

# YAHOO!

## Mobile App Continuous Delivery

PRESENTED BY **QC Li** | September 02, 2015

# About me

QC Li

Yahoo! Mobile App Engineer

qcli @ yahoo-inc . com



# Outline

- Continuous Delivery
- Continuous Delivery Pipeline
- Mobile App Continuous Delivery Pipeline

# Continuous Delivery



CD



# Continuous Delivery



Continuous Delivery (CD) is a software engineering approach in which teams keep producing valuable software in short cycles and ensure that the software can be reliably released at any time.

— Wikipedia

# Continuous Delivery

What we want is ...

- Low Cycle Time

- Cycle time: The time from deciding that you need to make a change to having it in production.

- High Quality

# Continuous Delivery

Automation almost everything

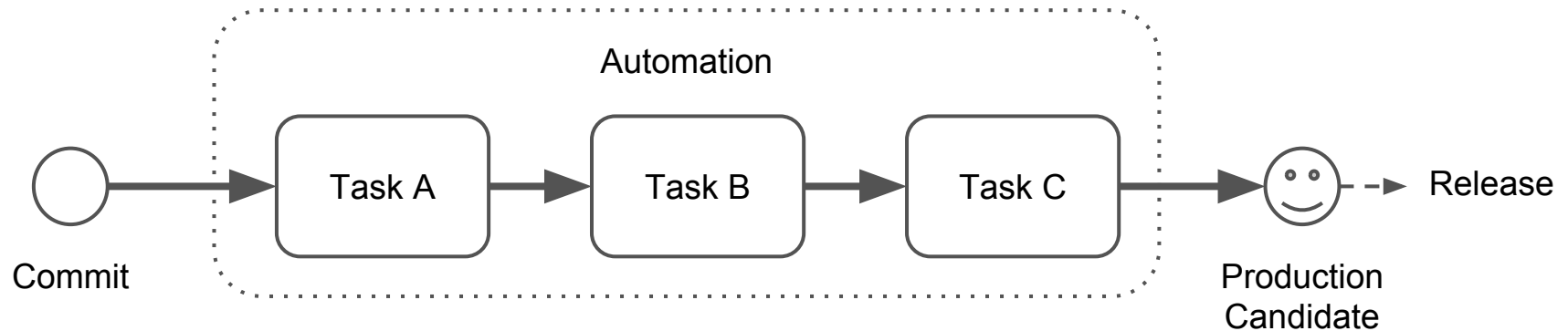
# Continuous Delivery

Commit to Production without human intervention

