



Bugzilla Anthropology

Preliminary Results

By Martin Best



Agenda

- Intro
- Bug Life Cycle
- Rapid Release Impact
- Drilling Down
- Triage
- Going Forward



Research Approach

- Interviews
 - 1 hour long
 - Results in 6-8 pages of dense point form summaries
- Metrics that provide trending
- How Bugzilla is being used?
 - What is working and what is not?
 - Are there any useful tools or hacks?
 - How are bugs moving through their life cycle?



Status

- Have completed 10 hours of interviews, including community, transcripts posted
- Have discovered existing tools that allow for the monitoring of high level trends
- Working with metrics to surface more trending information based on interview results
- Very large amount of detailed information gathered, too much to provide a useful summary in 30 mins.



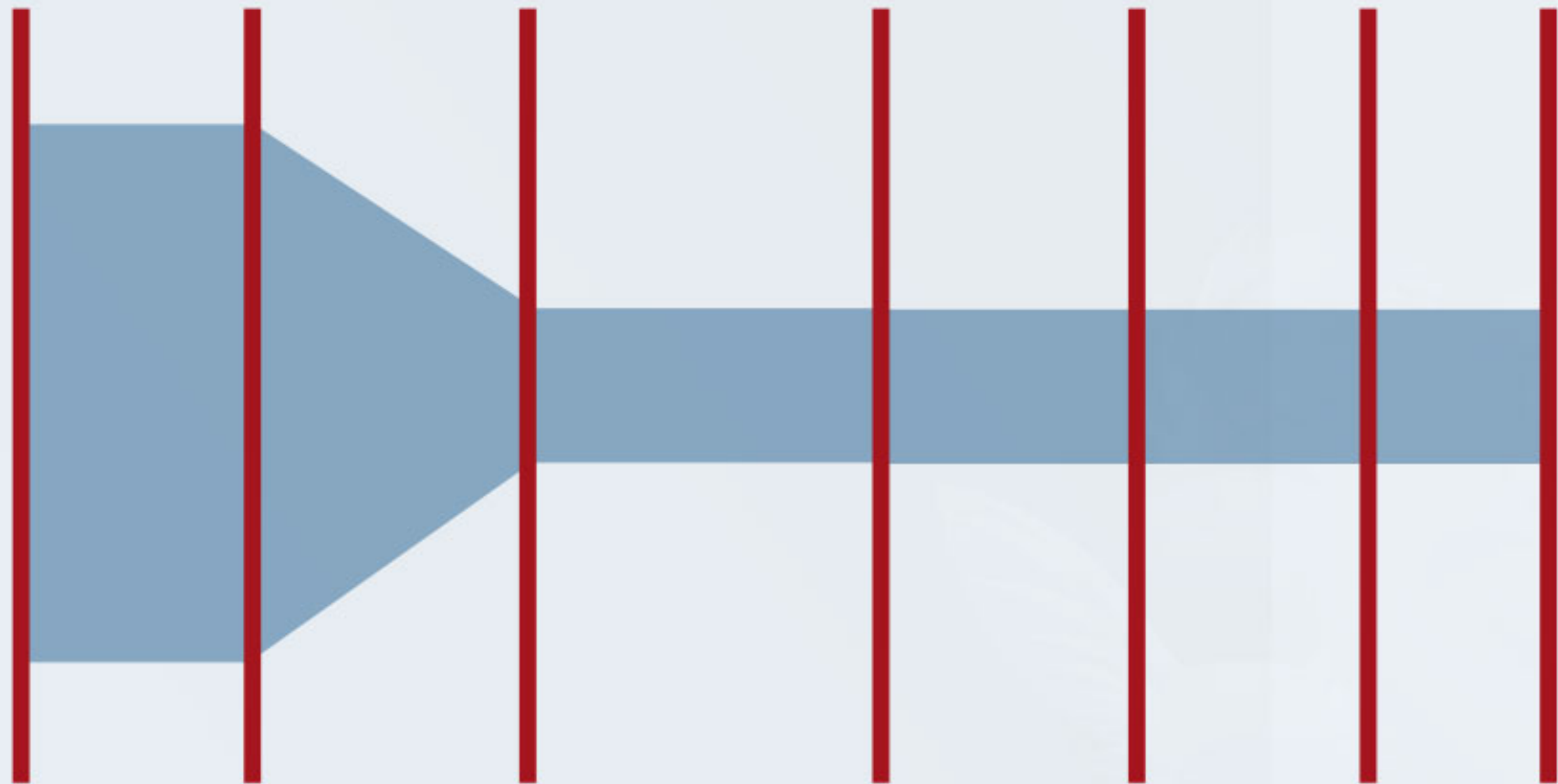
Bug Life Cycle



BUG LIFE CYCLE



SUBMIT TRIAGE CODING REVIEW VERIFY APPROVE SHIP





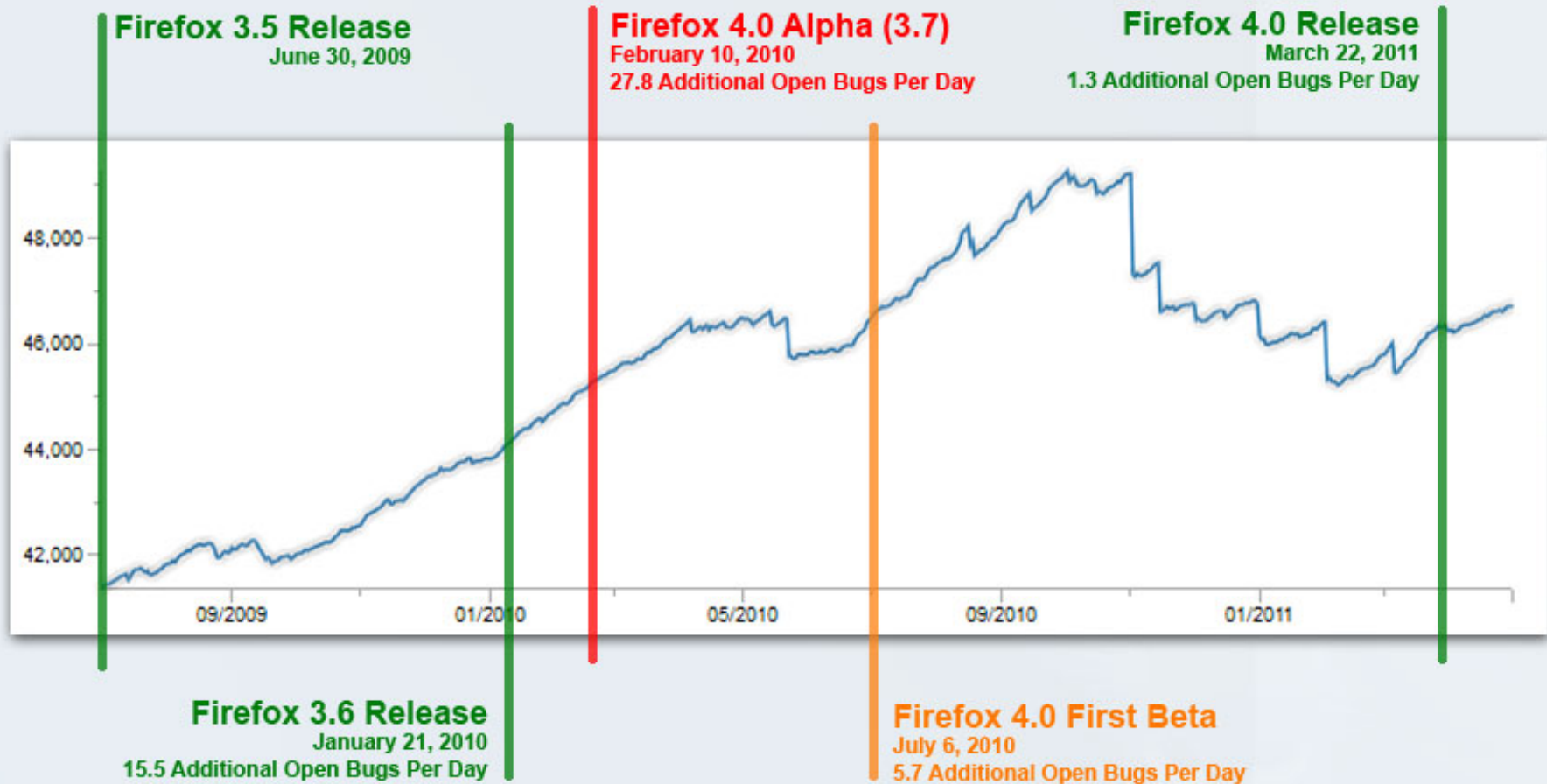
Rapid Release Impact



Open Bugs in Firefox and Core Over Time

From 3.5 to 4.0

3.5 to 4.0 - 7.8 Additional Open Bugs Per Day





Changes in Rapid Release

- Development, Aurora, and Beta all overlap
- 6 week development time frame for each release in total
- Increases the requirement to stay on top of bug volume to avoid build up of "must be fixed" bugs



STRATEGIC

QUALITY

INNOVATION

STRATEGIC



QUALITY



INNOVATION

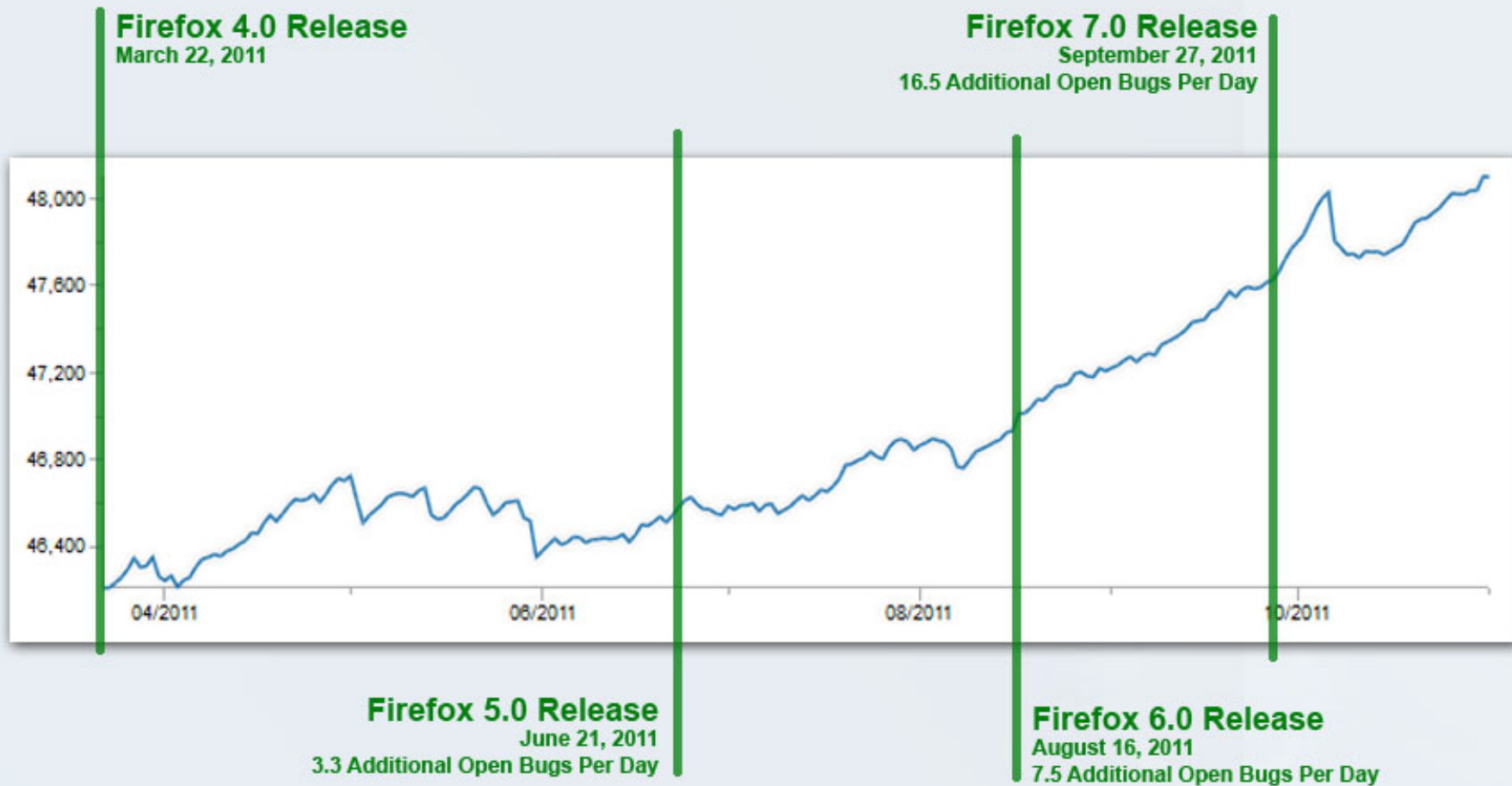


Open Bugs in Firefox and Core Over Time

From 4.0 to 7.0

4.0 to 7.0 - 7.5 Additional Open Bugs Per Day

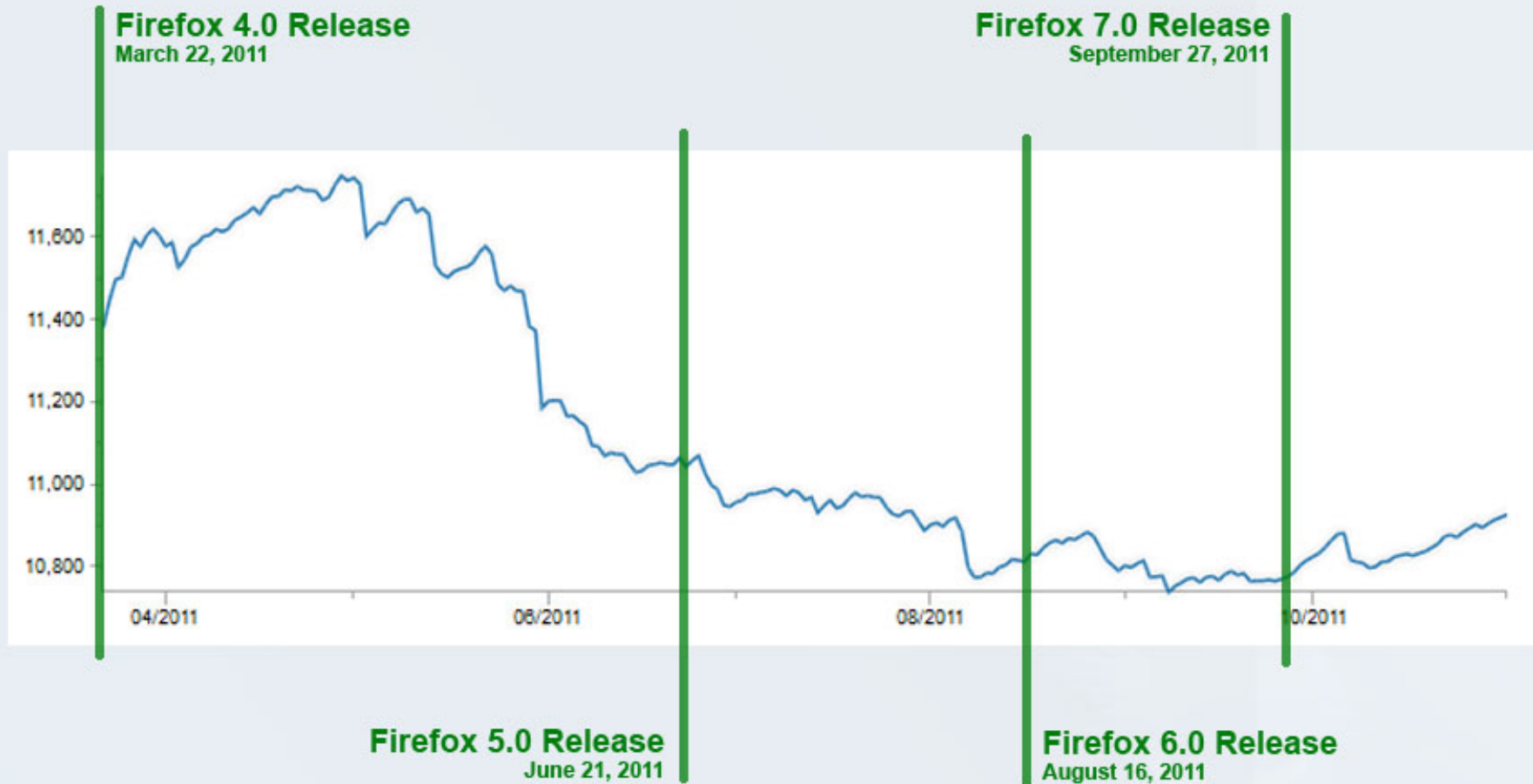
5.0 to 7.0 - 11.4 Additional Open Bugs Per Day





