

Software Engineering

Unit 3: Object-Oriented Software Estimation

Object-Oriented Software Estimation

3.1 Need of Object-Oriented Software Estimation

3.2 Loren and Kidd Estimation Method

3.3 Use case point estimation Method

3.4 Risk Management:

3.4.1 Introduction to Risk Management

3.4.2 Framework for Managing Risk

CE: 3.0 Object-Oriented Software Estimation

- After the requirements are gathered, the customer may like to know the estimate cost and time of the project.
- And to determine the cost and time of the project, the size and effort are required to measure.
- An accurate estimation has to be provided for any software project. But, in an early stage it is very difficult to determine, because of lack of information.
- To conduct effective software estimations, we must identify:
 - Scope/boundaries of the project
 - Size of the project
 - Effort of the project
 - Resources required in the project
 - Risk involved in the project

CE: 3.1

Need of Object-Oriented Software Estimation

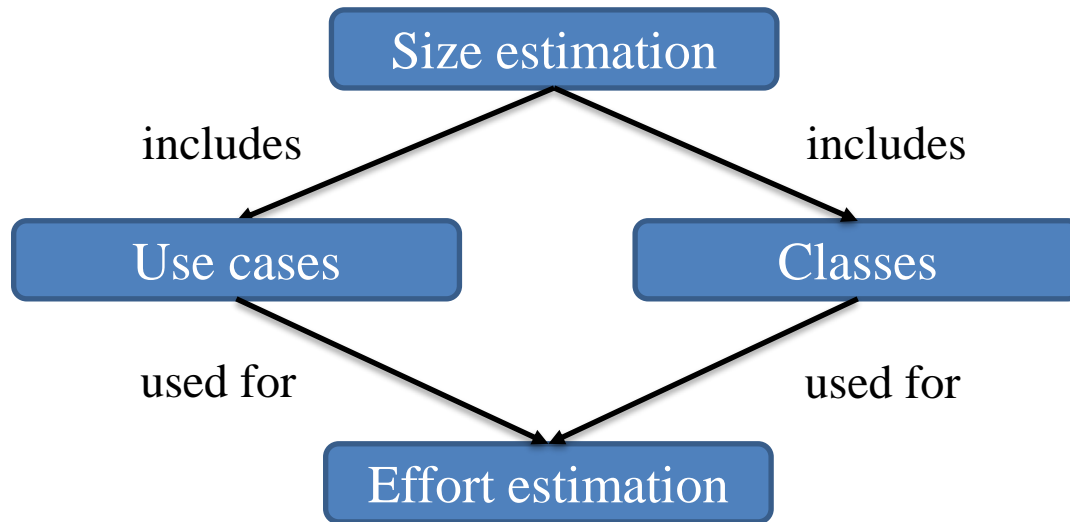
CE: 3.1 Need of Object-Oriented Software Estimation

- Traditional Software Estimations includes *lines of source code* and *function point analysis* for size and cost estimation.
- For *size*, following models are been used:
 1. COCOMO 81 (Constructive Cost Model)
 2. COCOMO II
- And for *cost*, following model is been used:
 - Putnam resource allocation model
- However, as the models are shifting from Traditional Software Estimation to Object-Oriented Software Estimation (OOSE), the estimation in OOSE may get change.

CE: 3.1 Need of Object-Oriented Software Estimation

- Object-Oriented Software Engineering uses UML (Unified Modelling Language) for creating models.
- Therefore, the *use case diagram* may be used to predict the size at an early stage of software development.
- *Classes* are also important element for measuring size in Object-Oriented Software.

CE: 3.1 Need of Object-Oriented Software Estimation



Framework of Object-Oriented Software Estimation

- The functionality of an Object-Oriented Software can be shown using use cases and these use cases can be transformed using classes.
- Therefore, the effort can be estimated using size estimation for Object-Oriented product.

CE: 3.2
Lorenz and Kidd
Estimation Method

CE: 3.2 Lorenz and Kidd Estimation Method

- Lorenz and Kidd provide two methods for estimation of number of classes:
 1. Use of scenario scripts
 2. Use of key and support classes

1. Use of scenario scripts:

- The number of scenario scripts may be used to estimate size of classes.
- i.e.

