Code Review

Short Guide

Why

- Mentoring.

- Communicate to future maintainer(s).

- Improve Quality, Mitigating Risks from bad code.

Principles

- Technical `Facts` and `data` overrule `opinions` and `personal preferences`.

- Stick to `Style Guide`. If it's not in, give priority to author.

- Escalate if needed. Don't leave it just because author and reviewer(s) are in disagreement.

- Don't make it personal.

What to looks at

- Design

- Functionality, Complexity

- Naming

- Tests

- Comment

- Style

Picking Reviewer(s)

- Whoever can help improve the quality.

- In case of modifying, give priority to previous author.

Code for Review

- A complete set of changes of code.

- In `PR`, `MR`, `CL` format

- Smaller is **faster** -> **better**

PR descriptions

- Title, First line
  - A complete sentence of what being done.
- Body
  - Informative part.
  - Make it clear with whatever means e.g.
    - include link(s) to related discussions
    - include diagrams
    - include screenshots

PR descriptions: Bad Examples

```
Fix Bug
# Fix what bug ?

Fix Test
# Fix what test ?

Complete Sprint
# Don't do this
```

PR descriptions: Good Examples

title: Clear and precise
Fix active expire timeout when db done the scanning

body
When db->expires_cursor==0, *it means the DB is done the scanning,*
we should exit the loop to avoid the useless scanning.

It is easy to see the active expire timeout in the modified test,
for example, let's assume that there is only 1 expired key in the
DB, and the size / buckets ratio is less than 1%, which means that
we will skip it in isExpiryDictValidForSamplingCb, and the return
value of expires_cursor is 0.

Because data.sampled == 0 is always true, so repeat is also
always true, we will keep scanning the DB, but every time it is
skipped by the previous judgment (expires_cursor = 0), *until the*
timelimit is finally exhausted.

ref: redis/redis#PR13030

Commenting in Code Review

- Be Respectful.

- Explain clearly.

- Give advice.

- Ask. If not sure about anything.

Key takeaways

- Code from Pair programming is considered as `reviewed`
  - but we should still do it anyway
- Code Review is a process in Quality Control, reducing risks from
  - Confused logic.
  - Bad performance from Bad design.
- Code Review is also a method for transferring knowledge.