

Get something new everyday

- Staging Project for openSUSE

Max Lin

Software Engineer

m_lin@suse.com

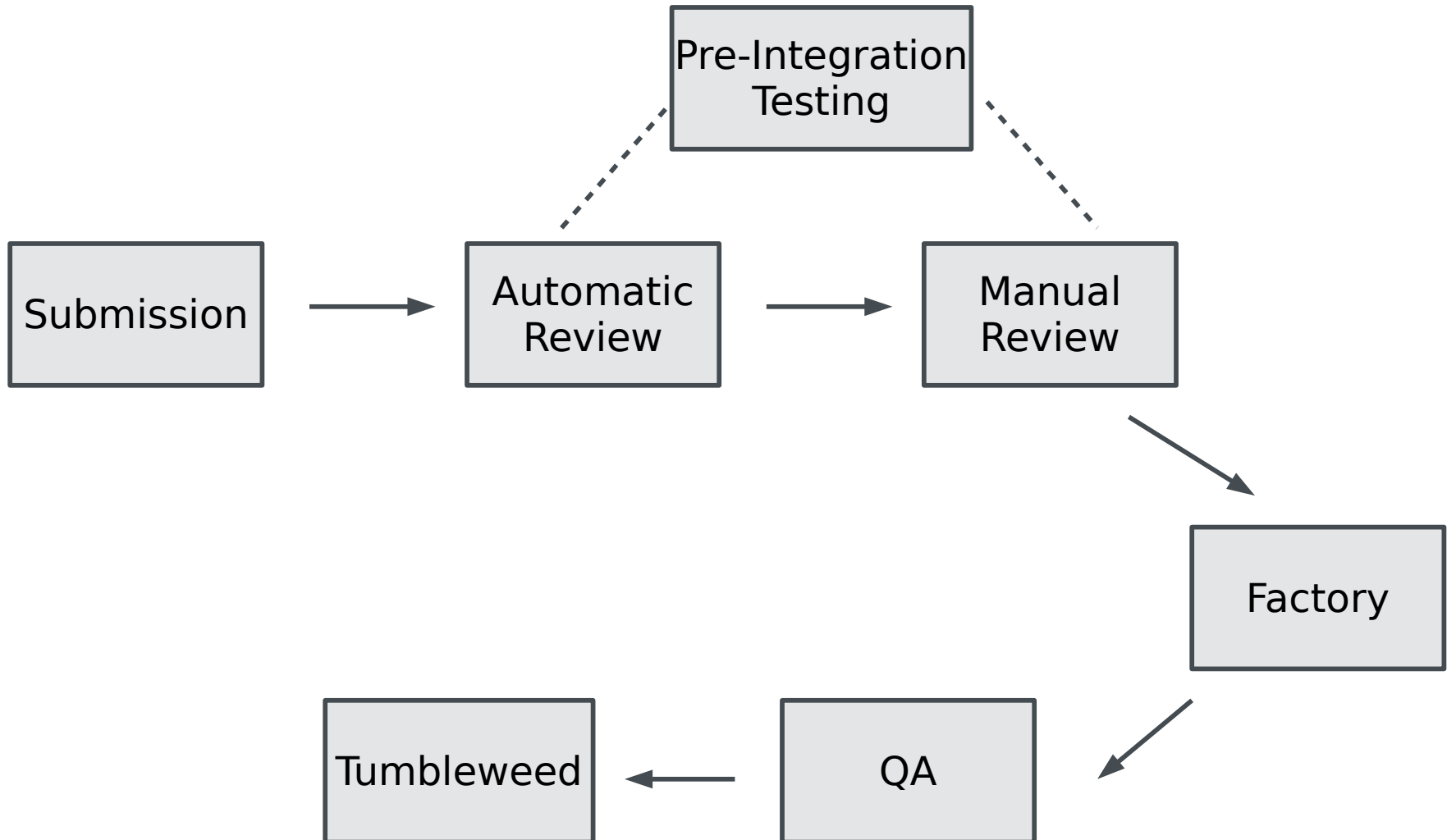


