Software Project Planning under Uncertainties

Kanika Arora & A.Yeshvini
B. Tech Dronacharya College ofEngineering,Gurgaon,India
kanikaarora286@gmail.com ; yeshvini.a.93@gmail.com

Abstract
Managing of risk and uncertainties throughout the course of a project has become one in all the priorities of the software system project manager.
Any analysis development comes square measure by oversized range of,
events may considerably modification the course of
events could type teams of connected eve nts or event chains. The paper discusses a projected methodology of modeling the system project planning mistreatment event chains, classification of the events and chains, identification of crucial chains, analysis of impact of the chains on project period, cost, and likelihood of project completion. The paper presents a sensible approach to modeling and visualizing event chains.

Introduction

Project programing is a vital step within the software
package development methopackage proj ectmanagers typically use programing to perform preliminary time and resource estimates, general steering, and analysis of project alternatives. one among the foremost challenges in software package project management is that it's tough to stick to the schedules as a result ofthe uncertainties associated with needs, schedules, personnel, tools, architectures, budgets, etc.This paper proposes a methodology for managing uncertainties based on an analysis of project events or groups of related events (event chains). The methodology can be easily understood by project managers who are not familiar with advanced statistical theory.

Heuristics and Biases

The problem associated with all the aforementioned methodologies lies in the estimation of project input variables: task durations, start and finish times, cost, resources, etc. If input uncertainties are inaccurately estimated, it will lead to inaccurate results regardless of the methodology of project scheduling.
Tversky and Kahneman [14] have proposed that limitations in human mental processes cause people to employ various simplifying strategies to ease the burden of mentally processing information to make judgments and decisions. During the planning stage, software project managers rely on heuristics or rules of thumb to make estimations. Under many circumstances heuristics lead to predictably faulty judgments or cognitive biases.

Following are short descriptions of some heuristics that affect the estimation of project variables for software project management.

The availability heuristic [2,13] is a rule of thumb in which decision makers assess the probability of an event by the ease with which instances or occurrences can be brought to mind. For example, project managers sometimes estimate task duration based on similar tasks that have been previously completed. If they are making their judgment based on their most or least successful tasks, it can cause inaccurate estimation.

The anchoring heuristic [14] refers to the human tendency to remain close to the initial estimate. For example, anchoring will lead to an overestimation of the success rate of the project with multiple phases because the chance of completion of each separate phase of the project can be an anchor in estimating the success rate for the whole project [9].

Judgments regarding the chance of a situation square measure influenced by quantity and nature of details within the situation during a approach that's unrelated to the particular chance of thesituation [12]. it's referred to as the representativeness heuristic. This heuristic will cause the "gambler's fallacy" or belief that a positive event is due as a result of a series of negative or undesirable events have already occurred.

Decision makers can be exposed to many cognitive and motivational factors that can lead to biases in perceptions. This effect is often referred to as selective perception.

For example, estimation of a task's cost can be influenced by the intention to fit the task into the project's budget. As a result, some of the project parameters can be overestimated.

Plus [11] has made some general recommendations for mitigating the negative impact of these and other heuristics. It is very important to keep accurate records and make estimations based on reliable historical data. Compound events should be broken into smaller events, which have known probabilities of occurrence. Discussion of best- or worst-case scenarios, for example the estimation of the most optimistic, the most likely, and the most pessimistic durations in PERT, can lead to unintended anchoring effects. To reduce dependence on motivational factors, Plous recommends the analysis of problems without taking expectations into account

Event Chains Methodology

The event chains methodology has been planned beat difficulties related to the estimation of project parameters, further on alter the method of project programing with uncertainties (schedule risk analysis) for computer code development.

According to the traditional project management methodology, the task (activity) is a continuous and uniform process. In reality, the task is affected by external events. These events can transform the task from one *state* to another. The state can be referred to as a process or part of the process with constant properties.
development.

In most cases, especially for research and development projects such as software development, it is difficult to predict potential events at the stage of project planning and scheduling. Events can occur stochastically during the course of a task. One task can be affected many multiple probabilistic events defined by the event properties: chance of occurrence, probabilistic

time of occurrence, and outcome (increase duration or cost, cancel task, assign or remove resource, etc.). These events will be included to the task's *list of events*. For example, during the course of development of the particular software feature, it may be discovered that the originally proposed software architecture is not appropriate. This discovery event may cause
the cancellation of the feature or even the project. It can also cause an increase in the task duration and cost. The chance of occurrence of this event based on the previous experience of development of similar tasks is 20%. Based on the same historical data, the event should occur during first two weeks of the development.

In addition to *probabilistic events*, there are also *conditional events*. A conditional event will occur if some conditions, related to project variables, are met. For example, if the task has reached a deadline, the event "cancel task" can be generated. It is possible to have a combined conditional probabilistic event. For example, if the deadline is reached, there is 20% chance that the task will be canceled.

The events can significantly affect the tasks, a group of tasks, and the whole project. Tasks within a group can have different relationships. It can be a summary task with subtasks. A group may also include tasks with joint resources or other common parameters, which can be affected by the same events. It is important to identify groups of tasks in order to simplify the process of modeling with events.

One event can lead to other events or create event chains. For example, an event of architectural change in the software can require refactoring of the software component. As a result, the resource will be pulled from another task, which will change a state: a task will be delayed. Therefore, one event (architectural change) may cause a chain reaction and eventually lead to major change in schedule for the whole project. Event chains can be presented by an *event chains diagram*

Fundamentally, calculations in event chains methodology square measure a variation of Monte Carlo simulations employed ancient risk analysis. throughout the
simulation method, project input variables(cost,
duration, begin and end time, likelihoo d completion) for every task are going to be calculated supported events. The results of calculation could be a distribution for
the length, begin and end time, success rate ,and value of the full project or any separate task.The results will be drawn withinthe type of frequencyor accumulative likelihood pl ots.

