

## openQA Project - coordination #91467

coordination # 91646 (Blocked): [saga][epic] SUSE Maintenance QA workflows with fully automated testing, approval and release

### [epic] Surface openQA failures per squad in a single place

2021-04-21 07:55 - vpelcak

|   |                                  |
|---|----------------------------------|
| <b>Status:</b> Resolved   | <b>Start date:</b> 2021-04-23    |
| <b>Priority:</b> Normal   | <b>Due date:</b>                 |
| <b>Assignee:</b> okurz  | <b>% Done:</b> 100%              |
| <b>Category:</b> Feature requests   | <b>Estimated time:</b> 0.00 hour |
| <b>Target version:</b> Ready  |                                  |
| <b>Difficulty:</b>  |                                  |
| <b>Description</b>  |                                  |
| <b>Motivation</b>   |                                  |
| As a squad team member, I want to be able to view Maintenance and Tumbleweed job group related problems that are relevant to my squad.  |                                  |
| Unlike <i>Product QA</i> where there is a single product version, one type of testing run, and tests in one job group per squad, the maintenance job groups are spread to 7 different versions of SLE, and in 3 different categories of testing. Even within those groups, both in the maintenance job groups and Tumbleweed job groups, there are separations which tests belong to which squad. |                                  |
| <b>Acceptance criteria</b>  |                                  |
| <ul style="list-style-type: none"><li><b>AC1:</b> Squad members can see which testcases their squad is responsible for within all maintenance job groups across all supported SLE versions are failing</li></ul>  |                                  |
| <b>Subtasks:</b>  |                                  |
| action # 91647: Making option to filter by flavor, test name on /tests/overview more pr...  | <b>Resolved</b>                  |
| action # 91650: Resolve the most recent builds per job group on /tests/overview when sh...  | <b>Resolved</b>                  |
| action # 91652: Remind about the use of openqa-review in squads   | <b>Resolved</b>                  |
| action # 92957: Add option to openqa-review to skip displaying all passed results   | <b>Resolved</b>                  |
| action # 93727: Publish openqa-review reports with "--skip-passed"  | <b>Resolved</b>                  |
| action # 94732: Provide link to /tests/overview of latest builds of all job groups with...  | <b>Resolved</b>                  |
| action # 94762: openqa-review: Add mode of single-line todo lists size:M  | <b>Resolved</b>                  |
| action # 96058: [spike] Filter test results on /tests or /tests/overview by regex match...  | <b>Resolved</b>                  |
| action # 97403: openqa-review: Polish job group section titles in todo-only mode size:S   | <b>Resolved</b>                  |
| action # 98258: No results on /tests/overview w/o build   | <b>Resolved</b>                  |
| action # 98445: improve description for "Test module" UI element as followup to #96058  | <b>Resolved</b>                  |
| action # 98460: Filter actual test results on /tests or /tests/overview by regex match ...  | <b>Resolved</b>                  |
| <b>Related issues:</b>  |                                  |
| Related to openQA Project - action #17252: notifications to maintainer on fai...  | <b>New</b> 2016-02-09            |
| Copied to openQA Project - coordination #91914: [epic] Make reviewing openQA ...  | <b>Blocked</b> 2016-02-09        |

### History

#1 - 2021-04-21 08:00 - okurz

- Project changed from openQA Infrastructure to openQA Project

- Category set to Feature requests

- Priority changed from Normal to Low

- Target version set to future

I think this fits better in "openQA Project" or do you consider a specific need for openqa.suse.de?

vpelcak wrote:

I think it would be very helpful if openQA was able to somehow automatically report failures to the chosen Rocket Chat channel.

Do you mean that test reviewers can select what patterns failing tests would match to and what Rocket.Chat channel to use for a configured set of tests, like a job group?

With [http://open.qa/docs/#\\_enable\\_custom\\_hook\\_scripts\\_on\\_job\\_done\\_based\\_on\\_result](http://open.qa/docs/#_enable_custom_hook_scripts_on_job_done_based_on_result) openQA has support for custom job hooks that could be used to do something like this. What we would need is at best a simple command line client to send messages to Rocket.Chat

**#2 - 2021-04-22 05:43 - tjrinski\_suse**

"As a squad team member, I want to be able to view Maintenance and Tumbleweed job group related problems that are relevant to my squad."

Unlike Product QA where there is a single product version, one type of testing run, and tests in one job group per squad, the maintenance job groups are spread to 7 different versions of SLE, and in 3 different categories of testing. Even within those groups, both in the maintenance job groups and Tumbleweed job groups, there are separations which tests belong to which squad. For that reason, manual daily clicking through all of them by all squads is not scalable, and we'd like to have an overview - not necessarily Rocket, or not necessarily only Rocket - that shows exactly the failing tests, and only the failing tests, that are relevant to us.

A summary of the Maintenance + Tumbleweed tests (which are similarly not split by squad) per squad is available at:

[https://confluence.suse.com/display/~tjrinski\\_suse/Maintenance+and+Tumbleweed+tests+per+squad](https://confluence.suse.com/display/~tjrinski_suse/Maintenance+and+Tumbleweed+tests+per+squad)

**#3 - 2021-04-22 06:08 - cdywan**

Thank you for adding your user story ☺☺☺

As an idea, it might be worth thinking of this as a view in the web UI, like a kind of saved query. Which then could be used for notifications as well. Since the tests each squad is interested in are different and a single notification wouldn't work.

**#4 - 2021-04-22 06:11 - cdywan**

- Description updated

**#5 - 2021-04-22 06:15 - cdywan**

- Description updated

**#6 - 2021-04-22 06:34 - cdywan**

- Subject changed from automated report of openQA failures via Rocket Chat to Surface openQA results per squad in a single place

Attempting to rename the ticket based on the use case rather than the specific suggestion.

**#7 - 2021-04-22 12:09 - okurz**

[vpelcak](#) can you please give a bit of context here why you created the ticket originally? I assume this is based on a meeting you had for a certain topic?

**#8 - 2021-04-22 12:26 - tjrinski\_suse**

[okurz](#) It was discussed in the yesterday's maintenance meeting, I asked Vit to file a ticket initially but it's now written by me and Christian, and based on George's original description of the idea.

It's related to the wish for squads to review their own tests, instead of qam-openqa reviewers, a bit similar to what's done in Product QA job groups already. But the maintenance (and Tumbleweed) tests are spread all over, so we'd want a single place per squad that shows the failures everywhere in their domain for review.

**#9 - 2021-04-22 12:48 - tjrinski\_suse**

Adding link to this done by QE Tools: <http://dashboard.qam.suse.de/blocked>

That's from incident point of view, but technically if 1) all items were expanded to show exact failing tests, 2) all non-failures were hidden, 3) sub-pages for squad specific filtering showing only those failing tests affecting the squad in question, we'd be close to goal... maybe? Aside from Tumbleweed.

**#10 - 2021-04-22 21:03 - okurz**

- Related to action #17252: notifications to maintainer on failed modules added

**#11 - 2021-04-22 21:16 - okurz**

- Subject changed from Surface openQA results per squad in a single place to [epic] Surface openQA results per squad in a single place

- Assignee set to okurz

- Priority changed from Low to Normal

- Target version changed from future to Ready

1. I wonder about the original request of "automatically inform (via Rocket.Chat) vs. the updated "a view of relevant test results". I see these as two different things with the crosssection of a definition of "interesting tests/scenarios/job groups". Can you all comment on which you would prefer?
2. What is the user stories (or multiple) behind that? Could you please specify something more goal oriented than "to see results"? I would like to come up with the best solution on the actual need
3. Specific question regarding Tumbleweed: I doubt you need to review Tumbleweed the same as for SLE. For Tumbleweed the tests are trusted enough so that the release managers review the overall test results and record rather specific tickets in case test maintainers can help. So, is Tumbleweed actually relevant in the same way?

