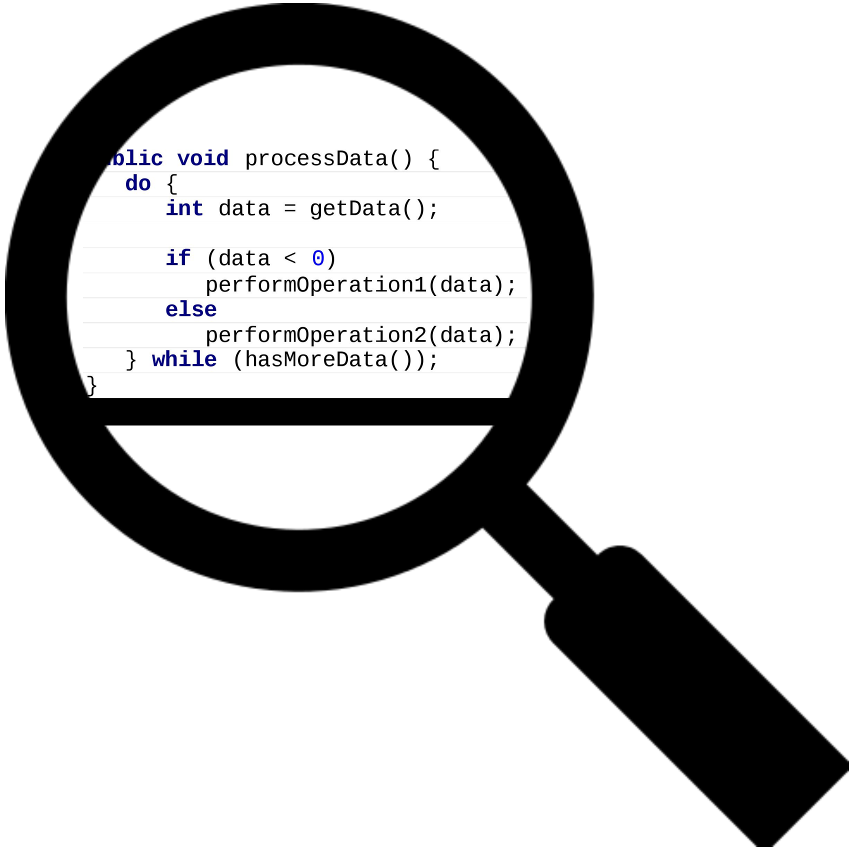
CS 301
FDD, TDD, ETVX

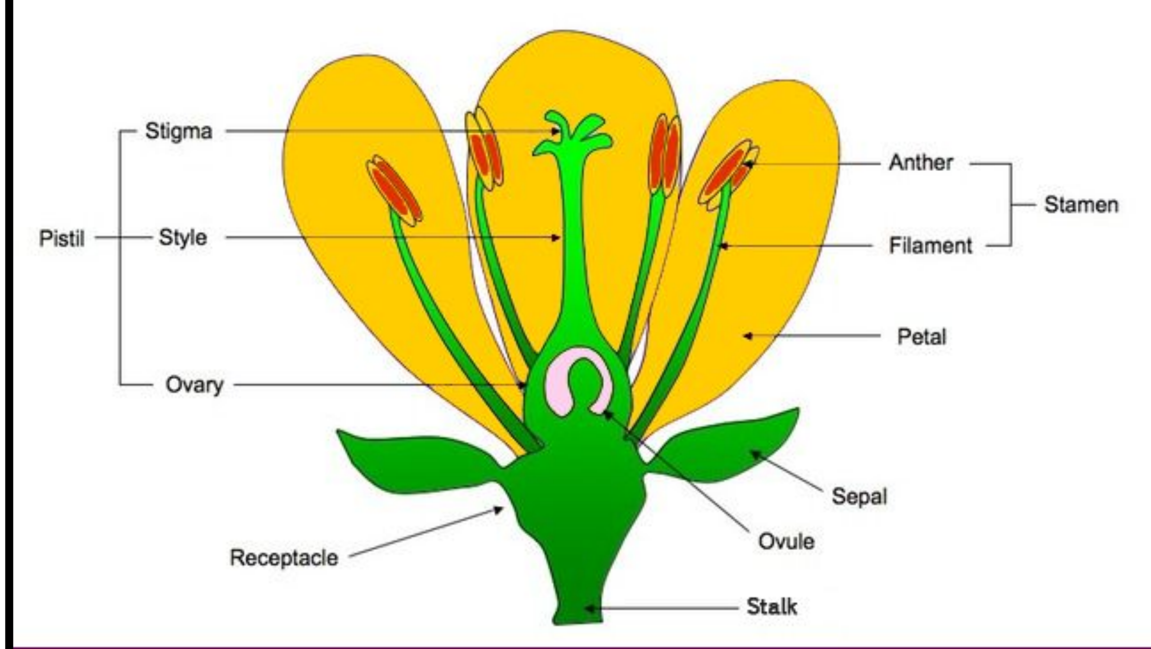
Eswaran Narasimhan

28/30 – Aug - 2024

Software Development Methodologies



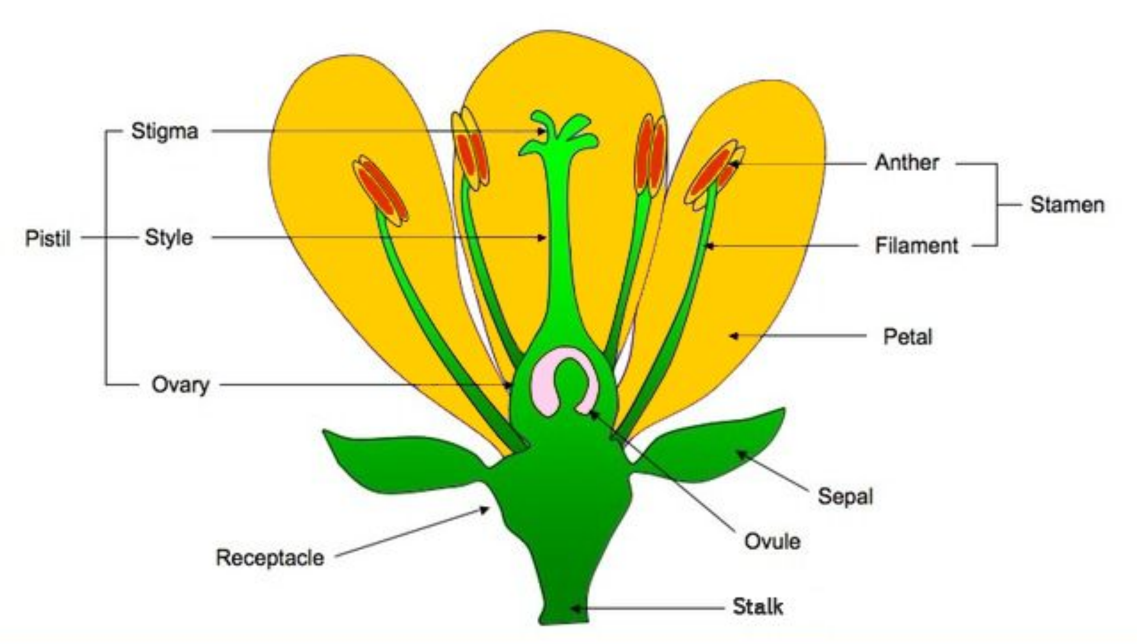
```
blic void processData() {
  do {
     int data = getData();

     if (data < 0)
        performOperation1(data);
     else
        performOperation2(data);
  } while (hasMoreData());
}
```



Parts of a flower

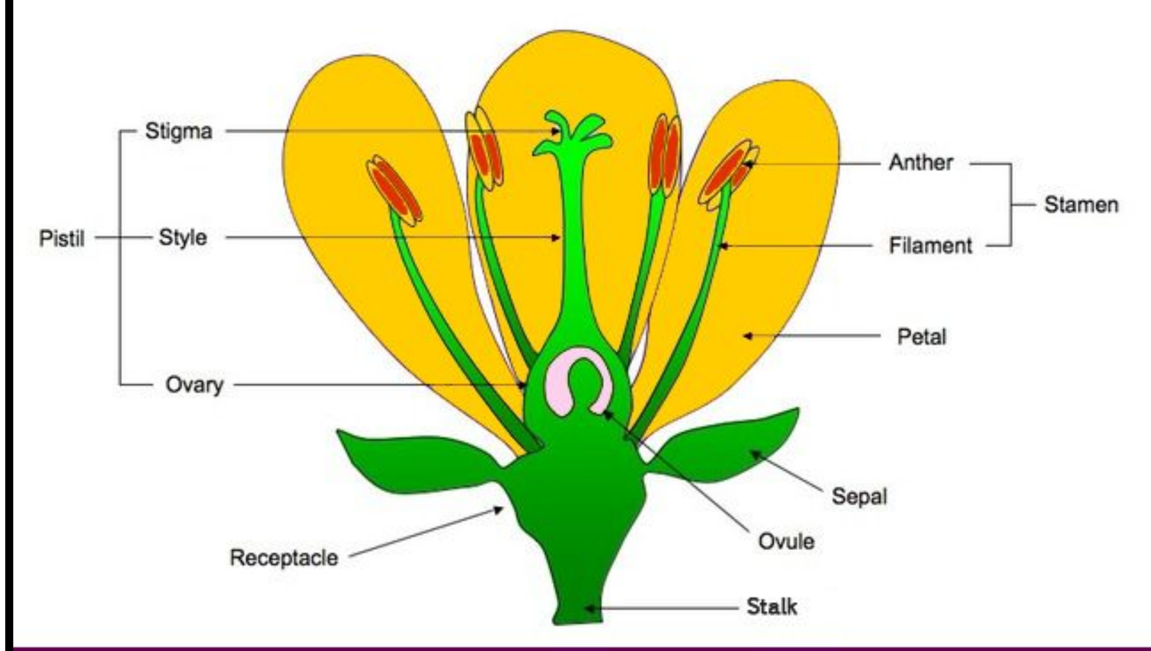
Feature Driven Development

- ❑ **Pistil**
  - Stigma
  - Style
  - Ovary
    - ❖ Ovule
- ❑ **Receptacle**
- ❑ **Stalk**
- ❑ **Sepal**
- ❑ **Petal**
- ❑ **Stamen**
  - Anther
  - Filament



