ABSTRACT

Time management plays an important role in project management; its processes include several components, and one of them is the activity duration estimation. In this thesis we review current methods and techniques used for duration estimation, then present a new way of using fuzzy logic (a mathematical technique used in artificial intelligence) to help a project manager better estimate activity durations.

Keywords: Time Management, Time estimation, Activity Duration, Duration Estimation, Project Management.

Executive Summary

The purpose of this thesis work is to develop a new methodology that improves the accuracy of time estimation in project time management. The algorithm created in this thesis was developed using fuzzy logic and designed to be used as an assistant tool for project managers. Fuzzy logic was used since it is a tool capable of modeling complex and uncertain or vague data using simple terminology such as IF-Then statements. This logic is perfect to deal with the uncertainty time constraints that can affect the time estimation in project development.

Our new developed methodology provides a quick and efficient tool for project managers in their use of project planning, by allowing the project manager to estimate project time based on similarities with other projects. This process is automated and the manager himself has no need to do any mathematical analysis with the other projects; the tool will do that when provided with the necessary information, which can be collected using several ways.

This thesis is divided as follows:

In chapter one, an introduction is made followed by literature review of current methods and techniques used in time estimation in project management, and the factors affecting it.

Chapter two reviews how information technology is used in current project management, and also the way artificial intelligence (a branch from I.T.) is used in this domain.

In chapter three, we introduce a new model for time estimating and this mathematical model is based on fuzzy logic. We will give a brief introduction of what fuzzy logic and its main principles are, then we propose our own model and an example of how our model can be used.

CHAPTER 1

PROJECT TIME MANAGEMENT

1.1 Introduction

An important aspect of Project Management is scheduling time accurately. This is a critical component of project planning as that will decide the deadline for the completion of a project. One important thing related to time management is credibility, since managers have to meet the deadlines they have committed to. When deadlines are not met and the timeline gets extended, project costs escalate accordingly [P. K. Suri, Bharat Bhushan, and Ashish Jolly. 2009]. This will impact the profitability of the project and it may lead to unnecessary stress and work load in the execution of the project.

Therefore a good estimation of the time period needed for the varied tasks in a project has a great importance in project planning. The usual problem is that a lot of managers underestimate the time needed to execute different types of jobs; this happens more frequently when the manager is not very familiar with certain type of tasks, which leads to improper judgment of time schedules [P. K. Suri, Bharat Bhushan, and Ashish Jolly. 2009].

Regardless of scope or complexity, any project goes through a series of stages during its life cycle. The Project Life Cycle refers to a logical sequence of activities to accomplish the project’s goals or objectives. Project development has increased importance in business due to strong competition and a fast changing business environment. Its cycle time is critical as it decides the success of a business. If the project is late to market by four months in a life cycle of five years, it loses one third of its profit [P. K. Suri, Bharat Bhushan, and Ashish Jolly. 2009].

At first there is an *Initiation or Birth phase*, in which the outputs and critical success factors are defined, followed by a *Planning phase*, characterized by breaking down the project into smaller parts/tasks, an *Execution phase*, in which the project plan is executed, and lastly a *Closure Project* activities must be grouped into phases or *Exit phase*, that marks the completion of the project. These usually are the general phases in a project's life cycle, which is independent

from the industry in which the project is done. It is still important to organize project phases into industry-specific project cycles because each industry sector involves specific requirements, tasks, and procedures when it comes to projects [T. A. Flynn. 2007]. But in this thesis our concern will be projects in general (no industry specific).

One true measure of the success of a project is doing the project “on time”. The other measure of project success is being “on budget” [George Angel. 2010]. These two areas (time and cost) are tightly linked and are the primary constraints (or challenges) project managers encounter on a day-to-day basis.

It has also been said that the highest source of conflict on a project is the schedule. More projects miss their target schedule dates than anyone would like to admit. The misses are often due to changes to scope or deliverables that affect the project’s scheduled completion date.

Time management, in a broad sense, involves both planning and execution. Time, unlike money, once spent, is gone and cannot be earned back. That is what makes time management an important knowledge area. There is no universally method or way to manage time. How time is managed and how activities are prioritized depend on the individual project manager. But there are some standard processes that, if followed, can help in this area [George Angel. 2010].

In this chapter we will review current methods or techniques used in project time management.