# Continuous Delivery Pipeline

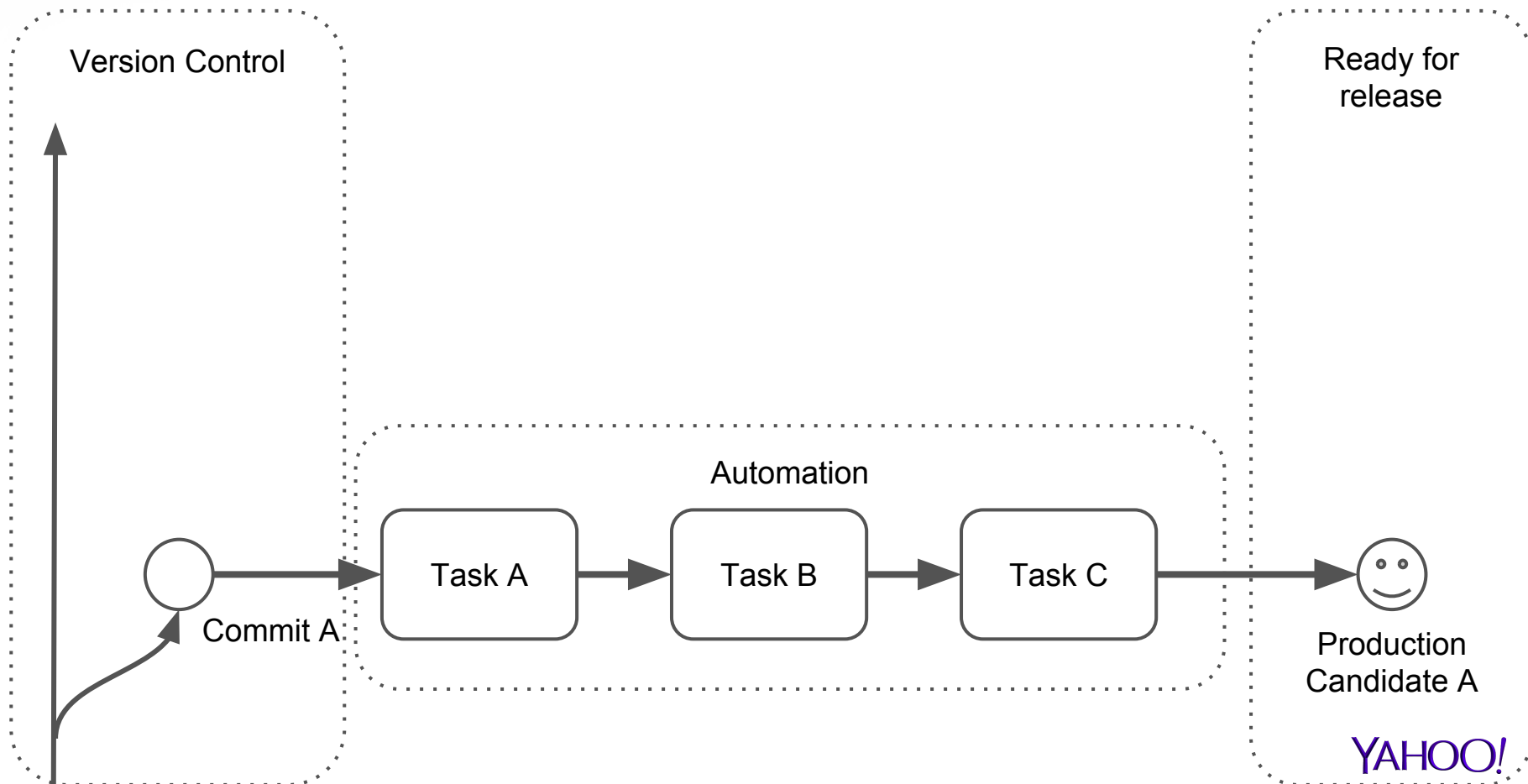
# Continuous Delivery Pipeline

- Workflow of tasks
- Automatically delivers code from commit to production



# Continuous Delivery Pipeline

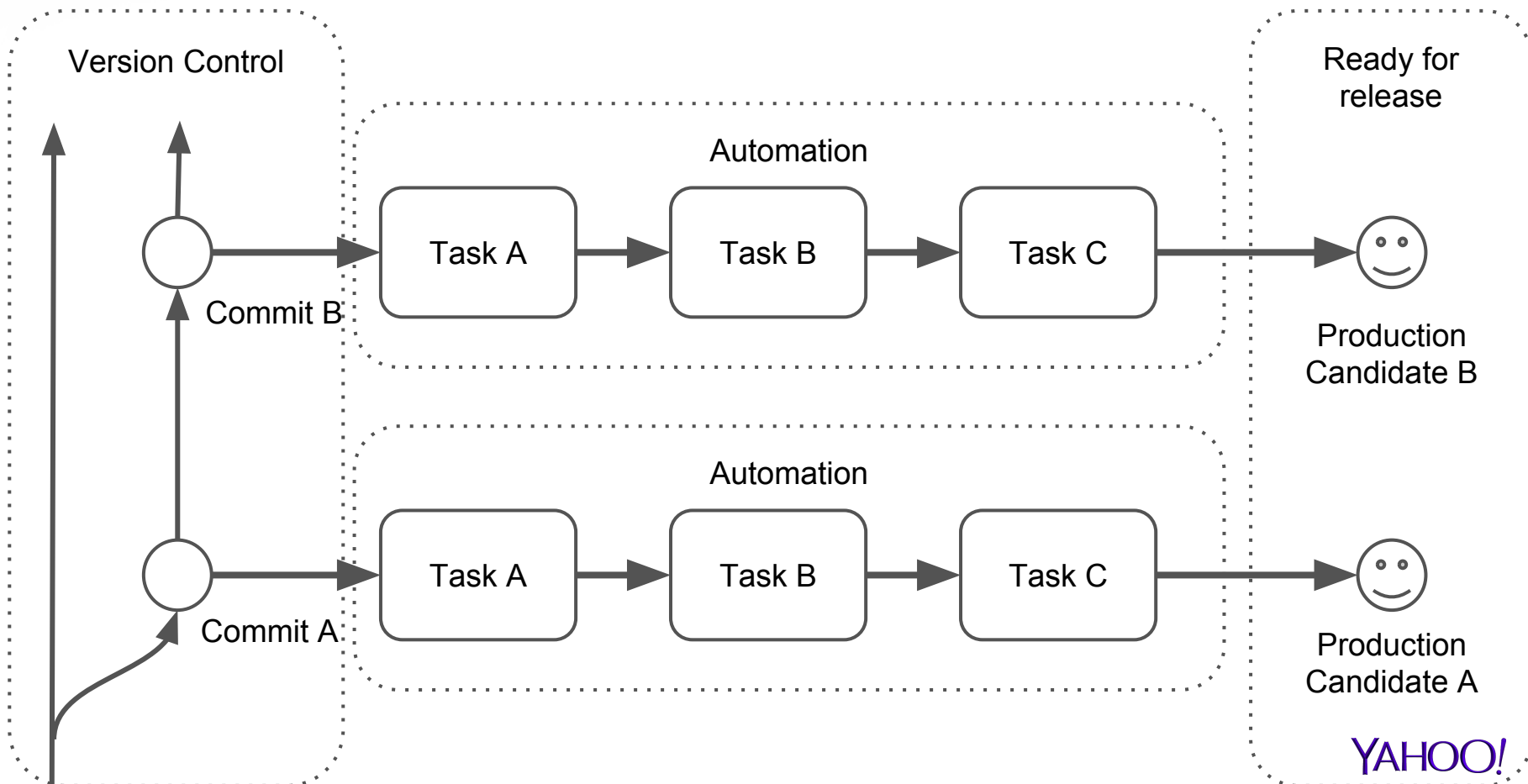
## ■ Keep everything in Version Control





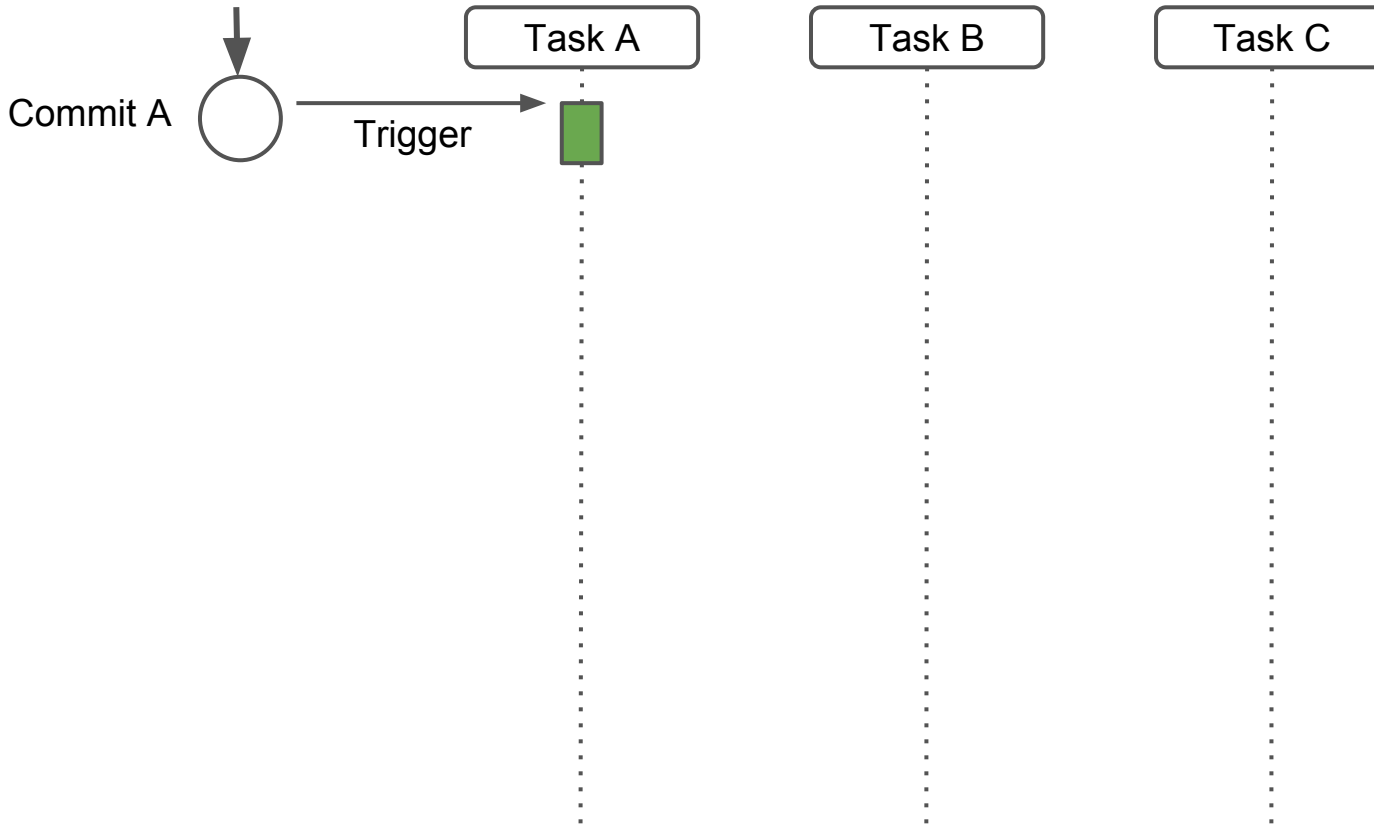
# Continuous Delivery Pipeline

## ■ Keep everything in Version Control



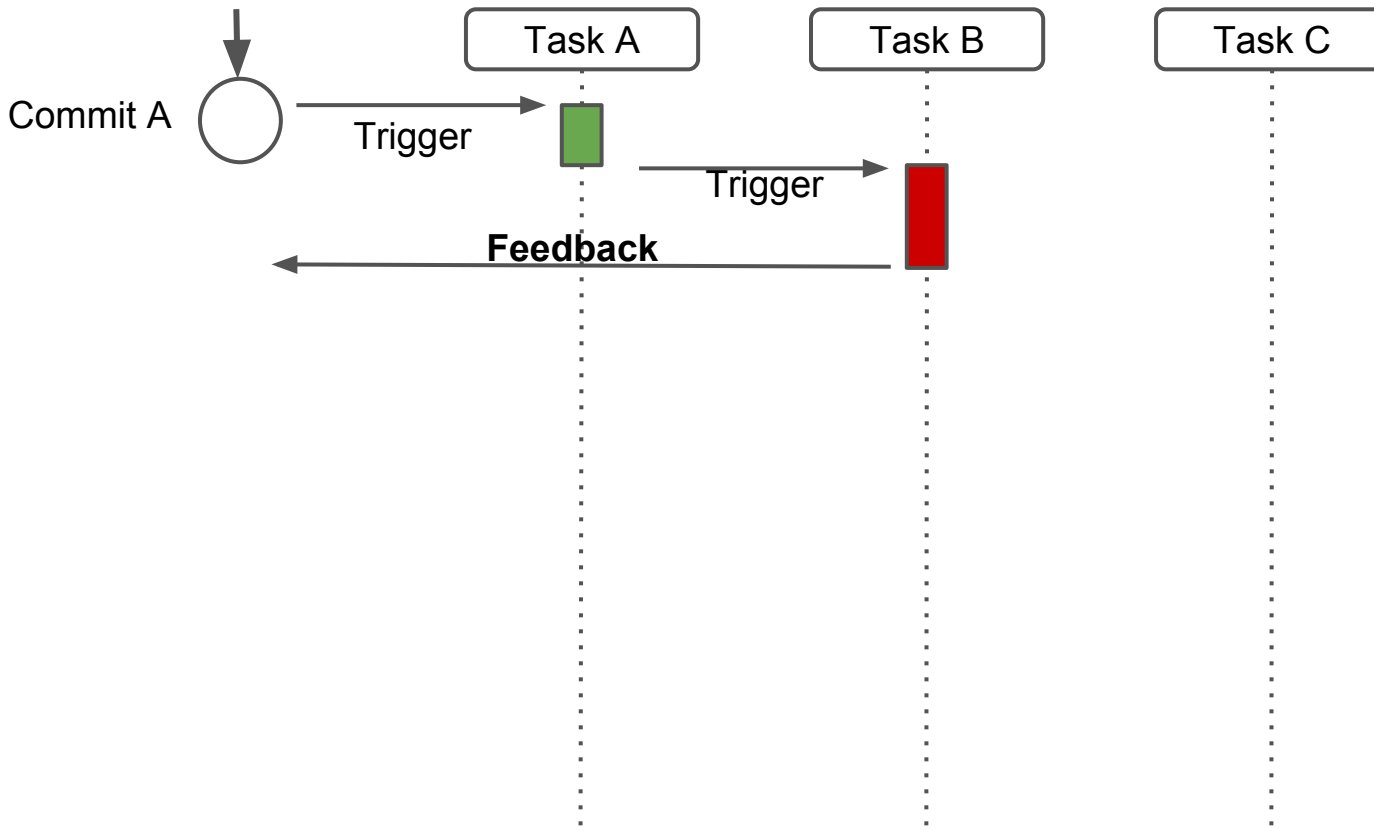
# Continuous Delivery Pipeline

- Each change should propagate through the pipeline instantly



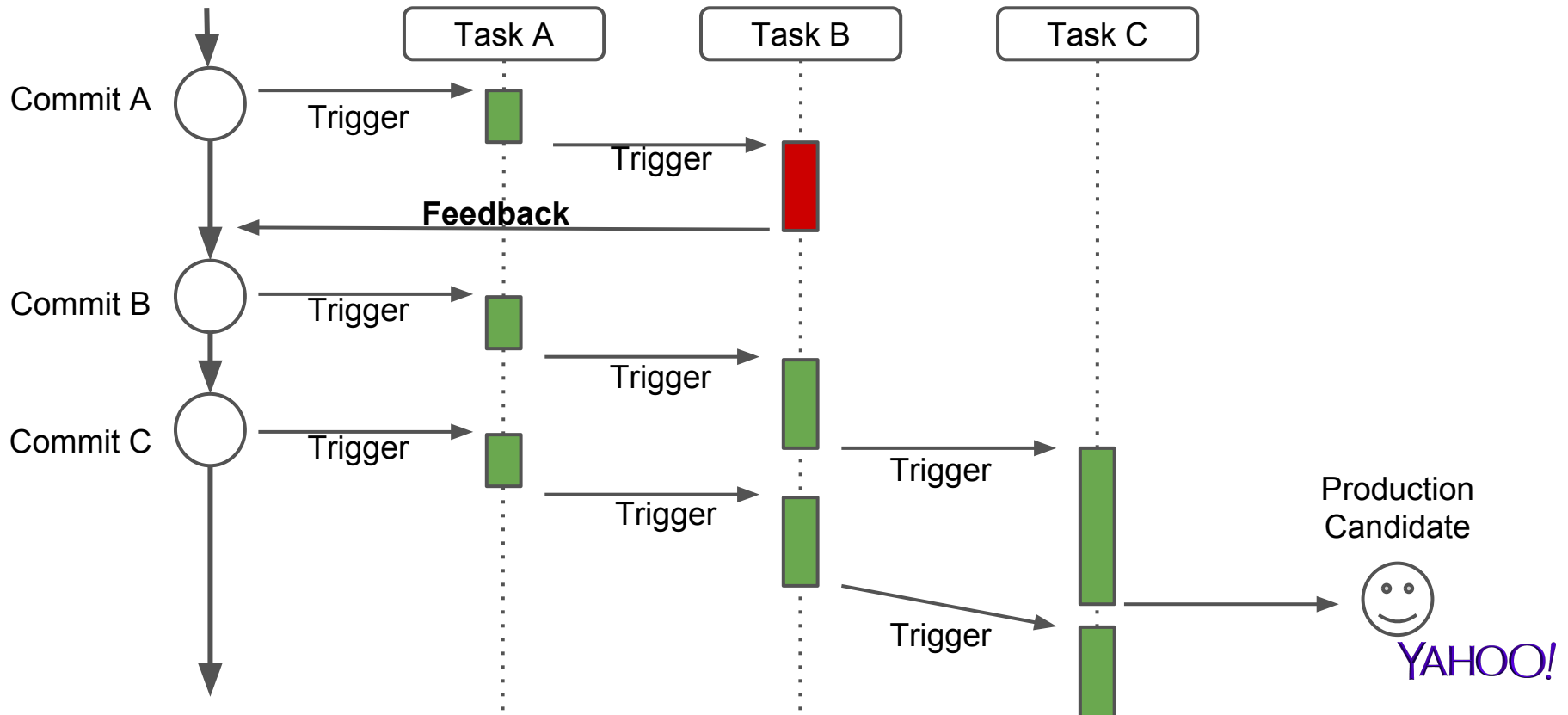
# Continuous Delivery Pipeline

- If any part of the pipeline fail, stop the line



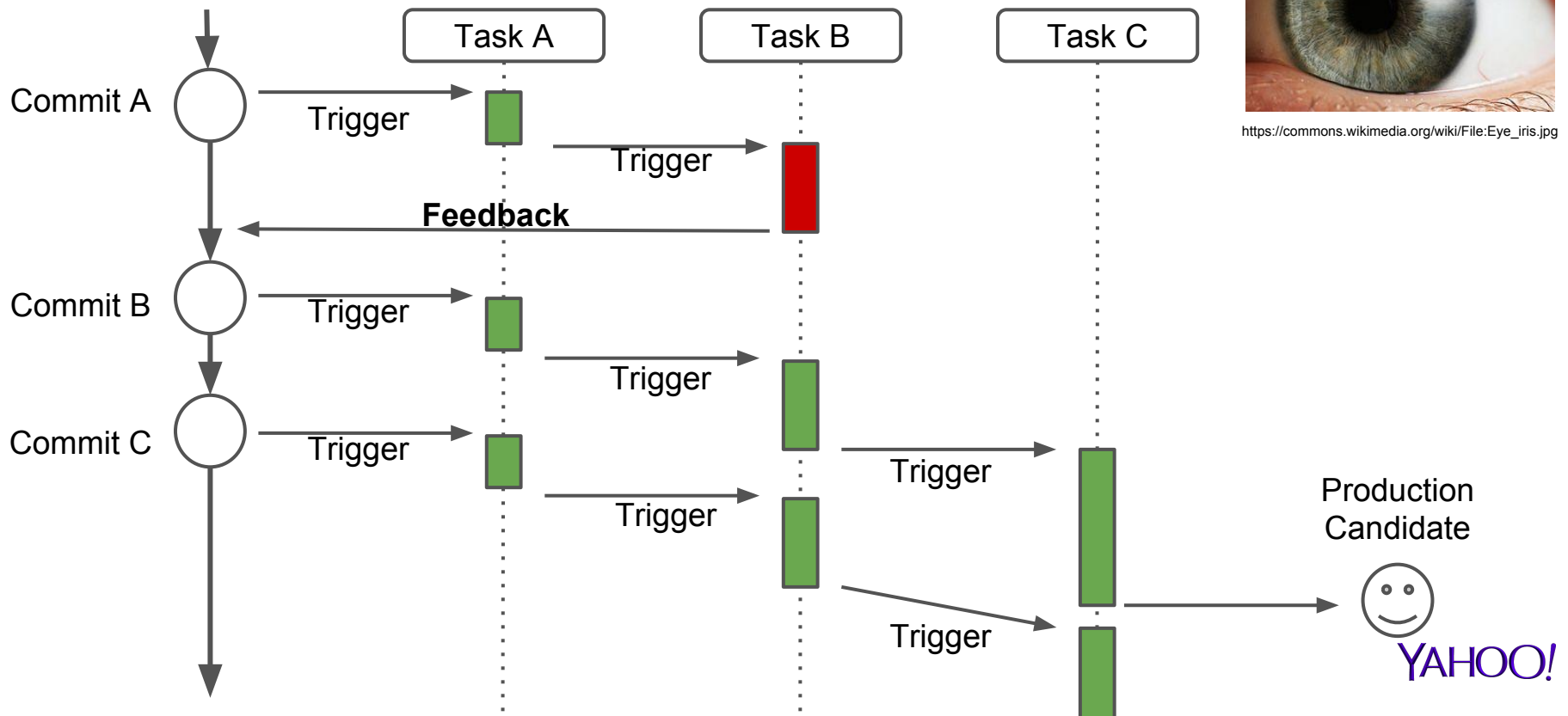
# Continuous Delivery Pipeline

- If any part of the pipeline fail, stop the line



# Continuous Delivery Pipeline

- Let the state of pipeline visible



# Continuous Delivery Pipeline

- Let the state of pipeline visible
  - Dashboard

Search all repositories

yahoo / fluxible build passing

My Repositories +

- ✓ yahoo/ember-intl #964  
Duration: 7 min 51 sec  
Finished: about 2 hours ago
- ✓ yahoo/fluxible-addons-react #14  
Duration: 3 min 6 sec  
Finished: about 13 hours ago
- ✓ yahoo/subscribe-ui-event #55  
Duration: 7 min 56 sec  
Finished: about 15 hours ago
- ✓ yahoo/generator-fluxible #122  
Duration: 2 min 56 sec  
Finished: about 16 hours ago

Current Branches Build History Pull Requests Settings

master Add eslint:recommended to .eslintrc # 547 passed  
Commit 66416c1  
Compare 3c010fa..66416c1  
ran for 3 min 4 sec  
4 days ago  
Michael Ridgway authored and committed

Build Jobs

✓ # 547.2	</> Node.js: 0.12	no environment variables set	1 min 24 sec
✓ # 547.3	</> Node.js: 0.10	no environment variables set	48 sec
! # 547.1	</> Node.js: iojs	no environment variables set	52 sec

Allowed Failures ?