Event chains methodology is designed to mitigate negative impact of heuristics related to estimation of project uncertainties:

1. The task duration, start and finish time, cost, and other project input parameters can be influenced by motivational factors such as total project duration to much greater extent than events and event chains. It happens because events cannot be easily translated into the duration, finish time, etc. Therefore, event chains methodology can help to mitigate certain effects of selective perception in project management.

2. The event chains methodology relies on estimation of duration based on focused work on activity and does not necessarily require low, base, and high estimation or statistical distribution; therefore, the negative effect of anchoring can be mitigated.

3. The probability of event can be easily calculated based on historical data. It helps to mitigate the effect of the availability heuristic.

Single nts

Single events are the building blocks of the comprehensive probabilistic model of the software development process.

Each event has a number of properties. The events can affect the whole project, a group of tasks, a particular task, or the resource. For example, if it is discovered that a selected software tool does not provide the required functionalities, all tasks that are using this tool can be delayed.

The following types of events are commonly used in the software development project:

- Start and end tasks or group of tasks,
- Duration of a task or duration of each task within the group can be increased or reduced,
- Tasks or each task within a group can be canceled,
- Resources can be reassigned or a new resource can be assigned, and
- Whole projects can be canceled.

duration can be increased by 20%.

One task will have a gaggle of reciprocally exclusive events. as an example, there's a 2 hundredthprobability that length of a task are redoubled by thirty fifth, a half-hour probability that lengthcan increase by 100%,and a five-hitter probability that task can ought to be canceled. instead, the task are often at the same time suffering from some combination of those events. For instance, there's a 2 hundredth probability that length and value are often redoubled along.

The next property of the event is chronological. This parameter can be deterministic, but in most cases it is probabilistic. For example, the event can occur between the start time and end time of the task minus two days, but will most likely occur two weeks after the task has started. This information can be represented by the triangular statistical distribution.

The time once the event happens is very important. If the event leads to the cancellation of the task, to calculate the task length, it's necessary to understand once it occurred. This data is additionallycrucial once pursuit of project performance so as to filter events that might have occurred before the particular date. Finally, in sure cases, it's essential to understand once the event has occurred to calculate the new length.

Generating the Baseline Project Schedule

The first step in planning processes victimization eve nt chains is extremely kind of like what project managers do victimization ancient methodologie s. The project schedules are going to be created and bestowed within the style of a Gantt chart. The project manager ought to specify input project parameters, like period, begin and end time, cost, etc., that are related to a “best case scenario” or a targeted work on activity.

Executing Simulation and Analysis

To generate a schedule with uncertainties, town simulations ought to be performed employing abaseline project schedule and an incident list. the amount of simulations will be outlined supportedall-time low likelihood of the prevalence of events. The simulation will
be stopped once the results of simulations converge: that's, once the most calculation outputs (duration of, finish time, project cost, etc.) within a given number of simulations remain close to each other. Unfortunately, because of the discrete nature of the event chains, simulations will converge relatively slowly. In reality, the number of simulations can be between a few hundred to a few thousand. However, using modern computer hardware, Monte Carlo simulations for realistic

Conclusions

The planned event chains methodology is applicable to completely different time-related business or technological processes. The methodology will be terribly effective in code project
management,wherever it will conside rably change a method with multiple uncertainties.contain multiple uncertainties. A process that utilizes this methodology can be easily used in different projects, regardless size and complexity, using off-the-shelf software tools.

References

[1] Beck K.: Extreme Programming Explained – Embrace Challenge. Addison-Wesley, Reading, MA. (2000)

[2] Carroll, J.S.: The effect of imagining an event on expectations for the event: An interpretation in terms of availability heuristic. Journal of Experimental Psychology, 17, (1978) 88-96

[3] Cho J.G., Yum B.J.: An Uncertainty Importance Measure of Activities in PERT Networks. International Journal of Production Research, 12, (1964) 460-470

[4] Futrell R.T., Shafer D.F., Shafer L.I.: Quality Software Project Management, Prentice Hall PTR, Upper Saddle River, NJ, (2002)

[5] Goldratt, E.: Critical Chain. North River Press, Great Barrington, Mass. (1997)

[6] Hulett D.T.: Schedule Risk Simplified, PM Network, July, (1996) 23-30

[7] Jeffries, R., Anderson A., Hendrickson C.: Extreme Programming Installed. Addison-Wesley, Reading MA. (2000)

[8] Klastorin T.: Project Management. Tools and Trade-Offs. Wiley, (2004)

[9] McCray G.E., Purvis R.L., McCray C.G.: Project Management Under Uncertainties: The Impact of Heuristics and Biases. Project Management Journal. Vol. 33, No. 1. (2002) 49-57

[10] Newbold R.C.: Project Management in the Fast Lane: Applying the Theory of Constraints. St. Lucie Press, Boca Raton, FL, (1998)

[11] Plous S.: The Psychology of Judgment and Decision Making, McGraw-Hill (1993)

[12] Tversky, A., Kahneman, D.: Belief in the law of small numbers. Psychological Bulletin, 76, (1971) 105-110

[13] Tversky, A., Kahneman, D.: Availability: A heuristic for judging frequency and probability. Cognitive Physiology, 5, (1973) 207-232

[14] Tversky, A., Kahneman, D.: Judgment Under Uncertainty: Heuristics and biases. Science, 185, (1974) 1124-\1130