1.2 Literature Review

There are a lot of techniques that can help making time estimation easier, and there are also some software tools that can also help in this domain.

An attempt has been made to design a simulator for time estimation of project management process using Erlang-4 distribution. The input for the simulator has been derived by using an algorithm for generating pseudo random numbers which follows Erlang-4 distribution. This simulator will be an asset to affordably keep track of the time of phases during the process of project management. [P. K. Suri, Bharat Bhushan, and Ashish Jolly. 2009]

[S. Kumanan, and O.V. Krishnaiah Chetty. 2005] described and exemplified by a case study the application of Petri nets to determine product development time.

[Naveen Aggarwal, Nupur Prakash, and Sanjeev Sofat. 2008] in their paper proposed a Content Management System Effort Estimation model (CMSEEM), and explained that using it, an activity can be estimated more accurately.

[D. Phan, D. Vogel, and Nunamaker. 1988.] [D. Phan. 1990] tried to assess to what extent, and for what reasons, software development projects encountered cost and schedule overruns.

A survey of cost estimation in some organizations was conducted by [F.J. Heemstra, and R.J. Kusters. 1989] [F. J. Heemstra. 1992]. The goal was to provide an overview of the state of the art of estimation and controlling software development costs. Estimation methods, original project estimates and actual effort were analyzed.

[F. Bergeron, and J.-Y. St-Arnaud. 1992] performed a study to identify estimation methods, and to what extent they were used. They also investigated how choice of method, and underlying factors and variables, influenced estimation accuracy.

[T.T. Moores, and J.S. Edwards. 1992] sought to investigate why there was an apparent lack of use of software cost estimating tools.

[G. Wydenbach, and J. Paynter. 1995] analyzed the estimation practices on basis of a survey.

[T. Addison, and S. Vallabh. 2002] investigated the perceptions of project managers on software project risks and controls. A “snowball sample” was used to identify 70 managers, of whom 36 returned the questionnaire. Although not a study with a focus on estimation, it reported on aspects related to budgets and plans.

There is another way of looking at projects, and that is object-oriented projects, in which we need to keep and communicate schedules at different levels of detail.

We will see details of such techniques in the next paragraph.

1.3 Time Management Techniques

Activity duration estimating is the process for "estimating the number of work periods that will be needed to perform each schedule activity" [George Angel. 2010]. Duration can be estimated in units being freely defined by the project. And the duration often is also influenced by the power of the resources and vice versa. Therefore, resource estimating and duration estimating as processes should be connected.

1.3.1 The Simulation Approach

[P. K. Suri, Bharat Bhushan, and Ashish Jolly. 2009] attempted in their paper to introduce a quantitative analytical framework for modeling effort-boxed development in order to uncover the effects on the overall development effort and the potential leverage that can be derived from incremental delivery in such projects. Models that predict product size as an exponential function of the development effort are used in the paper to explore the relationships between effort and the number of increments, thereby providing new insights into the economic impact of incremental approaches to effort-boxed software projects.

1.3.2 Content Management System Effort Estimation Model (CMSEEM)

In the research that accompanied this model, data available were collected from twelve completed projects taken from industry and seventy different projects completed by the students.

These projects are categorized based on their size and total/build effort ratio. The size of the project is estimated by using the modified object point analysis approach. The complexity of the project is determined by using a set of questionnaire which has to be filled by the project managers after completing the initial requirement analysis. The nominal size estimated after object point analysis is finalized using the adjustment factors which are calculated by considering the different characteristics of the system such as production and general system characteristics. The estimated effort is further phase wise distributed for better scheduling of the project. The proposed model is refined using the linear regression approach. Finally the model is validated using another questionnaire which has to be filled after completing the project. The proposed model shows a great improvement as compared to the earlier models used in effort estimation [Naveen Aggarwal, Nupur Prakash, and Sanjeev Sofat. 2008].

1.3.3 Object Oriented Approach

In object oriented management, the high-level/low-detail view is used to communicate information to the outside world (senior managers, customers); the detailed view is for team's use. Since many people are affected by the top level, it has low flexibility. At the low level of schedule detail, fewer people are affected, so we have greater flexibility. Experience has shown that development projects are more productive and less risky when the project manager has some flexibility in meeting internal milestones.

We maintain three levels of schedule, with increasing detail and flexibility: Master schedule, Macroschedule and Microschedule [Murray Cantor. 1998].

