

---

# Software Deployment

CS480 Software Engineering

<http://cs480.yusun.io>

February 8, 2015

Yu Sun, Ph.D.

<http://yusun.io>

[yusun@cpp.edu](mailto:yusun@cpp.edu)

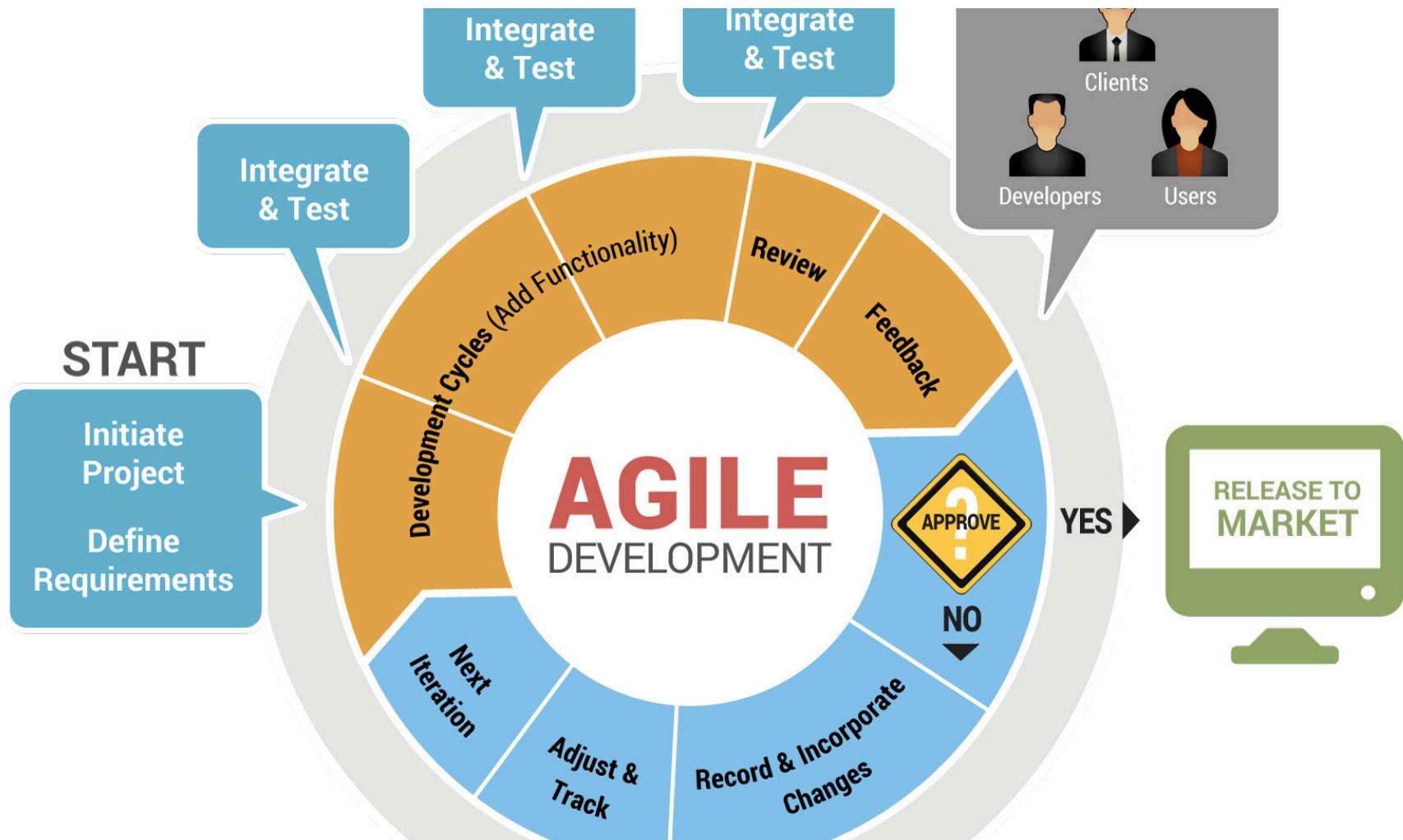


---

CAL POLY POMONA

---

# Agile Development



# Testing, Testing and Testing

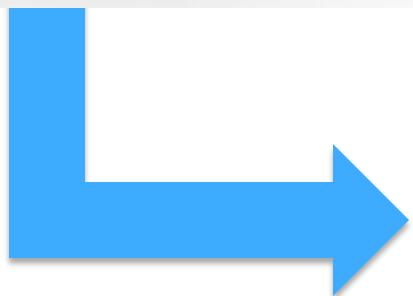
---

**DEV QA**



# Ready to Deploy

---



# What is Software Deployment?

---

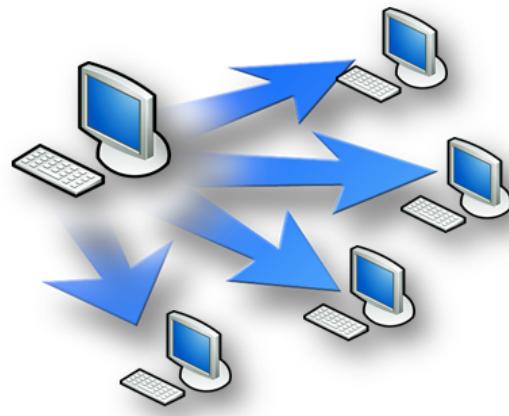
- ◆ *Software Deployment* is the art of deploying software artifacts produced as a result of build, on the target environment



# What is Software Deployment?

---

- ◆ Move your software from your local (dev) environment to the remote (prod) environment



# Deployment is Risky

---



Development



Production

# Deployment is Risky



A screenshot of the Amazon mobile website. The header includes the Amazon logo and "Prime" text. On the right, there are search and shopping cart icons. The main content area has a dark header bar with the text "New &amp; Used Offers". Below this, a red-bordered box contains the message: "The Amazon.com mobile service encountered a problem. Please try again shortly. (CS4)". At the bottom right of the box is a button labeled "OK".

A screenshot of a tweet from the official Netflix Twitter account (@Netflixhelps). The tweet reads: "We're aware that some members are experiencing issues streaming movies and TV shows. We're working to resolve the problem." Below the tweet are standard social media interaction buttons: "Reply", "Retweet", and "Favorite". At the bottom, there is a "RETWEETS" count of 14, followed by a row of small thumbnail images representing different movie and TV show covers. The timestamp at the bottom is "11:50 PM - 29 Jun 12 via web · Embed this Tweet".



# Deployment is Risky

---



# Deployment Tips

---

- ◆ Deployable Artifacts
- ◆ Repeatable Builds
- ◆ Consistent Environments
- ◆ Autonomous Packages
- ◆ Ease to-do/un-do Releases



# Deployable Artifacts

- ◆ Executable
- ◆ Easy to execute

```
for h3 in page.findAll("h3"):
    value = (h3.contents[0])
    if value != "Afdeling":
        print >> txt, value
import codecs
f = codecs.open("alle.txt", "r", encoding="utf-8")
text = f.read()
f.close()
# open the file again for writing
f = codecs.open("alle.txt", "w", encoding="utf-8")
f.write(value+"\n")
# write the original contents
```