Number of classes = 17 x number of scenario scripts

2. Use of key and support classes:

- Key classes are specific to business applications and are ranked with higher priority by the customer. These classes also involve many scenario.
- Support classes are common for many applications. These classes include user interface, back end classes and communications.

CE: 3.2 Lorenz and Kidd Estimation Method (Conti...)

The following are used to determine the support classes:

Interface type	Multiplier
No GUI	2.0
Text-based user interface	2.25
Graphical user interface	2.5
Complex graphical user interface	3.0

Support classes can be calculated as:

$$\text{Support classes} = \text{Number of key classes} \times \text{Multiplier}$$

Finally, the total number of classes is obtained by adding key classes and support classes.

$$\text{Total number of classes} = \text{Key classes} + \text{Support classes}$$

According to Lorenz and Kidd, each class requires 10 to 20 person days for implementation. Thus, effort can be calculated as:

$$\text{Effort} = \text{Total number of classes} \times (10 \text{ to } 20 \text{ person days})$$

This method is easy and simple to understand.

CE: 3.2 Lorenz and Kidd Estimation Method (Conti...)

EXAMPLE 4.1 An application consists of 15 scenario scripts and requires 15 person days to implement each class. Determine the effort of the given application.

Solution Number of classes = $17 \times \text{Scenario scripts}$
 $= 17 \times 15 = 255$

Effort = $255 \times 15 = 3825$ person days

EXAMPLE 4.2 Consider the database application project with the following characteristics:

1. The application has 45 key classes
2. A graphical user interface is required

Calculate the effort to develop such a project given 20 person days.

Solution Number of key classes = 45

Number of support classes = Number of key classes \times Multiplier
 $= 45 \times 2.5$
 $= 112.5$

Total number of classes = Number of key classes + Number of support classes
 $= 112.5 + 45$
 $= 157.5$

Effort = 157.5×20
 $= 3150$ person days

CE: 3.4

Risk Management

CE: 3.4 Risk Management

- Identifying and managing risks in the early phases of software development life cycle can prevent software companies from facing major tragedies.
- Risk is defined as the probability of some unexpected happening in the future, and a situation that can cause unwanted results.
- Risk management involves identifying and estimation the probability of risks with their order of harshness.

CE: 3.4.1 Introduction to Risk Management

What is risk?

- Risk = Probability of occurrence of an undesired event x Impact of occurrence of that event
- Risk can delay the delivery of the software and over budget a project.
- The risky project may not also meet specified quality levels.
- There are various types of risks that may cause threat for the completion of a project. They are:
 1. Technical risks
 2. Economical risks
 3. Deployment risks
 4. Environment risks

CE: 3.4.1 Introduction to Risk Management (Conti...)

1. Technical risks:

- It deals with the possibility and understanding of the problem.

2. Economical risks:

- It involve budget, time, personnel and quality risks.

3. Deployment risks:

- It consist of mishandling of the software, inadequate user training and ineffective maintenance activities

4. Environment risks:

- No matter how much high quality software is produced, it cannot run without environmental support.
- The security and safety of the workplace is a very important issue that should be addressed.

CE: 3.4.1 Introduction to Risk Management (Conti...)

- Risk may be rated as...
 1. *Urgent:* Risks that would cause high loss to the business.
 2. *High:* Risks that would prevent the delivery of the software.
 3. *Medium:* Risk may affect the company from meeting a milestone.
 4. *Low:* Routine risks with little or no impact.

CE: 3.4.2 Framework for Managing Risk

Risk management is a key part of project planning activities and the specific risky areas are highlighted in the plan. The project plan is expected to highlight both probability of failure and impact of the failure and to describe the steps to be taken in order to reduce the risk. Risk management consists of the following steps:

1. Risk identification
2. Risk analysis and prioritization
3. Risk avoidance and mitigation
4. Risk monitoring

Steps 1–3 are carried out iteratively in order to identify, analyse, prioritize and reduce risks. The risks should be identified and appropriate plans must be made to reduce the high prioritized risks. After reducing the high prioritized risks, the impact of reducing these risks should be reanalysed and also risks that may have been introduced should be identified. For example, if a new technology is introduced in order to resolve risk for delay in the schedule of the software. However, the hiring of new experts might introduce new risks. After identifying new risks, the risk reduction and removal activities may be planned again.

CE: 3.3

Use case point estimation
Method

*Thank
You*