Course: Software Project Management

Week 7: Software Project Estimation

Lecturer: Yimer Amedie (MSc.)

Addis Ababa Science and Technology University, Ethiopia

Contents

- Introduction
- Software project estimation
- Estimation approaches

Learning Outcomes

After completing this lesson, you will be able to:

- Define project estimation and explain its importance in the context of software project management
- Describe the key factors that influence software estimation accuracy
- Differentiate between various software estimation approaches.
- Apply selected estimation techniques to sample software project requirements to produce effort, cost, and duration estimates

Introduction

- In project management, the cost, time, effort and other resources determines the project success.
- So, adequate estimation of time, budget and other resources matters in software projects.
  - **66%** of software projects exceed budget or timeline (Standish Group, 2020).
  - Poor estimation leads to rushed work, technical debt, and stakeholder distrust.

Estimation Overview

- Estimation is the process of **approximating** the value, cost, time, or resources needed to complete a task or achieve a goal.
- Project estimation refers to the process of **predicting**
  - The time
  - Cost
  - Resources
  - Effort required to complete a project successfully.
- It’s management goal.

Estimation Overview

- Project estimation should answer the following questions.
  - How much will it cost?
  - How long will it take?
  - How many people are needed?

Challenges

→Scope creep

→Unknown dependency

→Human factor

Software Project Estimation

What is Software project estimation?

- A process of predicting the effort, time, cost, and resources required to complete a software project. This is to
  - Define realistic deadlines
  - Prevent overruns
  - Improve project control
  - Enhance client communication and trust
- An essential component (but challenging) for project success

Software Project Estimation

Why Software Project estimation is required?

Software Project Estimation

Over Vs Under estimation

- Accurate estimation of software development is critical, yet challenging.
- Sometimes the estimation (Time, Budget, Resources) may be **higher** or **lower** than actually required.
- Both significantly impact the project success.
    - lead to budget and time overruns, resource wastage, team burnout and project failures

Software Project Estimation

Estimation Issues	Causes
Overestimation	- Inexperience
	- Fear of blame
	- Vague requirements
	- Risk padding
Underestimation	- Optimism bias
	- Incomplete requirements
	- Stakeholder pressure
	- Tech unfamiliarity

Causes of Over

and

Under estimation

Software Project Estimation

Problem of Over and Under estimation

Aspect	Overestimation	Underestimation
Cost & Budget	Wasted money	Cost overruns
Time & Schedule	Delays disguised as padding	Missed deadlines
Team Performance	Reduced urgency	Burnout and low morale
Quality	May be maintained	Often compromised
Stakeholder Trust	Perception of inefficiency	Perception of incompetence
Business Impact	Lost opportunities	Reputation damage

Overestimation leads to staff work less hard & increases mgmt. overhead. Underestimation compromise quality

Software Project Estimation

Example: OLMS (Overestimation)

- **Scenario:** Estimating the time required to implement a **discussion forum** feature.
- **Initial Estimate (Estimated Time – ET):**
  - Development: **4 weeks,** Testing & Bug Fixes: **2 weeks,** **Total ET: 6 weeks.**
- **Actual Outcome (Actual Time – AT):**
  - Development completed in **2 weeks**, Testing took **1 week**, **Total AT: 3 weeks**
- **Reason for Overestimation:**
  - Assumed custom development was needed.
- **Impact:**
  - Wasted resources, Delayed other features.

Software Project Estimation

Example: OLMS (Underestimation)

- **Scenario:** Estimating the effort for **user authentication & role-based access control**.
- **Initial Estimate (Estimated Time – ET):**
  - Development: **2 weeks,** Testing: **1 week,** **Total ET: 3 weeks**
- **Actual Outcome (Actual Time – AT):**
  - Development took **4 weeks**, Testing: **2 weeks**, **Total AT: 6 weeks**
- **Reason for Underestimation:**
  - Security requirements, third-party API limitations.
- **Impact:**
  - Missed deadline, delaying the entire project, increased costs.