# Repeatable Builds

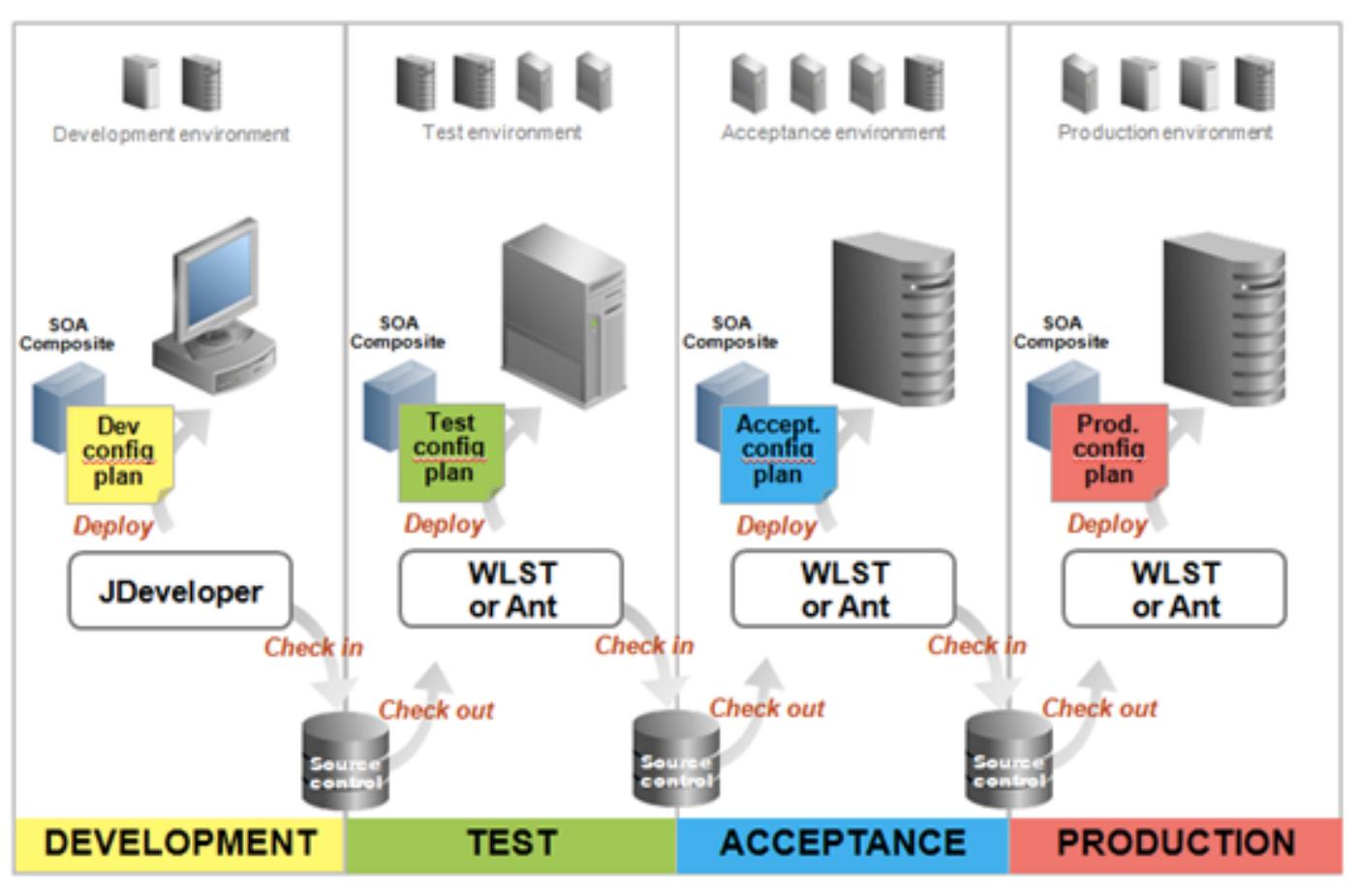
---

- ◆ One should have the provision of rebuilding the application exactly in the same way as it was built at the time of release
  - ◆ Same dependencies
  - ◆ Same versions
- ◆ IDEs do not work
- ◆ Use a build tool