#### #12 - 2021-04-23 04:15 - openqa\_review

- Due date set to 2021-05-07

Setting due date based on mean cycle time of SUSE QE Tools

#### #13 - 2021-04-23 12:13 - okurz

Talked with tjyrinki\_suse . We see two use cases:

1. Helping reviewing test failures, which ultimately users triggering changes to products should do themselves, e.g. the openSUSE release managers already do that, the SUSE SLE Maintenance coordination engineers should be able to do that but commonly do not do that right now.
2. Trying to find places for improvements, reducing the delta in test coverage, etc.

Related discussion: <https://confluence.suse.com/display/~vpelcak/Draft+-+Change+in+openQA+Review>

other ideas from SUSE QE Tools weekly meeting:

- use the "activity stream" not for users but for a configured set of "interesting tests"
- filtering by flavor on tests overview page is possible but not configurable in the filter box, could be extended -> [#91467](#)
- propose the use of "openqa-review" -> [#91652](#)
- Rocket.Chat bot ("csbot" is not able to do that, probably does not make sense that it learns that) -> [#91605](#)

#### #14 - 2021-04-23 13:37 - okurz

- Tracker changed from action to coordination

- Parent task set to [#91646](#)

Discussed with vpelcak. A view within openQA is distinct from feedback in smelt and OBS/IBS. So we should be very careful to not deepen the trenches between silos and teams, e.g. maintenance coordination engineers that receive comments on IBS about failed tests. IMHO maintenance coordination engineers can feel much more empowered and react quickly on test failures themselves after receiving such relevant test failure information. QE engineers can of course still review on a periodic base the test coverage and try to improve. However, I suggest to not overuse any openQA results view as a decision base for releaseability of products.

#### #15 - 2021-04-23 14:33 - okurz

- Status changed from New to Blocked

Created four specific subtasks, tracking

#### #16 - 2021-04-28 17:23 - okurz

- Copied to coordination [#91914](#): [epic] Make reviewing openQA results per squad easier added

#### #17 - 2021-04-28 17:26 - okurz

- Description updated

#### #18 - 2021-05-01 05:44 - okurz

- Status changed from Blocked to Resolved

With the subtask [#91652](#) done and the workshop about openqa-review completed I consider the work here done.

openqa-review can provide a view of openQA results in a single place, customizable per squad. The existing top-level reports are referenced in <https://progress.opensuse.org/projects/qa/wiki/Test-results-overview> . If anyone prefers a subset of the reported data I am sure we can accomodate that.

Further work as future improvement is planned in [#91914](#) to make reviewing results easier.

**#19 - 2021-05-03 07:47 - mgrifalconi**

Hello, I might have misunderstood something or be missing some context since I was on vacation at the end of last week, but:

- to my understanding, the goal was to have a place where each squad can see all its related tests (to later consider splitting the openqa review for updates on each squad instead of having a single reviewer)
- I see that in this ticket we agree that it is technically possible, but I don't see we have it done yet

As action items I can think of:

- agree on which test belongs to which squad (won't always be crystal clear separation, we need some compromise on who takes what in some cases)
- make sure there won't be orphan tests, each one need to have a squad
- implement a way to tag the test for each squad (maintainer field on perl module, on test suite? some query elsewhere with test names/suites/full jobgroups?)
- have some page where you can choose the squad and then see its related tests for each SLE version

Do we already have all of this? Have I misunderstood the goal?

EDIT: Or maybe all this things I am mentioning are to be discussed as part of [#91914](#) ?

**#20 - 2021-05-05 07:06 - okurz**

mgrifalconi wrote:

Hello, I might have misunderstood something or be missing some context since I was on vacation at the end of last week, but:

- to my understanding, the goal was to have a place where each squad can see all its related tests (to later consider splitting the openqa review for updates on each squad instead of having a single reviewer)

Correct. With the exception of not covering multiple instances I consider the reports linked on <https://progress.opensuse.org/projects/qa/wiki#Test-results-overview> as the place where each squad can see all its related tests. Of course the generic reports include more than just the tests for each squad. For this each squad would need to define job group and/or filter parameters that define the scope, similar to the reports that already exist for [QE Core and QE Yast](#)

- I see that in this ticket we agree that it is technically possible, but I don't see we have it done yet

Find existing reports on <https://progress.opensuse.org/projects/qa/wiki#Test-results-overview>, e.g. <http://s.qa.suse.de/test-status>

As action items I can think of:

- agree on which test belongs to which squad (won't always be crystal clear separation, we need some compromise on who takes what in some cases)

Sure. But I assume that every squad already did that and hopefully each job group, each test suite and each test module has a proper maintainer.

- make sure there won't be orphan tests, each one need to have a squad

This is a very good point. This is why I remind QE Project Managers that they must ensure that the overall results are reviewed and prevent such gaps.

- implement a way to tag the test for each squad (maintainer field on perl module, on test suite? some query elsewhere with test names/suites/full jobgroups?)

yes. The test maintainers field is already ensured within os-autoinst-distri-opensuse. At a time I ensured that each test suite has a maintainer but I am not aware of automatic checks that will ensure that. Also every job group should have maintainers but this has also already deteriorated.

- have some page where you can choose the squad and then see its related tests for each SLE version

Do we already have all of this? Have I misunderstood the goal?

EDIT: Or maybe all this things I am mentioning are to be discussed as part of [#91914](#) ?

Yes, that would be best. As this ticket here [#91467](#) was listed as "blocker" for process changes I wanted to provide a fast solution and split out "nice ideas for the future" elsewhere. I will note down the above ideas into [#91914](#)

**#21 - 2021-05-07 07:47 - tjyrinki\_suse**

okurz wrote:

- to my understanding, the goal was to have a place where each squad can see all its related tests (to later consider splitting the openqa review for updates on each squad instead of having a single reviewer)

Correct. With the exception of not covering multiple instances I consider the reports linked on <https://progress.opensuse.org/projects/qa/wiki#Test-results-overview> as the place where each squad can see all its related tests. Of course the generic reports include more than just the tests for each squad. For this each squad would need to define job group and/or filter parameters that define the scope, similar to the reports that already exist for [QE Core and QE Yast](#)

I think we would need to have an example of it being possible to have a filter that when used, only shows the failures in the job groups, test suites and individual test runs that are maintained by the squad. I guess this is <https://progress.opensuse.org/issues/91647>, but currently we are not able to say if it's possible to have a view for all squads that would be the pre-requisite for the openQA review proposal.

In addition, I think we'd need have a tooling creating automatic filtering based on Maintainer: in the .pm files in os-autoinst-opensuse. I think a ticket for that could be created. So, at first we'd likely use the manual filtering basically typing the sort of [https://confluence.suse.com/display/~tjyrinki\\_suse/Maintenance+and+Tumbleweed+tests+per+squad](https://confluence.suse.com/display/~tjyrinki_suse/Maintenance+and+Tumbleweed+tests+per+squad) in a form that the filtering form accepts (that page however only lists test suites, there might be even variation within the test suites but maybe those can be moved away so that each test suite is maintained by one squad), but when we get to point where each individual .pm, test suite and job group is correctly attributed to every applicable squad we could switch to the automated one. It's possible the currently available ways to set/imply maintainer for computer parsing are also not enough.