Unconfirmed Bugs in Firefox and Core Over Time

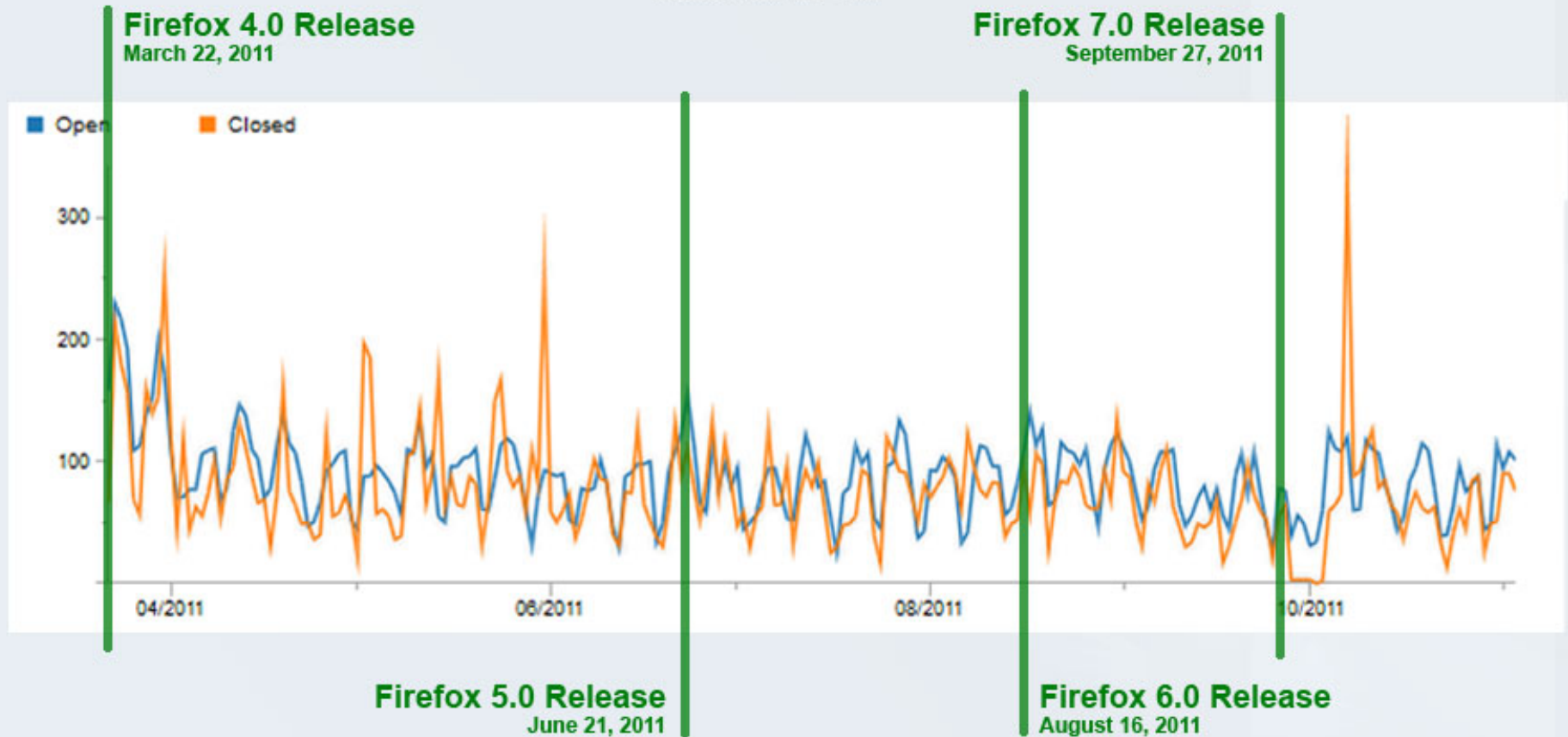
From 4.0 to 7.0





Daily Open and Close Status Change in Firefox and Core Over Time

From 4.0 to 7.0





Drilling Down



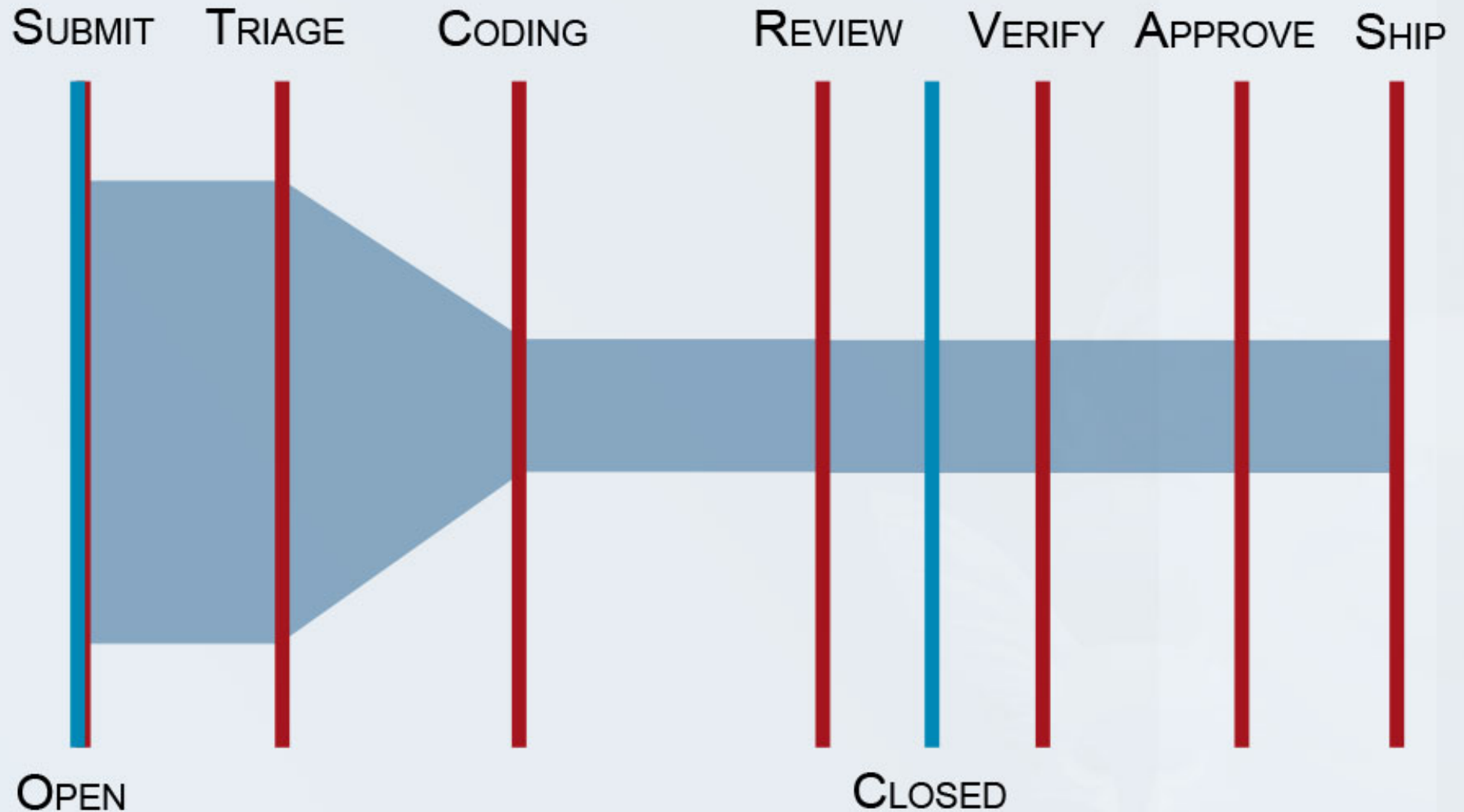


Why is the number of issues closed so flat?

- With 47 new staff added since January to Firefox and Core, it is surprising that this number is staying in the same range
- Is it a data error?
- Is it a seasonal trend?
- Is there friction due to new hires?
- Is there a bottle neck somewhere in the bug life cycle?



BUG LIFE CYCLE





Interview Data

- Search interview transcripts looking for trends that might indicate a bottle neck
- Focused in on the review queues
- Reviewed how different people are approaching the process
- Reviewed the differences in perspective between the reviewers and those submitting patches



Review Process Interview Summary

- Review process seems to cause a fair amount of concern in how it is currently being done
- Having to add a name manually focuses a lot of bugs on a few reviewers
