Release steps for v2.5

Steps needed to make a Grouper release

Final todo's

- WS tests (DONE)
- Look at jars not supposed to be there (from Chad) (DONE)
- Chris and Emily to announce and update wiki
- Emily check wiki make sure things are up to date
  - This week

Verifying cherry-picks from branch are in master

- https://stackoverflow.com/questions/7566416/how-to-see-which-commits-in-one-branch-arent-in-the-other

This is a somewhat tedious step, as not all commits are official cherry picks (they could be two similar commits), and some cherry picks don't get detected as identical changes (e.g. if there is a merge conflict).

As a first step, run `git checkout <branch>; git cherry master`. This will list all the commits in the branch. Commits with a minus (-) sign are already cherry picked. and commits with a plus (+) sign need further investigation. This will narrow down the investigation a lot. Alternatively, you can use `git log --left-right --graph --cherry-mark --oneline GROUPER_2_3_BRANCH...master` which will show the commit graph for both branches, with '<' for the commits only in the branch and '=' for cherry-picked commits.

For the commits needing attention, there is often no quick way to verify they have been applied to master. Probably the easiest thing to do is look at what changed using `git show <commit>`, see what changed, and then `git blame <filename>` in the master branch to see if the changes somehow made it to master. Github also has a git blame function, so you can compare changes in two different windows. Another way to compare, if you can find a master commit with the same commit message, is to do a `git show` in both commits and compare the differences (maybe using the diff command). If the only differences are in line numbers and spacing, the commits are a good match.

For all the commits that can't be accounted for, ask the original committers to look into why they never got cherry-picked.

Changes in git

Edit all the README.txt files (DONE)

Update Grouper specsheet software requirements page (DONE)

Grouper installer (Chris)

- grouper.installer.example.properties
- GiGrouperVersion.java

Grouper WS

- grouper-ws.base.properties
- If there is a new minor version with changes to the wsdl, add a new source folder for that version, copy the coresoap *.javas over to it, refactor to change the package, change the build.xml to build a WSDL for that version. Change GrouperService to be like a previous version but add new methods or change methods in it (DONE)
- Make sure that there is no coresoap (package) in use in src/grouper_ws_v1_6, src/grouper_ws_v2_0, etc. Take out all src/grouper_ws_vx_x except one, make sure they dont depend on each other (DONE)
- Generate the wssec aar's and commit them (DONE)
- Generate WSDLs and commit them (DONE)
- Add src/grouper-ws_v2_4 to the source list of the Maven build-help plugin in grouper-ws/grouper-ws/pom.xml (DONE)

Search for the old version (e.g. 2.2.2 or 2_2_2 or 2_2_002) in all the files (Chris) (TODO)

If its a minor release, change the release version policy page (DONE)

Git Release Commit

This is the single commit that will be the official release. All code work should be finished at this point.

Create a new branch: git checkout -b GROUPER_2_4_BRANCH

Edit all the pom.xmls <version> tags

- TO <version>2.4.0-SNAPSHOT</version>

```
mvn versions:set -DnewVersion=2.4.0

# if the git diff looks ok:
mvn versions:commit

# otherwise
# mvn versions:revert
```

Look for other snapshot references that needs to change: `find . -name pom.xml | xargs grep SNAPSHOT`

Commit in the new branch. Note, if this is in the master branch, or any other branch that Travis will act on, include "[ci skip]" in the commit message, so that it won't build and publish to the Sonatype snapshot repository.

Push the new branch (TODO needs an extra git command to track the remote?)

For the commit just pushed, tag as GROUPER_2_4_0 and push

```
git tag GROUPER_2_4_0
git push --tags
```

Or, if there is already a GROUPER_2_4_0 tag, these steps will override it in the remote repository:

```
# Retag:
# https://stackoverflow.com/questions/8044583/how-can-i-move-a-tag-on-a-git-branch-to-a-different-commit
# https://hackernoon.com/under-the-sink-removing-remote-git-tags-3d5644d67ea0

# (this will force delete the tag in the remote tag, since there is no source before the colon)
git push upstream :refs/tags/GROUPER_2_4_0

    # > To https://github.com/Internet2/grouper.git
    # >  - [deleted]                   GROUPER_2_4_0

# Force update the local tag to the current commit
git tag -fa GROUPER_2_4_0

# Dry run - note in the output that it will push the commit plus the tag
git push --dry-run upstream master --tags

git push upstream master --tags
```

In the branch, change pom 2.4.0->2.4.1-SNAPSHOT, update .travis.yml to include the 2.4 branch, commit, test build, commit, and push