[https://commons.wikimedia.org/wiki/File:Eye\\_iris.jpg](https://commons.wikimedia.org/wiki/File:Eye_iris.jpg)

# Continuous Delivery Pipeline

- Let the state of pipeline visible
  - Physically



# Continuous Delivery Pipeline

- Let the state of pipeline visible
  - Physically



<https://youtu.be/2ygPkb8KW70?t=2m55s>



# Continuous Delivery Pipeline

- Quick feedback
- Everyone is responsible for the quality
- Everyone is responsible for the delivery process

## Continuous Delivery Pipeline

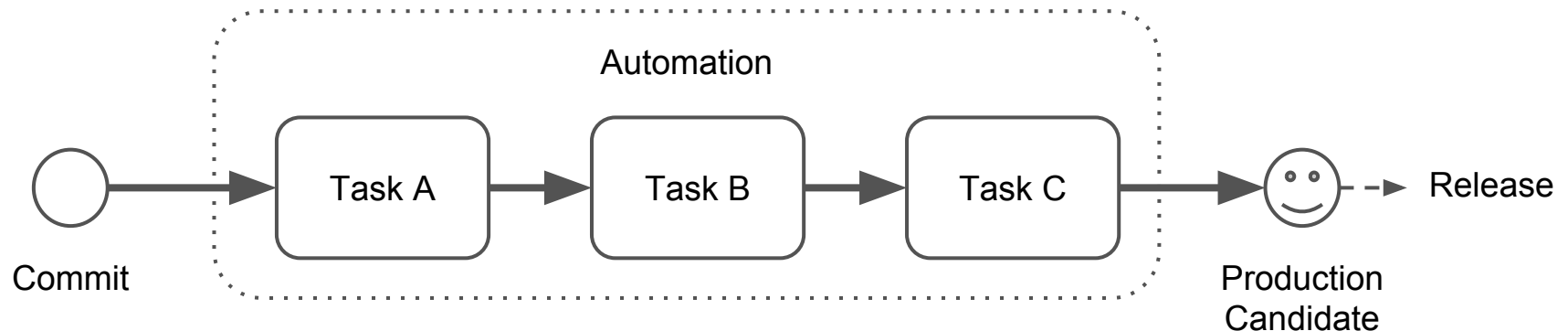
各取所需  
全程自動  
彈指即得  
快速回饋

— Plasma Chen, 2014  
Generated by 萌典字圖

YAHOO!

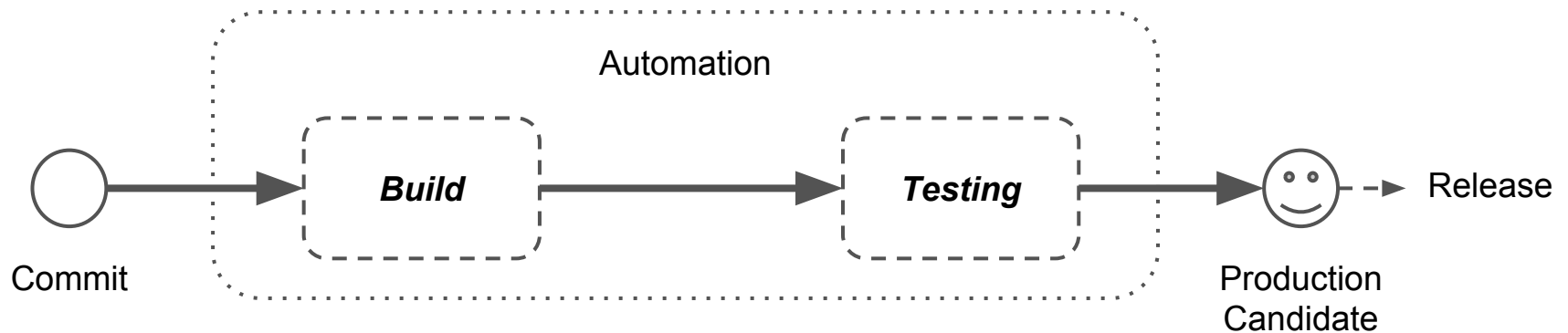
# Continuous Delivery Pipeline

- Automatically delivers code from commit to production
  - Build production from code
  - Testing



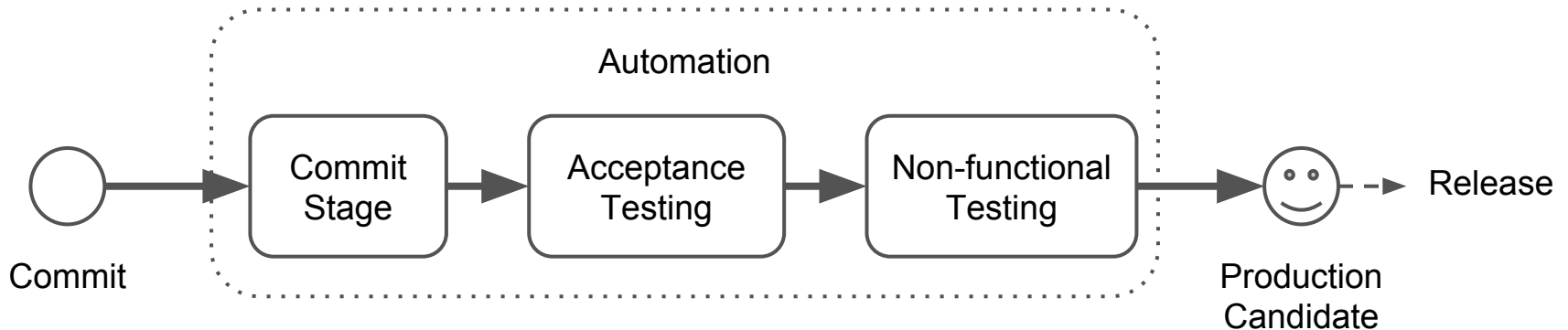
# Continuous Delivery Pipeline

- Automatically delivers code from commit to production
  - Build production from code
  - Testing



# Mobile App Continuous Delivery Pipeline

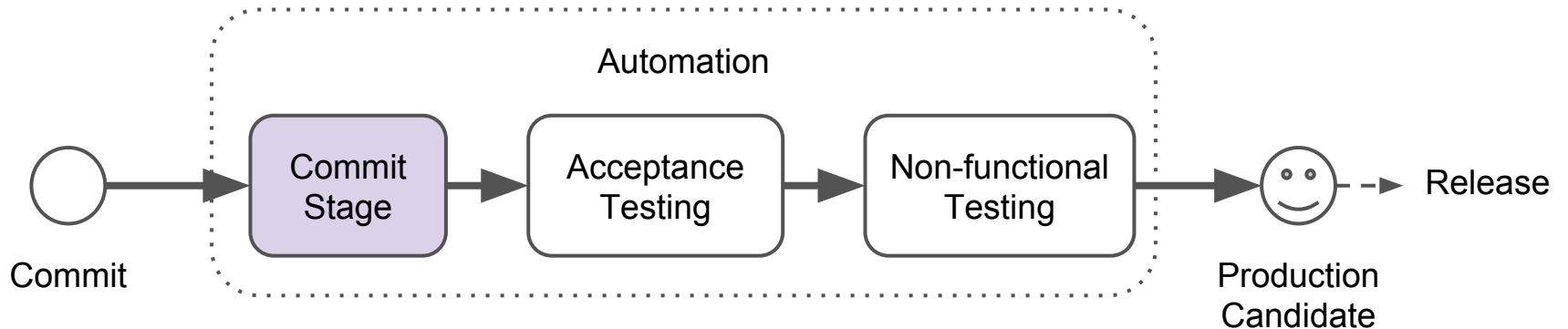
- Automatically delivers code from commit to production
  - Commit Stage
  - Automated Acceptance Testing Stage
  - Automated Non-functional Testing Stage



