REFACTORING

HOW TO DO IT
WITHOUT HARMING YOURSELF

CEZARY SANECKI

ABOUT ME

EX-CIVIL ENGINEER

WJUG LEADER

BLOGGER - CEZARYSANECKI.PL

TECH LEADER IN DECERTO

REFACTORING

REFACTORING - A CHANGE MADE TO THE INTERNAL STRUCTURE OF SOFTWARE TO MAKE IT EASIER TO UNDERSTAND AND CHEAPIER TO MODIFY WITHOUT CHANGING ITS OBSERVABLE BEHAVIOUR.

MARTIN FOWLER

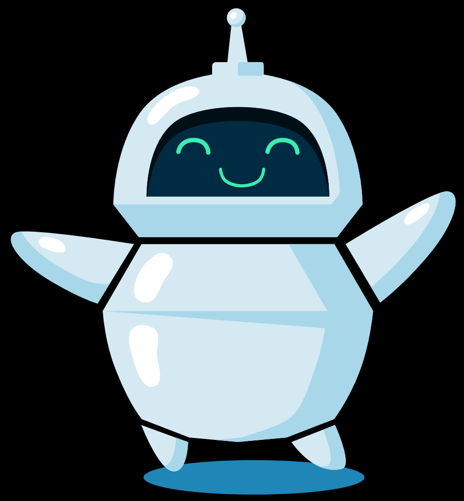
REFACTORING - A CHANGE MADE TO THE INTERNAL STRUCTURE OF SOFTWARE TO MAKE IT EASIER TO UNDERSTAND AND CHEAPIER TO MODIFY WITHOUT CHANGING ITS OBSERVABLE BEHAVIOUR.

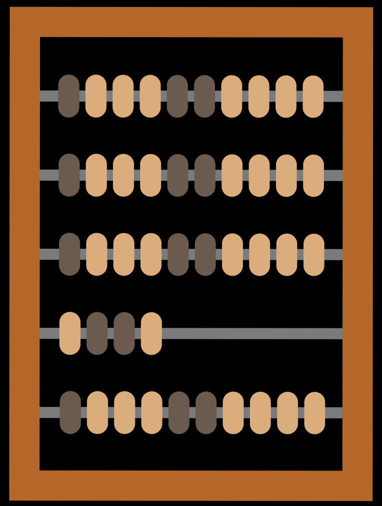
MARTIN FOWLER

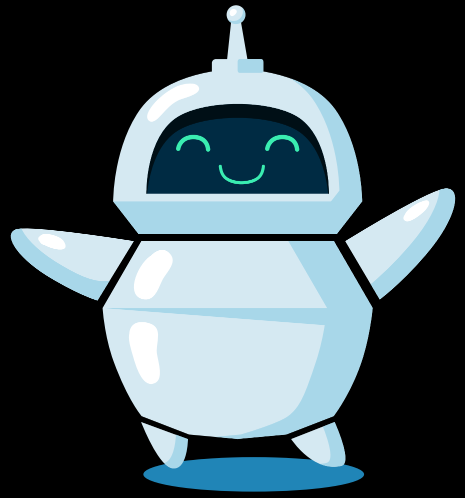
REFACTORING - A CHANGE MADE TO THE INTERNAL STRUCTURE OF SOFTWARE TO MAKE IT EASIER TO UNDERSTAND AND CHEAPIER TO MODIFY WITHOUT CHANGING ITS OBSERVABLE BEHAVIOUR.

