

# Principles of Software Construction: Objects, Design, and Concurrency

Software engineering in practice

Teams, branch-based development, and workflows

Josh Bloch

**Charlie Garrod**



# Administrivia

- Homework 5 team sign-up deadline Thursday 11:59 p.m.
  - Team sizes, presentation slots...
- 2<sup>nd</sup> midterm exam "in class" on Thursday
  - Please have mobile phone or some other way to scan documents
  - Review session today 6-8 pm EDT: <https://cmu.zoom.us/j/343150293>
- Required reading due next Tuesday:
  - Java Concurrency in Practice, Sections 11.3 and 11.4
- Homework 5 frameworks discussion
- Online format...

# Key concepts from last Thursday

- API design principles, part 2

# Key design principle: Information hiding

- "When in doubt, leave it out."

# Minimize mutability

- Classes should be immutable unless there's a good reason to do otherwise
  - Advantages: simple, thread-safe, reusable
    - See `java.lang.String`
  - Disadvantage: separate object for each value
- Mutable objects require careful management of visibility and side effects
  - e.g. `Component.getSize()` returns a mutable `Dimension`
- Document mutability
  - Carefully describe state space

# Fail fast

- Report errors as soon as they are detectable
  - Check preconditions at the beginning of each method
  - Avoid dynamic type casts, run-time type-checking

```
// A Properties instance maps Strings to Strings
public class Properties extends Hashtable {
    public Object put(Object key, Object value);

    // Throws ClassCastException if this instance
    // contains any keys or values that are not Strings
    public void save(OutputStream out, String comments);
}
```

# Subtleties of information hiding

- Prevent subtle leaks of implementation details
  - Documentation
  - Lack of documentation
  - Implementation-specific return types
  - Implementation-specific exceptions
  - Output formats
  - `implements Serializable`

# Don't let your output become your de facto API

- Document the fact that output formats may evolve in the future
- Provide programmatic access to all data available in string form

```
org.omg.CORBA.MARSHAL: com.ibm.ws.pmi.server.DataDescriptor; IllegalAccessException minor code: 4942F23E comp
  at com.ibm.rmi.io.ValueHandlerImpl.readValue(ValueHandlerImpl.java:199)
  at com.ibm.rmi.iiop.CDRInputStream.read_value(CDRInputStream.java:1429)
  at com.ibm.rmi.io.ValueHandlerImpl.read_Array(ValueHandlerImpl.java:625)
  at com.ibm.rmi.io.ValueHandlerImpl.readValueInternal(ValueHandlerImpl.java:273)
  at com.ibm.rmi.io.ValueHandlerImpl.readValue(ValueHandlerImpl.java:189)
  at com.ibm.rmi.iiop.CDRInputStream.read_value(CDRInputStream.java:1429)
  at com.ibm.ejs.sm.beans._EJSRemoteStatelessPmiService_Tie._invoke(_EJSRemoteStatelessPmiService_Tie.ja
  at com.ibm.CORBA.iiop.ExtendedServerDelegate.dispatch(ExtendedServerDelegate.java:515)
  at com.ibm.CORBA.iiop.ORB.process(ORB.java:2377)
  at com.ibm.CORBA.iiop.OrbWorker.run(OrbWorker.java:186)
  at com.ibm.ejs.oa.pool.ThreadPool$PooledWorker.run(ThreadPool.java:104)
  at com.ibm.ws.util.CachedThread.run(ThreadPool.java:137)
```



# Don't let your output become your de facto API

- Document the fact that output formats may evolve in the future
- Provide programmatic access to all data available in string form

```
public class Throwable {  
    public void printStackTrace(PrintStream s);  
    public StackTraceElement[] getStackTrace(); // since 1.4  
}
```

```
public final class StackTraceElement {  
    public String getFileName();  
    public int getLineNumber();  
    public String getClassName();  
    public String getMethodName();  
    public boolean isNativeMethod();  
}
```

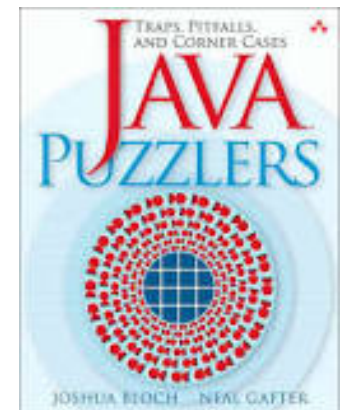
# Today: Toward software engineering in practice

- Two puzzlers
- Software engineering for teams
  - Challenges of working as a team
  - Tools and processes for teams
    - Branch-based development, et al.

# 1. “Time for a Change” (2002)

If you pay \$2.00 for a gasket that costs \$1.10, how much change do you get?

```
public class Change {  
    public static void main(String args[]) {  
        System.out.println(2.00 - 1.10);  
    }  
}
```



What does it print?

- (a) 0.9
- (b) 0.90
- (c) It varies
- (d) None of the above

```
public class Change {  
    public static void main(String args[]) {  
        System.out.println(2.00 - 1.10);  
    }  
}
```

What does it print?

(a) 0.9

(b) 0.90

(c) It varies

(d) None of the above: 0.89999999999999999999

Decimal values can't be represented exactly  
by `float` or `double`

# Another look

```
public class Change {  
    public static void main(String args[]) {  
        System.out.println(2.00 - 1.10);  
    }  
}
```

# How do you fix it?

// You could fix it this way...

```
import java.math.BigDecimal;
public class Change {
    public static void main(String args[]) {
        System.out.println(
            new BigDecimal("2.00").subtract(
                new BigDecimal("1.10")));
    }
}
```

Prints 0.90

// ...or you could fix it this way

```
public class Change {
    public static void main(String args[]) {
        System.out.println(200 - 110);
    }
}
```

Prints 90

# The moral

- Avoid `float` and `double` where exact answers are required
  - For example, when dealing with money
- Use `BigDecimal`, `int`, or `long` instead



## 2. “A Change is Gonna Come”



If you pay \$2.00 for a gasket that costs \$1.10, how much change do you get?

```
import java.math.BigDecimal;

public class Change {
    public static void main(String args[]) {
        BigDecimal payment = new BigDecimal(2.00);
        BigDecimal cost = new BigDecimal(1.10);
        System.out.println(payment.subtract(cost));
    }
}
```

# What does it print?

- (a) 0.9
- (b) 0.90
- (c) 0.89999999999999999999
- (d) None of the above

```
import java.math.BigDecimal;

public class Change {
    public static void main(String args[]) {
        BigDecimal payment = new BigDecimal(2.00);
        BigDecimal cost = new BigDecimal(1.10);
        System.out.println(payment.subtract(cost));
    }
}
```

What does it print?

