

INTRODUCTION TO SOFTWARE BUSINESS PRODUCT MANAGEMENT

Week 1 Day 2

Led by: Emily Crose

for

Oakland University

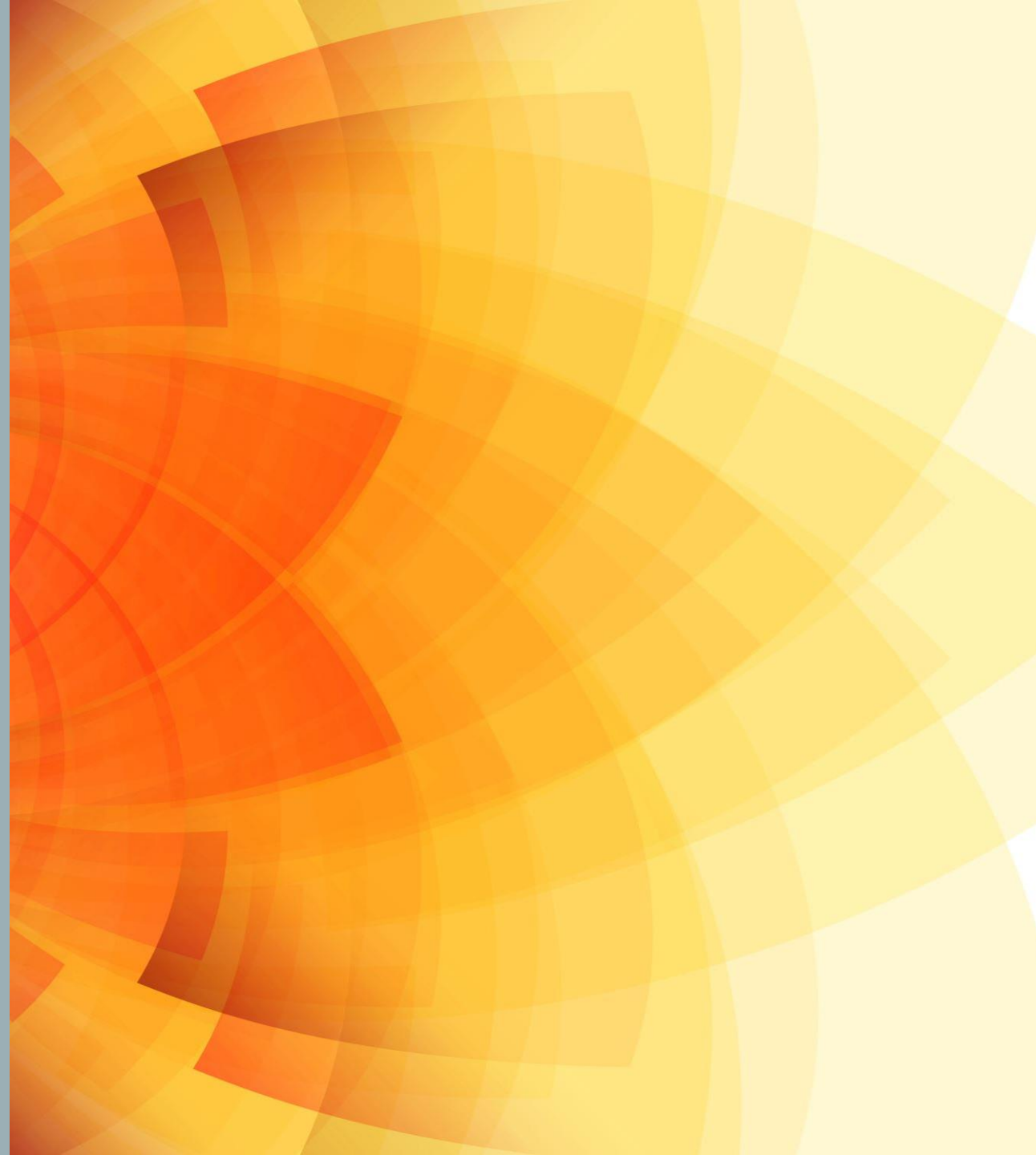


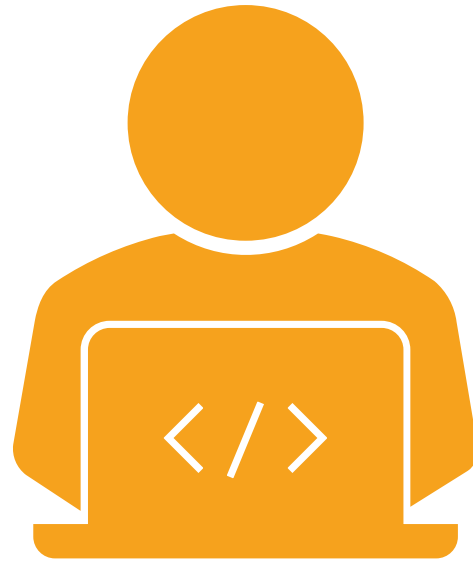
DAY 1 RECAP

QUESTION OR
CLARIFICATIONS?



MODERN SOFTWARE DEVELOPMENT CONCEPTS





WHAT IS A PROGRAM *REALLY*?

