Dataset and Analysis for the Commit Messages of the Linux Kernel Out-of-Memory Killer

Mouna Dhaouadi
PhD Candidate
Université de Montréal, Canada
http://www-labs.iro.umontreal.ca/~dhaouadm/

Bentley James Oakes
Assistant Professor
Polytechnique Montréal, Canada
https://bentleyjoakes.github.io/

Michalis Famelis
Associate Professor
Université de Montréal, Canada
https://michalis.famelis.info/
Linux Out-of-Memory Killer

Linux kernel:
• Development is through Git commits
• Culture for motivating/describing changes

Out-of-Memory Killer subsystem:
• When Linux runs out of memory, it calls OOM-Killer to avoid crashes
• Two broad steps:
  • Select “best” task to kill using heuristics
  • Force task to release memory and exit

What is the impact of changes?

- Dev works on "reclaiming used memory from the OOM victim".
- Find interesting commits from the Git history of OOM-Killer.

Manually uncovering why the system is the way it is can be hard!

- **Oom**: give the dying task a higher prirority
  - August 9th, 2010

- **Memcg**: give current access to memory reserves if it's trying to die
  - March 23rd, 2011

- **Oom - Kill**
  - remove_dying_task_prio()
  - April 14th, 2011

- **Mem**: introduce process_mrelease system call
  - September 2nd, 2021

- **Mm, Oom**: introduce oom reaper
  - March 25th, 2016

Suren's Challenge:

How does my decision impact previously established decisions?
How to make sure I will not cause conflicts with existing rationales?

Suren
Software Rationale

**Big** corpus of work on representing, structuring, extracting rationale

Useful to:
- understand the system
- learn from mistakes
- reuse solutions
- avoid conflicts

Little (Alkhadi’18, Sharma’21) about its characteristics in *real world systems*

No prior work on developer’s rationale in *code commit messages* of OSS

Alkadhi, Nonnenmacher, Guzman, and Bruegge. “How Do Developers Discuss Rationale?” SANER 2018
Sharma, Savarimuthu and Stanger. “Extracting Rationale for Open Source Software Development Decisions — A Study of Python Email Archives” ICSE 2021
Rationale in the OOM Killer Commit History

• Is rationale information present in commit messages?

• What are the factors that impact it?

• How does it evolve over time?

• How is it structured in commit messages?
Rationale in the OOM Killer Commits

1. Motivation
2. Dataset creation
3. Analysis
4. Conclusions
Rationale in the OOM Killer Commits

1. Motivation

2. Dataset creation

3. Analysis

4. Conclusions
Dataset Creation: Labelling

- Collect 418 commits
- Remove merge commits, filter code sentences
- 404 commits / 2234 sentences
- 3 annotators label sentences

Piloting
- 6 rounds → codebook + protocol
  - Resolve conflicts by discussion
- “Decision”, “Rationale”, “Supporting Facts”, “Inapplicable”

Batch annotations of the rest of the sentences
- Multiple classifications per sentence
- In disagreement, take classification union
- Fleiss kappa: Around 0.66 (fair to good agreement)
## Dataset

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit number</td>
<td>4</td>
</tr>
<tr>
<td>commit ID</td>
<td>C_kwDOACN7MtoAKGExOWNhZDA2OTE1OTdYjc5YzEyM2I4YTE5YTImYWJhNWFiN2Q5MGU</td>
</tr>
<tr>
<td>author name</td>
<td>Andrew Morton</td>
</tr>
<tr>
<td>committer name</td>
<td>akpm</td>
</tr>
<tr>
<td>message</td>
<td>mm/oom_kill.c: fix vm_oom_kill_table[] ifdeffery arm allnoconfig: mm/oom_kill.c:60:25: warning: ‘vm_oom_kill_table’ defined but not used [-Wunused-variable] 60</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://api.github.com/repos/torvalds/linux/git/commits/a19cad069f1597eb79c123b8a19a9faba5ab7d90e">https://api.github.com/repos/torvalds/linux/git/commits/a19cad069f1597eb79c123b8a19a9faba5ab7d90e</a></td>
</tr>
<tr>
<td>message_preprocessed</td>
<td>mm/oom_kill.c: fix vm_oom_kill_table[] ifdeffery</td>
</tr>
<tr>
<td>Decision</td>
<td>yes</td>
</tr>
<tr>
<td>Rationale</td>
<td>yes</td>
</tr>
<tr>
<td>Supporting Facts</td>
<td>no</td>
</tr>
</tbody>
</table>

Example dataset entry showing the structure

Available: [https://zenodo.org/records/10063089](https://zenodo.org/records/10063089)
Examples

**Sentence**

*mm, oom: introduce independent oom killer ratelimit state*

*printk_ratelimit() uses the global ratelimit state for all printks*

*The oom killer should not be subjected to this state just because another subsystem or driver may be flooding the kernel log*

*This patch introduces printk ratelimiting specifically for the oom killer.*

<table>
<thead>
<tr>
<th>Labelling</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td></td>
</tr>
<tr>
<td>Supporting Facts</td>
<td></td>
</tr>
<tr>
<td>Rationale</td>
<td></td>
</tr>
<tr>
<td>Decision</td>
<td></td>
</tr>
</tbody>
</table>

**Sentence**

*tlb: mmu_gather: Remove start/end arguments from tlb_gather_mmu()*

*The 'start' and 'end' arguments to tlb_gather_mmu() are no longer needed now that there is a separate function for 'fullmm' flushing*

*Remove the unused arguments and update all callers.*

<table>
<thead>
<tr>
<th>Labelling</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td></td>
</tr>
<tr>
<td>Rationale, Supporting Facts</td>
<td></td>
</tr>
<tr>
<td>Decision, Rationale</td>
<td></td>
</tr>
</tbody>
</table>
Dataset Description

Substantial overlap:

Decision word cloud.
Most frequent words:

Rationale word cloud.
Most frequent words:

Supporting Facts word cloud.
Most frequent words:
Rationale in the OOM Killer Commits

1. Motivation
2. Dataset creation
3. Analysis
4. Conclusions
Dataset Analyses and Research Questions

Presence of Rationale
• RQ1. How many commits contain rationale?
• RQ2. How much of the commit contains rationale?