- I see that in this ticket we agree that it is technically possible, but I don't see we have it done yet

Find existing reports on <https://progress.opensuse.org/projects/qa/wiki#Test-results-overview>, e.g. <http://s.qa.suse.de/test-status>

Those only work properly for Product QA where there is a single job group per squad, not subsets of tests and testsuites across 10+ job groups so that everything else is hidden from view. So the concrete examples mentioned above would be a pre-requisite.

Sure. But I assume that every squad already did that and hopefully each job group, each test suite and each test module has a proper maintainer.

This is a bit wrong assumption, as even though squads are aware of their territory, it's a very big job to ensure that all test suites and all individual tests have the proper maintainer set. Even that is not enough actually, as there are currently cases where the same test is maintained by a different squad depending on which product it's run, so all of these cases should be supported to have a view where only the area of squad is shown.

- make sure there won't be orphan tests, each one need to have a squad

This is a very good point. This is why I remind QE Project Managers that they must ensure that the overall results are reviewed and prevent such gaps.

I agree fully. There are community maintained tests on O3 side, but on OSD side everything should be maintained by some squad (or be deprecated).

- have some page where you can choose the squad and then see its related tests for each SLE version Do we already have all of this? Have I misunderstood the goal?

I think this was just the goal we need, have a per squad page. We don't have this, only pages that show also irrelevant bits and for example no possibility to hide everything but failures.

Yes, that would be best. As this ticket here [#91467](#) was listed as "blocker" for process changes I wanted to provide a fast solution and split out "nice ideas for the future" elsewhere. I will note down the above ideas into [#91914](#)

It can be moved there, but I current it feels like these ideas are still blockers too and there is no currently available solution that would be usable as is. That is, a single page that shows only failures, and only those failures that are in job groups, tests suites and individual tests that are maintained by the squad. Anything else will make the maintenance review too big cumbersome as all squads need to manually think which parts of the pages are irrelevant to them every day, and potentially browse several different pages.

**#22 - 2021-05-07 12:11 - cdywan**

- Status changed from Resolved to Feedback

Reading the latest comments after this was *Resolved* I'm genuinely confused. I thought [#91914](#) was meant to supersede this ticket, but you seem to be arguing what the OP's user story is and I'm not clear that it is covered by the new epic. Hence re-opening.

**#23 - 2021-05-08 05:08 - okurz**

[cdywan](#) I agree that it's a safer approach to reopen but I would have answered anyway. You can trust that it's very unlikely that I would overlook a

comment in progress but your approach is safer ;)

So let's try to clarify ...

tjyrinki\_suse wrote:

I think we would need to have an example of it being possible to have a filter that when used, only shows the failures in the job groups, test suites and individual test runs that are maintained by the squad. I guess this is <https://progress.opensuse.org/issues/91647>, but currently we are not able to say if it's possible to have a view for all squads that would be the pre-requisite for the openQA review proposal.

I am not sure I understand. So by default openqa-review only shows "failed" test results. openqa-review accepts selection parameters for job groups that should be included or excluded. What do you mean with "individual test runs"?

In addition, I think we'd need have a tooling creating automatic filtering based on Maintainer: in the .pm files in os-autoinst-opensuse. I think a ticket for that could be created. So, at first we'd likely use the manual filtering basically typing the sort of [https://confluence.suse.com/display/~tjyrinki\\_suse/Maintenance+and+Tumbleweed+tests+per+squad](https://confluence.suse.com/display/~tjyrinki_suse/Maintenance+and+Tumbleweed+tests+per+squad) in a form that the filtering form accepts (that page however only lists test suites, there might be even variation within the test suites but maybe those can be moved away so that each test suite is maintained by one squad), but when we get to point where each individual .pm, test suite and job group is correctly attributed to every applicable squad we could switch to the automated one. It's possible the currently available ways to set/imply maintainer for computer parsing are also not enough.

- I see that in this ticket we agree that it is technically possible, but I don't see we have it done yet

Find existing reports on <https://progress.opensuse.org/projects/qa/wiki#Test-results-overview>, e.g. <http://s.qa.suse.de/test-status>

Those only work properly for Product QA where there is a single job group per squad, not subsets of tests and testsuites across 10+ job groups so that everything else is hidden from view. So the concrete examples mentioned above would be a pre-requisite.

Would it help to extend openqa-review with filter parameters (blocklist/passlist) on test suite names? And would that be enough so that you can continue with <https://confluence.suse.com/pages/viewpage.action?spaceKey=-vpelcak&title=Draft+-+Change+in+openQA+Review>? this is something that we might be able to achieve within a reason timeframe.

Sure. But I assume that every squad already did that and hopefully each job group, each test suite and each test module has a proper maintainer.

This is a bit wrong assumption, as even though squads are aware of their territory, it's a very big job to ensure that all test suites and all individual tests have the proper maintainer set. Even that is not enough actually, as there are currently cases where the same test is maintained by a different squad depending on which product it's run, so all of these cases should be supported to have a view where only the area of squad is shown.

hm, given the dynamic and complicated nature of "I maintain this test module but only in this job group, but not for that architecture and elsewhere its someone else" this is a rather unrealistic expectation especially as that sounds like a very self-made and SUSE specific problem. It is just unlikely that following this path we will find a technical solution that suits the problem.

- make sure there won't be orphan tests, each one need to have a squad

This is a very good point. This is why I remind QE Project Managers that they must ensure that the overall results are reviewed and prevent such gaps.

I agree fully. There are community maintained tests on O3 side, but on OSD side everything should be maintained by some squad (or be deprecated).

- have some page where you can choose the squad and then see its related tests for each SLE version Do we already have all of this? Have I misunderstood the goal?

I think this was just the goal we need, have a per squad page. We don't have this, only pages that show also irrelevant bits and for example no possibility to hide everything but failures.

Yes, that would be best. As this ticket here [#91467](#) was listed as "blocker" for process changes I wanted to provide a fast solution and split out "nice ideas for the future" elsewhere. I will note down the above ideas into [#91914](#)

It can be moved there, but I current it feels like these ideas are still blockers too and there is no currently available solution that would be usable as is. That is, a single page that shows only failures, and only those failures that are in job groups, tests suites and individual tests that are maintained by the squad. Anything else will make the maintenance review too big cumbersome as all squads need to manually think which parts

of the pages are irrelevant to them every day, and potentially browse several different pages.

I am thankful that you already gave good examples above of challenging cases. I think we would be able to provide more help for the "happy path" and the common part of the problem but as an important requirement should be that "no test is overlooked" as well as "squads don't do duplicate work" I consider the following cornercases which should define the design:

1. test cases (that is both test suites or test modules) maintained by different people depending on the environment
2. test cases not maintained by anyone, e.g. due to miscommunication which can always happen a test case is not looked at by any team assuming that "another" team would handle
3. test cases that are mostly shared and that are critical so that it is likely that any person seeing these fail would take a look anyway, regardless who is the actual maintainer

I am not sure if it is a good idea or feasible to try to split the scopes of teams and have dedicated, pre-defined filtered views so that both case 1 and 3 can be handled efficiently without people doing duplicate review work as well as ensuring that case 2 is unlikely to happen.

