Get something new everyday

- Staging Project for openSUSE

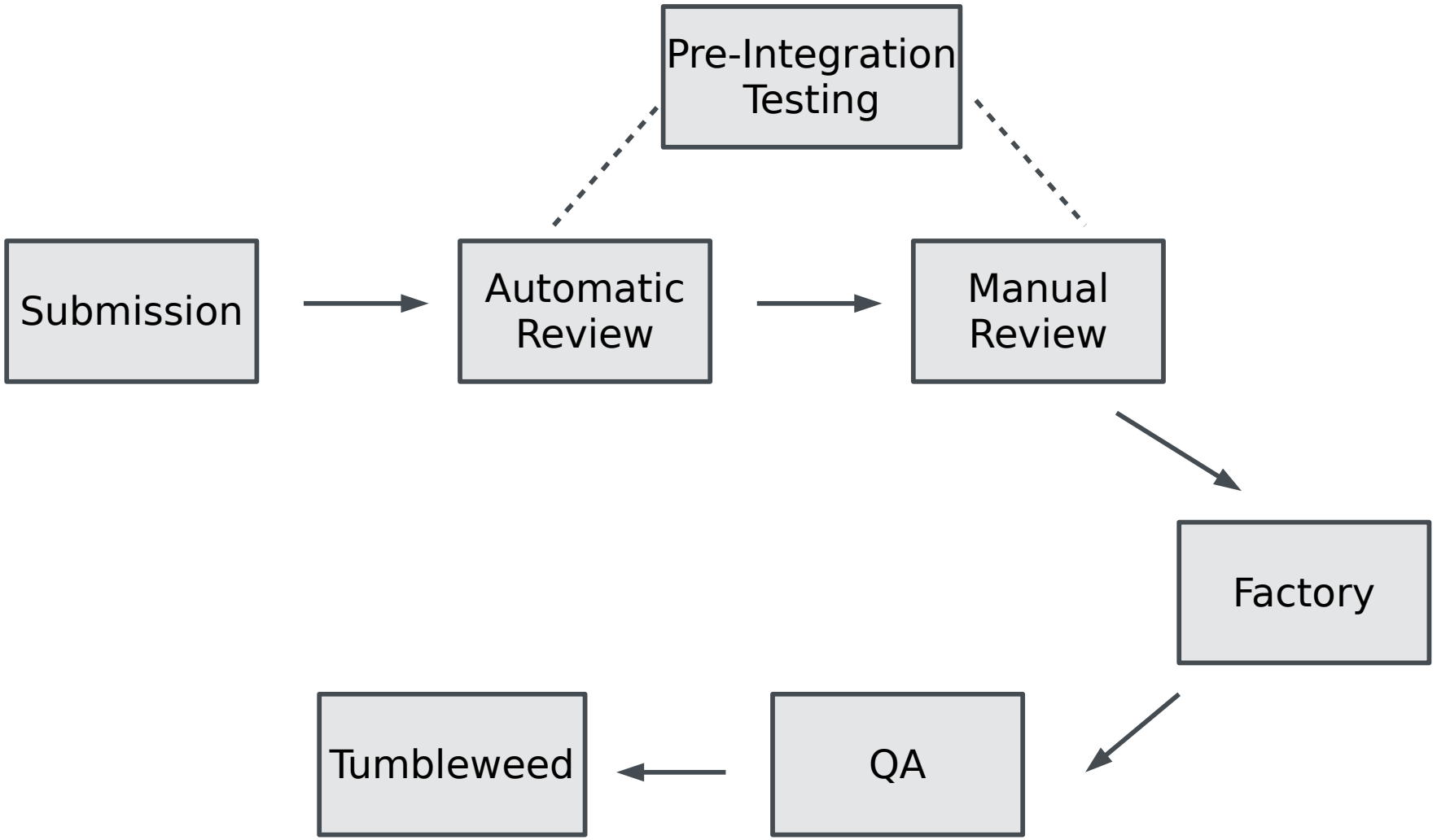
Max Lin
Software Engineer
