

# Characterization of the Xen Project Code Review Process: an Experience Report



Daniel Izquierdo-Cortazar, Lars Kurth, Jesus M. Gonzalez-Barahona,  
Santiago Dueñas, and Nelson Sekitoleko

# Problem

## Traditional Code Inspections

## Modern Code Review

- Mandatory code review
- Project/ Code review parameters: patch reviewers, size of the patch, time-to-merge, neutrality etc.

### Impact

-Product & process quality, cost and time to market

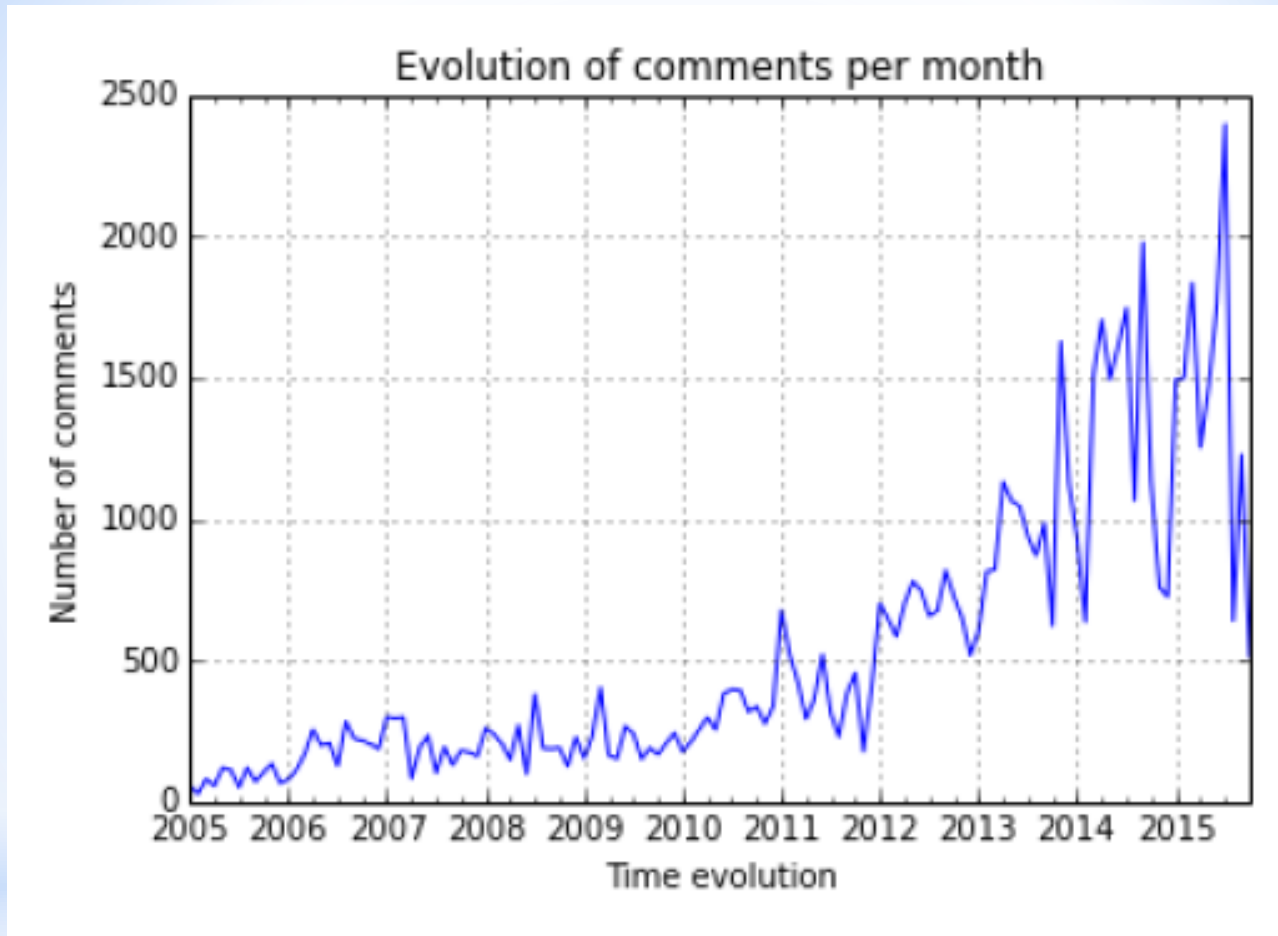
### Understand

Impact of code review the overall development effort:

Mandated reviewer  
checklists and in-person  
meetings

Lightweight, tool-supported

# Xen Project: virtualization platform powering [largest clouds](#) e.g [Amazon](#), [linux](#)



- Impact of [Code review process](#) on [development effort](#): mainly [Time-to-Merge](#)
- Verify: [perception](#): increased messages related to code review : Impact on project parameters: e.g. time to merge.
- [Quantitative data](#) suitable for [informed discussion](#): corrective action

# Research Questions

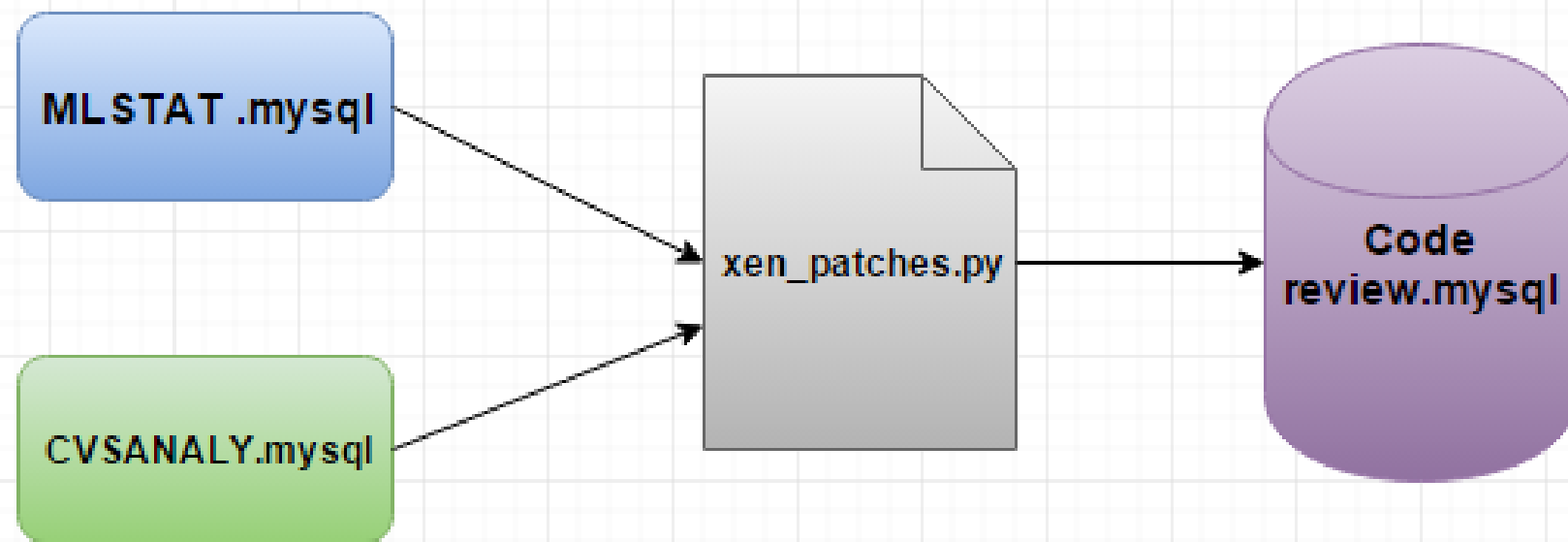
**RQ1:** Is **time-to-merge increasing**, measured from the moment a change is proposed to the moment it is finally merged into the code base?

**RQ2:** Is there an **impact of size** of “**combined patches**” (patchseries), measured as the number of individual patches in them, on **time-to-merge** for those patch series?

Year	No. of commits	No. of emails	No. of Patches
2002-2016	34,494	257,572	146,977

**Table 1:** Some parameters related to the activity observed in Xen repositories

# Methodology



- 1.Retrieval of mailing list information
- 2.Retrieval of Git information.
- 3.Detection and classification code review messages.
- 4.Merging information from CVSAAnaly (git) and ML-Stats (mailing list) databases
5. Analysis: Querying, Ipython notebook , Pandas --> Simple Statistics

# Methodology

Year	No. of commits	No. of emails	No. of Patches
2002-2016	34,494	257,572	146,977

Table 1: Some parameters related to the activity observed in Xen repositories

Year	No. of patches	No. of commits	No. of commits matching to patches
2011	1559	2181	581
2012	1907	2296	954
2013	2345	2503	1396
2014	2035	2332	1315
2015	2060	2204	1244

