

# INTRODUCTION TO SOFTWARE BUSINESS PRODUCT MANAGEMENT

- WEEK 3 DAY 2
- LED BY: EMILY CROSE
- FOR
- OAKLAND UNIVERSITY

# REVIEW OF DAY 1

QUESTIONS FROM  
DAY 1?



# TERMS TO LISTEN FOR

Interface/UI

- Visual elements of our application

UX (User Experience)

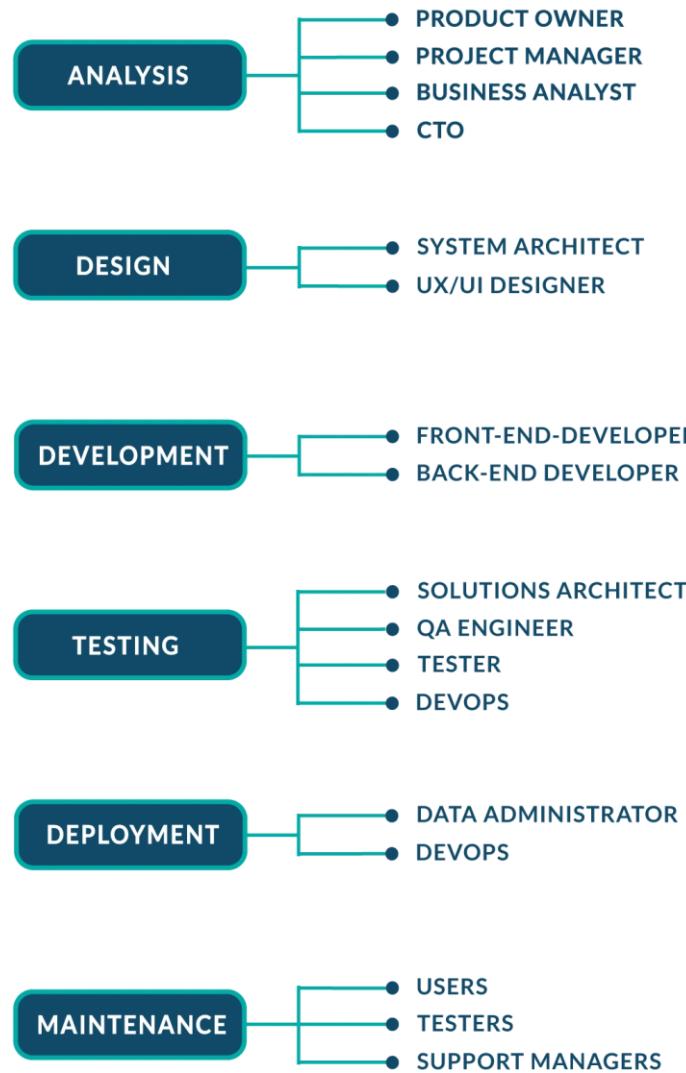
- Functionality of the app and its features

Topology

- Network architecture diagram

# THE SOFTWARE DEVELOPMENT LIFE CYCLE CONTINUED

# SDLC Phases



WE'VE DECIDED TO MAKE  
SOME SOFTWARE!

(Now WHAT?)