- Wide range of response time makes it hard to predict outcome
- So far all reviewers interviewed feel that getting a review turned around in 24 to 48 hours is important



Results

- How responsive is your team when an internal bug is created (1 slow, 10 fast)?
- How responsive are other teams to bugs you submit (1 slow, 10 fast)?
- Review process turn around (1 slow, 10 fast)?
- Approval process turn around time (1 slow, 10 fast)?

Individual Results

Average

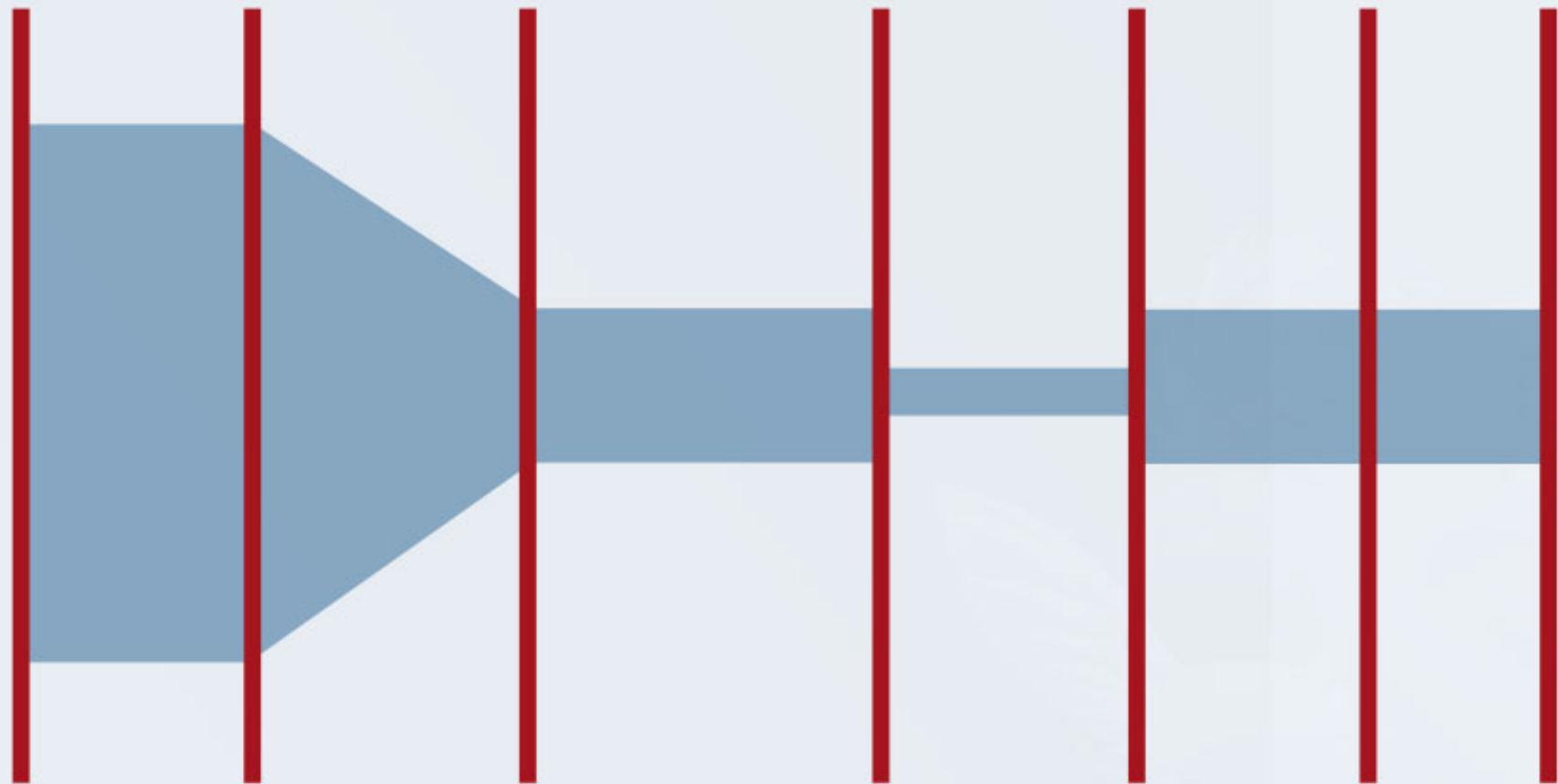
1	6-7	9	8	1-10	-	5	5	10	-	7.14
2	6-7	9	6	2-9	1-10	5	5-7	7	-	6.31
3	5	-	1-7	3-8	-	5	8	8	-	5.92
4	8	9	7-8	5-9	-	7	9	9	-	8.07



BUG LIFE CYCLE



SUBMIT TRIAGE CODING REVIEW VERIFY APPROVE SHIP





Next Steps

- Although there is much data to suggest there is an issue, no conclusive evidence has yet surfaced
- Metrics Team has been tracking data that is expected to give us a macro view of how the different review queues are changing over time
- Currently there is no interface for seeing this particular set of data
- Working with the Metric Team to get this information visualized