Agenda

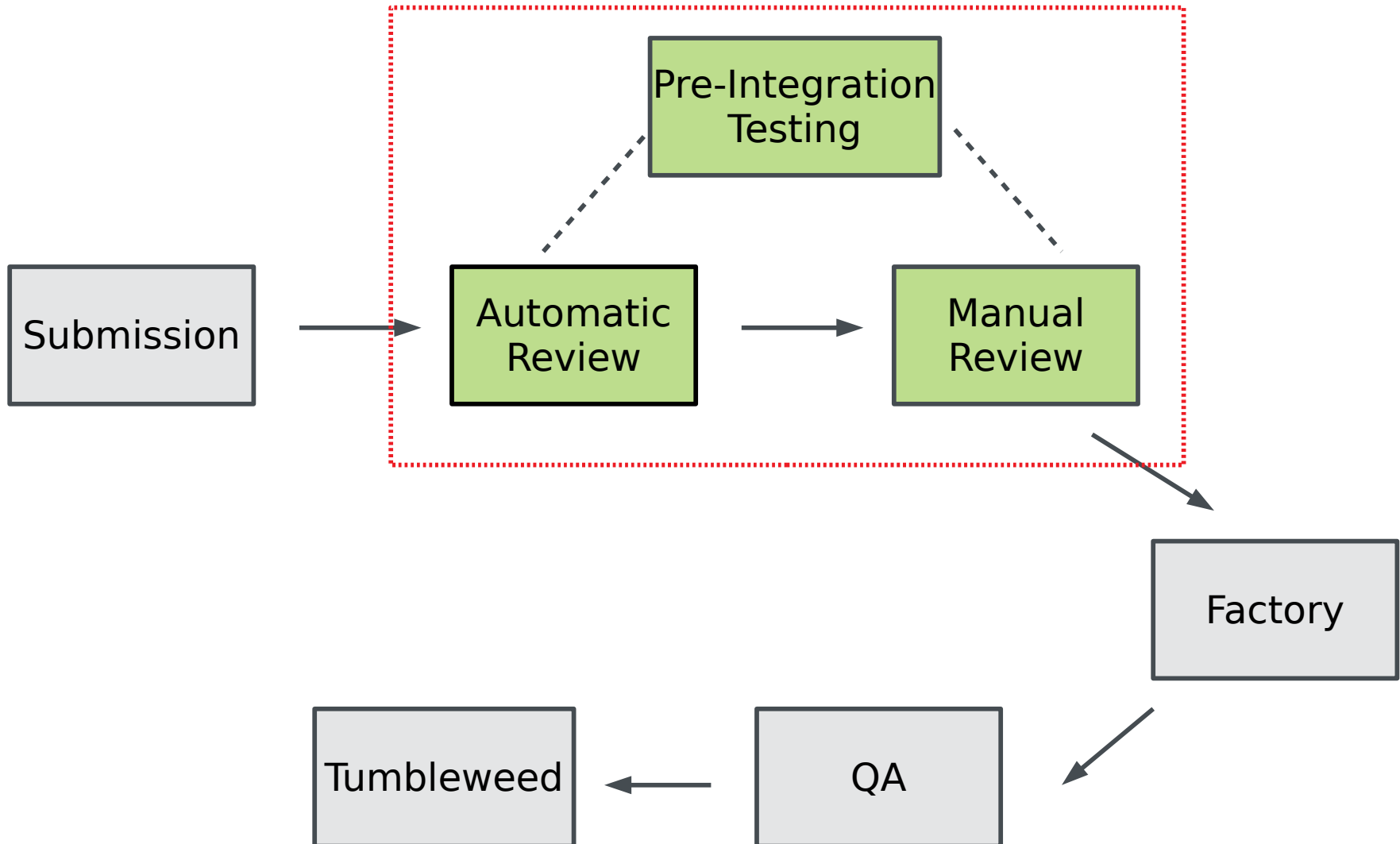
- Basic workflow
- Staging Project setup
- Dashboard
- Tools
- Case study