PROGRAM == CODE



CODE == INSTRUCTIONS

PARTS OF A PROGRAM

```
1
2 #include<stdio.h>
3 #include<conio.h>
4 int addNumbers(int a, int b); // function prototype
5
6 int main()
7 {
8     int n1,n2,sum;
9
10    printf(" \n Enter First Number : ");
11    scanf("%d",&n1);
12    printf(" \n Enter Second Number : ");
13    scanf("%d",&n2);
14    sum = addNumbers(n1, n2); // function call
15
16    printf(" \n Sum of two number = %d",sum);
17    getch();
18    return 0;
19 }
20 int addNumbers(int a,int b) // function definition
21 {
22     int result;
23     result = a+b;
24     return result; // return statement
25 }
```

Header Files

Function Prototype

Main Function

Variable Declaration

Pre Defined Function Call

User Defined Function Call

Function Declaration


Function Body



SCRIPTS VS. BINARIES

Compiled Language VS Interpreted Language

Comparison Chart

Compiled Language	Interpreted Language
The code of compiled languages can be executed directly by the computer's CPU.	A program written in an interpreted language is not compiled, it is interpreted.
The source code must be transformed into machine readable instructions prior to execution.	It does not compile the source code into machine language prior to running the program.
Compiled programs run faster than interpreted programs.	Interpreted programs can be modified while the program is running.
Delivers better performance.	Delivers relatively slower performance.
C, Fortran, and COBOL are languages used to produce compiled programs.	Java and C# are compiled into bytecode, the virtual interpreted language. 

COMPILED VS INTERPRETED SIMPLIFIED

Compiled

C, C++, Go, Fortran, Pascal

Language

"Compiling"

Machine Code

Ready to Run!

Interpreted

Python, PHP, Ruby, JavaScript

Language

Ready to Run!

"Interpreting"

