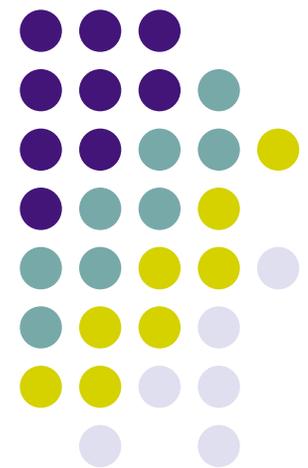
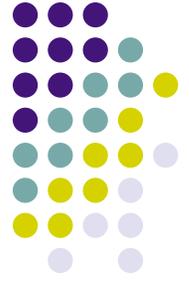


NU-WRF Repository Tutorial

March 11, 2011

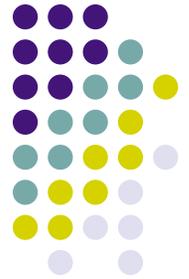




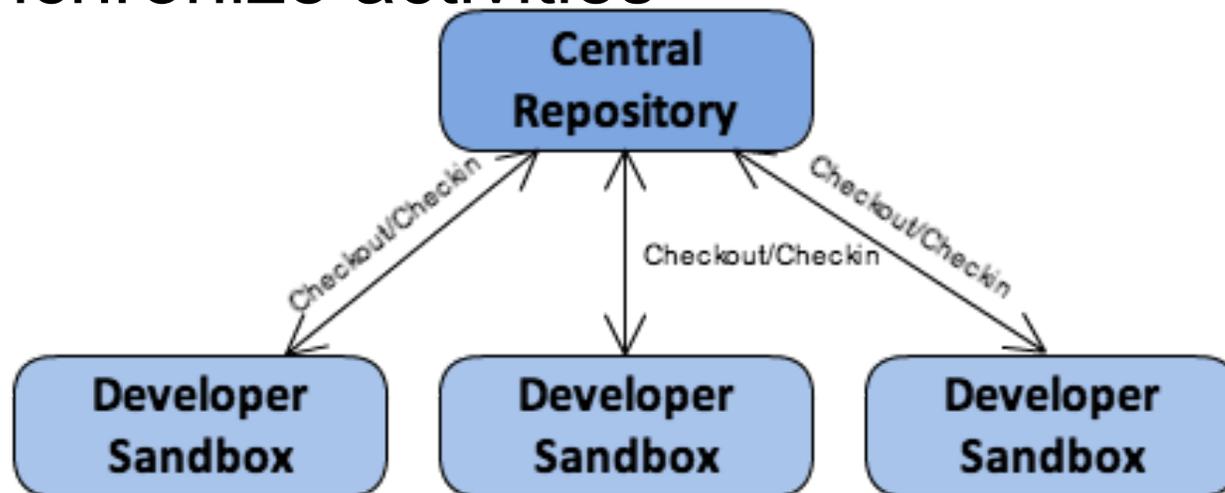
Outline

- Basics of a repository
- How developers generally use them
- Getting your Discover account set up to access the NU-WRF repository
- A few basic commands (like merging the latest changes into your code)

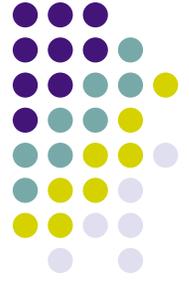
What is the NU-WRF repository?



1. Stores an ongoing history of changes for the NU-WRF project lifetime
2. Place where NU-WRF developers can synchronize activities



Why is revision control software useful?



NU-WRF uses **revision control software** called ***Subversion***

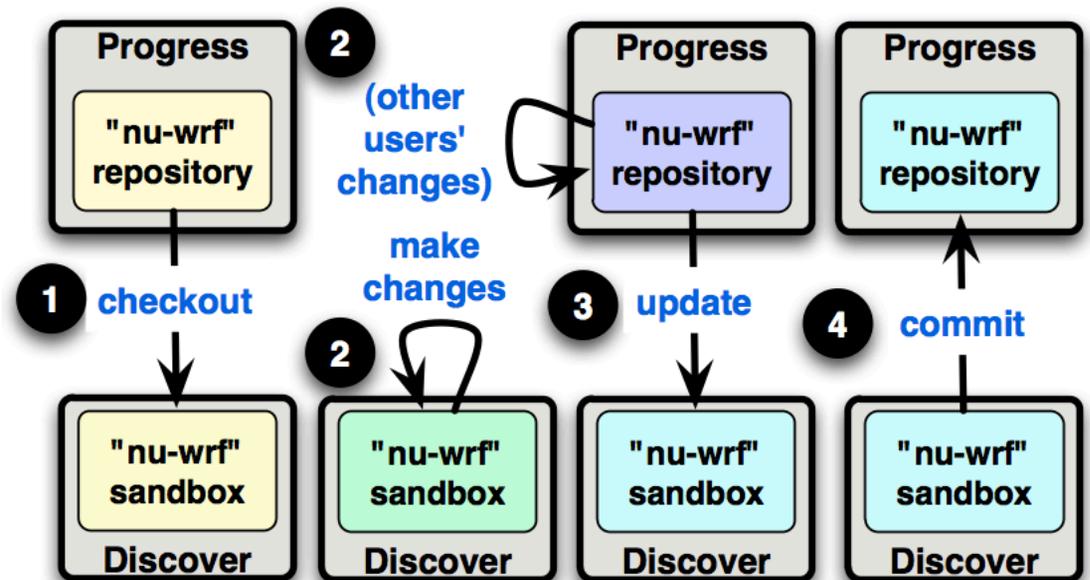
Revision control software advantages:

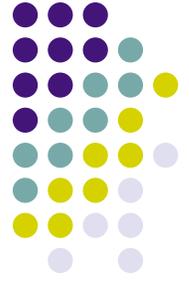
- Have a backup
- Revert to past project states
- Do (safer) transaction-based operations (avoid overwriting someone's change)
- Avoid/resolve code conflicts
- Support multiple product lines/versions

Typical Developer Process When Using Repositories



1. Checkout a local “sandbox” copy of the repository
2. Make your code modifications
3. Update your sandbox with recent changes in the repository from others
4. Check-in your work



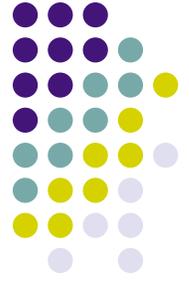


Setting up Access

Will now reference the directions on Modeling Guru for setting up your Discover account to use the repository:

<https://modelingguru.nasa.gov/docs/DOC-1834>

Explore NU-WRF Repository



NU-WRF code repository URL:

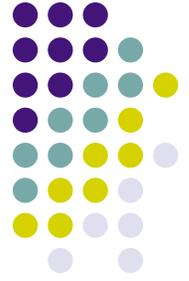
```
svn+ssh://progressdirect/svn/nu-wrf/code
```

List the repository:

```
svn ls <url>  
svn ls <url>/trunk  
svn ls <url>/tags
```

- **trunk** = generally the latest, but not always the most stable!
- **tags** = special snapshots, like releases
- **branches** = variants of the same code, such as with a special feature under development

Create a “Sandbox” Working Copy



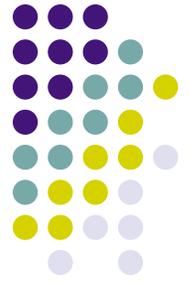
NU-WRF code repository URL:

```
svn+ssh://progressdirect/svn/nu-wrf/code
```

Checkout the project:

```
svn co <url>/trunk <localname>
```

See What You've Changed Since Checkout



Edit a file:

```
cd <localname>
```

```
emacs CHANGELOG.TXT #change the file, save
```

See all you've changed since checkout:

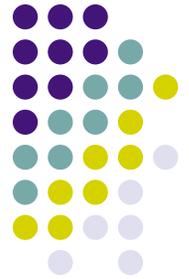
```
svn status
```

```
svn diff CHANGELOG.TXT
```

Brief key for 'svn status':

- **M** = modified file
- **C** = conflicting changes in local/repository file
- **?** = new local file
- ***** = newer version of file in repository

See What Both You AND Others Have Changed



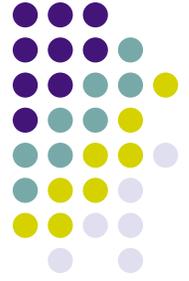
Also show changes in the remote repository:

```
svn status --show-updates
```

Brief key for 'svn status':

- **M** = modified file
- **C** = conflicting changes in local/repository file
- **?** = new local file
- ***** = newer version of file in repository

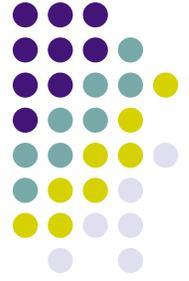
Merge Latest Repository Code into Your Sandbox



Update sandbox with the latest changes:

`svn update`

- This will automatically merge all the latest code in the repository into your sandbox.
- It may be useful to perform `svn status` first.



Basic Subversion Commands

- **svn ls** – list files in repository
- **svn co | export** – checks out the code
- **svn add | rm | cp | mv** – adds, removes, copies, or moves code in the repository
- **svn update** – merges latest server code into your local sandbox
- **svn commit** – push changes back to server
- **svn diff** – reports differences between two versions of a repository
- **svn status** – reports change status of all files