Let's imagine we would define a single page that shows results for QE-Core based on your definition in

[https://confluence.suse.com/display/~tjyrinki\\_suse/Maintenance+and+Tumbleweed+tests+per+squad](https://confluence.suse.com/display/~tjyrinki_suse/Maintenance+and+Tumbleweed+tests+per+squad) – which is already becoming hard to read and likely hard to get right when following a plain format which is not a multi-dimensional table :D – if I understand that correctly that would mean that QE-Core would see and hence need to review any test failure of "qam-minimal+base" regardless of the product/version/branch/arch and regardless of the actual test modules, right? But another squad would look at specific test modules, let's say [https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/tests/boot/boot\\_to\\_desktop.pm](https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/tests/boot/boot_to_desktop.pm) is looked at by yutao [yuwang@suse.com](mailto:yuwang@suse.com) as listed in the test module as maintainer. I can imagine what will happen most likely: Maybe both do duplicate work at the beginning but very quickly people will assume "whenever boot\_to\_desktop pops up as failed, let's hope the other will take care (I have already so much to do)". Also, a scenario like "qam-minimal+base" would be used in cases like "on submission" tests, see <https://confluence.suse.com/pages/viewpage.action?pageId=723878219>. Would you want to take a look into "qam-minimal+base" failing due to obvious problems with the submitted grub2/kernel/plymouth/systemd submission? Just having the submitter informed (which is already done automatically e.g. over IBS comments) is enough, the submitter can still ask QE-Core for help, be it over a progress ticket or ping you in chat, right? Another case would be where many jobs fail because of a common problem, let's say some syntax error. What good would it be if every squad tries to solve that on their squad-view? In these cases there are other views or communication channels that are better. In summary for this paragraph I want to say: Likely it's better to use different views for the different use cases and the different questions of a periodic review and we are happy to help improve all of these different views, e.g. be it openQA view enhancements, openqa-review, chat notifications, custom bash snippets with API calls. By the way, have you seen [#91542#note-12](#)? Maybe based on that you can have something that helps as well? Not saying that it is "the solution" but part of a bigger picture.

Also, are you aware of [https://wiki.suse.net/index.php/RD-OPS\\_QA/openQA\\_review](https://wiki.suse.net/index.php/RD-OPS_QA/openQA_review)? If not then that could explain some problems we see :D

Oh, and I don't think it's wise to say to be "blocked" on

<https://confluence.suse.com/pages/viewpage.action?spaceKey=~vpehcak&title=Draft+--+Change+in+openQA+Review> waiting for an imaginary new openQA feature when we already have the same tests that you maintain used for so much more than just the job groups you look at, e.g. SLE staging, openSUSE staging, Carwos, etc.

What I have seen working out best so far is if people take a look at both /tests/overview of openQA as well as openqa-review output and work down these "todo" reports using more generic, bigger report which help to provide context and help each other until there are no more unhandled, unreviewed test failures in openQA.

#### #24 - 2021-05-18 07:31 - tjyrinki\_suse

okurz wrote:

I am not sure I understand. So by default openqa-review only shows "failed" test results. openqa-review accepts selection parameters for job groups that should be included or excluded. What do you mean with "individual test runs"?

Basically ability to list also test suites under specific job groups that would be included in the report, and under those test suites possibly only individual test modules.

I tried using openqa-review with some job groups to see whether the output could be grepped and combined somehow to form that "squad page" that we're hoping to have.

Eg openqa-review -J [https://openqa.suse.de/group\\_overview/308](https://openqa.suse.de/group_overview/308) says eg there's a problem on mau-extratests1 but these issues make it hard to use that output alone as part of overall review page:

1. it does not say under which flavor the failure is under, there could be different squad handling a failure depending on the flavor
2. it does not have url associated to the failing
3. it does not list and link to which test module is failing (direct link to actual failure)
4. I don't see a way to grep the output so that all headers and such would be hidden, only test failures of those tests that are maintained by the squad

It'd seem we'd by minimum need to be give parameters job group -> flavor -> test suite, ie only show results for flavor(s) under the job group, and under that flavor only the specific test suites. Also it'd be useful if the output as is would be usable without trying to grep the correct lines, ie output failures with links to the failing point, and possible related bugs and tickets linked all on the same line, instead of eg empty sections, division by archs etc. More like a dense list of failures. I tried --short-failure-str --abbreviate-test-issues but it didn't do much to the basic very verbose mode of output.

Would it help to extend openqa-review with filter parameters (blocklist/passlist) on test suite names? And would that be enough so that you can continue with <https://confluence.suse.com/pages/viewpage.action?spaceKey=~vpehcak&title=Draft+--+Change+in+openQA+Review>? this is

something that we might be able to achieve within a reason timeframe.

I think that flavor + test suite would likely be enough so that job group + flavor + passlist of test suite names. I've mentioned also a test suite could have individual modules maintained by different squads, but if those cases are found it makes more sense to separate those to different test suites.

The flavor would be needed since there are different flavors maintained by different squads, but which may have the same test suites running - however the respective squad should review the flavor results as a whole.

With that I think we'd have a way of having the most rudimentary tool so that doing some tens of openqa-review runs, we'd have a page that would contain the relevant failures.

The output from openqa-review should include "just the right links" and possibility to clean up the output.

The preferred output (meta)format would be:

fail: job group, product, flavor, [test suite](#), (architecture, linked product bug, linked ticket)

So that each failure would have a single line with all relevant information related to the failure, instead of there being sections like "architecture", "new product bugs", "skipped tests" etc.

If we could have that kind of output, we could codify eg <https://confluence.suse.com/display/qasle/Tests+Maintained+by+QE+Core> into openqa-review commands that would result in this squad page we're hoping to get.

I am thankful that you already gave good examples above of challenging cases. I think we would be able to provide more help for the "happy path" and the common part of the problem but as an important requirement should be that "no test is overlooked" as well as "squads don't duplicate work" I consider the following cornercases which should define the design:

1. test cases (that is both test suites or test modules) maintained by different people depending on the environment
2. test cases not maintained by anyone, e.g. due to miscommunication which can always happen a test case is not looked at by any team assuming that "another" team would handle
3. test cases that are mostly shared and that are critical so that it is likely that any person seeing these fail would take a look anyway, regardless who is the actual maintainer

The not overlooking tests is a good point, and makes it maybe more important not for each team to manually craft openqa-review commands they think are correct, but instead have a global page, configuration for which could be git managed where the maintenance areas could be defined and also the page would automatically show eg test suites currently "dropped out" ie not included on any squads' views.

most likely: Maybe both do duplicate work at the beginning but very quickly people will assume "whenever boot\_to\_desktop pops up as failed, let's hope the other will take care (I have already so much to do)"

I think the review part would be able to be clear - squad reviews their flavors & test suites. But if there's a module specific problem in a module that is general and maintained by someone else, the maintainer would be asked to look at it. However, like qam-openqa reviewers currently do, it'd be squads' responsibility to make sure something happens about a problem in case of failures, but not necessarily they automatically are supposed to fix all problems, just be aware.

Oh, and I don't think it's wise to say to be "blocked" on

<https://confluence.suse.com/pages/viewpage.action?spaceKey=~vpelcak&title=Draft+-+Change+in+openQA+Review> waiting for an imaginary new openQA feature when we already have the same tests that you maintain used for so much more than just the job groups you look at, e.g. SLE staging, openSUSE staging, Carwos, etc.