(a) 0.9

(b) 0.90

(c) 0.899999999999999999999999

(d) None of the above:

0.8999999999999999999999991118215802998747  
6766109466552734375

We used the wrong `BigDecimal` constructor

## Another look

The spec says:

```
public BigDecimal(double val)
```

Translates a double into a BigDecimal which is the exact decimal representation of the double's binary floating-point value.

```
import java.math.BigDecimal;
```

```
public class Change {  
    public static void main(String args[]) {  
        BigDecimal payment = new BigDecimal(2.00);  
        BigDecimal cost = new BigDecimal(1.10);  
        System.out.println(payment.subtract(cost));  
    }  
}
```

# How do you fix it?

```
import java.math.BigDecimal;
```

Prints 0.90

```
public class Change {  
    public static void main(String args[]) {  
        BigDecimal payment = new BigDecimal("2.00");  
        BigDecimal cost = new BigDecimal("1.10");  
        System.out.println(payment.subtract(cost));  
    }  
}
```

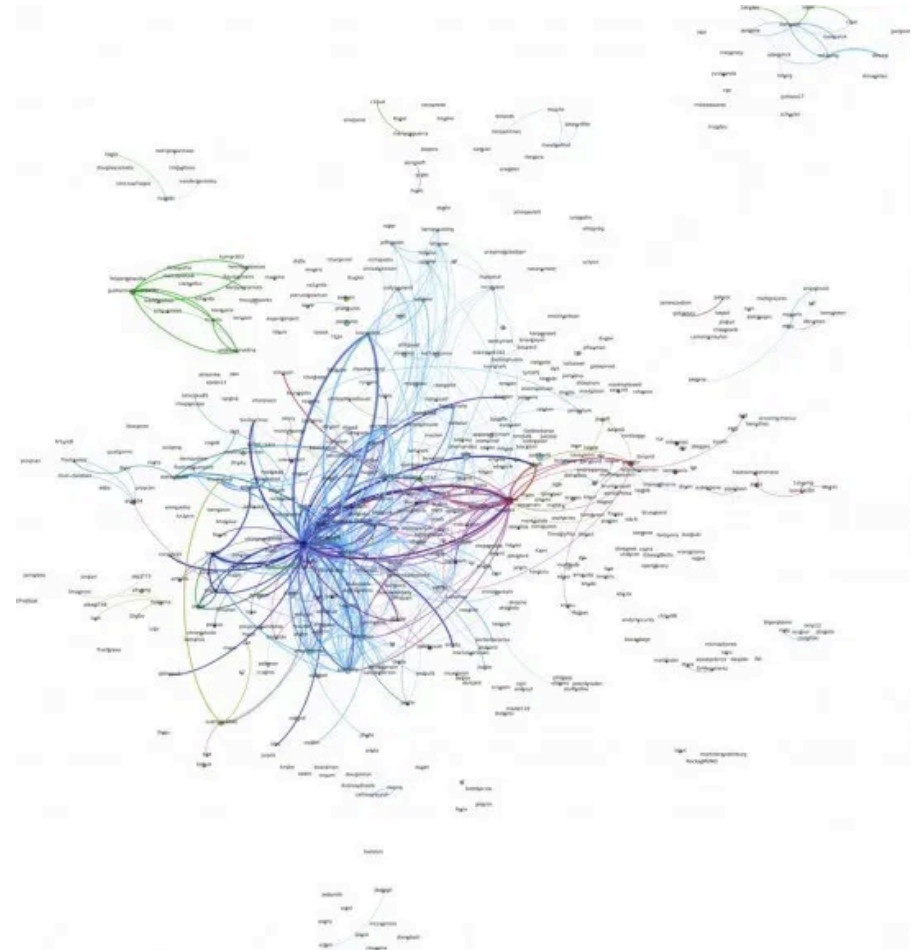
# The moral

- Use new `BigDecimal(String)`, not new `BigDecimal(double)`
- `BigDecimal.valueOf(double)` is better, but not perfect
  - Use it for non-constant values.
- For API designers
  - Make it easy to do the commonly correct thing
  - Make it hard to misuse
  - Make it possible to do exotic things

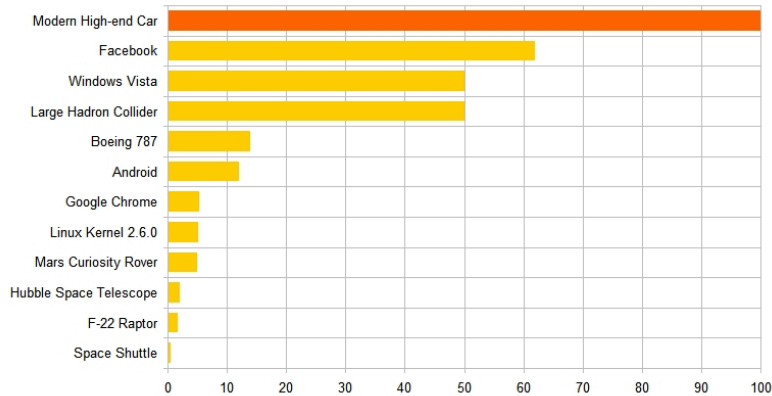
# Today: Toward software engineering in practice

- Two puzzlers
- Software engineering for teams
  - Challenges of working as a team
  - Tools and processes for teams
    - Branch-based development, et al.

