

SWE

Software Engineering (SE):

1. Formulate a problem
2. Analyse the problem
3. Searching for solutions fitting the problem
4. Deciding upon solution
5. Specifying the solution

Object-Oriented Software Engineering (OOSE):

1. Requirements elicitation
2. Analyse
3. System Design
4. Object Design
5. Implementation
6. Testing

1. Requirements elicitation:

Steps:

1. Identifying actors.
e.g. user, admin, IT guy

2. Identifying scenarios

3. Identifying use cases

A scenario is an instance of a use case.

e.g. login use case with scenarios:

- google;
- fb;
- system credentials.

4. Refining use cases

e.g.

Scenario:

Name: LogInGoogle

Actors: User

Flow of events:

1. User accesses system;
2. User is redirected to Login;
3. User chooses LogInWithGoogle;
4. User is logged in.

Use Case:

Name: LogIn

Actors: user, admin, IT guy

(Functional req.) Precondition: Actor is not logged in.

(Functional req.) Flow of events:

1. User accesses system;
2. User is redirected to Login;

3. User chooses LogInWithGoogle;

4. User is logged in.

(Functional req.) Past condition: Actor is logged in.

(Non-functional req.) Quality requirements: Actor is logged in within a maximum of 3 seconds response time.

5. Assessing relationships between actors and use cases.

Types:

1. Non-functional requirements:

Functionality

Usability

Reliability

Performance

Supportability

+

2. Functional requirements:

Use case diagram

Analysis object analysis model

System design design goals

system decomposition \Leftrightarrow off-the-shell components

Object design custom objects: SOLID

Single-responsibility principle

Open-closed principle

Liskov substitution principle

Interface segregation principle

Dependency inversion principle