MARTIN FOWLER

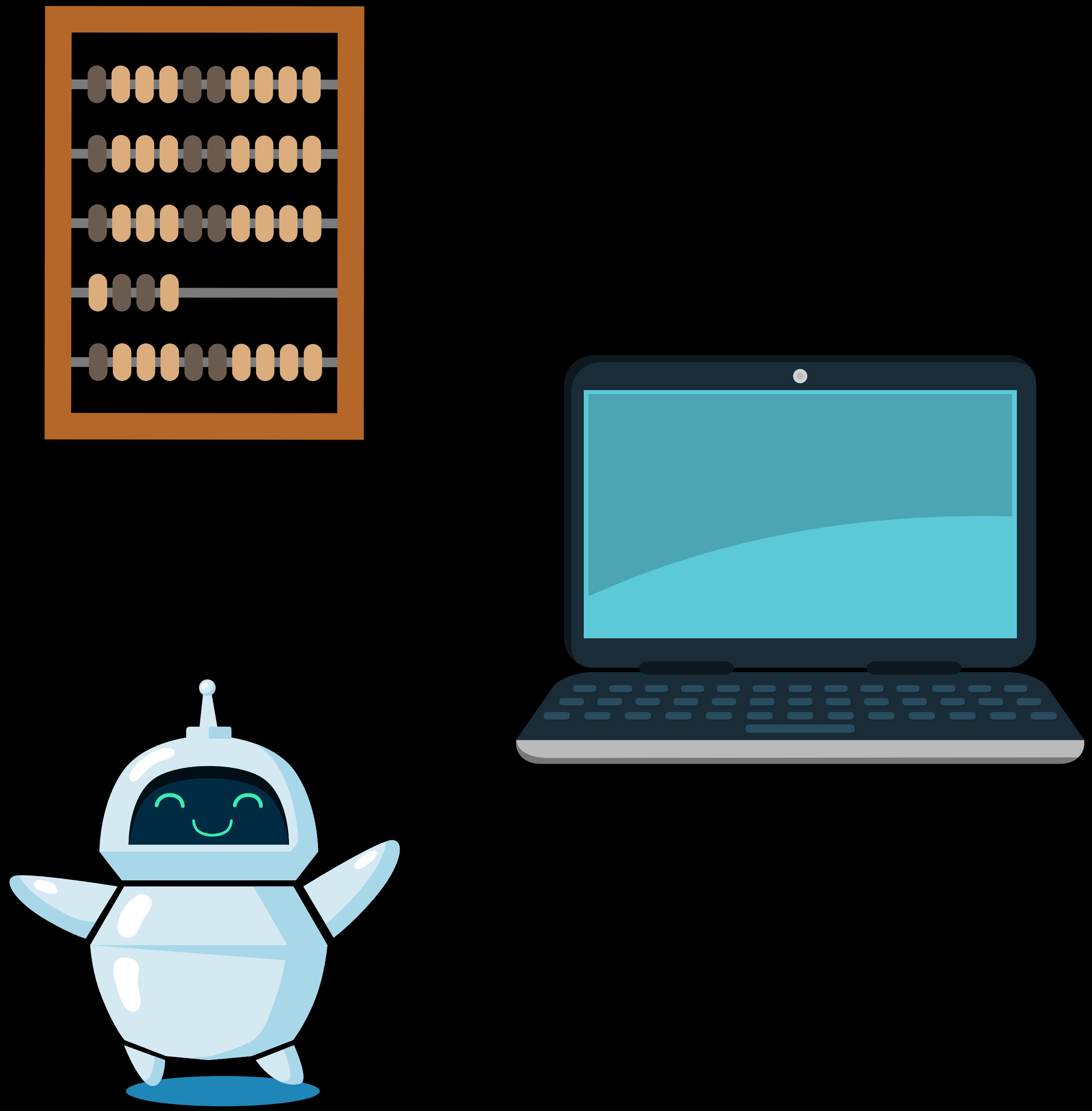
OBSERVABLE BEHAVIOR





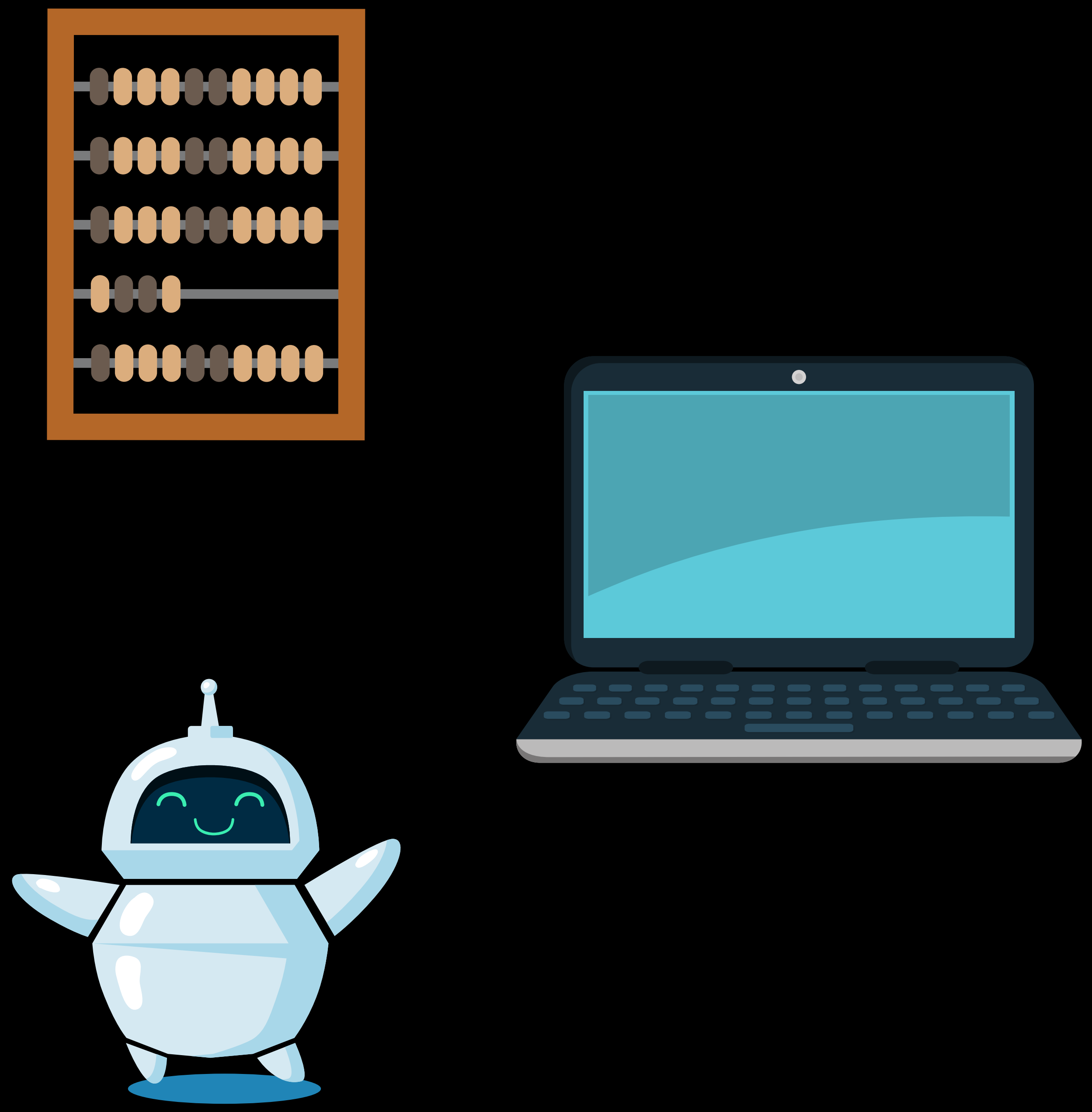


WHAT?



WHAT?

HOW?



PROJECT CHARACTERISTICS

FUNCTIONAL

NON-FUNCTIONAL

FUNCTIONAL

~~NON-FUNCTIONAL~~

FUNCTIONAL

OPERATIONAL

DEVELOPMENTAL

WHAT? FUNCTIONAL

HOW? OPERATIONAL

DEVELOPMENTAL

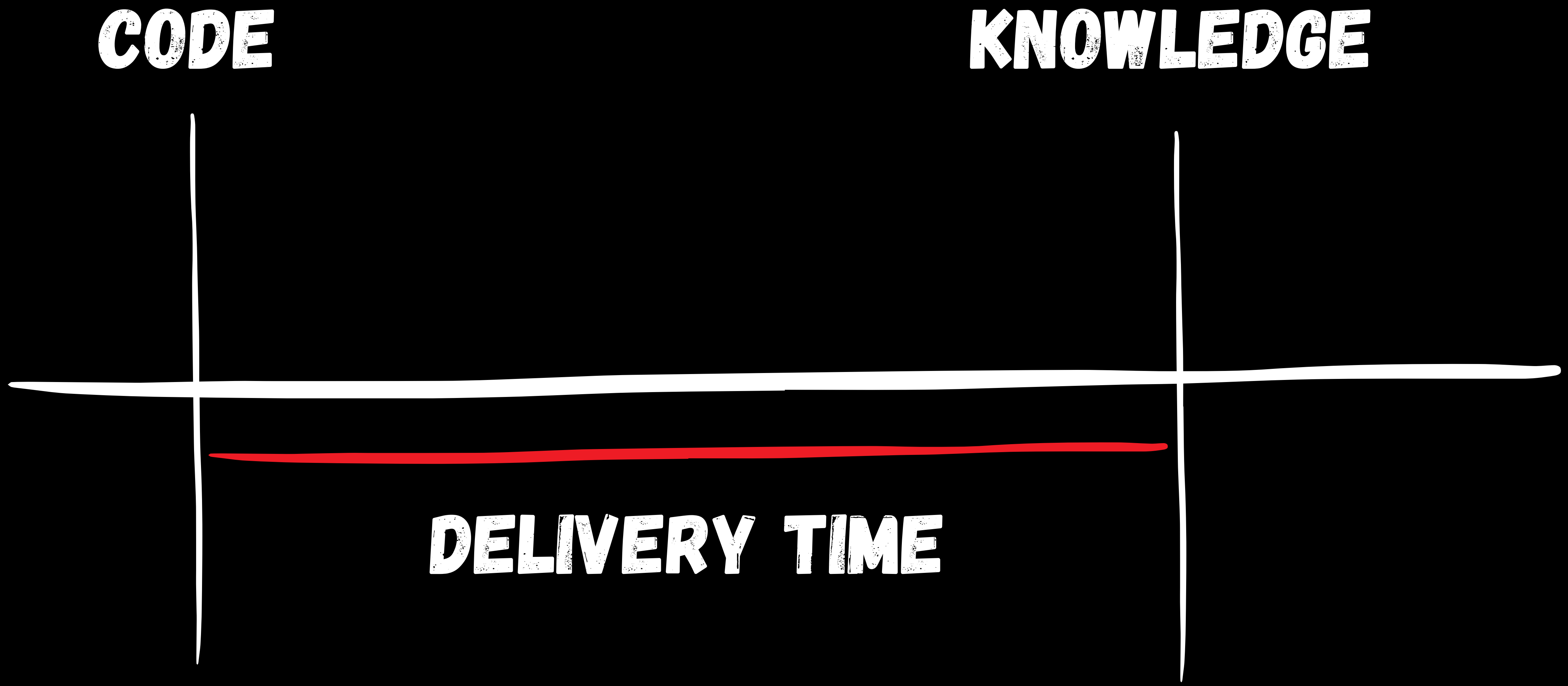
DEVELOPMENTAL CHARACTERISTICS

WHY DO WE NEED TO REFACTOR?

REFACTORING - A CHANGE MADE TO THE INTERNAL STRUCTURE OF SOFTWARE TO MAKE IT EASIER TO UNDERSTAND AND CHEAPIER TO MODIFY WITHOUT CHANGING ITS OBSERVABLE BEHAVIOUR.