mlin@suse.com

Agenda

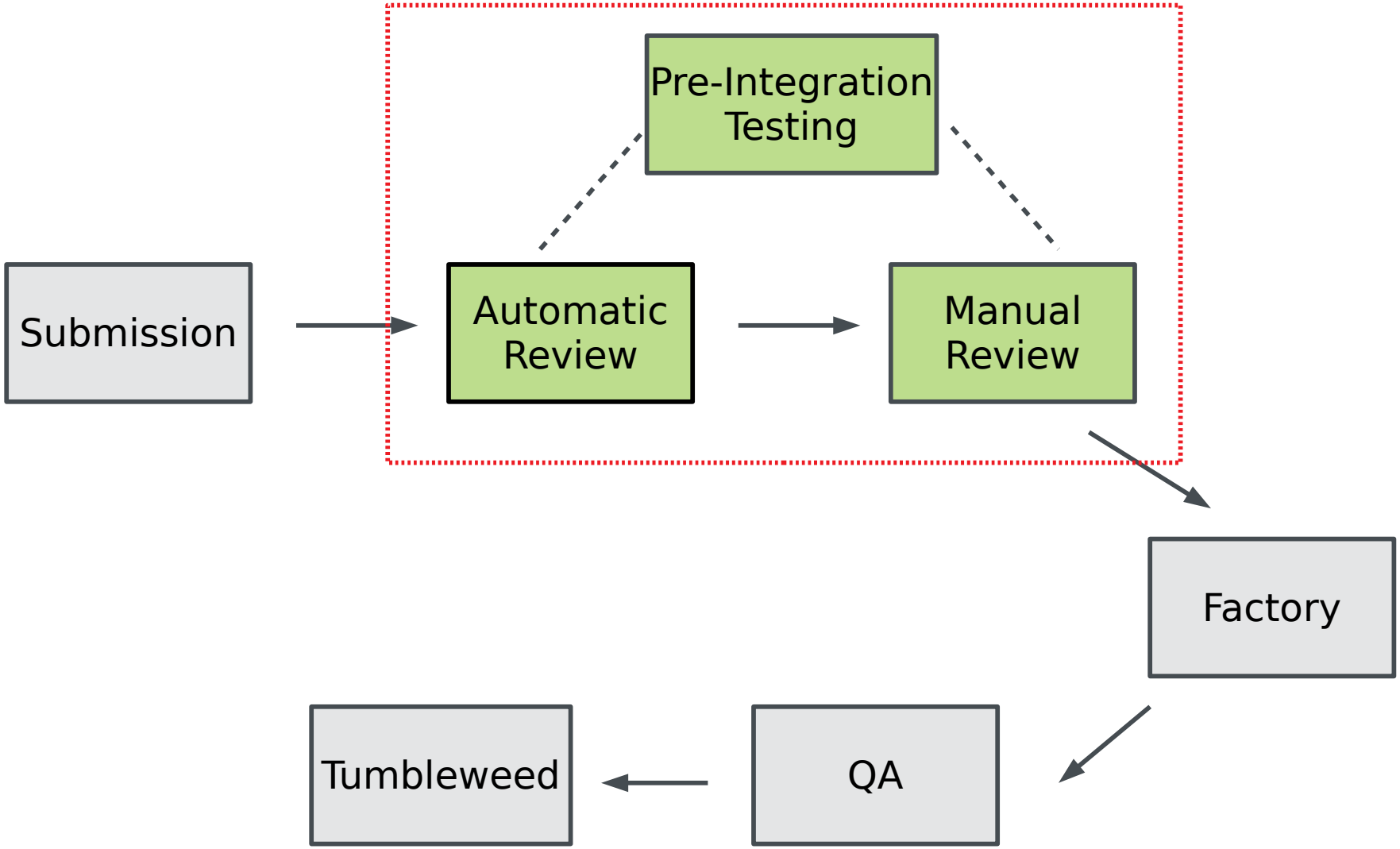
- Basic workflow
- Staging Project setup
- Dashboard
- Tools
- Case study