Parts of a flower

Feature Driven Development

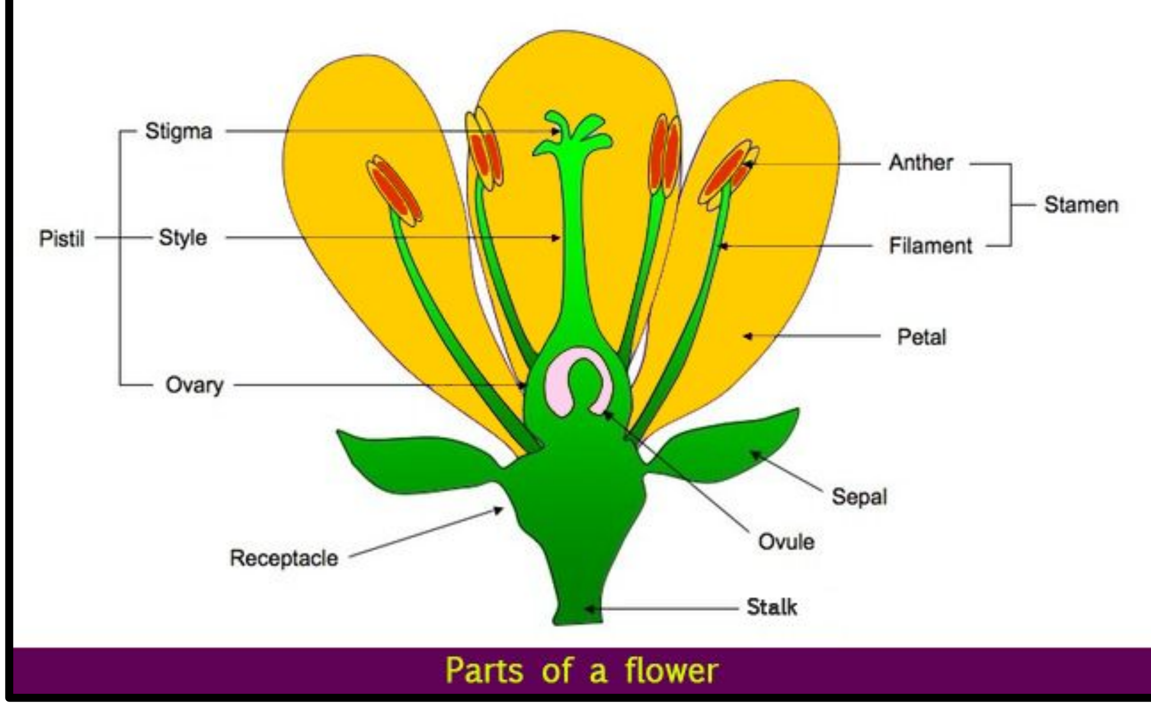


Parts of a flower

<action><result><object>

- ❑ Compute the total tax
- ❑ Validate the entered password
- ❑ Authenticate the credentials of the user

Feature Driven Development



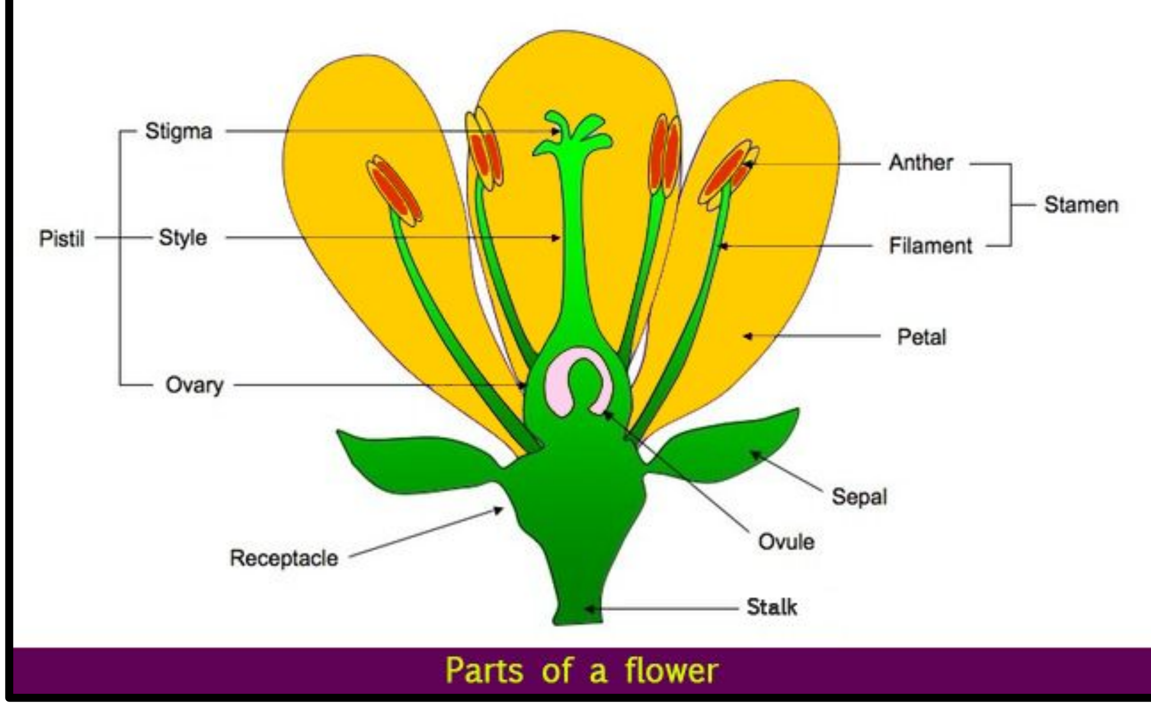
<action><result><object>

- ❑ Compute the total tax
- ❑ Validate the entered password
- ❑ Authenticate the credentials of the user

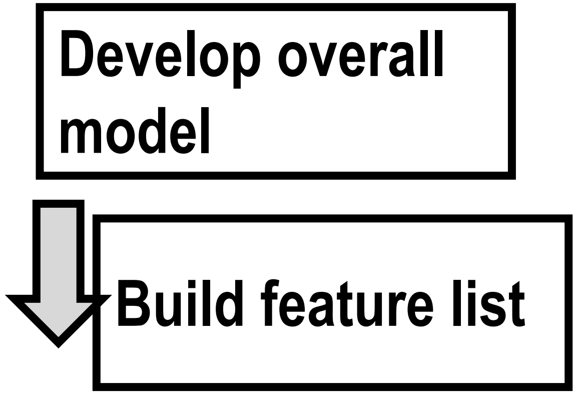
- ❑ Write 5 features in Moodle
- ❑ Write 3 features in PayTM