Schedule Planning

Both the master schedule and the macroschedule should be maintained in a project management tool as a single project file, which is created by entering the WBS (work breakdown structure) into a tool and assigning start and end dates, or durations and dependencies. If we choose the latter, a tool will compute the dates for us.

Next, we add the program milestones; then identify which items are part of the master schedule and which belong to the more detailed macroschedule. Tools such as Microsoft Project let us identify the master schedule items in a text field so that the master schedule can be viewed by filtering on that field. In addition, we can enter master schedule milestone dates as *fixed*, so that the tool can highlight the impact of macroschedule slips.

If delivery is years away, it is essential that our master schedule include several builds with incremental functionality. As a rule of thumb, the builds should be completed once a year. The delivery or demonstration of each build not only gives us and our management an unequivocal view of the project's progress, it also sets a milestone that serves to focus the team.

There are four principles to consider as we create a master schedule. All address some aspect of risk management:

- Visible content. Prioritize the content that the customer must have, as well as that what we feel needs customer review and concurrence. The sooner we discover that the customer has a problem, the better.

- Technical uncertainty. If we are not sure how long it will take to develop a certain component, it makes sense to start it as early as possible.

- Critical functionality. Any core functionality required to deliver a working system should be in the first build. Our goal in the first build should be to deliver a (partially) working system. A working system allows for customer feedback, which in turn allows for adjustments in the second build. This helps minimize customer satisfaction risk.

- Resource leveling. Schedule functionality to reduce the chance of having to change staffing needs from build to build. It is often difficult to bring staff in and then take them off a project. Less than optimal use of staff adds to schedule and budget risk.

1.3.4 Other Techniques

Activity Duration Estimates are estimated durances for each activity given as values (ranges are indicated by +...- addons or percent failure values)

Duration includes the actual amount of time worked on an activity plus the elapsed time.

Effort is the number of workdays or work hours required to complete a task. Effort does not normally equal duration. People doing the work should help create estimates, and the PM should review them. Duration estimating is assessing the number of work periods (hours, days, weeks,) likely to be needed to complete each activity. Duration estimates always include some indication of the range of possible results, for example, 2 weeks + or – 2 days or 85% probability that the activity will take less than 3 weeks.

1.3.4.1 Expert judgement is used to determine how long an activity might be.

1.3.4.2 Analogous estimating is a method for estimating by comparing the actual unit with similar already realized / executed units. Therefore analogous estimating uses historical information and expert judgment.

Analogous estimating means that we use the actual time frame from a previous, similar project as the basis for estimating the time frame for the current project. The underlying premise is that this project is analogous to that project. Because we know how long that project took, we can estimate how long this project will take. Analogous estimating is used when little information is known about the current project or when two projects appear similar. Of course, if they appear similar but, in fact, are not, the estimate is inaccurate. Analogous estimation is good for ball-park figures but not for precise estimations of a project or timeline development.

1.3.4.3 Parametric estimating computes the durations by formulas which operate on more or less unsure values. Parametric modeling involves using variables from the project description in a mathematical formula to develop an estimate. For example, if we know the number of units that will be involved and how long it takes to create one unit, we can estimate the amount of time it takes to create 1000 units. If it takes our team 10 minutes a unit and we need 1000 units then it will take 10,000 minutes, which is 167 hours or a little over four weeks for

development. The formula is simple: the number of units multiplied by the amount of time each unit takes to create equals the total amount of effort needed for development.

1.3.4.4 Three-point estimating is a method which estimates not only one value (duration), but the value, which most likely will be reached, and an optimistic value and a pessimistic value. And on the base of these estimated values the method computes variances and probabilities.

Three-Point Estimating is a quantitative analysis technique that assigns numerical values to define a range of possible out-turns so that Risk Analysis may be carried out to better inform decisions. Three-Points Estimating may also be referred to as PERT (Program Evaluation and Review Technique). It makes sense in the real world that we do not really know how long a particular activity will take, specially talking about certain activities such as research and development. In this case, we can look at the project completion time in a probabilistic fashion and for each activity we can define:

Optimistic time estimate: an estimate of the minimum time an activity will require.

Most likely time estimate: an estimate of the normal time an activity will require.

Pessimistic time estimate: an estimate of the maximum time an activity will require.