# Software engineering is inherently collaborative



Software Size (million Lines of Code)



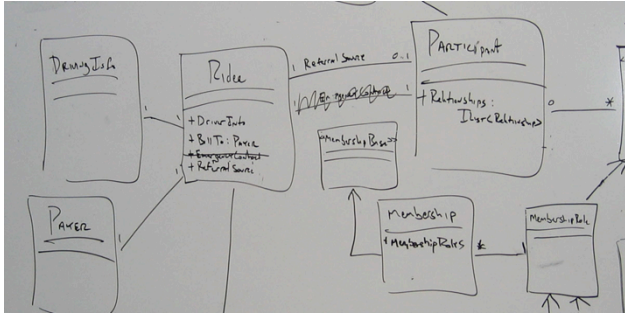


# Challenges of working as a team:

# Challenges of working as a team: **Aligning expectations**

- How does the team make decisions?
- How do you divide the work?
- Does the team share the same goals and incentives?
- What happens when work isn't completed as expected?
- When do team members like to work?
- What other commitments do your team members have?
- Where will you get the work done?
- ...

# Decide what to build, then design the API



## Basic Process:

- (1) Determine minimal feature set
- (2) Draw UML on the whiteboard.
- (3) Sketch out your API on paper
- (4) Write example code
- (5) Review
- (6) Repeat

```
// A collection of elements (root of the collection hierarchy)
public interface Collection<E> {

    // Ensures that collection contains o
    boolean add(E o);

    // Removes an instance of o from collection, if present
    boolean remove(Object o);

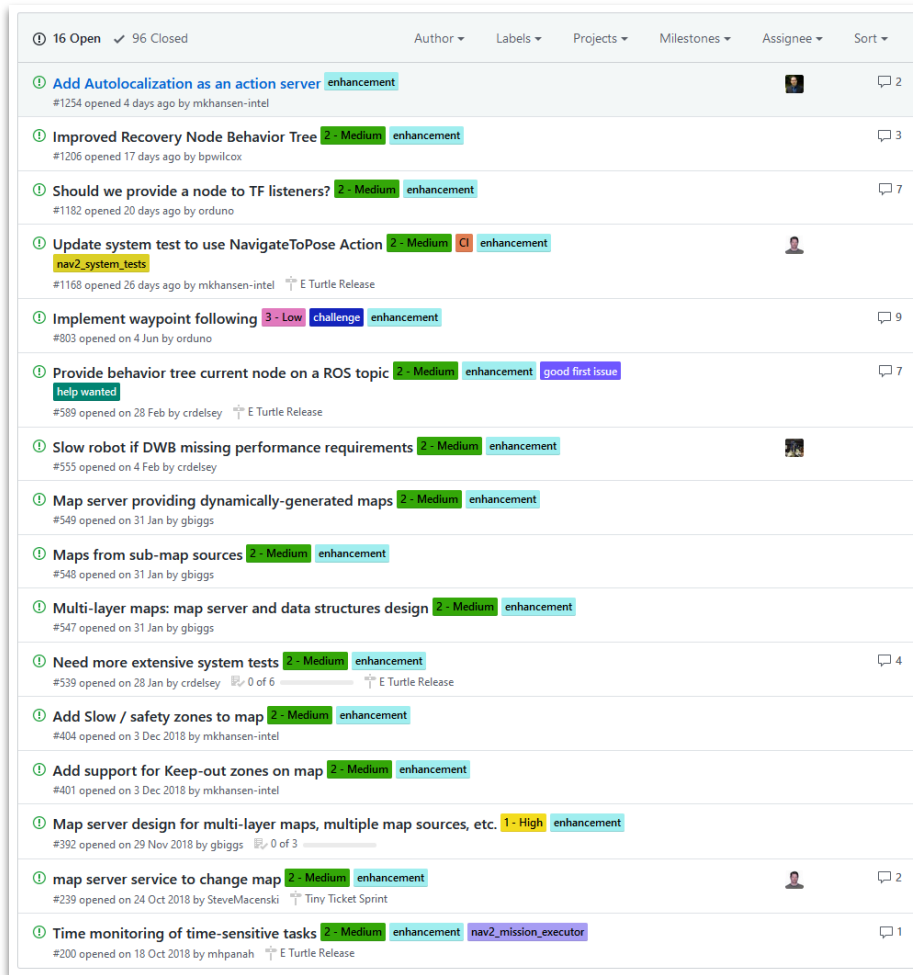
    // Returns true iff collection contains o
    boolean contains(Object o);

    // Returns number of elements in collection
    int size();

    // Returns true if collection is empty
    boolean isEmpty();

    ... // Remainder omitted
}
```

