

# CVS Tutorial

## An Introduction to Using

CVS

Versions System

Concurrent

\* collaboration

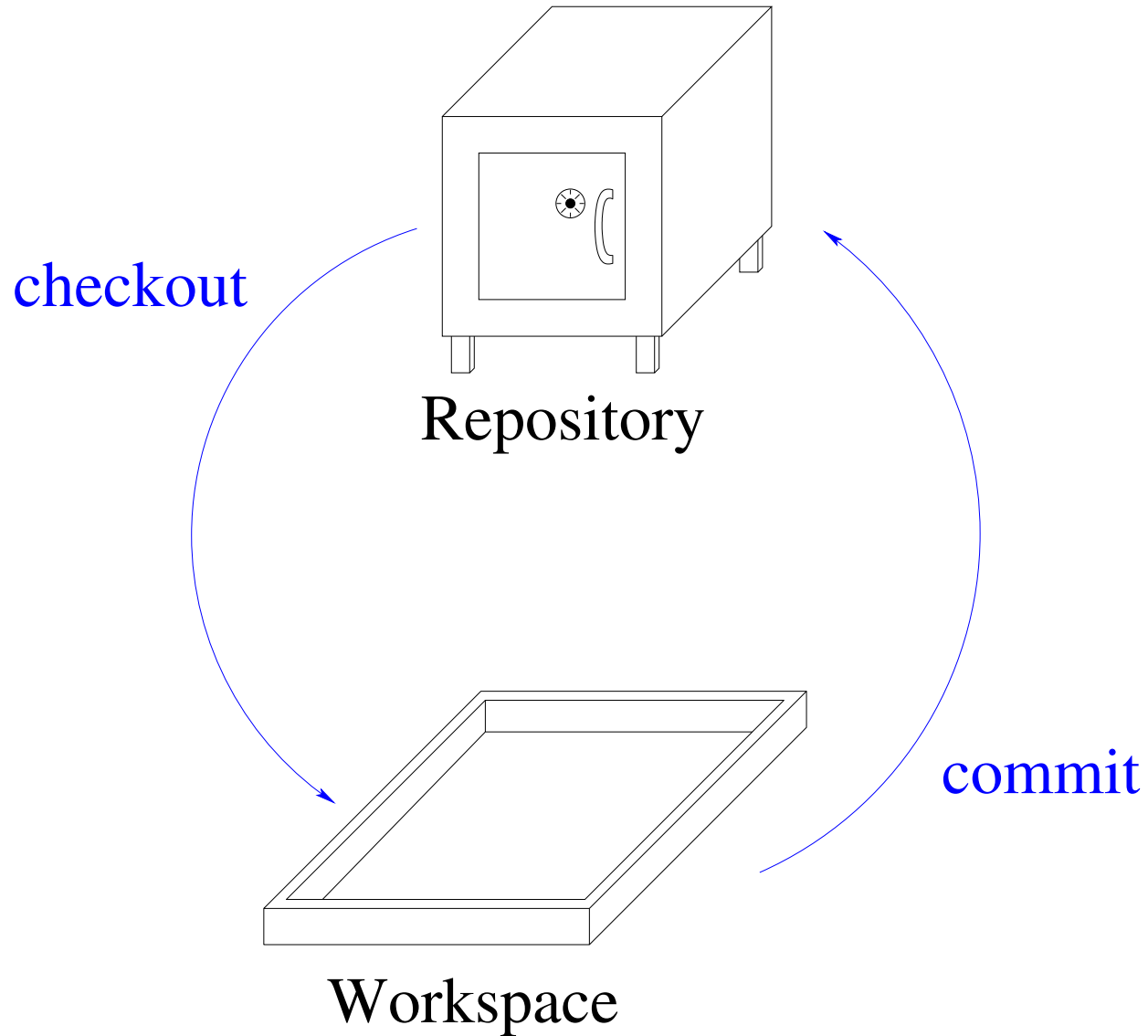
\* history: when? why?

