The Emerging of the Virtual QA Team for Linux Kernel

Philip Li philip.li@intel.com
Agenda

- Status Quo of the Trend
- Involvement of 0-Day CI
- Challenges and Discussion
The community has established the decentralized and independent quality assurance.

The testing equips more capabilities and becomes modern along the way.

Eventually it results in a full functional QA team that works virtually across the community ...
The advancement in test methodology decouples the traditional need to map code change to test case in a few areas.

The adoption of engineering in each testing level.

New ideas keep coming.
More and more players join in this field to have their own focus and priority to contribute to the kernel.

The connection with developer has different models to ease the adoption.

Test services are moved to cloud (if not all).
• Bisection becomes a standard part to provide firsthand analysis.
• A central place for result storage and report customization.
• Volunteers form the roles such like “regression maintainer” to remind/drive the closure of regression.
The collaboration with developers is more supportive, with ease to reproduce tools.

The focus on release quality (criteria) to assist maintainers to decide whether it is all ready.
Now Link them All

- The full functional testing effort is established with modernized methodology (advancement but not necessary replacement).
- Conceptually, a virtual QA team merges. It does the testing by less solos of various testing effort, tighter connection with developer, and extra volunteered ownerships.
0-Day CI Involvement
Recent Focus

- Expand the coverage of shift left to reduce the direct reports against mainline.
- Improve bisection effectiveness.
- Introduce machine learning, such as log analysis and test selection.
- Involve to contribute more to community.
Challenges and Discussion

There’re a lot of discussions regarding improving the overall testing. And this is one more :)

Linux Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022
Large scale testing (usually by the robots) is involved late sometimes (for most sub-systems).

Is there purposely designed coverage for the change like new feature to be tested reasonably (need exposed by submitter).
The constraints of platforms

Test resource is not enough to achieve systematic testing

Test Plan/Coding – Test Coverage

- The constraints of platforms
- Test resource is not enough to achieve systematic testing
Test Plan/Coding – Test Suites

- Test compatibility.
- Test applicability, same version to run on mainline, stables.
Test Execution – Shift Left

- Shift left to reduce the finding on mainline.
Test Execution – Test Methodology

- For example ...
  - Bisection.
  - Test scope selection.
  - Test duplication, like randconfig.
Test Execution – Developer

- How could the specialized tests running by developer be integrated to existing test framework if feasible?
What is the feasible AI we can adopt? Any practical examples.
Bug Reports/Scrubs - Long Tail Bug Findings

- Long tail bug findings, the top 10 “reported by” covers ~50%. Can automation further help here?

<table>
<thead>
<tr>
<th>v5.19 - v6.0-rc4</th>
</tr>
</thead>
<tbody>
<tr>
<td>reported-by [992]</td>
</tr>
</tbody>
</table>

195 19.7% kernel test robot
72 7.3% syzbot
66 6.7% Abaci Robot
50 5.0% Stephen Rothwell
44 4.4% Hulk Robot
32 3.2% Dan Carpenter
16 1.6% Zeal Robot
10 1.0% Hacash Robot
10 1.0% Nathan Chancellor
10 1.0% Randy Dunlap
487 49.1% <others>
Bug Reports/Scrubs – Regression

- Is every valid regression going to be solved?
- How to prevent regression being missed?
Bug Reports/Scrubs – Data Mining

- High level picture regarding trend and comparison between releases.
- The status of remaining issues (not necessary regression).
- ...
Bug Fix/Release

- Assist developer to reproduce and root cause
- Reproduce and debug but may be lack of environment (HW).
- Leverage qemu or docker image if issue can be reproduced there.
- Assist maintainer to judge the quality in time analysis for the component
- Acceptable criteria from corresponding maintainer.
This is not to make a grand unified theory, but a step to look for more collaborations and hints for future testing

Transparency and trust
Enhanced traceability
Highest quality possible validation
Linux Plumbers Conference

Dublin, Ireland  September 12-14, 2022