Software Project Management

Lecture # 5

Outline

- **Estimation**
  - Introduction
  - Estimation and Risk
  - Project Planning Activities
    - Software Scope and Feasibility
    - Resources
    - Software Project Estimation
    - Decomposition Techniques
    - Empirical Estimation Models

Introduction

- Software Project Management begins with 'Project Planning' activities which include:
  - Estimation
  - Scheduling
  - Risk Analysis
  - Quality Management Planning
  - Change Management Planning

What is Estimation?

- Estimation is to determine how much…
  - Money/cost,
  - efforts,
  - resources and
  - time

  will be required to build a specific software based system or product.

Who does Estimation?

- Estimation is done by software project managers using information solicited from
  - Stakeholders,
  - Software engineers and
  - Software metrics data from past projects.

Estimation Steps – Summary

- Description of product scope.
- Decomposition of problem into set of smaller problems.
- Each sub problem is estimated using historical data, software metrics and experience (from past projects) as guides.
- Problem complexity and risks are considered before final estimate is made.

Estimation and Risk

- Estimation carries inherent risk & risk leads to uncertainty.
- Estimation risk is measured by the degree of uncertainty in the quantitative estimates established for resources, cost, and schedule.
- Availability of comprehensive historical information and software metrics (from past projects) helps establish better estimates and hence reduces risk factors.

Estimation and Risk (Contd.)

- If project scope is poorly understood or requirements are subject to change then uncertainty and estimation risk become dangerously high.
- Thus variability in software requirements means instability in cost and schedule.
- However, a project manager should not become obsessive about estimation as modern software engineering approaches are iterative in nature hence making it possible to revise estimates.

Software Project Planning Process

- Software project planning provides a framework that enables the manager to make reasonable estimates of resources, cost and schedule.
- Although there is inherent uncertainty, the team embarks on a project plan.
- But, this plan must be adapted and updated as the project progresses.

Software Project Planning Activities

- Establish project scope
- Determine feasibility
- Analyze risks
- Define resources
- Estimate cost and effort
- Develop project schedule

Lets discuss these in detail …

Software Scope

- It is defined in one of the following ways:
  - A narrative description developed after communication with all stakeholders
  - A set of use-cases developed by end-user
- It describes
  - functions & features to be delivered to end-user
  - Input and output data
  - “content” presented to users as they use the software
  - Performance considerations (processing and response time, etc)
  - Constraints (limits placed on software by external hardware, available memory, or existing systems), interfaces and reliability that bound the system

Software Feasibility

- Feasibility check is conducted after scope identification.
- It addresses questions like
  - Can we build software to meet this scope?
  - Is the project feasible?
- These questions are very crucial but often overlooked either by software engineers or by the impatient customers and managers.

Software Feasibility (Contd.)

- Putnam and Myers address feasibility in four dimensions
  - Technology
    - Is the project technically feasible? Is it within state of the art? Can defects be reduced as needed?
  - Finance
    - Is it financially feasible? Can the development be completed at a cost that the software organization, the client or the market can afford
  - Time
    - Will the project's time-to-market beat the competition?
  - Resources
    - Does the organization have enough resources needed to succeed?

Resources

- After scope and feasibility, next comes estimation of resources. Three major categories of resources are:
  - People/Human resources
  - Reusable software components
  - Development environment (s/w & h/w tools)

- Each resource has 4 characteristics
  - Description of resource
  - Statement of availability
  - Time when resource will be required
  - Duration of time when resource will be applied

1.Human resources

- This estimation involves
  - Selecting *Skills* (required to complete development)
  - Specifying *organizational positions* (manager, senior s/w engr, ..) and *specialty*
  - Determining *number of people* based on development effort
    - For small projects, single person can do all s/w engg tasks
    - For large projects, more number of people involved which may be geographically distributed. So, location of resource also specified
    - No. of people can be determined after estimating development effort (e.g., person-months)

2. Reusable Software Resources

- CBSE emphasizes the creation and reuse of software building blocks (components)
- 4 categories of software components
  - ***Off-the-shelf components***
    - Ready-to-use existing software acquired from third party (COTS) or from (internal) past projects
  - ***Full-experience components***
    - Existing specifications, designs, code, test data from past projects similar to software to be developed (for current project). May require little modifications

2. Reusable Software Resources

- ***Partial experience component***
  - Existing specifications, designs, code, test data from past projects related to software to be developed (for current project) but will require substantial modifications
- ***New components***
  - Software components that must be built for current project

3. (Development) Environment resources

- Software Engineering Environment (SEE) includes hardware and software support for a software project
- Hardware and software elements availability and time window must be specified

Software Project Estimation

- **Options** for cost and effort estimates:
  - Delay estimation until late in project
    - Not a practical approach
  - Base estimation on similar past projects
    - Reasonable approach but not always successful
  - Use simple decomposition techniques to generate estimates
    - Divide and conquer approach. Divide project into major activities/functions and make estimates
  - Use some empirical model for estimation
    - Complements decomposition techniques

- **Which option is better?**
  - Each approach can be used as a cross-check for the other

Decomposition Techniques

- Decomposition can be performed in two aspects
  - Decomposition of problem
  - Decomposition of process
- Estimation uses one or both forms of decompositions. But before this, software size must be estimated.

Software Sizing

- Proper estimation of software size and mapping of size estimate to human effort, calendar time and cost are important things which contribute to accuracy of overall software project estimation.
- In project planning, size refers to a quantifiable outcome.
  - **Direct approach** – size is measured as LOC
  - **Indirect approach** – size is measured as function-points

Software Sizing (Contd.)

- Putnam and Myers suggested 4 different approaches for sizing problem
  - Fuzzy logic sizing
  - Function point sizing
  - Standard component sizing
  - Change sizing

Other types of estimations…

- Problem based estimation (LOC-based and FP-based)
- Process-based estimation
- Use-case based estimation

References

- Today’s lecture contents were taken from chapter 23 – “Software Engineering, A Practitioner's Approach” by Roger Pressman