Virtual Machine

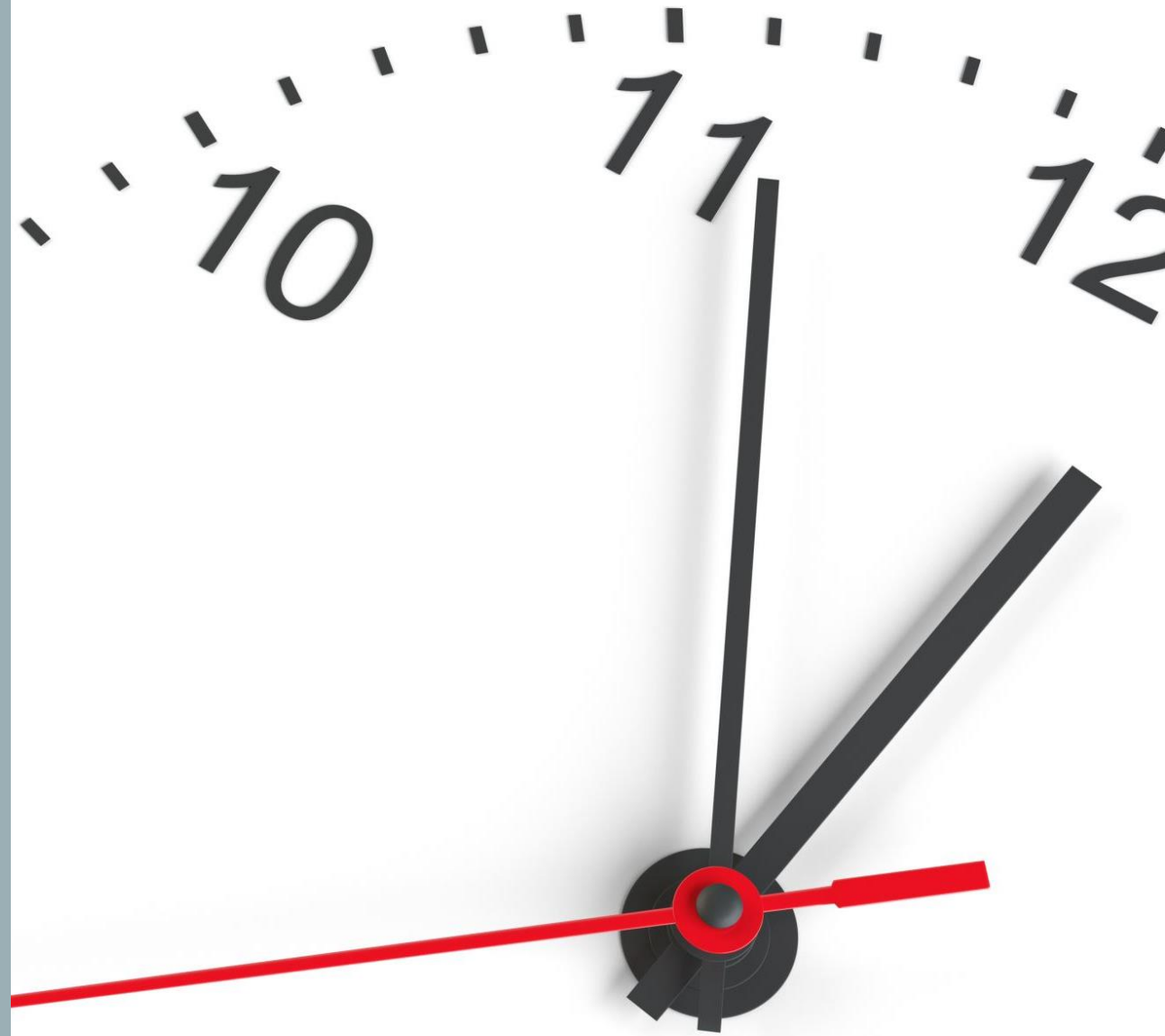
Machine Code

LET'S LOOK AT SOME CODE

<https://github.com/hexa-decim8/girltalk/blob/master/girltalk.sh>



10 MINUTE BREAK



HOW A PROGRAM WORKS



HOW FIREFOX WORKS

1. Running program gets its own area in RAM to hold code and data. Copy instructions to there.

CPU

RAM

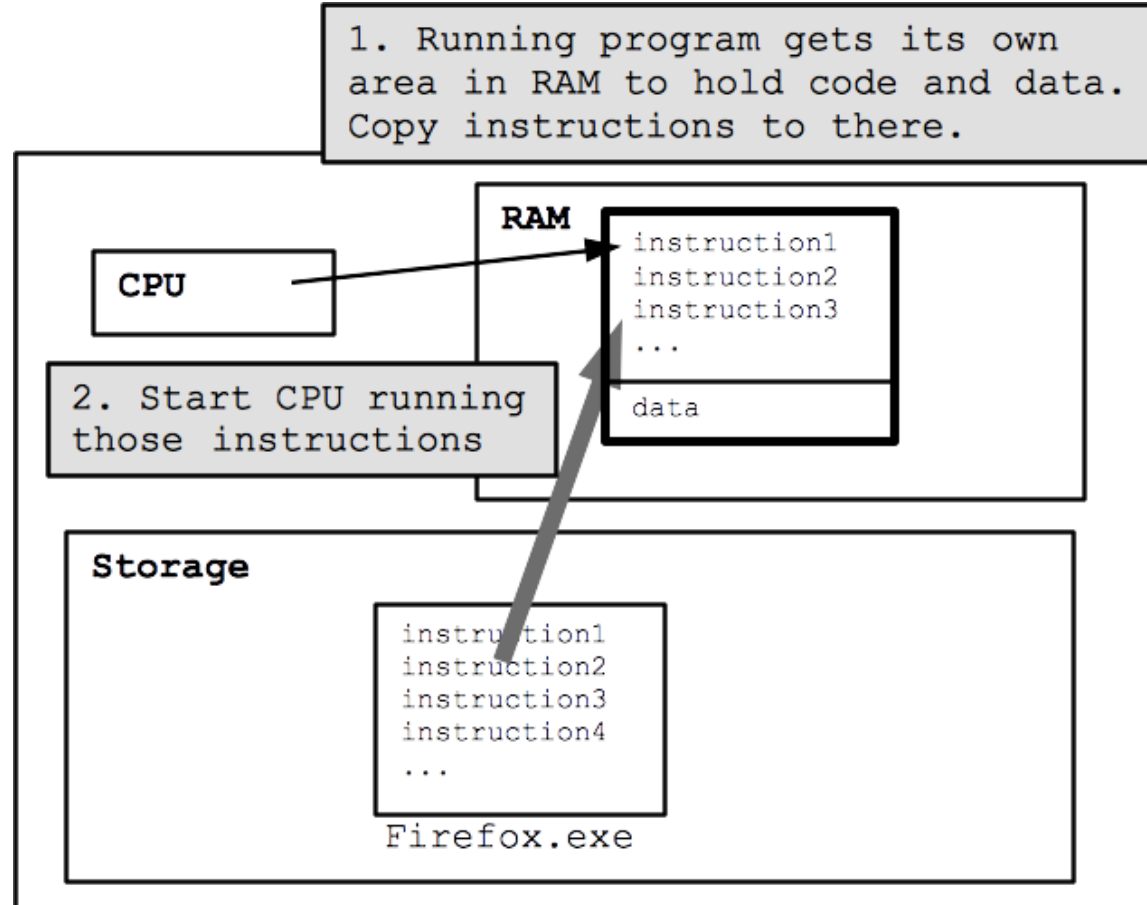
instruction1
instruction2
instruction3
...
data

