Software Project Management

Durga Prasad Mohapatra
Professor
CSE Deptt.
NIT Rourkela

Project Estimation Techniques cont…

- A taxonomy of estimating methods
- Size Estimation

A taxonomy of estimating methods

- Top-down or Bottom-up - activity based, analytical
- Parametric or algorithmic models e.g. function points
- Expert Judgement - just guessing?
- Analogy - case-based, comparative
- Price to win

Pricing to win

- ‘Price to win’ is setting a target that is likely to win business when tendering for work
- The project costs whatever the customer can spend on it.
- Advantages: You get the contract
- Disadvantages: Costs do not accurately reflect the work required. Either:
  - (1) the customer does not get the desired system or
  - (2) the customer overpays.

Pricing to win

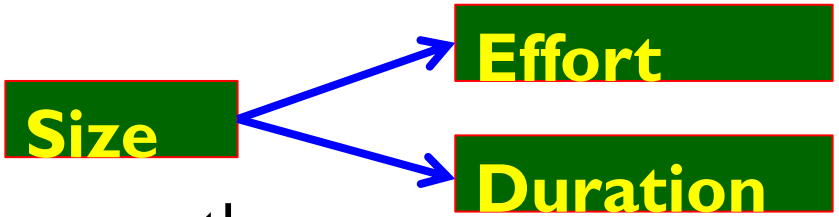
- This approach may seem unethical and unbusiness-like….
  - However, when detailed information is lacking it may be the only appropriate strategy…
- Which is the most ethical approach?
  - The project cost is agreed on the basis of an outline proposal and the development is constrained by that cost
  - A detailed specification may be negotiated or an evolutionary approach used for system development

Project Parameters to be Estimated

- For project planning, we need:
  - Effort (cost)
  - Duration
- Hard to estimate effort (or cost) or duration directly from a problem description.
- Effort and Duration can be measured in terms of project size (indirect metric)

Project Parameters to be Estimated cont …

- Size is a fundamental measure of work
- Based on the estimated size, two parameters are estimated:
  - **Effort**
  - **Duration**



- Effort is measured in person-months:
  - **One person-month is the effort an individual can typically put in a month.**

What is size? A Measure of Work…

- Project size is a measure of the problem complexity in terms of the effort and time required to develop the product.
- Two metrics are popularly used to measure project size:
  - Source Lines of Code (SLOC)
  - Function point (FP)
- SLOC is conceptually simple
  - But, FP is now-a-days favoured over SLOC
  - Because of the many shortcomings of SLOC.

Lines of code

- What's a line of code?
  - Originally proposed when programs were typed on cards with one line per card;
  - What happens when statements in Java span several lines or where there can be several statements on one line?
- What programs should be counted as part of the system? GUI? Built-in Class?
- Initially software development consisted of only writing code…

Lines of Code – Some Terminologies

- LOC ≡ Line of Code
- KLOC ≡
  - Thousands of LOC
- KSLOC ≡
  - Thousands of Source LOC
- NCKSLOC ≡
  - New or Changed KSLOC

LOC: Few things counter intuitive…

- The lower the level of the language, the more productive the programmer is:
  - The same functionality takes more code to implement in a lower-level language than in a high-level language.
- The more verbose the programmer, the higher the productivity:
  - Measures of productivity based on lines of code suggest that programmers who write verbose code are more productive than programmers who write compact code.

Major Shortcomings of SLOC

- Size can vary with coding style.
- Focuses on coding activity alone.
- Correlates poorly with quality and efficiency of code.
- Penalizes higher level programming languages, code reuse, etc.

Major Shortcomings of SLOC

- Difficult to estimate at start of a project from problem description
  - Only way to estimate is to make a guess…
  - So not useful for project planning
- Only a code measure
- Programmer-dependent
- Measures lexical/textual complexity only,
  - Does not consider structural or logical complexity.