\* examine old revisions

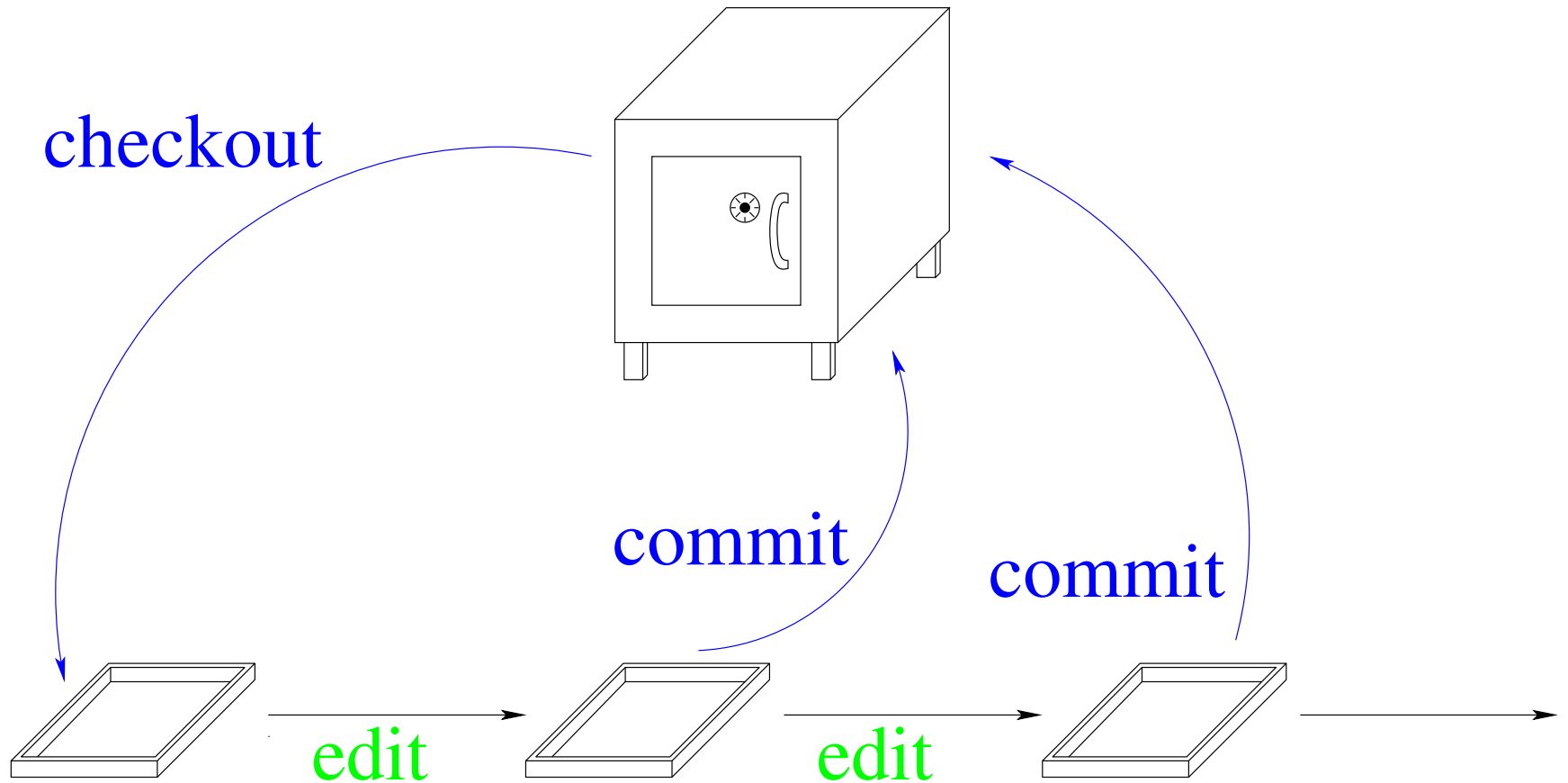
\* bugfix releases

*Steve Robbins*

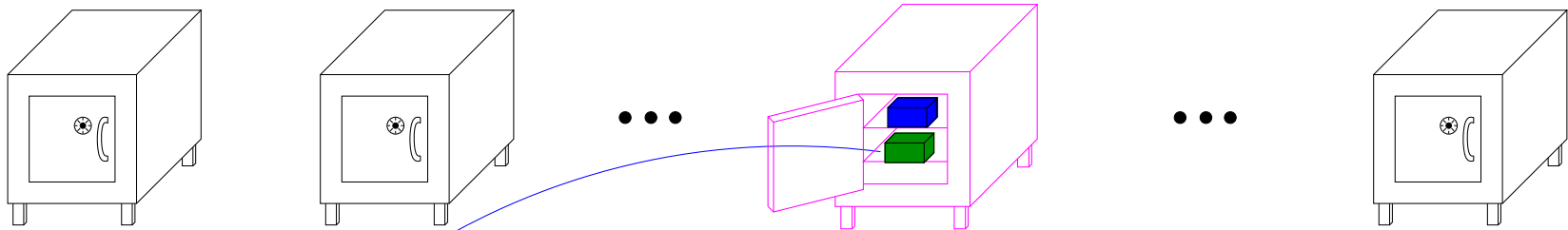
# CVS Tutorial



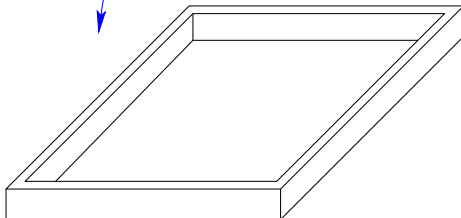
# CVS Tutorial



# CVS Tutorial



`cv` `-d` repository-name `checkout` module-name



```
> cv -d /software/examples checkout hello
```

```
cv
```

 checkout: Updating hello

```
U hello/Makefile
```

```
U hello/hello.c
```

```
U hello/world.c
```

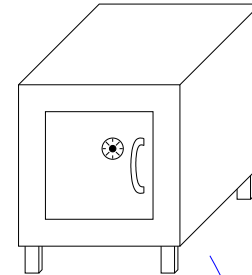
```
U hello/world.h
```

# CVS Tutorial

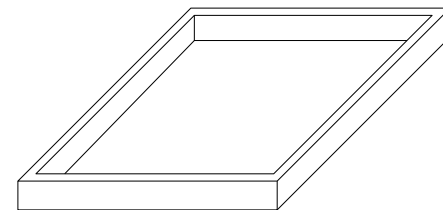
```
> cd ~/src
> cvs -d /software/examples checkout hello
cvs checkout: Updating hello
U hello/Makefile
U hello/hello.c
U hello/world.c
U hello/world.h
```

```
> cd hello
> ls
CVS    Makefile hello.c world.c world.h
```

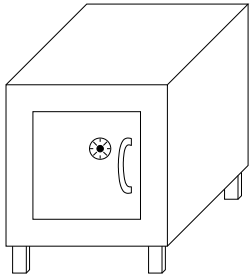
```
> ls CVS
Entries  Repository Root
```



checkout



# CVS Tutorial



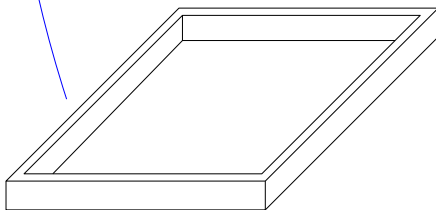
```
> cvs commit  
cvs commit: Examining .
```

```
CVS: -----  
CVS: Enter Log. Lines beginning with 'CVS:' are removed automatically  
CVS:  
CVS: Committing in .
```

```
( $CVSEEDITOR  
  $EDITOR )
```

```
Checking in hello.c;  
/software/examples/hello/hello.c,v <-- hello.c  
new revision: 1.2; previous revision: 1.1  
done
```