I understand there could be endless associated feature requests. I just try to see a path that each squad could review their results with a one click daily instead of dwelling into 20+ job groups, two openQA instances etc. and filtering manually (in head) the results to see which affect our squad or not. So we'd need to be able to do filtering where the filtering is done separately, and that is granular enough that the squad specific maintainerships are used. And then have a full single results page that does not need to be crafted manually but is automatically generated based on that filtering.

What I have seen working out best so far is if people take a look at both /tests/overview of openQA as well as openqa-review output and work down these "todo" reports using more generic, bigger report which help to provide context and help each other until there are no more unhandled, unreviewed test failures in openQA.

There are 20+ /tests/overview page that would need to be checked, and again manually filtered by everyone looking at them. Currently just one person in qam-openqa review looks at them and even that is deemed too much work currently, but when distributed to squads people from multiple squads could be looking at the same pages, all doing that clicking through and manually filtering, and since the person would change every team member would need to remember fully the maintenance areas of the squad.

openqa-review would have the potential, with the output changed and flavor+test suite filtering added, but there should be a single place where configuration of the filtering used would be something visible to everyone so that all squads could check others' pages too. The fact that tens of openqa-review calls would be still needed to gather a single squad page is not a problem, as that could be hidden in the background in the web page generator.

I agree there are also communication related topics to handle, but the reviewing topic is anyway only about the very first step, that someone notices a failure and acts, similar to qam-openqa doing currently full-time (but we don't want anyone to do that full time, and surely not even more people doing



that full time) but distributed to squads and including both product, maintenance and openSUSE QA.

## #25 - 2021-05-18 12:46 - okurz

tjyrinki\_suse wrote:

okurz wrote:

I am not sure I understand. So by default openqa-review only shows "failed" test results. openqa-review accepts selection parameters for job groups that should be included or excluded. What do you mean with "individual test runs"?

Basically ability to list also test suites under specific job groups that would be included in the report, and under those test suites possibly only individual test modules.

I tried using openqa-review with some job groups to see whether the output could be grepped and combined somehow to form that "squad page" that we're hoping to have.

Eg openqa-review -J [https://openqa.suse.de/group\\_overview/308](https://openqa.suse.de/group_overview/308) says eg there's a problem on mau-extratests1 but these issues make it hard to use that output alone as part of overall review page:

1. it does not say under which flavor the failure is under, there could be different squad handling a failure depending on the flavor

ok, including the flavor is possible but before investing the effort let's discuss the other points first

1. it does not have url associated to the failing
2. it does not list and link to which test module is failing (direct link to actual failure)

You mean after you called openqa-review yourself? Look for the "-T" option which can be specified multiple times. Then you will have URLs to failing jobs

1. I don't see a way to grep the output so that all headers and such would be hidden, only test failures of those tests that are maintained by the squad ... It'd seem we'd by minimum need to be give parameters job group -> flavor -> test suite, ie only show results for flavor(s) under the job group, and under that flavor only the specific test suites.

Not sure how we should be able to define that on a different level than because it seems you want to cover multiple dimensions of instance+product+version+flavor+arch+testsuite+modules in a single view.

Also it'd be useful if the output as is would be usable without trying to grep the correct lines, ie output failures with links to the failing point, and possible related bugs and tickets linked all on the same line, instead of eg empty sections, division by archs etc. More like a dense list of failures. I tried --short-failure-str --abbreviate-test-issues but it didn't do much to the basic very verbose mode of output.

Did you try --no-empty-sections? The "dense list of failures" sounds useful. I think coming up with something like that from the /tests route on openQA can be more straightforward.

Would it help to extend openqa-review with filter parameters (blocklist/passlist) on test suite names? And would that be enough so that you can continue with <https://confluence.suse.com/pages/viewpage.action?spaceKey=~vpelcak&title=Draft+-+Change+in+openQA+Review> ? this is something that we might be able to achieve within a reason timeframe.

I think that flavor + test suite would likely be enough so that job group + flavor + passlist of test suite names. I've mentioned also a test suite could have individual modules maintained by different squads, but if those cases are found it makes more sense to separate those to different test suites.

yes, separating into different test suites was also the approach taken so far within former squads. Especially between QE YaST and former QE Functional this is how it was done. Keep in mind though that this already increased the discrepancy between openqa.opensuse.org and openqa.suse.de and will likely increase the gap. And the things are not split up further on openqa.opensuse.org for a good reason: Mainly because the *main* reviewers there are the people that directly benefit and have the products-under-test under control, i.e. openSUSE Release Managers.

The flavor would be needed since there are different flavors maintained by different squads, but which may have the same test suites running - however the respective squad should review the flavor results as a whole.

With that I think we'd have a way of having the most rudimentary tool so that doing some tens of openqa-review runs, we'd have a page that would contain the relevant failures.

The output from openqa-review should include "just the right links" and possibility to clean up the output.

The preferred output (meta)format would be:

fail: job group, product, flavor, [test suite](#), (architecture, linked product bug, linked ticket)

So that each failure would have a single line with all relevant information related to the failure, instead of there being sections like "architecture",

"new product bugs", "skipped tests" etc.

If we could have that kind of output, we could codify eg <https://confluence.suse.com/display/qasle/Tests+Maintained+by+QE+Core> into openqa-review commands that would result in this squad page we're hoping to get.

I think for this something custom built based on openQA API calls might be a potential alternative in the long run rather than doing that within openqa-review because it's again a different focus. The main focus for openqa-review was "What is the overall status of my product(s)" but here we are looking for "what problems that we maintain need work" and we might better off coming up with a separate approach for that ... in the long run :)

I am thankful that you already gave good examples above of challenging cases. I think we would be able to provide more help for the "happy path" and the common part of the problem but as an important requirement should be that "no test is overlooked" as well as "squads don't do duplicate work" I consider the following cornercases which should define the design:

1. test cases (that is both test suites or test modules) maintained by different people depending on the environment
2. test cases not maintained by anyone, e.g. due to miscommunication which can always happen a test case is not looked at by any team assuming that "another" team would handle
3. test cases that are mostly shared and that are critical so that it is likely that any person seeing these fail would take a look anyway, regardless who is the actual maintainer

The not overlooking tests is a good point, and makes it maybe more important not for each team to manually craft openqa-review commands they think are correct, but instead have a global page, configuration for which could be git managed where the maintenance areas could be defined and also the page would automatically show eg test suites currently "dropped out" ie not included on any squads' views.

That would only work if on that level we again programatically check that "each test/flavor/testsuite/whatever is mentioned on at least one squad's review list" which sounds complicated and convoluted. I would go with: "As long as there are unreviewed tests someone is not done yet" :)

...

What I have seen working out best so far is if people take a look at both /tests/overview of openQA as well as openqa-review output and work down these "todo" reports using more generic, bigger report which help to provide context and help each other until there are no more unhandled, unreviewed test failures in openQA.

There are 20+ /tests/overview page that would need to be checked, and again manually filtered by everyone looking at them. Currently just one person in qam-openqa review looks at them and even that is deemed too much work currently, but when distributed to squads people from multiple squads could be looking at the same pages, all doing that clicking through and manually filtering, and since the person would change every team member would need to remember fully the maintenance areas of the squad.

openqa-review would have the potential, with the output changed and flavor+test suite filtering added, but there should be a single place where configuration of the filtering used would be something visible to everyone so that all squads could check others' pages too.