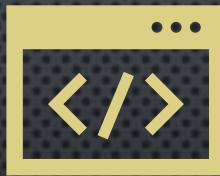


DESIGN STAGE

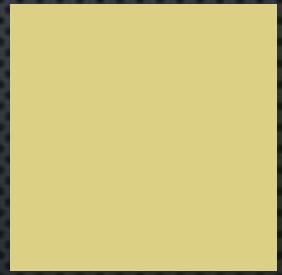
# DESIGN ROLES



System architect



UX Designer



UI Designer

# DESIGN ACTIVITIES



UX MOCKUPS



DATABASE  
REQUIREMENTS



PROTOTYPING



VISUALS

Layout

Visual Design

Branding

Research

Usability Testing

Personalization

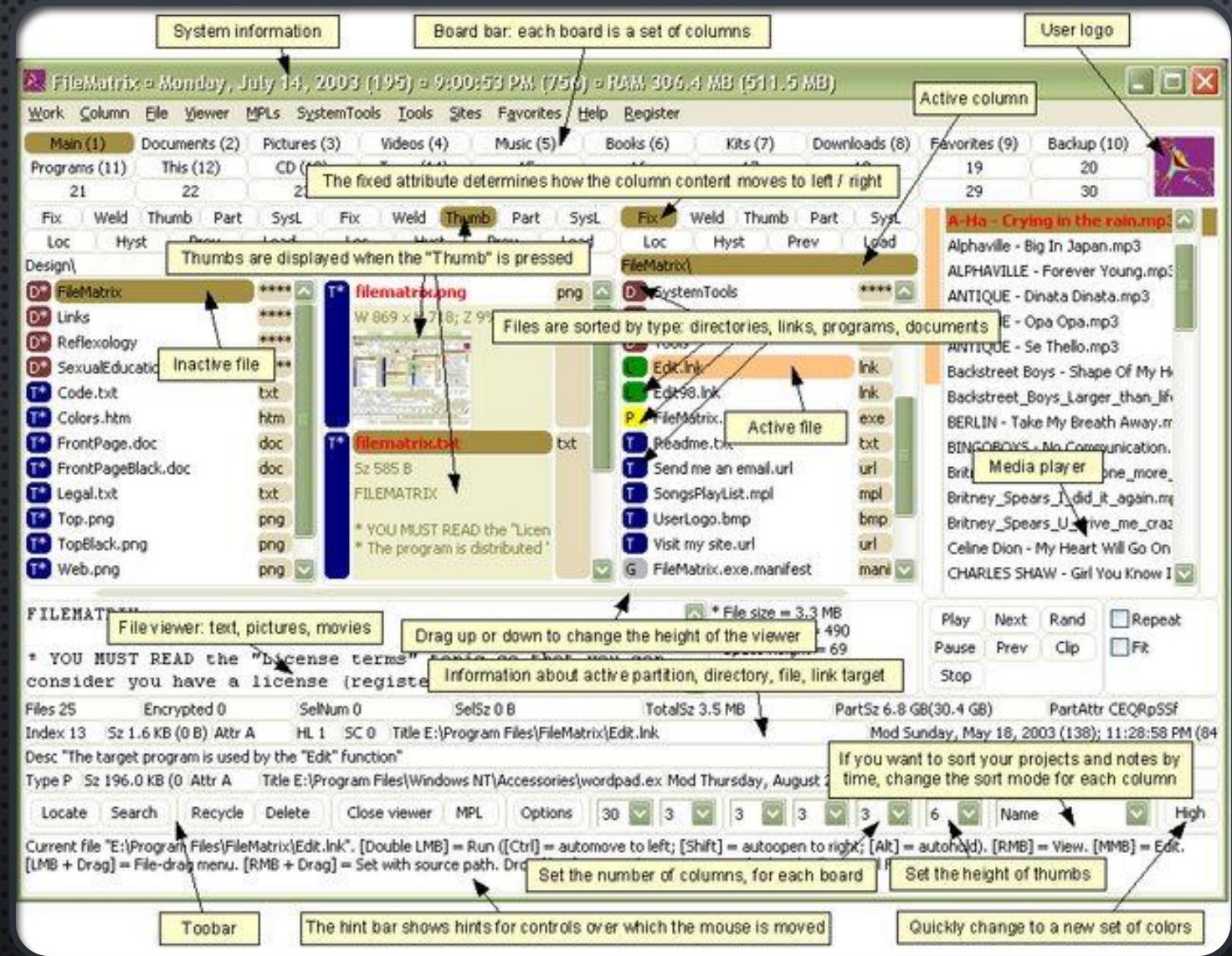


UI & UX



# BRANDING

# CURSED UI



# MINIMALIST UI

The image displays a modern, minimalist user interface (UI) for an administrator. The left side features a vertical sidebar with a blue header containing the title "Administrator". Below the header, the sidebar includes a "Dashboard" button, a "Products" section with a right-pointing arrow, a "My Wallet" section with a right-pointing arrow, a "Transactions" section with a downward/upward arrow, and two sub-options: "Income" and "Outcome". At the bottom of the sidebar is a "Settings" icon. A large blue rectangular area at the bottom of the sidebar contains the text "Go Pro" and the subtext "Administrator Pro is packed with premium features, and more!", followed by a white "Upgrade now" button.

The main content area is titled "Dashboard" and contains several key components:

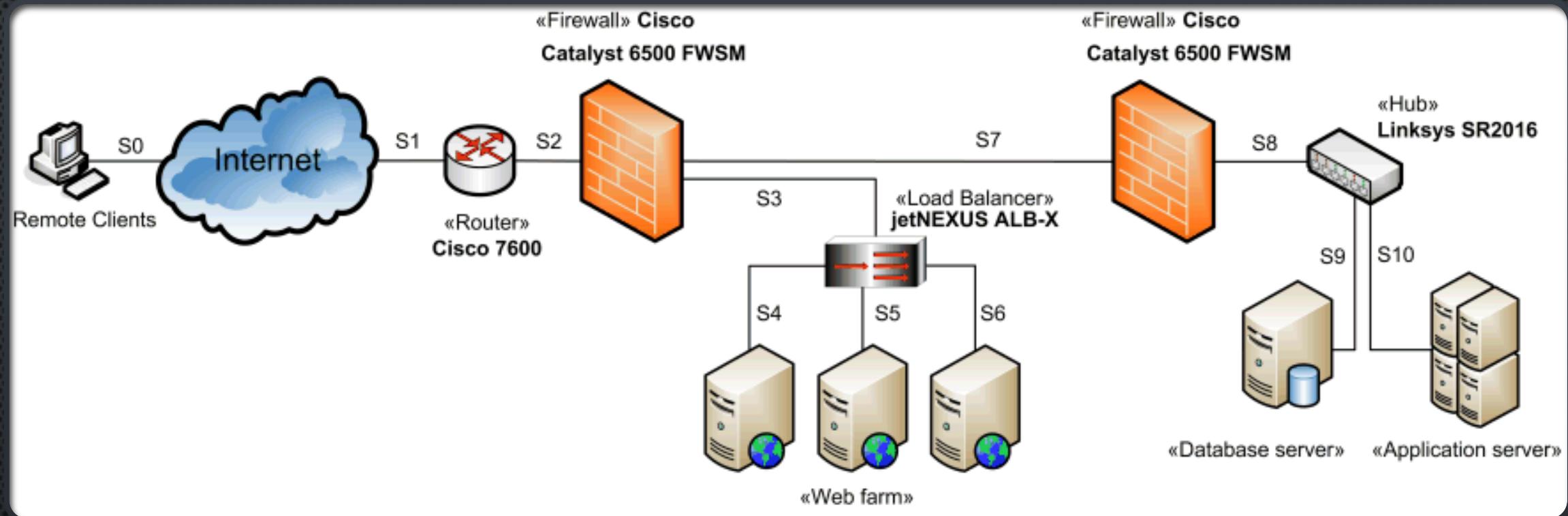
- Earnings Overview:** A line chart showing monthly earnings from January to December. The chart shows a general upward trend with some fluctuations. A specific point on the line for June is highlighted with a blue dot and a vertical line, indicating a detailed view or selection.
- Product Category:** A donut chart illustrating the distribution of products across four categories: Healthcare (dark blue), Fashion (light blue), Otomotif (grey), and Electronic (white).
- Direct Message:** A section for managing direct messages. It lists four messages from users: Stephanie, Cameron Williamson, Kathryn Murphy, and Robert Fox, each with a small profile picture and a snippet of their message content.
- All Products:** A table listing three products: Women's Sock, Black T-shirt, and School Bag. The table includes columns for "Info Products", "Price", "Stock", and "Active". Each product row has a toggle switch and a three-dot menu icon.



