

## QA - action #93799

### teregen: Improvement of usability of disabled testcases notification

2021-06-10 10:23 - vpelcak

|   |        |                        |            |
|---|--------|------------------------|------------|
| <b>Status:</b>  | New    | <b>Start date:</b>     | 2021-06-10 |
| <b>Priority:</b>  | Low    | <b>Due date:</b>       |            |
| <b>Assignee:</b>  |        | <b>% Done:</b>         | 0%         |
| <b>Category:</b>  |        | <b>Estimated time:</b> | 0.00 hour  |
| <b>Target version:</b>  | future |                        |            |
| <b>Description</b>  |        |                        |            |
| <b>User story</b>   |        |                        |            |
| <p>There is a new recent feature in template generator.<br/>If the automation squad disables the testcase in aggregated tests (for example in order to fix it), this is automatically detected for the testers to identify missing testcase and cover the missing tests by manual testing.</p> <p>As a tester, I then occasionally encounter stuff like:</p> <p>Example:</p> <p><a href="http://qam.suse.de/testreports/SUSE:Maintenance:19738:241935/log">http://qam.suse.de/testreports/SUSE:Maintenance:19738:241935/log</a></p>   |        |                        |            |
| <b>regression tests:</b>  |        |                        |            |
| <p>WARNING: One or more test suites are missing compared to last runs with these source rpms.<br/>Please, check openQA manually. Missing test suites are:</p> <ul style="list-style-type: none"><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-sles4sap_hana_node01 / x86_64<br/>sle-12-SP3-Server-DVD-SAP-Incidents-x86_64-Build:19600:libxml2-qam-sles4sap_hana_node01@64bit-sap-qam</li><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-create_hdd_sles4sap_gnome / x86_64<br/>sle-12-SP3-Server-DVD-SAP-Incidents-x86_64-Build:19600:libxml2-qam-create_hdd_sles4sap_gnome@64bit</li><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-sles4sap_scc_gnome_hana_cli / x86_64<br/>sle-12-SP3-Server-DVD-SAP-Incidents-x86_64-Build:19600:libxml2-qam-sles4sap_scc_gnome_hana_cli@64bit-sap-qam</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; qam-incidentinstall / ppc64le<br/>sle-12-SP3-Server-DVD-Incidents-Install-ppc64le-Build:19600:libxml2-qam-incidentinstall@ppc64le</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; mau-extratests2 / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-mau-extratests2@64bit</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; qam-incidentinstall / s390x<br/>sle-12-SP3-Server-DVD-Incidents-Install-s390x-Build:19600:libxml2-qam-incidentinstall@zskvm</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; mau-extratests-zypper / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-mau-extratests-zypper@64bit</li><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-incidentinstall-sap / x86_64<br/>sle-12-SP3-Server-DVD-SAP-Incidents-Install-x86_64-Build:19600:libxml2-qam-incidentinstall-sap@64bit</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; mau-extratests-phub / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-mau-extratests-phub@64bit</li><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-sles4sap_scc_gnome_netweaver / x86_64<br/>sle-12-SP3-Server-DVD-SAP-Incidents-x86_64-Build:19600:libxml2-qam-sles4sap_scc_gnome_netweaver@64bit</li><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-sles4sap_hana_node02 / x86_64<br/>sle-12-SP3-Server-DVD-SAP-Incidents-x86_64-Build:19600:libxml2-qam-sles4sap_hana_node02@64bit-sap-qam</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; qam-allpatterns+addons / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-qam-allpatterns+addons@64bit</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; mau-webserver / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-mau-webserver@64bit</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; qam-incidentinstall / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-Install-x86_64-Build:19600:libxml2-qam-incidentinstall@64bit</li><li>• Maintenance: SLE 12 SP3 SAP Incidents -&gt; qam-incidentinstall-sap / ppc64le<br/>sle-12-SP3-Server-DVD-SAP-Incidents-Install-ppc64le-Build:19600:libxml2-qam-incidentinstall-sap@ppc64le</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; mau-filesystem / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-mau-filesystem@64bit</li><li>• Maintenance: SLE 12 SP3 Incidents -&gt; mau-extratests1 / x86_64<br/>sle-12-SP3-Server-DVD-Incidents-x86_64-Build:19600:libxml2-mau-extratests1@64bit</li></ul> |        |                        |            |