# Break up tasks into GitHub Issues

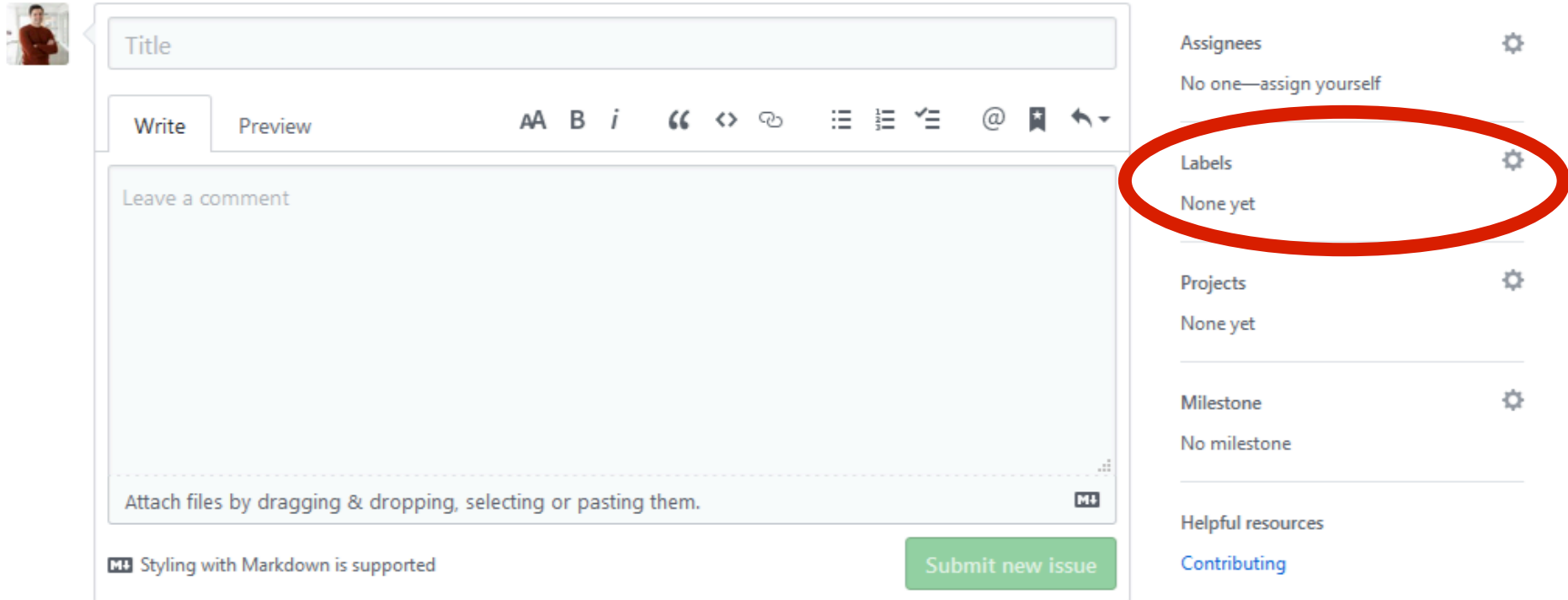


Issues can represent both tasks and bugs that need to be fixed.

Issues should be:

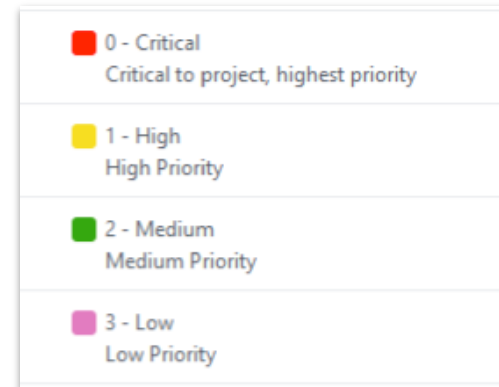
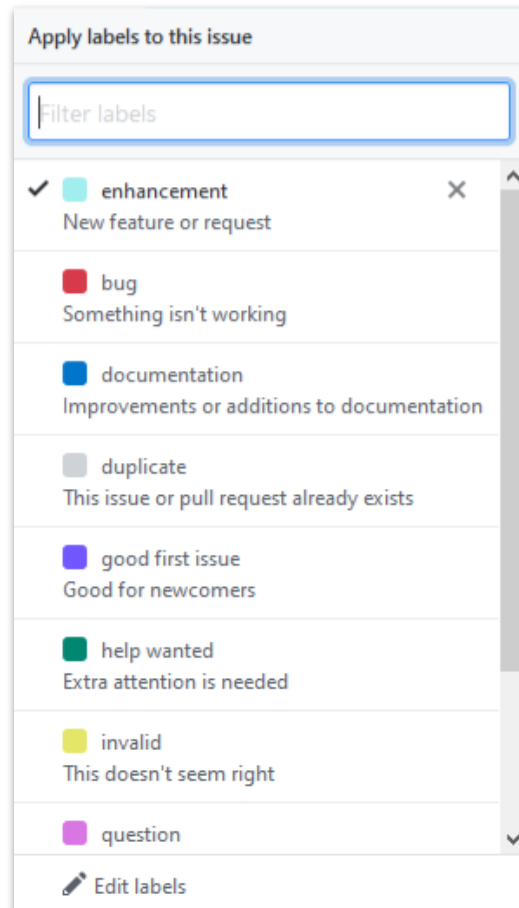
- a *reasonable* chunk of work
- focused and cohesive

# Break up tasks into GitHub Issues



The image shows the GitHub 'New Issue' form. On the left, there is a profile picture of a person. The form has a 'Title' input field at the top. Below it are 'Write' and 'Preview' tabs, followed by a rich text editor toolbar with icons for bold, italic, quote, code, link, list, and other formatting options. The main body of the form is a large text area with the placeholder text 'Leave a comment'. At the bottom of the text area, there is a note: 'Attach files by dragging & dropping, selecting or pasting them.' and a small 'M4' icon. Below the text area, there is a message: 'Styling with Markdown is supported' and a green 'Submit new issue' button. On the right side of the form, there are several sections: 'Assignees' with 'No one—assign yourself' and a gear icon; 'Labels' with 'None yet' and a gear icon, which is circled in red; 'Projects' with 'None yet' and a gear icon; and 'Milestone' with 'No milestone' and a gear icon. At the bottom right, there is a 'Helpful resources' section with a link to 'Contributing'.

# Use labels to indicate priority and differentiate bugs from features



# Consider using milestones (e.g., HW5a, HW5b)

Labels Milestones

4 Open ✓ 12 Closed Sort ▾

<h3>Tiny Ticket Sprint</h3> <p>Due by October 25, 2019 Last updated about 3 hours ago</p> <p>Lets get some tiny tickets out the week before we all head to China...<a href="#">(more)</a></p>	<div><div style="width: 16%;"></div></div> <p>16% complete 20 open 4 closed</p>
<h3>E Turtle Release</h3> <p>Due by November 22, 2019 Last updated about 4 hours ago</p>	<div><div style="width: 63%;"></div></div> <p>63% complete 29 open 50 closed</p>
<h3>September</h3> <p>⚠ Past due by 22 days Last updated 3 days ago</p>	<div><div style="width: 81%;"></div></div> <p>81% complete 5 open 22 closed</p>
<h3>F Turtle Release</h3> <p>No due date Last updated 4 days ago</p>	<div><div style="width: 0%;"></div></div> <p>0% complete 1 open 0 closed</p>

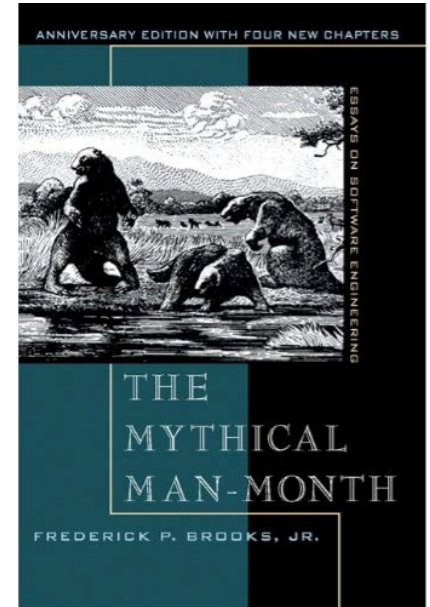
How does a large software project  
get to be one year late?