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

Staging Project Setup

Rings

	Description	Packages
Ring-0	The bootstrap cycle of factory	102
Ring-1	Based on bootstrap and building minimalX DVD	1043
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>
```

Stagings

- openSUSE:Factory:Staging:A
- openSUSE:Factory:Staging:A:DVD
- openSUSE:Factory:Staging:B
- openSUSE:Factory:Staging:B:DVD
- openSUSE:Factory:Staging:C
- openSUSE:Factory:Staging:C:DVD
- …
- openSUSE:Factory:Staging:J
- openSUSE:Factory:Staging:J:DVD

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.

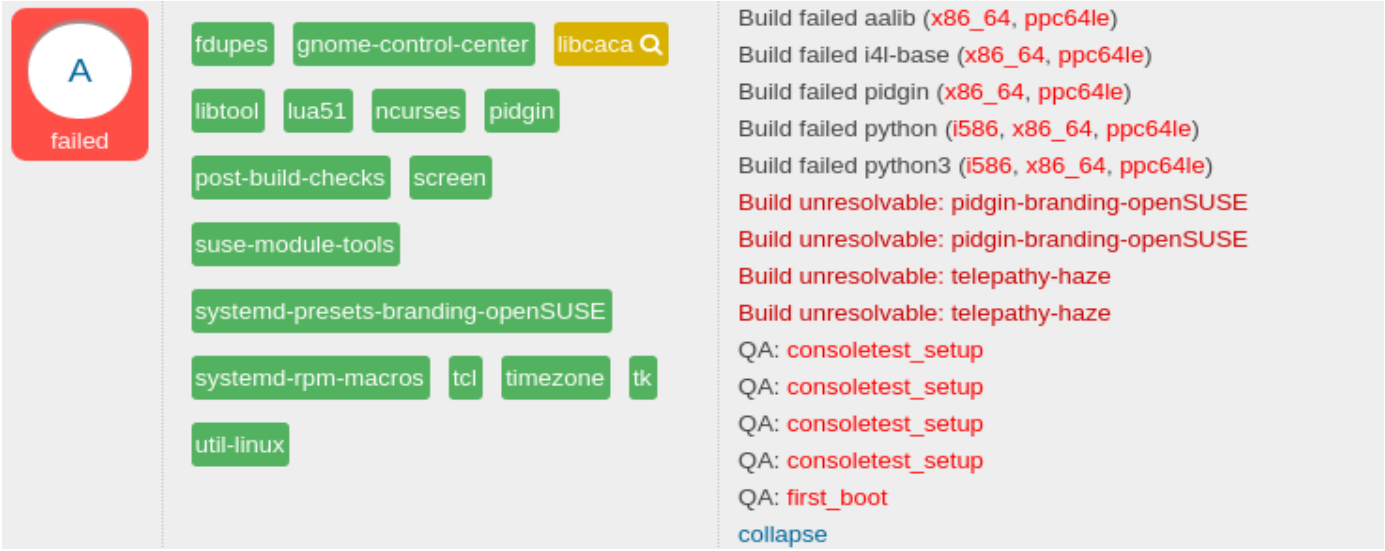
The adi Staging (Ad-Interim)

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

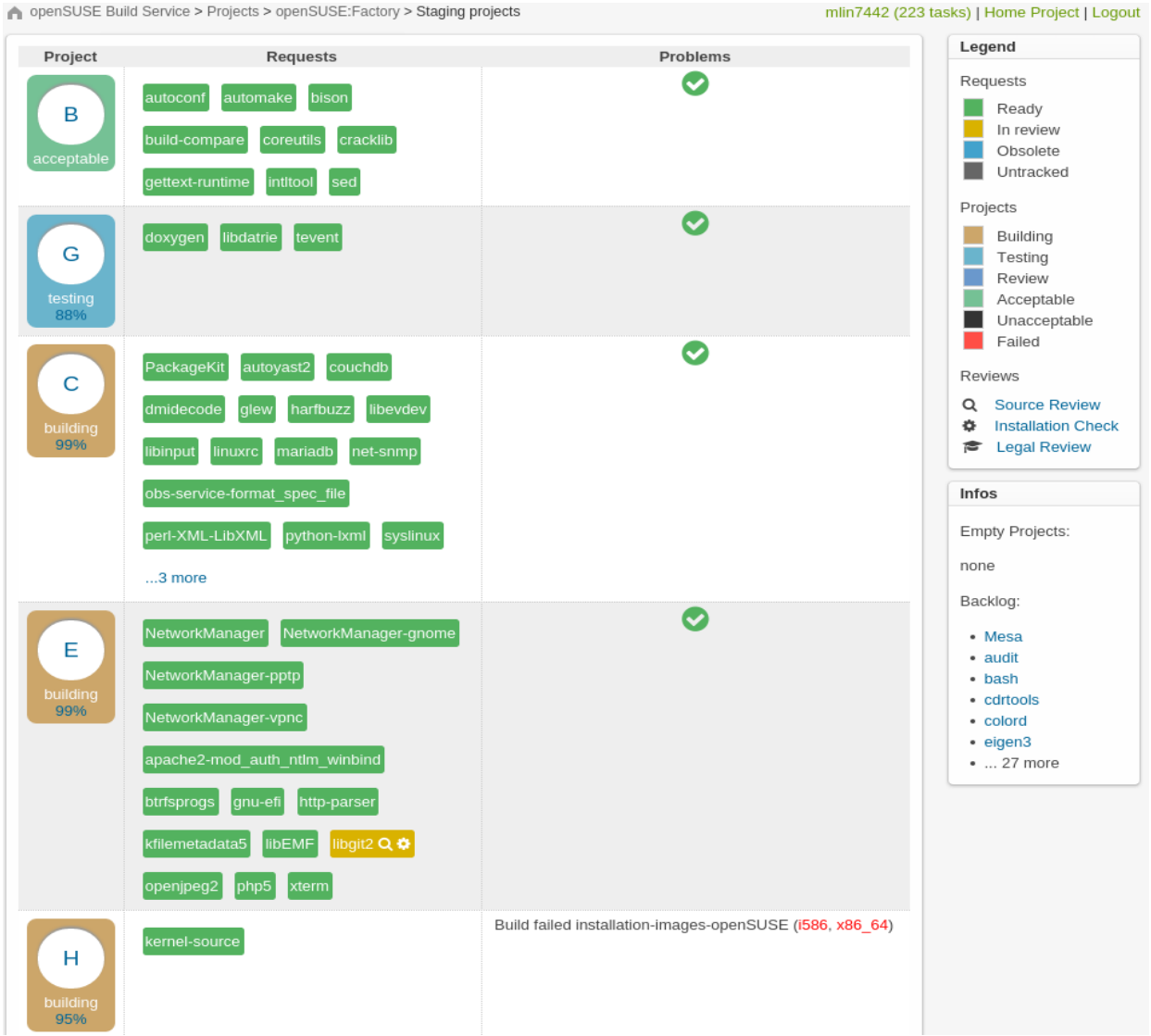
Dashboard

Adding request to staging

- osc staging select LETTER REQUESTS



Staging 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