Project managers can use three-point estimating to gain a greater degree of control over how the end value is calculated. The end value is the weighted average of three estimates. To do three-point estimating, in its simplest form for a particular task or activity, we ask the staff resource for their best, most likely and worst estimates. Add the Pessimistic estimate to four times the most likely, then we add the optimistic and divide by six. This gives us our estimate, which is a slightly more balanced view of how long the task or activity is likely to take. The calculation is (P + 4M +O / 6).

1.3.4.5 Precedence Diagramming Method (PDM)

PDM is also referred to as Activity-on-Node (AON). The PDM puts the activities in boxes, called nodes, and connects the boxes with arrows. The arrows represent the relationship and the dependencies of the work packages. Example:

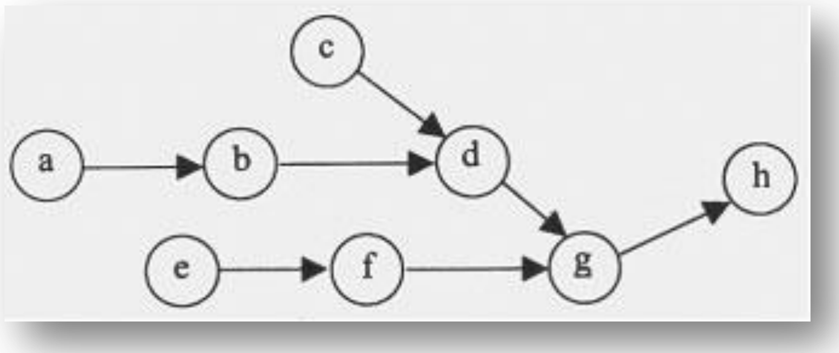


Fig. 1 An example of PDM

PDM includes four types of dependencies or precedence relationships: Finish-to-Start, Finish-to-Finish, Start-to-Start and Start-to-Finish. They are illustrated in the following figure:

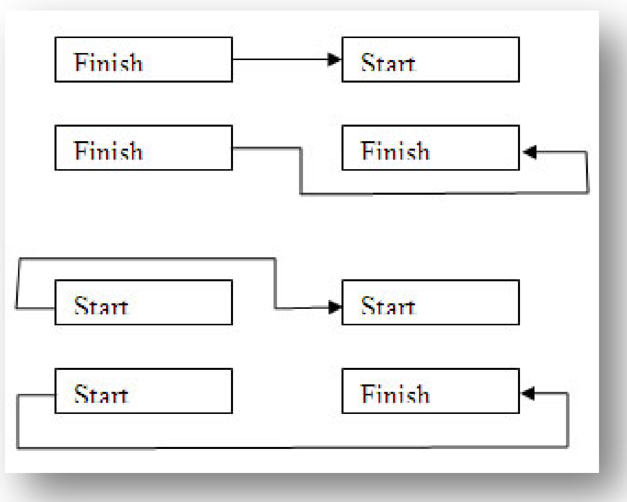


Fig. 2 PDM dependencies relationships

1.3.4.6 Arrow Diagramming Method

The arrow diagramming method (ADM) is similar to the PDM, except that all dependencies Activities and durations are generally depicted on the arrows. For this reason, the ADM diagram is also called the Arrow-on-Arrow (AOA) diagram. Example:

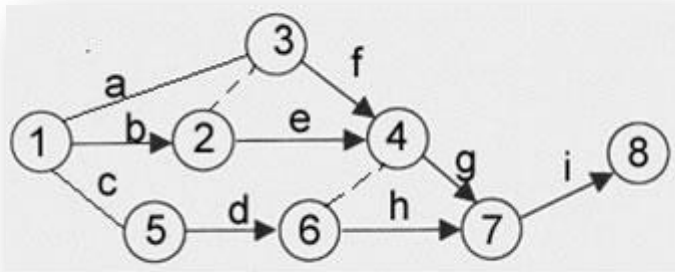


Fig. 3 An example of ADM

ADM uses only finish-to-start dependencies and can require the use of “dummy” relationships called dummy activities, which are shown as dashed lines, to define all logical relationships correctly. Since dummy activities are not actual schedule activities (they have no work content), they are given a zero value duration for schedule network analysis purposes.