MARTIN FOWLER

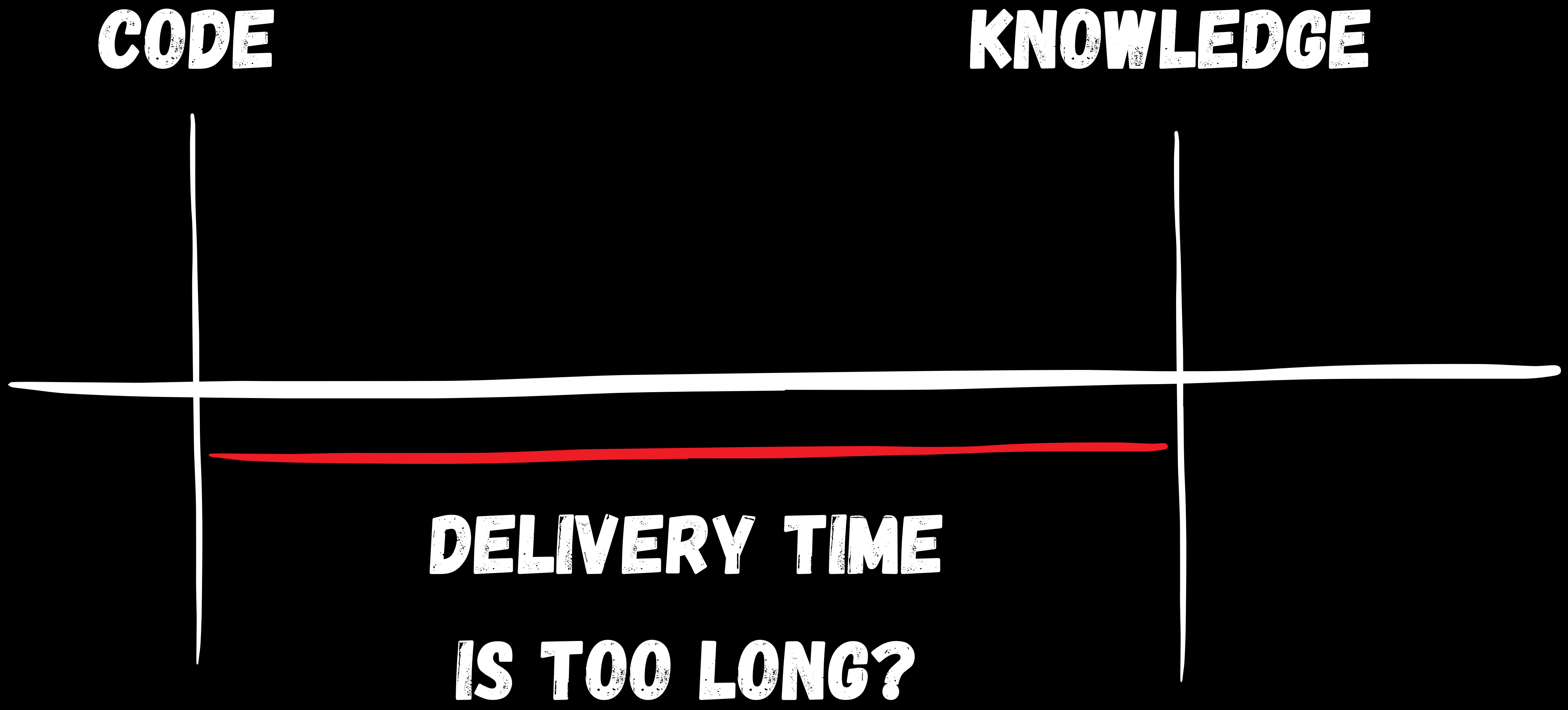
WE CODE KNOWLEDGE!



CODE

KNOWLEDGE

DELIVERY TIME



CODE

KNOWLEDGE

DELIVERY TIME

IS TOO LONG?

CHANGE IN REQUIREMENTS

LACK OF KNOWLEDGE

ASAP/TECHNICAL DEBT

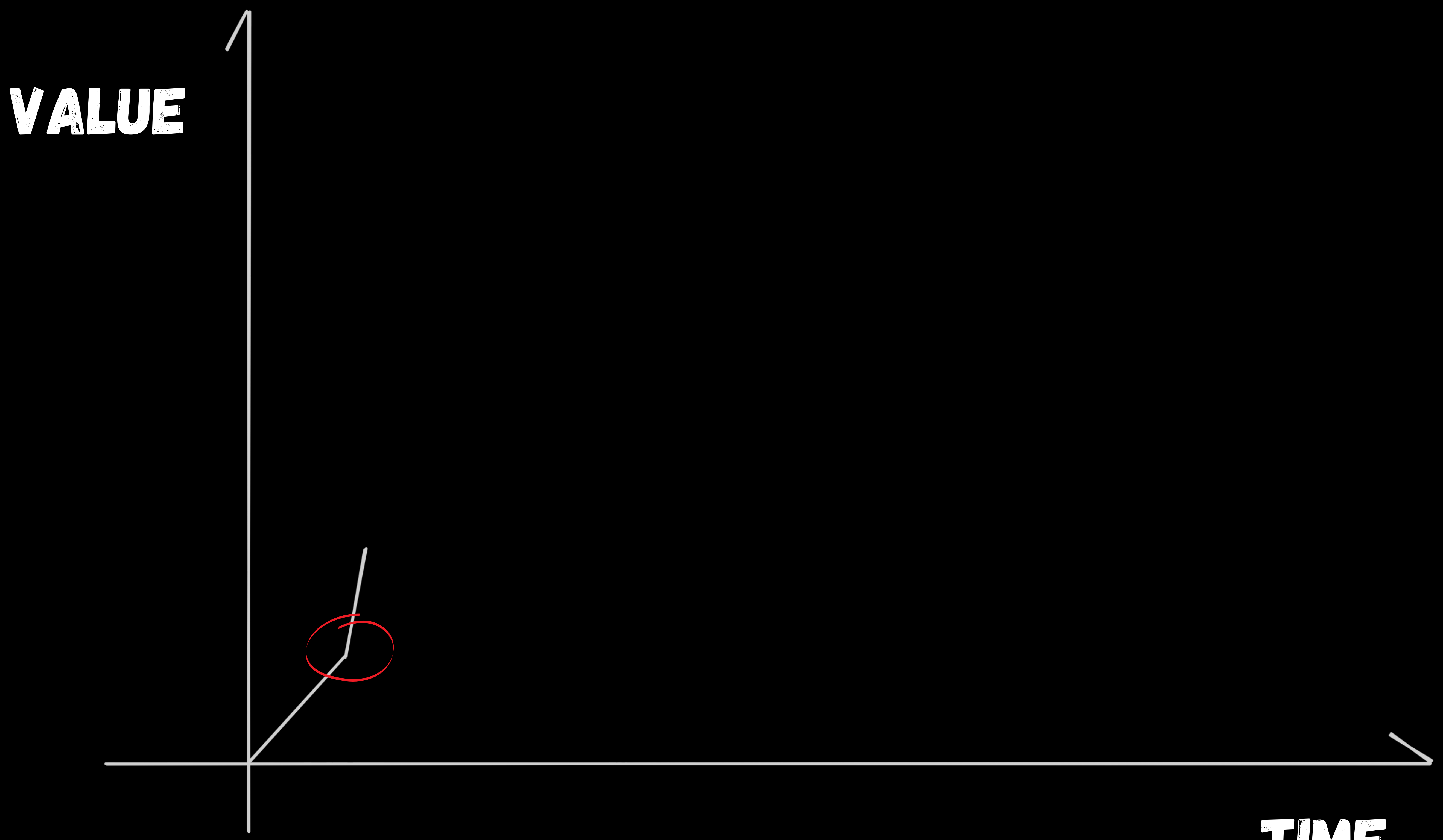
SOMETIMES WE CREATE OUR OWN COMPLEXITY.