Feature Driven Development

Develop overall model

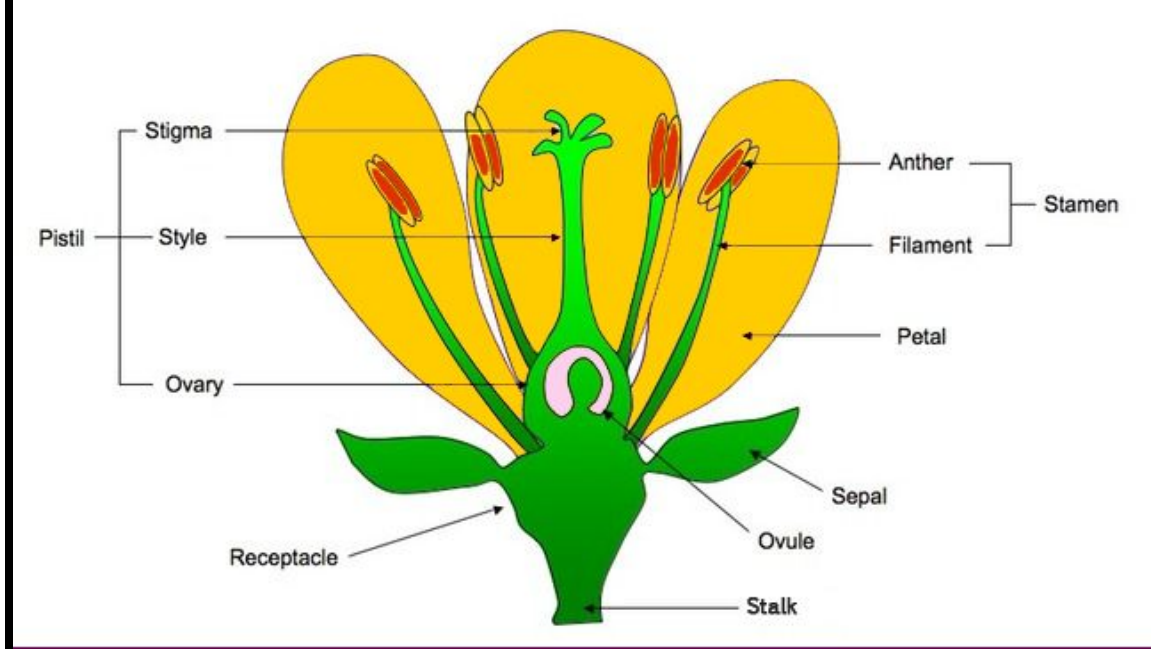


Feature Driven Development



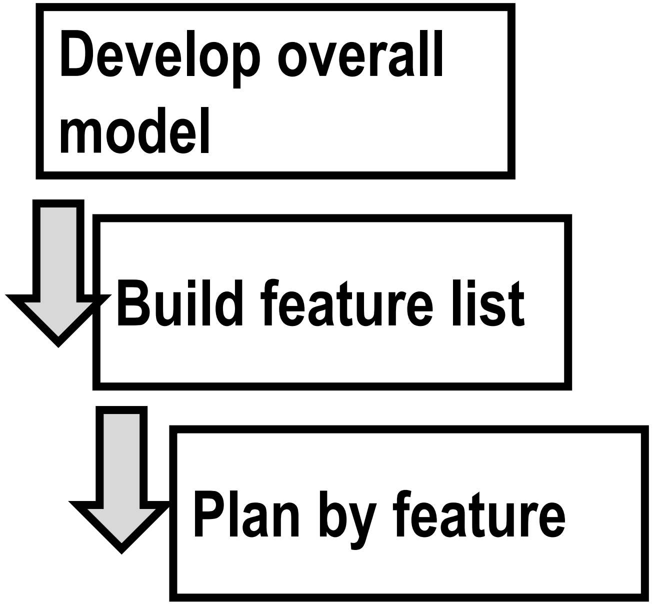
Develop overall model

Build feature list



Parts of a flower

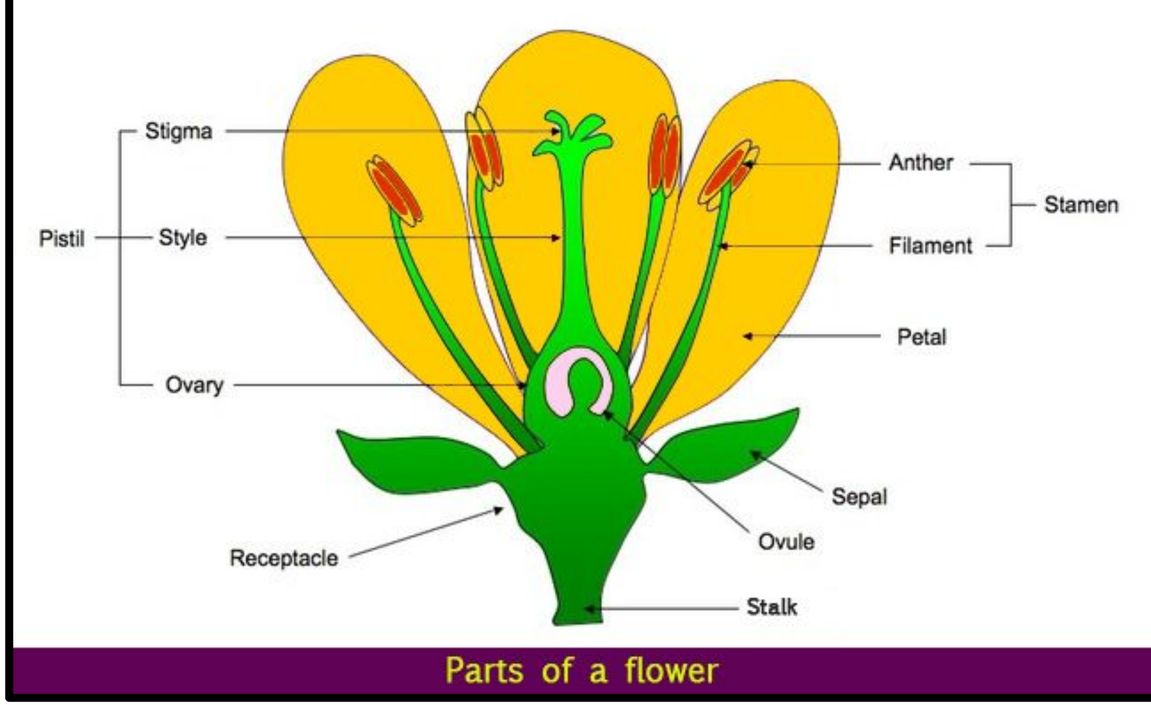
Feature Driven Development



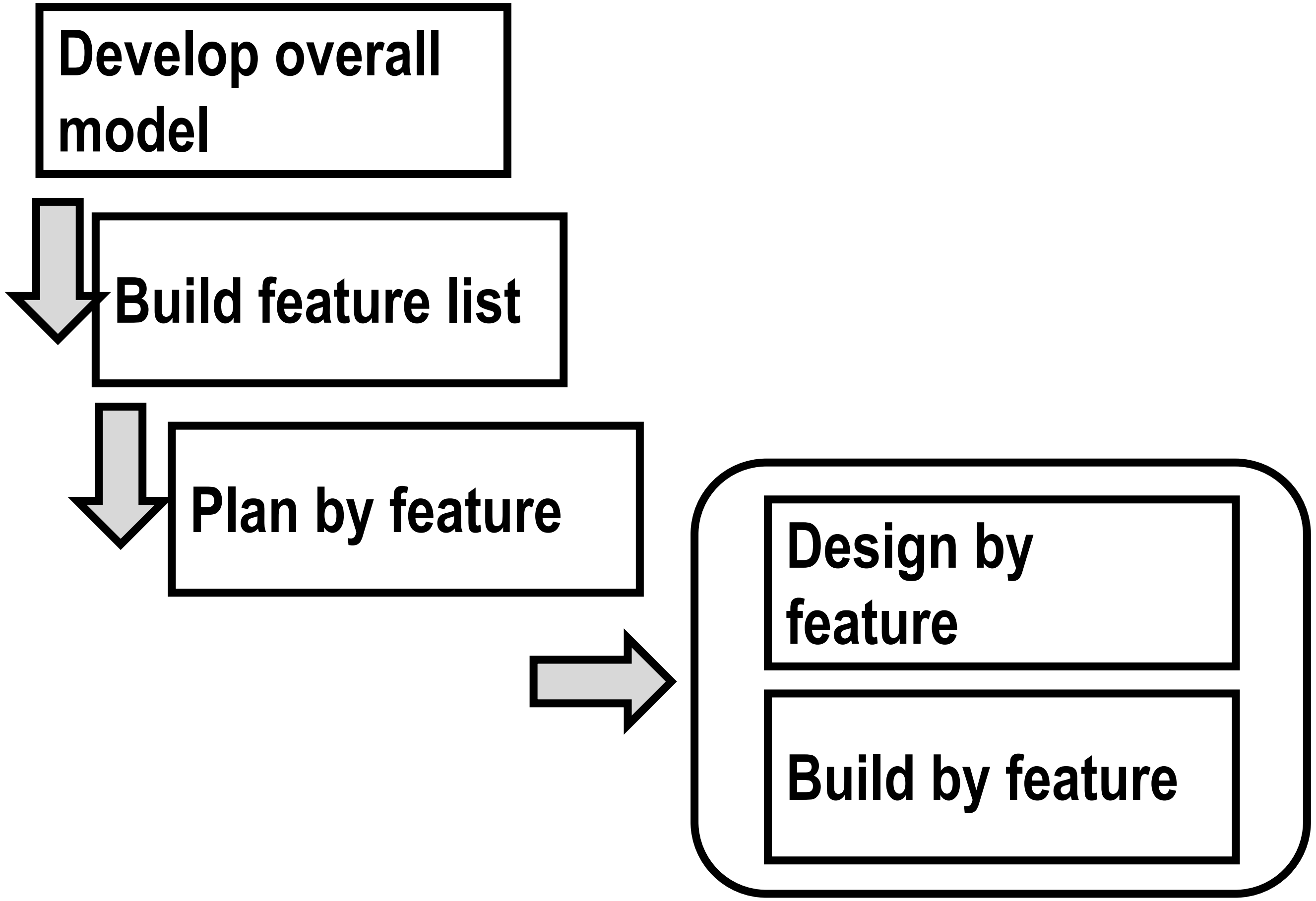
Develop overall model

Build feature list

Plan by feature



Feature Driven Development



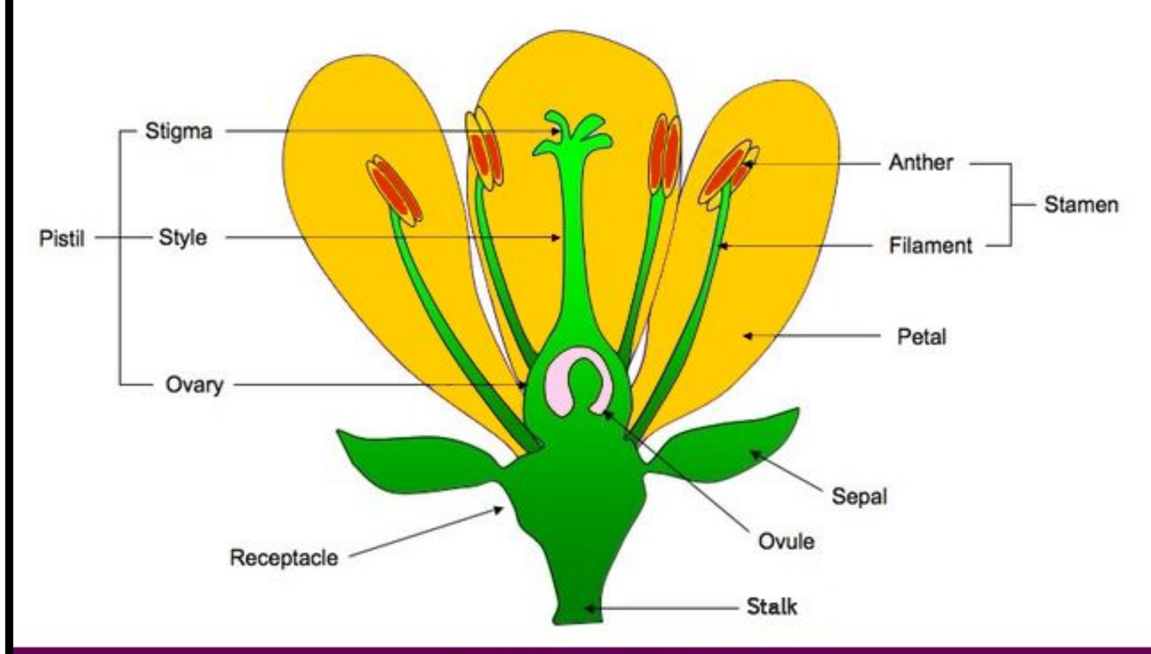
Develop overall model

Build feature list

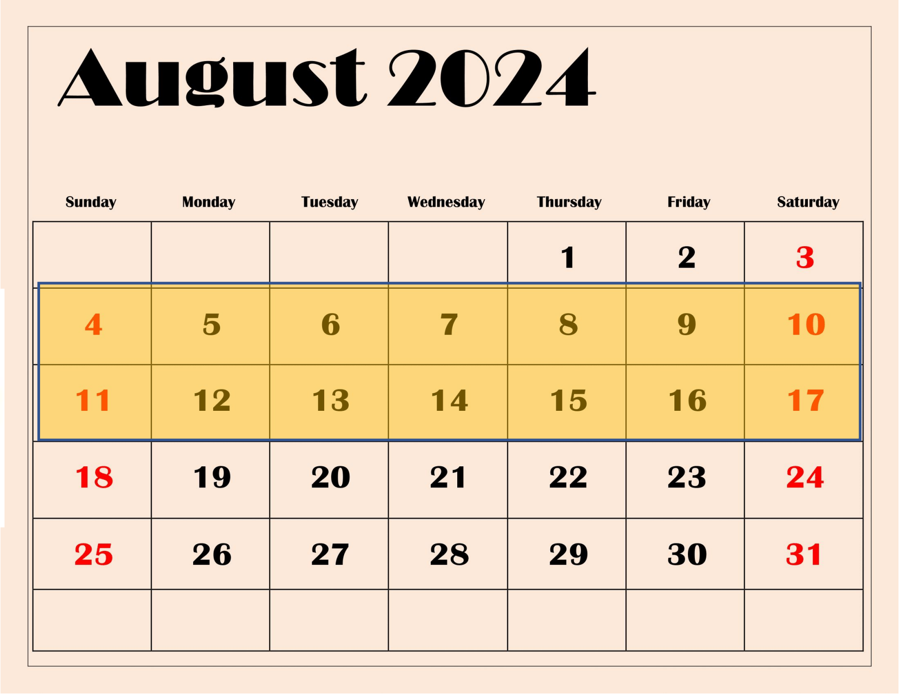
Plan by feature

Design by feature

Build by feature



Parts of a flower



August 2024

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

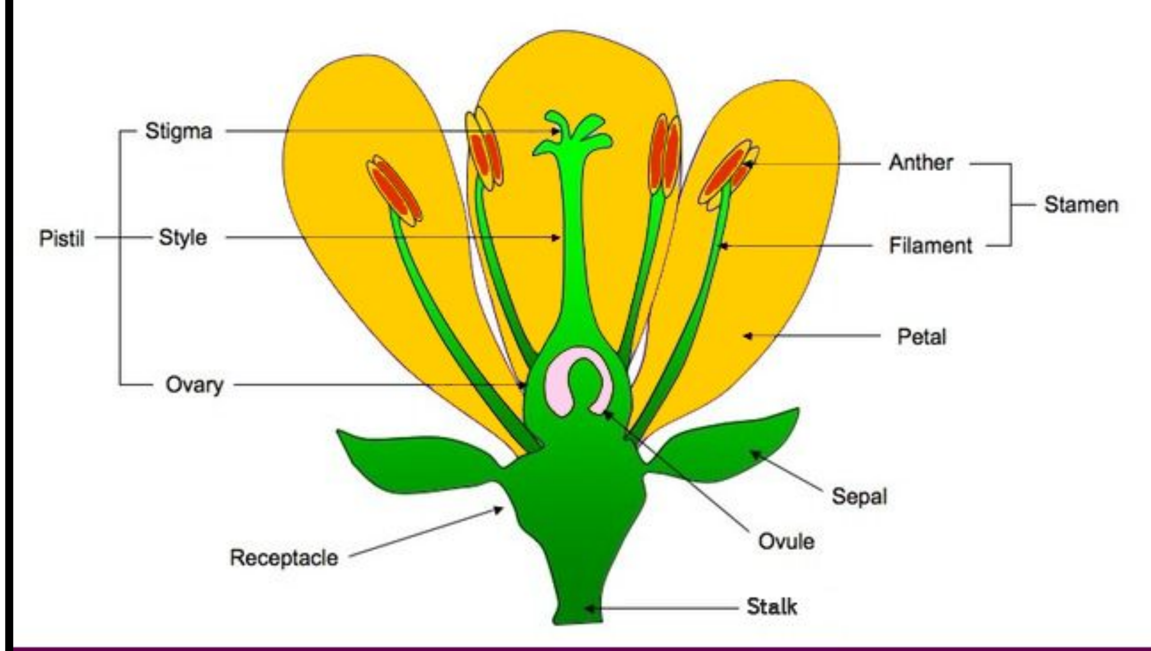
Feature Driven Development

Advantages

□ Large scale projects
□ Improving quality