# Mobile App Continuous Delivery Pipeline

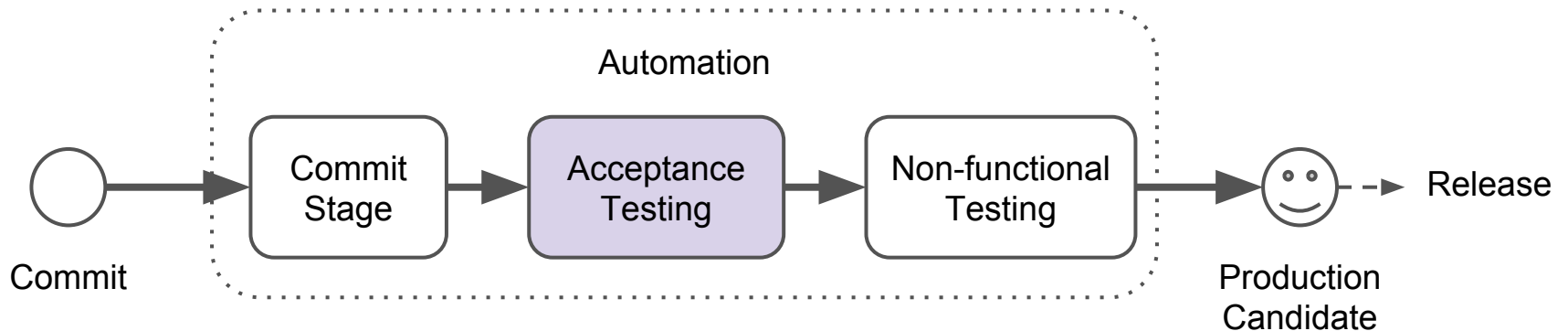
## ■ Commit Stage

- Build App
- Assert App works at technical level



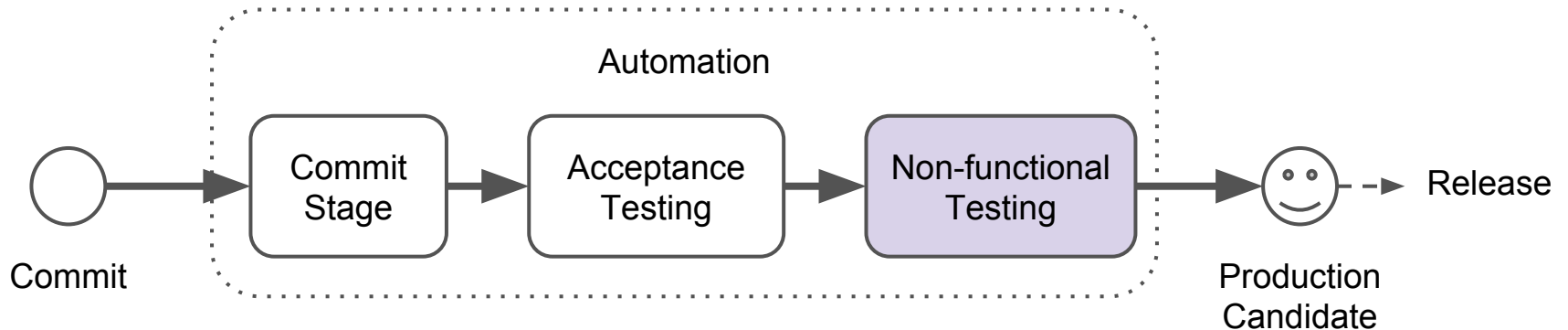
# Mobile App Continuous Delivery Pipeline

- Automated Acceptance Testing Stage
  - Assert App works at functional level



# Mobile App Continuous Delivery Pipeline

- Automated Non-functional Testing Stage
  - Assert App works at non-functional level

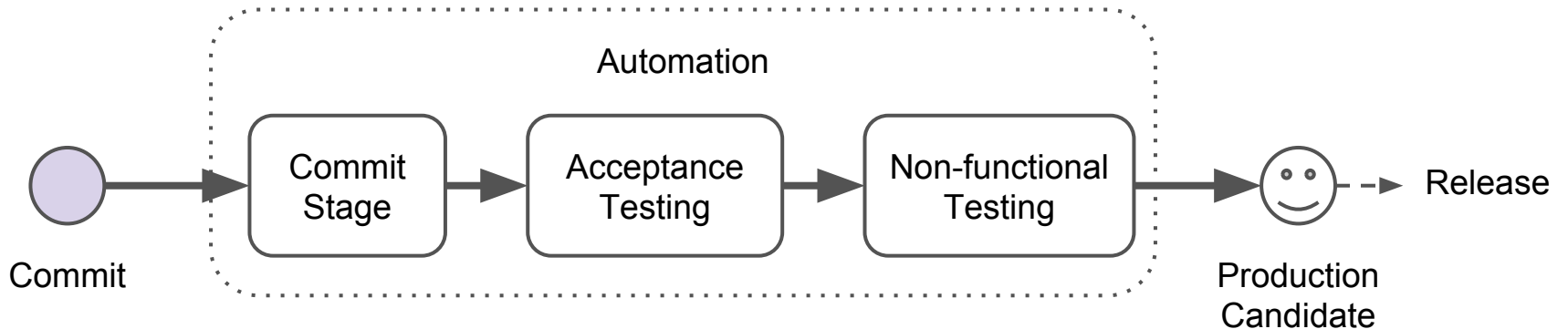




Pre Commit

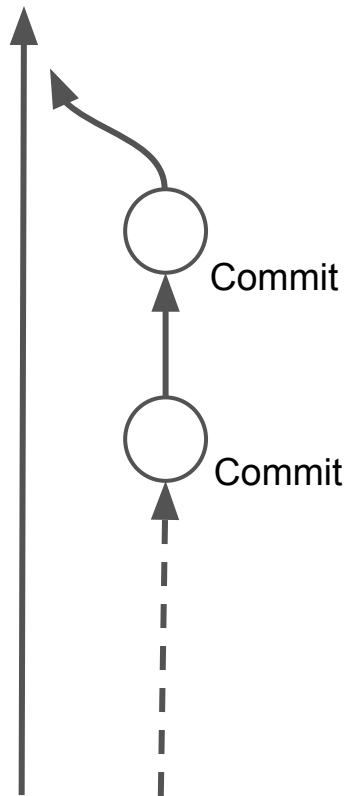
# Pre Commit

- Developing, make changes
- Build locally
- Test locally (e.g. Unit test)
- Commit after it can be built successfully and passing test



# Pull Request

- After commit change locally, create a pull request



# Code Review

- Changes need be reviewed by other developers
- DO NOT merge it without code review

 @@ -2,6 +2,9 @@			
2	"B": {	2	"B": {
3	"desc": "B"	3	"desc": "B"
4	},	4	},
		5	+ "C": {
		6	+   "desc": "C"
		7	+ }
5	"D": {	8	"D": {
6	"desc": "D"	9	"desc": "D"
7	}	10	}
			

# Code Review

- Changes need be reviewed by other developers
- DO NOT merge it without code review

```
8     var secondNum = 2;  
9     var expected = 3;  
10  
11    +// This is a test :\  
12     var result = Add(firstNum, secondNum);  
13
```

# Merge

- Merge after it can be built successfully and passing test

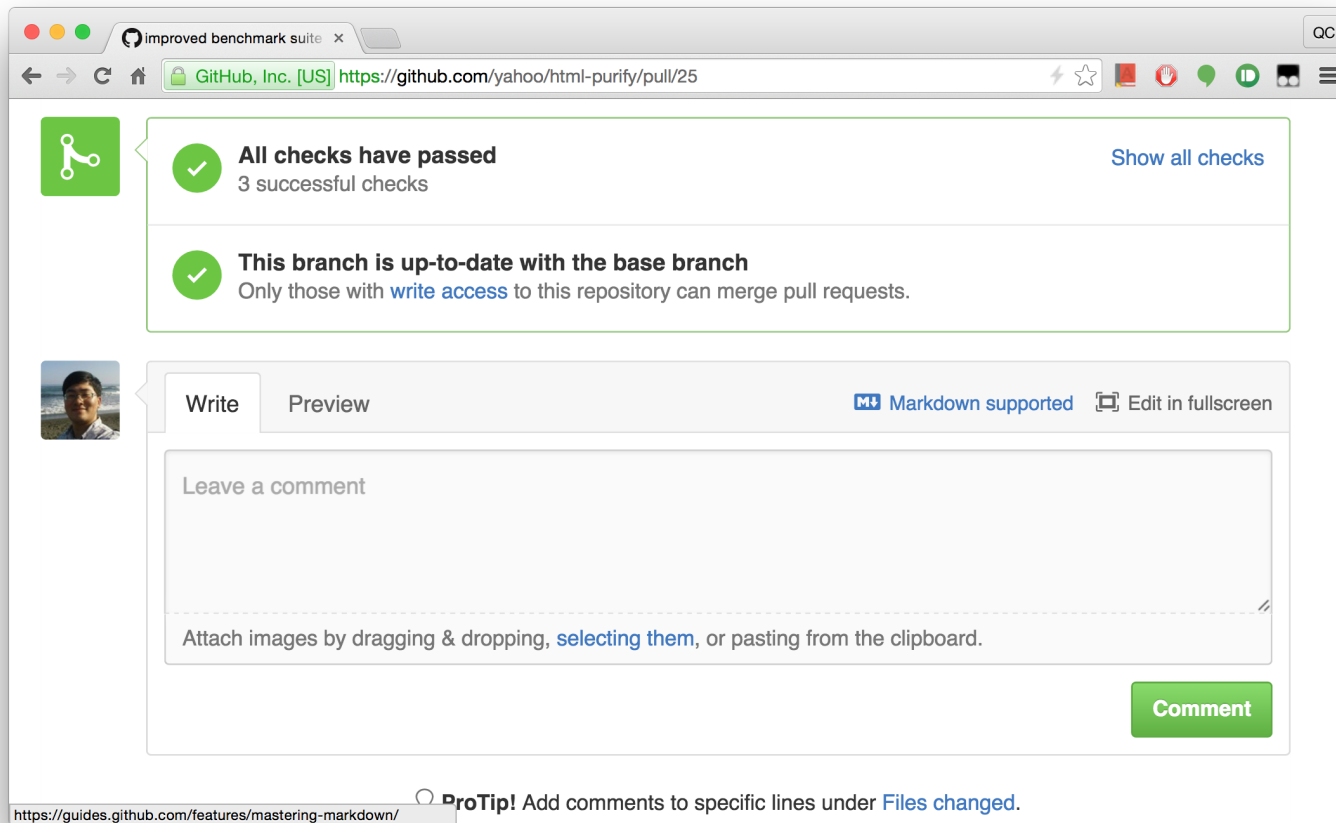
The screenshot displays the GitHub interface for the 'yahoo/html-purify' repository. The main content area shows a list of pull requests, all of which are marked as merged with a green checkmark. The pull requests are:

- #25: improved benchmark suite ✓ (opened 15 days ago by adon-at-work, 3 comments)
- #24: disable canonicalizing IE cond. comments ✓ (opened 15 days ago by adon-at-work, 3 comments)
- #23: Fix tag balancing ✓ (opened 15 days ago by adon-at-work, 7 comments)
- #22: Add proper quotes ✓ (opened 21 days ago by adon-at-work, 14 comments)
- #21: preserve quoting in attribute values ✓ (opened 22 days ago by maditya, 2 comments)

At the bottom of the list, there is a ProTip: `no:milestone` will show everything without a milestone.

# Merge



- Merge after it can be built successfully and passing test





The screenshot shows a web browser window with the URL <https://github.com/yahoo/html-purify/pull/25>. The page displays two green checkmarks indicating that all checks have passed and the branch is up-to-date with the base branch. Below this, there is a comment section with a text area and a 'Comment' button. A 'ProTip!' is visible at the bottom of the page.

improved benchmark suite x

GitHub, Inc. [US] <https://github.com/yahoo/html-purify/pull/25>

  **All checks have passed** [Show all checks](#)  
3 successful checks

 **This branch is up-to-date with the base branch**  
Only those with [write access](#) to this repository can merge pull requests.

 **Write** **Preview** [Markdown supported](#) [Edit in fullscreen](#)

Leave a comment

Attach images by dragging & dropping, [selecting them](#), or pasting from the clipboard.

**Comment**

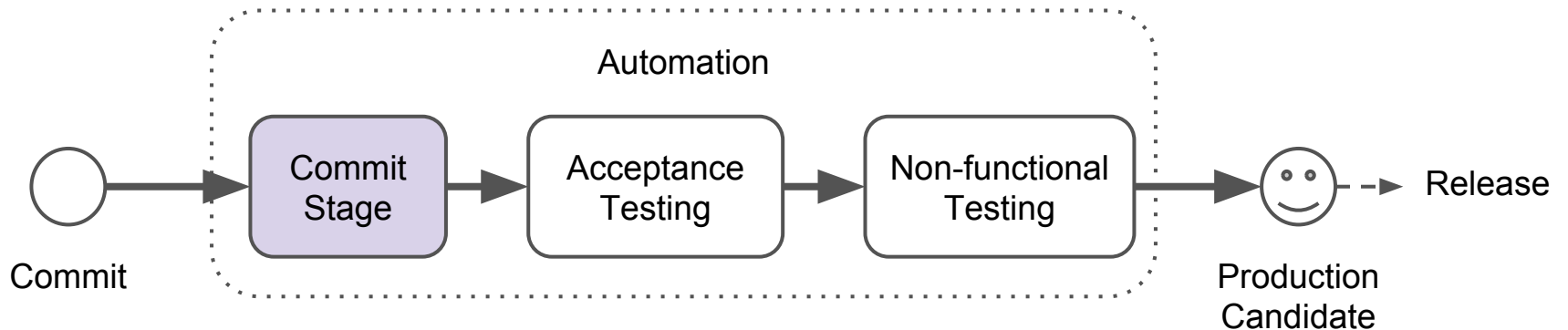
<https://guides.github.com/features/mastering-markdown/> **ProTip!** Add comments to specific lines under [Files changed](#).

**Commit Stage**



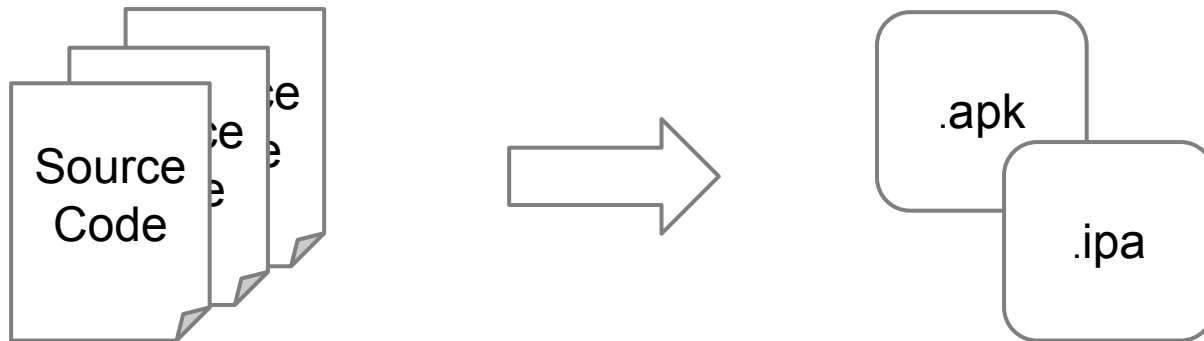
# Commit Stage

- Assert App works at technical level
- Build
- Unit test



# Build

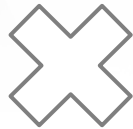
- Compile source code to build App



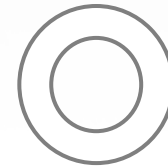
# Unit Test

- Test a method with input, then check if the output is expected
- Don't repeat yourself
- Arrange-Act-Assert pattern

# Unit Test



```
1 // method
2 var Add = function(a, b){
3     return a + b;
4 };
5
6 // arrange
7 var firstNum = 1;
8 var secondNum = 2;
9 var expected = firstNum + secondNum;
10
11 // act
12 var result = Add(firstNum, secondNum);
13
14 // assert
15 check(expected, result);
```



```
1 // method
2 var Add = function(a, b){
3     return a + b;
4 };
5
6 // arrange
7 var firstNum = 1;
8 var secondNum = 2;
9 var expected = 3;
10
11 // act
12 var result = Add(firstNum, secondNum);
13
14 // assert
15 check(expected, result);
```

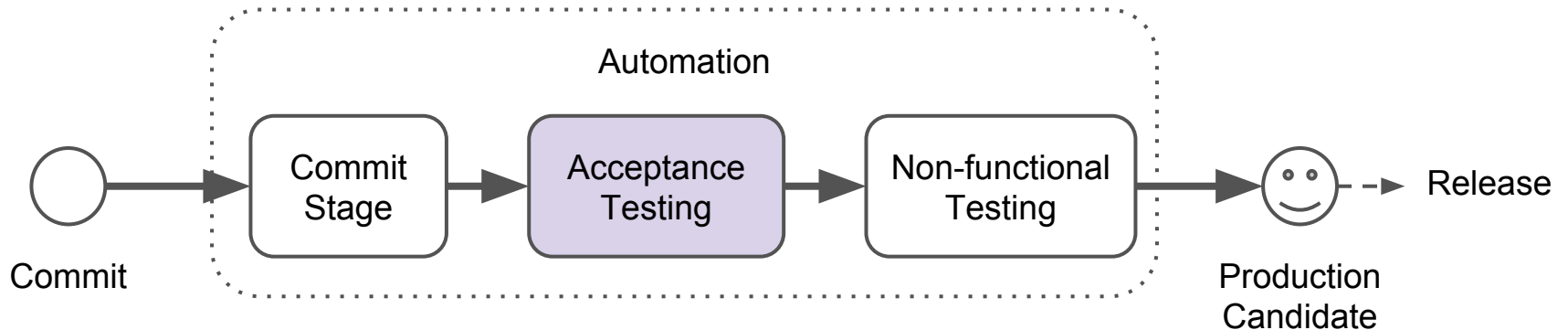
# Analysis

- Check its health
  - Test coverage
  - Duplicated code
  - Number of warnings
  - ...

