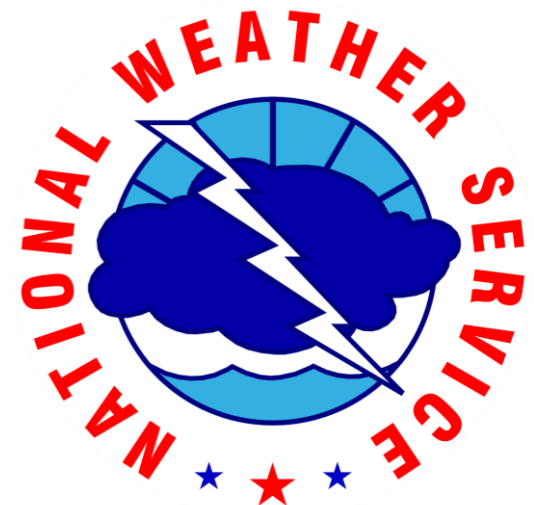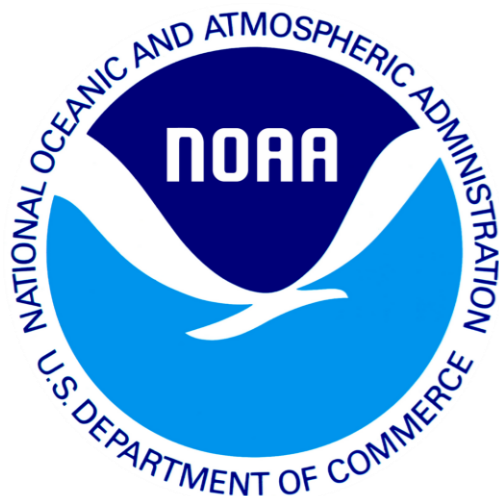# Adopting a Collaborative Software Development Process in the Scientific Community

NOAA's 32nd Climate Diagnostics and Prediction Workshop (CDPW)

Mike Charles
DOC/NOAA/NWS/NCEP/CPC

Motivation

To advocate improved software development and project management techniques, including *collaborative software development* and the use of *project management software*.

Who does this apply to?
- Universities
- Government
- Private sector
- Any software developers

From now on, for the sake of brevity, "collaborative work" will include use of project management software.

Disadvantages of Non-Collaborative Work

Non-Collaborative Work

A developer works alone

- Work goes unchecked by others
- If the owner leaves, nobody knows how the software works
- There are no ideas contributed by others during development

Non-Collaborative Work

A developer doesn't log changes

- Hard to revert to a previous version of the code
- Can't remember what they did last week
- Can't tell why a specific change was made

Non-Collaborative Work

A developer doesn't create documentation

- Hard for others to adopt code
- Hard for backup code owners to fix bugs and/or run code
- Hard to share code with others

Advantages of Collaborative Work

Collaborative Work

How should developers be working?

- ***Collaboration*** - Working with others leads to more ideas and creativity, and more QCing of code
- ***Tracking Changes*** - Using *revision control software,* tracking changes to code and keeping a log makes it easier to revert to older versions of the code, and keep track of work
- ***Documentation*** - Creating documentation and putting comments in code makes the code easier to understand and edit, and makes it easier to share code with others

Collaborative Work

How do we know these methods are better?

Open source software is a great model:
- Started in 1983 - GNU Project - free software
- Open source software is free to edit and distribute
- Usually developed collaboratively
- Usually contains sufficient documentation
- Meant to be shared with many others
- Linux operating system entirely based in open source
- Essentially, open source practices promote better software development

Project Management Software

Collaborative Work

Software to help with collaborative development...

- *Wiki* – A web site developed collaboratively by a community of users, allowing any user to add and edit content.
- *Revision Control* – Management of changes to documents and software, including a history of all changes made and logging for each change.
- *Issue Tracker* – Software that manages a list of issues or tasks that need to be fixed or completed.

There are open source versions of all of these.

Wiki

- Webpage containing text, pictures, and other information that can easily be edited by a team of people
- For example, developers can create a new wiki page for each component of a piece of software (eg. gui, database backend, etc.)
- Developers can also use wiki pages to document scripts and source code

Wiki - Trac

Can format text and images any way you want

Coding Standards

EPG

In order to keep coding practices uniform across all of the developers, we need to follow some guidelines about how to write code. A great resource for us to begin with is the set of ⇨coding standards developed by the Engineering Process Group (EPG). Please see that documentation for more information.