- Maintenance: SLE 12 SP3 Incidents -> mau-sles-robot-fw / x86\_64  
sle-12-SP3-Server-DVD-Incidents-x86\_64-Build:19600:libxml2-mau-sles-robot-fw@64bit-2gbram
- Maintenance: SLE 12 SP3 SAP Incidents -> qam-sles4sap\_scc\_gnome\_saptune / x86\_64  
sle-12-SP3-Server-DVD-SAP-Incidents-x86\_64-Build:19600:libxml2-qam-sles4sap\_scc\_gnome\_saptune@64bit
- Maintenance: SLE 12 SP3 Incidents -> mau-extratests-kdump / x86\_64  
sle-12-SP3-Server-DVD-Incidents-x86\_64-Build:19600:libxml2-mau-extratests-kdump@64bit
- Maintenance: SLE 12 SP3 SAP Incidents -> qam-sles4sap\_hana\_supportserver / x86\_64  
sle-12-SP3-Server-DVD-SAP-Incidents-x86\_64-Build:19600:libxml2-qam-sles4sap\_hana\_supportserver@64bit
- Maintenance: SLE 12 SP3 Incidents -> mru-install-minimal-with-addons / x86\_64  
sle-12-SP3-Server-DVD-Incidents-x86\_64-Build:19600:libxml2-mru-install-minimal-with-addons@64bit

in the testreport.

Testers do not understand it and don't know how to interpret it.

## Acceptance criteria

- **AC1:** it is clear to the tester what he needs to do when he sees message like the one above
- **AC2:** ideally, information about what testing needs to be done to cover for the disabled testcase should be easily accessible

I gave this high priority, because there is a danger, that additional testing won't happen and updates are released with reduced test coverage with risk of regressions.

### Related issues:

|  |                 |                   |
|--|-----------------|-------------------|
| Related to QA - action #90914: [teregen] Add overview for stored coverage data   | <b>New</b>      | <b>2021-04-09</b> |
| Related to QA - action #90401: [teregen] Integrate coverage information in a ... | <b>Resolved</b> | <b>2021-03-29</b> |
| Related to QA - action #88127: [tools][qem] Test coverage DB for maintenance ... | <b>Closed</b>   | <b>2021-02-08</b> |

## History

### #1 - 2021-06-10 11:10 - jbaier\_cz

- Related to action #90914: [teregen] Add overview for stored coverage data added

### #2 - 2021-06-10 11:11 - jbaier\_cz

- Assignee deleted (jbaier\_cz)

Unassigning myself as there is nothing I can do at this moment, this needs more input from the user-base first.

### #3 - 2021-06-10 12:31 - cdywan

Notes:

- By definition the tests listed here are no longer there
- It might be useful to lookup previous jobs for these cases
- A diff of the YAML (if the YAML was changed) and GitLab diff if applicable

Suggested approach:

- Include descriptions of tests that previously passed but were removed

### #4 - 2021-06-10 12:38 - okurz

- Due date set to 2021-07-01

- Assignee set to okurz

- Target version set to Ready

To be able to better understand and prioritize, could you try to adapt the description of the ticket according to the template <https://progress.opensuse.org/projects/openqav3/wiki/#Feature-requests> ? Also, can you explain why you see this as "High" priority?