# Automated Acceptance Testing Stage

# Automated Acceptance Testing Stage

- Assert App delivers the value to user is expected
- Smoke test
- Functional test



# Smoke Test

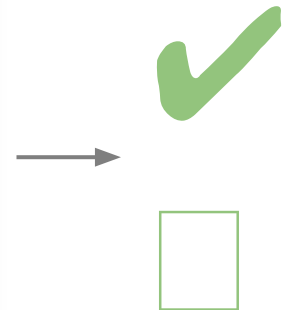
- Check App can provide most important features
- Test through UI automation





# Functional Test

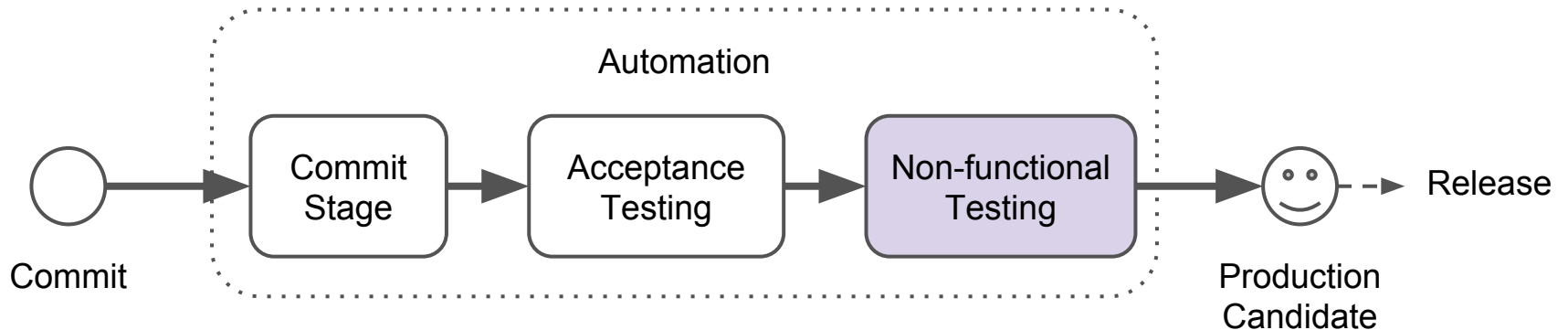
- Check App provide main features we want to deliver
- Test through UI automation



# Automated Non-functional Testing Stage

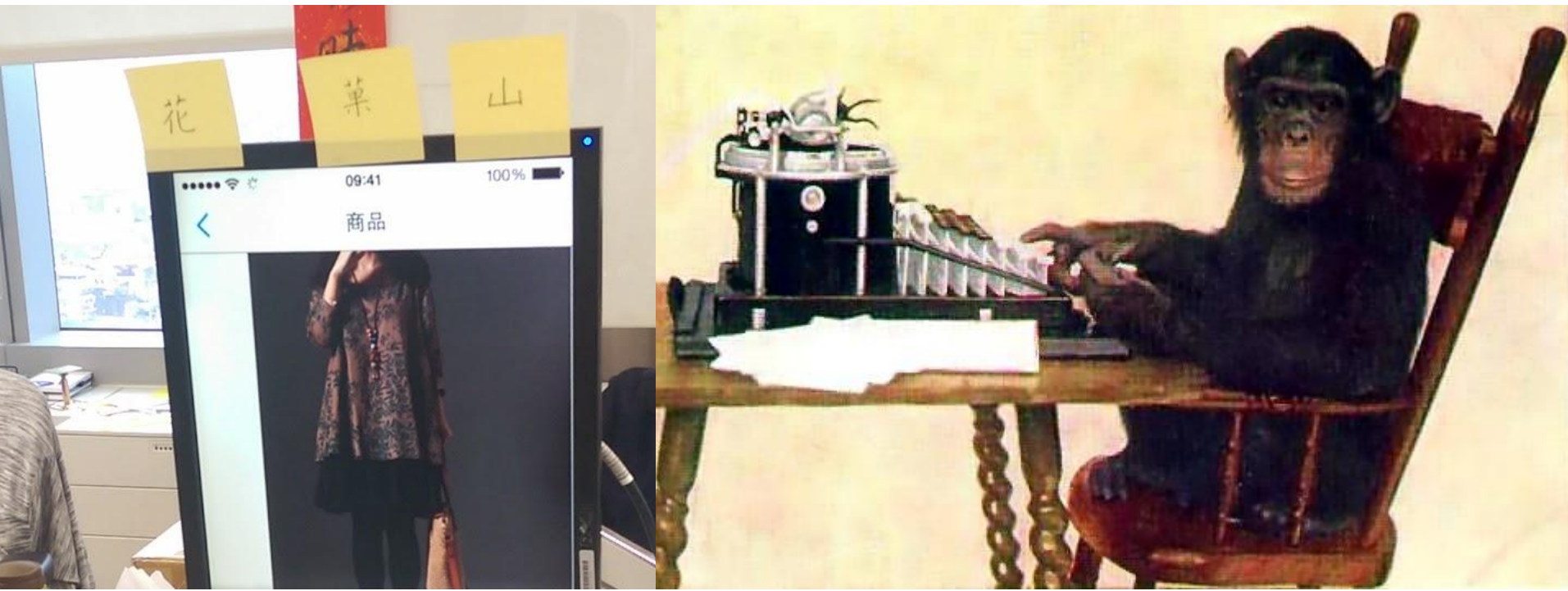
# Automated Non-functional Testing Stage

- Assert App works at non-functional level
- Stability Test
- Performance Test



# Stability Test

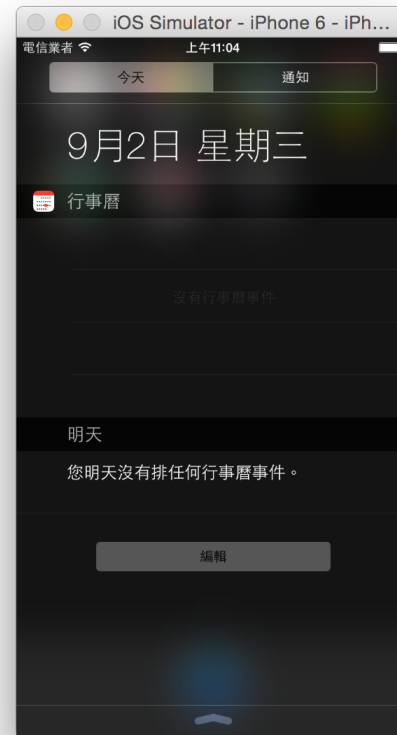
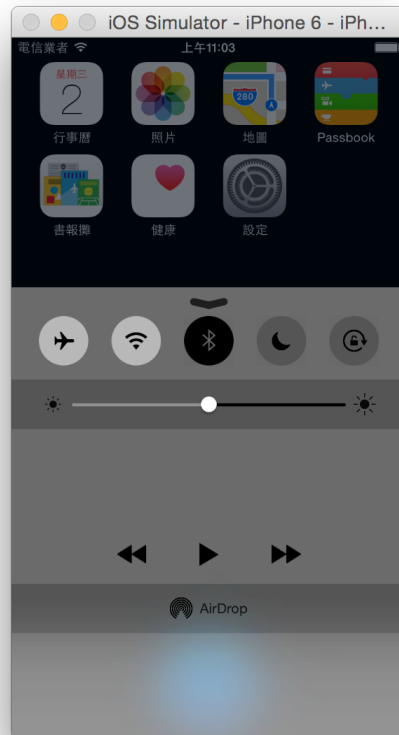
- Monkey test



<https://commons.wikimedia.org/wiki/File:Monkey-typing.jpg>

# Stability Test

## ■ Monkey test



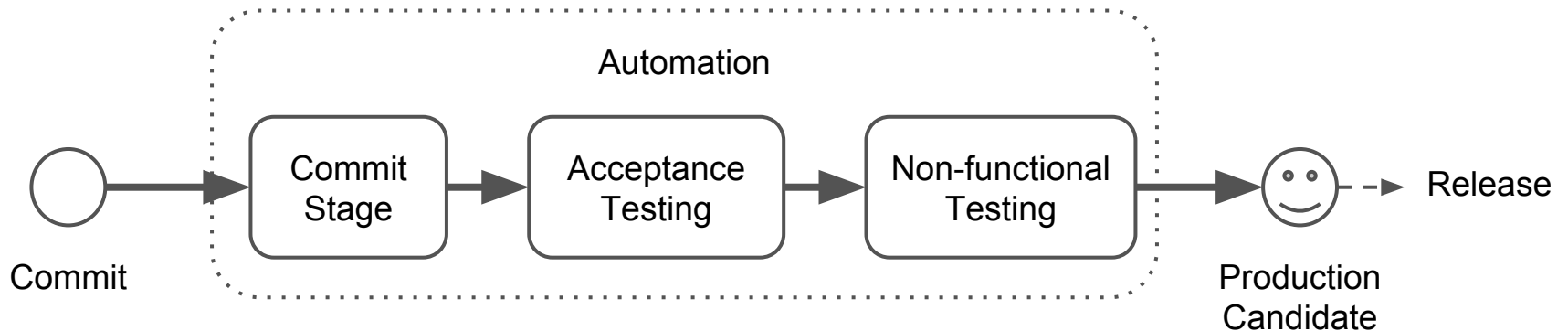
# Performance Test

- Cold start time
- Memory usage
- FPS
- ...

