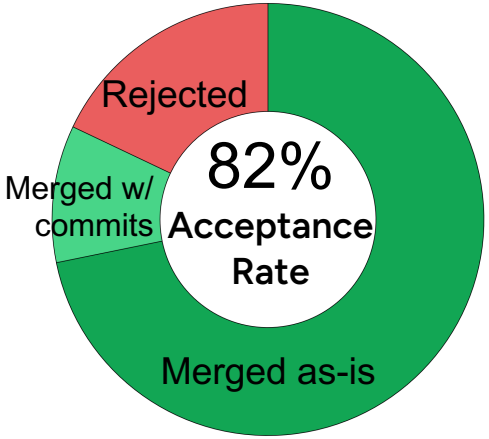
Quantifying cognitive reviewing load is valuable future work

Time-to-close only partially reflects review ease. We need to move beyond time-based metrics to quantify effort.

Pomona Architecture



PR & Questionnaire Outcomes



2h 14m
Med. Decision Time

10/12
Engineers would try Pomona

(Some) Lessons Learned

1. Agents don't have to generate to be useful.

2. Task prioritisation is as important as task identification.

3. Quantifying cognitive load is key future work.

More insights in our full paper!

Contact me!
davejjwilliams.github.io
david.williams.22@ucl.ac.uk