Further Difficulties with SLOC

- SLOC can become ambiguous due to rapid changes in programming methodologies, languages, and tools:
  - Language advancements
  - Automatic source code generation
  - Custom software and reuse
  - Object-orientation

Expert Judgment-based Techniques

1. Basic expert judgment
2. Weighted average estimating
3. Consensus estimating
4. Delphi

Basic Expert Judgment

- One or more experts predict software costs.
  - Process iterates until some consensus is reached.
- Advantages: Relatively simple estimation method. Can be accurate if experts have direct experience of similar systems
- Disadvantages: Very inaccurate if there are no experts available!

Basic Expert Judgment Method: Steps

- Coordinator presents each expert with a specification and an estimation form.
- Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.
- Experts fill out forms anonymously
- Coordinator prepares and distributes a summary of the estimation on an iteration form.
- Coordinator calls a group meeting, specially focusing on having the experts discuss points where their estimates varied widely.
- Experts fill out forms, again anonymously, and Steps 4 and 6 are iterated for as many rounds as appropriate.

Basic Expert Judgement Method cont. …

- An expert is familiar with and knowledgeable about the application area and the technologies
- Particularly appropriate where existing code is to be modified.
- Research shows that expert judgement in practice tends to be based on analogy…

Stages

- Identify significant features of the current project
- Look at previous project(s) with similar features
- Observe differences between the current and previous projects
- Find out the possible reasons for error (risk)
- Take measures to reduce uncertainty

Estimation by Analogy

- The cost of a project is computed by comparing the project to a similar project in the same application domain:
- Advantages:
  - May be accurate if project data available and people/tools the same
- Disadvantages:
  - Impossible if no comparable project has been tackled.
  - Needs systematically maintained cost database

Basic Expert Judgement: Cons

- Hard to quantify
- It is hard to document the factors used by the experts or expert-group.
- Expert may be biased, optimistic, or pessimistic, even though they have been decreased by the group consensus.
- The expert judgment method always complements the other cost estimating methods such as algorithmic method.

Weighted average estimates

- Weighted average estimating is also known as sensitivity analysis estimating.
- Three estimates are obtained rather than one.
  - Best case (O = optimistic), worst case (P = pessimistic) and most likely (M = median).
  - This provides a more accurate estimate than when only one estimate is used.
- These are then used in the following formula:

Estimated effort = (O + 4M + P) / 6

Consensus estimating

Steps in conducting a consensus estimating session:

- A briefing is provided to the estimating team on the project.
- Each person is provided with a list of work components to estimate.
- Each person independently estimates O, M and P for each work component.
- The estimates are written up on the whiteboard.
- Each person discusses the basis and assumptions for their estimates.
- A revised set of estimates is produced.
- Averages for the O, M and P values are calculated.
- These values are used in the formula.

Delphi Estimation

- A variation of consensus estimation technique
- Team of Experts and a coordinator.
- Experts carry out estimation independently:
  - mention the rationale behind their estimation.
  - coordinator notes down any extraordinary rationale:
    - circulates the estimation rationale among experts.
- Experts re-estimate.
- Experts never meet each other
  - to discuss their viewpoints.

Delphi

- Delphi is an expert survey in two or more "rounds".
- Starting from the second round, a feedback is given (about the results of previous rounds).
- The same experts assess the same matters once more - influenced by the opinions of the other experts
- **important: anonymity**

Delphi Method: Steps

1. Coordinator presents each expert with a specification & an estimation form.
2. Coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.
3. Experts fill out forms anonymously
4. Coordinator prepares and distributes a summary of the estimation on an iteration form.
5. Coordinator calls a group meeting, specially mentioning the noted rationale where the estimates varied widely.
6. Experts fill out forms, again anonymously, and steps 4 and 6 are iterated for as many rounds as appropriate.

Types of Estimation Techniques

- Though there are many techniques of estimating, they can broadly be classified into:
  - Top-down
  - Bottom-up
- What about:
  - Algorithmic models?
  - Expert opinion?
  - Analogy ?
  - Price to win?