**commit**



# CVS Tutorial

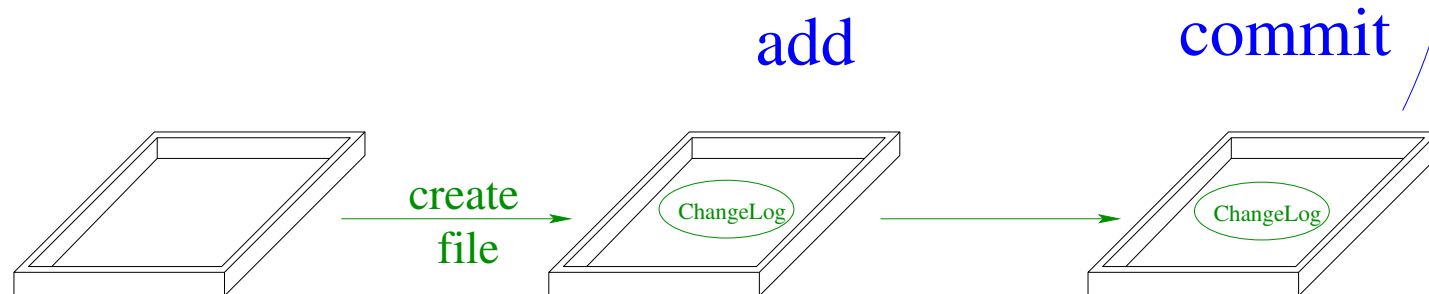
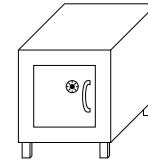
## > **cv**s add README

cv

s add: scheduling file 'README' for addition  
cvs add: use 'cvs commit' to add this file permanently

## > **cv**s commit

cv

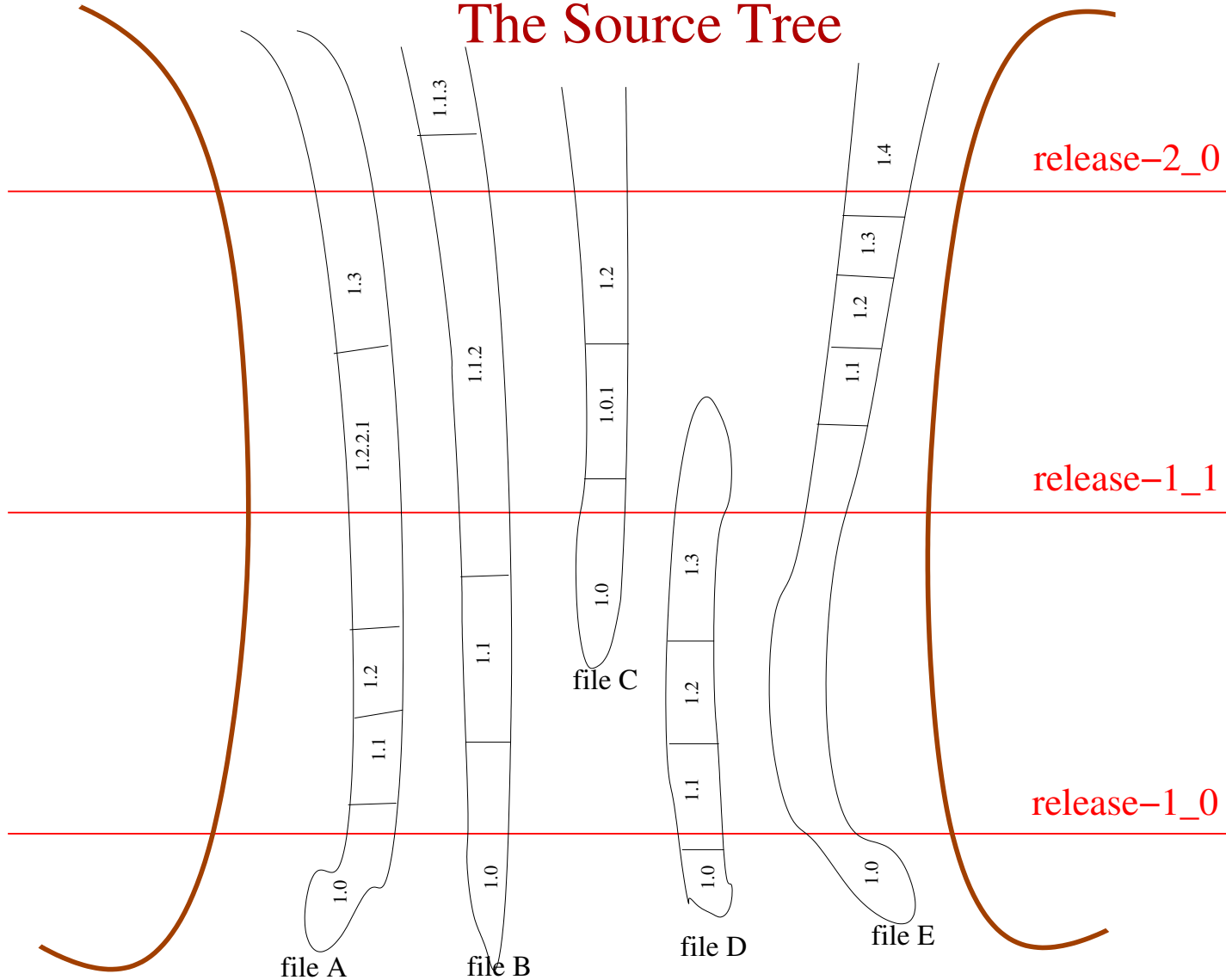
s commit: Examining .  
RCS file: /software/examples/hello/README,v  
done  
Checking in README;  
/software/examples/hello/README,v <-- README  
initial revision: 1.1  
done