How does a large software project  
get to be one year late?

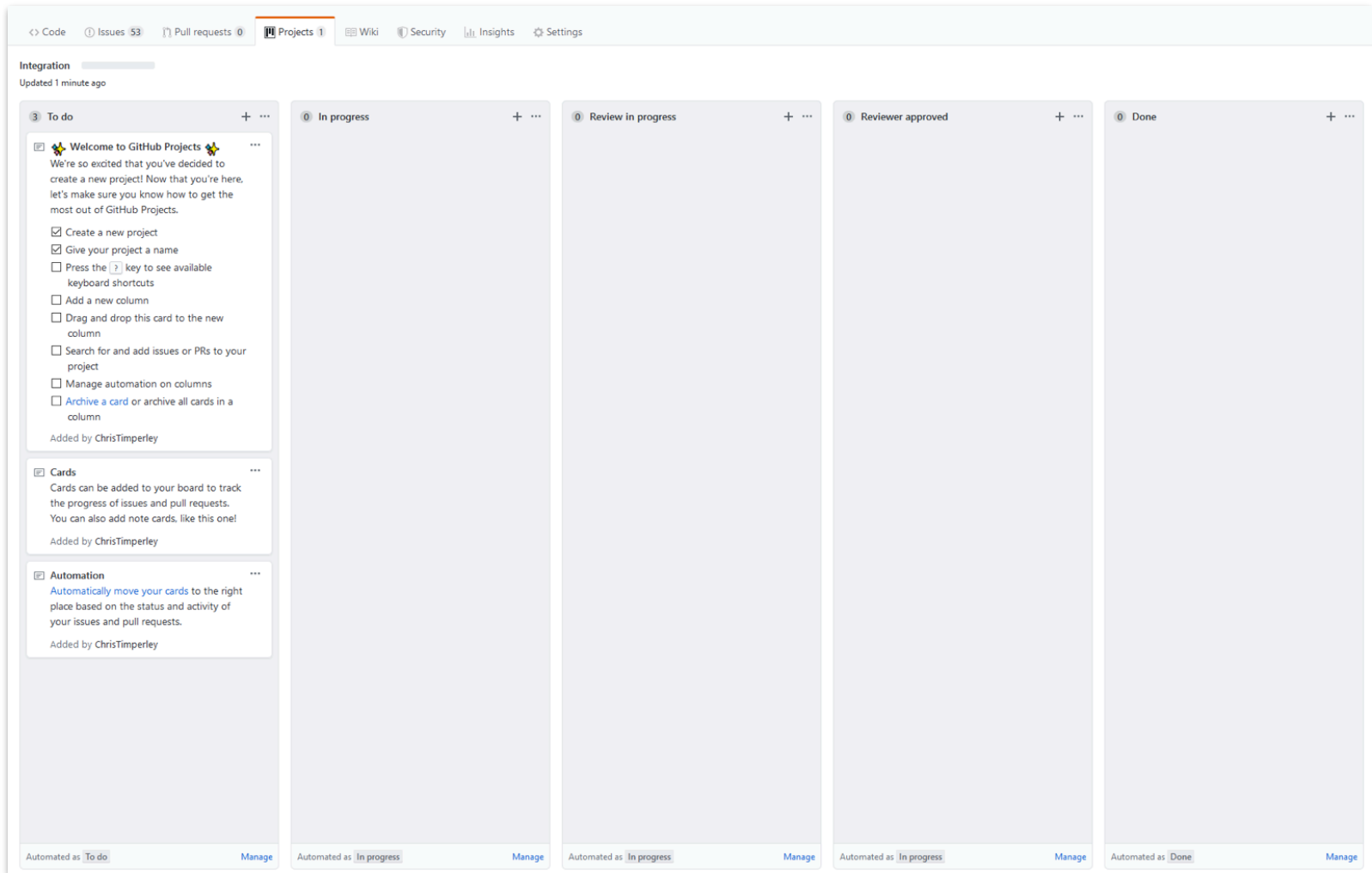
One day at a time.

— Fred Brooks, *The Mythical Man-Month*



[https://en.wikipedia.org/wiki/The\\_Mythical\\_Man-Month](https://en.wikipedia.org/wiki/The_Mythical_Man-Month)

# Use a simple Kanban board to measure progress

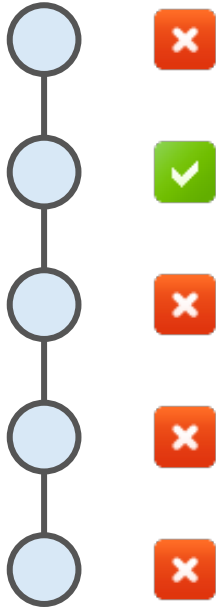


# Use a simple Kanban board to measure progress

The screenshot displays a Trello Kanban board for the project 'MagicBox - Development'. The board is organized into four columns: Backlog, To Do, In progress, and Done. Each column contains a list of tasks, each represented by a card with a title, a description, and various labels and tags. The 'Backlog' column has 7 cards, 'To Do' has 5, 'In progress' has 4, and 'Done' has 8. A search bar at the top right allows filtering cards. A '+ Add column' button is visible on the right side of the board. The board is automatically synced with Trello, Asana, and Jira.