Basic Workflow

Factory development model



Factory development model



The review process

- Factory-Auto: checks basic rules and adds additional reviewers
- Legal-Auto: checking if the license of the package is in the permitted license database
- Review Team: human check of the request
- Repo-Checker: a more in depth automatic check
- Factory-Staging: review by Staging Master (manually and assisted by repochecker and openQA)

Pre-Integration testing: RepoChecker

- Build result verification of submission
- The binaries installable
- Check the conflicts with other packages
- Check the build cycle in Factory
- Check the validation of *deletereq* request
- Leave the comment on the request once found an issue
- `osc check_repo` command

Pre-Integration testing: openQA

Currently the test plan covering...

- Normal installation
- Installation on RAID partition(software RAID)
- Installation with cryptlvm enabled
- Installation on UEFI environment
- Boot succeeded after the system installed by above way
- MinimalX/GNOME/KDE environment work
- Update system to current staging

The adi Staging (Ad-Interim)

- Mainly for *non-rings* packages submission
- Ensure the package can built against Factory
- RepoChecker is involved
- It can not handle *deletereq* request!

Staging Project Setup

Rings

	Description	Packages
Ring-0	The bootstrap cycle of factory	102
Ring-1	Based on bootstrap and building minimalX DVD	1044
Ring-2	A pretty much complete DVD	937

Staging Project project setup

```
<link project="openSUSE:Factory:Rings:0-Bootstrap"/> # depends  
<link project="openSUSE:Factory:Rings:1-MinimalX"/>
```

```
<repository name="standard" rebuild="direct" linkedbuild="all">  
  <path project="openSUSE:Factory:Staging:A"  
repository="bootstrap_copy"/>  
  ...  
</repository>  
<repository name="images" rebuild="direct" linkedbuild="all">  
  <path project="openSUSE:Factory:Staging:A"  
repository="standard"/>  
  ...  
</repository>  
<repository name="bootstrap_copy">  
  <path project="openSUSE:Factory:Staging" repository="standard"/>  
  ...  
</repository>
```

Staging Project project setup

```
<link project="openSUSE:Factory:Rings:2-TestDVD"/>
```

```
<repository name="standard" rebuild="direct" linkedbuild="all">  
  <path project="openSUSE:Factory:Staging:A"  
repository="standard"/>  
  ...  
</repository>  
<repository name="images" rebuild="direct" linkedbuild="all">  
  <path project="openSUSE:Factory:Staging:A:DVD"  
repository="standard"/>  
  ...  
</repository>
```