**Remove files:** rm filename; **cv**s remove filename

**Rename files:** remove then add

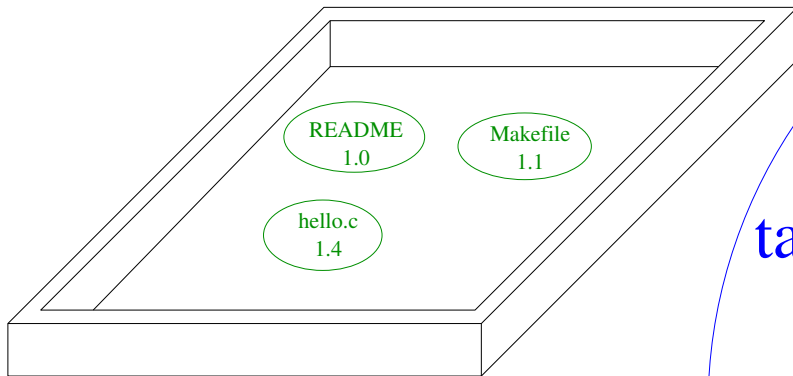
# CVS Tutorial

## The Source Tree

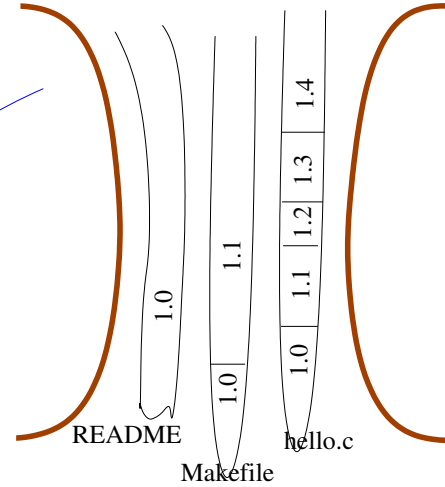




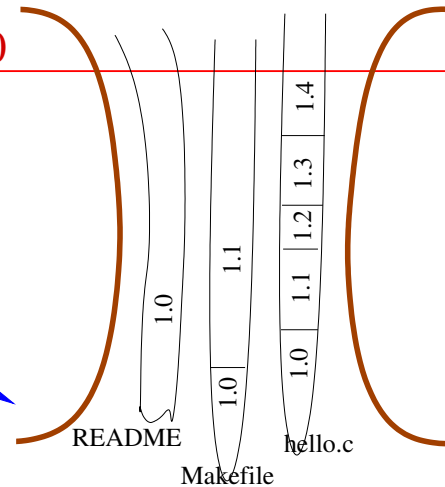
# CVS Tutorial



tag



release-1\_0



```
> cvs tag release-1_0
```

```
cvs tag: Tagging .
```

```
T Makefile
```