Column	Task Title	Task Description	Labels
Backlog	Create Docker image for project	magicbox/ration-admin-server#10 opened by jflory?	enhancement, priority:low
	Throw errors for missing dependencies	magicbox/agg-logic-mobility#12 opened by jflory?	enhancement, help wanted
	Curling results in empty response, server crashes	magicbox/ration-admin-server#5 opened by jflory?	bug, help wanted, priority:crit
	Serve aggregations by admin level	magicbox/open-api#22 opened by mikelab	
	Base API in Swagger Spec Has Extra /	magicbox/open-api#28 opened by sherrida	
	Use arguments to overwrite default configurations	magicbox/agg-logic-mobility#24 opened by carlosdiaz	enhancement
	Return hasProbsData in API response for schools	magicbox/open-api#24 opened by ayana217	
expired token warning gets overridden by \$placeholder library and return html instead of json	magicbox/open-api#18 opened by mikelab		
To Do	Colombia shapello (mpis.json) should be fetched from api rather than be required in react app.	magicbox/maps2#6 opened by mikelab	
	Fix "file_loader" issue in react-app/src/public/lib.js	magicbox/maps#50 opened by jflory?	bug
	Comment and refactor code in action-select-country.js	magicbox/maps#40 opened by mikelab	
	Add disclaimer popup on load	magicbox/maps#39 opened by mikelab	
	Reconfigure login code to work in production	magicbox/maps#38 opened by mikelab	
In progress	Add contributing guidelines placeholder (closes #22)	magicbox/open-api#42 opened by jflory?	enhancement
	Determine where to begin writing tests and refactoring	magicbox/maps#34 opened by mikelab	
	Map relationships between apps, repos, and azure resources.	magicbox#15 opened by mikelab	in progress, internal organization, priority:crit
	Mobility: Map should display data for mobility being displayed	magicbox/maps#32 opened by mikelab	
Done	Resume test suite (run-tests.sh), add comments	magicbox/open-api#41 opened by jflory?	enhancement
	Refactor WebglLayer.js	magicbox/maps#31 opened by jflory?	bug
	Create production/staging/idev deployment slots for Mobility Mapping	magicbox#16 opened by mikelab	enhancement, internal organization, priority:crit
	Set up automatic CI testing for new pull requests	magicbox#6 opened by jflory?	enhancement, priority:high
	Add loading spinner to mobility map	magicbox/maps#17 opened by mikelab	
	Add spinner mobility	magicbox/maps#4 opened by carlosdiaz	enhancement
	Add CodeClimate maintainability + test coverage checks to CI	magicbox/open-api#40 opened by jflory?	enhancement
	Add TravisCI and CodeClimate tests		

# Single-branch development doesn't scale to teams



Master

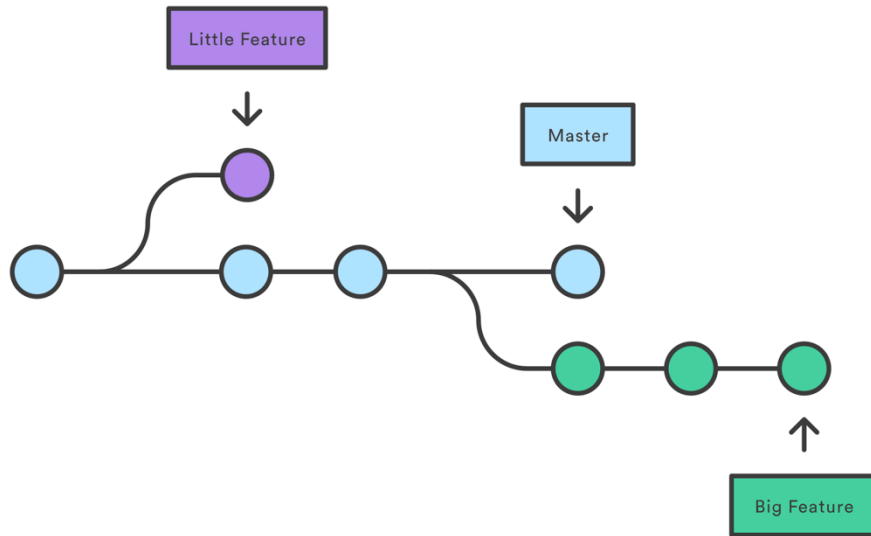
```
merge_conflict -- bash -- 80x24
bash python bash bash
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
dhcp-10-101-250-155:merge_conflict SUYEONSON$ git add index.html
dhcp-10-101-250-155:merge_conflict SUYEONSON$ git commit -m 'tryna make a merge conflict'
[master ee28024] tryna make a merge conflict
 1 file changed, 1 deletion(-)
dhcp-10-101-250-155:merge_conflict SUYEONSON$ git push
To git@github.com:suyeonson/merge_conflict.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:suyeonson/merge_conflict.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
dhcp-10-101-250-155:merge_conflict SUYEONSON$
```

build failure

# Use simple **branch-based development**



Commits on Oct 20, 2019

Added file checking methods to FileSystem (#28) ...

ChrisTimperley committed yesterday ✓

Verified

da32e4a



Commits on Oct 19, 2019

Implemented basic filesystem API (#27) ...

ChrisTimperley committed 2 days ago ✓

Verified

73d331e



Added workaround for shell calls without both stdout and stderr (#26) ...

ChrisTimperley committed 2 days ago ✓

Verified

06aa050



Added Container class for holding Docker container details (#24) ...

ChrisTimperley committed 3 days ago ✓

Verified

05c61e8



Commits on Oct 13, 2019

Added DockerDaemon for maintaining connections to daemon (fixes #21) (...)

ChrisTimperley committed 8 days ago ✓

Verified

79ad8e7



Added environ method to Shell (#20) ...

ChrisTimperley committed 9 days ago ✓

Verified

4494af4



Added basic popen to shell (fixes #6) (#19) ...

ChrisTimperley committed 9 days ago ✓

Verified

c7f9374



Add encoding and text parameters to Shell commands (fixes #9) (#17) ...

ChrisTimperley committed 9 days ago ✓

Verified

cef114c



## Create a new branch for each feature.

- allows parallel development
- no dealing with half-finished code
- no merge conflicts!