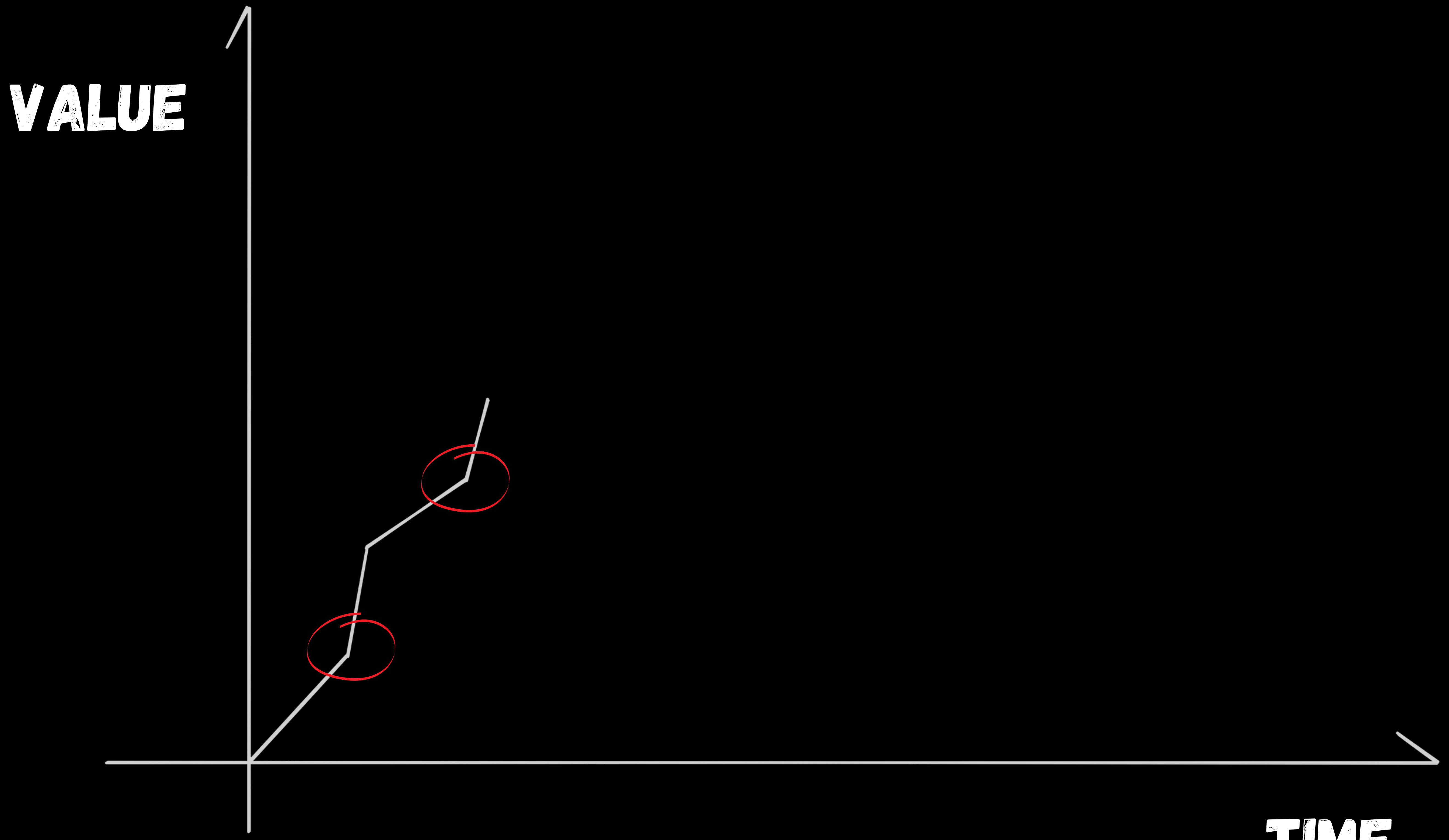
KEVLIN HENNEY

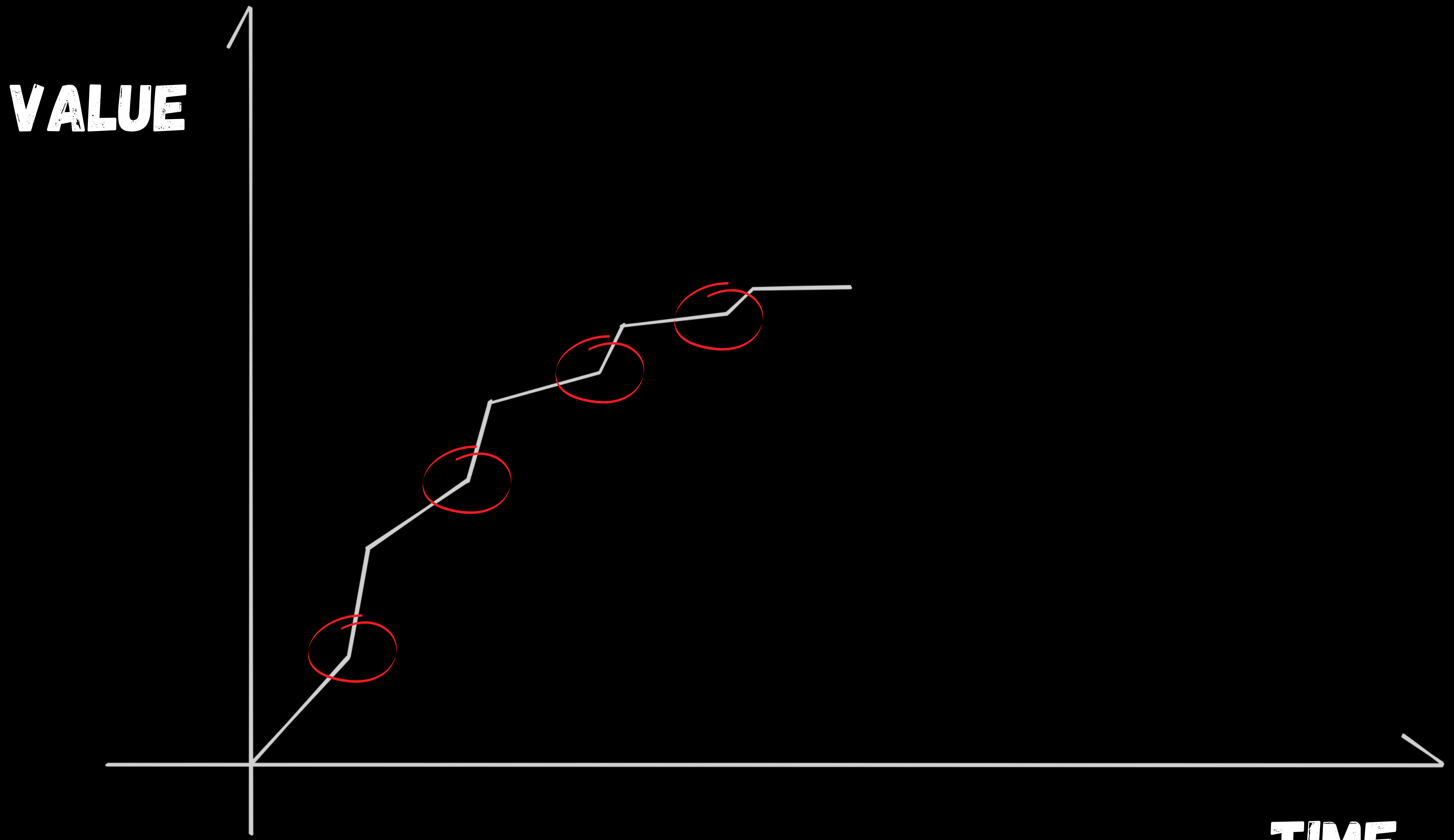
CDD

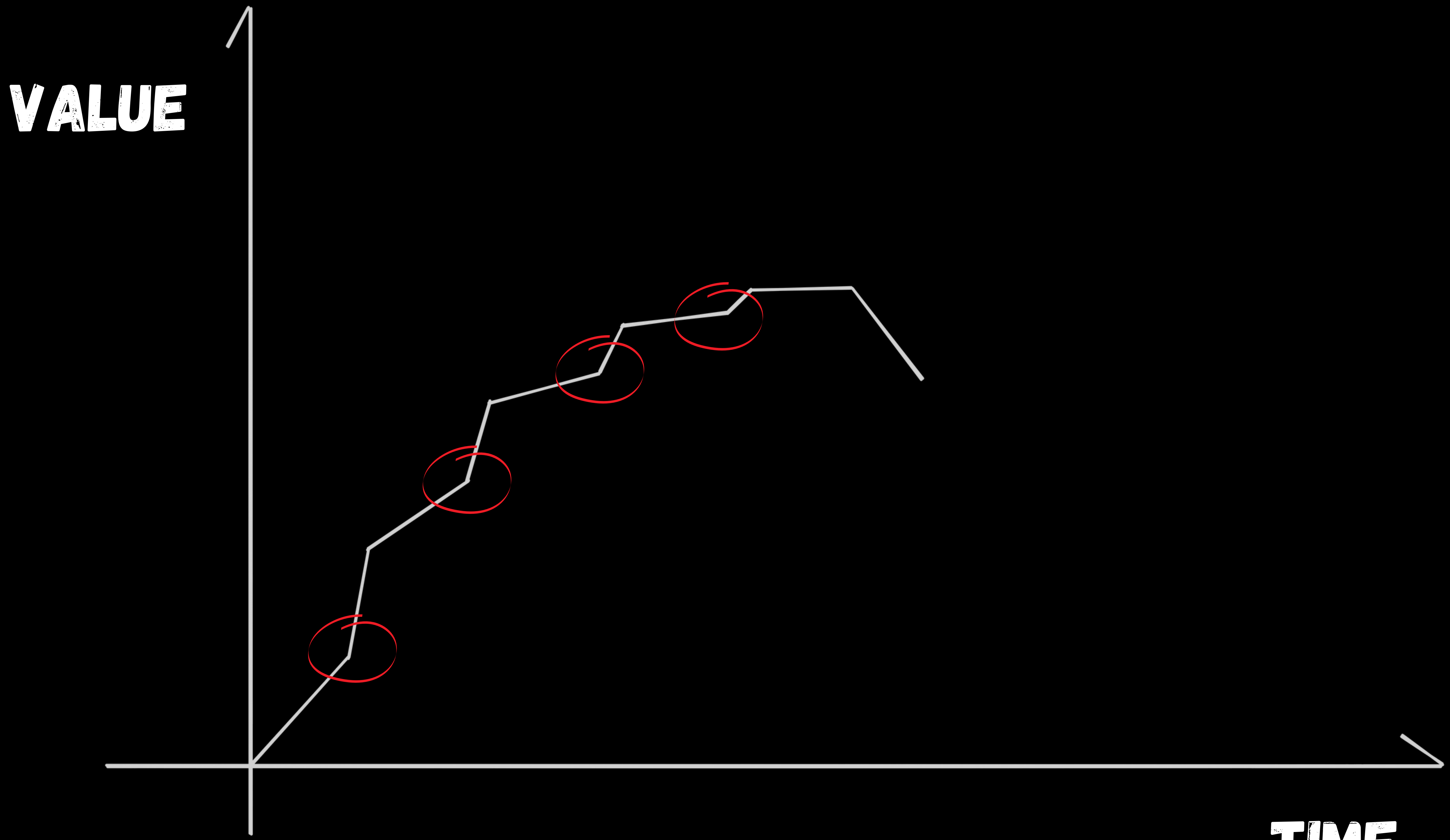
CV DRIVEN-DEVELOPMENT

TECHNICAL DEBT

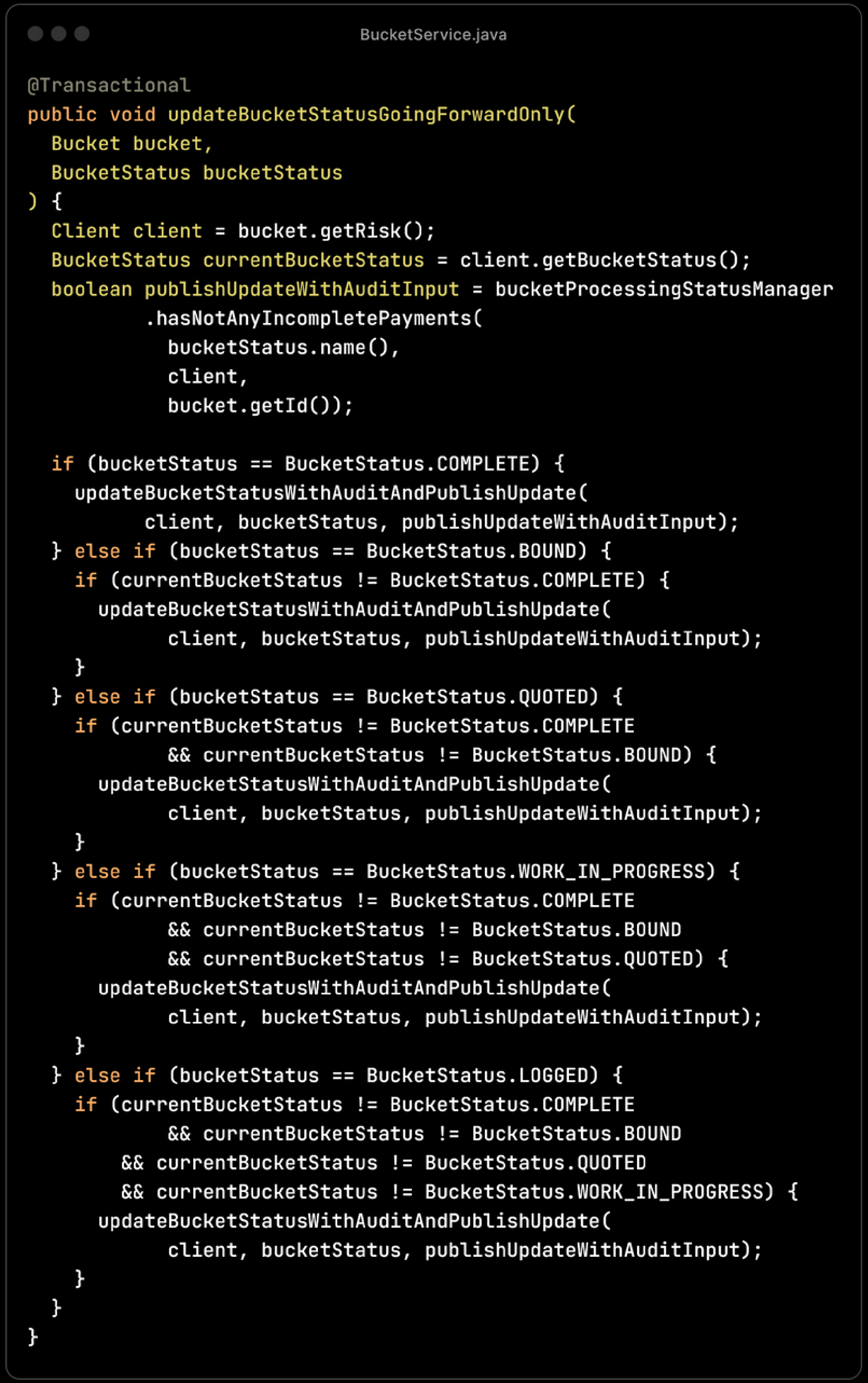

VALUE
TIME


VALUE
TIME


VALUE
TIME

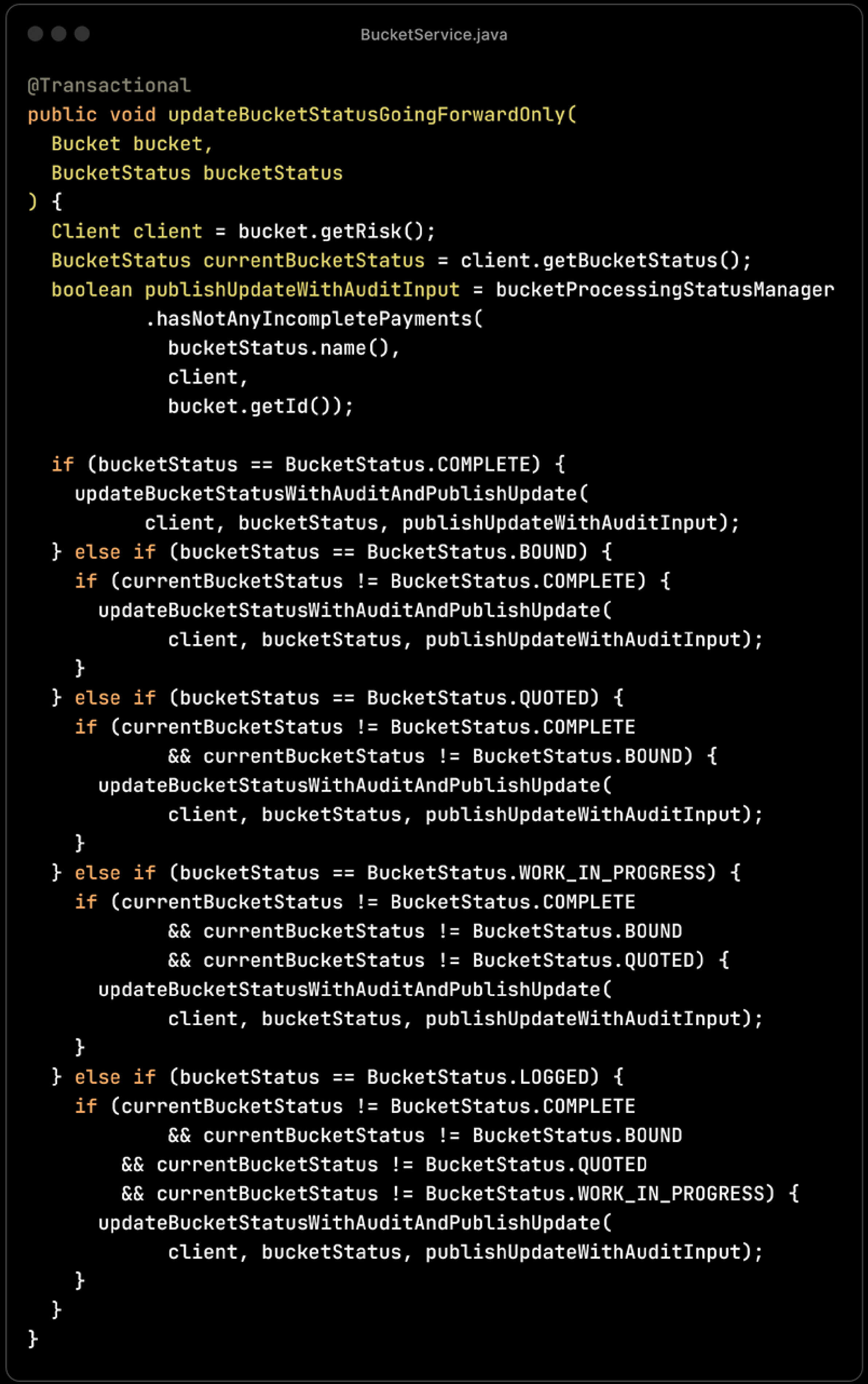

VALUE
TIME

WHEN AND WHAT DO WE REFACTOR?



```java
@Transactional
public void updateBucketStatusGoingForwardOnly(
  Bucket bucket,
  BucketStatus bucketStatus
) {
  Client client = bucket.getRisk();
  BucketStatus currentBucketStatus = client.getBucketStatus();
  boolean publishUpdateWithAuditInput = bucketProcessingStatusManager
          .hasNotAnyIncompletePayments(
            bucketStatus.name(),
            client,
            bucket.getId());

  if (bucketStatus == BucketStatus.COMPLETE) {
    updateBucketStatusWithAuditAndPublishUpdate(
          client, bucketStatus, publishUpdateWithAuditInput);
  } else if (bucketStatus == BucketStatus.BOUND) {
    if (currentBucketStatus != BucketStatus.COMPLETE) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  } else if (bucketStatus == BucketStatus.QUOTED) {
    if (currentBucketStatus != BucketStatus.COMPLETE