### #5 - 2021-06-10 14:11 - vpelcak

- Description updated

### #6 - 2021-06-10 18:21 - jbaier\_cz

- Related to action #90401: [teregen] Integrate coverage information in a presentable way into test template added

#7 - 2021-06-10 18:22 - jbaier\_cz

- Related to action #88127: [tools][qem] Test coverage DB for maintenance updates added

#8 - 2021-06-10 22:09 - okurz

vpelcak wrote:

## Acceptance criteria

- **AC1:** it is clear to the tester what he needs to do when he sees message like the one above

This sounds like extending the message "Please, check openQA manually". You would only expect better instructions within that text, right? Or elsewhere as well?

- **AC2:** ideally, information about what testing needs to be done to cover for the disabled testcase should be easily accessible

That implies that manual testing would be able to do what an automated openQA test scenario based on a complete test suite does. It would be possible for removed test modules, e.g. just the "firefox" test module, but AFAIK that is not handled at all by test report templates. But I see that as effectively impossible when complete test scenarios are unscheduled. I know that this comes up from time to time that people think they can do manually for single updates what an automated test in openQA can do but only a very tiny subset of tests would be possible to execute manually nowadays within a reasonable time.

I am just picking *one* of the above mentioned test suites, possibly the *smallest* one, "mau-extratests-zypper". Take for example <https://openqa.suse.de/tests/6224114> showing one successful run of such scenario. The test is composed of the test modules. Let's imagine we could ask a human tester to follow all instructions from the test modules. That is effectively following all instructions that show up also in the log file <https://openqa.suse.de/tests/6224114/file/autoinst-log.txt>. One could search for all "<<<" markers which are commands that openQA types. But that accounts for 2300 lines in the mentioned example, unlikely to be followed by humans. Instead let's look at modules themselves. First "boot\_to\_desktop", ok, feasible to achieve by a tester just judging from the name. Then comes "zypper\_lr\_validate". In openQA this takes 8s. The instructions are visible in [https://openqa.suse.de/tests/6224114/modules/zypper\\_lr\\_validate/steps/1/src](https://openqa.suse.de/tests/6224114/modules/zypper_lr_validate/steps/1/src) corresponding to [https://github.com/os-autoinst/os-autoinst-distrib-opensuse/blob/master/tests/console/zypper\\_lr\\_validate.pm](https://github.com/os-autoinst/os-autoinst-distrib-opensuse/blob/master/tests/console/zypper_lr_validate.pm). Very hard to read as the test module needs to be very special depending on SLE service packs. But a test for an incident update likely needs to understand these specifics as well. And then there are many more test modules, like "zypper\_ref", "zypper\_info", "check\_interactive\_flag", "validate\_packages\_and\_patterns", "zypper\_extend", "coredump\_collect". Again, asking humans to redo what an automated test system does is impossible.

The best approach I see is:

1. If tests produce false positives and it wasn't a mistake that the tests have been added, e.g. someone just added new tests that were never there, then revert what introduces the test regression and retrigger tests for the incident updates
2. If in the rare exception that one really really really thinks it's the right approach to revert to manual testing, e.g. urgent security fix while all openQA tests are broken due to infrastructure problem, then rely on expertise of professional testers of how the update can be tested best as we can not formalize consistent and complete manual test instructions that could replace the automated tests

I gave this high priority, because there is a danger, that additional testing won't happen and updates are released with reduced test coverage with risk of regressions.

ok, but it's not a recent regression, is it? So basically as a safeguard or workaround until we find a better solution we/you should remind or explain manually what it means.

#9 - 2021-06-11 08:12 - vpelcak

okurz wrote:

vpelcak wrote:

## Acceptance criteria

- **AC1:** it is clear to the tester what he needs to do when he sees message like the one above

This sounds like extending the message "Please, check openQA manually". You would only expect better instructions within that text, right? Or elsewhere as well?

Yes. Having better instructions what needs to be done in general would help. The block of the text below, while great start, is hard to decode for the people.

- **AC2:** ideally, information about what testing needs to be done to cover for the disabled testcase should be easily accessible

That implies that manual testing would be able to do what an automated openQA test scenario based on a complete test suite does. It would be possible for removed test modules, e.g. just the "firefox" test module, but AFAIK that is not handled at all by test report templates. But I see that

as effectively impossible when complete test scenarios are unscheduled. I know that this comes up from time to time that people think they can do manually for single updates what an automated test in openQA can do but only a very tiny subset of tests would be possible to execute manually nowadays within a reasonable time.