Triage

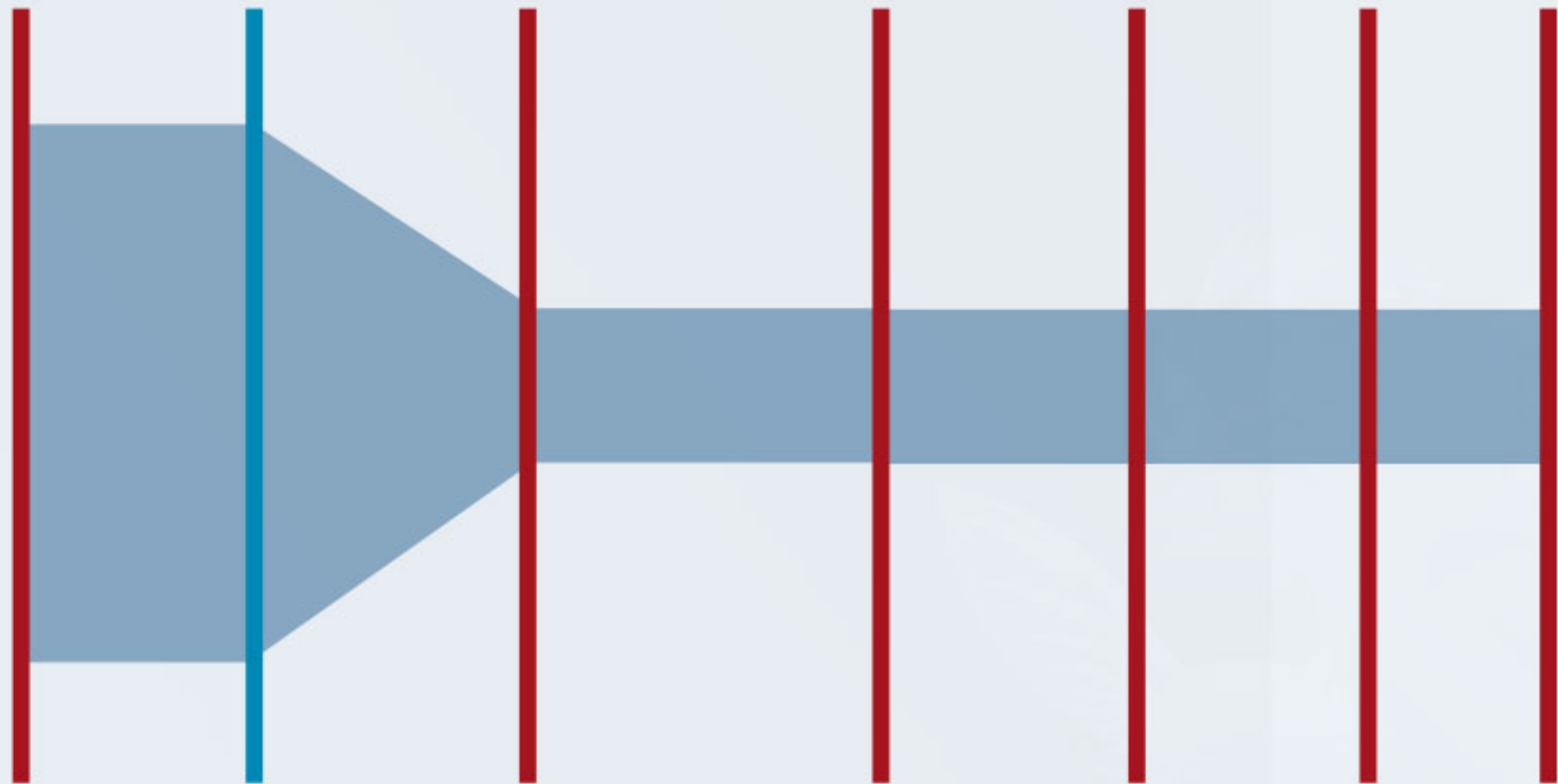




BUG LIFE CYCLE



SUBMIT TRIAGE CODING REVIEW VERIFY APPROVE SHIP





What Activities Are Included in Triage?

1. Separate out service requests
2. Clarification
3. Correct Component
4. Regression Window
5. Repro Steps
6. Severity (fix or not)
7. Assign or add to a component cue



Interview Statistics Relating to Triage

- On a scale of 1 to 10, how confident are you that your group knows about every bug that is important?
- (1 no confidence, 10 total awareness)

Individual Results

Average

1	5	1	7	3	1	7	7	1-2	3.72
---	---	---	---	---	---	---	---	-----	------



Comments From Those That Answered 1

- 100% that there are untracked bugs. Anytime you go through all the bugs, you find, "Wow, that's really important and everyone missed it." This was recently reinforced by the graphics.
- Graphics currently has about 2500 bugs. No one is aware of the full range of issues in that list due to volume. This is the list that is properly filed and does not included concerns about issues yet triage.



Graphics Bug Kill Day

- Started from oldest and started closing bugs
- 1/6 were considered actionable
- 1 in 40 had patches that could be landed
- Older bugs have a higher likelihood no longer being relevant



Triage in Firefox and Core

- Triage done primarily by Engineers
- Triage often done by watching new submit bug mail
- If initial wave of bug mail is missed by all developers of that component, there is a good chance that a bug will be lost unless a reporter calls attention to it