# Consistent Environment

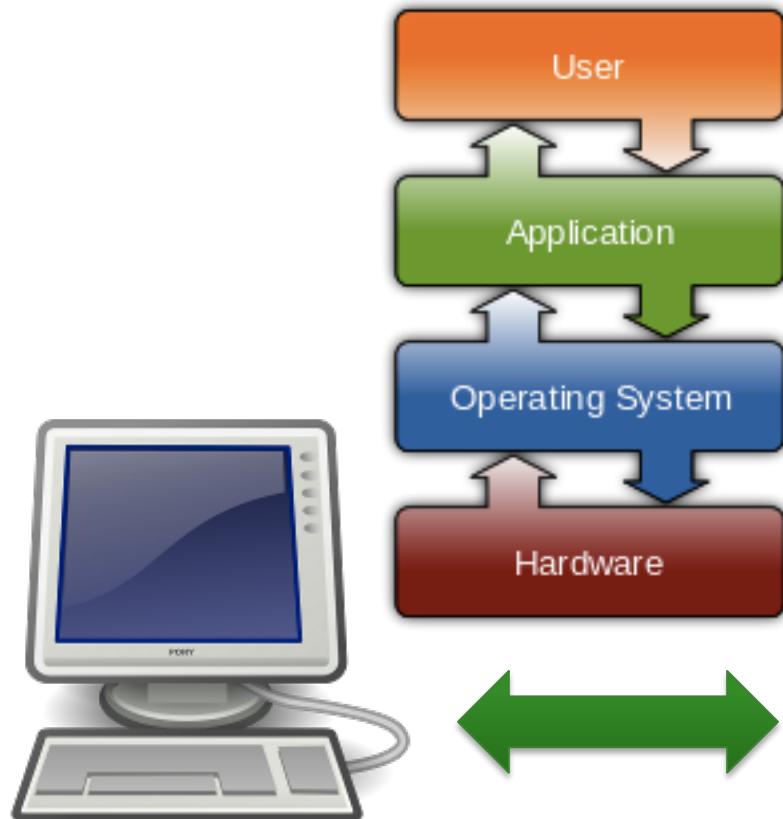
- ◆ An environment includes all the components needed to build and run the application



# Consistent Environment

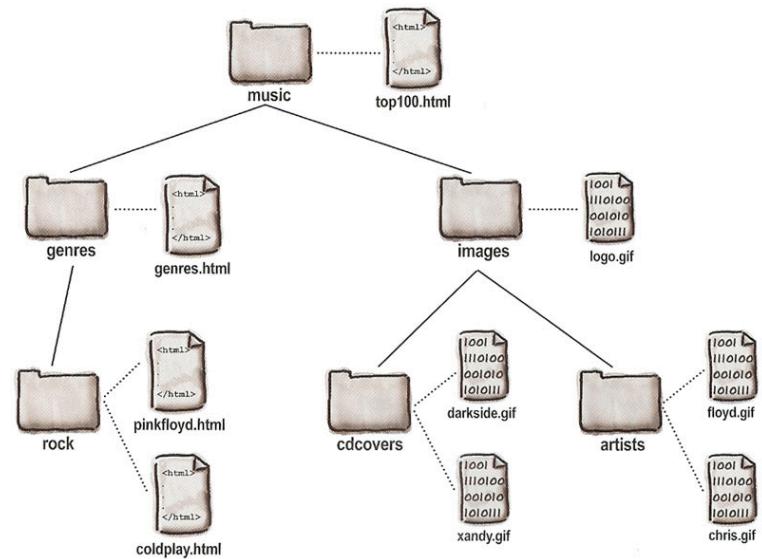
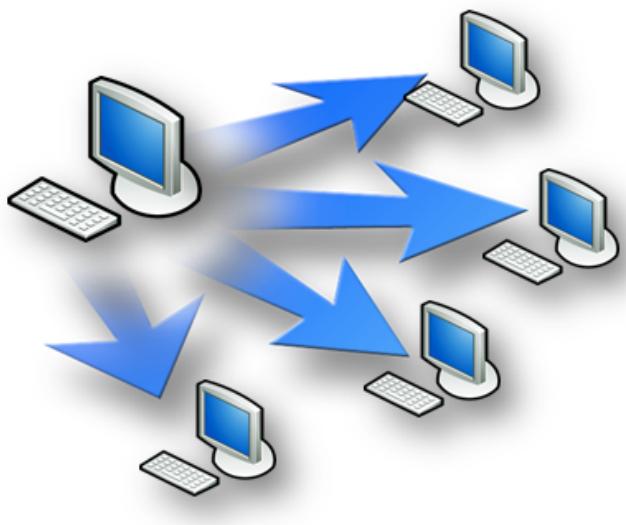
---

- ◆ An environment includes all the components needed to build and run the application



# Autonomous Packages

- ◆ Autonomous Packages means the deployable package should exist independently
- ◆ The more autonomous the deployable package is the more easier to deploy and maintain.



# Easy Todo/Undo/Redo

---

- ◆ Ensure software releases can be easy to
  - ◆ Install
  - ◆ Uninstall
  - ◆ Rollback