I see your point. OTOH, we used to work in that kind of mode already. And I certainly hope that this situation won't be too frequent.

The best approach I see is:

1. If tests produce false positives and it wasn't a mistake that the tests have been added, e.g. someone just added new tests that were never there, then revert what introduces the test regression and retrigger tests for the incident updates
2. If in the rare exception that one really really really thinks it's the right approach to revert to manual testing, e.g. urgent security fix while all openQA tests are broken due to infrastructure problem, then rely on expertise of professional testers of how the update can be tested best as we can not formalize consistent and complete manual test instructions that could replace the automated tests

You listed just few of possible options. There could be a version bump causing testcase not working, long time unstable test...

I gave this high priority, because there is a danger, that additional testing won't happen and updates are released with reduced test coverage with risk of regressions.

ok, but it's not a recent regression, is it? So basically as a safeguard or workaround until we find a better solution we/you should remind or explain manually what it means.

No, it is not a regression indeed.

#### #10 - 2021-06-11 11:36 - okurz

The best approach I see is:

1. If tests produce false positives and it wasn't a mistake that the tests have been added, e.g. someone just added new tests that were never there, then revert what introduces the test regression and retrigger tests for the incident updates
2. If in the rare exception that one really really really thinks it's the right approach to revert to manual testing, e.g. urgent security fix while all openQA tests are broken due to infrastructure problem, then rely on expertise of professional testers of how the update can be tested best as we can not formalize consistent and complete manual test instructions that could replace the automated tests

You listed just few of possible options. There could be a version bump causing testcase not working, long time unstable test...

That makes no difference to me as *still* a human tester will not be possible to conduct the test in depth as the automated tests are able to do. What I have not mentioned as option is of course what should have been decided in before: That such version update can only be approved as soon as the automated tests have been updated with exception of what I described as option 2 which should not apply for a "version bump causing testcase not working". The risk of releasing a product regression breaking customer use cases is just too great.

As the list of missing tests can get long I would suggest to abbreviate after e.g. 3 lines with ... and point to a test overview page on openQA instead, e.g. <https://openqa.suse.de/tests/overview?distri=sle&build=:19924:webkit2gtk3> but as we can not store openQA jobs indefinitely this has a limitation. All openQA results are deleted unless marked as "important". Rather than "saving the data elsewhere because we don't have enough space in openQA" I would say if we need results for longer in openQA then we will find a solution in openQA. So as alternative to saving a list of tests elsewhere we can discuss keeping results longer for certain job groups or we can mark builds or jobs as "important" and then they are kept around longer. This needs corresponding space on OSD reserved for that purpose of course.

#### #11 - 2021-06-18 12:14 - okurz

- Subject changed from *Improvement of usability of disabled testcases notification* to *teregen: Improvement of usability of disabled testcases notification*

- Assignee deleted (okurz)

- Priority changed from *High* to *Low*

- Target version changed from *Ready* to *future*

I agree that the mentioned ideas would help but currently we need to focus on other needed changes before we can go back to UX improvements in SUSE-internal-only tools.

#### #12 - 2021-07-07 07:51 - okurz

- Due date deleted (2021-07-01)

points from [#93799#note-4](#) have been addressed hence removing due-date

**#14 - 2021-09-16 09:12 - vpeicak**

Hello.

How is the situation about this?

I feel that the priority is way too low.

When I said that it is not a regression, I think I was not correct.

Yes, it is not a regression in the matter of notifications to the people, but every unscheduled test poses a risk of regression in the test coverage and subsequently in the product.

That needs to be addressed.

OTOH some unschedules could be for a good reason (useless test for example). Perhaps it is possible to somehow address that, too.

Maybe we should refine the ticket.