            && currentBucketStatus != BucketStatus.BOUND) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  } else if (bucketStatus == BucketStatus.WORK_IN_PROGRESS) {
    if (currentBucketStatus != BucketStatus.COMPLETE
            && currentBucketStatus != BucketStatus.BOUND
            && currentBucketStatus != BucketStatus.QUOTED) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  } else if (bucketStatus == BucketStatus.LOGGED) {
    if (currentBucketStatus != BucketStatus.COMPLETE
            && currentBucketStatus != BucketStatus.BOUND
        && currentBucketStatus != BucketStatus.QUOTED
        && currentBucketStatus != BucketStatus.WORK_IN_PROGRESS) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  }
}
```



```java
@Transactional
public void updateBucketStatusGoingForwardOnly(
  Bucket bucket,
  BucketStatus bucketStatus
) {
  Client client = bucket.getRisk();
  BucketStatus currentBucketStatus = client.getBucketStatus();
  boolean publishUpdateWithAuditInput = bucketProcessingStatusManager
          .hasNotAnyIncompletePayments(
            bucketStatus.name(),
            client,
            bucket.getId());

  if (bucketStatus == BucketStatus.COMPLETE) {
    updateBucketStatusWithAuditAndPublishUpdate(
          client, bucketStatus, publishUpdateWithAuditInput);
  } else if (bucketStatus == BucketStatus.BOUND) {
    if (currentBucketStatus != BucketStatus.COMPLETE) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  } else if (bucketStatus == BucketStatus.QUOTED) {
    if (currentBucketStatus != BucketStatus.COMPLETE
            && currentBucketStatus != BucketStatus.BOUND) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  } else if (bucketStatus == BucketStatus.WORK_IN_PROGRESS) {
    if (currentBucketStatus != BucketStatus.COMPLETE
            && currentBucketStatus != BucketStatus.BOUND
            && currentBucketStatus != BucketStatus.QUOTED) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  } else if (bucketStatus == BucketStatus.LOGGED) {
    if (currentBucketStatus != BucketStatus.COMPLETE
            && currentBucketStatus != BucketStatus.BOUND
        && currentBucketStatus != BucketStatus.QUOTED
        && currentBucketStatus != BucketStatus.WORK_IN_PROGRESS) {
      updateBucketStatusWithAuditAndPublishUpdate(
            client, bucketStatus, publishUpdateWithAuditInput);
    }
  }
}
```