There are some concepts that we will need to clarify:

- *Dependencies*

There are several types of dependencies:

- Mandatory dependencies: often involve physical or technological limitations of the work; for example, a prototype must be built before it can be tested.
- Discretionary dependencies: Also referred to as preferred logic, preferential logic or soft logic and are based on knowledge of best practices within a particular application area.
- External dependencies: any input that is needed from another project or source outside the project team.

- *Float, Leads & Lags*
- Free Float - The amount of time an activity can be delayed without delaying the next scheduled activity's start date.
- Total Float - The amount of time an activity can be delayed without delaying the project's finish date.
- Lead - allows an acceleration of the successor activity.

- Lag - directs a delay in the successor activity.

1.3.4.7 Reserve analysis is the process of computing a reserve into the estimated results.

1.3.4.8 Software Tools and Charts

- **myPmpsFactory** offers a structured template for activity lists which allows to insert the durations (work days) of the activities. This template can be edited.
- **Planner** offers a project schedule file which also allows to insert the durations of the activities. Next is a screenshot of the planner software viewing tasks and their estimated duration.

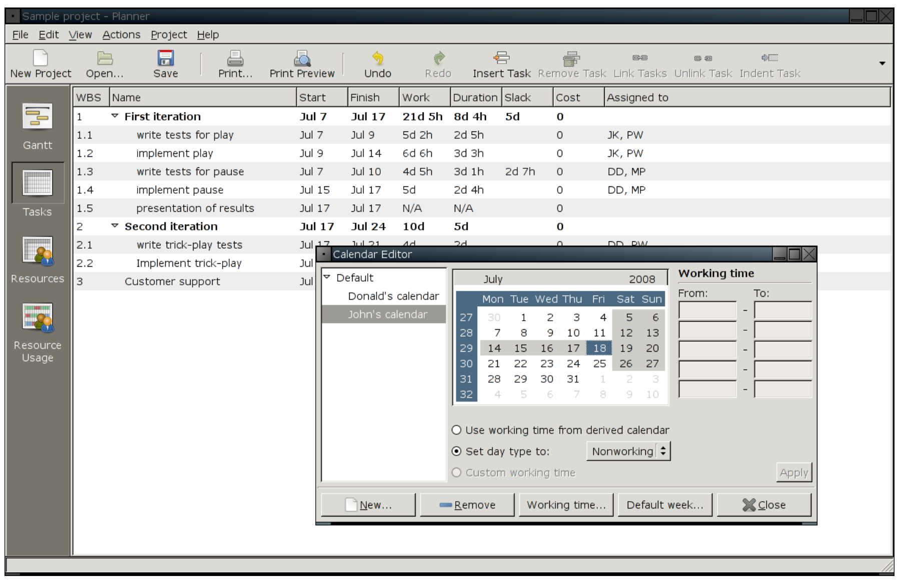


Fig. 4 A screenshot of Planner

- **Gantt Charts:** Schedules are usually maintained in a Gantt chart. The chart is a visual representation of the planned tasks and events. The column on the left lists the WBS according to key events such as formal review meetings or artifact delivery. Such events are called milestones. Milestones do not have preset durations and do not have budgets associated with them. The chart shows the placement of the tasks and events in time, which is depicted from left to right. Logically, each task bar starts at the beginning of the task and ends at the conclusion of the task [Murray Cantor. 1998.].

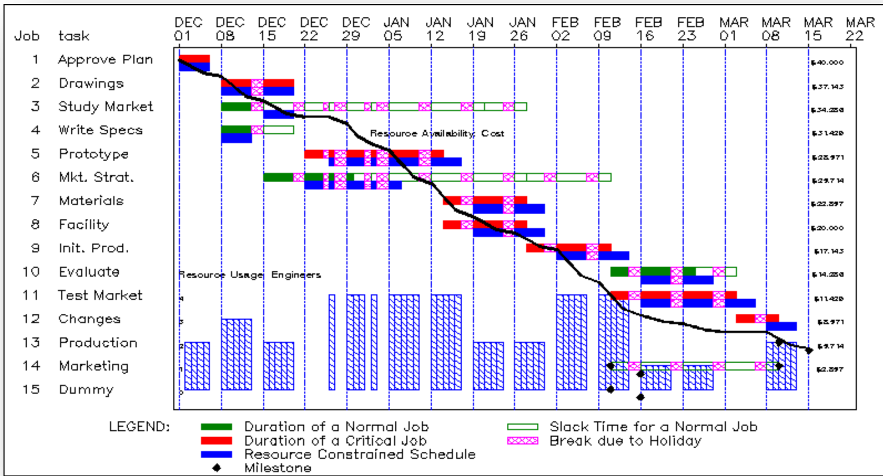


Fig. 5 A Gantt chart example

A project manager needs to become familiar with Gantt charts. Software tools, such as Microsoft's Project or Primavera's SureTrak, are available for building and maintaining Gantt charts. Gantt charts have other useful features, such as resource leveling and critical path analysis [Murray Cantor. 1998].