Software Project Estimation

The basis for accurate software project estimation

1. Historical data
2. Parameters to be estimated
   - Duration ➔ months
   - Effort ➔ Work-month (wm), person-month(pm)
3. Measure of work
   - Size of the project ➔ **Independent Variable**
   - Effort and time ➔ **Dependent Variable**

Software Project Estimation

- Software project estimation is difficult due to
  - The complexity and invisibility of software
  - Changing technologies
  - Subjective nature of estimating
  - Political implication
  - Lack of homogeneity of project experience

Software Project Estimation

- Common estimation **metrics** for software projects
  - ✅🎯 **Scope** (features/requirements) → 30 requirements
    - **size** (line of code, function point, story point)
  - 🕒 **Effort** (person-month) → 180 person-month
  - 💲 **Cost** (budget required) → $700,000
  - 🕒 **Duration** (calendar time) → 6 months

Phases of Estimation Process

- It involves several phases to ensure that project requirements, costs, and timeframes are well understood and accurately predicted.

1. **Requirement Gathering and Analysis**
2. **Choosing Estimation Techniques**
3. **Estimating Resources and Effort**
4. **Estimating Cost**
5. **Risk Assessment and Adjustments**
6. **Review and Validate**

Phases of Estimation Process

1 Requirement Gathering and Analysis

- Understand and collect all relevant information about the project.
  - Gather detailed requirements from stakeholders.
  - Analyze the scope of work.
  - Clarify uncertainties and ambiguities in the requirements.
  - Break down high-level tasks into smaller, more manageable components.

Phases of Estimation Process

2 Choosing Estimation Techniques

- Decide on the methods or techniques to use for estimating.
  - Select appropriate estimation methods
    - Expert judgment, analogy-based estimation, parametric estimation, bottom-up estimation, etc.
  - Understand the limitations and assumptions behind each technique

Phases of Estimation Process

3 Estimating Resources and Effort

- Estimate the effort, resources, and time required for each task or component.
  - Estimate the effort required for each task
    - hours, days, or person-days.
  - Identify the necessary resources
    - Team members, tools, software).
  - Factor in possible constraints
    - Team skill levels, availability.

Phases of Estimation Process

4 Estimating Cost

- Estimate the total cost associated with the project.
  - Calculate the direct costs
    - labor, equipment, materials
  - Consider indirect costs
    - overhead, administrative costs
  - Factor in contingency costs to account for risks or uncertainties

Phases of Estimation Process

5 Risk Assessment and Adjustments

- Evaluate the potential risks and uncertainties that might affect the estimate
  - Identify potential risks that could affect the project
    - technology changes, resource availability.
  - Adjust estimates for uncertainties or risks, using buffers or contingency plans.
  - Revisit earlier phases to incorporate adjustments due to risks.

Phases of Estimation Process

6 Review and Validate

- Finalize and validate the estimates to ensure accuracy
  - Conduct a peer review or expert review of the estimates.
    - Ensure that they are realistic and complete
  - Validate the assumptions and data used in the estimation process.
  - Adjust the estimates as necessary based on feedback