```
mlin@innoko:~> osc staging check A
 -- FAILED Project openSUSE:Factory:Staging:A still needs attention
   - libcaca: 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
mlin@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
- Featurea is broken, caught by openQA
- etc.

Failure Case 1

- Old binaries tainted the other packages build, bad packages didn't triggered rebuild on OBS somehow. See the failure after merge to Factory.

- Example: 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

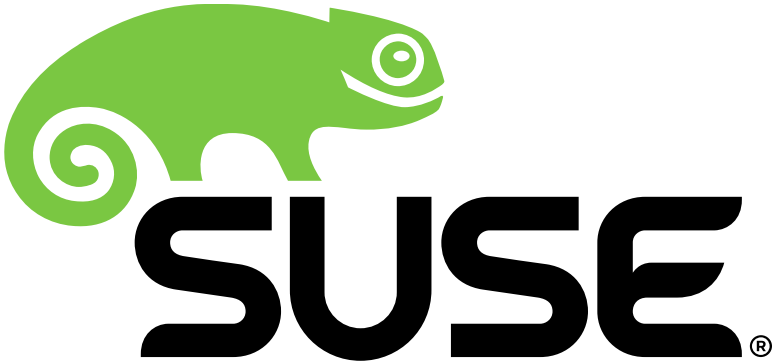
- Example: yast2 software manager crashed in UI mode
- Bug: https://bugzilla.novell.com/show_bug.cgi?id=941398

Failure Case 4

- Dependency submission didn't puts 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, they are puts in different staging, **unluckily** both are build succeeded in its staging, after merge them to Factory, PkgA fails to build with PkgB.
- Rarely happens!

Q&A

Thank you!


SUSE®
We adapt. You succeed.

Corporate Headquarters
Maxfeldstrasse 5
90409 Nuremberg
Germany

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

Join us on:
www.opensuse.org