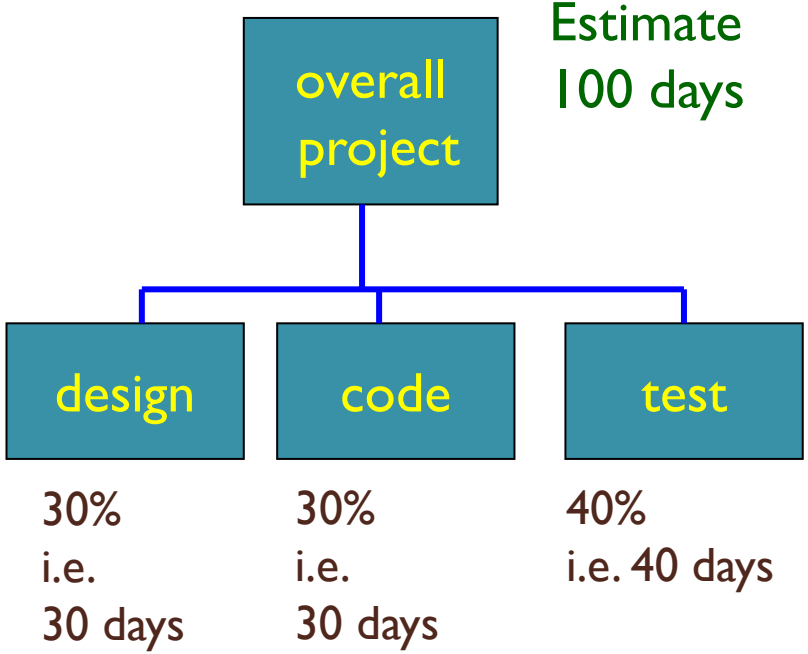
Bottom-up versus top-down

- Bottom-up
  - identify all tasks that have to be done – so quite time-consuming
  - use when you have no data about similar past projects
- Top-down
  - produce overall estimate based on project cost drivers based on past project data
  - divide overall estimate between jobs to be done