10 MINUTE BREAK

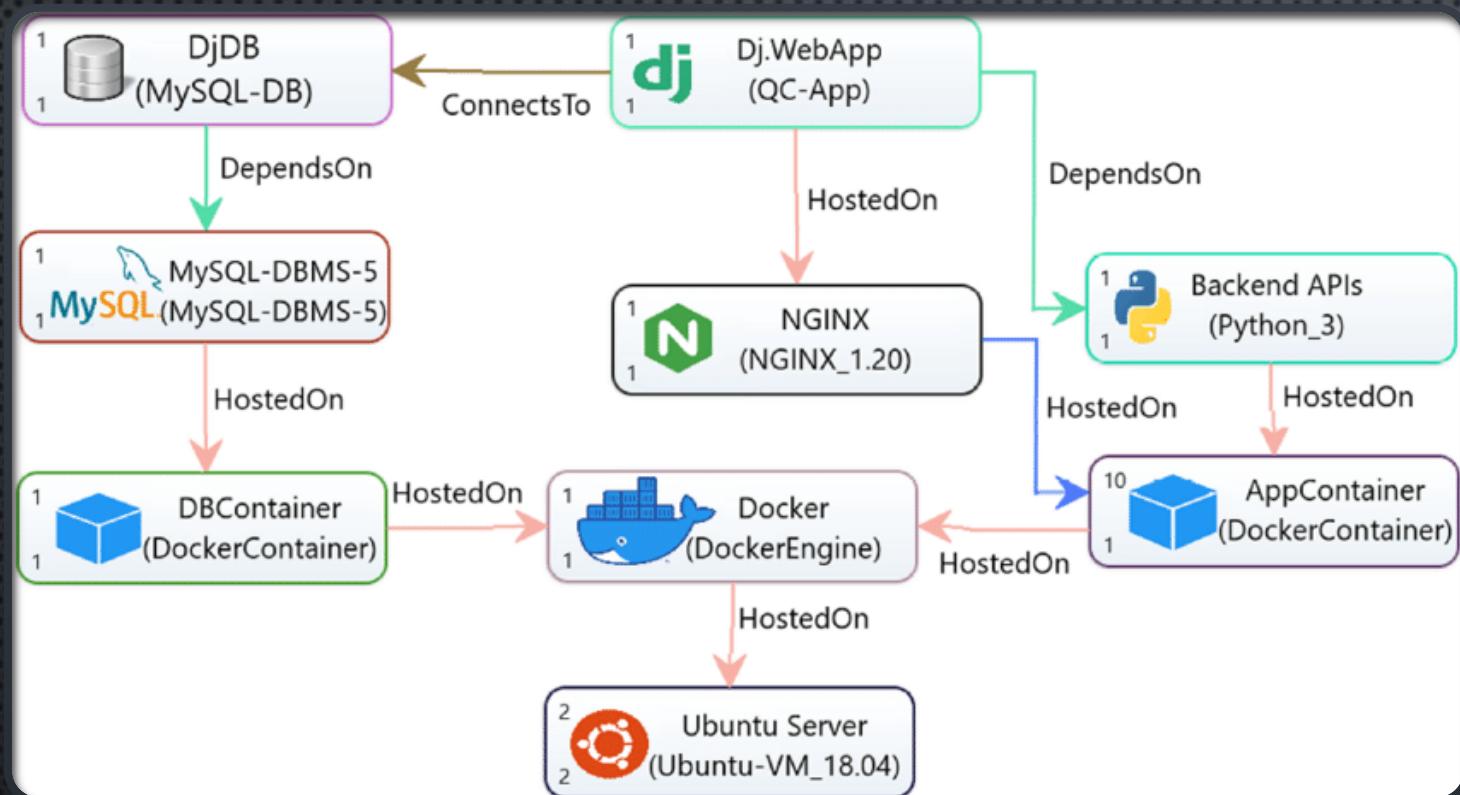
A black and white photograph showing several large, curved, concrete architectural elements. These structures overlap and curve outwards, creating a sense of depth and geometric complexity.

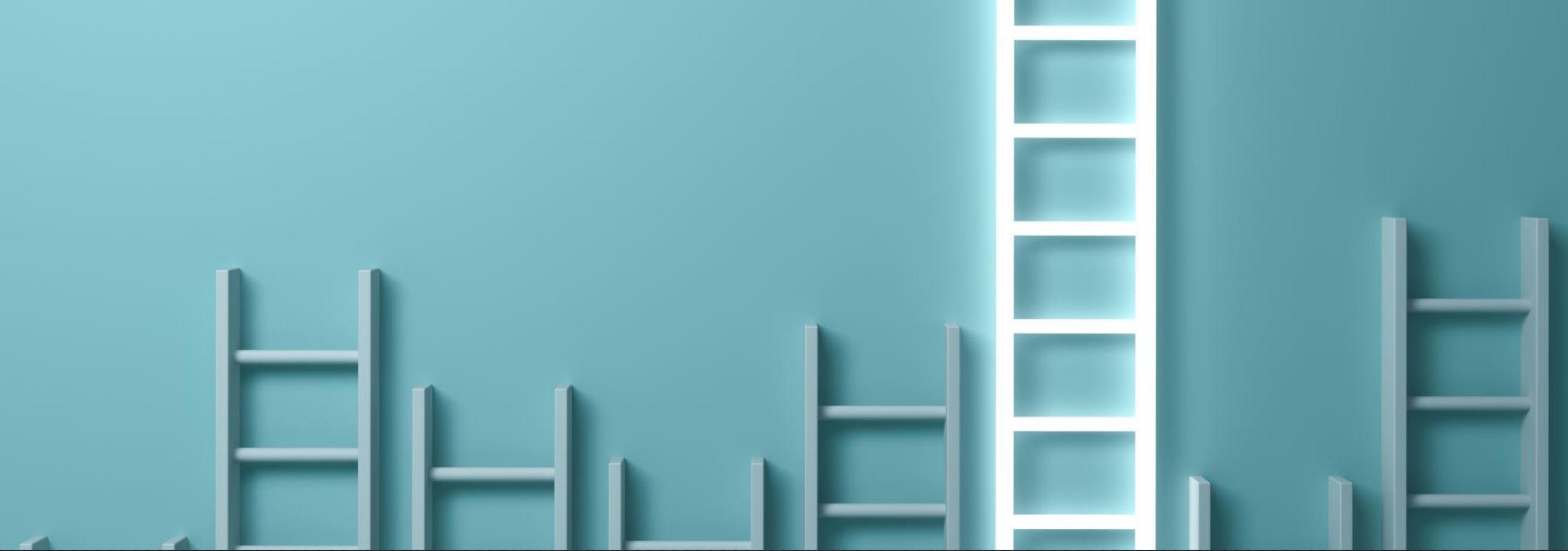
# ARCHITECTURE & DEPENDENCIES



# NETWORK TOPOLOGY

# DEPENDENCY DIAGRAM





DEVELOPMENT STAGE



DEVELOP  
PHASE  
ACTIVITIES

# DEVELOPMENT PHASE ACTIVITIES CONT'

- BUILDING INFORMATION SOURCES
  - BACK-END DATABASES
  - BUILD FEEDS FOR ONGOING INFORMATION TRANSFER
- BUILDING INFORMATION CONSUMERS
  - FED COMPONENTS OF APP

CONSIDERATION: PUSH, OR  
PULL?

