R&D Processes & Best Practices

Confidential, internal use only – v1.1, 28/10/2011

Revision history

Version	Date	Author	Changes
1.0	02/08/11	J. SIMON	Initial version
1.1	28/10/11	J.SIMON	Added “Production Issues of the day” in section 2.2 Added “Escalation of catastrophic bugs” in section 3.2 Updated “Release Management” in section 4

Table of contents

1. Build rules

Owner: Romain

Before you commit code, you have to compile not only your code, but the entire repository.

Before starting the compilation of the entire repository, do a GIT update, to include new code that has been committed since your last GIT update.

If you break the build, you have to send an email to *devteam* to say:

a) that you are the one who broke the build
b) explain where the build break comes from (did you follow all the rules, what didn't work?).

This will allow us to have stats on why the build is broken (and yes also on who breaks the build, which might come handy when we discuss salary raises).

If there is a build break, nobody is allowed to commit anything until the build break is resolved.

2. Bug tracking

Owner: Romain

2.1. Scope

The bug tracker (http://alert.criteo.prod) is used to track and resolve:

- Bugs [use type = "Bug Report"]
- Issues (that might not be bugs, like investigating something tricky) [use type = "Bug Report"]
- Some types of requests to the production team (please use the proper bug category under "Production") [use type = "Request"]
    - New or improved hardware
    - PRM changes
    - Deploying SQL patches
- Feature requests [use type = "Feature Request"]

2.2. Communicating on open bugs

Please open a bug rather than sending emails to *devteam* about issues to be fixed (especially emails with a big stack trace).

All information about the bug should be saved in the bug tracker, so that everyone can access it, including while the issue is investigated or later on, after the bug was closed and nobody remembers anything about it. In particular, it is crucial to explain what has been done to fix the issue.

We often see bugs being closed as "fixed" with no other information. This is not enough. We would like to have a description of how the bug was fixed and whenever applicable, GIT information corresponding to the bug fix.

You all receive a daily e-mail with bugs assigned to YOU (aka "Bugs assigned to you", duh). Please read it and take action.

In addition, the Production team sends another daily e-mail listing bugs detected by our monitoring systems (aka "Production issues of the day"). Again, please read it and take action: these issues are usually very important and can cause serious technical & business issues.

2.3. Filling a bug report

2.3.1. Who should a bug be assigned to?

When you open a bug, if you don't pick an owner, it will be chosen automatically depending on the bug category. Most of the time, this is the right thing to do.

Generally speaking, a bug should be assigned to the person who has the next action item. The bug could be assigned back and forth between people depending on who has the next move.

Please do not assign a bug to multiple owners (we actually removed this functionality in flyspray) as very often multiple owners = no owner at all.

If not sure who to assign a bug to, please pick someone in the "R&D Managers" group.

2.3.2. What if a bug is assigned to me but I'm the not the right person to fix it?

It is your responsibility in this case to reassign the bug to someone else.

2.3.3. What language should I use in the bug tracker?

English only please.

2.3.4. What bug category should I use?

Choose the most appropriate leaf category. If no category is appropriate, go up in the tree and use more general categories. Only as a last resort use the "I'm not sure" category. We changed the category tree a bit so that they better match to current teams.

2.3.5. Which severity level should I use?

- Critical = very severe business impact, we are losing or might lose serious money , whoever the bug is assigned to must stop everything else and work on the bug as top priority.
- High = important business impact, potential for losing money if the bug is not fixed rather quickly, however no clear immediate impact. We want to solve this kind of bugs in a matter of a few days.
- Standard = No short term impact, but something to fix for the long run.

Note that feature requests and requests should be at "Standard", unless they are linked to higher severity level bug fix.

2.3.6. Filling the due date field

We are using the "due date" field as a way to tell when the issue will be fixed. It is important to give visibility on this so that whoever is impacted by the bug can plan accordingly or discuss the due date of it is not appropriate.

If a bug is assigned to you (and provided you are the right person, otherwise please reassign right away), we expect you to fill the due date:

- ASAP for "critical" bugs
- At most 24h (not counting week ends and days off of course) after it was opened for "high" severity bugs.