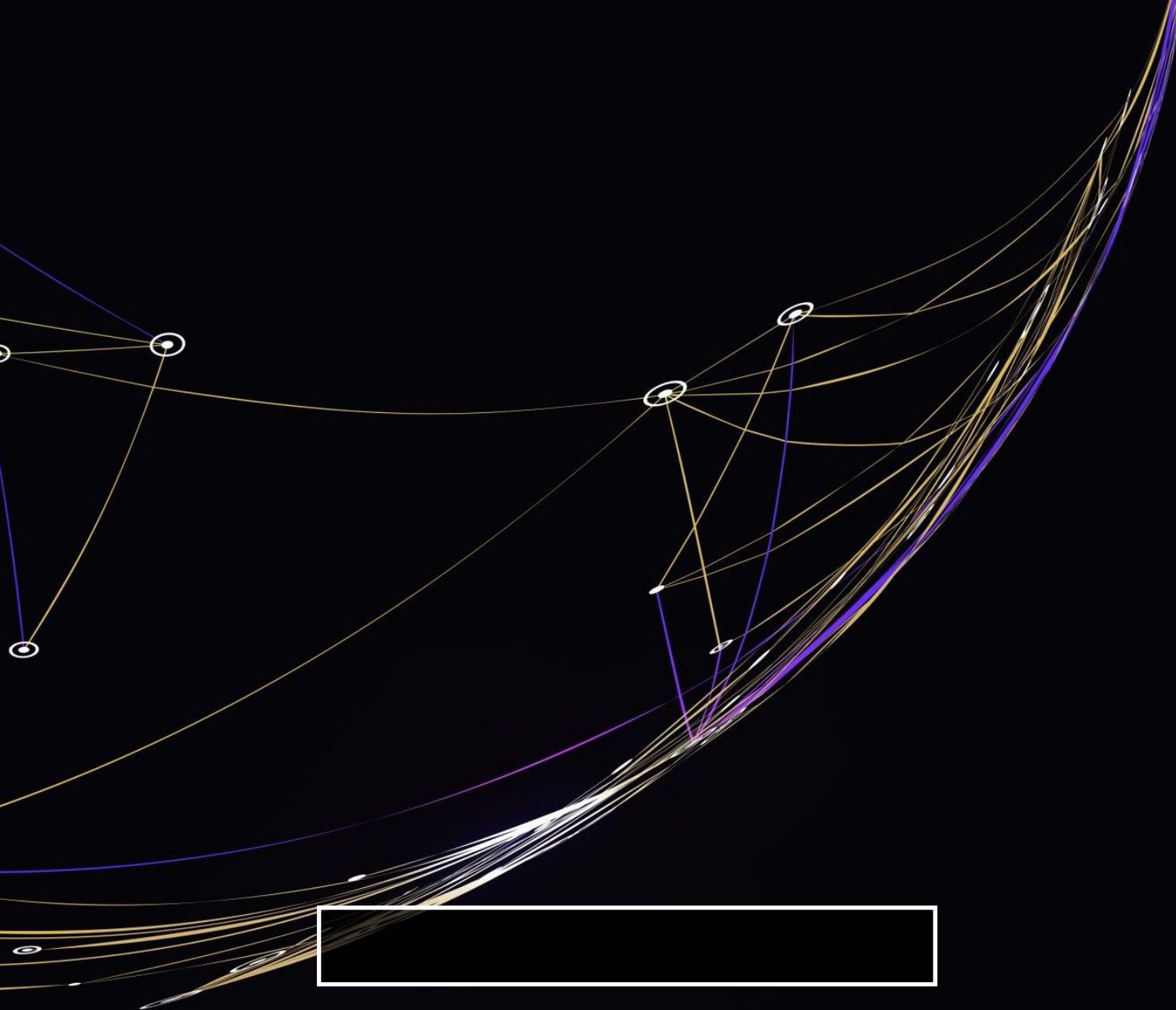
2. Start CPU running those instructions

Storage

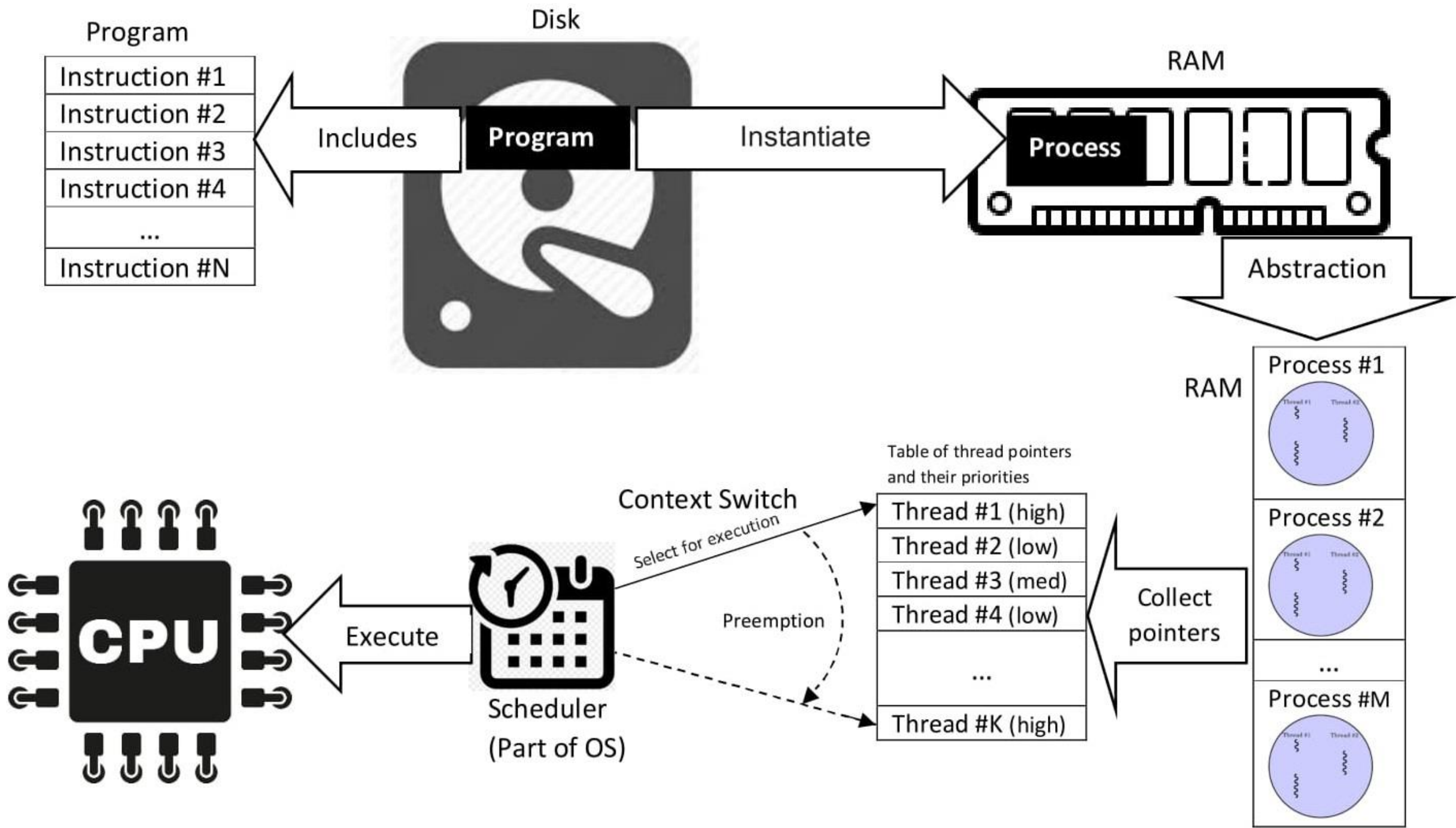
instruction1
instruction2
instruction3
instruction4
...