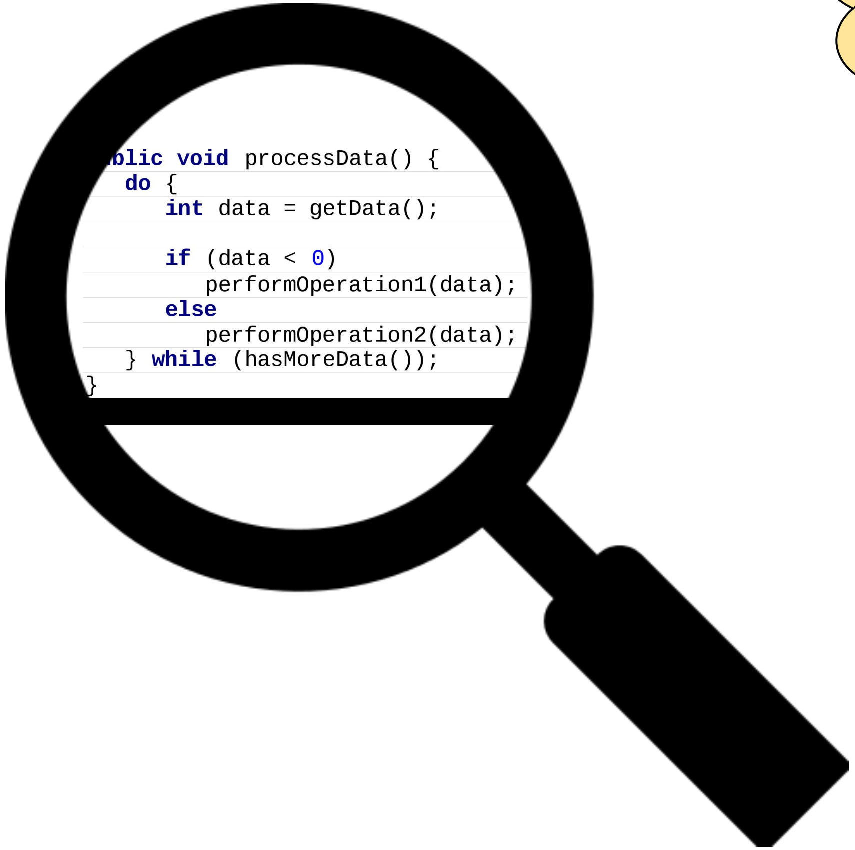
Disadvantages

□ Dynamic deadlines
□ Skill dependent
□ Scarce Documentation at detailed level



Parts of a flower

Test Driven Development

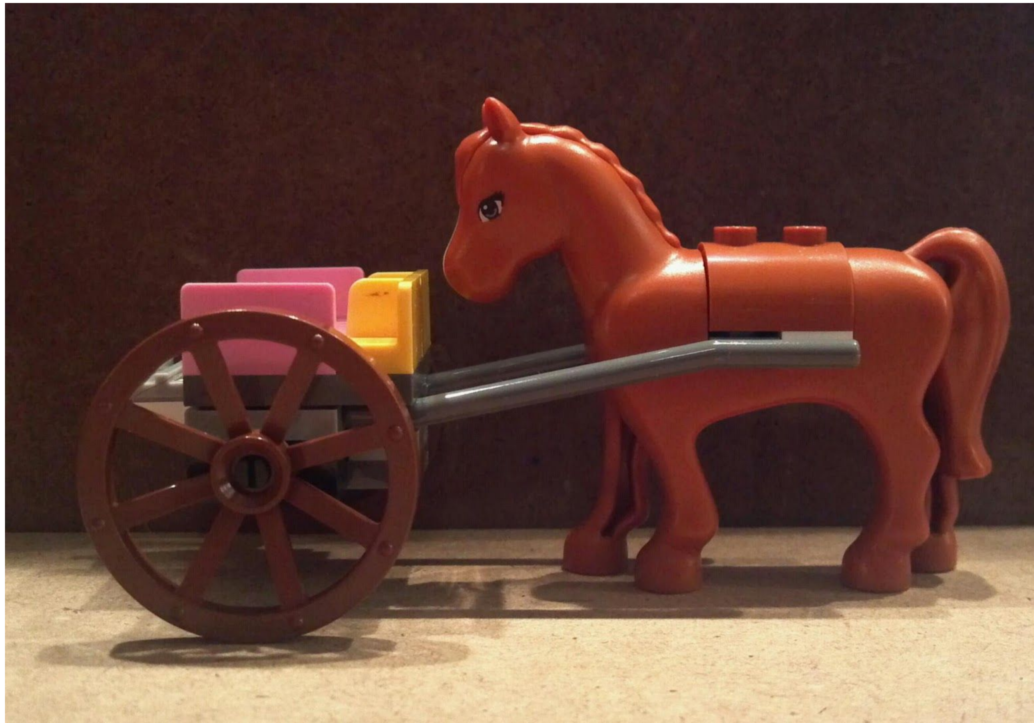


```
blic void processData() {
   do {
      int data = getData();

      if (data < 0)
         performOperation1(data);
      else
         performOperation2(data);
   } while (hasMoreData());
}
```

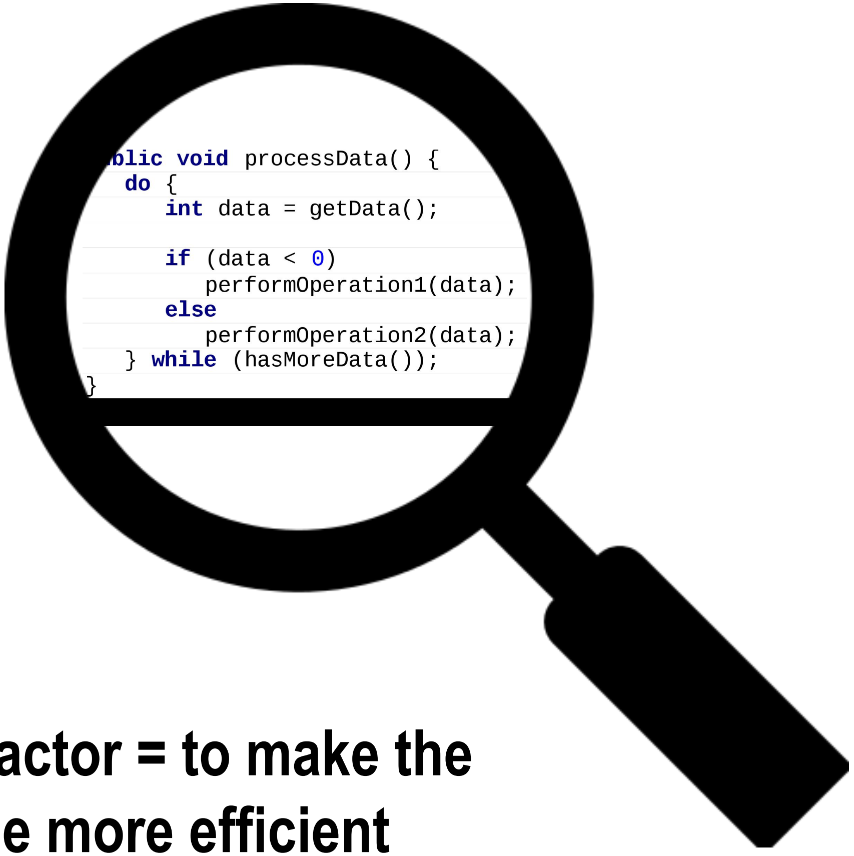
Doing the right thing.

Doing the thing right.

http://bestandworstever.blogspot.com/2013/01/worst-thing-to-put-before-horse-ever.html

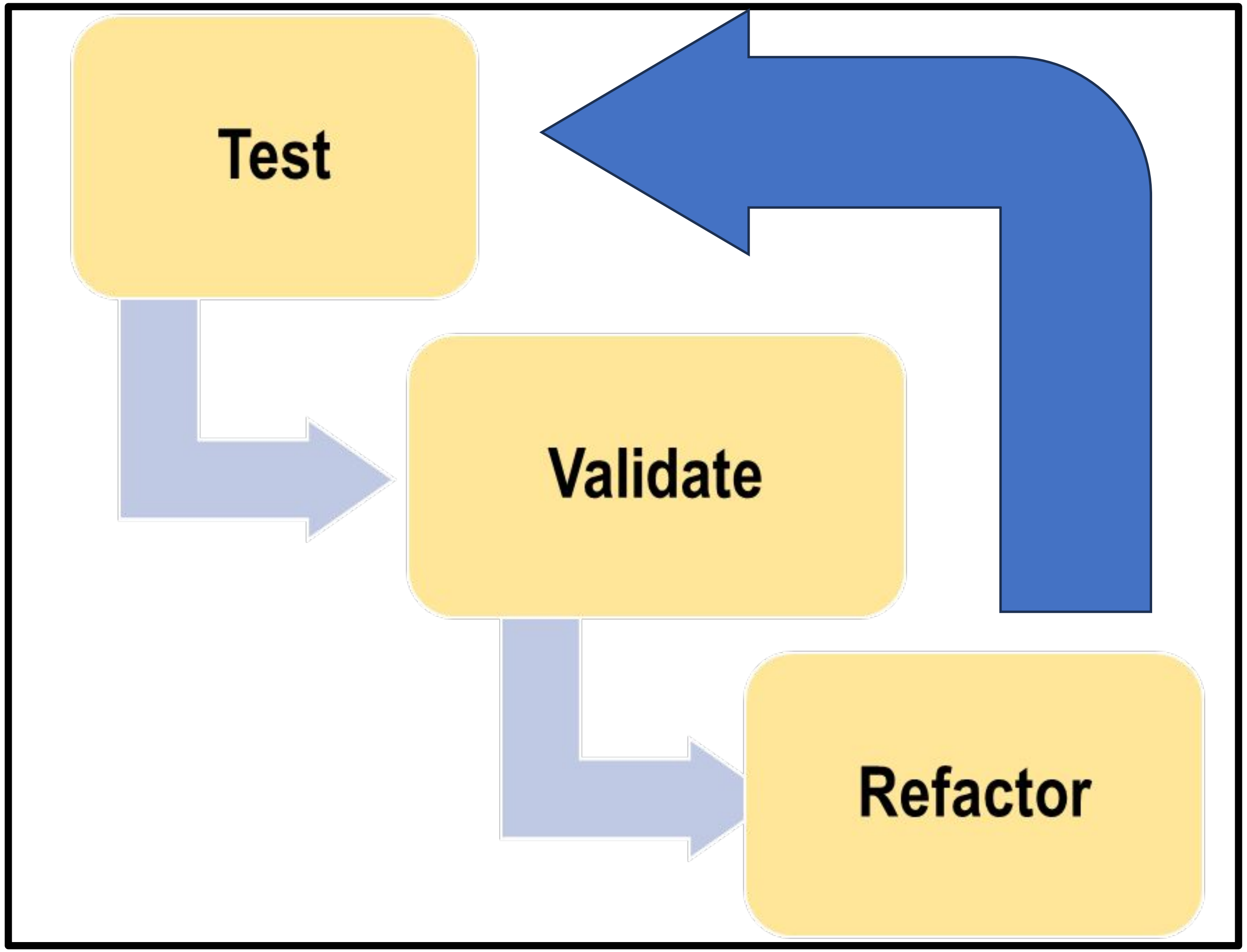


Test Driven Development



```
blic void processData() {
  do {
     int data = getData();

     if (data < 0)
        performOperation1(data);
     else
        performOperation2(data);
  } while (hasMoreData());
}
```