Release

# Mobile App Continuous Delivery Pipeline

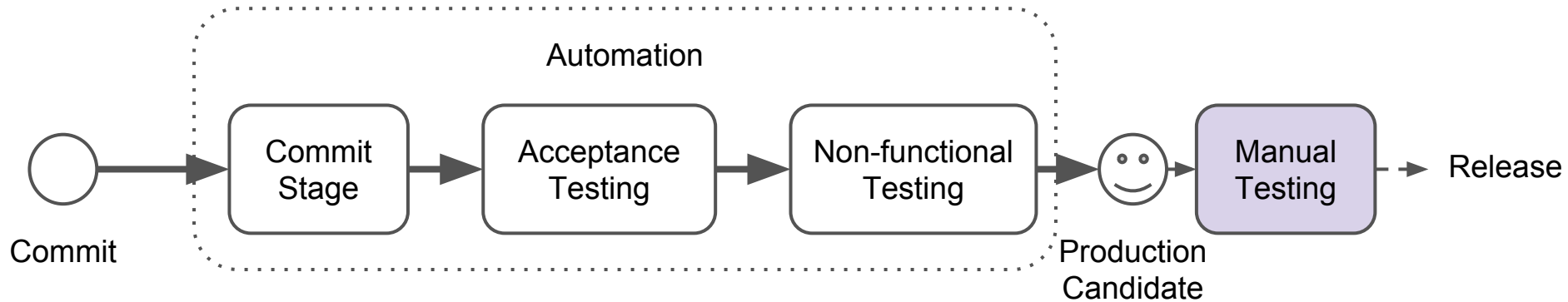
- There is always a gap between ideal and reality





# Mobile App Continuous Delivery Pipeline

- There is always a gap between ideal and reality



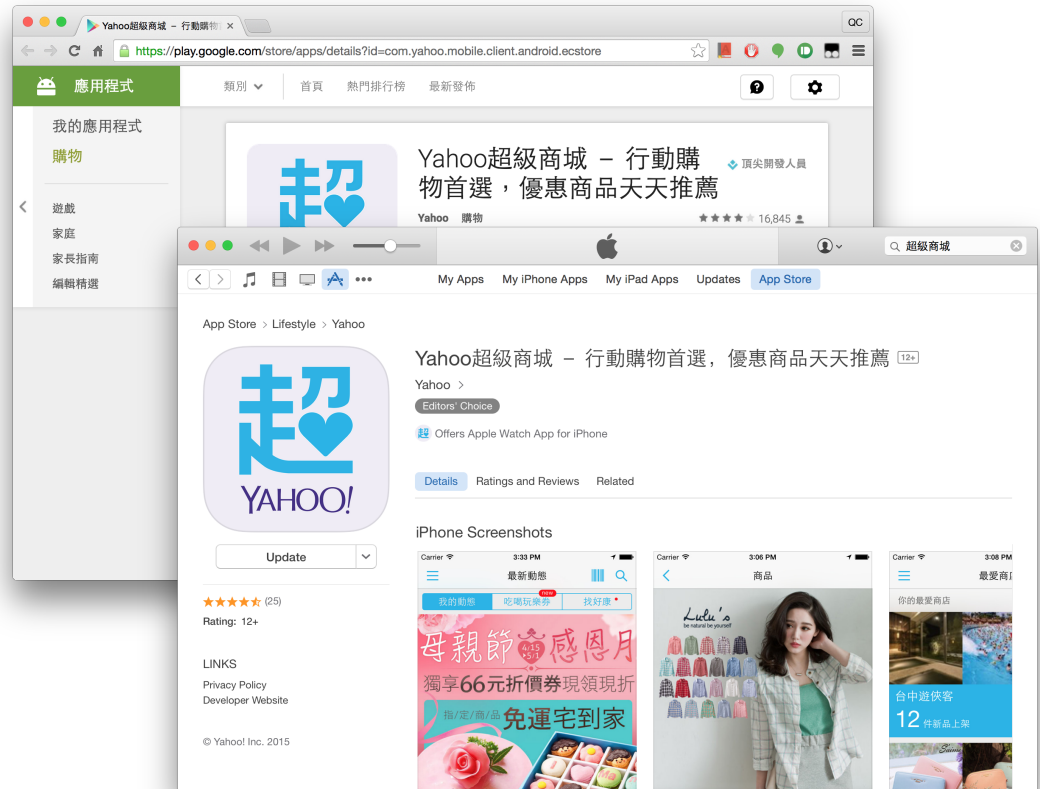
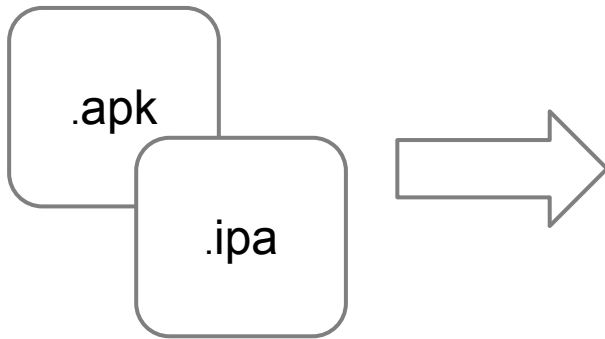
# Manual Testing

- Something can not be tested automatically



# Release

- Submit the production candidate to Google Play / App Store



## After Release

- If find an error in production...

哀公問：「弟子孰為好  
學？」孔子對曰：「有  
顏回者好學，不遷怒，  
不貳過，不幸短命死矣  
！」

—《論語》  
Generated by 萌典字圖

YAHOO!

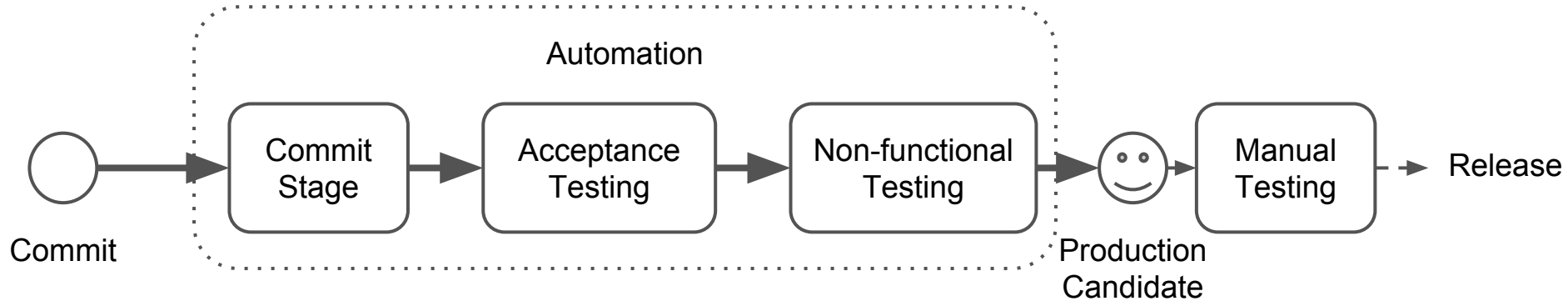
## After Release

- Add test cases to prevent it, 「不貳過」

哀公問：「弟子孰為好學？」孔子對曰：「有顏回者好學，不遷怒，不貳過，不幸短命死矣！」

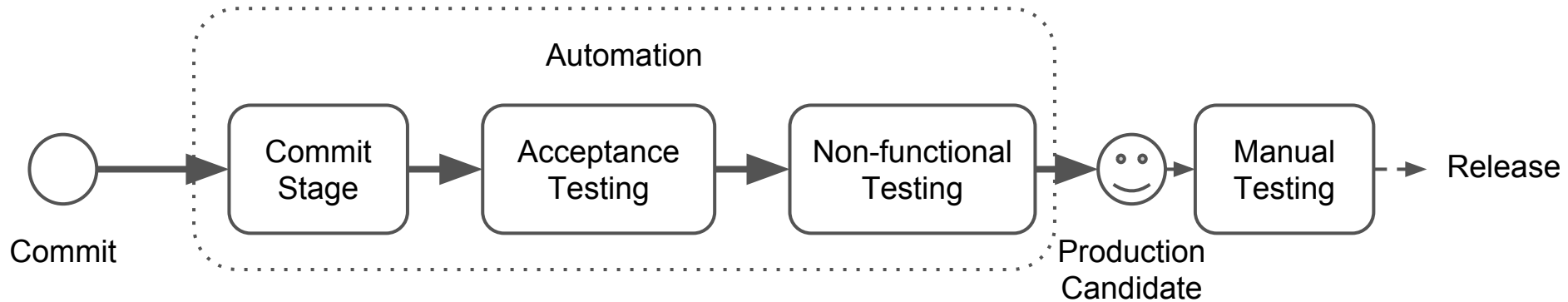
# Mobile App Continuous Delivery Pipeline

- Low cycle time
- High quality



# Mobile App Continuous Delivery Pipeline

Commit to Production (almost) without human intervention



Q & A 