Sounds like a wiki page that needs diligent moderation work likely by project managers that are independent of squads. So far I have not seen something like this work out in practice.

The fact that tens of openqa-review calls would be still needed to gather a single squad page is not a problem, as that could be hidden in the background in the web page generator.

correct. Letting scripts call openqa-review 10x is more efficient than humans doing the same manually :)

So I assume you don't see the acceptance criterion "Squad members can view relevant tests in a single place" covered because even if there is a place where squad members can see all relevant tests (from one instance) without filtering and getting rid of headers it's too hard to read, right?

#### **#26 - 2021-05-21 06:56 - vpealcak**

- Subject changed from [epic] Surface openQA results per squad in a single place to [epic] Surface openQA failures per squad in a single place

#### **#27 - 2021-06-09 21:21 - okurz**

- Related to action #92957: Add option to openqa-review to skip displaying all passed results added

#### **#28 - 2021-06-09 21:26 - okurz**

[tjyrinki\\_suse](#) there are some open points in [#91467#note-25](#) where I would appreciate your answer.

In the meantime we are progressing with some users stories where we have a better defined expectations. One thing that you also brought up is to skip sections of the openqa-review report that include only "passed" results. Please see the generated output in [https://w3.nue.suse.com/~okurz/openqa\\_suse\\_de\\_skip\\_passed.html](https://w3.nue.suse.com/~okurz/openqa_suse_de_skip_passed.html) in contrast to the default [https://w3.nue.suse.com/~okurz/openqa\\_suse\\_de\\_status.html](https://w3.nue.suse.com/~okurz/openqa_suse_de_status.html).

#### **#29 - 2021-06-24 06:38 - tjyrinki\_suse**

okurz wrote:

[tjyrinki\\_suse](#) there are some open points in [#91467#note-25](#) where I would appreciate your answer.

In the meantime we are progressing with some users stories where we have a better defined expectations. One thing that you also brought up is to skip sections of the openqa-review report that include only "passed" results. Please see the generated output in [https://w3.nue.suse.com/~okurz/openqa\\_suse\\_de\\_skip\\_passed.html](https://w3.nue.suse.com/~okurz/openqa_suse_de_skip_passed.html) in contrast to the default [https://w3.nue.suse.com/~okurz/openqa\\_suse\\_de\\_status.html](https://w3.nue.suse.com/~okurz/openqa_suse_de_status.html).

Hello. I think we addressed or at least felt like addressed those issues in a subsequent call.

There were some faults with openqa-review output that were fixed with correct parameters, and some that were not. And we agreed that given that we already usually should have a relatively short list of failures, maybe a rather global list with at most a few tens of failures could be started with.

The format could be one line per failure, as given example above, so that the list would be very dense but also include all information on one page without extra clicks. The idea would be that it would be human parseable quickly enough that even though there'd be like 7 people reviewing the list, it wouldn't cause the same amount of time consumption (and frustration) as clicking through tens of pages.

#### **#30 - 2021-06-25 20:20 - okurz**

ok. We will progress further on the ideas that you have mentioned and see what is feasible and how it can be represented. For openqa-review there is an intermediate step with multiple reports now collected on

<https://openqa.io.suse.de/openqa-review/>

Also we have now a possibility to better resolve "the latest build" within the tests overview pages of openQA so we can have pages like

<https://openqa.suse.de/tests/overview?todo=1&groupid=366&groupid=308&groupid=232&groupid=165&groupid=218&groupid=280&groupid=108&groupid=54#>

showing test results that need review from all most recent aggregate test results.

Can you please state if you see the above fulfilling the specified acceptance criterion? If not, please help me to find a new acceptance criterion so that we can know when the work on the epic is sufficient.

#### **#31 - 2021-07-02 08:34 - okurz**

There is also "openqa-revtui" from <https://github.com/grisu48/openqa-mon> which can show the current status of jobs and any need for review depending on the selected scope of tests. With [#94762](#) we plan to come up with something similar but for interactive review work maybe "openqa-revtui" is better, also because it has dynamic updates on jobs depending on rabbitMQ messages.

[tjyrinki\\_suse](#) Can you please state if you see one or multiple of the the above fulfilling the specified acceptance criterion? If not, please help me to find a new acceptance criterion so that we can know when the work on the epic is sufficiently done.

#### **#32 - 2021-07-15 10:14 - okurz**

- Status changed from Feedback to Resolved

No response. With this and all the recent updates I try again to consider the epic completed. Nevertheless the work continues in [#91914](#)

#### **#33 - 2021-07-15 14:17 - vpelcak**

- Description updated

- Status changed from Resolved to In Progress

#### **#34 - 2021-07-16 04:09 - openqa\_review**

Setting due date based on mean cycle time of SUSE QE Tools

#### **#35 - 2021-07-17 04:09 - openqa\_review**

Setting due date based on mean cycle time of SUSE QE Tools

#### **#36 - 2021-07-17 07:49 - okurz**

- Status changed from In Progress to Workable

With the extension of the acceptance criteria I will need to look into moving more subtasks from the follow-up epic here

#### **#37 - 2021-07-19 12:25 - okurz**

- Status changed from Workable to Blocked

Moved more tickets here. [#94762](#) is currently in progress and should fulfill the wish stated by [tjyrinki\\_suse](#) and objectively fulfilling AC1. However, assuming that AC1 might mean the more restricted version "Squad members can see only testcases their squad is responsible for and not any other failures" then both current open subtasks [#94762](#) as well as [#92921](#) will not be able to cover that. Likely that version of AC would not be feasible to

reach at all as there can always be overlapping cases.

Again, with [#91650](#) and [#94732](#) openQA provides links to test overview pages that show combined all test results within a parent job group. This allows to have queries ready like <https://openqa.suse.de/tests/overview?groupid=366&groupid=308&groupid=232&groupid=165&groupid=280&groupid=218&groupid=108&groupid=54&todo=1> which show all openQA test failures within the aggregate test results that need review which I consider should be very helpful and hopefully in combination with the other mentioned features enough to resolve if you agree. Still, setting tickets to "Blocked" to wait for the two subtasks [#94762](#) and [#92921](#)

#### **#38 - 2021-07-21 12:44 - okurz**

With [#94762](#) we have a TODO list with individual jobs needing review per line, see [https://openqa.io.suse.de/openqa-review/openqa\\_suse\\_de\\_todo\\_only.html](https://openqa.io.suse.de/openqa-review/openqa_suse_de_todo_only.html) . We would be happy to receive your feedback.

#### **#39 - 2021-08-05 07:06 - tjrinki\_suse**

Thank you for the new feature! I think it's a great start. It's not maybe usable right away because the list is much longer that one would have hoped for (hope was 30-40 lines) and it'd need more polishing. Those could be added as additional work items.

It's not well readable at the moment since the job group sections are not separated and everything is a <li>. Under a job group the sorting is not obvious, maybe it's per flavor, and it would help if there was a subheader for each flavor since teams would need to skip certain flavors under certain job groups.

Secondly, I'd hope there was a way to customize the view - in a way that the selections can be bookmarked - to hide certainly "not interesting to us" job groups and flavors. That might help to get the length under control so that it would be feasible to review daily by every squad.

I can anyway see this feature is likely very near to providing the minimal requirements for the daily reviews to start, after polishing and deploying to main OSD.

#### **#40 - 2021-08-17 09:18 - okurz**

tjrinki\_suse wrote:

[...]