Refactor = to make the code more efficient and rewrite better

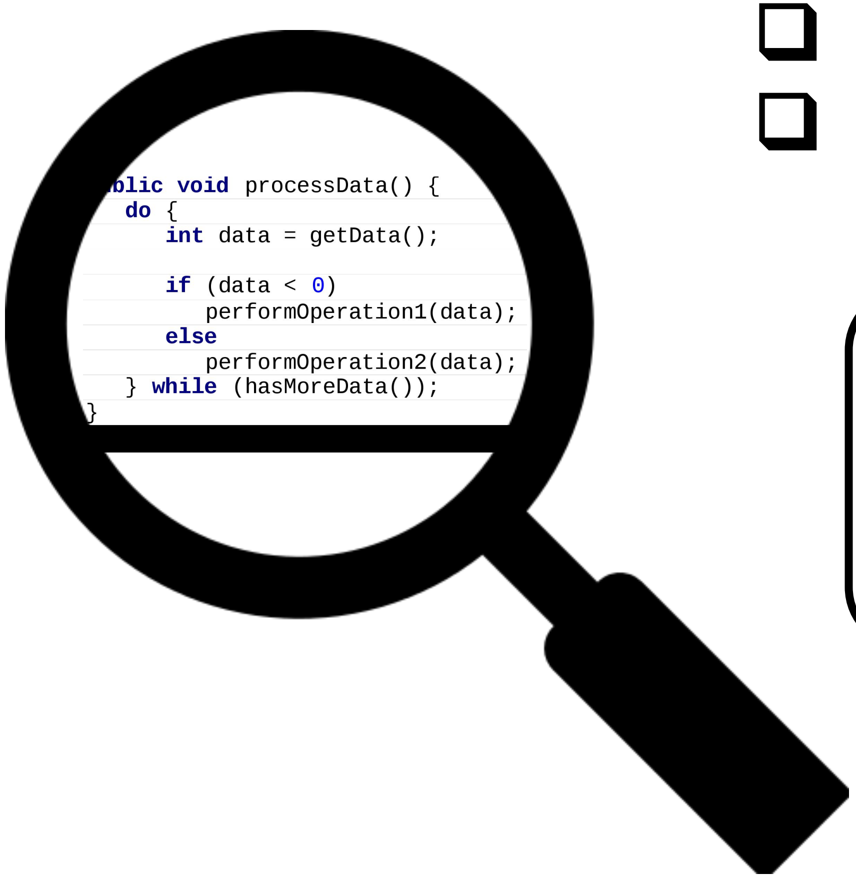


Test

Validate

Refactor

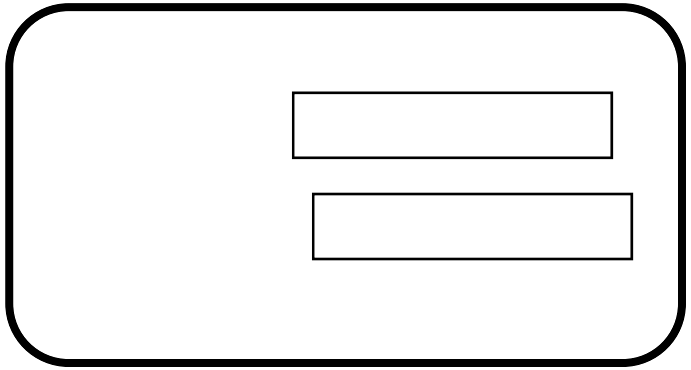
Test Driven Development



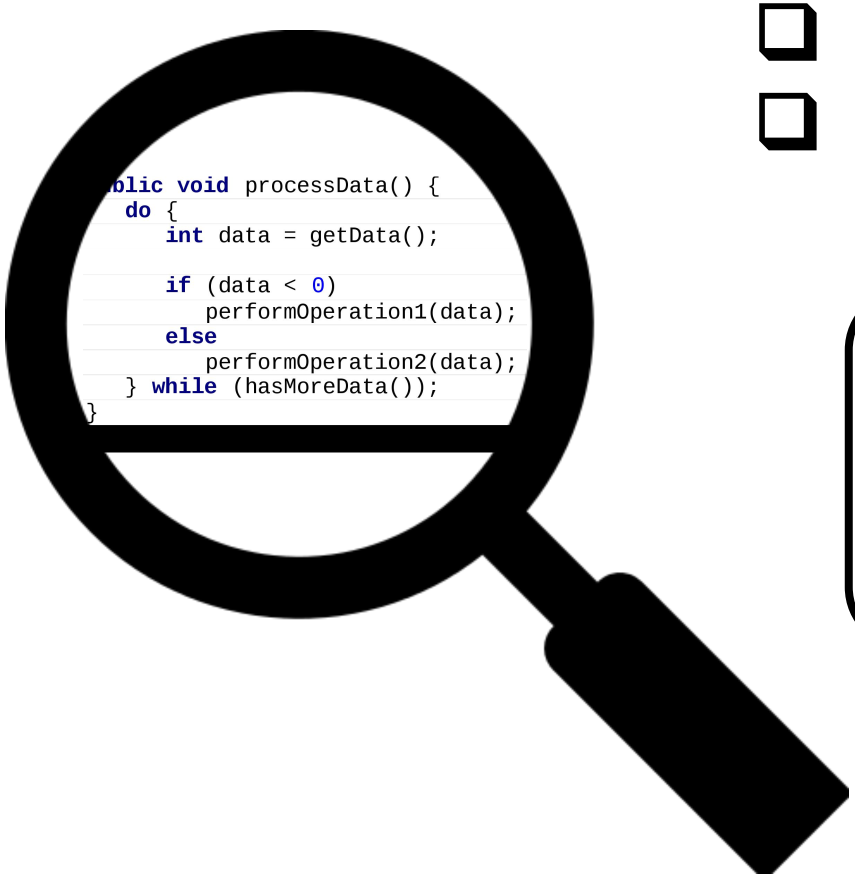
```
blic void processData() {
  do {
      int data = getData();

      if (data < 0)
          performOperation1(data);
      else
          performOperation2(data);
  } while (hasMoreData());
}
```

Screen with two inputs

- ❑ User id
- ❑ Password



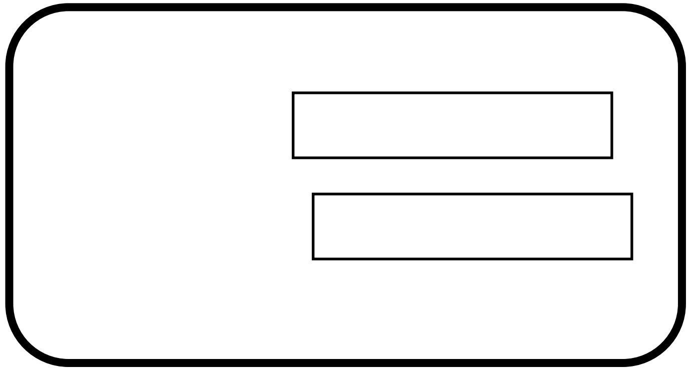
Test Driven Development



```
ublic void processData() {
  do {
      int data = getData();

      if (data < 0)
          performOperation1(data);
      else
          performOperation2(data);
  } while (hasMoreData());
}
```

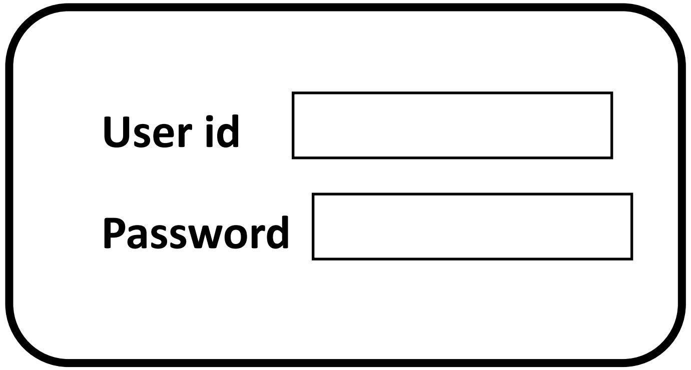
Screen with two inputs

- □ User id
- □ Password

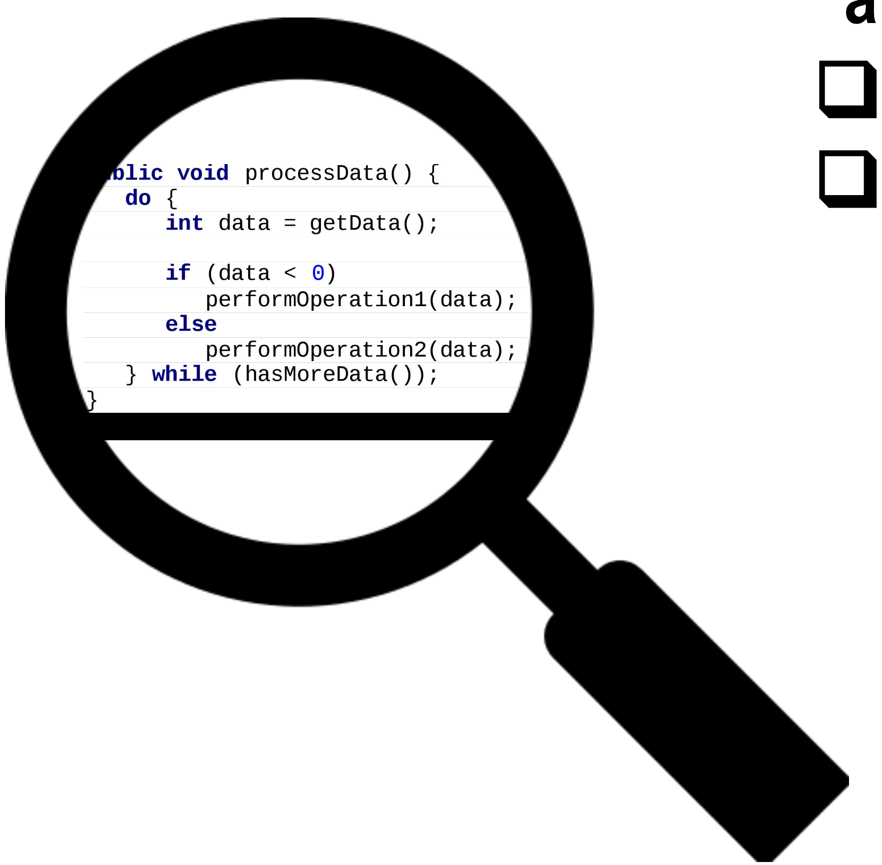


Screen with two inputs and two captions

- □ User id
- □ Password



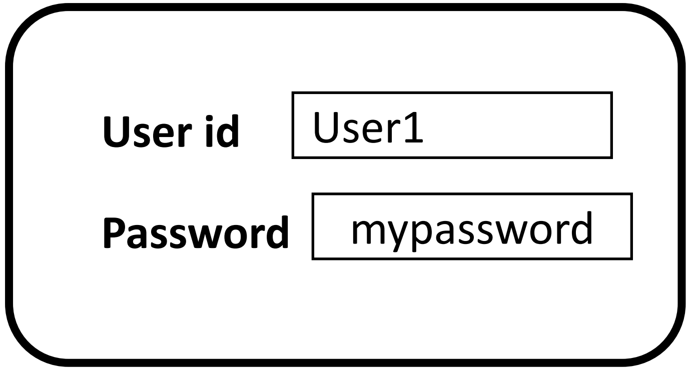
Test Driven Development



```
blic void processData() {
  do {
      int data = getData();

      if (data < 0)
          performOperation1(data);
      else
          performOperation2(data);
  } while (hasMoreData());
}
```