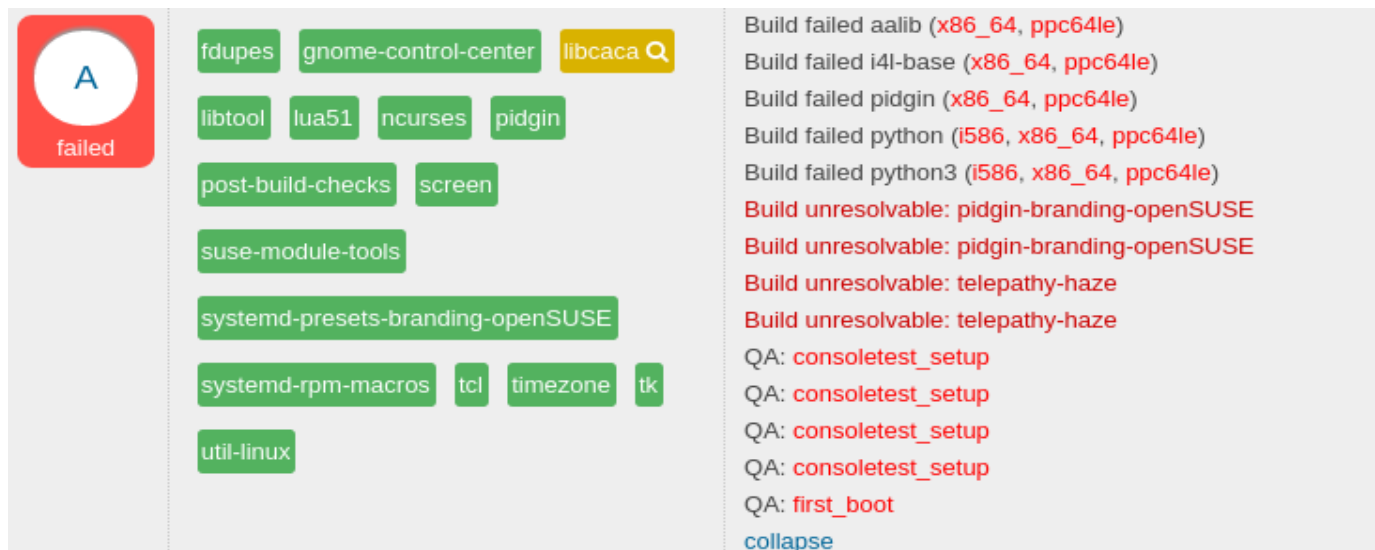
Support packages

- Test-DVD-ppc64le: puts a .kiwi file, in order to generating the relevant ISO file.
- Test-DVD-x86_64: puts a .kiwi file, in order to generating the relevant ISO file.
- bootstrap-copy: aggregate the bootstrap packages from Ring-0. not in :DVD.

Dashboard

Adding request to staging



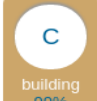
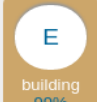
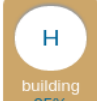
- osc staging select LETTER REQUESTS



The screenshot displays the OBS staging interface. On the left, a red square icon with a white circle containing the letter 'A' and the word 'failed' below it indicates the build status. To the right of this icon, a list of packages is shown in green boxes: fdupes, gnome-control-center, libccata (with a magnifying glass icon), libtool, lua51, ncurses, pidgin, post-build-checks, screen, suse-module-tools, systemd-presets-branding-openSUSE, systemd-rpm-macros, tcl, timezone, tk, and util-linux. On the far right, a list of build failures is shown in red text: 'Build failed aalib (x86_64, ppc64le)', 'Build failed i4l-base (x86_64, ppc64le)', 'Build failed pidgin (x86_64, ppc64le)', 'Build failed python (i586, x86_64, ppc64le)', 'Build failed python3 (i586, x86_64, ppc64le)', 'Build unresolvable: pidgin-branding-openSUSE', 'Build unresolvable: pidgin-branding-openSUSE', 'Build unresolvable: telepathy-haze', 'Build unresolvable: telepathy-haze', 'QA: consoletest_setup', 'QA: consoletest_setup', 'QA: consoletest_setup', 'QA: consoletest_setup', 'QA: first_boot', and 'collapse' in blue text.

Staging Dashboard

openSUSE Build Service > Projects > openSUSE:Factory > Staging projects mlin7442 (223 tasks) | Home Project | Logout

Project	Requests	Problems
 B acceptable	autoconf automake bison build-compare coreutils cracklib gettext-runtime intltool sed	✓
 G testing 88%	doxygen libdatrie tevent	✓
 C building 99%	PackageKit autotast2 couchdb dmidecode glew harfbuzz libevdev libinput linuxrc mariadb net-snmp obs-service-format_spec_file perl-XML-LibXML python-xml syslinux ...3 more	✓
 E building 99%	NetworkManager NetworkManager-gnome NetworkManager-pptp NetworkManager-vpnc apache2-mod_auth_ntlm_winbind btrfsprogs gnu-efi http-parser kfilemetadata5 libEMF libgit2 openjpeg2 php5 xterm	✓
 H building 95%	kernel-source	Build failed installation-images-openSUSE (i586, x86_64)

