Software Project Planning: Overview

- Software planning involves estimating how much time, effort, money, and resources will be required to build a specific software system. After the project scope is determined and the problem is decomposed into smaller problems, software managers use historical project data (as well as personal experience and intuition) to determine estimates for each. The final estimates are typically adjusted by taking project complexity and risk into account. The resulting work product is called a project management plan.

Estimation Reliability Factors

- Project complexity
- Project size
- Degree of structural uncertainty (degree to which requirements have solidified, the ease with which functions can be compartmentalized, and the hierarchical nature of the information processed)
- Availability of historical information

Project Planning Objectives

- To provide a framework that enables software manager to make a reasonable estimate of resources, cost, and schedule.
- Project outcomes should be bounded by 'best case' and 'worst case' scenarios.
- Estimates should be updated as the project progresses.

Software Scope

- Describes the data to be processed and produced, control parameters, function, performance, constraints, external interfaces, and reliability.
- Often functions described in the software scope statement are refined to allow for better estimates of cost and schedule.

Customer Communication and Scope

- Determine the customer's overall goals for the proposed system and any expected benefits.
- Determine the customer's perceptions concerning the nature if a good solution to the problem.
- Evaluate the effectiveness of the customer meeting.

Feasibility

- Technical feasibility is not a good enough reason to build a product.
- The product must meet the customer's needs and not be available as an off-the-shelf purchase.

Estimation of Resources

- Human Resources (number of people required and skills needed to complete the development project)
- Reusable Software Resources (off-the-shelf components, full-experience components, partial-experience components, new components)
- Development Environment (hardware and software required to be accessible by software team during the development process)

Software Project Estimation Options

- Delay estimation until late in the project.
- Base estimates on similar projects already completed.
- Use simple decomposition techniques to estimate project cost and effort.
- Use empirical models for software cost and effort estimation.
- Automated tools may assist with project decomposition and estimation.

Decomposition Techniques

- Software sizing (fuzzy logic, function point, standard component, change)
- Problem-based estimation (using LOC decomposition focuses on software functions, using FP decomposition focuses on information domain characteristics)
- Process-based estimation (decomposition based on tasks required to complete the software process framework)

Empirical Estimation Models

- Typically derived from regression analysis of historical software project data with estimated person-months as the dependent variable and KLOC or FP as independent variables.
- Constructive Cost Model (COCOMO) is an example of a static estimation model.
- The Software Equation is an example of a dynamic estimation model.

Risk: Overview

- Risks are potential problems that might affect the successful completion of a software project. Risks involve uncertainty and potential losses. Risk analysis and management are intended to help a software team understand and manage uncertainty during the development process. The important thing is to remember that things can go wrong and to make plans to minimize their impact when they do. The work product is called a Risk Mitigation, Monitoring, and Management Plan (RMMM).

Risk Strategies

- Reactive strategies - very common, also known as fire fighting, project team sets resources aside to deal with problems and does nothing until a risk becomes a problem
- Proactive strategies - risk management begins long before technical work starts, risks are identified and prioritized by importance, then team builds a plan to avoid risks if they can or minimize them if the risks turn into problems

Software Risks

- Project risks - threaten the project plan
- Technical risks - threaten product quality and the timeliness of the schedule
- Business risks - threaten the viability of the software to be built (market risks, strategic risks, management risks, budget risks)
- Known risks - predictable from careful evaluation of current project plan and those extrapolated from past project experience
- Unknown risks - some problems simply occur without warning

Risk Identification

- Product-specific risks - the project plan and software statement of scope are examined to identify any special characteristics of the product that may threaten the project plan

- Generic risks - are potential threats to every software product (product size, business impact, customer characteristics, process definition, development environment, technology to be built, staff size and experience)

Risk Impact

- Risk components - performance, cost, support, schedule
- Risk impact - negligible, marginal, critical, catastrophic
- The risk drivers affecting each risk component are classified according to their impact category and the potential consequences of each undetected software fault or unachieved project outcome are described

Risk Projection (Estimation)

- Establish a scale that reflects the perceived likelihood of each risk
- Delineate the consequences of the risk
- Estimate the impact of the risk on the project and product
- Note the overall accuracy of the risk projection to avoid misunderstandings

Risk Table Construction

- List all risks in the first column of the table
- Classify each risk and enter the category label in column two
- Determine a probability for each risk and enter it into column three
- Enter the severity of each risk (negligible, marginal, critical, catastrophic) in column four
- Sort the table by probability and impact value
- Determine the criteria for deciding where the sorted table will be divided into the first priority concerns and the second priority concerns
- First priority concerns must be managed (a fifth column can be added to contain a pointer into the RMMM)

Assessing Risk Impact

- Factors affecting risk consequences - nature (types of problems arising), scope (combines severity with extent of project affected), timing (when and how long impact is felt)
- If costs are associated with each risk table entry Halstead's risk exposure metric can be computed (RE = Probability * Cost) and added to the risk table.

Risk Assessment

- Define referent levels for each project risk that can cause project termination (performance degradation, cost overrun, support difficulty, schedule slippage).
- Attempt to develop a relationship between each risk triple (risk, probability, impact) and each of the reference levels.
- Predict the set of referent points that define a region of termination, bounded by a curve or areas of uncertainty.
- Try to predict how combinations of risks will affect a referent level.

Risk Refinement

- Process of restating the risks as a set of more detailed risks that will be easier to mitigate, monitor, and manage.

- CTC (condition-transition-consequence) format may be a good representation for the detailed risks (e.g. given that <condition> then there is a concern that (possibly) <consequence>).

Risk Mitigation, Monitoring, and Management

- Risk mitigation (proactive planing for risk avoidance)
- Risk monitoring (assessing whether predicted risks occur or not, ensuring risk aversion steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problems)
- Risk management and contingency planing (actions to be taken in the event that mitigation steps have failed and the risk has become a live problem)

Safety Risks and Hazards

- Risks are also associated with software failures that occur in the field after the development project has ended.
- Computers control many mission critical applications in modern times (weapons systems, flight control, industrial processes, etc.).
- Software safety and hazard analysis are quality assurance activities that are of particular concern for these types of applications and are discussed later in the text.

Risk Information Sheets

- Alternative to RMMM in which each risk is documented individually.
- Often risk information sheets (RIS) are maintained using a database system.
- RIS components - risk id, date, probability, impact, description, refinement, mitigation/monitoring, management/contingency/trigger, status, originator, assigned staff member.

REFER TO CHAPTER 5 and 6 OF ***“Pressman. R. S., Software Engineering a practitioners Approach. 5th Edition”*** ACCORDINGLY.