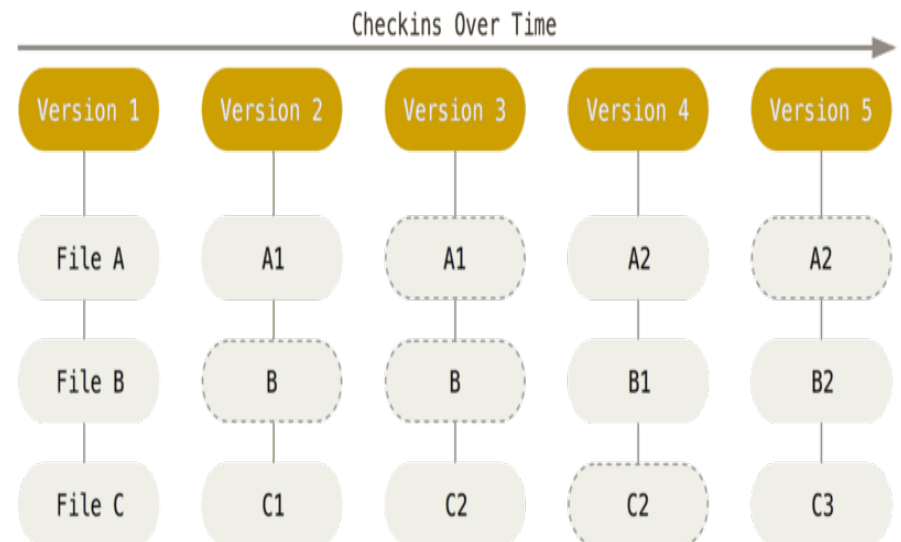
Every commit to “master” should pass your CI checks.

build

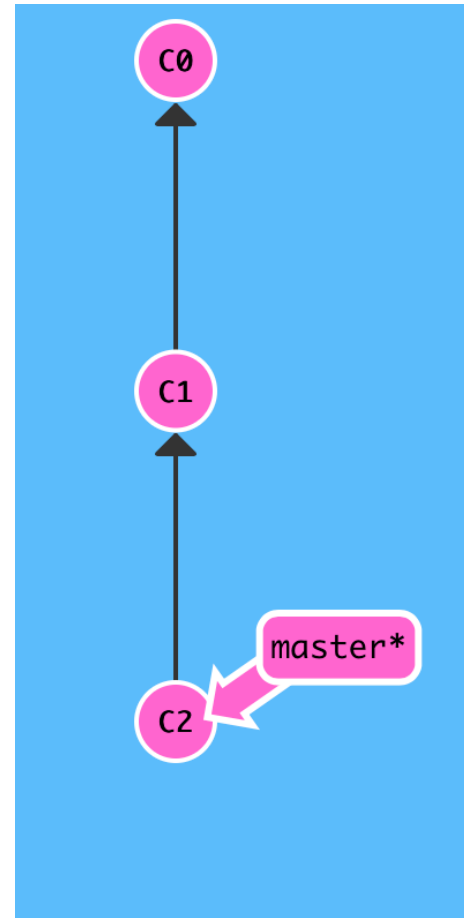
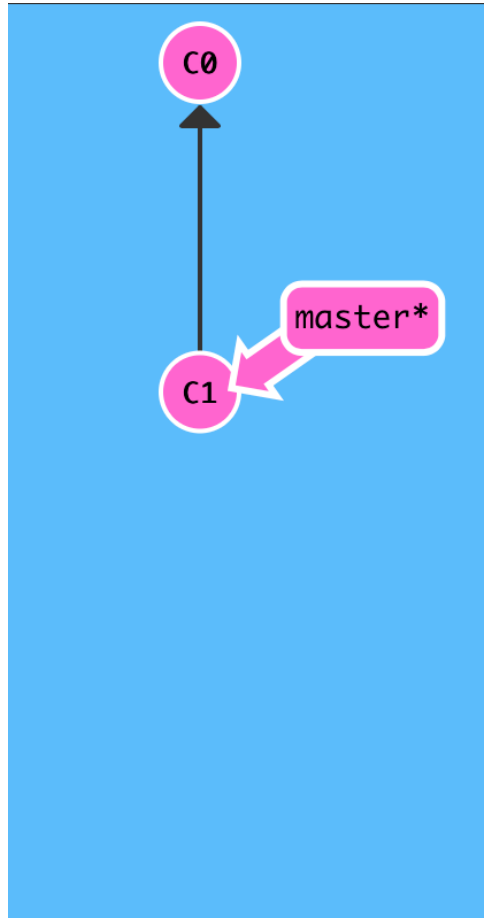
passing

# Git, practically

- Git stores each version as a snapshot
- If files have not changed, only a link to the previous file is stored
- Each version is referred by the SHA-1 hash of the contents

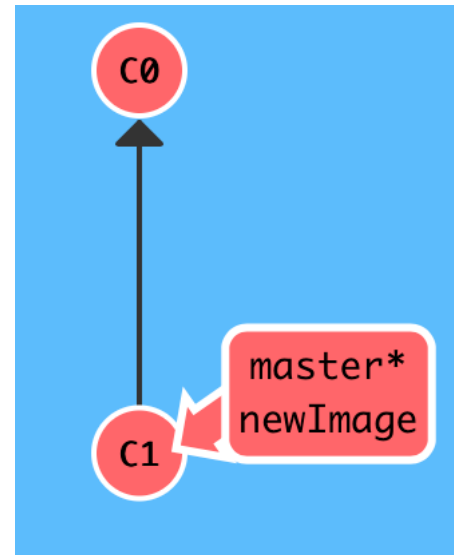
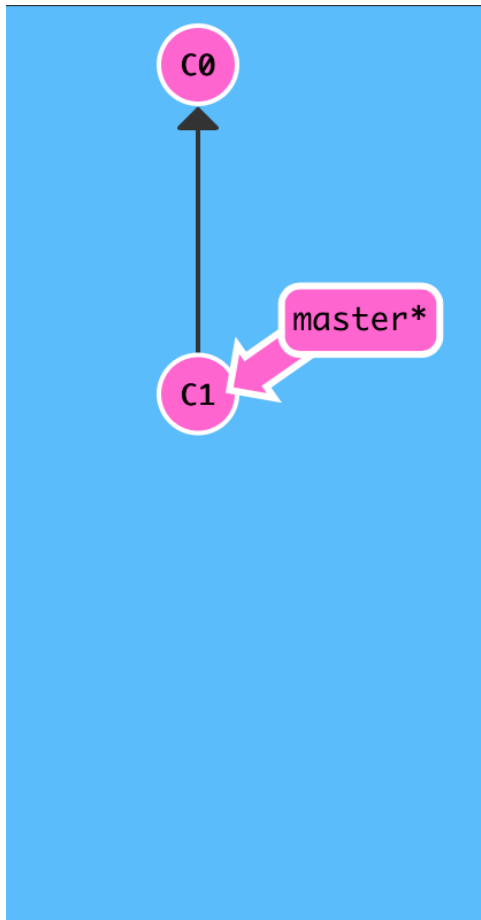