Firefox.exe





DETAILED
EXECUTION



COMPILING



What is compiling?

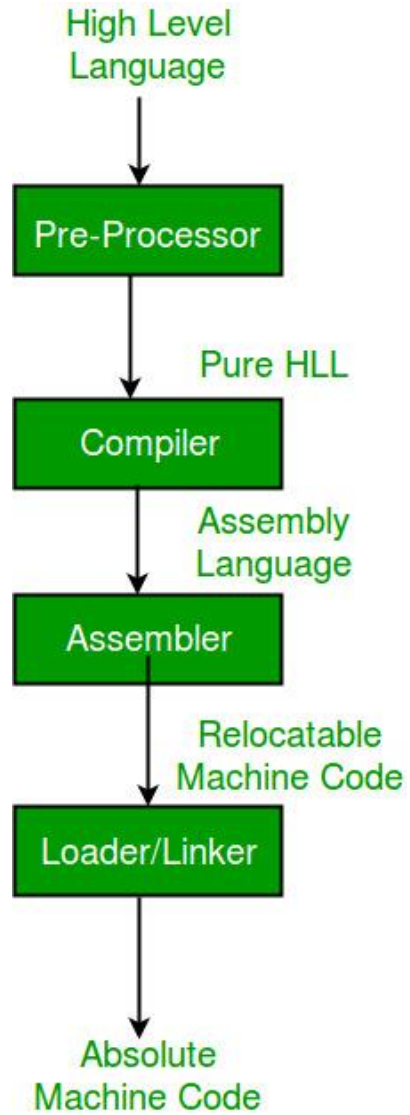


A set of instructions that has been compiled becomes an executable 'binary'

We may just call this an 'executable'