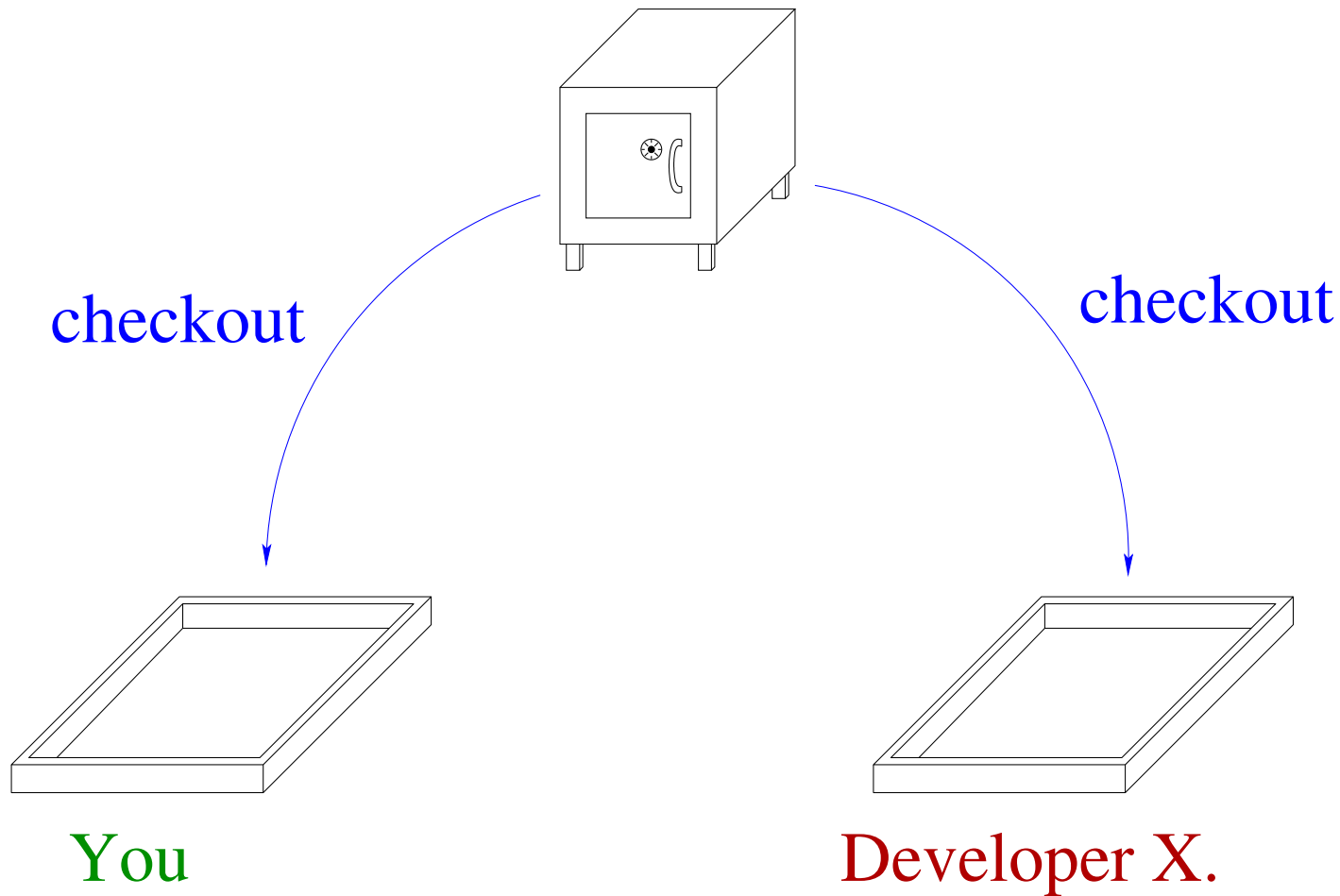
```
T README
```

```
T hello.c
```

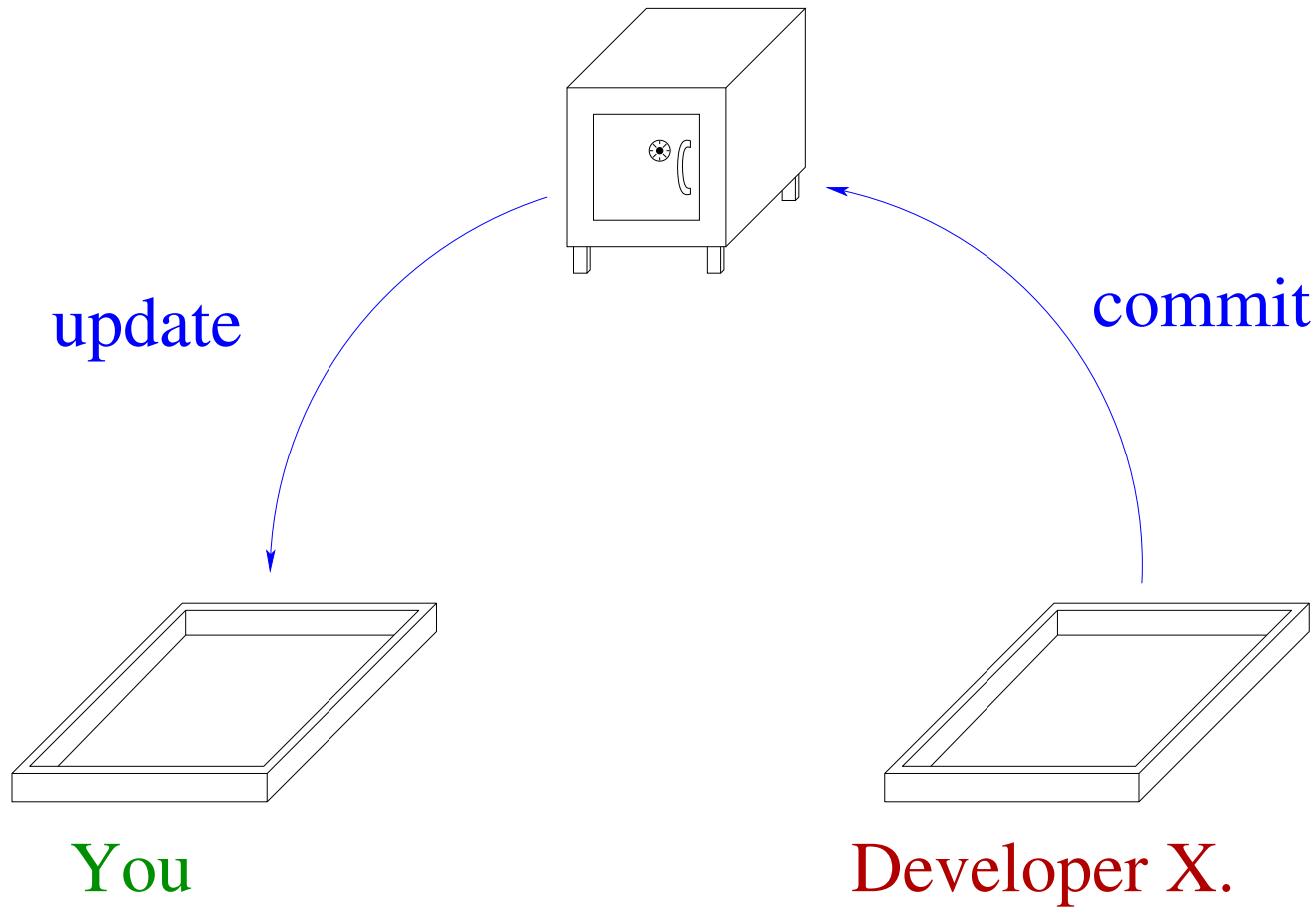
**Moral: use `cvs tag -c ...`**

# CVS Tutorial

## Collaborative Development



# CVS Tutorial



```
> cvs update
cvs update: Updating .
U ChangeLog
U README
```

# CVS Tutorial

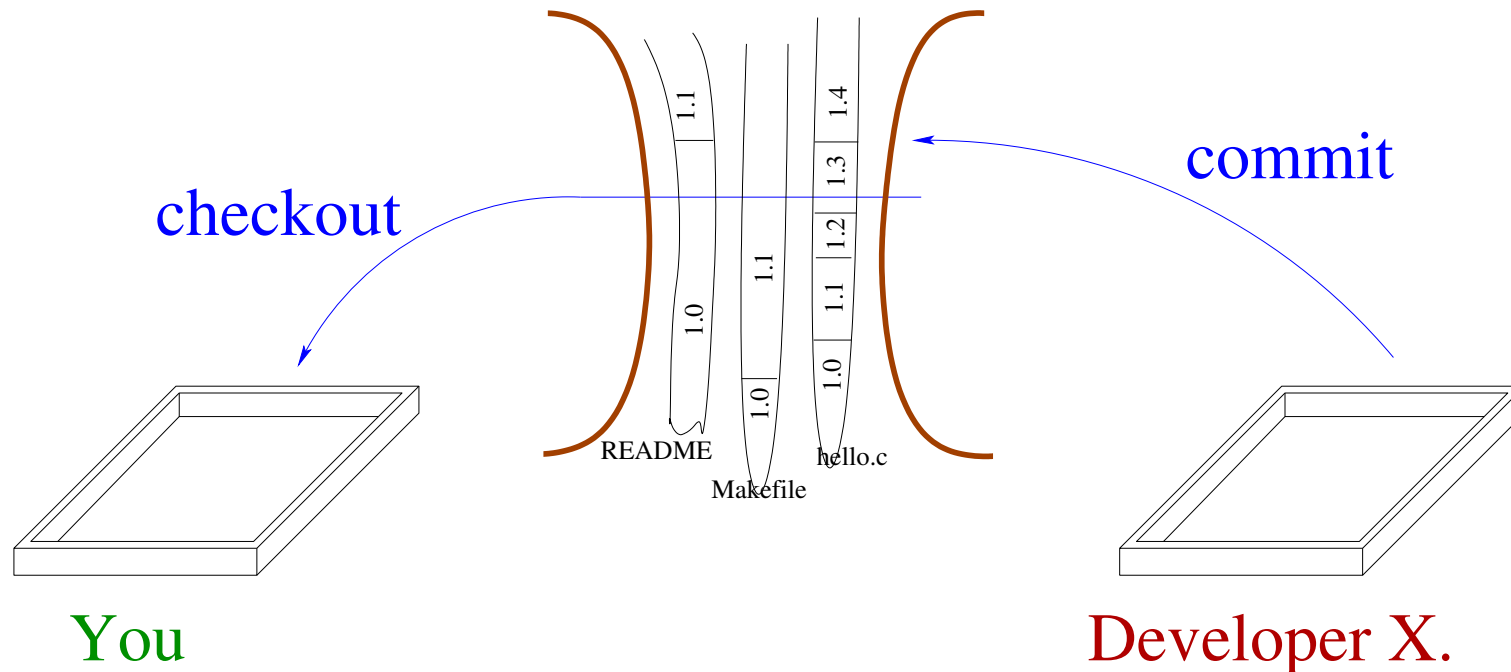
## > cvs commit

cvs commit: Examining .

cvs commit: Up-to-date check failed for 'README'

cvs commit: Up-to-date check failed for 'hello.c'