Screen with two inputs and two captions

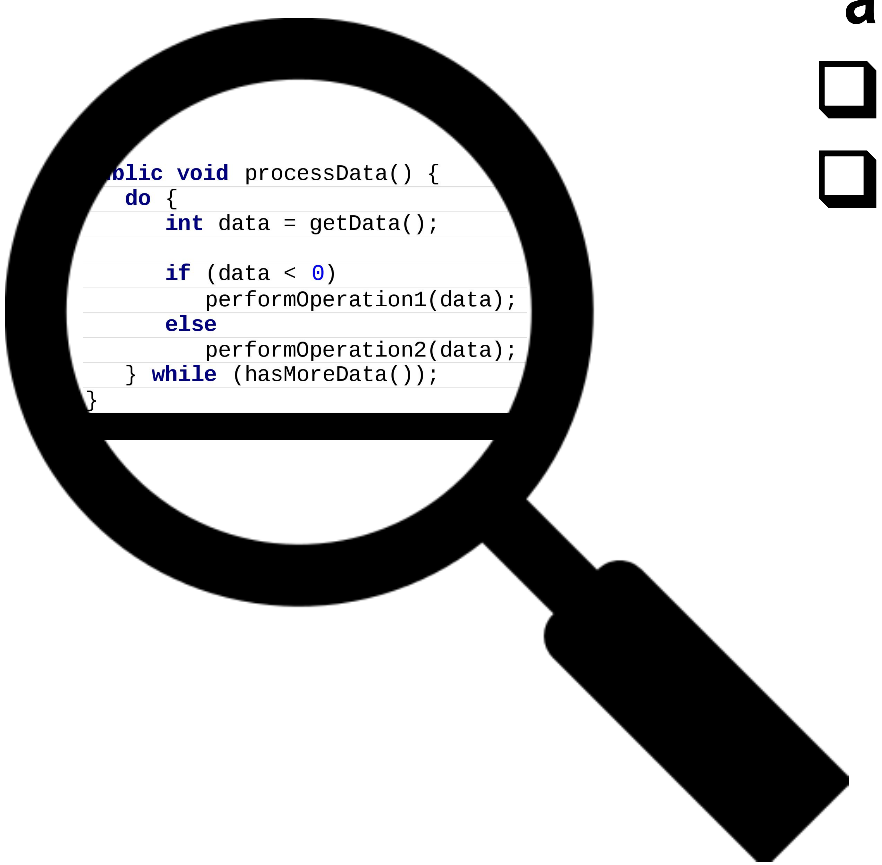
- ❑ User id
- ❑ Password



User id User1

Password mypassword

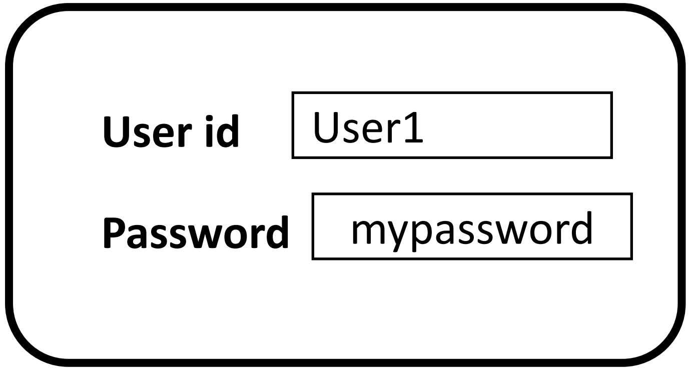
Test Driven Development



```
blic void processData() {
  do {
      int data = getData();

      if (data < 0)
          performOperation1(data);
      else
          performOperation2(data);
  } while (hasMoreData());
}
```

Screen with two inputs and two captions

- □ User id
- □ Password

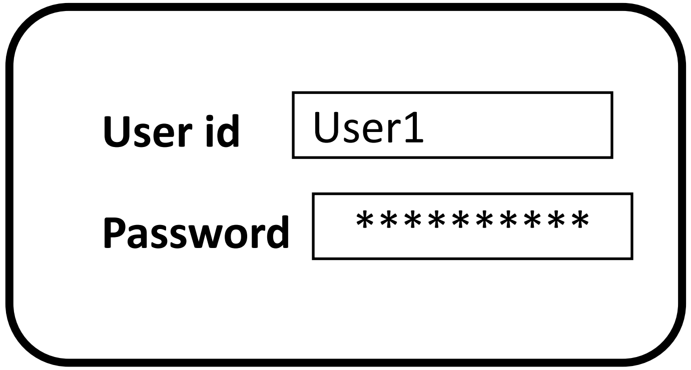


User id: User1

Password: mypassword

Screen with two inputs and two captions

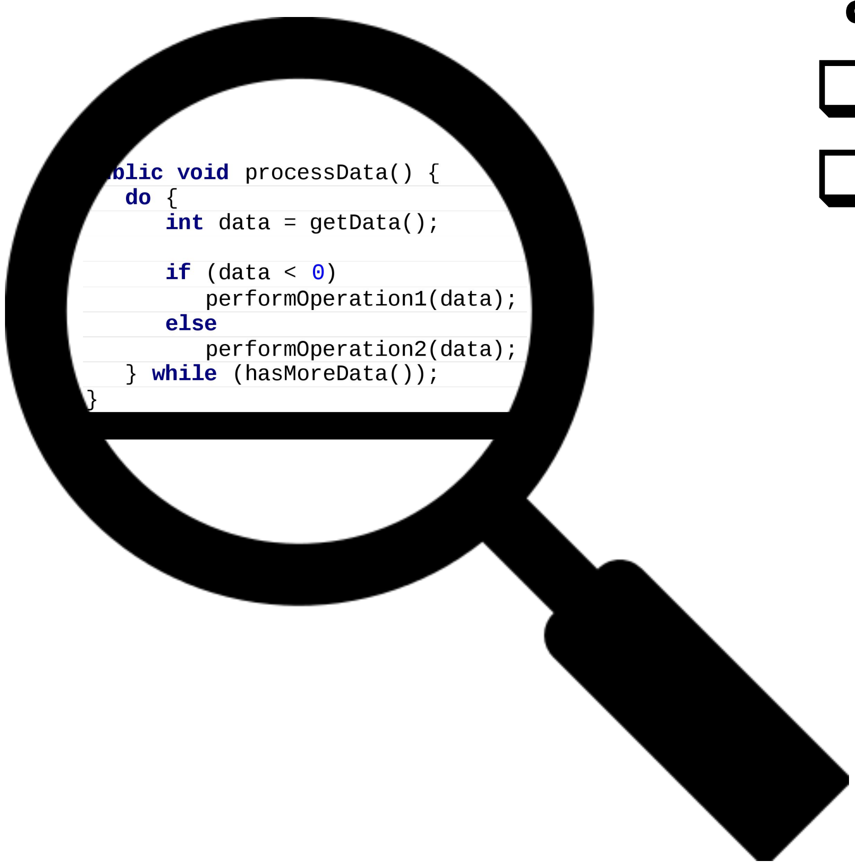
- □ User id
- □ Password with masking



User id: User1

Password: **********

Test Driven Development



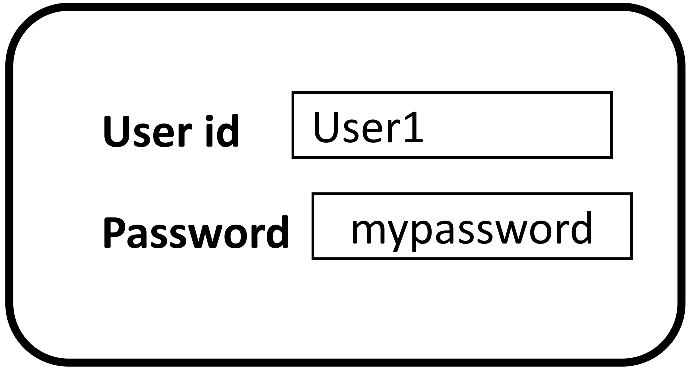
```
blic void processData() {
   do {
      int data = getData();

      if (data < 0)
         performOperation1(data);
      else
         performOperation2(data);
   } while (hasMoreData());
}
```

