1. **Install and configure Git**

   a. ***Install Git***

   Go to 'https://git-scm.com/download' and select to download Git version

   // Make sure to add Git into your System path: ***C:\Program Files (x86)\Git\cmd***

   b. ***Add SSH key to Stash account***

   This step is now optional since the remote uses HTTP by default.

   ❖ ***Generate SSH key***

   Before you generate an SSH key, you can check to see if you have any existing SSH keys.

   - Open Git bash.
   - Enter ***ls -al ~/.ssh*** to see if existing SSH keys are present:
   - Check the directory listing to see if you already have a public SSH key.

   If you don't have an existing public and private key pair, or don't wish to use any that are available to connect to Stash, and then generate a new SSH key.

   - Open Git bash.
   - Paste the text below, substituting in your Stash email address.

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

   - When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.

```
Enter a file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]
```

   - At the prompt, type a secure passphrase.

```
Enter passphrase (empty for no passphrase): [Type a passphrase]
Enter same passphrase again: [Type passphrase again]
```

   ❖ ***Adding your SSH key to the ssh-agent***

   Ensure ssh-agent is enabled:

   - If you are using Git bash, turn on ssh-agent

```
# start the ssh-agent in the background
eval "$(ssh-agent -s)"
Agent pid 59566
```

   - If you are using another terminal prompt**,** such as Git for Windows, turn on ssh-agent

```
# start the ssh-agent in the background
eval $(ssh-agent -s)
Agent pid 59566
```

Add your SSH key to the ssh-agent. If you used an existing SSH key rather than generating a new SSH key, you'll need to replace ***id_rsa*** in the command with the name of your existing private key file.

```
$ ssh-add ~/.ssh/id_rsa
```

❖ ***Adding a new SSH key to your Stash account***

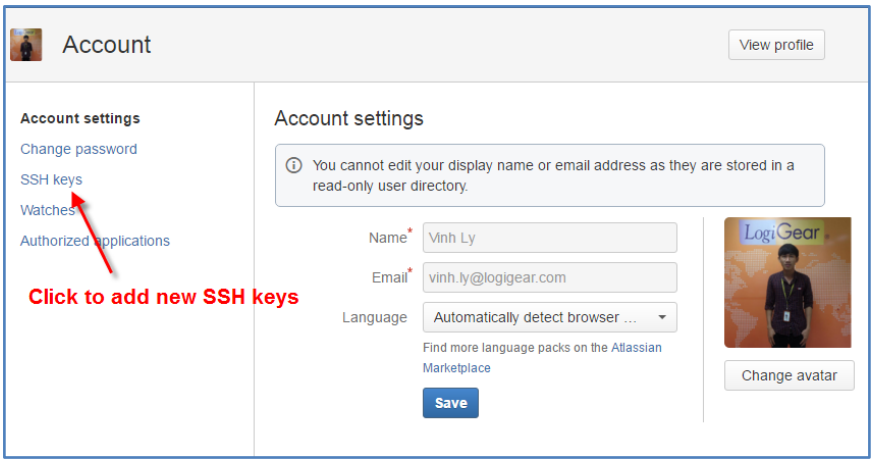
To configure your Stash account to use your new (or existing) SSH key, you'll also need to add it to your account.

- Copy the SSH key to your clipboard

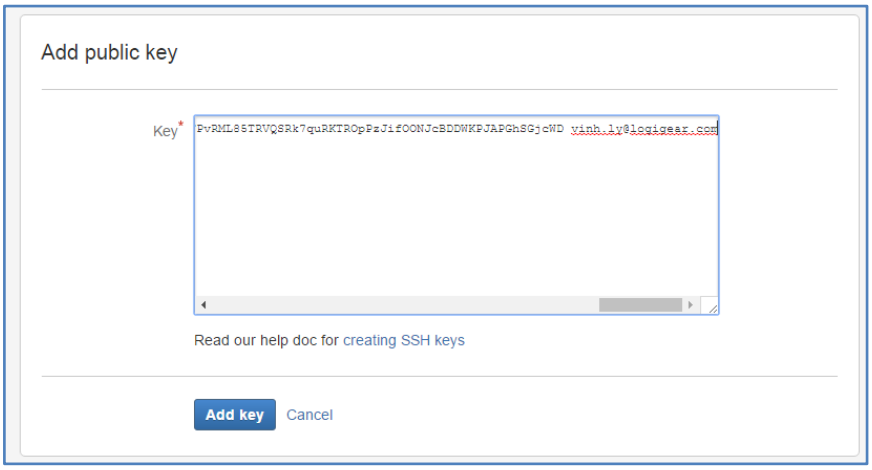
  If your SSH key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

```
$ clip < ~/.ssh/id_rsa.pub
# Copies the contents of the id_rsa.pub file to your clipboard
```