Legend

Requests

- Ready
- In review
- Obsolete
- Untracked

Projects

- Building
- Testing
- Review
- Acceptable
- Unacceptable
- Failed

Reviews

- Source Review
- Installation Check
- Legal Review

Infos

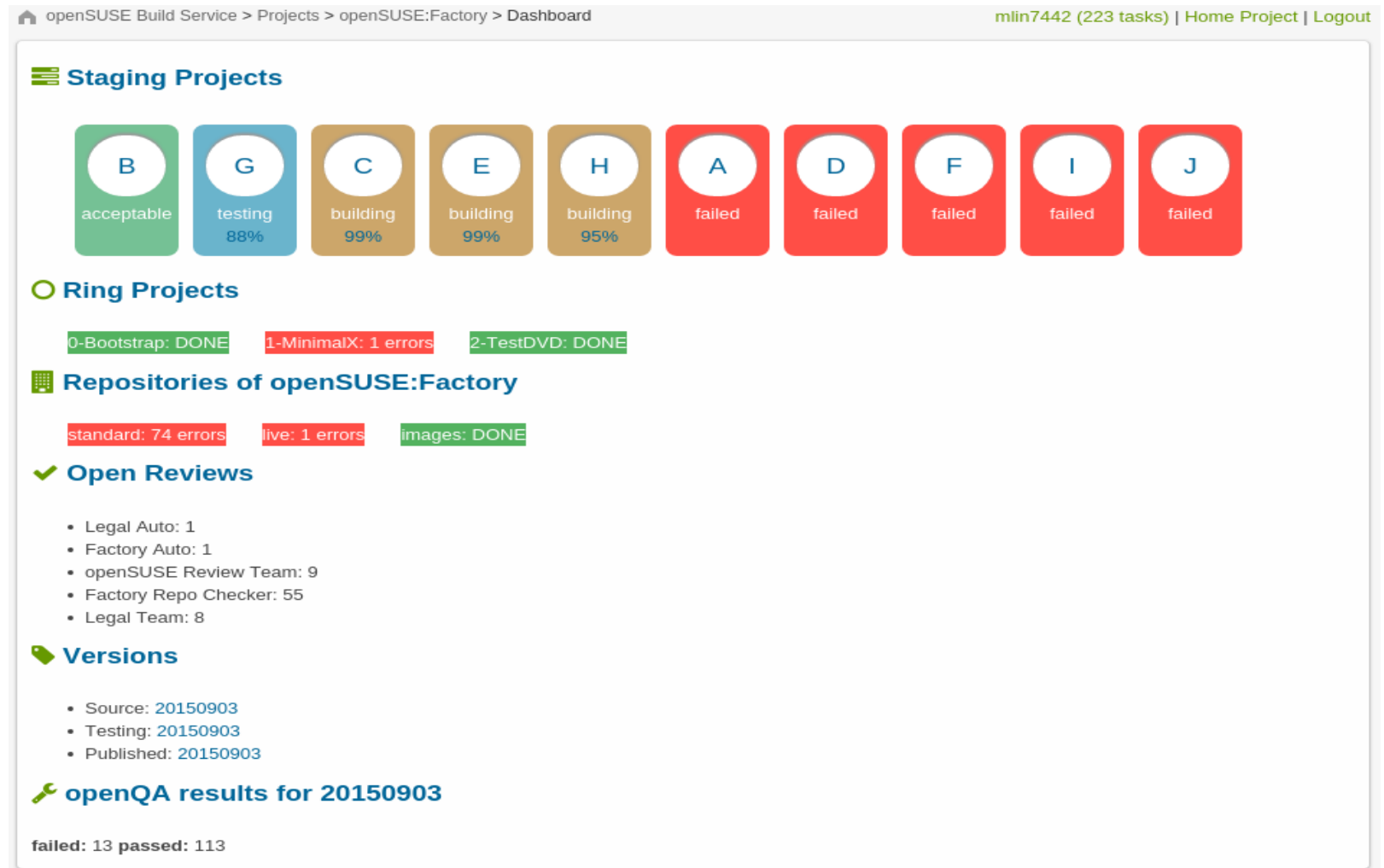
Empty Projects:

none

Backlog:

- Mesa
- audit
- bash
- cdrtools
- colord
- eigen3
- ... 27 more

Factory Dashboard



Tools

osc staging command

- accept: accept all requests in Staging:<LETTER>
- check: shows all staging projects status
- cleanup_rings: try to cleanup rings content and print out problems
- freeze: rebase the sources of the project's links
- list: dump the requests what open review by factory-staging
- select: add requests to the project
- unselect: remove requests from the project ie. pushing them back to the backlog

<https://github.com/openSUSE/osc-plugin-factory>

osc check_repo command

- Behind of RepoChecker
- Download the needed binaries
- Execute *installcheck* against binaries
- Execute *findfileconflicts* against binaries
- Build result in devel project verification
- Check new cycle created
- Can force skip review by --skip argument
- Can force skip cycle check by --skipcycle argument

ISO file generating

- Mirrored the binaries from OBS project
- Execute rpm2solv against all binaries and generate the solv file
- There are 2 testcase files as template represents the basic requirement of packages
- Adds the declare statement of solv file to the correspond testcase
- Execute testsolv against testcase then get the packages list(or detect the problem)
- Parsing the packages list above and add to kiwi file
- Update the kiwi file back to Test-DVD-{\$arch} on OBS

openQA




Testing on openQA

Test result overview

Overall Summary of [Staging Projects](#) build 209.2

Passed: **3** Failed: **1**

Flavor: Staging-DVD





Test	x86_64
RAID1	
cryptlvm	 boot_encrypt
minimalx	
rescue_system	

Test result overview

Overall Summary of [Staging Projects](#) build 387.2

Passed: **1** Soft Failure: **1** Failed: **2**

Flavor: Staging2-DVD

Test	x86_64
gnome	 first_boot
kde	 amarok firefox_audio 6 more
miniuefi	
update_staging	

Check the status

```
m1in@innoko:~> osc staging check A
-- FAILED Project openSUSE:Factory:Staging:A still needs attention
- libcacca: Missing reviews: opensuse-review-team
- Following packages are broken:
  python (standard): failed
  python3 (standard): failed
  python (standard): failed
  python3 (standard): failed
  python (standard): failed
  python3 (standard): failed
- openQA's overall status is failed for https://openqa.opensuse.org/tests/82027

-- For subproject openSUSE:Factory:Staging:A:DVD
- Following packages are broken:
  aalib (standard): failed
  i4l-base (standard): failed
  pidgin (standard): failed
  pidgin-branding-openSUSE (standard): unresolvable
  telepathy-haze (standard): unresolvable
  aalib (standard): failed
  i4l-base (standard): failed
  pidgin (standard): failed
  pidgin-branding-openSUSE (standard): unresolvable
  telepathy-haze (standard): unresolvable
- openQA's overall status is failed for https://openqa.opensuse.org/tests/82179
m1in@innoko:~> osc staging check B
++ Acceptable staging project openSUSE:Factory:Staging:B
```

Case Study

Successful Cases

- New cycle added
- File conflicts
- Build failures
- Missed requirement
- Feature is broken, caught by openQA
- etc.

Failure Case 1

- Old binaries tainted the other packages build, bad packages didn't re-triggered build on OBS somehow. See the failure after merge to Factory.
- Example1: mdadm with new systemd
- Example2: kdebase4-runtime with new NM

Failure Case 2

- Staging Project didn't test Live ISO boot, and even didn't generate the LiveISO there.
- Example: the changes in wodim caused TW snapshot can not boot on USB stick.
- Bug:
https://bugzilla.opensuse.org/show_bug.cgi?id=939456

Failure Case 3

- The feature test doesn't covered by openQA
- Example: yast2 software manager crashed in UI mode
- Bug:
https://bugzilla.novell.com/show_bug.cgi?id=941398

Failure Case 4

- Dependency submission didn't put in the same staging, after merge both to Factory, the issue appear.
- Example: PkgA buildrequire PkgB, and both have new changes and submit to Factory almost at the same, and they are put in different staging, unluckily both are build succeeded in its staging and nothing red in each staging, after merge them to Factory, PkgA fails to build.
- Rarely happens!

Q&A

Thank you!





Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

+49 911 740 53 0 (Worldwide)
www.suse.com

Join us on:
www.opensuse.org

Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