Screen with two inputs and two captions

- □ User id
- □ Password

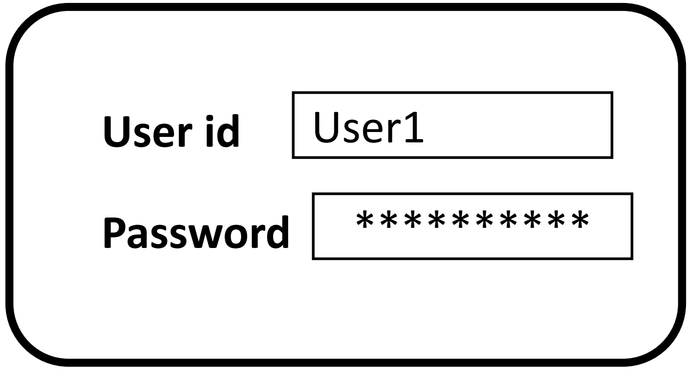
Screen with two inputs and two captions

- □ User id
- □ Password with masking



User id User1

Password mypassword

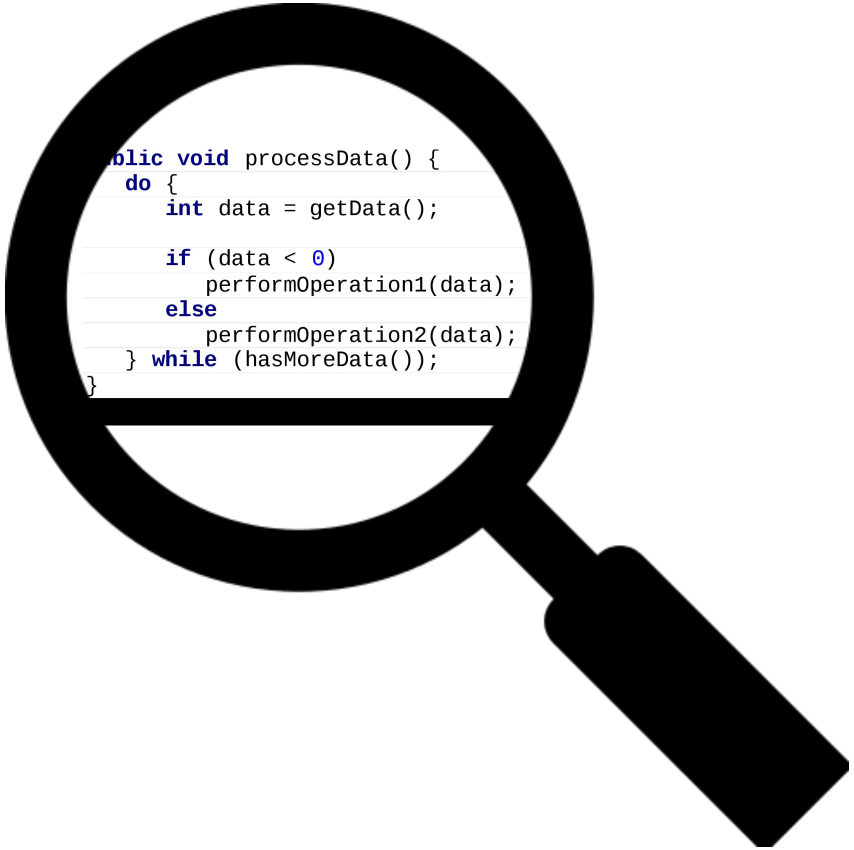


User id User1

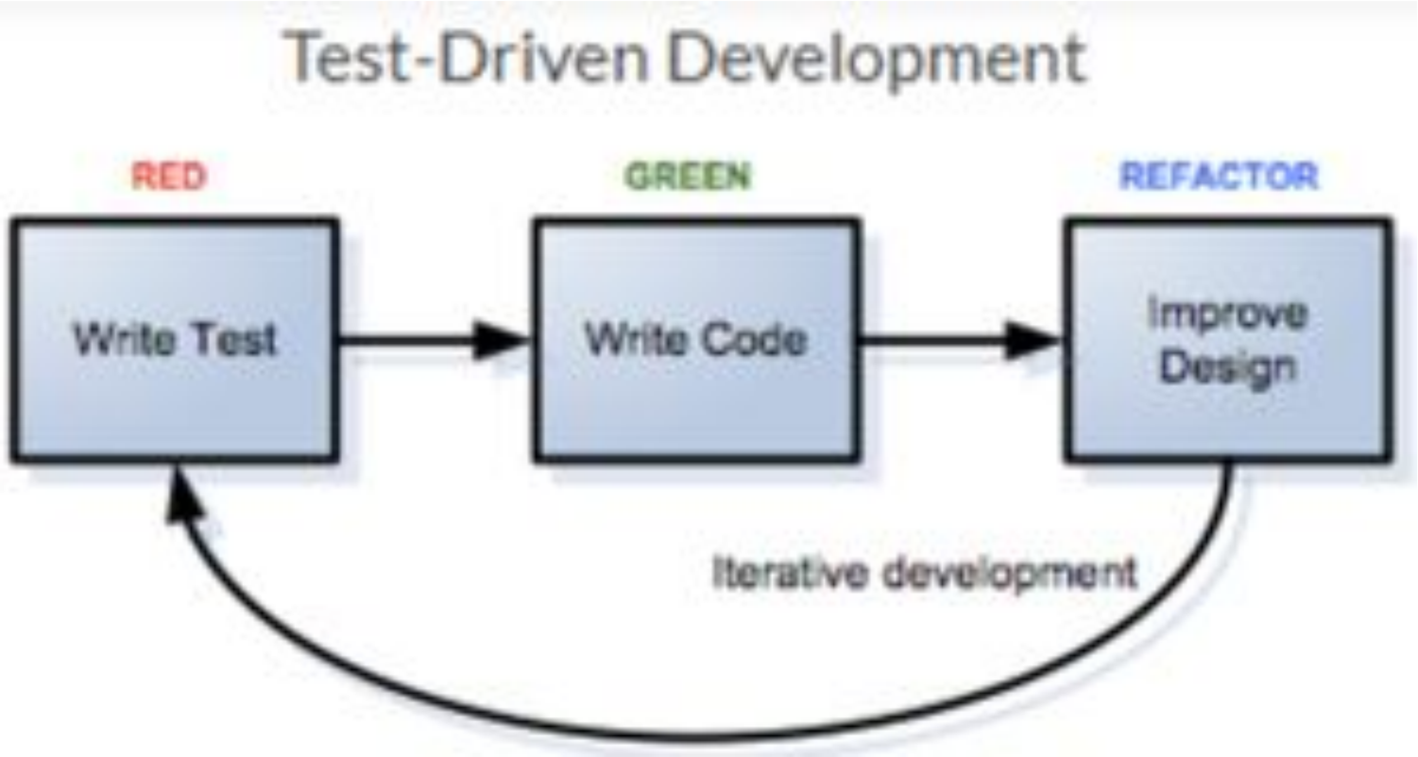
Password **********

Submit button, reset, forgot password, Captcha

Test Driven Development

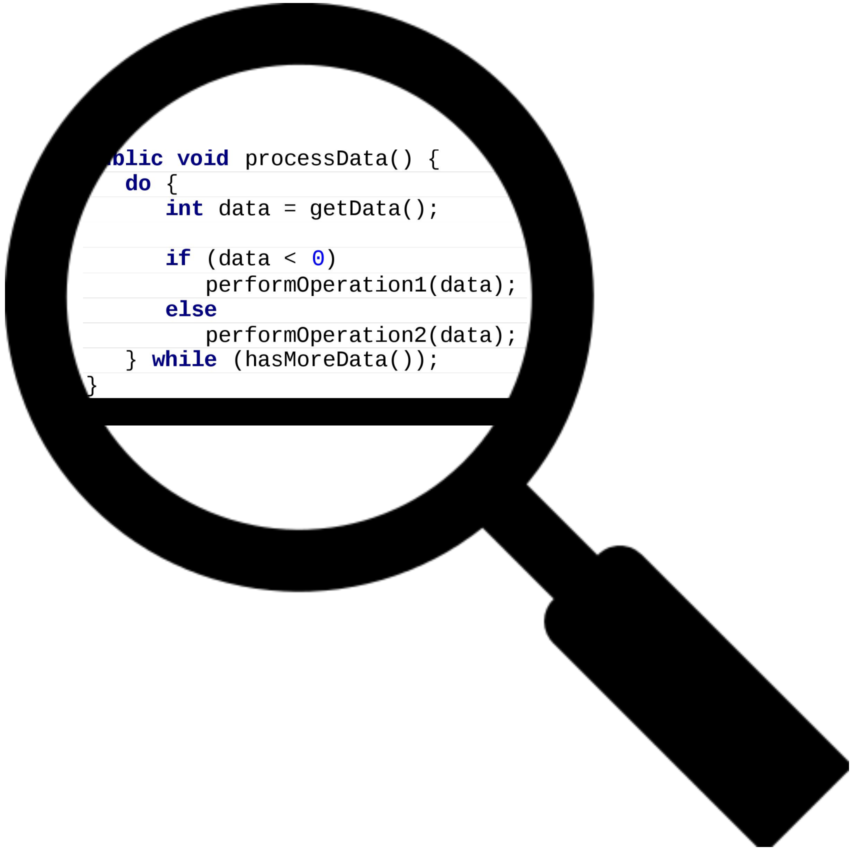


```
blic void processData() {
  do {
     int data = getData();

     if (data < 0)
        performOperation1(data);
     else
        performOperation2(data);
  } while (hasMoreData());
}
```



https://cio-wiki.org/wiki/Test_Driven_Development_(TDD)

Test Driven Development



```
blic void processData() {
    do {
        int data = getData();

        if (data < 0)
            performOperation1(data);
        else
            performOperation2(data);
    } while (hasMoreData());
}
```

Advantages

❑ Simple, elegant, modular code

❑ Shift Left

❑ Concurrent documentation.

❑ Maintenance of code.