cvs [commit aborted]: correct above errors first!



**Moral: update before commit.**

# CVS Tutorial

## Update (revisited)

> **cv**s update

cv

s update: Updating .

U ChangeLog

M Makefile

RCS file: /software/examples/hello/hello.c,v

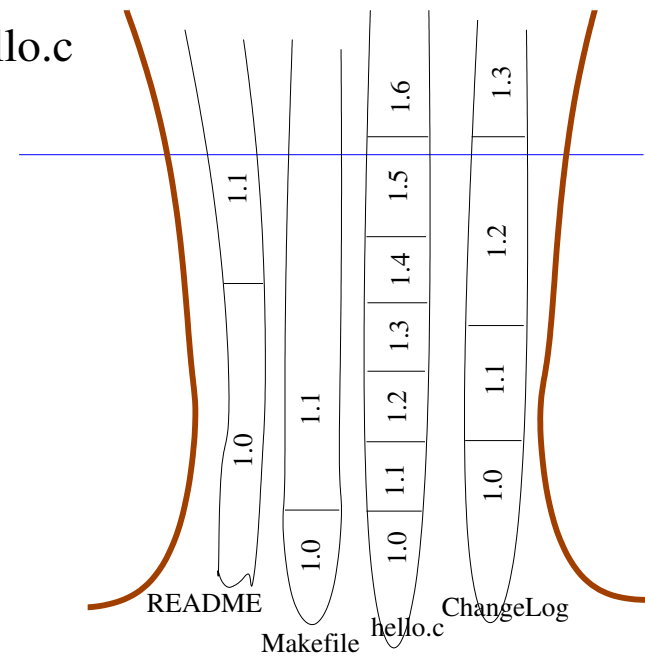
retrieving revision 1.5

retrieving revision 1.6

Merging differences between 1.5 and 1.6 into hello.c

M hello.c

? hello



# CVS Tutorial

## > cvs update

cvs update: Updating .