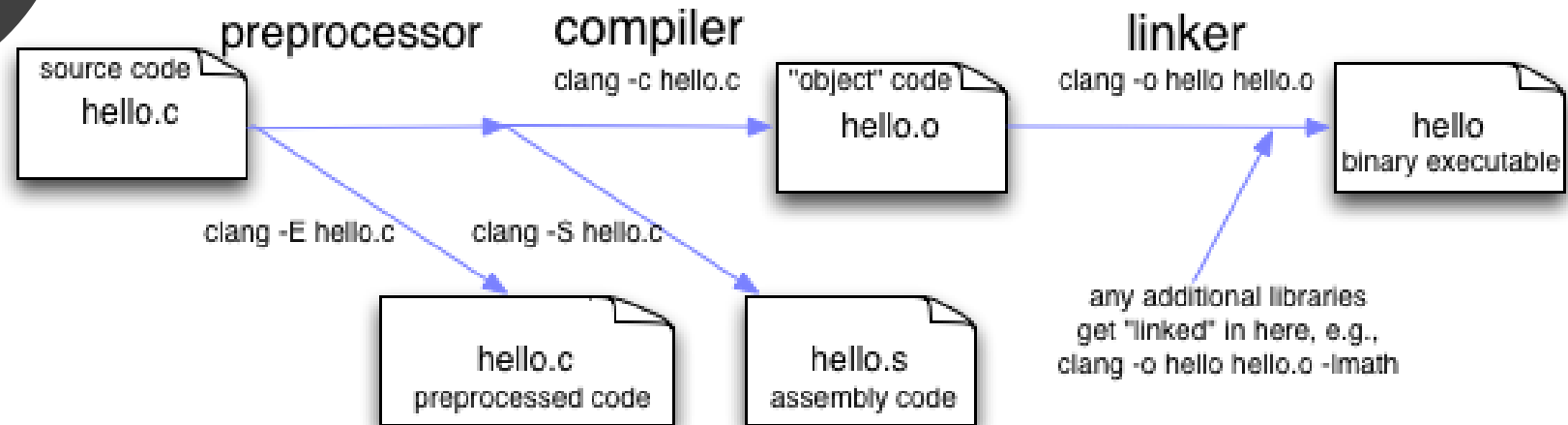


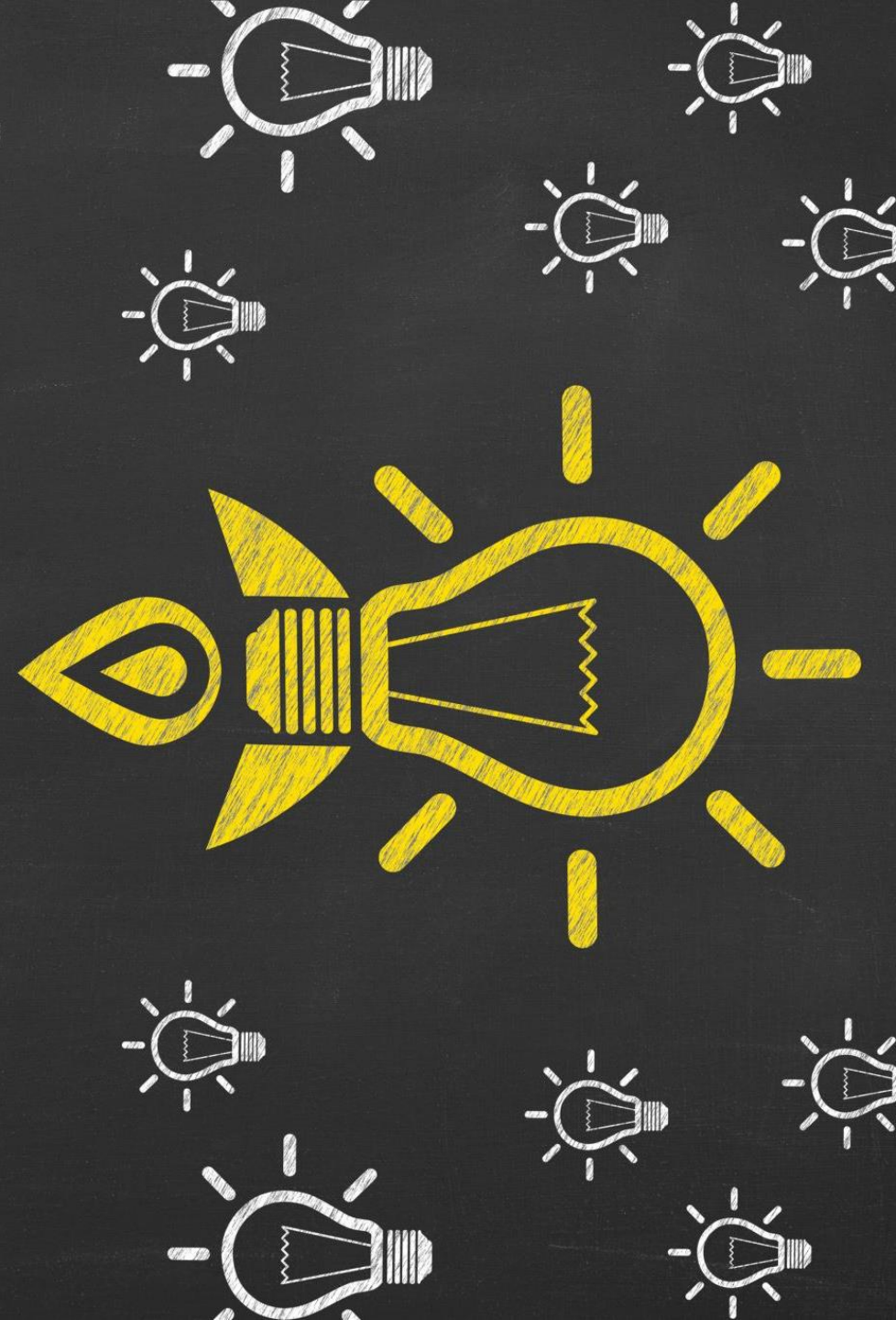
Scripts do not compile!



COMPILE PROCESS

COMPILING C IN PRACTICE





IMAGINE AN
OUTSTANDING APP

The background of the image is a dark teal color, overlaid with a repeating pattern of speech bubbles. Each speech bubble is a different color (red, yellow, purple, grey) and contains a large, dark blue question mark. The bubbles are scattered across the entire frame, creating a textured, question-filled background.

WHY WOULD WE MAKE AN APP?



HOW DO WE BUILD A
QUALITY APP?

WHAT IS THE PROBLEM?