Table 3: Number of patches, commits, and commits corresponding to patches identified in the combined database (matched commits)



# Results : RQ1:Time-to-merge is under control.

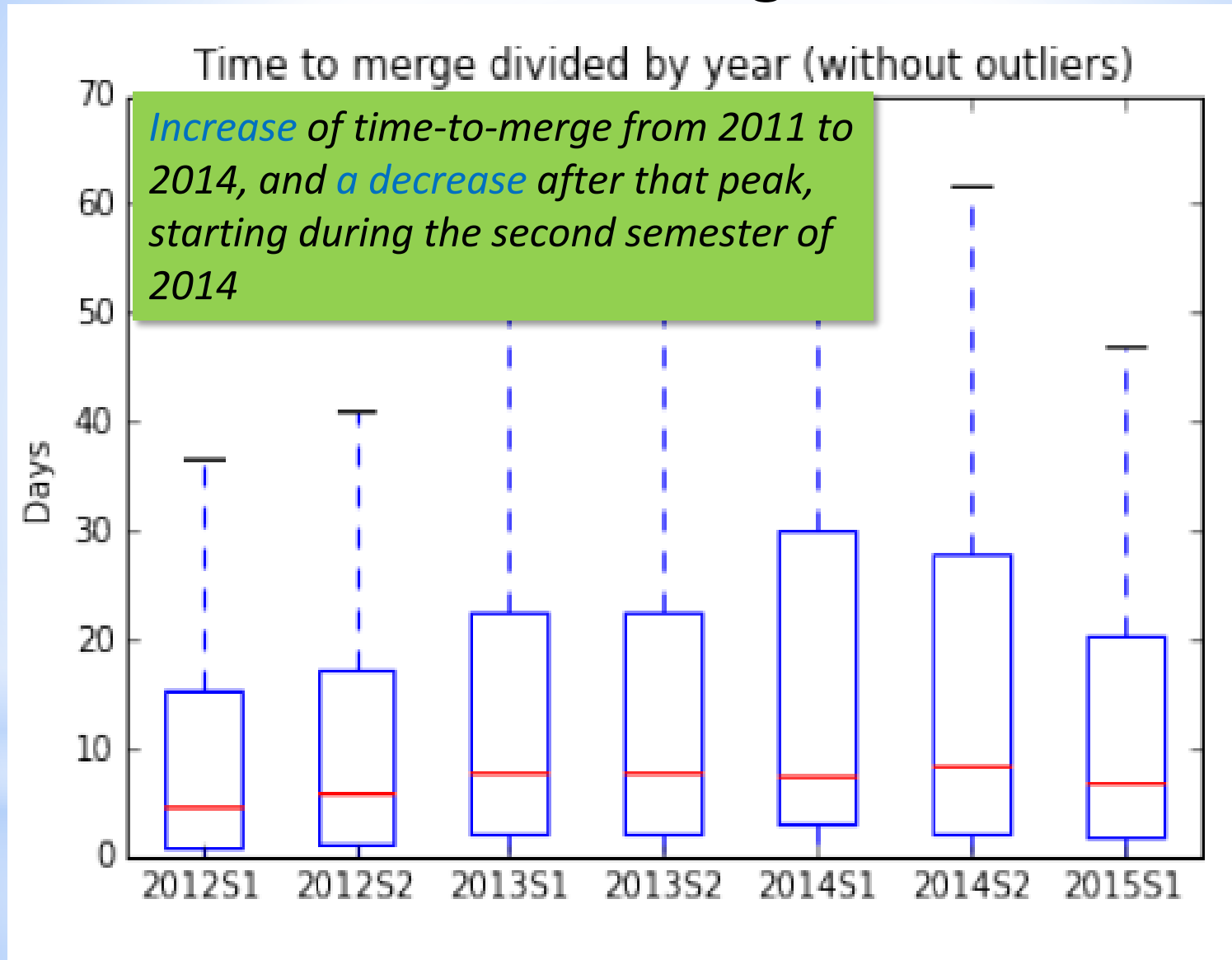


Figure 3: Evolution of time to merge

## RQ2: Time-to-merge is behaving in a similar way for all sizes of patch series

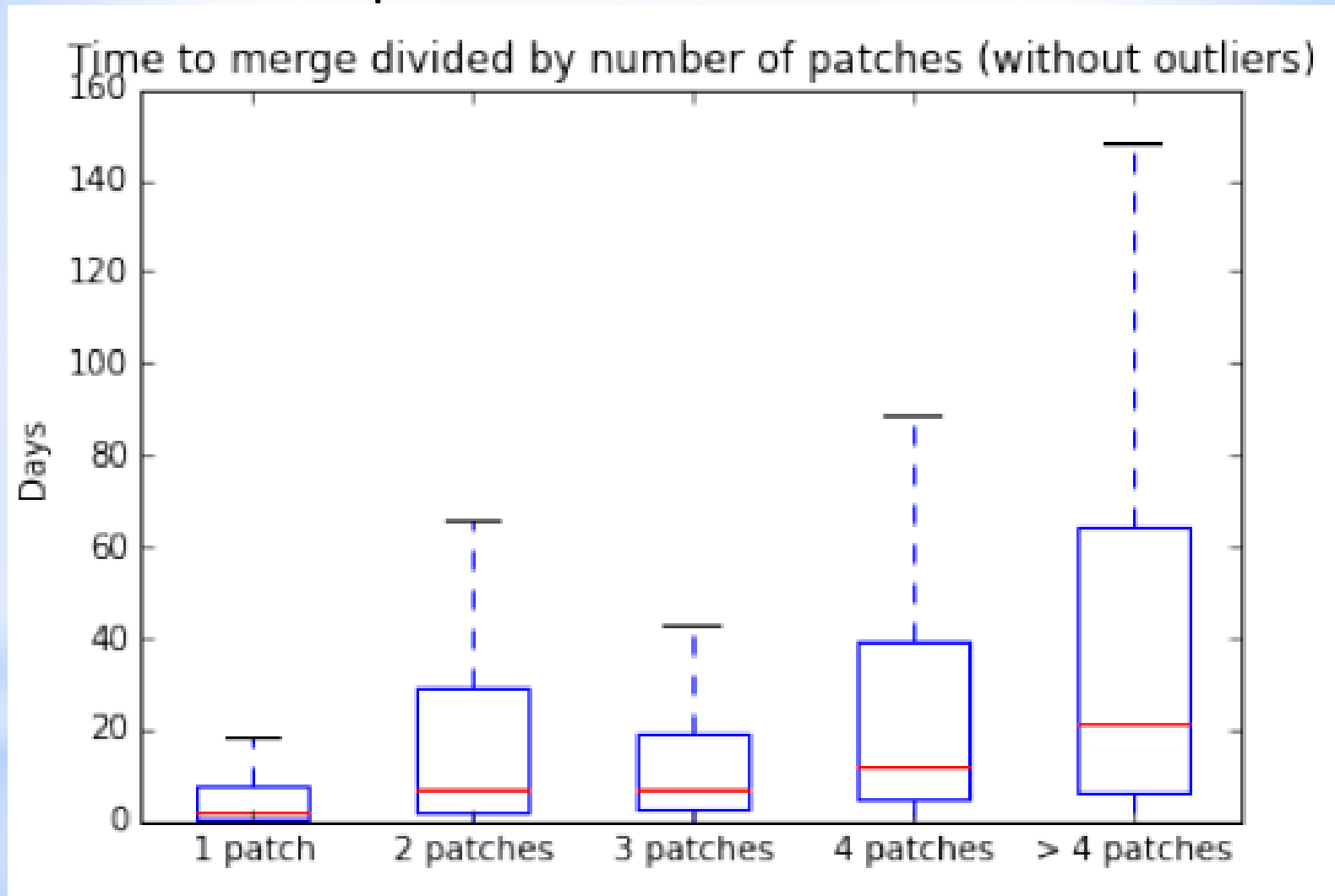