Bottom-up estimating

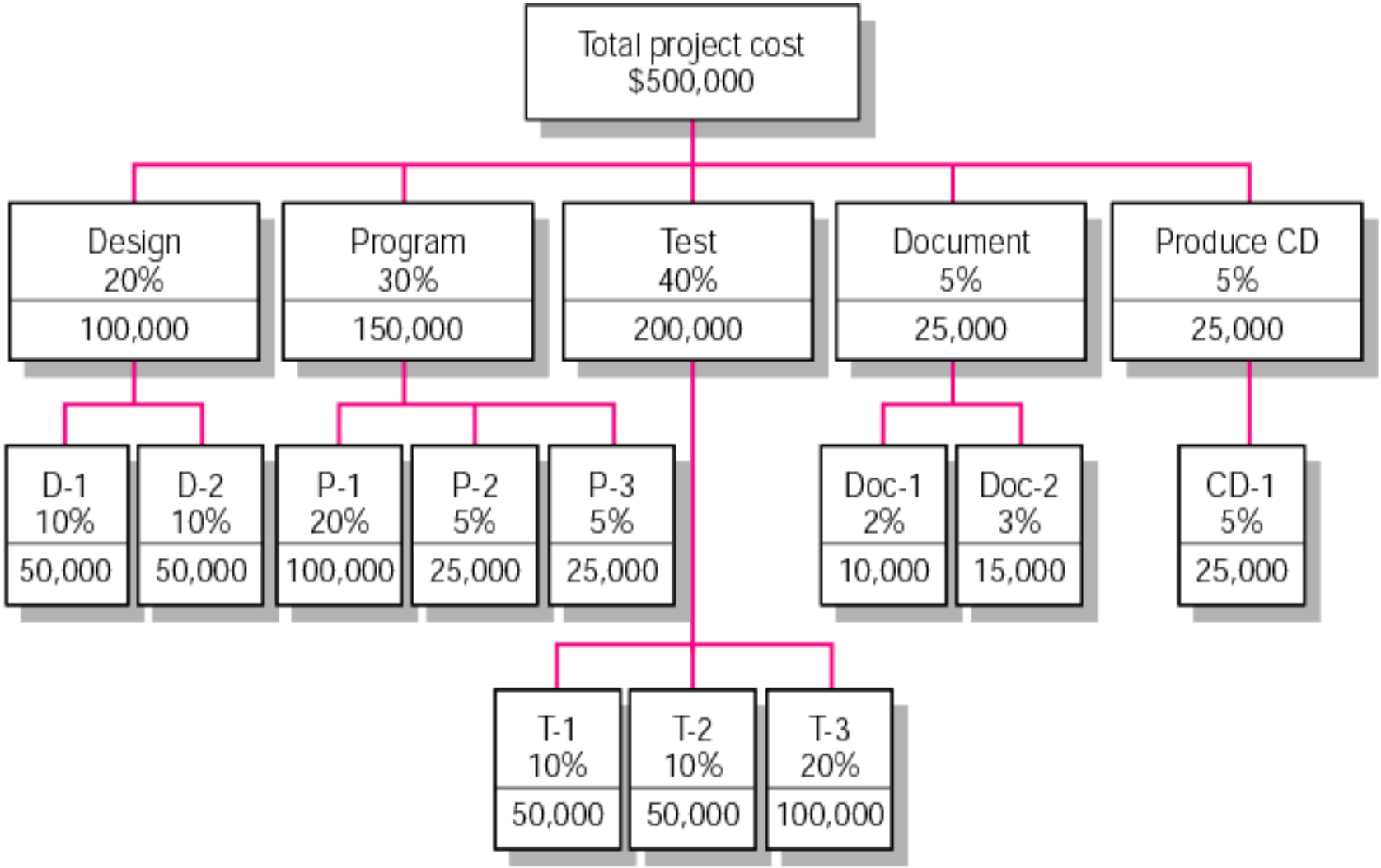
1. Break the project activities into smaller and smaller components
   - Stop when you get to what one person can do in one/two weeks
2. Estimate costs for the lowest level activities
3. At each higher level calculate estimate by adding estimates for lower levels

Top-down estimates



- Produce overall estimate using effort driver(s)
- Distribute proportions of overall estimate to components

Top-down Example



Top-Down Estimating: Pros

- It accounts for system-level activities such as integration, documentation, configuration management, etc.:
  - Many of these may be ignored in bottom-up estimating methods.
- It requires minimal project details:
  - It is usually faster, easier to implement.

Top-Down Estimating: Cons

- It often does not take into consideration the difficult low-level problems:
  - Tends to underestimate and overlook complexities of low-level components.
- It provides no detailed basis for justifying decisions or estimates.

Bottom-up Estimating: Pro

- It permits the software group to estimate in an almost traditional fashion:
  - Each group estimates components for which it has adequate experience.
- It is more stable because the estimation errors in the various components more or less balance out.

Bottom-up Estimating: Con

- It may overlook many of the system-level costs:
  - Integration,
  - configuration management,
  - quality assurance, etc.
- It may be inaccurate because the necessary information may not be available in the early phase.
- It tends to be more time-consuming.

Criteria for a Good Algorithmic Model

- Defined—clear procedure
- Accurate
- Objective—avoids subjective factors
- Results understandable
- Stable— Results valid for a wide range of parameter values
- Easy to Use
- Causal—future data not required

Algorithmic Models

- For project planning, we need:
  - Effort (cost)
  - Duration
- Hard to estimate effort (or cost) or duration directly from a problem description.
- Effort and Duration can be measured in terms of certain project characteristics that correlate with it:
  - Called size