I can anyway see this feature is likely very near to providing the minimal requirements for the daily reviews to start, after polishing and deploying to main OSD.

the provided solution is already live for OSD. I would plan any polishing only after seeing that users use it. So, with this, do you see the current solution enough to fulfill the ticket?

#### **#41 - 2021-08-17 11:33 - cdywan**

tjrinki\_suse wrote:

Secondly, I'd hope there was a way to customize the view - in a way that the selections can be bookmarked - to hide certainly "not interesting to us" job groups and flavors. That might help to get the length under control so that it would be feasible to review daily by every squad.

I seem to remember discussing this on a call, but it doesn't seem like a comment was added here. Sorry about that.

Did you see the YAML which supports setting the arguments for reports freely? I'm not sure all features are covered yet, but it should help iterating towards the desired format: <https://gitlab.suse.de/openqa/openqa-review/-/blob/master/review-jobs.yml>

#### **#42 - 2021-08-18 06:24 - tjrinki\_suse**

- *File review-report-mockup.png added*

okurz wrote:

the provided solution is already live for OSD. I would plan any polishing only after seeing that users use it. So, with this, do you see the current solution enough to fulfill the ticket?

I think the right question might be "would users use it if it would be more polished", and after staring at it a bit yes I could see myself happily using it. Of course I can only speak for myself. I tried mocking up a view and thinking what would be needed for it to be good, see example attached. It has

- Job group names with different color, font size etc (color: #f00; font-size:large; in this example)
- Flavors displayed in bold text
- Failed modules not hidden under hover, but displayed directly, and also are a direct link to the module failure (no need to scroll)
- Maintainer of the failed modules displayed (this would also encourage to fix where eg proper squad maintainer is not shown)

I can see there could be technical problems implementing such changes though, as if I understand correctly the html is currently generated from text output, which loses the context information like whether the string being printed is a job group, flavor, a test etc (not sure). That would be needed for correct formatting.

Second problem would be that it seems to be the ordering of failures seems to be somewhat random under a job group, and it'd need to be so that every failure in each flavor is listed in one go, so that the headers in my mock up would work and the tests would also be listed alphabetically under the flavor similar to actual openQA job test run page like <https://openqa.suse.de/tests/overview?distri=sle&version=15-SP2&build=391.14&groupid=321>

cdywan wrote:

Secondly, I'd hope there was a way to customize the view - in a way that the selections can be bookmarked - to hide certainly "not interesting to us" job groups and flavors. That might help to get the length under control so that it would be feasible to review daily by every squad.

I seem to remember discussing this on a call, but it doesn't seem like a comment was added here. Sorry about that.

Did you see the YAML which supports setting the arguments for reports freely? I'm not sure all features are covered yet, but it should help iterating towards the desired format: <https://gitlab.suse.de/openqa/openqa-review/-/blob/master/review-jobs.yml>

That does look good, with that customization one could eg list the certain maintenance job groups + Functional job group, and the list would be shorter. Squads could offer their own custom yaml likely. If the flavors would be marked as in my mockup and entries sorted, I'd also argue that it'd visually then be easy enough to skip flavors even multiple times a day so that there wouldn't need to be a feature to filter certain flavors out.

Then "one more thing" would be ability to have on that single page errors from both OSD and O3 (certain job groups).

#### #43 - 2021-08-23 14:59 - okurz

tjyrinki\_suse wrote:

[...]

Do I understand you correctly that if we implement (more or less) the visual changes you proposed for the openqa-review reports that you would be ok to set this ticket to "Resolved"?

#### #44 - 2021-08-25 06:29 - tjyrinki\_suse

okurz wrote:

Do I understand you correctly that if we implement (more or less) the visual changes you proposed for the openqa-review reports that you would be ok to set this ticket to "Resolved"?

Ack, with the proposed visual/data/sorting changes it'd seem like a great page, together with the fact we can create new pages like that by contributing the YAML mentioned.

#### #45 - 2021-08-25 07:03 - tjyrinki\_suse

- *File review-report-mockup2.png added*

As discussed with Vit, here's another mockup that simply shows the idea that the page would have proper links at the failures that have a ticket or bug filed, as usual in openQA. Then it would be easy for reviewers to check which failures still need immediate action (first being filing a ticket).

#### #46 - 2021-08-27 06:42 - okurz

tjyrinki\_suse wrote:

As discussed with Vit, here's another mockup that simply shows the idea that the page would have proper links at the failures that have a ticket or bug filed, as usual in openQA. Then it would be easy for reviewers to check which failures still need immediate action (first being filing a ticket).

The full reports like [https://openqa.io.suse.de/openqa-review/openqa\\_suse\\_de\\_status.html](https://openqa.io.suse.de/openqa-review/openqa_suse_de_status.html) already show the failed tests that have ticket links. The "todo-only" page only shows the tests that do not have any label, i.e. need review

#### #47 - 2021-08-31 11:47 - cdywan

okurz wrote:

tjyrinki\_suse wrote:

As discussed with Vit, here's another mockup that simply shows the idea that the page would have proper links at the failures that have a ticket or bug filed, as usual in openQA. Then it would be easy for reviewers to check which failures still need immediate action (first being filing a ticket).

The full reports like [https://openqa.io.suse.de/openqa-review/openqa\\_suse\\_de\\_status.html](https://openqa.io.suse.de/openqa-review/openqa_suse_de_status.html) already show the failed tests that have ticket links. The "todo-only" page only shows the tests that do not have any label, i.e. need review

Notes from my conversation with [tjyrinki\\_suse](#):

- ~~We could make the sections collapsible and expandable~~ We should find a way to make reports readable out of the box, and rather have a separate report than trying to find "the" solution for everyone
- The **todo** could be a lot more useful
  - If we sort it alphabetically
  - If we can have flavor headers
  - If we can include ticket links
  - For the purposes of this conversation, we probably want **a variant** of the existing todo list with new options configured in the YAML

#### #48 - 2021-08-31 12:17 - okurz

cdywan wrote:

okurz wrote:

tjyrinki\_suse wrote:

As discussed with Vit, here's another mockup that simply shows the idea that the page would have proper links at the failures that have a ticket or bug filed, as usual in openQA. Then it would be easy for reviewers to check which failures still need immediate action (first being filing a ticket).

The full reports like [https://openqa.io.suse.de/openqa-review/openqa\\_suse\\_de\\_status.html](https://openqa.io.suse.de/openqa-review/openqa_suse_de_status.html) already show the failed tests that have ticket links. The "todo-only" page only shows the tests that do not have any label, i.e. need review

Notes from my conversation with [tjyrinki\\_suse](#):

- ~~We could make the sections collapsible and expandable~~ We should find a way to make reports readable out of the box, and rather have a separate report than trying to find "the" solution for everyone

agreed. I assume the current reports are already readable enough. We can beautify in the future.

- The **todo** could be a lot more useful
  - If we sort it alphabetically
  - If we can have flavor headers

yes, but I hope you agree that both are not a strong requirement for the current epic

- If we can include ticket links

That does not make sense because the todo-only-list shows entries that do not have a ticket yet

- For the purposes of this conversation, we probably want **a variant** of the existing todo list with new options configured in the YAML

what YAML? And which variant?

#### #49 - 2021-08-31 12:26 - tinita

okurz wrote:

cdywan wrote:

- For the purposes of this conversation, we probably want **a variant** of the existing todo list with new options configured in the YAML

what YAML? And which variant?