RCS file: /software/examples/hello/hello.c,v

retrieving revision 1.9

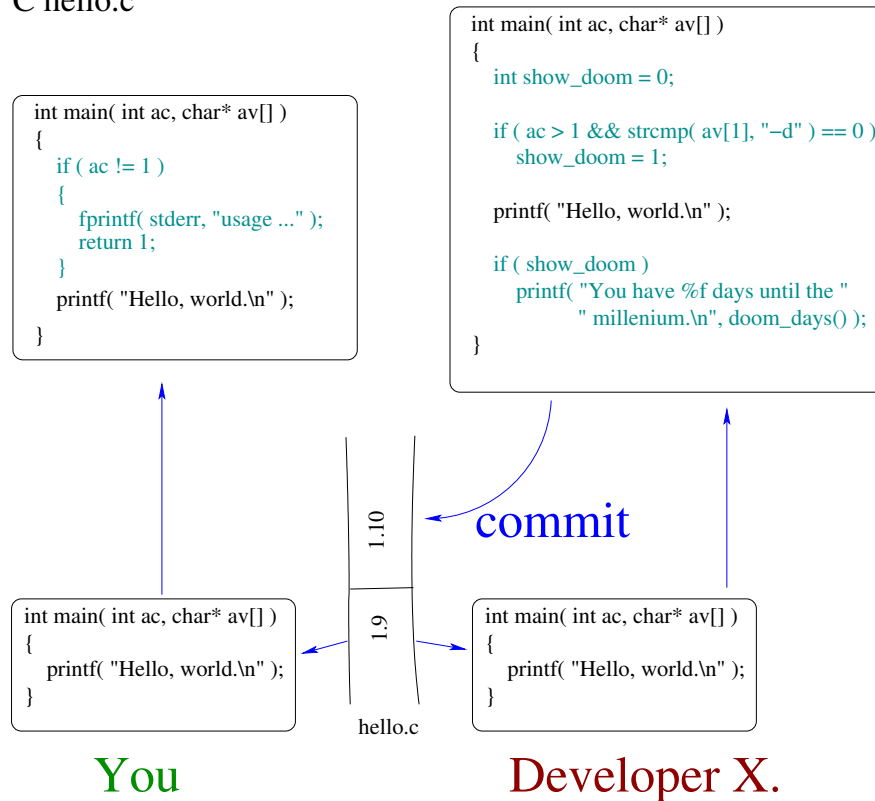
retrieving revision 1.10

Merging differences between 1.9 and 1.10 into hello.c

rcsmerge: warning: conflicts during merge

cvs update: conflicts found in hello.c

C hello.c



# CVS Tutorial

## Conflict Resolution

```
int main( int ac, char* av[] )
{
    if ( ac != 1 ) {
        fprintf( stderr, "usage ..." );
        return 1;
    }
    printf( "Hello, world.\n" );
}
```

hello.c before update

```
int main( int ac, char* av[] )
{
    int show_doom = 0;

    if ( ac > 1 && strcmp( av[1], "-d" ) == 0 )
        show_doom = 1;

    printf( "Hello, world.\n" );

    if ( show_doom )
        printf( "You have %f days until the "
              "millenium.\n", doom_days() );
}
```

revision 1.10

```
int main( int ac, char* av[] )
{
    <<<<<<< hello.c
    if ( ac != 1 ) {
        fprintf( stderr, "usage: hello\n" );
        return 1;
    }

    =====
    int show_doom = 0;

    if ( ac > 1 && strcmp( av[1], "-d" ) == 0 )
        show_doom = 1;

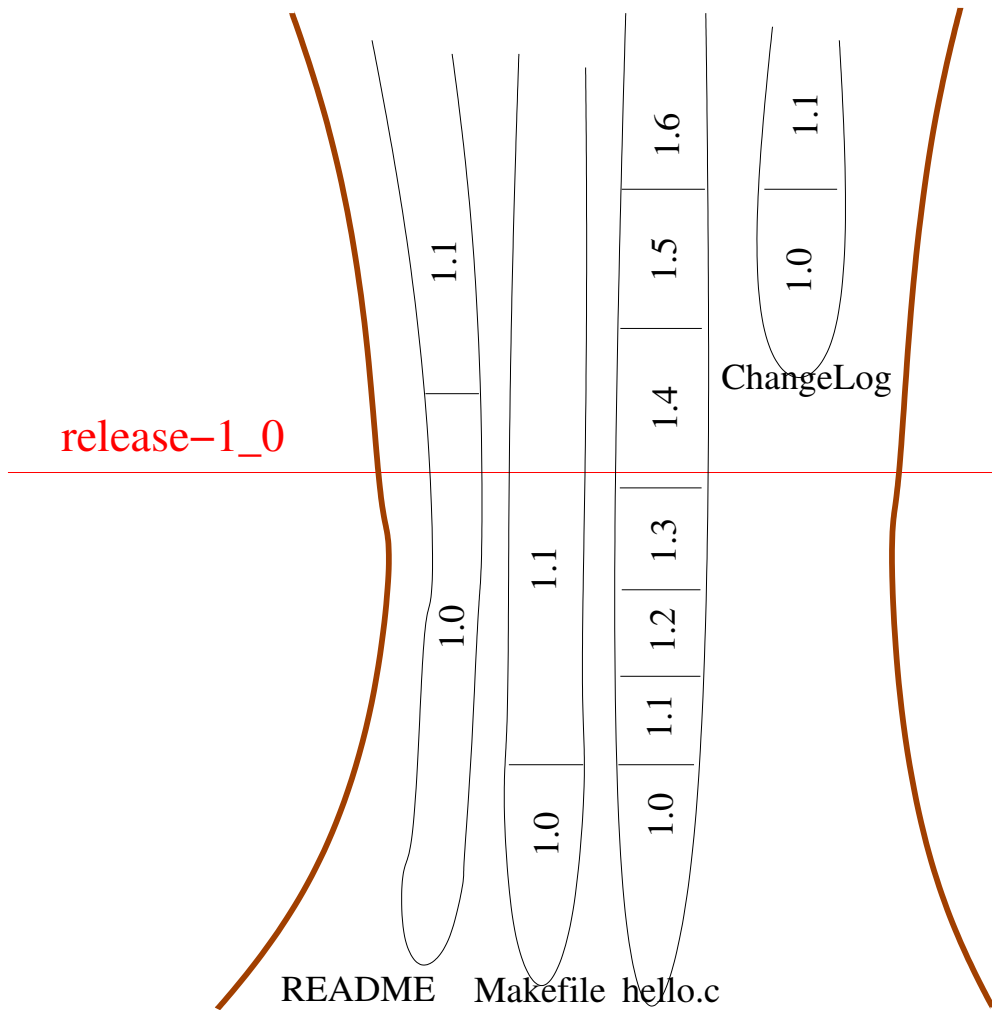
    >>>>>>> 1.10
    printf( "Hello, world.\n" );

    if ( show_doom )
        printf( "You have %f days until the "
              "millenium.\n", doom_days() );
}
```