- **Using PERT**

In recent years, the range of project management applications has greatly expanded. Project management concerns the scheduling and control of activities in such a way that the project can be completed in as little time as possible.

PERT is a well known technique with proven value in managing large-scale projects. The traditional PERT (Program Evaluation and Review Technique) model uses beta distribution as the distribution of activity duration and estimates the mean and the variance of activity duration using "pessimistic", "most likely" and "optimistic" time estimates proposed by an expert. In the past several authors have modified the original PERT estimators to improve the accuracy [N. Ravi Shankar, K. Surya Narayana Rao, and V. Sireesha. 2010].

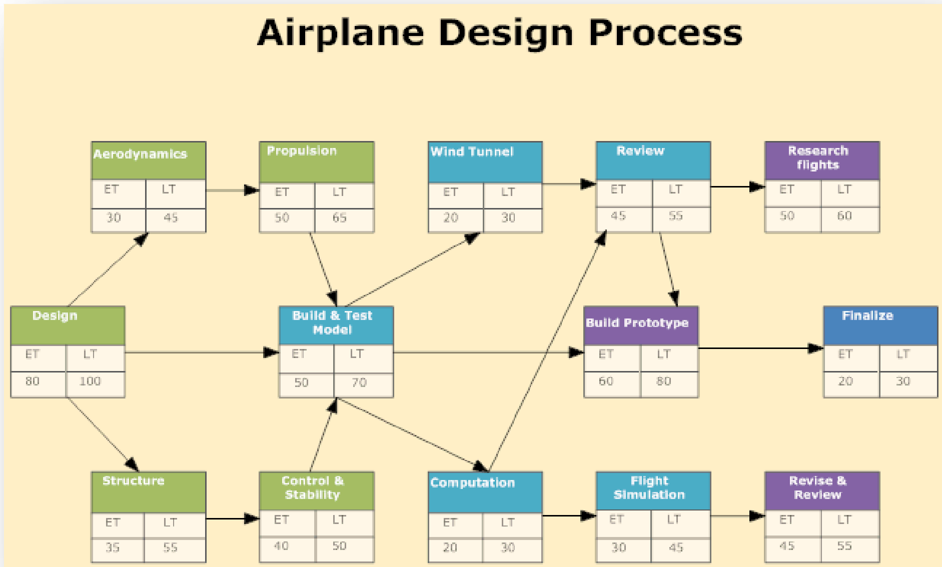


Fig. 6 An example of Pert model

There are 2 types of PERT: Traditional PERT approximation and Ginzburg's PERT approximation

1.3.4.9 TMS

To effectively manage time on a project, it is recommended to use a time management (TM) system. A TM system is a designed combination of processes, tools, and techniques that allow the project manager and team to identify, analyze, sequence, and estimate the duration for all project activities.

A TM system usually includes a resource calendar, resource assignment matrix (RAM), and a network diagram to help identify the overall duration and critical path activities of the project [George Angel. 2010].

Managing the schedule involves getting focused on the importance of supply-and-demand concepts. It is about making sure we have everything we need - resources, people, tools, materials - to get the job rolling and keep it rolling to bring the project in on time [Angel, George. 2010].

Time management, as a key knowledge area, includes six processes that center around effectively managing the project's work packages and activities (the actual work of the project). The ultimate goal is to bring the project to successful completion on time according to the approved delivery date. One thing that makes time management difficult is that time is constantly moving, and like the many dynamics we face on our projects, the milestones and target dates can be moving targets.