In master, change pom 2.4.0-SNAPSHOT->2.5.0-SNAPSHOT, test build, commit, and push

Handle Copyright According to Policy

Run the copyright wizard on the entire branch for java files. Or you can diff in version control and go through the new files. Note, only new files should have diffs... this is the custom copyright. You can update the year for new files, don't update the year for existing files. The year should be used and not a range. That is our policy on copyrights. (dont want to change all files)

```
Copyright 2015 Internet2

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

  http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

Make sure the subjects.sql and quickstart.xml file are in the release directory (copy from 2.3.0)

Misc Checks

- Make sure all tables, views, and cols (of tables and views) have comments in the DB (oracle or postgres). (TO DO on a rainy day)

Security Review

Review UI / WS code changes* from the previous version with respect to Open Web Application Security Project (OWASP) top 10 web application security vulnerabilities

If possible run a security scan against a test instance - University of Pennsylvania via Chris?

*We have not carried out a thorough security review of the existing code base for any version of Grouper. We should do that in order for the incremental reviews to be adequate

Testing

Run API (SuiteDefault) JUnit tests (set true in all JUnit test includes in grouper.properties). Also run the Grouper Installer, it should end in success for the client connecting to the WS and you should be able to use the UI.

API JUnit Tests

Database	Install	Upgrade
HSQLDB		
MySQL (with utf/bin collation table types)	Bill	Bill
Oracle	Shilen	Shilen
PostgreSQL	Chad	Chad

- MySQL windows
- MySQL unix with case sensitive table names
- Postgres
- Oracle

```
create role GrouperRole;
GRANT connect, resource, CREATE view TO GrouperRole;
CREATE USER groupertest IDENTIFIED BY <pass>;
GRANT connect, resource, GrouperRole TO groupertest;
```

- Hsql

```
ps -ef | grep java   --- kill hsql
wget http://www.internet2.edu/grouper/release/x.y.z/grouper.apiBinary-x.y.z.tar.gz
cd grouper.apiBinary-2.0.0
cd bin
java -cp ../lib/jdbcSamples/hsqldb.jar org.hsqldb.Server -port 9001 -database.0 file:grouper -dbname.0 grouper &
./gsh.sh -registry -runscript
./gsh.sh -test -all
```

Try the UI with a few different languages in the browser request (en without US, french, something not common)

Try checkConfig tests.

Upgrade from x.y-1 to x.y (e.g. 2.3 to 2.4).

- Install 2.3.0
  - Download and run 2.4 installer
- Open the UI, browse around
- Upgrade to 2.4.0: v2.4 Upgrade Instructions from v2.3
  - Download the installer
-
- Open the UI, browse around

Install the grouper installer

- Download grouper installer
- Run: java -jar grouperInstaller.jar
- Try the UI that was installed
- Make sure gsh.sh and gsh works in api, ws, ui for 2.4.0

Database	Linux	OS X (10.6)	Solaris	Windows
HSQLDB	CH			
MySQL				
Oracle				
PostgreSQL				
MsSQL (NO LONGER SUPPORTED)				

Web Service JUnit Tests (Grouper WS and Grouper Client) (DONE)

WS samples (DONE)

WS javadoc (generate, commit, test). Make sure new operations / args / etc are documented in the WS doc page (TODO?)

UI internationalization tests (TODO?)

Test API

```
# try to see if someone else is testing, look when process started, wait if so or email list...
[appadmin@i2midev6 ~]$ ps -ef | grep gsh | grep test | grep all

getGrouperSource.sh GROUPER_2_4_BRANCH
cd  /home/mchyzer/tmp/grouperDownload/build_mchyzer/grouper-GROUPER_2_4_BRANCH
cd grouper
./start-hsql.sh