# git commit



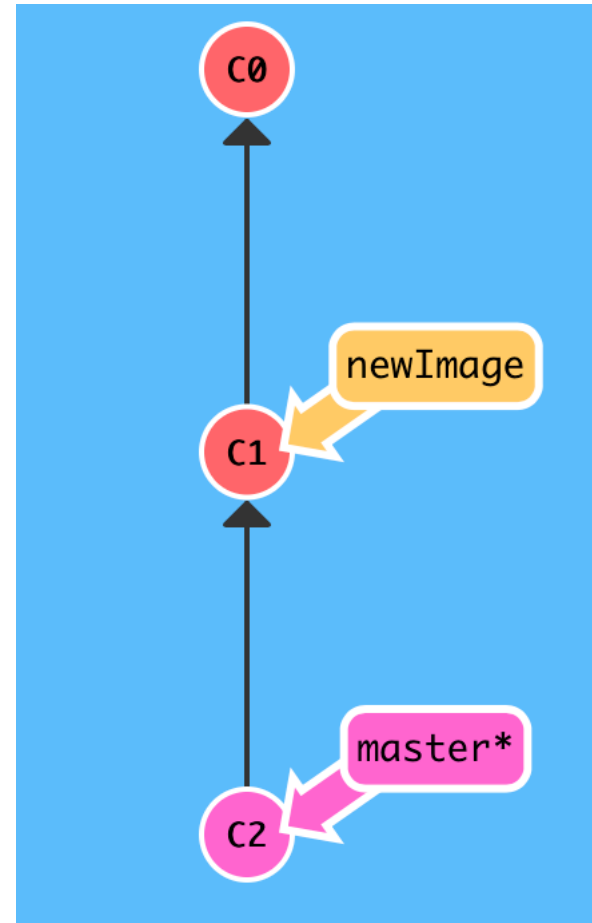
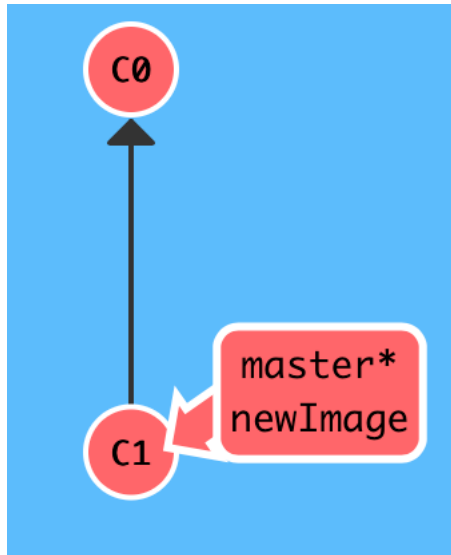
Graphics by <https://learngitbranching.js.org>

# git branch newImage

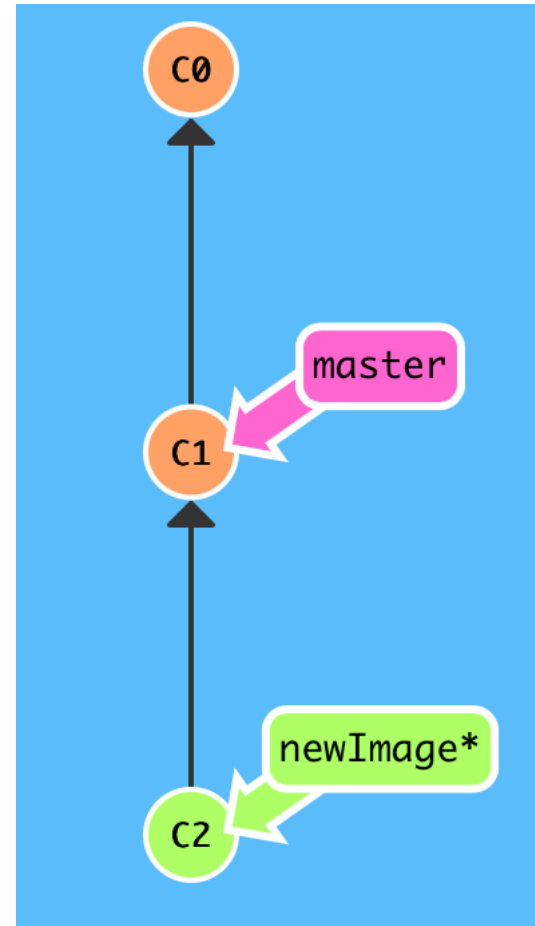
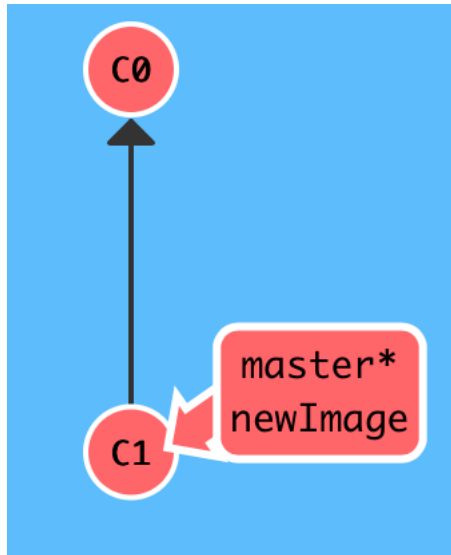




# git commit



git checkout newImage; git commit



# Summary

- Identify and discuss risks within your team
  - Get to know your teammates, and agree on your process
- Use standard tools to improve your process