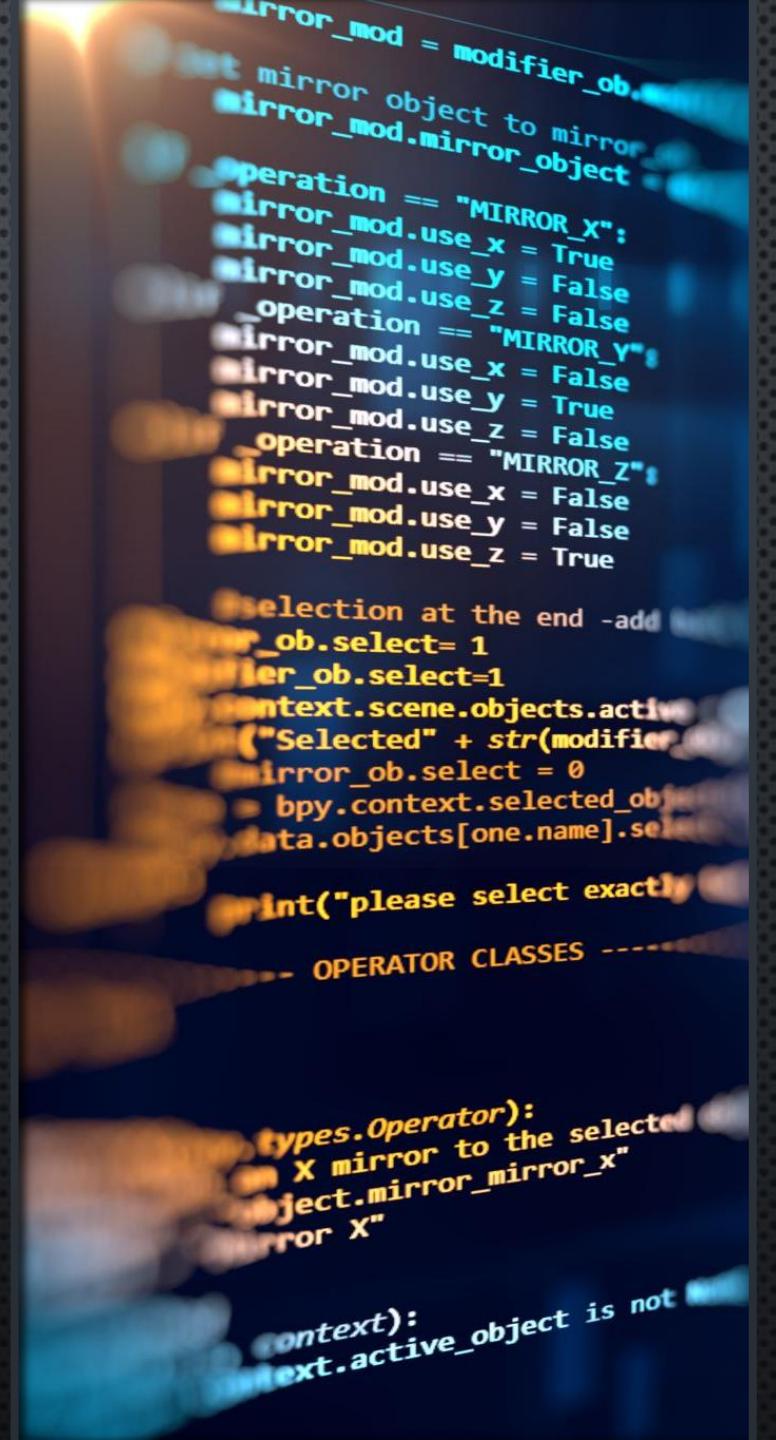
# DEVELOPER ROLES

Front-End  
Developer

Back-End  
Developer

# FRONT-END DEVELOPERS

- DEAL WITH UI/UX PROBLEM SOLVING
  - WORK WITH VISUAL PROGRAMMING LANGUAGES
- DESIGN INTEGRATION MODULES TO ATTACH TO BACK-END INFORMATION SOURCES

A photograph showing a person's hands pointing at a computer monitor. The monitor displays a block of Python code. The code appears to be a script for a 3D modeling application, specifically Blender, based on the use of bpy and context variables. The code handles different mirroring operations (MIRROR\_X, MIRROR\_Y, MIRROR\_Z) and manages object selection. The background is dark, making the white text of the code stand out.

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object

operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False

operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False

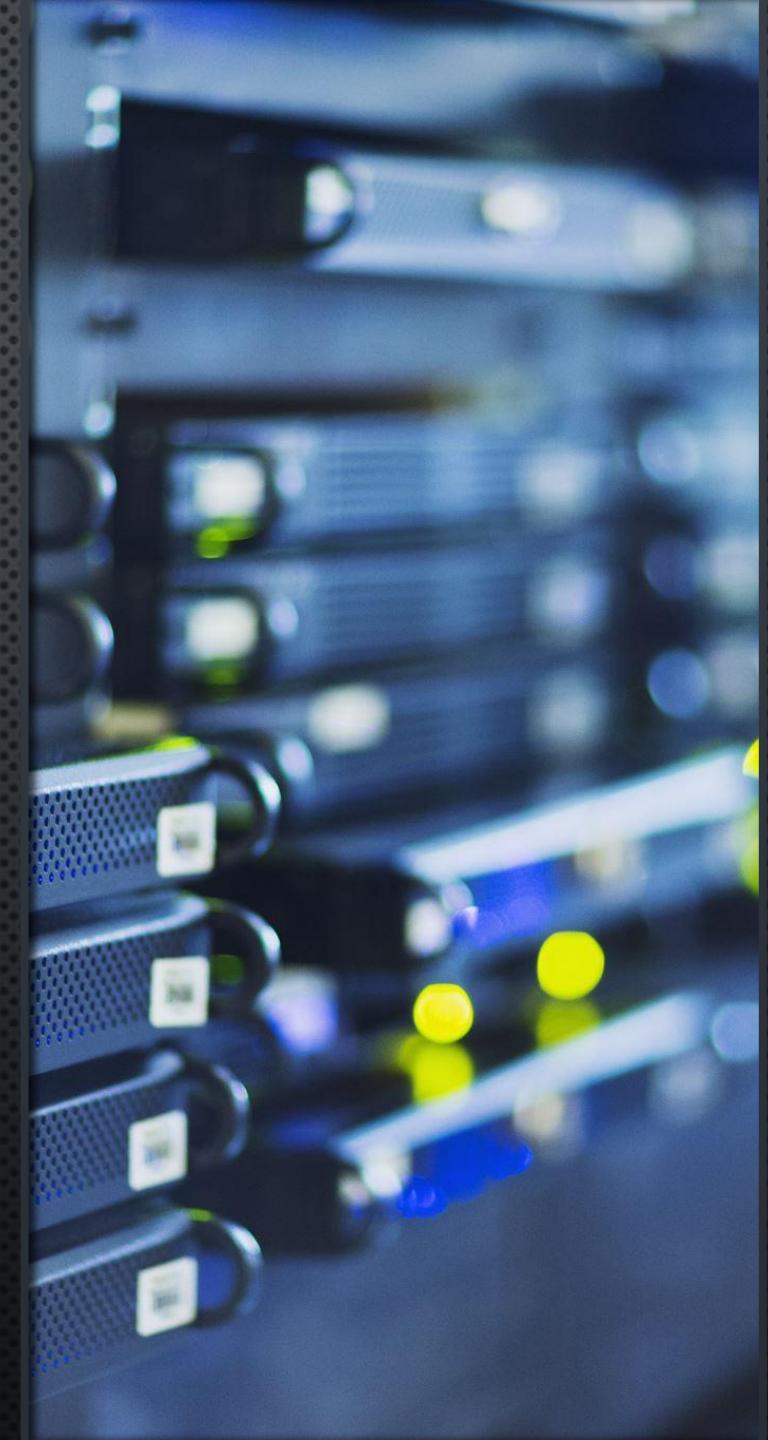
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