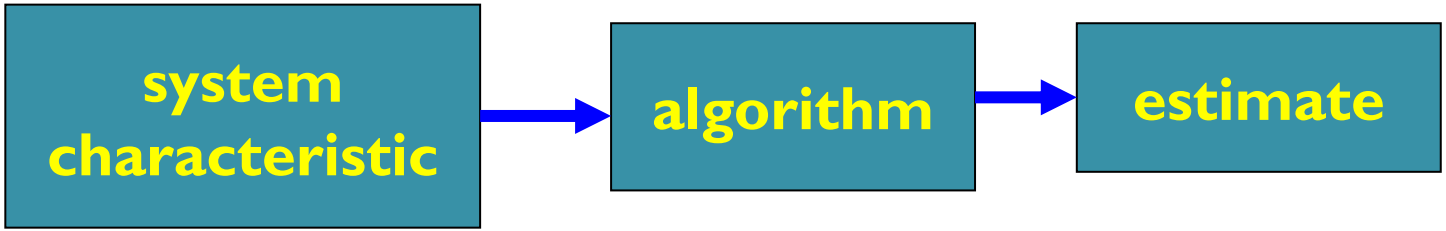
Software Size

- What exactly is the size of a software project?
  - How do you measure it?
- Any characteristic of software that is easily measured and correlates with effort.
  - SLOC
  - Function point

Algorithmic/Parametric models

- COCOMO (lines of code) and function points examples of these
- A Problem with LOC based models (COCOMO etc):

but what is desired is



Algorithmic Methods: Pro

- It is able to generate repeatable estimations.
- It is easy to modify input data, refine and customise formulas.
- It is efficient and able to support a family of estimations or a sensitivity analysis.
- It can be meaningfully calibrated to previous experience.

Algorithmic Methods: Cons

- It lacks means to deal with exceptional conditions, such as:
  - exceptional teamwork, exceptional match between skill-levels and tasks, etc.
- Poor sizing inputs and inaccurate cost driver rating will result in inaccurate estimation.
- Some factors such as experience cannot be easily quantified.

Model Calibration

- Many models are developed for specific situations and are, by definition, calibrated to that situation.
- Such models usually are not useful outside of their particular environment.
  - Calibration is needed to increase the accuracy of one of these general models.
  - Calibration is in a sense customising a generic model.
- Items that can be calibrated in a model include:
  - product types, operating environments, labour rates and factors, various relationships between functional cost items, etc.

Some recommendations

- Do not depend on a single cost or schedule estimate.
- Use several estimating techniques or cost models:
  - Compare the results and determine the reasons for any large variations.
- Document the assumptions made when making the estimates.
- Monitor the project to detect when assumptions that turn out to be wrong jeopardize the accuracy of the estimate.
- Improve software process:
  - **Maintain a historical database**

Simplistic Model

- estimated effort = (system size) / productivity
- As example:

  system size = lines of code

  productivity = lines of code per day

- productivity = (system size) / effort
  - based on past projects

What is wrong with the simplistic model?

Example 1

Consider a transaction project of 38,000 lines of code, what is the shortest time it will take to develop? Productivity is about 400 SLOC/staff month

Effort = $(\text{productivity})^{-1}$ (size)

= (1/.400 KSLOC/SM) (38 KSLOC)

= 2.5 (38) ≈ 100 SM

Min time = .75 T= $(.75)(2.5)(\text{SM})^{1/3}$

≈ $1.875(100)^{1/3}$

≈ 1.875 x 4.63 ≈ 9 months

Summary

- Discussed the Price to win estimation method.
- Presented Size estimation.
- Also, discussed Expert Judgement – based estimation techniques such as
  - Basic judgement
  - Weighted average estimating
  - Consensus estimating
  - Delphi Technique
- Explained the Analogy based estimation.
- Discussed top-down and bottom-up estimation techniques
- Discussed the algorithmic/parametric models for project estimation.

References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Sixth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.
2. R. Mall, *Fundamentals of Software Engineering*, Fifth Edition, PHI Learning Pvt. Ltd., 2018.

Thank you