Factors impacting Rationale
• RQ3. Does the quantity of rationale reported depend on the commit message size?
• RQ4. Does the quantity of rationale reported depend on the developer experience?

Evolution of rationale over time
• RQ5. How does rationale evolve over time?
• RQ6. How does rationale evolve over time for the five core contributors?

Structure of commit messages
• RQ7. In what order do the categories mostly appear?
Dataset Analyses: Presence of Rationale

Presence of Rationale

- RQ1. How many commits contain rationale?
- RQ2. How much of the commit contains rationale?

\[
\text{rationale density}\% = \frac{\text{number of commits that contain rationale}}{\text{total number of commits}}
\]

- 98.9\% of commits contain rationale

\[
\text{average rationale density} = \frac{\sum \text{rationale density}}{\text{number of commits that contain rationale}}
\]

- About 60\% of sentences per commit contain rationale
Dataset Analyses: Factors impacting Rationale

Factors impacting Rationale

- RQ3. Does the quantity of rationale reported depend on the commit message size?

- RQ4. Does the quantity of rationale reported depend on the developer experience?

Most the commits have fewer than 15 sentences

A lot of the short commits (fewer than 6 sentences) have a high rationale density (> 60%).

As a commit becomes longer, the tendency is between 40% to 60% of sentences to contain rationale information.
Only 5 developers wrote more than 16 commits.

All the other developers wrote fewer than 16 commits; most of them, fewer than 10 commits.

More experienced developers' commits have a consistent rationale density near 60%.

Factors impacting Rationale

- RQ3. Does the quantity of rationale reported depend on the commit message size?
- RQ4. Does the quantity of rationale reported depend on the developer experience?
Dataset Analyses: Evolution of rationale over time

- **RQ5.** How does rationale evolve over time?
- **RQ6.** How does rationale evolve over time for the five core contributors?

**Rationale density** consistently high at around 0.6.

**Decision density** consistently high (> 0.5).

**Supporting facts** density typically low (< 0.6).

In early and late years, decision density > rationale density >> supporting facts density.

In middle years, all converge at around 0.55, supporting facts density always at bottom.
Rationale density was consistent around 0.6 for all the years before 2020, but it dropped to around 0.4 in 2020 and 2021 and went up to 0.8 in 2022.

The number of commits varies considerably each year.

Usually, the top contributors write short commits (< 8 sentences)
Dataset Analyses: Structure of commit messages

Common Structure:

1. Decisions
2. Supporting Facts
3. Rationale
4. More decisions

Distribution of the categories over the normalized positions of the sentences of the commit messages

- RQ7. In what order do the categories mostly appear?
Rationale in the OOM Killer Commits

1. Motivation
2. Dataset creation
3. Analysis
4. Conclusions
## Rationale in the OOM Killer

| Presence | Commit messages **almost always** contain rationale information.  
On average, around **60% of the message** contains rationale information |
| --- | --- |
| Impacting factors | The **quantity** of rationale information reported does **not** depend on the commit message **size** or developer **experience**.  
**Experienced** developers have a rationale density around **60%**. |
| Evolution over time | Rationale density is **consistent** (~0.6).  
Decision density is always **high** (> 0.5).  
Supporting facts density is **lower** (< 0.6).  
More experienced developers write **short** commit messages (fewer than eight sentences). |
| Structure of commit messages | Developers tend to **start and end** their commit messages with **Decisions**.  
Rationale and Supporting Facts appear in the **middle** of the commit, with Supporting Facts usually preceding Rationale sentences |
Dataset and Analysis for the Commit Messages of the Linux Kernel Out-of-Memory Killer

An empirical contribution to better understand rationale in-the-wild.

Software Rationale

**Big** corpus of work on representing, structuring, extracting rationale

Useful to:
- understand the system
- learn from mistakes
- reuse solutions
- avoid conflicts

Little (Alkhadi’18, Sharma’21) about its characteristics in real world systems

No prior work on developer’s rationale in code commit messages of OSS

Rationale in the OOM Killer

**Presence**

Commit messages *almost always* contain rationale information.

On average, around 60% of the message contains rationale information

**Impacting factors**

The quantity of rationale information reported does not depend on the commit message size or developer experience.

Experienced developers have a rationale density around 60%.

**Evolution over time**

Rationale density is consistent (~0.6).

Decision density is always high (> 0.5).

Supporting facts density is lower (< 0.6).

More experienced developers write short commit messages (fewer than eight sentences).

**Structure of commit messages**

Developers tend to start and end their commit messages with Decisions.

Rationale and Supporting Facts appear in the middle of the commit, with Supporting Facts usually preceding Rationale sentences

Next Steps:
- Improve dataset quality and richness
- Compare with other Linux modules, other OSS projects
- Automate rationale classification

Dataset: [https://zenodo.org/records/10063089](https://zenodo.org/records/10063089)