PROJECT LIFE CYCLE

GREEN FIELD

GREEN FIELD

STABILIZATION

GREEN FIELD

STABILIZATION

MAINTENANCE

GREEN FIELD

[✓ STABILIZATION]

MAINTENANCE

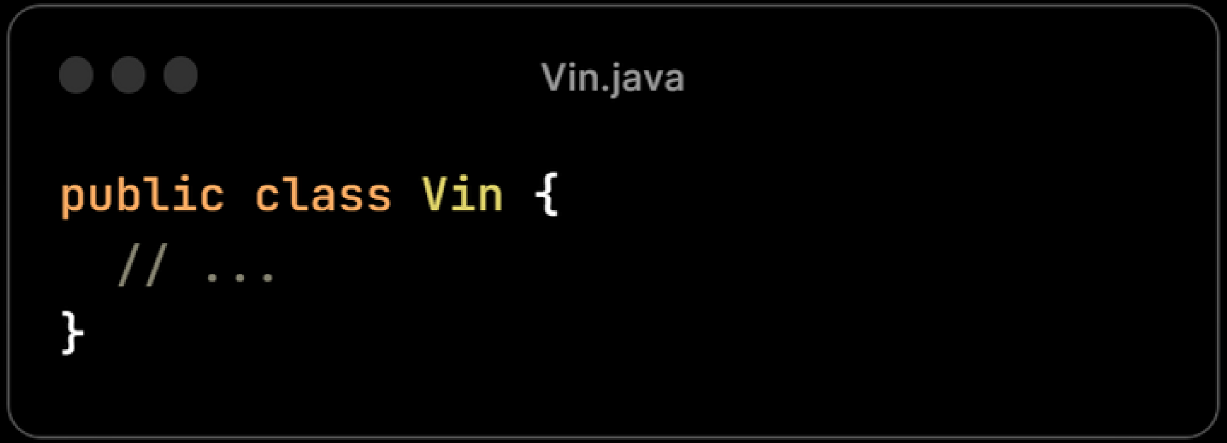
ONE COMMIT

5 MINUTES

EVERY DAY

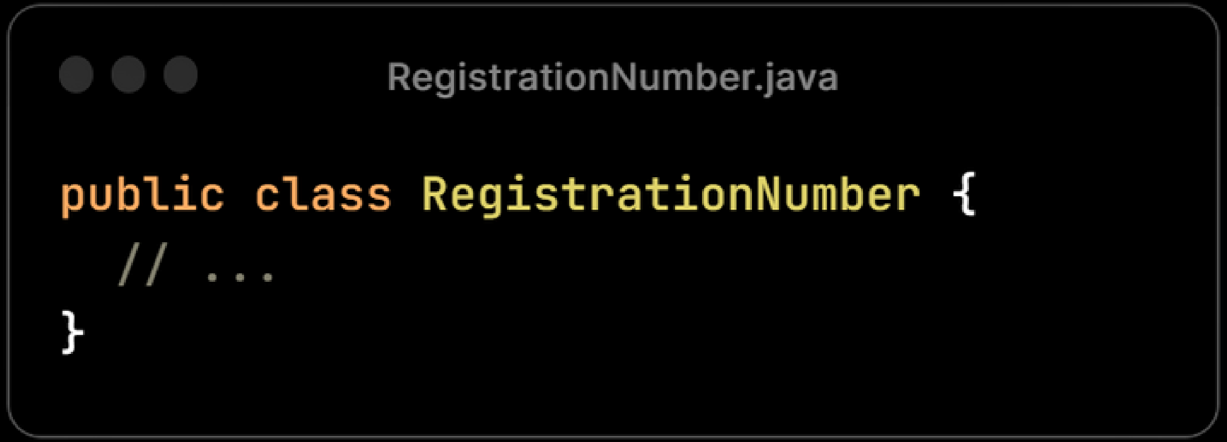
JUMP INTO THE CODE!