selection at the end -add
    ob.select= 1
    mirror_ob.select=1
    context.scene.objects.active
        ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects
        [one.name].select
    print("please select exactly one object")
- OPERATOR CLASSES ---

types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    or X"
context):
    context.active_object is not
```

# BACK-END DEVELOPERS

- WORK WITH DATA DISTRIBUTION
  - DATABASE ADMINISTRATORS
  - SYSTEM ENGINEERS
- AVAILABILITY IS KEY!
- ARE DATA SCIENTISTS NEEDED?



# TESTING PHASE

THE MOST IMPORTANT STAGE?

# TESTING ROLES

---

Solutions Architect

---

Quality Assurance Engineer

---

Software Tester

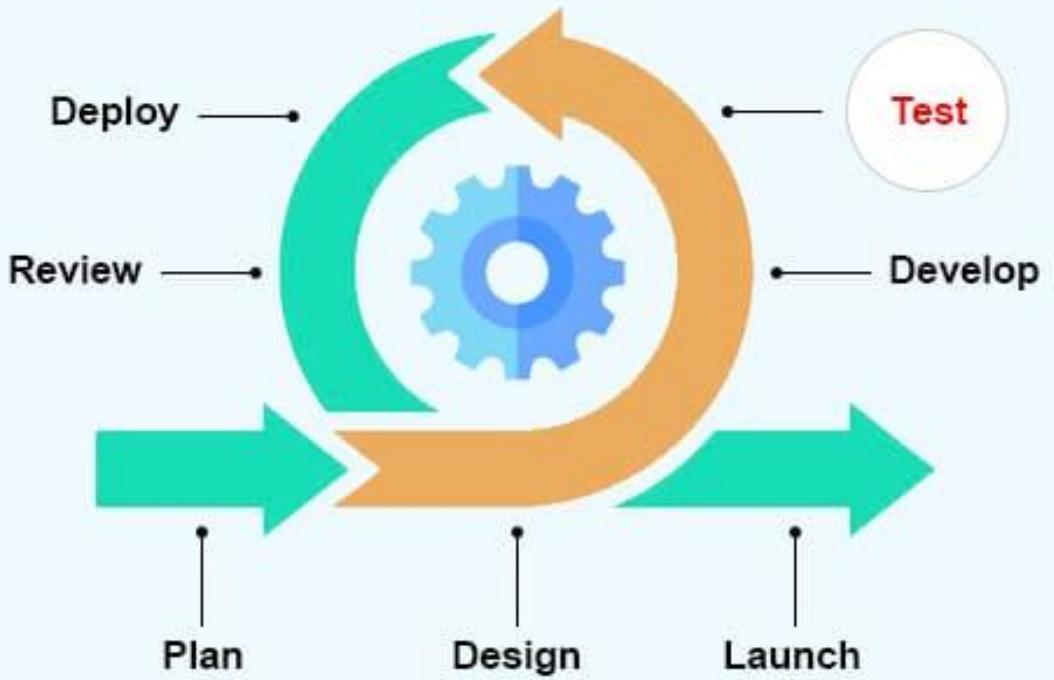
---

DEVOPS

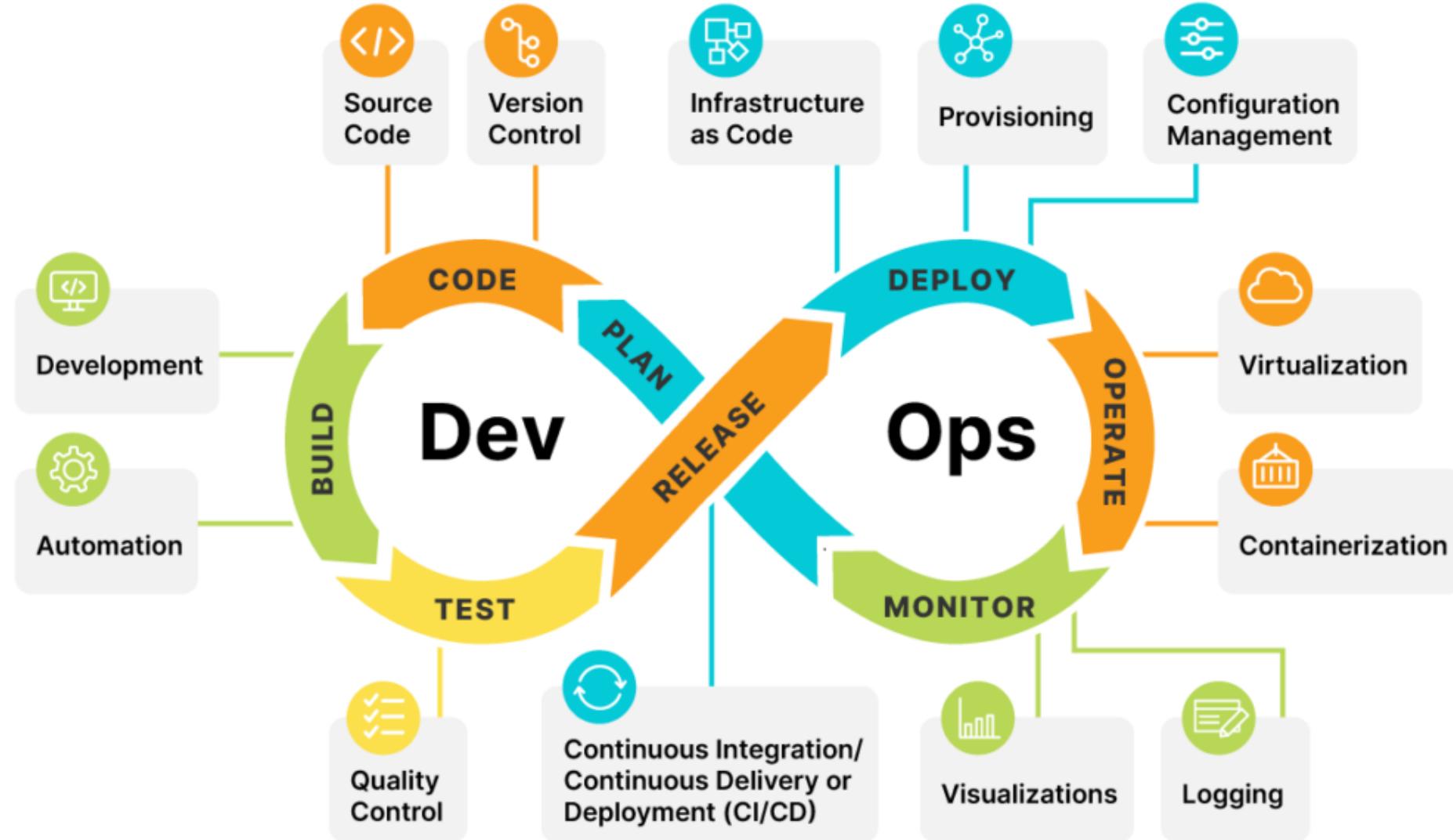


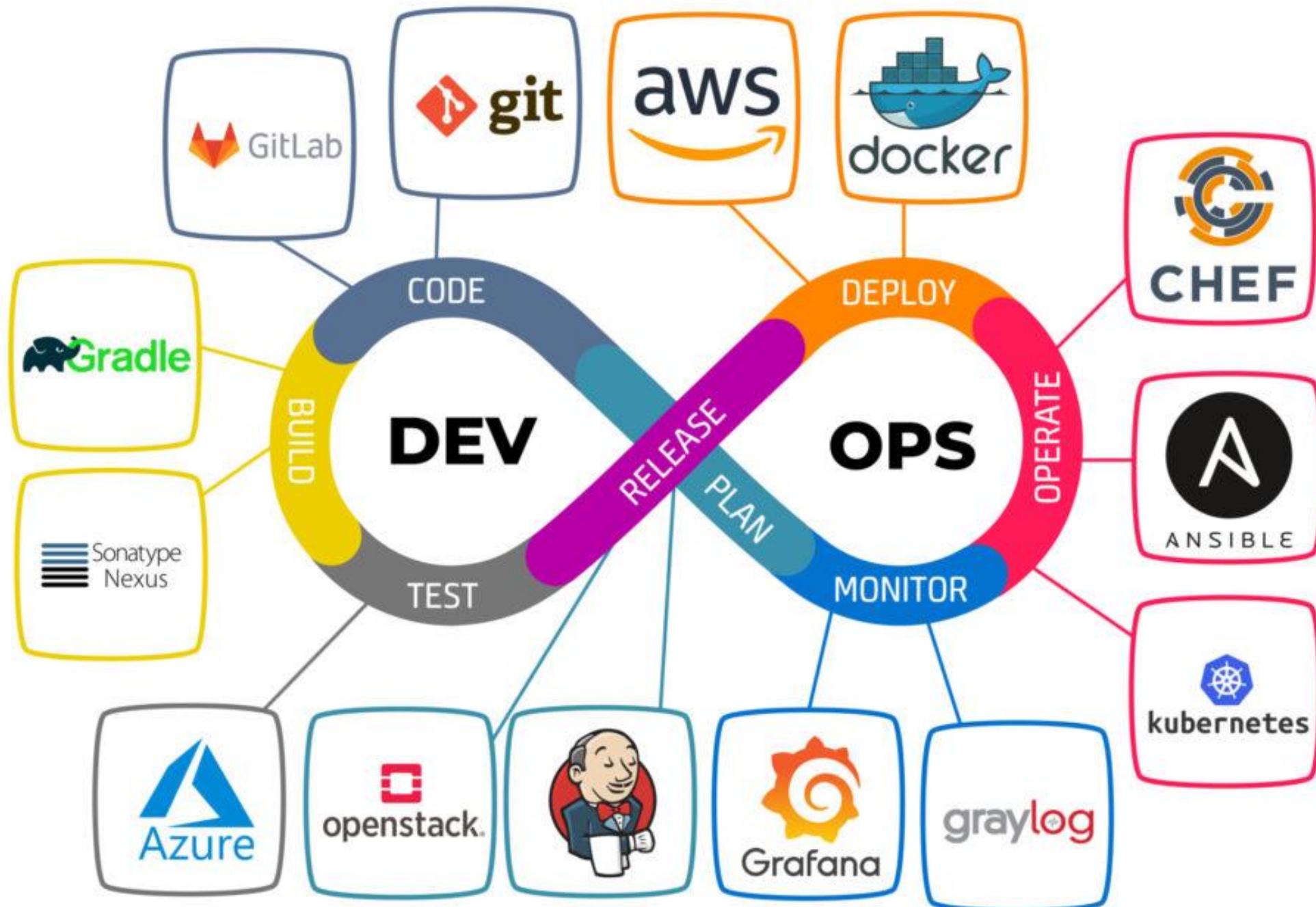
10 MINUTE BREAK

WHAT IS DEVOPS??

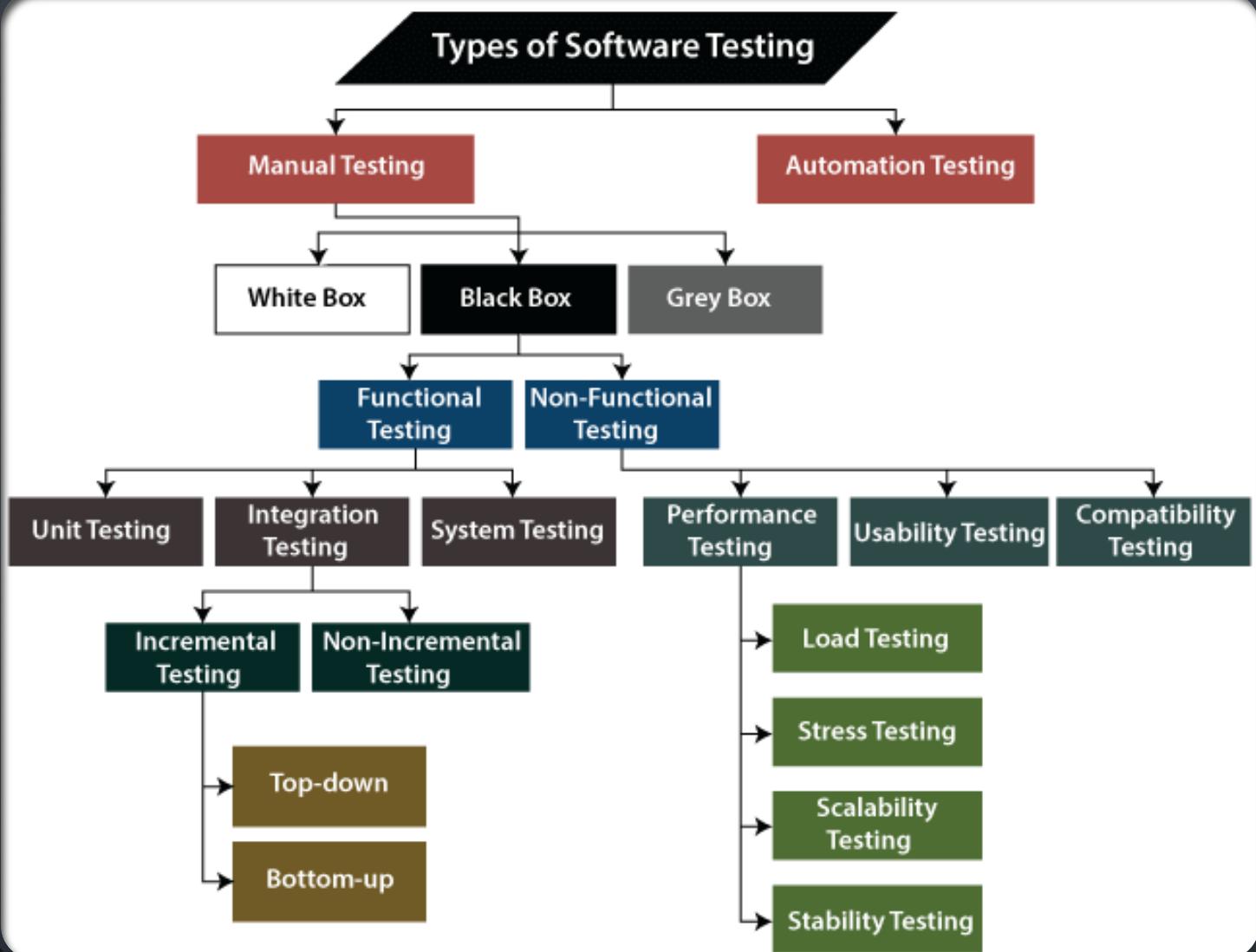