Phases of Estimation Process

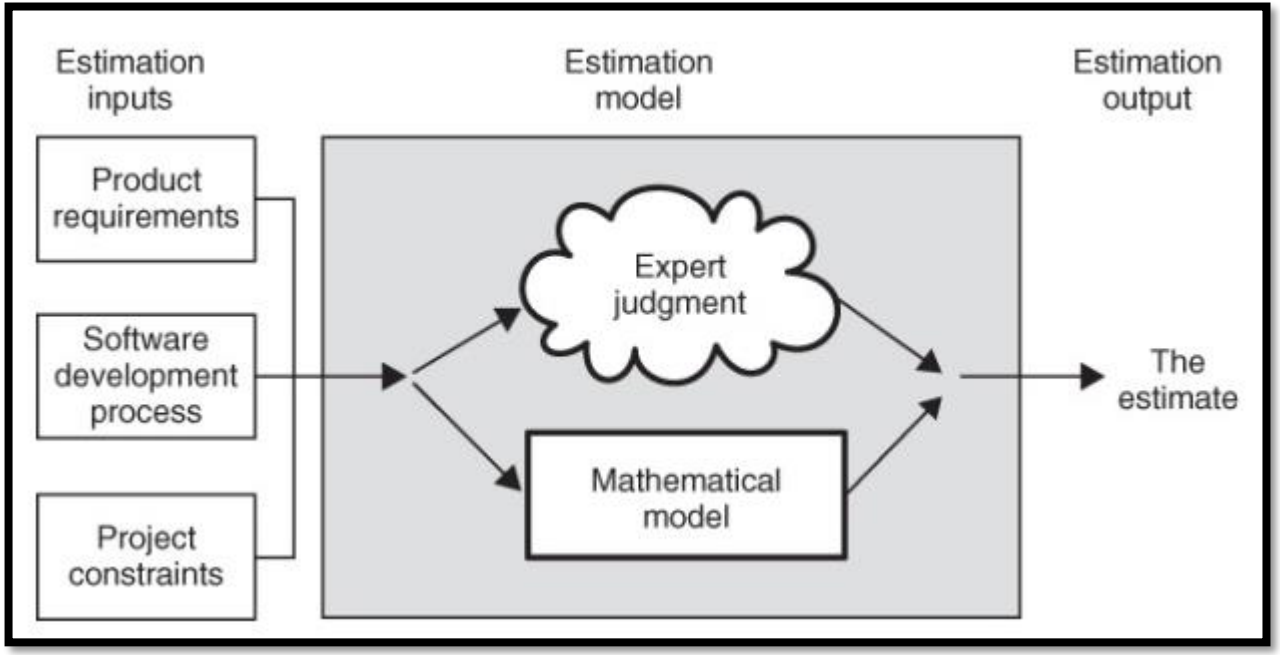


Figure 1: Perception of an Estimation Process (Abran, 2015)

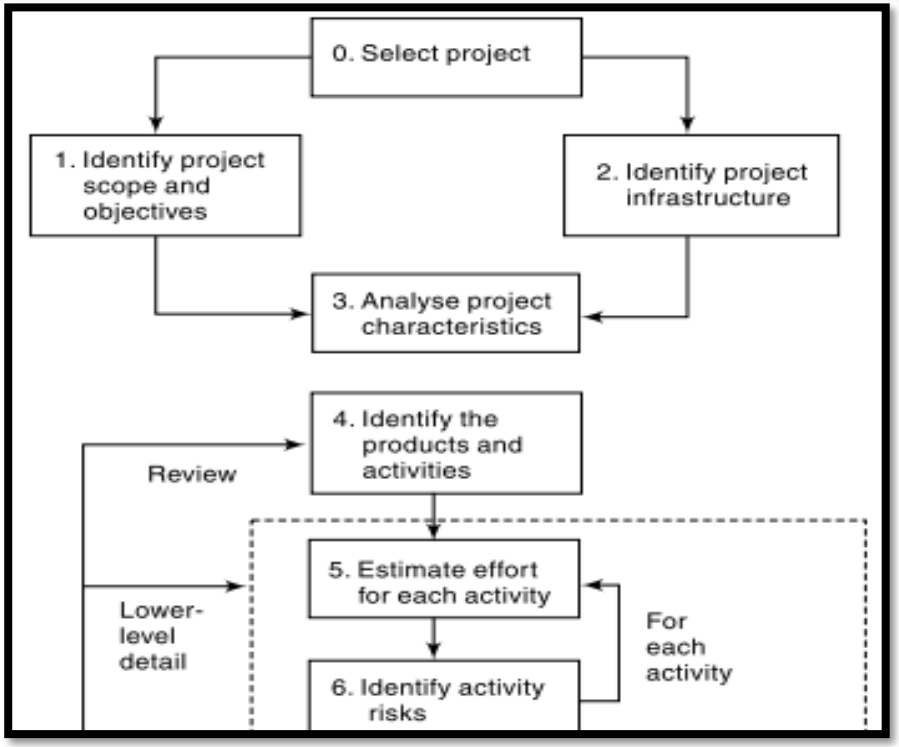
Discussion

1. **How poor estimation happen?**
   - Poor estimation
     - lack of historical data or experience, underestimating complexity
     - Optimism bias (over confidence in tools/techniques),
     - Failure to breakdown task
     - Poor communication

Estimation Across the Project Lifecycle

- The effort, time and cost estimation for software projects occurs
  - during the initiation and
  - at the beginning of the planning phases.
- Estimation is iterative,
  - It starts vague (initiation) and sharpens with each project management phases, reflecting growing certainty.