❑ Learning economies

❑ End – user view

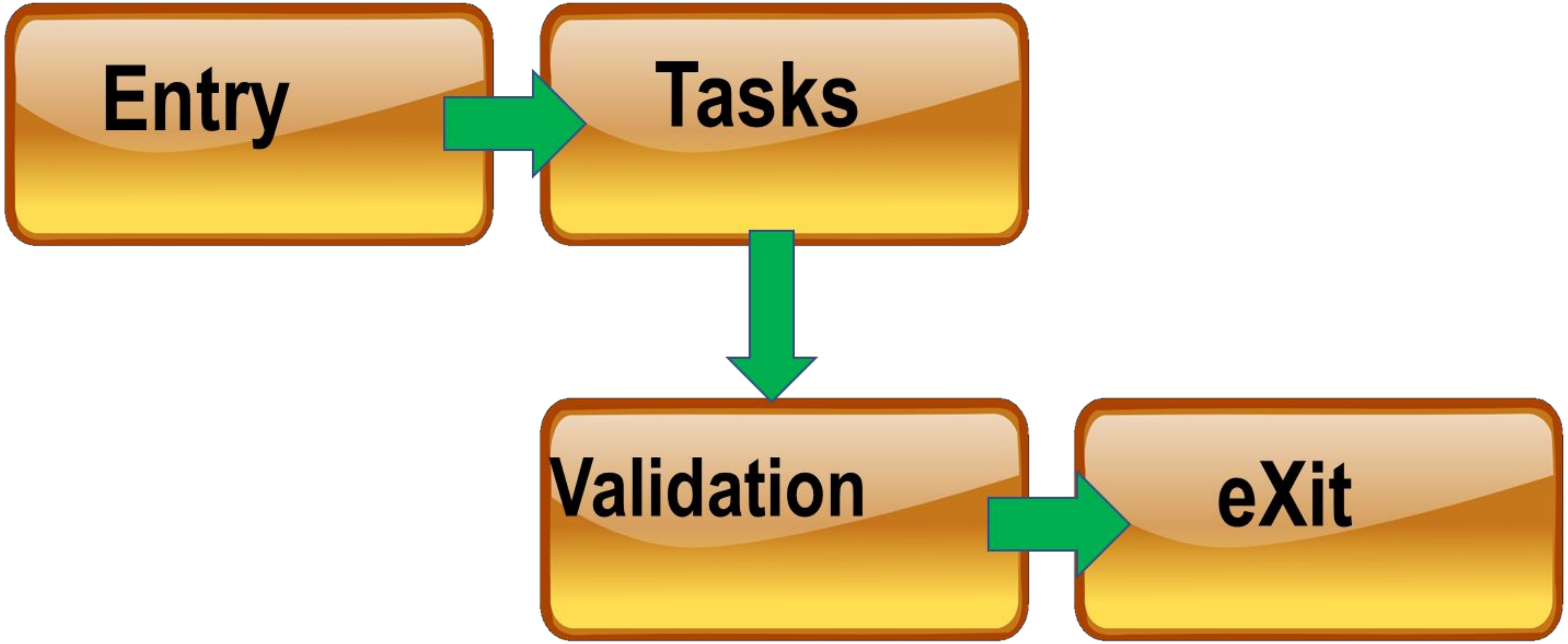
Disadvantages

❑ Cart-before-the-horse effect

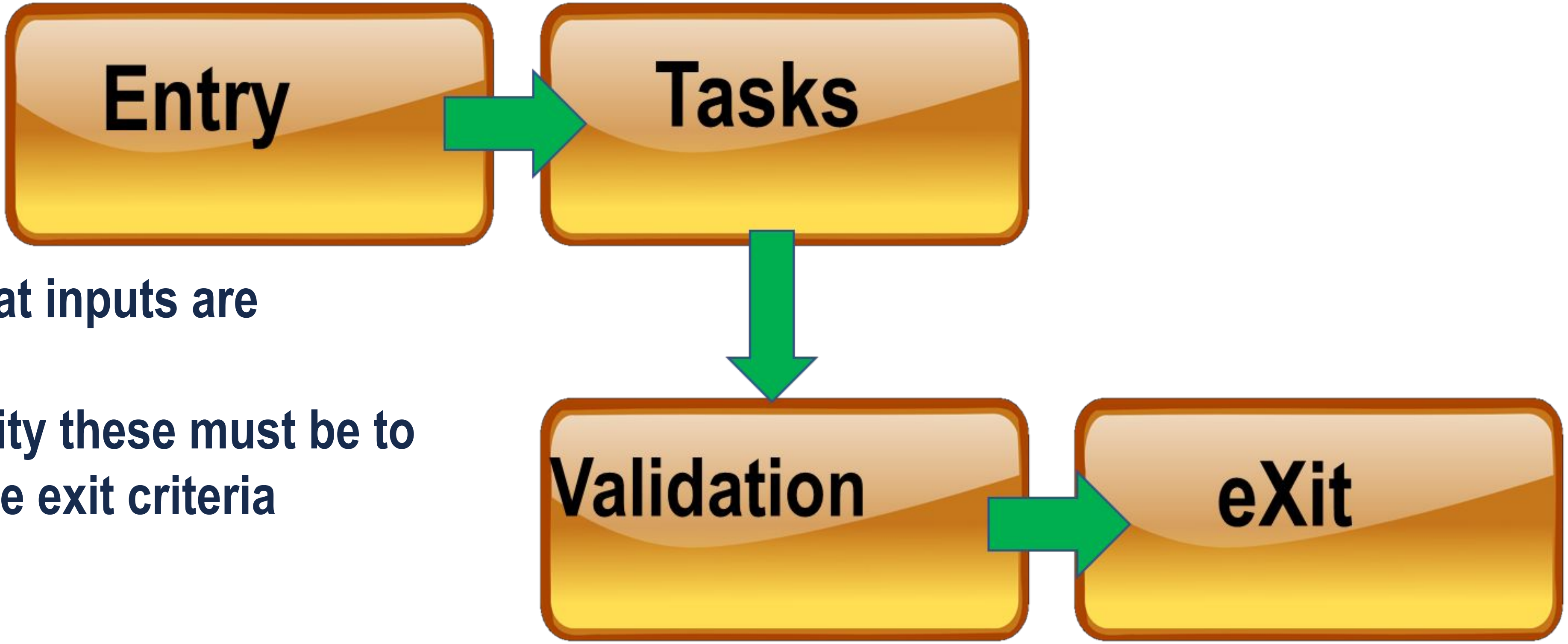
❑ Refactoring could become challenging

❑ Design volatility

ETVX Model of Tracking Tasks

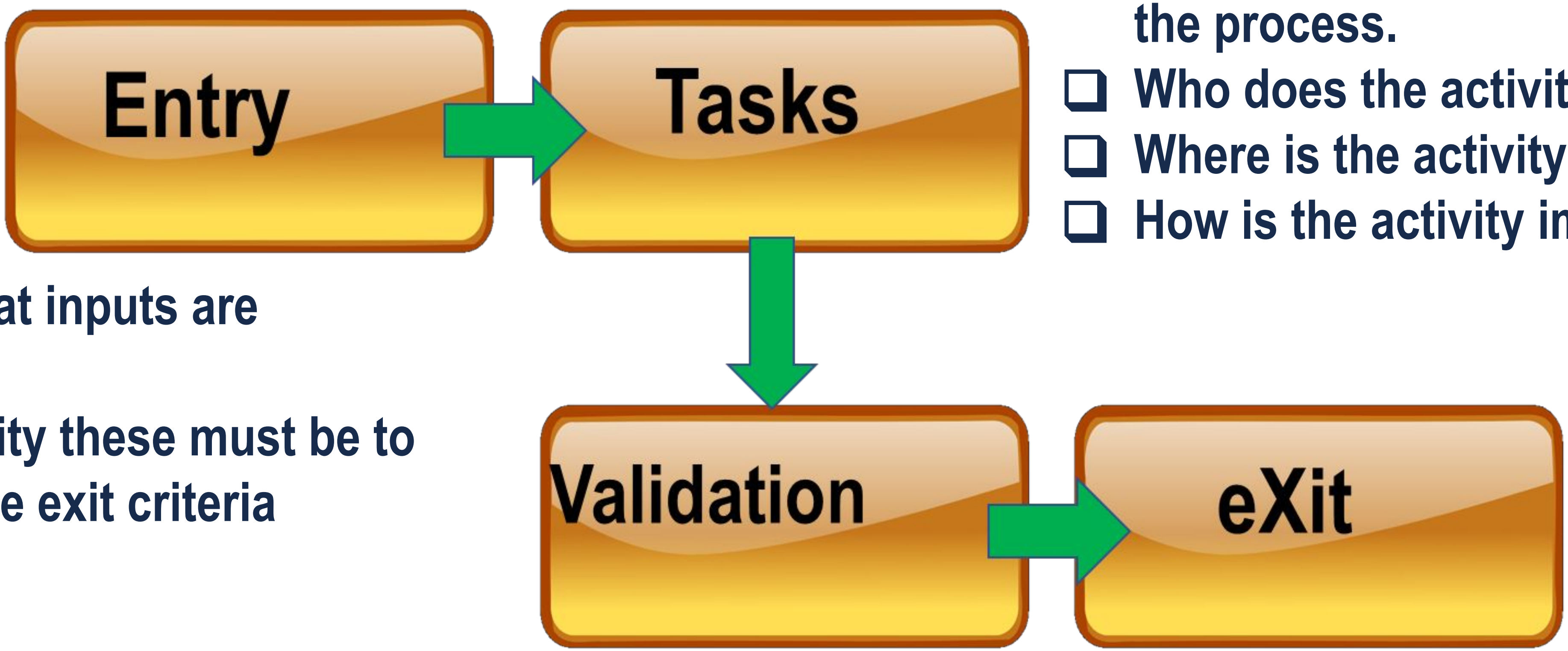


ETVX Model of Tracking Tasks



- ❑ Define what inputs are required
- ❑ What quality these must be to achieve the exit criteria

ETVX Model of Tracking Tasks



Entry

Tasks

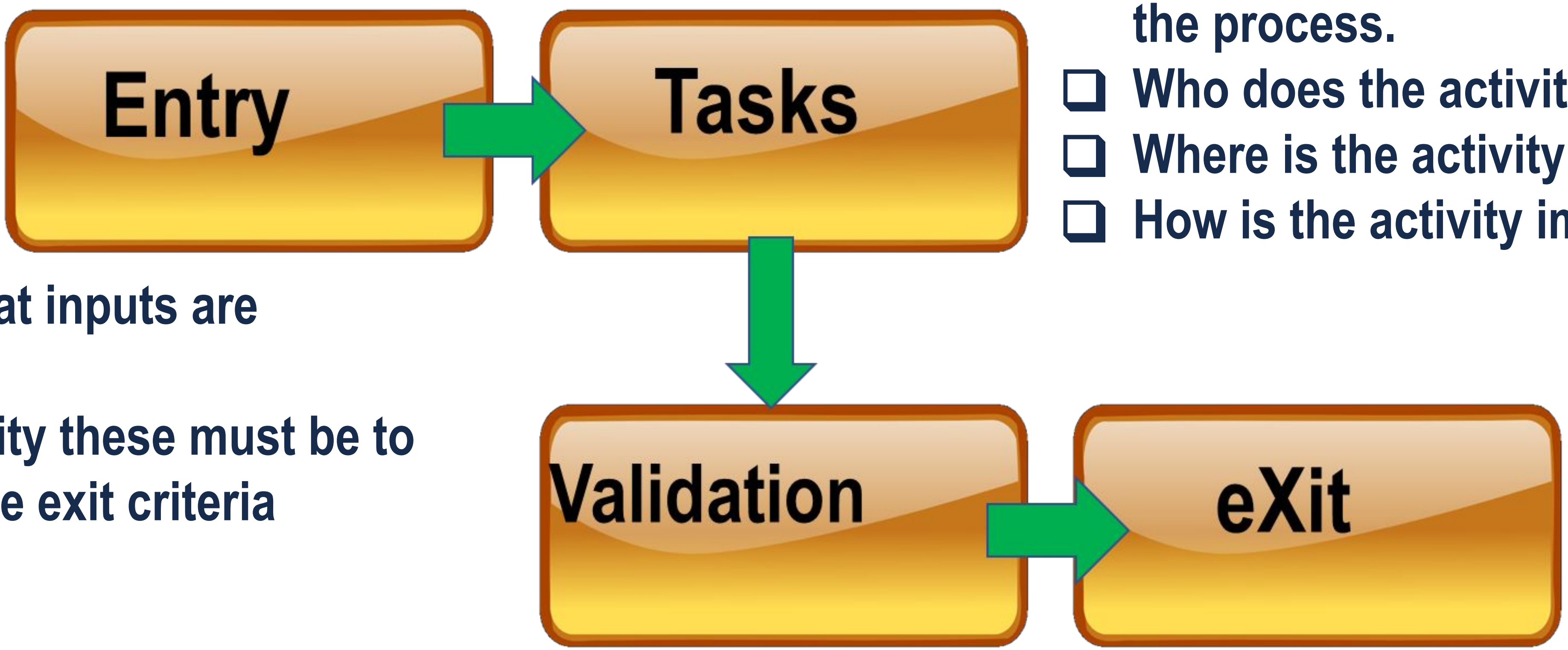
Validation

eXit

- □ Specify the actions that make up the process.
- □ Who does the activity?
- □ Where is the activity performed?
- □ How is the activity implemented?

- □ Define what inputs are required
- □ What quality these must be to achieve the exit criteria

ETVX Model of Tracking Tasks

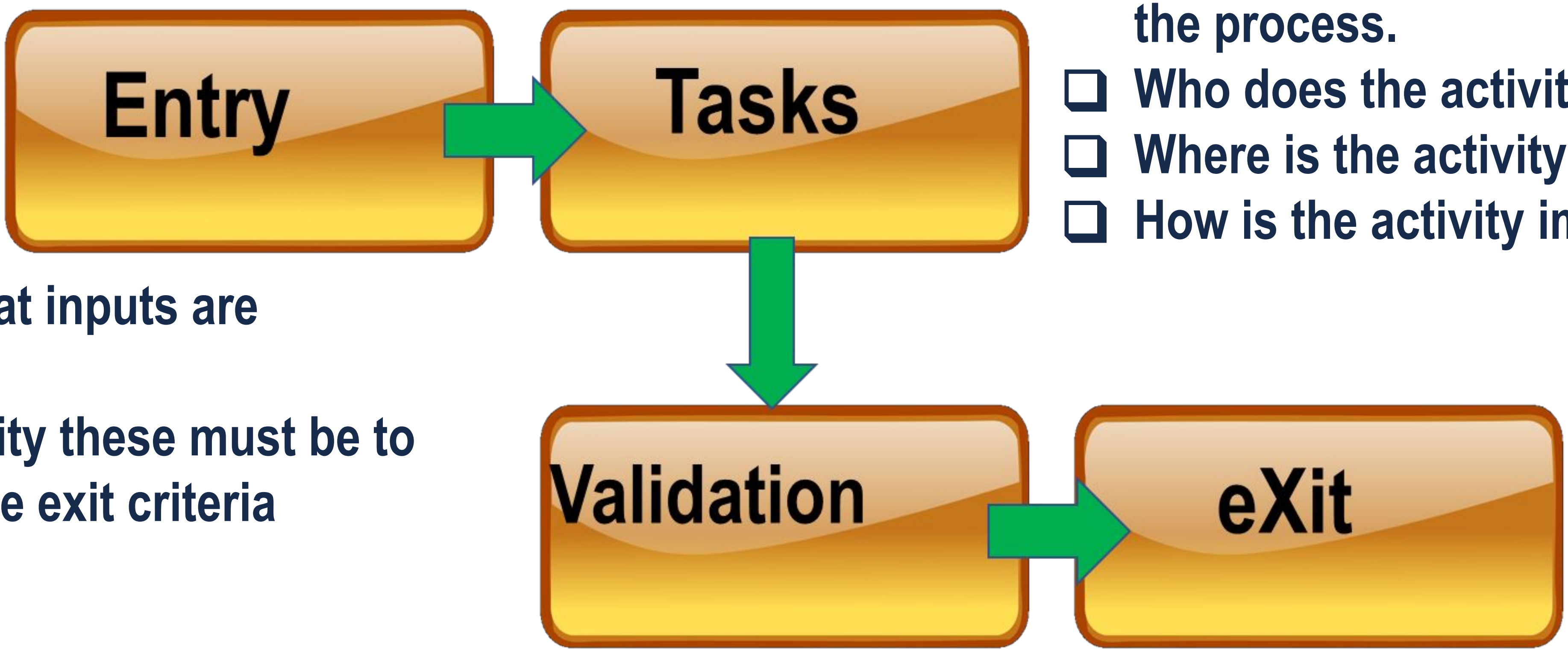


- Specify the actions that make up the process.
- Who does the activity?
- Where is the activity performed?
- How is the activity implemented?

- Define what inputs are required
- What quality these must be to achieve the exit criteria

- Identify test points within the process
- Define the tests and criteria for checking at these points
- This enables problems to be caught as quickly as possible

ETVX Model of Tracking Tasks



Entry

Tasks

Validation

eXit

□ Specify the actions that make up the process.

□ Who does the activity?

□ Where is the activity performed?

□ How is the activity implemented?

□ Define what inputs are required

□ What quality these must be to achieve the exit criteria

□ Identify test points within the process

□ Define the tests and criteria for checking at these points

□ This enables problems to be caught as quickly as possible

□ Define what outputs and quality level is required to meet the requirements of the process.

□ When does the activity end?