Figure 4: Time to merge patchsets (days) by number of patches



# RQ2:Time-to-merge is behaving in a similar way

Small and large patch series show the same pattern

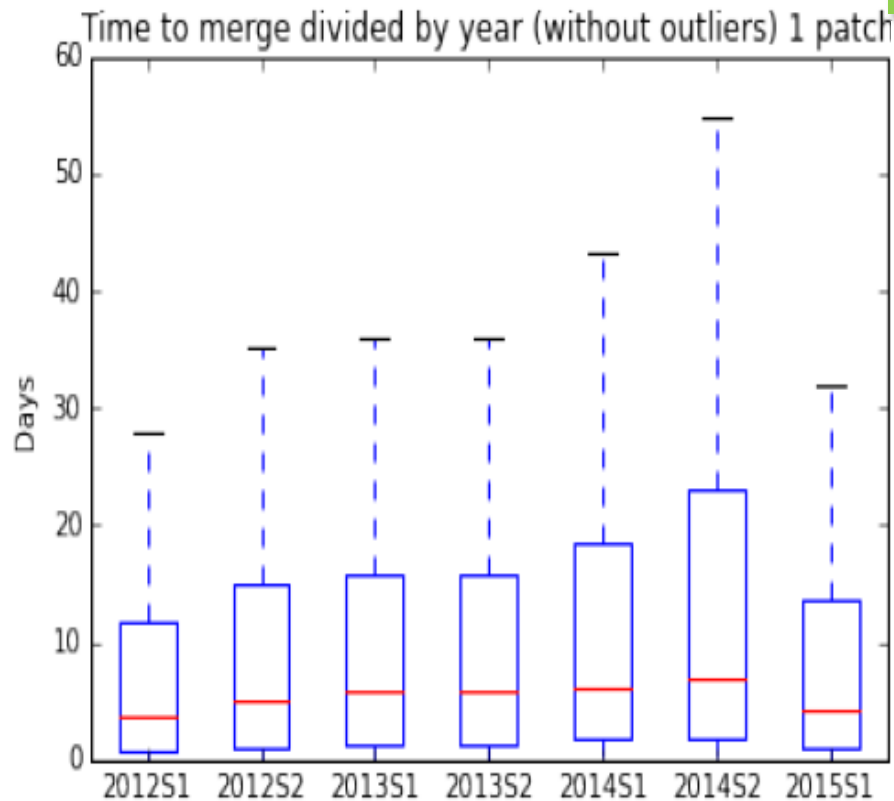


Figure 5: Time to merge (days) for patch series composed of a single patch..

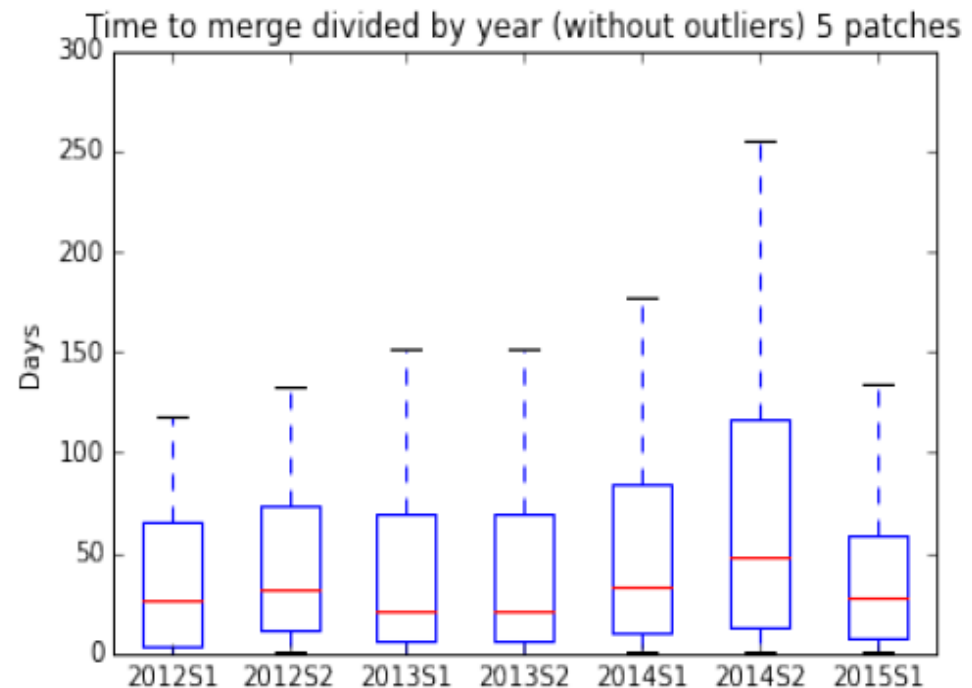


Figure 6: Time to merge (days) for patch series composed of more than 4 patches.

# Conclusion, Future work and Take away

## → Conclusion

- **Time-to-merge** is under control: Xen community
- **Small** and **large** patch series show the **same pattern**.
- **Other metrics**: time to commit, time to re-work a patch, cycle time , time to review

## → Future

- **Matching algorithm**: email threads to git commits
- Conducting similar study (and more): **Linux Kernel**