Estimation Across the Project Lifecycle



- In the stepwise project planning, software estimation takes place in the step 3 and 5 (Bob Huagoes, 2011).
  - Step 3: High level estimate
  - Step 5: detail estimate

Estimation Approach

Software Project estimation techniques (Bob Haghes, 2011)

- Algorithmic Models
- Expert Judgement
- Analogy
- Top-down
- Bottom-up
- Use case points
- Function Point Analysis
- Agile Estimation

Estimation Approach

Algorithmic models

- Use mathematical formulas to predict the required effort based on specific project characteristics (called **"effort drivers"**).
- These models often take into account factors such as **software size**, **complexity**, and the **environment** in which the software will be implemented.
- Particularly **useful for large-scale software projects** where manual estimation may be unreliable

Estimation Approach

Algorithmic models

- **COCOMO (Constructive Cost Model)**
  - Developed by Barry Boehm, COCOMO estimates effort based on **source lines of code (SLOC)** and complexity factors.
  - Types
    - **Basic:** Rough estimate based on project size.
    - **Intermediate:** Adjusts for cost drivers (e.g., team experience, complexity).
    - **Detailed:** Phase-wise effort estimation.

Estimation Approach

Algorithmic models

- **COCOMO (Constructive Cost Model)**
  - **Formula:**

$$\text{Effort} = a \times (\text{KLOC})^b$$

  - **KLOC** = Thousands of Lines of Code
  - **a, b** = Constants (depends on project type:
    - Organic,
    - Semi-Detached,
    - Embedded

Estimation Approach

Algorithmic models

COCOMO Constants for Different Project Types

Project Type	a	b	Example Projects
Organic (Small, simple)	2.4	1.05	Internal tools, scripts
Semi-Detached (Medium)	3.0	1.12	E-commerce, online learning systems
Embedded (Complex)	3.6	1.20	Flight control, real-time systems

Estimation Approach

Algorithmic models

- **Example: Online Learning Management System**
  - **Project Scope:**
    - **Features:** User registration, course enrollment, video streaming, quizzes, admin dashboard.
    - **Team:** 5 developers, 2 testers.
    - **Project Type:** Semi-Detached (moderate complexity).

Estimation Approach

Algorithmic models

- **Example: Online Learning Management System**
  - **Step 1: Estimate KLOC**
  - **Assume:**
    - **Frontend (React):** 15,000 LOC
    - **Backend (Node.js):** 20,000 LOC
    - **Database (SQL):** 5,000 LOC
    - **Total = 40,000 LOC➔ 40 KLOC**

Estimation Approach

Algorithmic models

- **Example: Online Learning Management System**
  - **Step 2: Select COCOMO Model**
    - For **Semi-Detached** projects: **a = 3.0**, **b = 1.12**
  - **Step 3: calculate effort**

$$\text{Effort} = a \times (\text{KLOC})^b$$

    - Effort= $3.0 \times (40)^{1.12}$ =3x63.5 ≈190.5person−months
- Therefore, the duration of the project is:
  - **Duration = 190.5/7 = 27.2 months.**

Estimation Approach

Algorithmic models

- **What if the team size is not determined?**
  - First calculate duration (time to complete the project):

$$\text{Duration} = c \times (\text{Effort})^k$$

Where:

- $c = 2.5$ (constant for all project types)
- k depends on project complexity:
  - **Organic**: $k = 0.38$
  - **Semi-Detached**: $k = 0.35$
  - **Embedded**: $k = 0.32$

For OLMS

Given: c =2.5, E = 190.5, k =0.35

Duration = $2.5 \times (190.5)^{0.35}$

≈14.2 months

Team size = Effort/Duration

≈13.4 people

Estimation Approach

Algorithmic models

- **Our initial estimation was 7 people and the COCOMO calculation is 13 people, what is its implication?**
  - Either the project will take longer time 27.2 months with 7 people, or
  - You need to hire more people (~13) to meet the 14.2-month deadline.

Estimation Approach

Algorithmic models

- **How much it will cost?**
  - Project cost can be obtained by multiplying
    - the estimated effort (in man-month, from the effort estimate) **with**
    - the manpower cost per month
  - Others (overhead or admin cost)

Project Cost = (Effort X Manpower Cost) + Other

Estimation Approach

- **Analogy (Comparative Estimation)**
  - A similar, completed, project is identified and its actual effort is used as the basis of the estimate.
  - Case based reasoning
    - **Source case, Target case**
  - Adjust estimates based on differences in scope, complexity, or technology.
  - Data-driven, useful when past projects are similar.
  - Requires a database of past projects
  - Less accurate if the new project is significantly different.