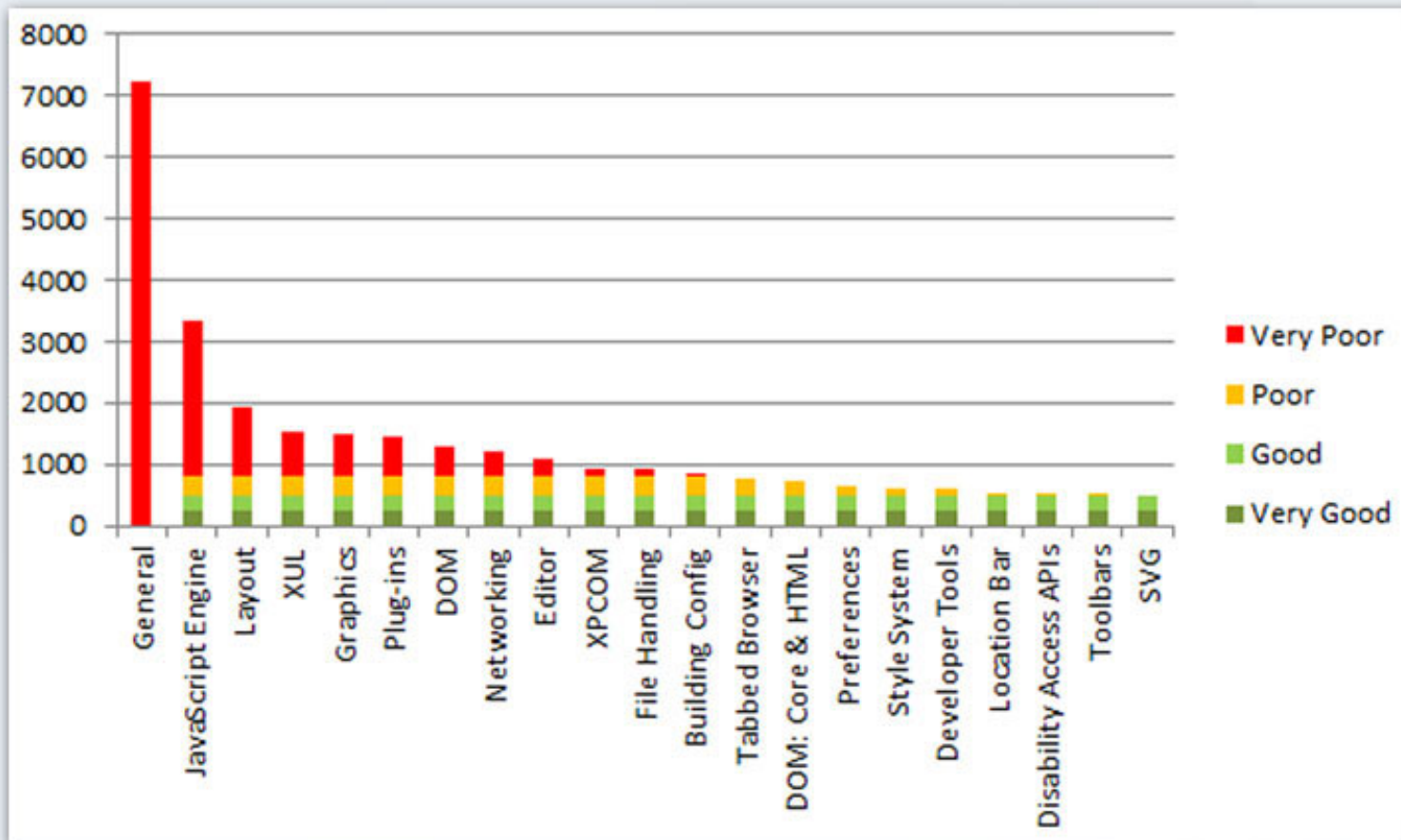


Challenges to Firefox and Core Triage

- Several managers have talked about the difficulty of excessively large queues making it hard to stay aware of important bugs
- When asked at what volume these issues becomes a problem, numbers from 250-800 have been suggested
- Intend to start drilling into this further with follow up questions

Number of Bugs per Component

How Well a Single Manager Can Maintain Awareness





Going Forward





Finishing Research

- We are currently targeting a total of 20 interviews and will re-evaluate if more are needed
- Develop web based reports that will allow for trend tracking on more points along the bug life cycle
- Targeting End of January to have research completed and move towards developing solutions



Questions We Need People Thinking About

- How will we move past research and start using what we have learned?
- Are there areas of the bug life cycle that are not being studied?
- Is our interpretation of the data valid?
- What else do we need to look into?



Slides Removed





How is Triage Being Done

- Three different approaches to triage
 - Triage in a component of Firefox and Core
 - Triage in General
 - Triage in Fennec



Triage in Fennec

- Responsibility divided between QA and Engineers
- QA does a first pass on the bugs looking for duplicates, setting tracking flags, checking that it is in the right component, make sure it is important
- Engineers take over and handle repro steps, regression window, and assignment
- QA will set tracking flags to let Engineers know a bug is ready, and Engineers will set an assistance flag to let QA know they need a hand



Proposed Triage Solutions Mentioned During Interviews

- Creation of a next step system in Bugzilla (suggested by Jesse Ruderman on his blog)
- Better monitoring of trends
- Bugkill days to keep bug counts reasonable
- Internal triage group to help assist and coordinate community
- Better dashboards to help improve bug visibility rather than relying on bug email



Triage in General

- Primarily done by community
- Community starts at the top of the list and tries to move bugs forward in the triage process
- Will sometimes create searches to reduce the number of bugs in a list
- Attempt to adhere to the rules of each component when it comes to use of meta data

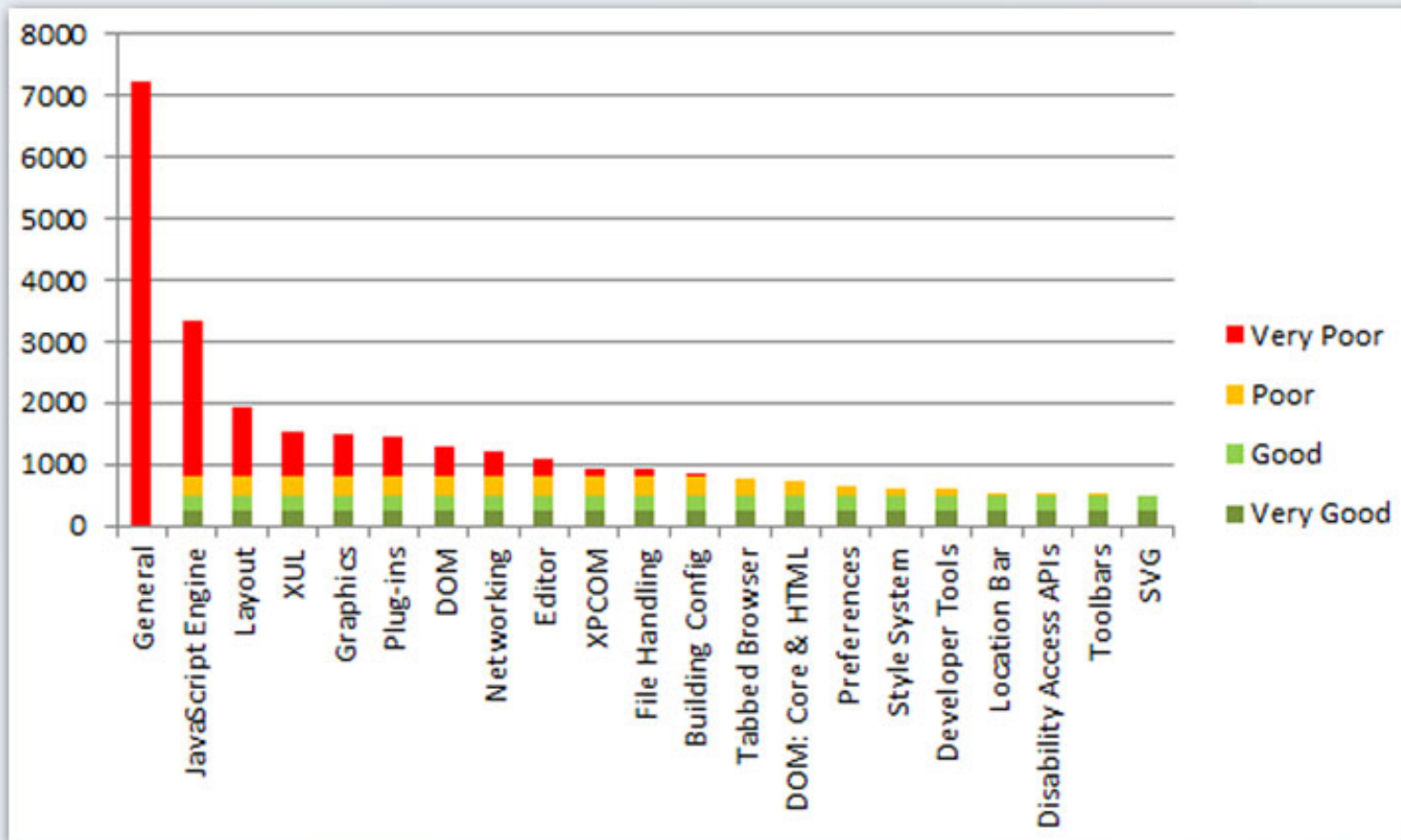


Challenges to Triage General

- No way to know what stage of the triage process a bug is in
- Hard to figure out what a bug needs without reading the comments
- When a bug is triaged it is often not picked up by a developer if left in General
- No clear internal owner of General
- When asked about general, developers assumed someone else is handling that