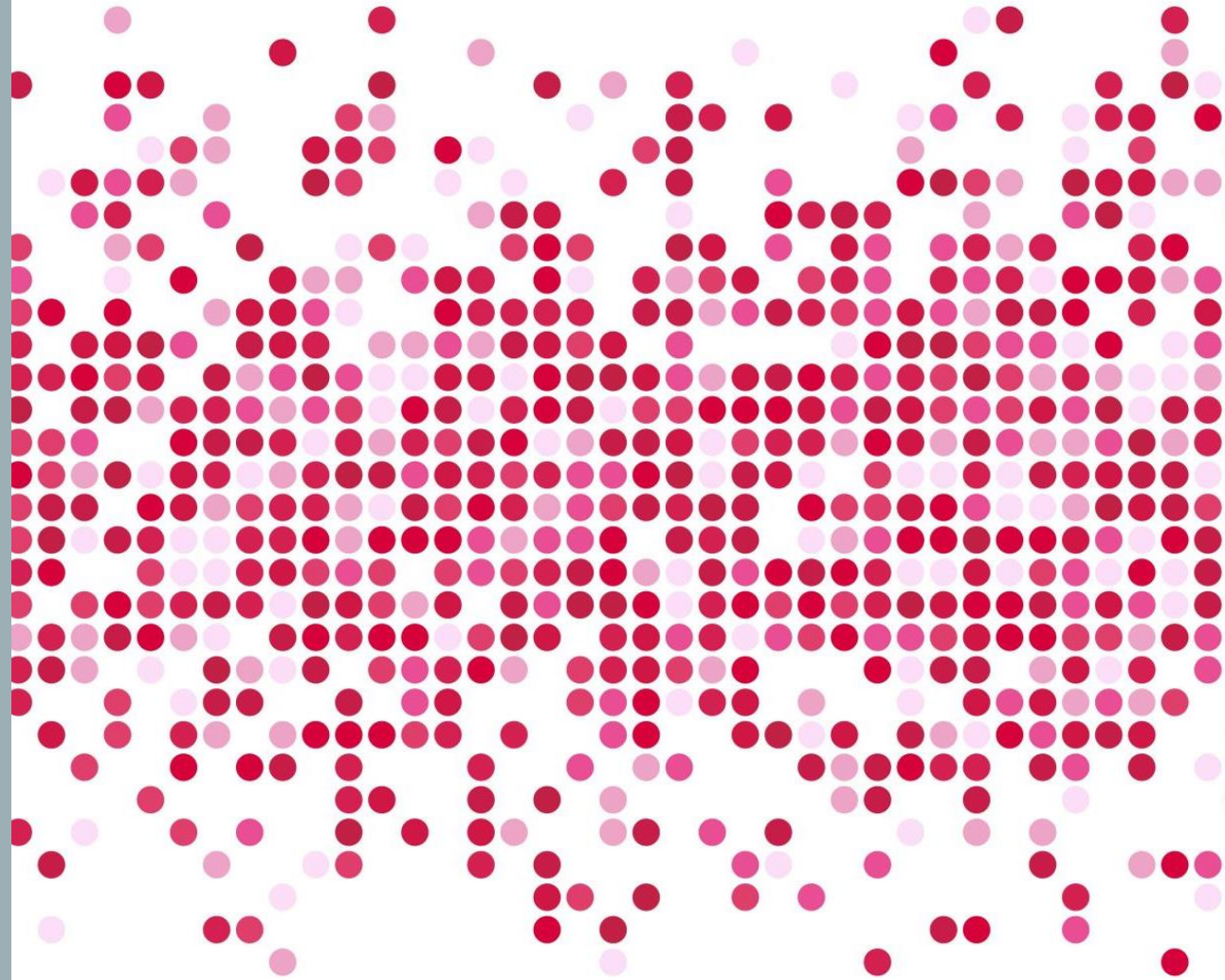
- What problems does this app solve?
- How does it solve these problems?





SYNTHESIZING A SOLUTION

GENERATING REQUIREMENTS





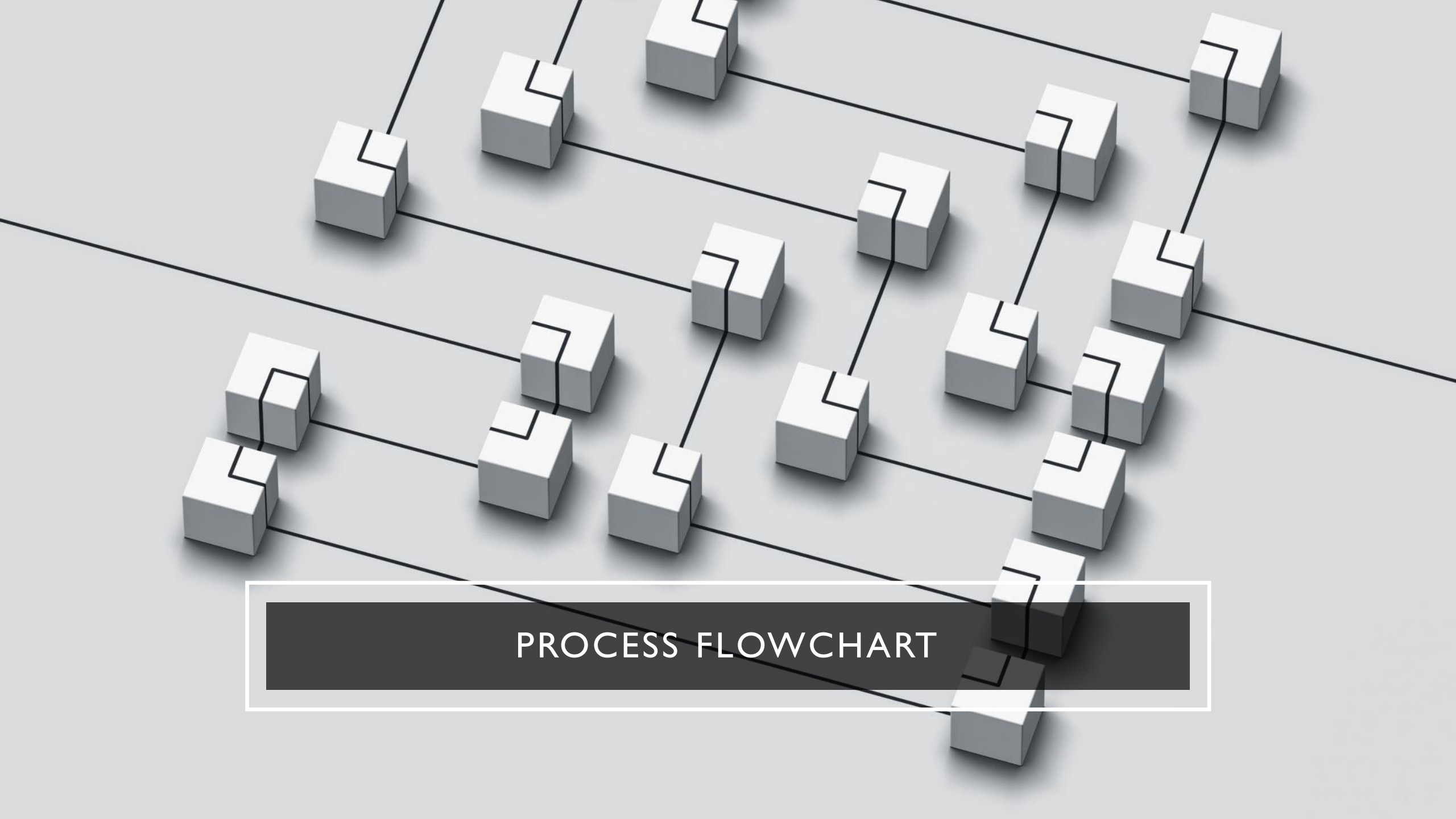
WHO MAKES SOFTWARE?