cd /home/mchyzer/tmp/grouperDownload/build_mchyzer/grouper-GROUPER_2_4_BRANCH/grouper
ant dist
./bin/gsh.sh -registry -runscript
./bin/gsh.sh -test -all
```

Packaging and releasing

- ssh to i2mibuild.internet2.edu
- cd to ~mchyzer

Update Software+Download page

https://spaces.at.internet2.edu/display/Grouper/Grouper+Downloads (DONE)

- Make sure you have copied this to the archives page per above
- freshen links and rows in software download table
- freshen cvs info at bottom of page
- update release date

Review product pages

- Ensure that each page identifies which version it's current for, as in "Building the Grouper API as of v1.4.1" as an h2 at the top.
- Review incoming and outgoing links info for each page.
- If there's new features with new pages, update the Grouper+Product page to appropriately reference the new pages.
- Is it all there?
- Are there pages in the Working Group or Community areas that have graduated to being core product doc? Negotiate with originator to move them over to the product pages.

Update GrouperWG/Home

Just update the "NEW!!" message on this page. Maybe review the Background section to see if it could use some freshening. Maybe add a news item

Review Training Videos

- Determine which Grouper training videos need to be updated. Are new training videos required?

Review Impact, if any, on TIER Grouper Deployment Guide (DONE?)

- Consider if the new release should lead to any changes/updates in the TIER Grouper Deployment Guide
- if yes, discuss with Bill Thompson, Lafayette College

Update Grouper Wiki Documentation

- Update the Grouper Downloads page (DONE)

- Update the spec sheet (DONE)
- Update the Grouper Deployment Guide (DONE)
- Update wiki pages for all changed features.
- Add a new selection to the Grouper Admin Guides TOC if needed. See here https://spaces.at.internet2.edu/x/UAfw
    - Be sure that Grouper wiki doc pages for new features are moved from the Development Items are and area and instead linked from the Admin Guides area.
- Review the Glossary, remembering this could be a first place new people go to learn important terms. It seems rusty to Emily but not exactly sure how to improve it. HELP appreciated.
- Be sure that wiki pages on new Features are linked correctly from the Admin Guides TOC page

Notify about the release

- Compose email to grouper-dev@internet2.edu and grouper-users@internet2.edui and grouper-announce@internet2.edu with highlights of the new release and link to the Grouper Downloads page
- email to mw-announce [at] internet2.edu email list
- Highlights should resemble those on the vN.N Release Notes page and maybe even be identical.
- Change jira admin to make that version released, and if the next build or version isnt there, add it

Other Outreach

- update Grouper website features page to highlight newest Grouper features
- Social Media: Announce the new Grouper release on the InCommon and Internet2 Facebook and Twitter accounts as appropriate (Jessica Coltrin coordinates InCommon social media as of April 2020)
- work with Jessica Coltrin (and possibly Sara Aly, Internet2 Communications Manager,) around other possible publicity/press releases etc.
- A Grouper Team member (DaveL for the Grouper 2.2 release) will create a Youtube video (using same tools as the Grouper Training videos) highlighting new features
    - promote this video in coordination with the Internet2 Marketing and Communications Team
- schedule a webinar to promote the new release

Release in sonatype

The Maven step will package all the artifacts, gpg sign them (creating *.asc for all the objects) then upload them to the Sonatype staging repository

Go to https://oss.sonatype.org/#stagingRepositories and there should be a newly created grouper repository with the artifacts inside. The login and password for Sonatype can be found in ~/.m2/settings.xml.

Maven central is https://search.maven.org/search?q=grouper

Sonatype URL immediate repo:

- https://oss.sonatype.org/service/local/repositories/releases/content/edu/internet2/middleware/grouper/grouper-ws-scim/2.5.2/grouper-ws-scim-2.5.2.jar
- https://oss.sonatype.org/#nexus-search;quick~grouper

If the objects are all there and look ok, click the close icon to freeze the repository. At that point, you should test the objects, by trying to building something, using the staging repository as if it were the release repository. In your .m2/settings.xml, add:

```
  <profiles>

    <profile>
      <id>grouper-stage</id>
      <repositories>
        <repository>
          <id>grouper-stage</id>
          <url>https://oss.sonatype.org/service/local/repositories/eduinternet2middlewaregrouper-1014/content<
/url>
          <releases>
            <enabled>true</enabled>
          </releases>
        </repository>
        <!-- ... -->

  <activeProfiles>
    <activeProfile>grouper-stage</activeProfile>
  </activeProfiles>
```

If it builds ok, click the release icon for the repository, and then it will be moved to the official release repository. Within a few hours, it may be synchronized to other Maven repositories.

If something isn't right with the staging repository, just drop it and the Maven build will create a new one with a different increment number.

Workflow for container including CI

Build

1. add tag (GROUPER_RELEASE_a.b.c) and push

CI (travis)

1. tag triggers a build
2. download that zip archive from github (github makes archives available for any tag)
3. unpack archive
4. Parse the tag to convert GROUPER_RELEASE_a.b.c to a.b.c
5. abort if the version from the tag doesn't match version in grouper.version
6. update maven versions to the new version (`mvn versions:set -DnewVersion=a.b.c`)
7. build and deploy to staging
8. Mail committer that either the staged jars in Maven Central are ready to release, or any errors

INSTALL

1. input parameter to grouper installer is grouper version
2. download that zip archive from github (github makes archives available for any tag)
3. download the jar plus dependencies from maven
4. assemble the container

Grouper Failing Tests

Test	Owner	Status and date	Reason it was failing