Estimation Approach

- **Analogy (Comparative Estimation)**
  - **Example: Online Learning Management System**
  - Suppose you previously built a **Course Management System** with:
    - Effort: **300 person-day,** Cost: **$600,000**
  - Estimating a new **Online Learning System** with:

    ☑ Same features as before (**Course Management**)

    ➕ Additional modules (**Payment System, Assessments**)

Estimation Approach

- **Analogy (Comparative Estimation)**
  - **Example: Online Learning Management System**
  - **Estimation Steps:**
    - Base estimate: **300 days** (for Course Management, same as before).
    - Add **100 days** for Payment System (based on another past project).
    - Add **50 days** for Assessments.
    - **Total Estimate:**
      - **450 days** (~1,000,000 *at* 150/hr, 12 hrs/day)

Estimation Approach

- **Expert judgement**
  - Relies on experienced professionals to estimate based on intuition, expertise, and domain knowledge.
  - Experts analyze requirements and provide estimates.
  - May use Delphi technique (multiple experts discuss & converge).
  - Useful for novel projects, but it may be subjective and biased

"Ask your senior dev: ‘How long did a similar feature take last year?’"

Estimation Approach

- **Top-down**
  - Starts with an overall estimate based on high-level requirements.
  - Breaks down the project into major components and allocates estimates.
  - Often based on expert judgment, historical data, or industry benchmarks.
- **Bottom-up**
  - Breakdown the project into its component tasks and estimate it (WBS)
  - Adding up the calculated effort for each activity to get an overall estimate.

Estimation Approach

Top-down VS Bottom-up

Factor	Top-Down	Bottom-Up
Speed	Faster	Slower
Accuracy	Low (Broad estimates)	High (Detailed breakdown)
Best Used When	Early-stage planning, rough budget	Detailed planning, precise scheduling
Risk of Error	Higher (may miss details)	Lower (validated per task)

Estimation Approach

Example: **Online Learning Management System**

- **Top-down**
  - Total project estimate ➔ 6 months, $200,000
    - User management (20%)
    - course management (35%)
    - Assessment (20%)
    - Admin Dashboard (25%)
  - Then for each task in each module

Estimation Approach

Example: **Online Learning Management System**

- **Bottom-up**
  - Breakdown the system in to modules → **Course Management Module**
  - Identify tasks

Task	Effort (Person-Days)	Cost ($)
Design Course	5	1,000
Develop Course Upload	10	2,000
Implement Search & Filters	8	1,600
User Enrollment	7	1,400
Testing & Bug Fixes	5	1,000
Total	35	7,000

Similarly, we estimate all other modules and sum them up.

Estimation Approach

- Pitfalls to Avoid
  - ☒ "It’s Just a Small Feature!"
    - ☒ Small tasks compound (e.g., "Add button" → redesign layout).
  - ☒ Ignoring Historical Data
    - ☒ *"Last year’s login module took 3x longer → why assume this one won’t?“*
  - ☒ Not Updating Estimates
    - ☒ Fixed estimates in dynamic projects **=** guaranteed miss.

Estimation Approach

- Best practices
  - ☑ Involve the whole team
    - ☑ Devs/testers know hidden complexities
  - ☑ Re-estimate Often
    - ☑ Adjust after each sprint/milestone.
  - ☑ Add Buffers
    - ☑ 20-30% for unknowns (e.g., bug fixes).
  - ☑ Document Assumptions
    - ☑ We assumed API docs would be ready by Week 2.

Summary

- Estimate the effort, time and cost required to complete the project.
- **Key Components are**
  - Effort Estimation (person-hours/days)
  - Time Estimation (duration)
  - Cost Estimation (budget)
  - Resource Estimation (team, tools)

"Estimation is a dark art but with practice, you'll get better at seeing in the dark."

Common Estimation Techniques

- Parametric Estimation
- Expert Judgment
- Analogous Estimation
- Bottom Up
- Use Case / Function Point Analysis

References

1. Standish Group. (2020). CHAOS Report 2020: Project success and failure rates. Standish Group International. Retrieved from https://www.standishgroup.com/
2. Bob Hughes, M. C. (2011). Software Project Management (5th ed.). McGraw-Hill.
3. Abran, A. (2015). Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers. Wiley & Sons, Inc., Hoboken, New Jersey.