- At most 72h after it was opened for “standard” severity bugs.

There should also be a note acknowledging the issue and a first analysis of what is going to be done to fix it.

After filling the due date field for the first time, please make sure you do resolve the issue by the due date or change the due date accordingly.

Generally speaking, reply promptly to bug tracker requests.

Note that we don’t fill the due date for PRM or SQL patches request as they are always expected to be handled the next business day.

Note that feature requests (especially if they require non trivial amount of work) can be closed with reason “Added to backlog”, and are then handled like any other feature in the appropriate backlog.

2.3.7. Additional information on SQL Patches and PRM changes

You should use this information:

- Project: “**Product’**
- Task type: “**Feature request**”
- Alert type: “**SQL patch**” or “**PRM**”

A few common sense guidelines to help the Production team handle them more efficiently:

- The sooner you create the tickets, the better. Especially if you need them for the next release.
- Please do not assign tickets: tickets are auto-assigned to Julien, who will dispatch them to the best person, depending on priority, availability, etc
- All communication about a ticket should be done within the ticket itself, not through e-mails which are impossible to follow.
- You may use the “Severity” field and the text field to give an indication on how urgent the request is. Feel free to indicate a requested date as well.

3. Escalation

Owner: Romain

3.1. Escalation process

It is meant to handle cases where the tech team must be involved to solve issues (bugs or complex cases) with our platform, our product.

The escalation manager drives the issues to resolution with the proper level of priority (depending on business impact) and involving the proper resources. He also analyses the issues to tackle root causes and suggest long term improvements.

Business teams should NOT bring their issues to developers or the escalation manager directly. So that the whole thing remains scalable, I&S teams must be involved first. If I&S can't handle an issue, they will escalate it to the escalation manager.

Issues must be tracked by opening bugs in our bug tracker (will be done by I&S normally). Additional information regarding the issue, including estimated time/date of resolution, will be added to the bug. Of course, for urgent issues which require immediate attention, feel free to also send an email to the "escalation" alias.

If there are discussions through email anyway, always copy the "escalation" alias.

Finally, the escalation process is NOT meant to solve internal IT issues (your accounts, printers, blackberries, email not working, etc...). A different team is in charge of that, and the proper thing to do is to open a ticket for your issues by simply sending an email to the "helpdesk" alias. If you're not getting the appropriate level of responsiveness, please send Romain an email.

3.2. Escalation process for catastrophic issues

3.2.1. Defining catastrophic issues

- Anything impacting the look or performance of advertiser / publisher websites:
- Buggy passback code, buggy banner code that destroys the publisher web site
- High latency on javascript loader on advertisers web sites
- ·Anything that breaks:
    - Trackers on advertisers (especially javascript loader)
    - Invocation code of our ad server
    - Our banners HTML code
    - Passback on Criteo Publisher Marketplace
    - C-POP + API
    - Criteo Publisher Marketplace interface + API
- Anything that has a <u>major</u> impact on Criteo's gross margin (CTR drop, tagging issue, …)
- Any possible security / data leak issue
- A problem you would create a ticket for IS NOT a catastrophic bug ("my campaign can't start", etc)

3.2.2. Before getting in touch

If possible, please try to get I&S to quickly confirm the issue. This will eliminate 90% of false positives.

When you're convinced the issue is real, please prepare an e-mail with the following information

- WHO: your name, position, e-mail address & phone where you can be reached right away
- WHAT: concise description of the problem (5 lines max)
- WHEN: how long ago the problem started / was detected
- WHERE: impacted countries / advertisers / publishers, screen shots, 'traceroute' / 'nslookup' / 'Firebug' output, URLs where the problem is visible, customer-side monitoring information, etc. Anything that can speed up the investigation and resolution is welcome.

3.2.3. Getting in touch

Send the e-mail you prepared to astreinte@criteo.com ('astreinte' means 'on-duty' in French).

Make sure that you cc: comprod@criteo.com, escalation@criteo.com, j.simon@criteo.com

Use a very explicit subject, such as "PRODUCTION EMERGENCY"

For extra safety, ask for delivery & read confirmation e-mails

At this point, you probably woke up a few people. Please give them a few minutes to emerge and understand what happened.