Here are the six processes of the Project Time Management knowledge area:

1. Define activities (Planning process group) - This involves identifying specific activities needed to produce the approved project deliverables.
2. Sequence activities (Planning process group) - This involves identifying and documenting relationships (for example, predecessor and successor) between the project activities.
3. Estimate activity resources (Planning process group) - This involves estimating the type and quantities of material, people, equipment, or supplies needed to perform each activity.
4. Estimate activity durations (Planning process group) - This involves approximating the number of work periods needed to complete individual activities with the estimated number of resources available.
5. Develop schedule (Planning process group) - This involves analyzing the activity sequences, durations, resource requirements, and schedule constraints to create the project schedule.
6. Control schedule (Monitoring and Controlling process group) - This is the process of monitoring the progress of the project to update status and manage changes to the schedule baseline [George Angel. 2010].

1.3.4.10 Agile/Scrum Project Management

Traditional projects development has followed a classic "waterfall" method, as in software projects where initiatives were carefully analyzed, designed, documented, coded, tested, and ultimately delivered to the customer – sometimes years after inception. By then, it was not

uncommon for business' needs to have changed and for the resulting system to fall short of customers expectations.

In response to this limitation, "Agile" methodologies – which are light on formality - began to emerge in the 1990's [Anne Loeser. 2006]. Scrum, which is one of several Agile approaches, was first developed and presented in 1995. Although Scrum was intended for management of software development projects, it can be used as a general project management approach.

The primary objective of Agile development is to deliver value early in the Project Lifecycle based upon customer and market demands. The ability to deliver value early and the ability to adapt to change, is considered to be a major contributor in making Agile Development one of the more rapidly growing trends in technology.

Below is a list of Scrum characteristics in contrast with traditional models:

- a dynamic Product Backlog of prioritized work to be done;
- a Release Plan to deliver larger initiatives across multiple Sprints, with the highest priority first;
- a Sprint Planning Meeting in which backlog items are selected for iterations(s) called Sprints (which are usually 30 days in length);
- a living Sprint Backlog or Sprint Plan of tasks to be done within the Sprint;
- a brief daily Sprint Meeting or Scrum, at which each team member's progress is disclosed, upcoming work is described and committed to, and impediments are raised;
- a Demo at which Product Owners (and occasionally developers) demonstrate accomplishments during the Sprint;
- a Retrospective, at which team members reflect about the past sprint and make recommendations about future improvements or changes.

Scrum is facilitated by a Scrum Master whose primary responsibility is to remove obstacles hindering the team from delivering the Sprint goal. The Scrum Master is not the leader of the team because the Sprint team is self-organizing. Instead, the Scrum Master acts as a

facilitator for issues resolution and communication, rather than as a manager controlling the team. The Scrum Master notes and removes obstacles, safeguards the Scrum process, facilitates collaboration, while the Project Manager is held ultimately accountable for traditional projects, the entire Sprint team - including the Scrum Master - shares responsibility for consummating the Sprint's objectives.

Scrum diverges from the approach to Project Management exemplified in the Project Management Body of Knowledge (PMBOK) - which has as its goal quality through application of a series of prescribed processes, documentation, and controls overseen by the Project Manager. In contrast, the Agile movement espouses that people and their interactions with each other are the key to creating value.

1.3.4.11 Critical Chain Project Management

According to PMBOK Critical chain method is a schedule network analysis technique that modifies the project schedule to account for limited resources. It mixes deterministic and probabilistic approaches to schedule network analysis.

CCPM takes advantage of the best practices of:

- PMBOK: Planning and control processes.
- TOC (Theory of Constraints): Remove bottleneck to resolve constraints.
- Lean: Eliminate waste.
- Six Sigma: Reduce Variations.

CCPM helps to overcome following phenomenon.

- Parkinson's Law: Work expands to fill the available time.
- Student Syndrome: People start to work in full fledge only when deadline is near.
- Murphy's Law: What can go wrong will go wrong.
- Bad Multi Tasking: Bad multitasking can delay start of the successor tasks.

CCPM is based on:

- Resource constrained situations.
- Optimum use of Buffer (amount of time added to any task to prevent slippage of schedule)
    - Project Buffers (PB): Amount of buffer time at the end of the project.
    - Feeding Buffers (FB): Amount of buffer time at the end of a sequence of tasks.
    - Resource Buffers (RB): It is an alert that is used to indicate that resource is needed to perform a task. This alert can be set few days before a resource is actually needed.