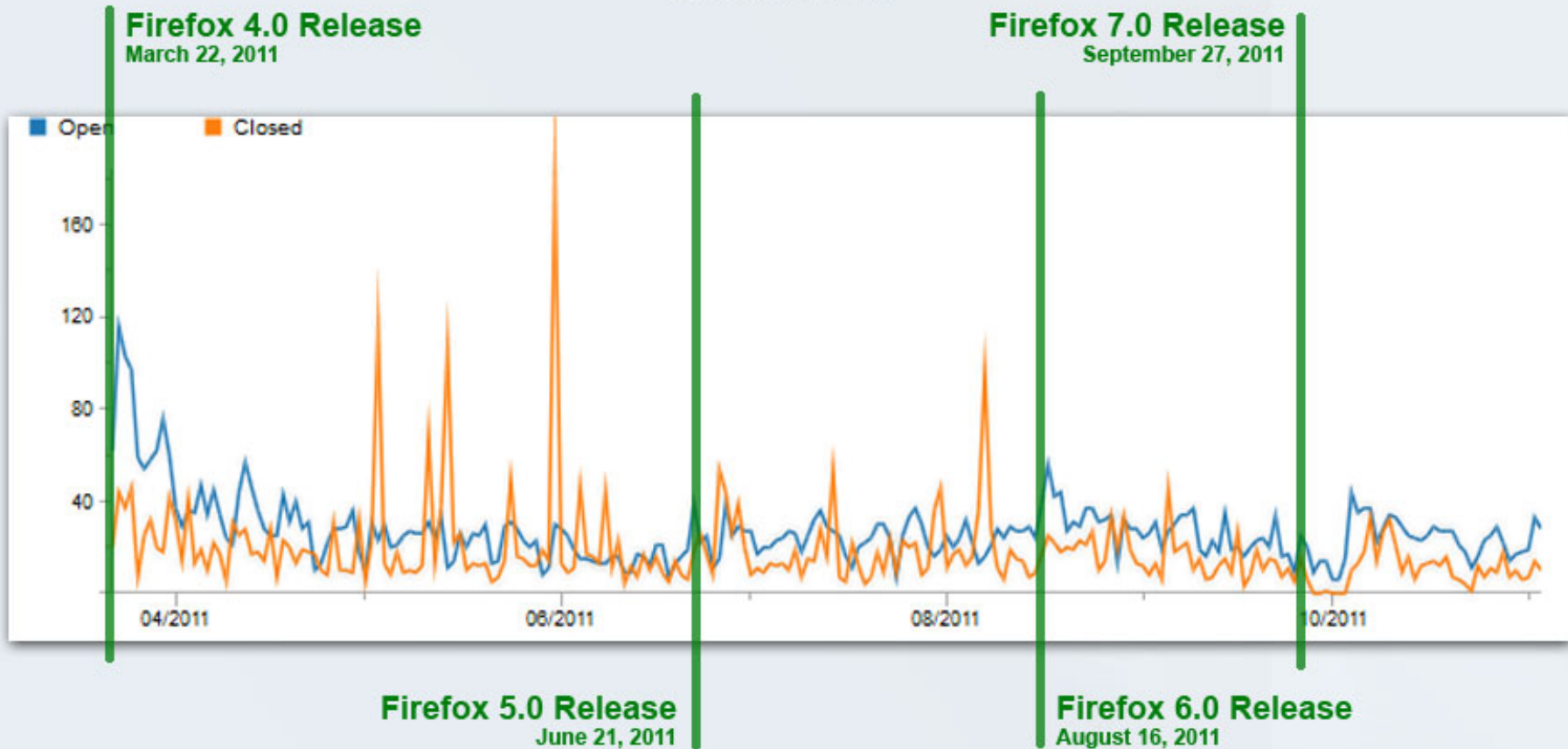
Number of Bugs per Component

How Well a Single Manager Can Maintain Awareness



Daily Open and Close Status Change in Firefox General Over Time

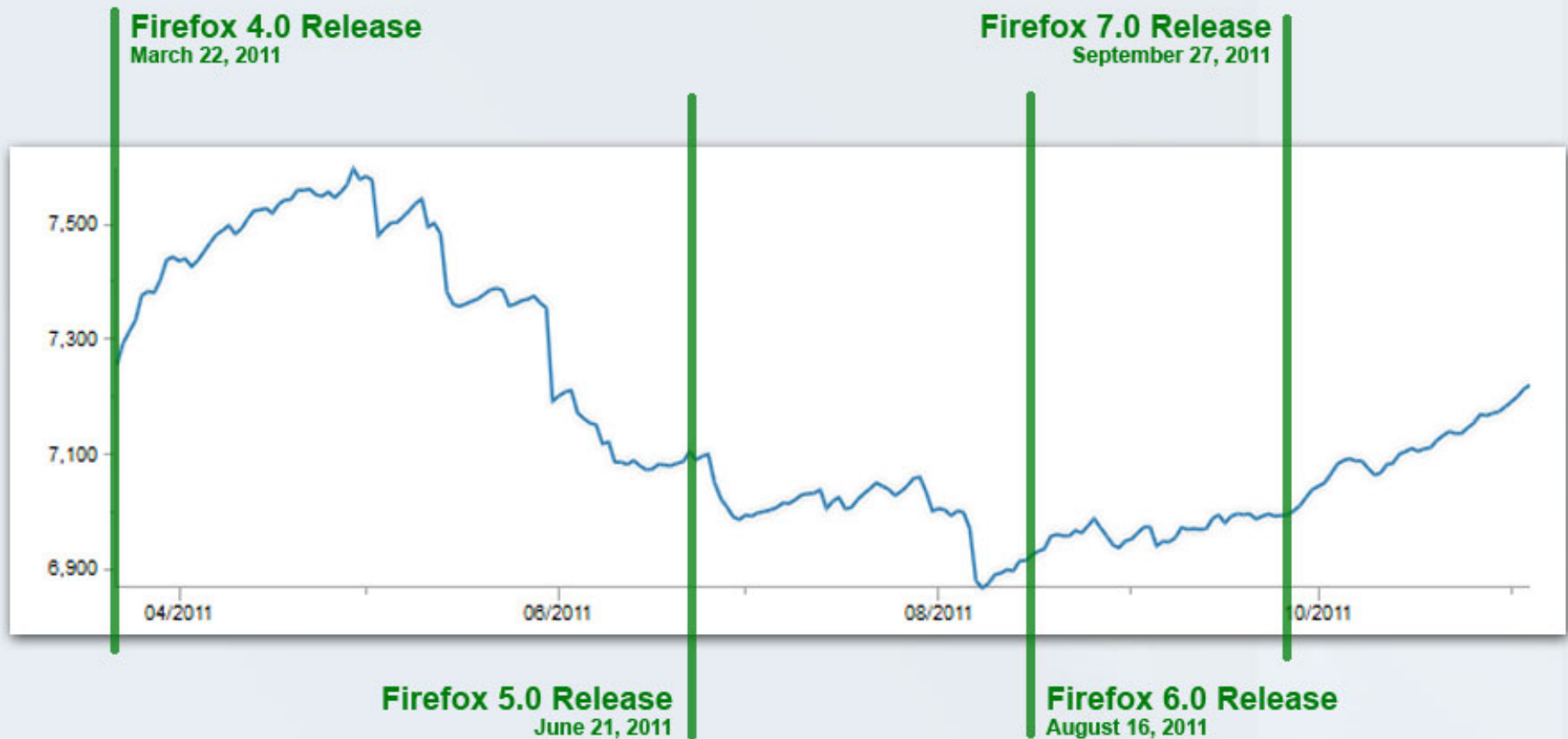
From 4.0 to 7.0





Open Bugs in Firefox General Over Time

From 4.0 to 7.0





Scope of This Presentation

- Data gathered via interviews is considerable and one could easily spend 30 minutes on one of the many areas being studied
- Will attempt to give some insight into several key areas we are drilling into currently
- Data will be provided in more detail in the coming days and weeks