3.2.4. Waiting

The on-duty engineer will get you back within 20 minutes by phone or e-mail to acknowledge the incident report.

In the meantime, please STAY CALM. Don't flood other people with e-mails or phone calls. It will only create confusion.

Instead, please try to gather additional info that could help to diagnose & resolve the issue.

3.2.5. Investigating

The on-duty engineer will start to work on the incident and provide feedback within 20-30 minutes. If this is a false alarm or if the severity is not critical, he will downgrade the incident to bug status, send a short summary by e-mail… and go back to bed.

If this incident is indeed critical, he will keep working and escalate the issue to the R&D management team. Appropriate and periodic communication will then be sent during the resolution of the incident

3.2.6. Escalating

If you don't hear from the on-duty engineer within 20 minutes of your first e-mail, or if there's a disagreement on the severity of the issue, or if you think the issue isn't handled correctly, you can escalate the issue directly to me

- Cell phone: REDACTED
- Personal e-mail: REDACTED

If you don't hear from me within 20 minutes, you can escalate the issue to Romain

- Cell phone: REDACTED
  Personal e-mail: REDACTED

If you don't hear from Romain within 20 minutes, then unleash Hell. Use all means necessary to raise the alarm.

3.3. Internal communication on technical issues

Non-R&D teams need more information on technical issues as they appear, so we developed a report that can be found here:

http://reporting.criteo/reports/Pages/Report.aspx?ItemPath=%2fOperations%2fBugs+impacting+business&SelectedSubTabId=GenericPropertiesTab&SelectedTabId=ViewTab

The URL for the report when not in the office internal network is:

https://reporting.criteo.com:442/reports/Pages/Report.aspx?ItemPath=%2fOperations%2fBugs+impacting+business&SelectedSubTabId=GenericPropertiesTab&SelectedTabId=ViewTab

This report is based on our bug tracker and lists bugs that have a wide impact and that you might want to know about.

Which bugs are selected for this report? In the bug tracker, there is a "Tracking" field on each bug. By default, it's set to "None". If you set it to "Business Impact", the bug will show up in the report. We'll rely on each of you to use your judgment and set the field appropriately when you open a bug or edit it. The right question to ask yourself when you decide on flagging the bug or not is "Does everybody need to know about this issue?".

3.4. Global performance issues

3.4.1. Scope

Every 2 or 3 months, we have to investigate a situation that looks strange: numbers (impressions, clicks, gross, etc…) are down in a country or globally and we don't know if there is a technical bug of if this is just the result of seasonality or some big accounts (publishers or advertisers) being down for some external or commercial reasons. It could also be a combination of the two. The goal of this process is to come to a conclusion when such a situation arises.

3.4.2. Process

Starting point: it should be handled like any other technical escalation: someone looks at the numbers and thinks there is a global issue. This person should contact the local I&S team that might already have some explanation.

If I&S don't know what is going on, they escalate to the escalation manager by opening a bug in the "product" project and assigning it to the escalation manager. If emails are exchanged, they should always copy the "escalation" alias (and the SWAT team, see below). This master bug should then be updated with any information/finding along the way so that everyone knows the current status and the details of what has been done to investigate.

The escalation manager does the first round of investigation with the tech team: are there known issues that explain the situation?

If nothing comes up, the escalation manager involves the BI team. The master bug is assigned to the BI team leader who can reassign to someone in his team. At this point, we have a BI owner. The escalation manager should make sure that the BI owner has been identified and has accepted the case.

The responsibility of the BI team is to provide quantitative explanations to the global situation. The idea is to single out specific issues with some accounts, take them out of the picture and then continue the investigation on the bulk of the remaining accounts until everything is explained. Starting point could be trends by countries, types of publishers, etc… Whenever something that is identified as a bug is uncovered (examples: big advertiser down , CTR prediction off compared to actual values, RTB bids completely changed compared to previous week, etc…), they work with the escalation manager to open a separate bug (which will then be followed by the escalation manager).