DEVELOPERS

“PSEUDO” CODE

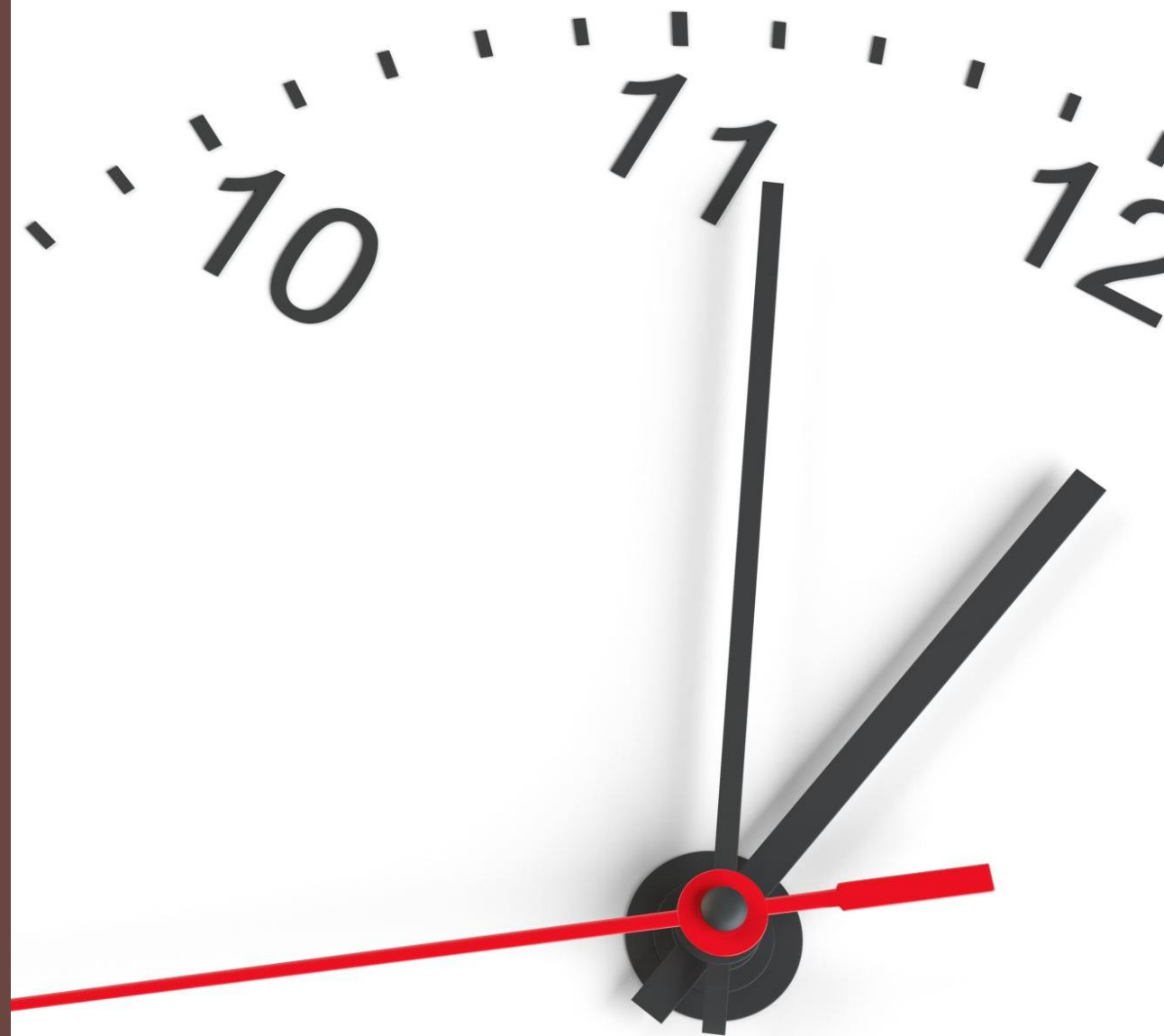


PROCESS FLOWCHART



SEPARATION OF DUTIES

10 MINUTE BREAK



TESTING

Does the code work as expected?

Does the code solve the problem we identified?

What does the code work on?

Does the code function with all of our supported systems?

Do we even know what possible systems exist in the field?



CODE DISTRIBUTION



PROBLEM DISCOVERY



WHO ARE OUR USERS?



Do we know who they are?



**What kind of sensitive information
might we be processing?**

Personally identifiable information?

HIPAA information?

Legally sensitive?

SUPPORTING THOSE CUSTOMERS

How do our customers
contact us?

How do we contact
them?

Do we have field
support?

What is our process for
addressing customer
concerns/complains?

What is our process for
addressing customer
feature requests?
• Ticketing system?

QUESTIONS?



REVIEW DAY 2



PREVIEW DAY 3