Naming Variables

CamelCase versus Underlines

- Variables should utilize CamelCase between words that are part of a phrase:

```
fileName = '/sandwich/banana/pickle.txt'
```

and an underscore should separate parts of the filename that are not part of a phrase:

```
fileName_sandwich = '/sandwich/banana/pickle.txt'
fileName_salad    = '/salad/apple/mustard.txt'
```

Objects

- When creating an object, the object's name will be the class name plus *Obj*:

```
Data dataObj = new Data();
```

- When a variable is a comma-delimited String that gets split and looped over (eg. users can choose to get Heidke for several models), the comma-delimited version should get *List* appended to the name, and the version that gets looped over (after the split) should retain the original name:

```
fcstSourceList = "off, auto, klein";
for (String fcstSource : fcstSourceList) {
    System.out.println("Forecast source = " + fcstSource);
}
```

Wiki - Trac

Creating a Wiki page
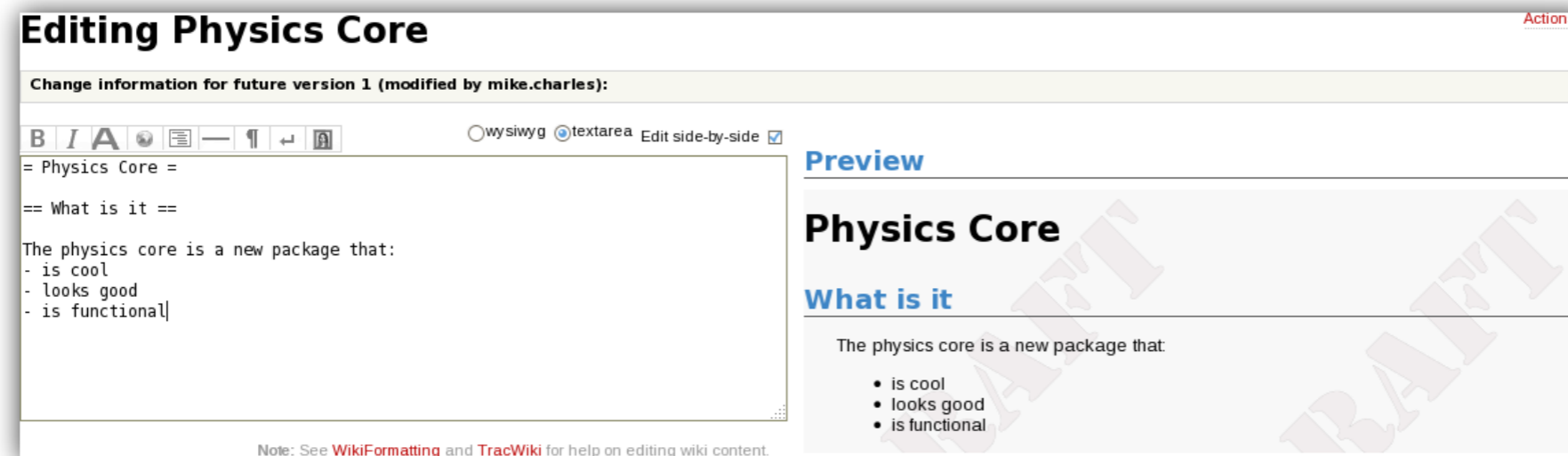
Step 1 - Enter the URL of a page you want to make

http://trac.mycenter.gov/PhysicsCore

Step 2 - Click "Create this page", using an optional template

Create this page Using the template: (blank page)

Step 3 - Add your content

Editing Physics Core

Change information for future version 1 (modified by mike.charles):

wysiwyg textarea Edit side-by-side ☑

```
= Physics Core =

== What is it ==

The physics core is a new package that:
- is cool
- looks good
- is functional
```

Preview

Physics Core

What is it

The physics core is a new package that:

- is cool
- looks good
- is functional

Note: See WikiFormatting and TracWiki for help on editing wiki content.