If the global situation is particularly nasty or if the BI team is stuck in the investigation, we set-up an investigation SWAT team that meets as often as needed to decide on the next steps. SWAT team is made of: BI owner, escalation manager, Franck Le Ouay, Romain Niccoli, Nicolas Messelet (or central BD owner) + whoever is deemed important for the particular case. SWAT team should always be copied on all emails about the situation.

Key findings should be communicated to the prod_events alias (provided the issue is serious enough).

When the global performance drop is explained (not necessarily solved), the master bug is assigned back to the escalation manager who can then close it (after checking that all relevant information is in the bug).

The other bugs opened along the way for specific issues continue to follow the normal escalation process until resolution.

3.4.3. Follow-up

When the case is resolved, the escalation manager should conduct a post-mortem focused on (as appropriate):

- What can be done to detect such a situation automatically in the future?
- What can be done to prevent such a situation from happening again?
- Did we have the right tools to investigate the situation? If not, what do we need?

The conclusions of the post-mortem should be appended to the master bug.

4. Release management

Owner: Julien

Please see: http://intranet/wiki/Wiki%20Pages/Release%20Management.aspx

Additional information:

- The CTR Server must not be deployed on the same day as CAS.
- If stats are late, all deployments are postponed.

5. Incident management

Owner: Julien

Every Monday, we're holding a weekly meeting dedicated to the follow-up and resolution of production incidents:

- On-duty alerts
- Application/infrastructure issues, especially those undetected by our monitoring system
- Release issues (failed / rollbacked)

Mandatory participants:

- The on-duty guy who didn't sleep ;)
- Damien
- Romain
- Myself
- Anyone owning one action linked to the resolution of an incident

The list of incidents and the action plan are detailed in a document called "Incident Management.xls", which is is stored at http://intranet/engineering. An updated version will also be sent to *devteam* after each meeting.

If you're the owner of at least one action, you are required to:

- Work on the investigation / resolution. These tasks are HIGH PRIORITY.
- Keep me updated before the next meeting
- Attend the meetings until the action is 100% complete

If you do not own any action… patience, your time will come :-P

6. Infrastructure supplier management

Owner: Julien

Supplier competition is mandatory. There is no room for long-lasting relationships, friendships, etc if they don't deliver the best deal.

Suppliers must be evaluated on technology, roadmap, price, support, delivery times, etc. Depending on projects, these criteria will have different weights.

Supplier selection is my prerogative. I will listen to everyone but at the end of the day, I decide who we select & who we drop.

No supplier contact happens without me knowing and approving.

No quote gets signed without me knowing and approving.

7. Infrastructure change control management

Owner: Julien

7.1. Scope

"**Change**": a non-routine technical operation performed by a member of the Production team (or by a contractor) on live production infrastructure, involving a definite risk to our business operations, even if it is perfectly executed.

"**Non-routine**": something that isn't part of your daily stuff. Granted, it's not a perfect definition and you'll have to use your best judgment. If you're unsure whether one of your actions should follow this process or not, PLEASE ask me BEFORE proceeding! Obviously, this excludes emergency actions that have to be performed during on-duty hours (or possibly during normal office hours).

"**Technical operation**": anything that modifies the behavior of the system

- Explicit modification of the system configuration
- Impact on system performance, for example running a heavy one-shot job on a database server

"**Live production infrastructure**": any system involved in servicing real-life traffic & services

"**Risk**": the possibility of disrupting real-life traffic & services, with even a minor impact on business operations

If you're unsure whether one of your actions must follow this process or not, PLEASE ask BEFORE proceeding!

7.2. Process

Any change must be:

- **Registered** as early as possible by creating a new ticket in the "Infrastructure change" project : http://alert.criteo.prod/index.php?project=7 → ticket status is "**New**"
- **Explained & discussed** during the weekly Production meeting (every Tuesday), in order to refine the operation and to assess the actual risk(s)
- **Approved** by me → I will set the ticket status is "**Approved**"
- **Planned** in the least risky way (e.g. late at night) & communicated to all relevant parties well in advance (all the more so if it is very disruptive) → I will set the ticket status to "**Planned**" and fill in the "**Due Date**" field.
- **Announced** by e-mail right when it starts and once it's complete (or rolled back) → ticket status is "**Done**" (or "**Rolled back**")