STRING VIN



```java
public class Vin {
  // ...
}
```

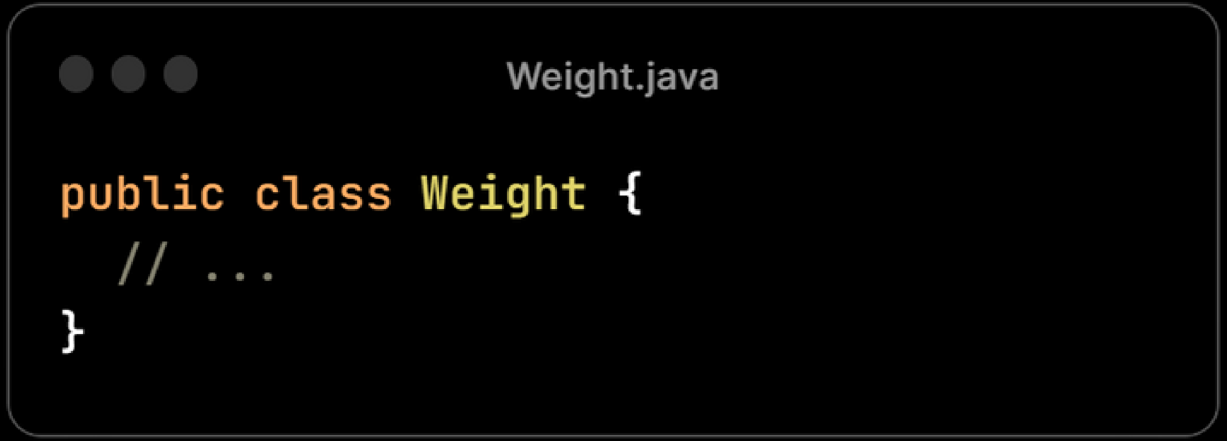
STRING REGISTRATION NUMBER



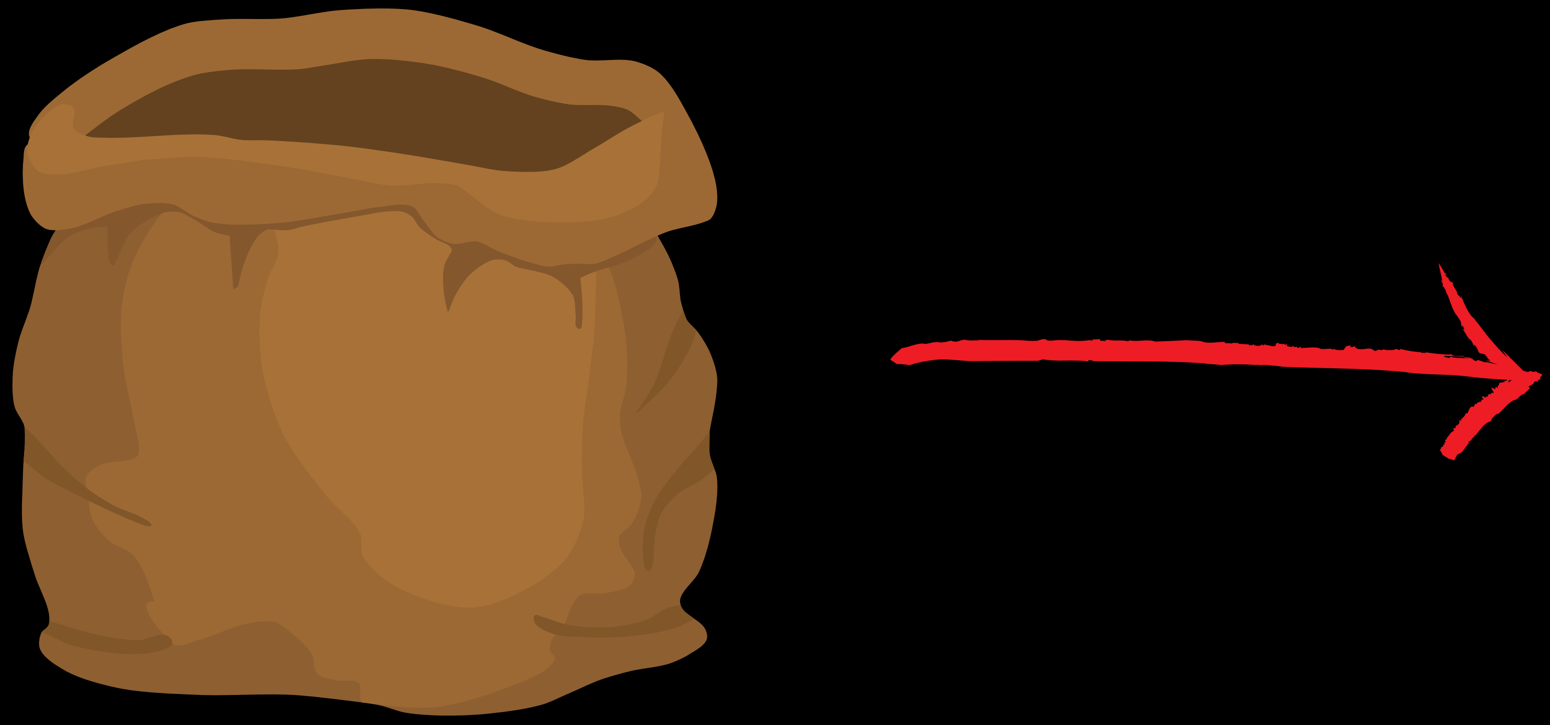
```java
public class RegistrationNumber {
  // ...
}
```

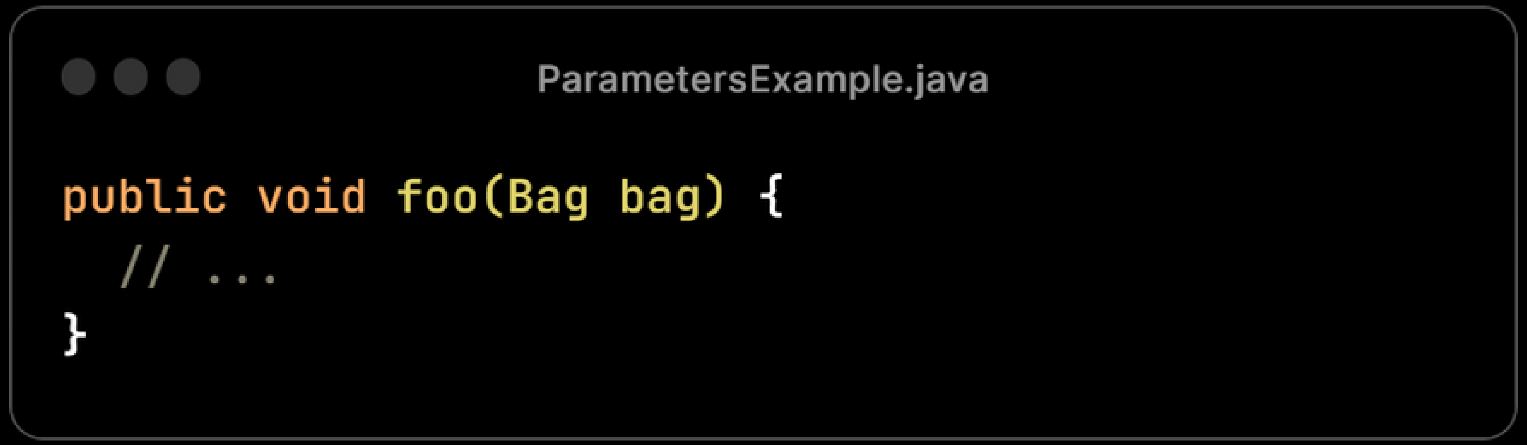
DOUBLE WEIGHT VALUE

STRING WEIGHT UNIT



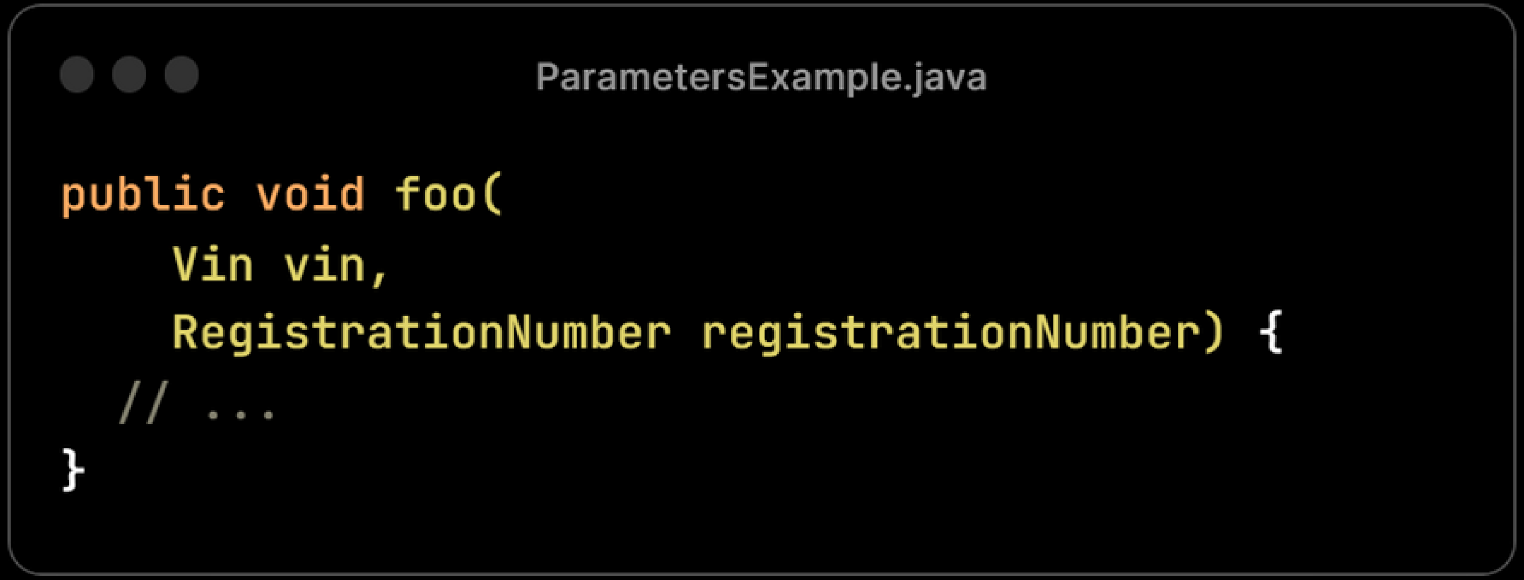
```java
public class Weight {
  // ...
}
```





ParametersExample.java

```java
public void foo(Bag bag) {
  // ...
}
```



ParametersExample.java

```java
public void foo(
    Vin vin,
    RegistrationNumber registrationNumber) {
  // ...
}
```

SUMMARY

REFACTOR TO...

INCREASE TESTABILITY

DESCREASE COGNITIVE LOAD

ACQUIRE KNOWLEDGE

THANKS
FOR YOUR ATTENTION!

@CEZARY.SANECKI

CEZARYSANECKI

CEZARY-SANECKI

CEZARYSANECKI.PL