Wiki - Trac

Keeps history of all Wiki edits

Change History for Coding Team/Coding Standards

View changes

		Version	Date	Author	Comment
○	◉	19	17 months	mike.charles	Added a link to ../Throwing Exceptions
◉	○	18	22 months	mike.charles	Fixed the link to the EPG coding standards PDF
○	○	17	2 years	mike.charles	Renamed all links to using CamelCase
○	○	16	2 years	mike.charles	Added a section on using Javadoc
○	○	15	2 years	mike.charles	Cleaned up the "comments" section
○	○	14	2 years	mike.charles	developing...
○	○	13	2 years	mike.charles	Reorganized
○	○	12	3 years	mike.charles	Added a naming convention for variables that are really comma-delimited ...
○	○	11	3 years	melissa.ou	
○	○	10	3 years	mike.charles	Added information about naming specific kinds of variables
○	○	9	3 years	mike.charles	Added some information about the variables used to pass information ...
○	○	8	3 years	mike.charles	Changed the section on variable naming convention
○	○	7	3 years	mike.charles	Added a section for naming Java packages
○	○	6	3 years	mike.charles	Removed all sections specifically discussing Matlab features
○	○	5	3 years	mike.charles	Switched from the cable CSS class 'wiki' to 'custom', since I realized ...
○	○	4	3 years	mike.charles	Name changed from Coding_standards to Coding_Team/Coding_standards
○	○	3	3 years	mike.charles	Added a table of contents
○	○	2	3 years	mike.charles	Added additional guidelines
○	○	1	3 years	mike.charles	Added the help block section

View changes

Wiki - Trac

Visual diff between any 2 versions

Timestamp: *(multiple changes)*
Author: *(multiple changes)*

Comment: *(multiple changes)*

View differences [inline ▲▼]

◉ Show [2] lines around each change
◯ Show the changes in full context

Ignore:
  ☐ Blank lines
  ☐ Case changes
  ☐ White space changes

[Update]

☐ Unmodified   🟩 Added   🟥 Removed   🟨 Modified

Coding Team/Coding Standards   **Tabular** | Unified

v17	v19	
3	3	= Coding Standards =
4	4	== EPG ==
5		In order to keep coding practices uniform across all of the developers, we need to follow some guidelines about how to write code. A great resource for us to begin with is the set of [http://www2.cpc.ncep.noaa.gov/cpc/CF/documentation.shtml coding standards] developed by the Engineering Process Group (EPG). Please see that documentation for more information.
	5	In order to keep coding practices uniform across all of the developers, we need to follow some guidelines about how to write code. A great resource for us to begin with is the set of [http://www2.cpc.ncep.noaa.gov/cpc/CF/documentation/EPG_Programmers_Manual_infomap.pdf coding standards] developed by the Engineering Process Group (EPG). Please see that documentation for more information.
6	6	
7	7	== Naming Variables ==
...	...	
204	204	|| climateRegion || NCDC Regional Climate Centers ||
205	205	|| state || US States ||
	206	
	207	== Throwing Exceptions
	208	
	209	See [[wiki:../ThrowingExceptions]] for standards on throwing Exceptions in Java.

Revision Control

What is Revision Control?

- Managing and tracking changes to code, documents, and directories.