A DIFFERENT  
WAY TO  
LOOK AT THE  
SDLC





# TESTING STAGE ACTIVITIES



# FUNCTIONAL VS NON- FUNCTIONAL TESTING

## Functional

- x Unit testing
- x Integration testing
- x Sanity testing
- x System testing
- x Smoke testing
- x Interface testing
- x Regression testing
- x Acceptance testing

## Non- Functional

- x Performance testing
- x Load testing
- x Stress testing
- x Security testing
- x Volume testing
- x Compatibility testing
- x Reliability testing
- x Usability testing
- x Compliance testing
- x Recovery testing
- x Localization testing

### Black box testing

Testers do not know much how the product is built. test the product as a end user.

### White box testing

Internal perspective of a system.  
Mainly the code structure.  
Happens at unit level.

### Grey box testing

Testers have partial information of the product.  
Helps find bugs, clients are unaware of.

# WHY IS TESTING SO IMPORTANT?



PRE-RELEASE



EASY TO FIX  
PROBLEMS FOUND

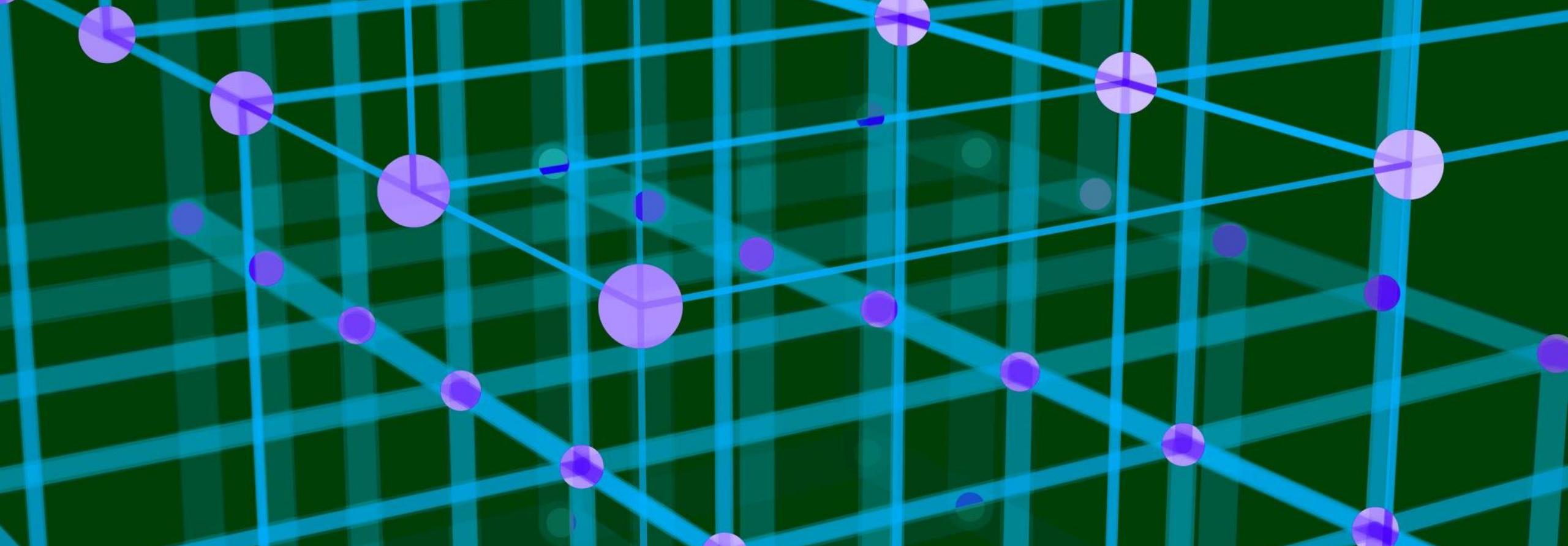


DISCOVERY OF  
MISSING PIECES



REVIEW  
ASSUMPTIONS

# OUTCOMES



# DEPLOYMENT PHASE

# DEPLOYMENT QUESTIONS

- HOW WILL OUR USERS GET THIS APP?
  - APP STORE?
  - DIRECT DOWNLOAD/SIDELOADING?
- HOW WILL USERS GET FUTURE UPDATES?

# DEPLOYMENT ROLES

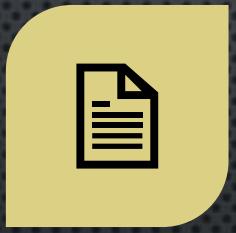
- DATA ADMINISTRATOR
- DEVOPS



# DEPLOYMENT ACTIVITIES



FINAL TESTING STEPS



INITIAL  
DOCUMENTATION  
PHASE



MAKING SOFTWARE  
PACKAGE AVAILABLE  
TO CUSTOMERS



CUSTOMIZATION AND  
DEVELOPMENT  
TESTING



INSTALLATION TO  
STAGING SERVERS



DOCUMENTATION  
DOCUMENTATION  
DOCUMENTATION