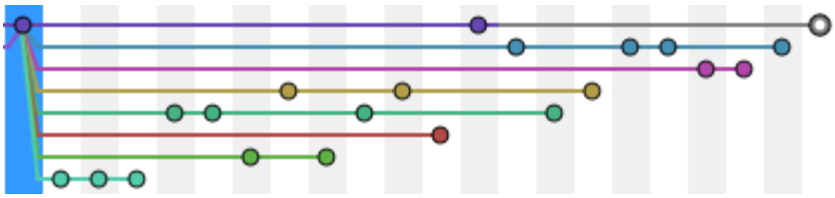
- Go to Stash Account Setting, select 'SSH Keys' menu

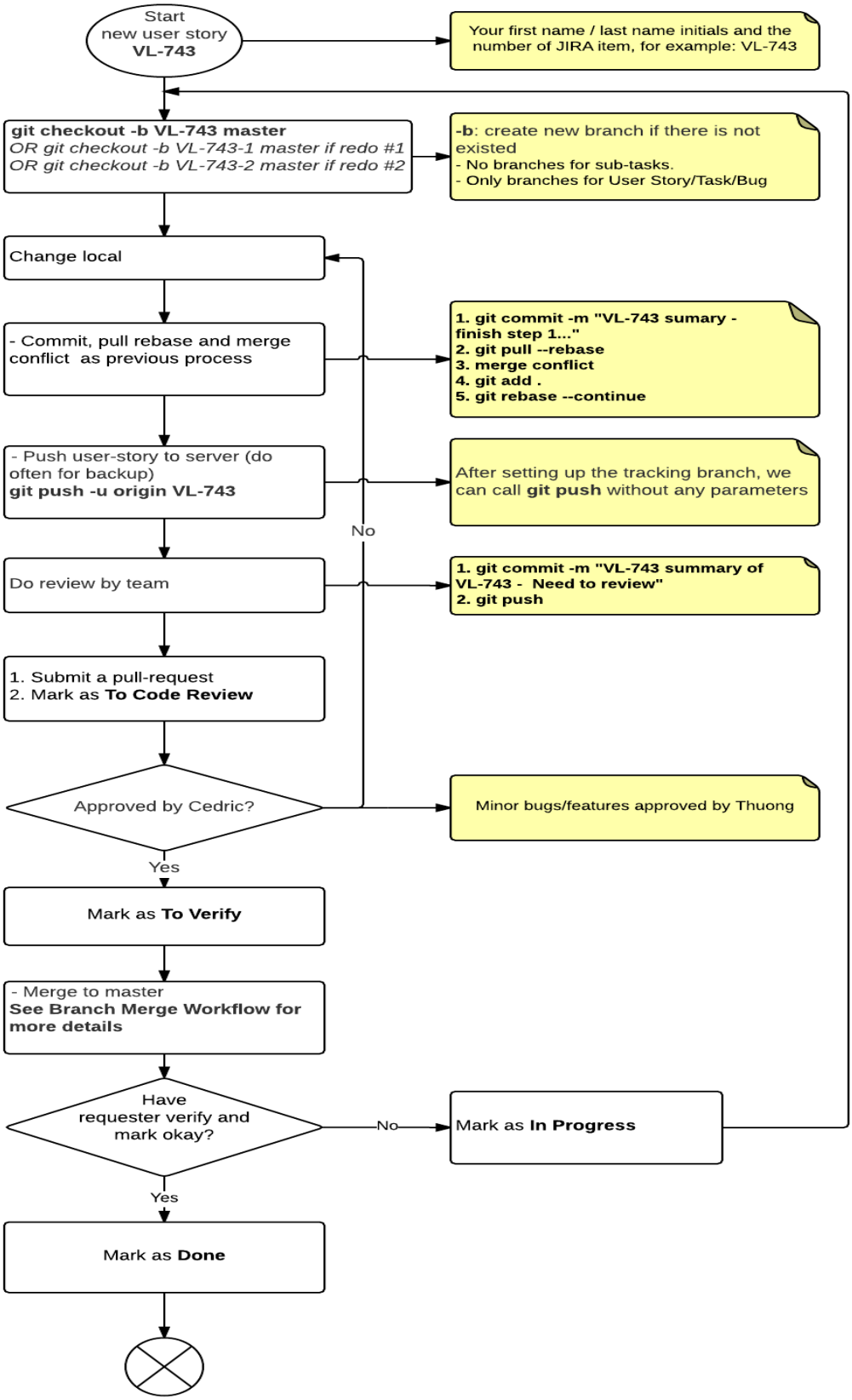


- Click on 'Add key' button then paste your key into key field

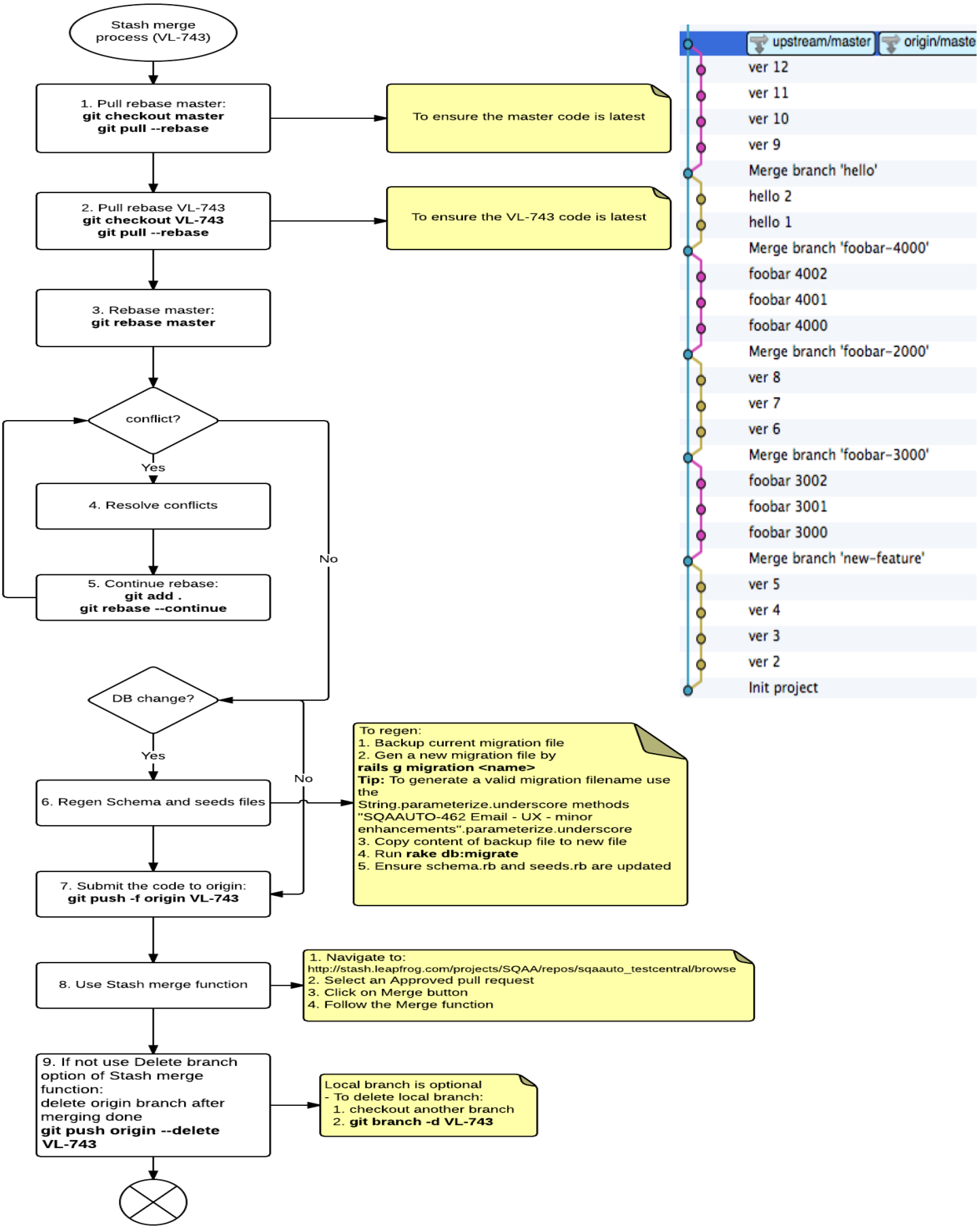


2. Git working process

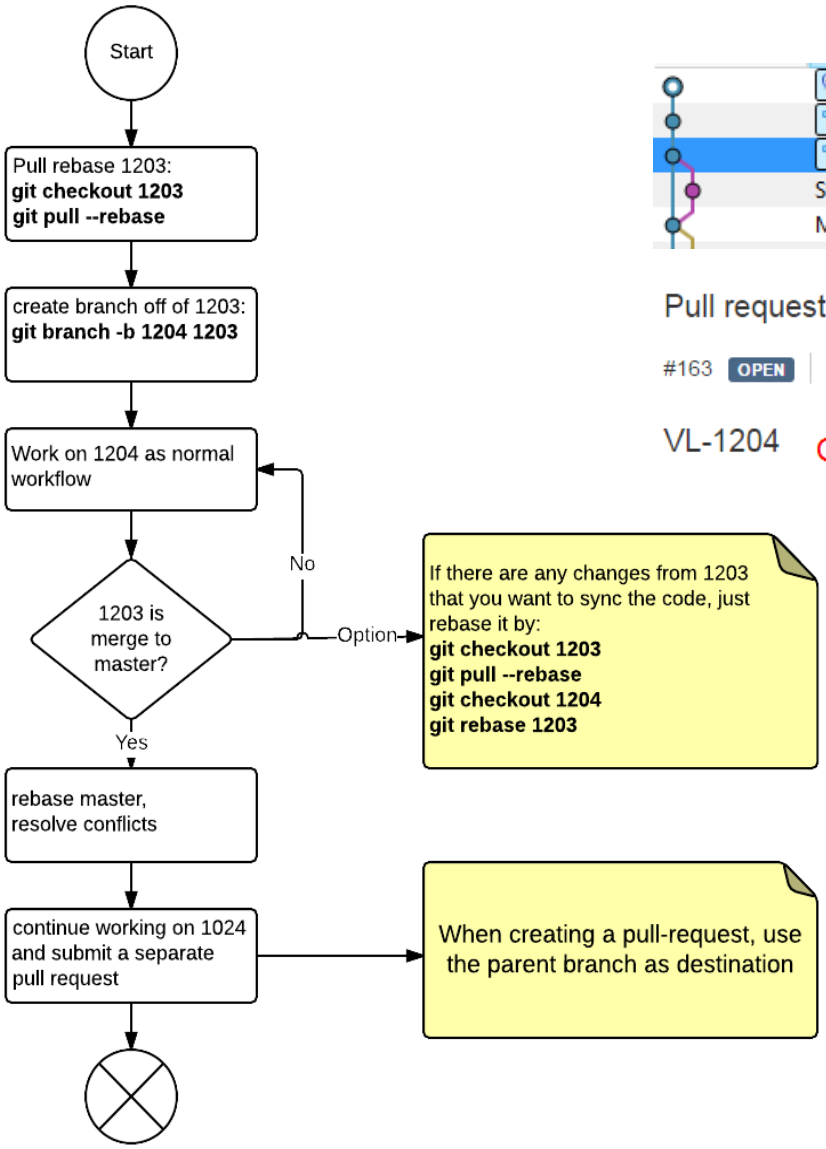


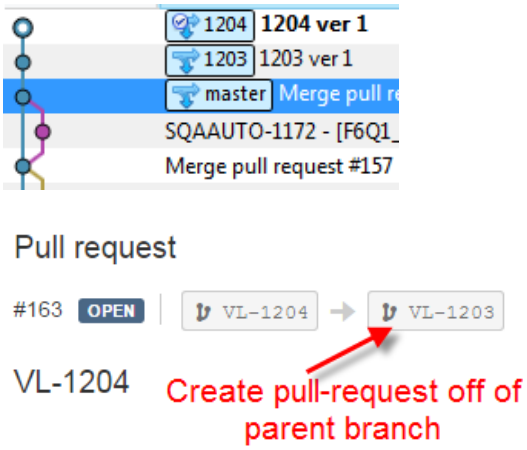


3. Merging process



4. Working with dependent Branch





5. Working with Sub-module

- Main project: ssh://git@stash.leapfrog.com:7999/sqaa/sqaauto_narnia.git
- Sub-module project: ssh://git@stash.leapfrog.com:7999/sqaa/sqaauto_outpost.git

1. Add an existing Stash repository (sqaauto_outpost) as a submodule on a Main project (sqaauto_narnia)
   git submodule add ssh://git@stash.leapfrog.com:7999/sqaa/sqaauto_outpost.git

2. Clone a Main project that included a sub-module project
   git clone --recursive ssh://git@stash.leapfrog.com:7999/sqaa/sqaauto_narnia.git

   // At the first time getting code, go to 'sqaauto_outpost' folder and checkout to Master branch
   git checkout master

3. Update the checkout to the master/branch of the sub-module repository.
   git submodule update –remote [branch name]

4. Working with sub-module project (checkout, create new branch, work with branch, rebase, merge…)
   All are similar to work with a normal project (Refer to 2 above sections: **Git working process** and **Merging process**)