BUG LIFE CYCLE



SUBMIT

TRIAGE

CODING

REVIEW

VERIFY APPROVE

SHIP





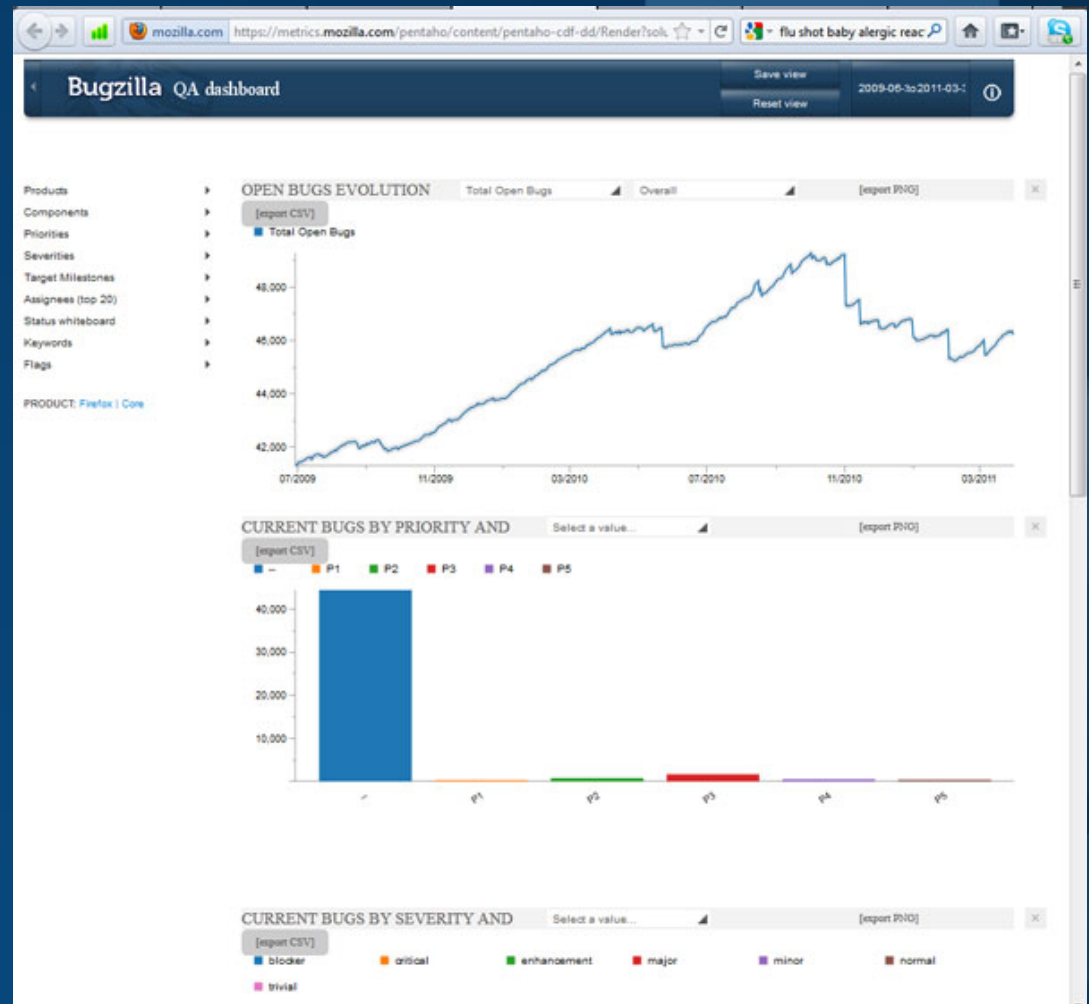
Monitoring

- The key points on the bug life cycle represent a point where a new person will be introduced to the bug
- These points should be monitored for backlogs
- Many stages are traceable and metrics has the data



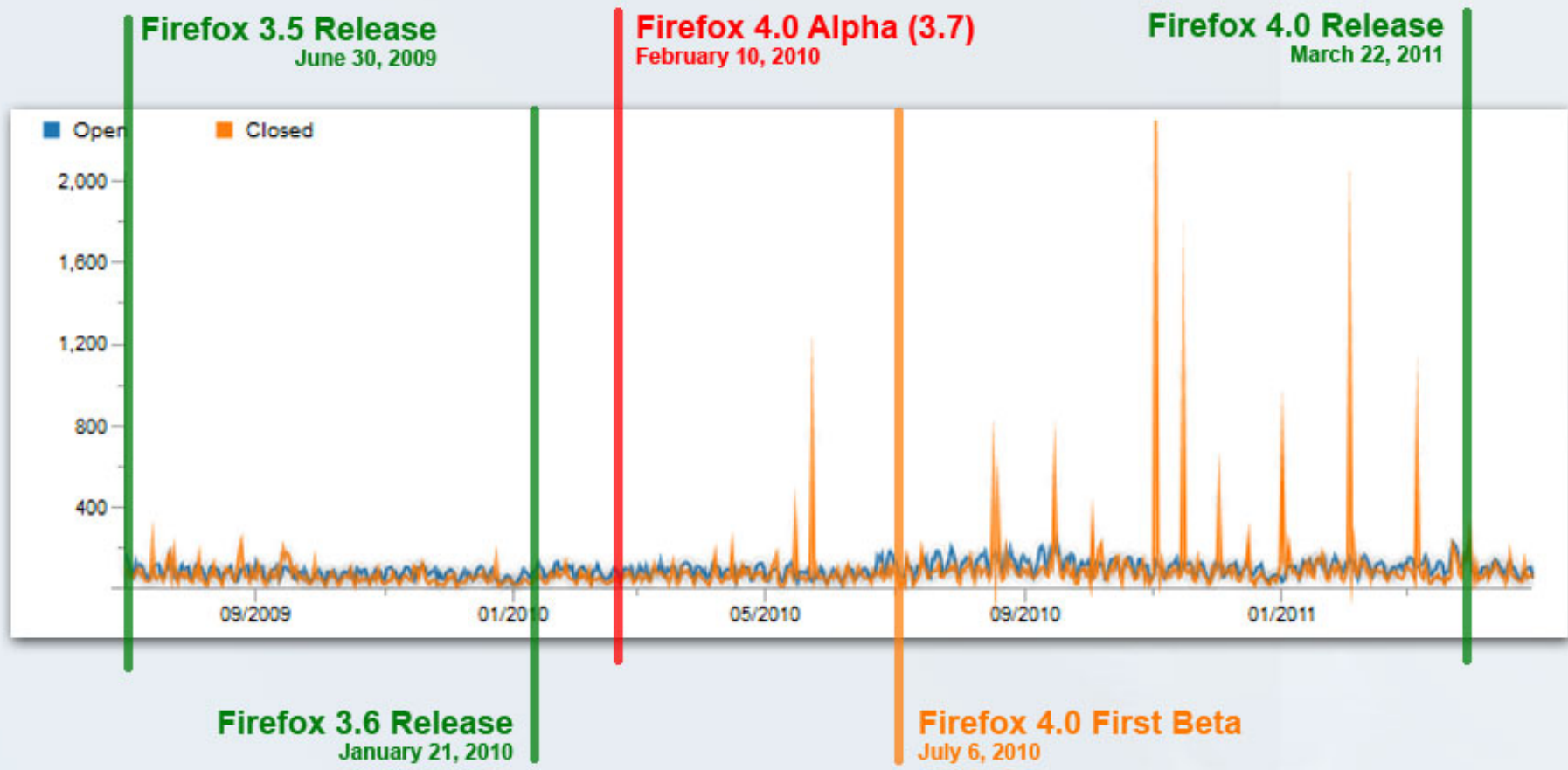
Macro Statistics Tool Find

- One of the first discoveries was the very useful QA Dashboard
- Much of the macro data presented has come from this tool
- Not 100% functional due to lack of use





Daily Open and Close Status Change in Firefox and Core Over Time From 3.5 to 4.0





Interview Statistics

- Several numerical range questions were asked during interviews to help find potential bottlenecks
- These included
 - Speed of response to submitted bugs
 - Speed of review and approval process
 - How confident they were that they were aware of everything they should be



Questions Relating to Speed

1. How responsive is your team when an internal bug is created (1 slow, 10 fast)?
2. How responsive are other teams to bugs you submit (1 slow, 10 fast)?
3. Review process turn around (1 slow, 10 fast)?
4. Approval process turn around time (1 slow, 10 fast)?