hello.c after update

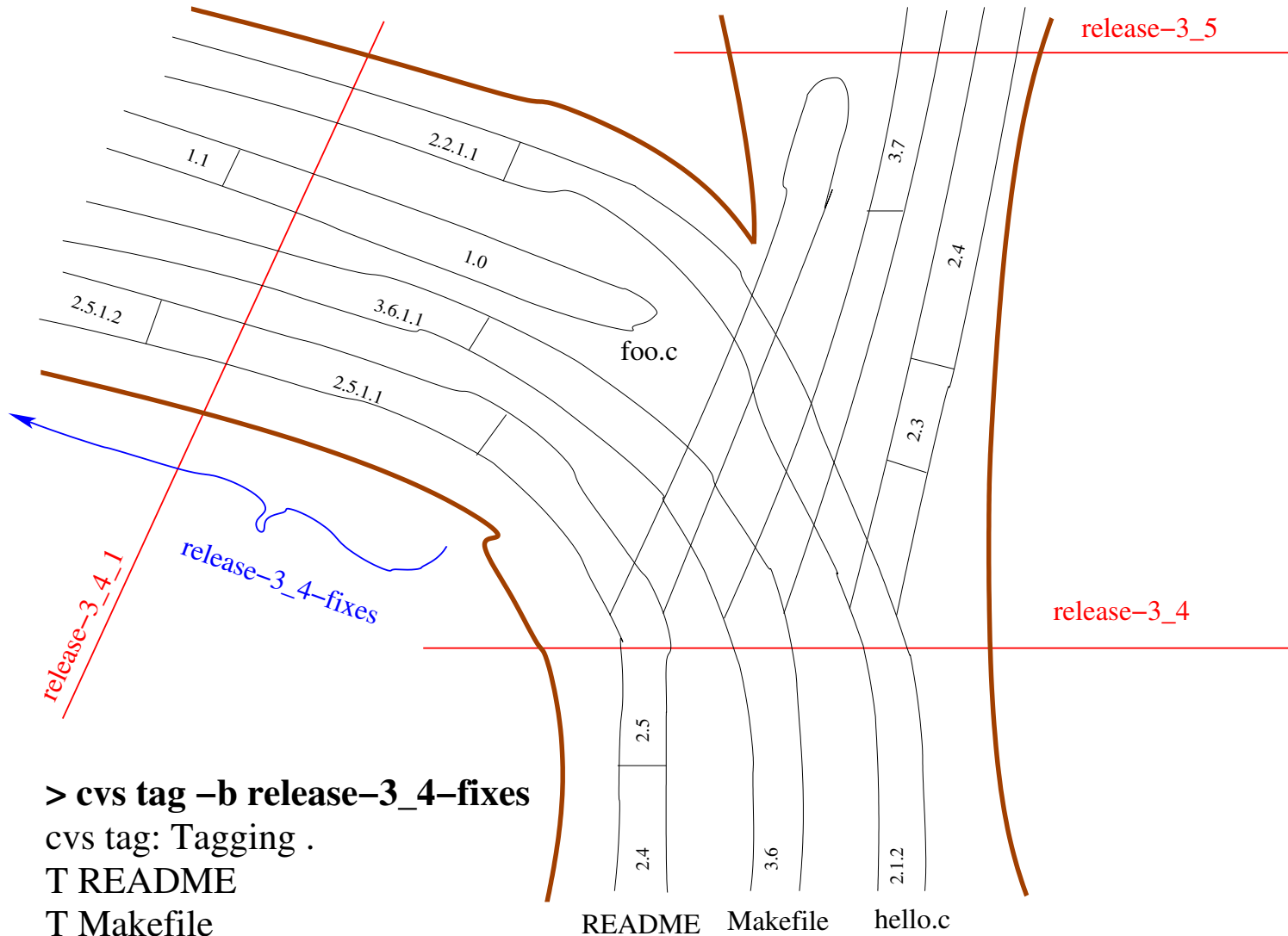
# CVS Tutorial

## Why Branch?





# CVS Tutorial



> **cvstag -b release-3\_4-fixes**

cvstag: Tagging .

T README

T Makefile

T hello.c



# CVS Tutorial

## Command Summary

`cv`s `-d repository` `init`  
= create a new repository

`cv`s `-d repository` `import` `module` `vendor-tag` `release-tag`  
= create a new module

`cv`s `-d repository` `checkout` [`-r tag`] `module`

`cv`s `update` [`-r tag`]

`cv`s `add` `file ...`

`cv`s `remove` `file ...`

`cv`s `commit`

`cv`s `tag -c` [`-b`] `tag`

`cv`s `log`

= list log messages, tags, etc

`cv`s `status`

= up-to-date, locally-modified, etc

`cv`s `diff -r rev1 -r rev2` `file`

= difference between specified revisions