# USER ACCEPTANCE TESTING



The background of the slide features a complex network of white dots connected by thin white lines, set against a gradient background that transitions from dark purple at the top to black at the bottom.

MAINTENANCE PHASE

# MAINTENANCE ROLES



Users



Testers



Support managers



Field Support Engineers

# MAINTENANCE ACTIVITIES

- BREAK/FIX
- PRODUCT SUPPORT
- DEMONSTRATION/FIELD SUPPORT



# COMMUNICATION



WHERE ARE OUR “FRONT DOORS”?

THEN WHAT?

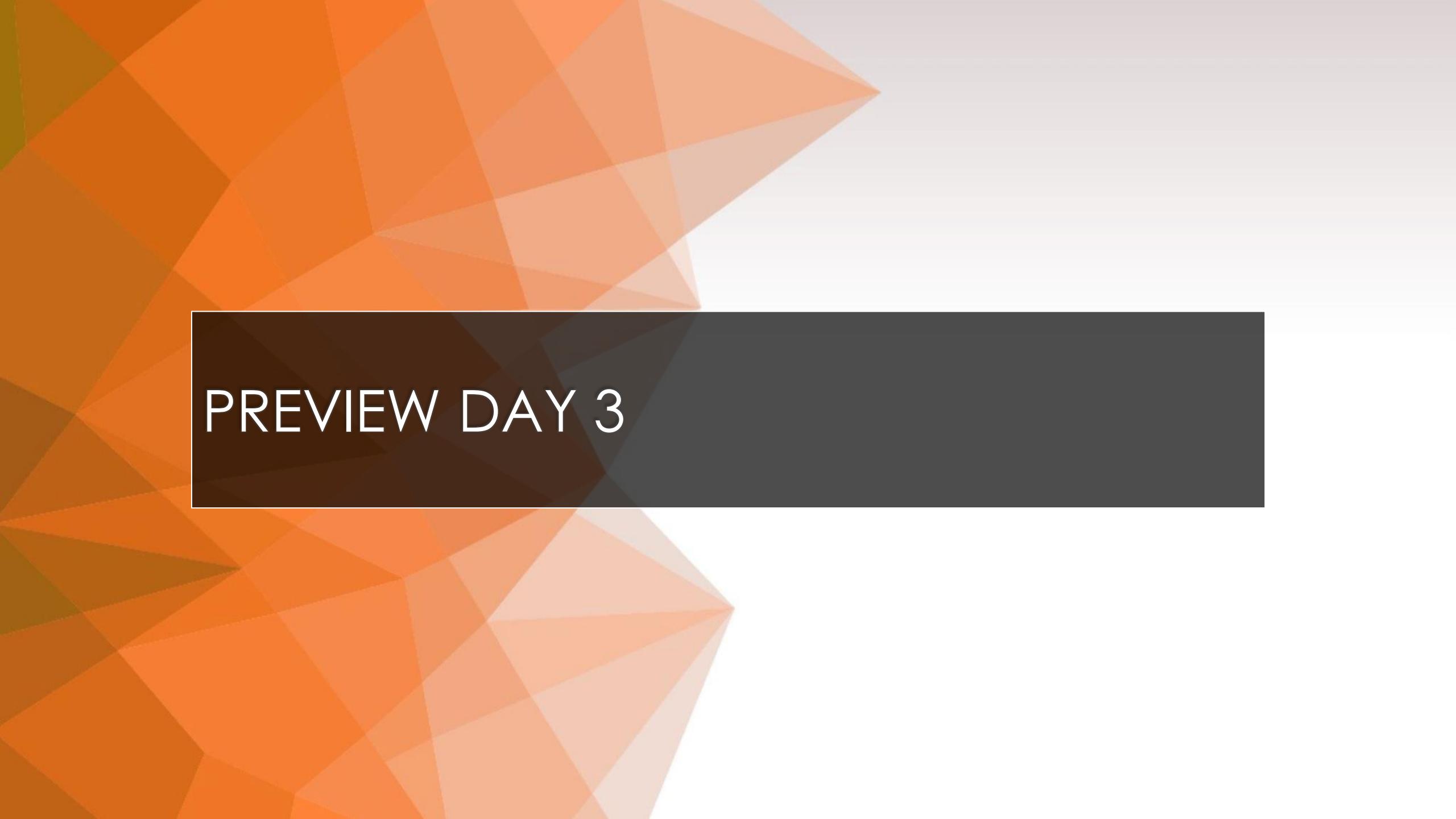


THIS IS THE SONG  
THAT DOESN'T END

# REVIEW DAY 2

QUESTION OR  
CLARIFICATIONS?



The background features a complex, abstract geometric pattern composed of numerous triangles. These triangles are primarily colored in shades of orange, from bright tangerine to deep burnt orange, and some are accented with darker brown or tan. The arrangement is organic and non-repeating, creating a sense of depth and movement.

PREVIEW DAY 3

SEE YOU NEXT TIME!