<https://gitlab.suse.de/openqa/openqa-review/-/blob/master/review-jobs.yml>

#### #50 - 2021-08-31 12:29 - tinita

Before adding more pages to the YAML config, we should probably look into [#97109](#) because the generation can already take up to 4 hours and, and still times out from time to time although we already set a high timeout value.

**#51 - 2021-08-31 12:56 - okurz**

Agreed, one more reason to not invest more effort in a solution for which we don't know how many users rely on it.

**#52 - 2021-08-31 13:24 - tjyrinki\_suse**

From my notes with cdywan, I believe a workable combination with lesser new things to implement would be to use complex per-squad overview query in the likes of this one <https://is.gd/ZGkmE2> I got from Michael a few weeks ago, together with an enhanced todo page. The former would hopefully give overall view to all failures within squads' plate (with caveat of not being able to filter non-interesting flavors under interesting job groups out), and the latter for giving the quickest view to failures that have no action done at all.

In this way, and as pointed out todo page actually hides failures that have a bug/ticket, out of the mockup we'd need just the visible job group and flavor headers and tests sorted in a similar way they are in overview (currently todo list order seems more random ie it hops from one flavor to another and then back). Would be nice to have also direct failing modules links + the Maintainer field from each failing module under the test suite that is failing.

As we're trying to get closer to 0 failures by wanting to unschedule things quicker if a fix is not immediately doable, these two pages even with eg 20 job groups for one squad should be tolerable, especially the todo-only one.

okurz wrote:

- The **todo** could be a lot more useful
  - If we sort it alphabetically
  - If we can have flavor headers

yes, but I hope you agree that both are not a strong requirement for the current epic

Currently I find the todo page is not very usable for the purpose of this ticket because the sorting and styling problems, even if there would be squad specific YAML already in place, so those two would be in my opinion the most critical ones, and then the direct link to module + maintainer less critical.

**#53 - 2021-08-31 13:52 - cdywan**

One thing I just realized I forgot to mention: I think a workshop about this, with the current reports and Timo's input as a basis, could help get more input from other stakeholders. So that we can avoid making this too specific while others are looking into other solutions of their own.

**#54 - 2021-08-31 19:54 - okurz**

tjyrinki\_suse wrote:

From my notes with cdywan, I believe a workable combination with lesser new things to implement would be to use complex per-squad overview query in the likes of this one <https://is.gd/ZGkmE2> I got from Michael a few weeks ago, together with an enhanced todo page. The former would hopefully give overall view to all failures within squads' plate (with caveat of not being able to filter non-interesting flavors under interesting job groups out), and the latter for giving the quickest view to failures that have no action done at all.

Yes, this is my expectation at best as well.

In this way, and as pointed out todo page actually hides failures that have a bug/ticket, out of the mockup we'd need just the visible job group and flavor headers and tests sorted in a similar way they are in overview (currently todo list order seems more random ie it hops from one flavor to another and then back). Would be nice to have also direct failing modules links

I still don't understand that part. The report overview page shows multiple reports and only one is the "todo-page". The intention of the "todo-page" is really to show only what needs review, all the other reports do show ticket references if they exist.

the Maintainer field from each failing module under the test suite that is failing.

This is a nice idea but very likely not something we will manage this year.

As we're trying to get closer to 0 failures by wanting to unschedule things quicker if a fix is not immediately doable, these two pages even with eg 20 job groups for one squad should be tolerable, especially the todo-only one.

Exactly, I think the same.

Currently I find the todo page is not very usable for the purpose of this ticket because the sorting and styling problems, even if there would be squad specific YAML already in place, so those two would be in my opinion the most critical ones

Ok, understood.

and then the direct link to module + maintainer less critical.

Same as above, that part I don't understand yet.

**#55 - 2021-09-02 07:38 - tjyrinki\_suse**

okurz wrote:

In this way, and as pointed out todo page actually hides failures that have a bug/ticket, out of the mockup we'd need just the visible job group and flavor headers and tests sorted in a similar way they are in overview (currently todo list order seems more random ie it hops from one flavor to another and then back). Would be nice to have also direct failing modules links  
I still don't understand that part. The report overview page shows multiple reports and only one is the "todo-page". The intention of the "todo-page" is really to show only what needs review, all the other reports do show ticket references if they exist.

With 'overview' I meant sorting similar way they are on "Test result overview" page like <https://openqa.suse.de/tests/overview?distri=sle&version=15-SP3&build=188.15&groupid=373> - that is, alphabetically, flavor by flavor instead of in more random order where one can't easily visually skip flavors that are not interesting.

**#56 - 2021-09-14 06:06 - tjyrinki\_suse**

Very nice to see improvements happening as there are now job group headers  
[https://openqa.io.suse.de/openqa-review/openqa\\_suse\\_de\\_todo\\_only.html](https://openqa.io.suse.de/openqa-review/openqa_suse_de_todo_only.html) !

**#57 - 2021-09-14 07:00 - okurz**

Thanks a lot! You are very quick to find this :)

**#58 - 2021-09-16 15:02 - okurz**

This epic was discussed among runger, vpelcak, hrommel1, okurz, tjyrinki\_suse and both okurz and runger suggested to not further wait and implement feature requests based on real-life feedback from users, i.e. QE engineers conducting daily openQA test review according to <https://confluence.suse.com/display/~vpelcak/Draft+-+Change+in+openQA+QE+Maintenance+Review> while diligently monitoring the adherence to process to ensure no increase in review delay. In other words: Keep maintsec happy by identifying and fixing false-positive test results quickly so that updates are approved where possible by our automatic components

**#59 - 2021-09-26 09:50 - okurz**

Hi, I am happy to report that all current subtasks have been resolved. In my opinion especially with [https://openqa.io.suse.de/openqa-review/openqa\\_suse\\_de\\_todo\\_only.html](https://openqa.io.suse.de/openqa-review/openqa_suse_de_todo_only.html) as well as the possibility to filter the openQA test overview page by test module content we have progressed a long way and can provide multiple alternatives how to present "single overview places" for squads hence calling the epic resolved.

A little bit more details about the test overview page of openQA:

It's possible to filter test overview pages by test module content with a search expression as a regex. This allows for example to filter any /tests/overview page with something like `module_re=Maintainer.*qe-core&modules_result=failed`, e.g. [https://openqa.opensuse.org/tests/overview?module\\_re=Maintainer.\\*qe-core&modules\\_result=failed&distri=microos&distri=opensuse&version=Tumbleweed&build=20210924&groupid=1#](https://openqa.opensuse.org/tests/overview?module_re=Maintainer.*qe-core&modules_result=failed&distri=microos&distri=opensuse&version=Tumbleweed&build=20210924&groupid=1#) or alike

Further improvements and follow-up tasks are defined in related epics

**#60 - 2021-09-26 09:50 - okurz**

- Status changed from Blocked to Resolved

**#61 - 2021-09-29 15:10 - vpelcak**

If this is resolved, can we see a demo of how to do this in our Friday Workshop?

**#62 - 2021-09-29 15:12 - okurz**

Absolutely!

**Files**

---

|                           |         |            |               |
|---------------------------|---------|------------|---------------|
| review-report-mockup.png  | 52.8 KB | 2021-08-18 | tjyrinki_suse |
| review-report-mockup2.png | 55.6 KB | 2021-08-25 | tjyrinki_suse |