Revision Control - Subversion

Keeps history & log messages of all changes

Revision Control - Subversion

Merges changes from multiple developers
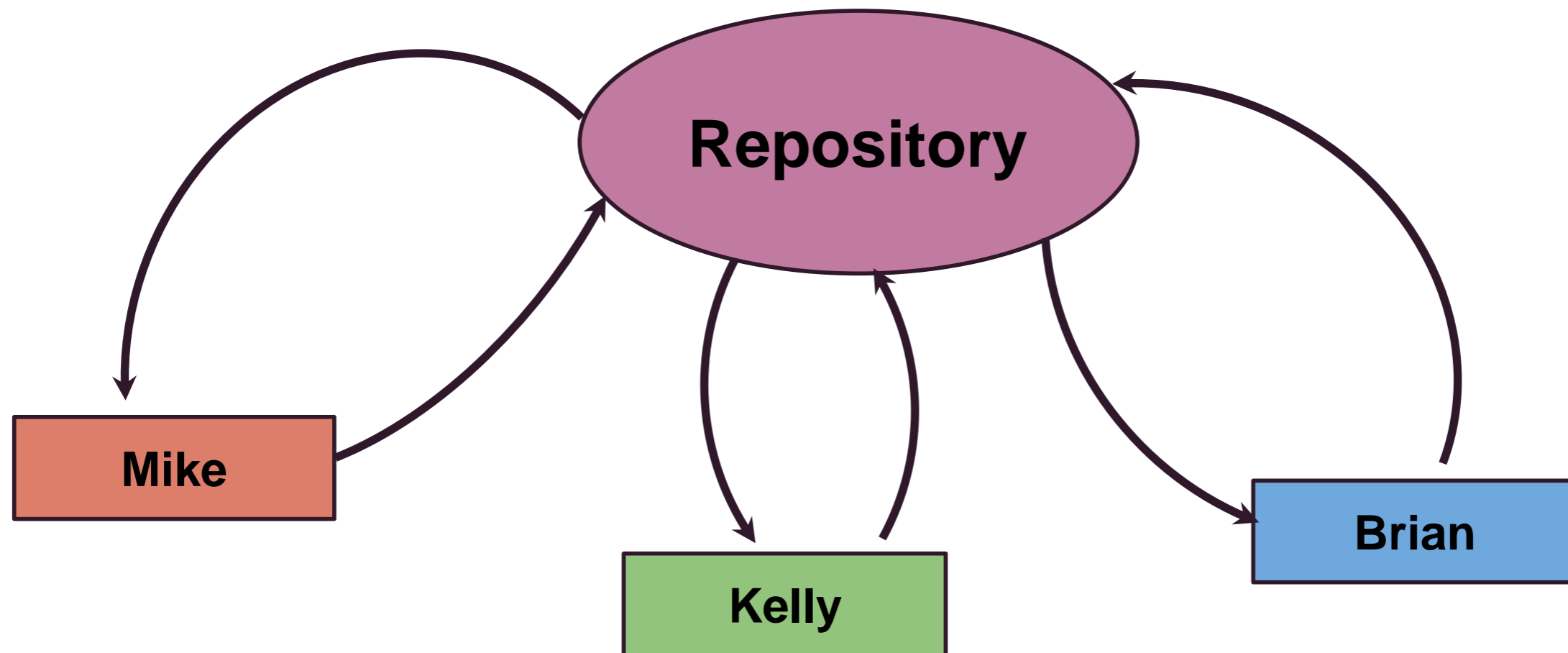
Code is copied from repository, edited in each developer's isolated environment, then sent back to the central repository

Issue Tracking

What is issue tracking?

- Managing a list of issues or tasks
- Manage a list of "tickets" which can be assigned to different developers and store a history of comments made by any of the developers

Issue Tracking - Trac

Create ticket for a specific task or issue

Create New Ticket

Properties

Summary: Drop-down menu not working on home page

Reporter: mike.charles

Description: **B** *I* **A** ⊕ ☰ — ¶ ↵ 🔲   ○wysiwyg ●textarea You may use WikiFormatting here.

> The drop-down menu on the home page doesn't open when the button is clicked. It works in Firefox, but not in Internet Explorer.

Type: defect

Priority: critical

Milestone: 1.0

Component: Web

Version: trunk

Keywords: firefox, ie, javascript

Cc: Edit

Due date: 10/23/2012

Hours spent: 0

Owner: bill.gates

☐ I have files to attach to this ticket

[Preview] [Create ticket]

Issue Tracking - Trac

Anyone can comment on tickets, reassign, etc.

Issue Tracking - Trac

List tickets using optional filters

Subversion/Trac Combo

- RSS feed support
- Email notification of changes to ticket
- Create milestones
- 100s of plugins available for Trac
- View source code & Subversion code history within Trac
- Connect specific tickets to specific code revisions
- Use Trac to keep track of all work done on a project

Summary

- Current practice in many organizations is for developers to work alone on software
- Better practice is to work collaboratively
- Wikis help a team share documentation
- Version control helps track changes to code and revert to any previous version
- Issue tracking helps developers assign tasks and track work on issues and tasks
- We should all use open source software as a model for efficient, creative work

Thank you

Trac - http://trac.edgewall.org
Subversion - http://subversion.apache.org

Any questions - email me at mike.charles@noaa.gov