CHAPTER 2
ARTIFICIAL INTELLIGENCE IN PROJECT MANAGEMENT

Artificial Intelligence is a field of study that seeks in explaining and emulating intelligent behavior in terms of computational process. Its use in project management works by perceiving an environment and maximizing the chance of making a process successful [Abdomerovic, Muhamed. 2010].

We will speak first about using Artificial intelligence in economics and finance, then we will discuss its use in project management.

2.1. Artificial Intelligence in Economics

There are many possibilities for its use, and in the following paragraphs we will list some of them.

In Economic Forecasting

Usually, the mechanism of economic forecasting relies on econometric models with some limitations. Innovations in Artificial Intelligence hold out several new possibilities in this regard; these innovations can be used for correcting existing defects, thereby improving the performance of established models, or they can be used to bring about new approaches in modeling and in economic theory itself.

Over the past years, the development of numeric computing capacity established a strong presence of econometric models in the forecasting and diagnosing of macro-economic events. For all their efforts in improving technologies, designers of more powerful systems, along with users of them, have had to concede the limitations inherent in their approach. The problem lies in the difficulty of interpreting the results rendered by the models. These econometric models are simulation models, and they represent an economic structure in a state of equilibrium and show how this state changes over time [Paliés, Odile, and Mayer, Jacques. 2004].

Advances in Artificial Intelligence brought new approaches to the processing of information: modeling cognitive processes. Here the expertise and knowledge of both economist and econometrician are represented in order to produce an economic forecast by simulating their behavior [Paliés, Odile, and Mayer, Jacques. 2004].

2.2. Artificial Intelligence in Project Management

During the last few years, project management went through some serious development and a lot of software solutions were developed for it, which in most cases were oriented to preparation of project schedules, for example Microsoft Project, and this was fairly good for a long period of time.

Recently, a necessity for new software solutions appeared in order to answer the challenges of this fast developing area of knowledge. Some of the main problems at the moment are the following:

- Analyzing business process - analyzing and optimization of the business processes in the company;
- Adjusting of resource and project schedule; in larger projects the challenge of adjusting these two schedules always exists;
- Managing multiple schedules: often in the course of work with projects we need to work simultaneously with more than one schedule; also in the project schedule we often have tasks behind time or emerging new tasks; it would be very useful in these cases if the project management system offers us solutions for optimizing and reorganizing the time schedules or prepares a few solutions among which we could choose the most optimal;
- Using Smart wizards - recently increasingly valued are solutions which with the help of smart questions could prepare the main part of the project time schedule, thus leaving the Project Manager with doing only some final adjustments;
- Next generation of ERP systems: ERP systems have already been on the market for 20 years and have gone through a major development. They are at the moment irreplaceable tools in arrangement and analyzing the business processes in a company. Yet, nowadays,

this is not sufficient; there is need that the systems take and offer decisions on their own, something that needs artificial intelligence [Vasilev, Ivaylo. 2008].

Project management is an artistic skill of getting things done using qualitative and quantitative methodologies. There are many quantitative approaches currently used in project management in order to access project, project status and to gain knowledge from historical data. Qualitative implementation of Artificial Intelligence techniques in project management are used to yield effective results. Project management is a systematic approach of planning, organizing, and managing resources in order to achieve project goals and objectives.

In [Gorakavi, Pavan Kumar. 2010], there is an implementation of Predicate calculus – an Artificial Intelligence knowledge representation concept in building Project Management advisor. This intelligent system plays a vital role in project planning.

Artificial Intelligence is a study of computer science that perceives an environment and maximizes the chance of making a process successful. The two major fundamentals of Artificial Intelligence are knowledge representation and search. *Knowledge representation* is a methodology of addressing the problem by describing in a formal language. *Search* is a problem solving technique that systematically explores a space of problem states [Gorakavi, Pavan Kumar. 2010].

Predicate Calculus is a formal language which describes properties and relationship among objects in public domain that requires qualitative reasoning rather than arithmetic calculation for their solution. Predicate calculus contains truth symbols, constant symbols, variable symbols, functional symbols, predicates and atomic sentences.

Resource planning plays a vital role in project planning. Under estimation or over estimation of a task results in yielding a bitter result. This problem was studied in [Gorakavi, Pavan Kumar. 2010] and was approached using a predicate calculus expression and build a reasoning system from the expression. Planning system is illustrated the following figure:

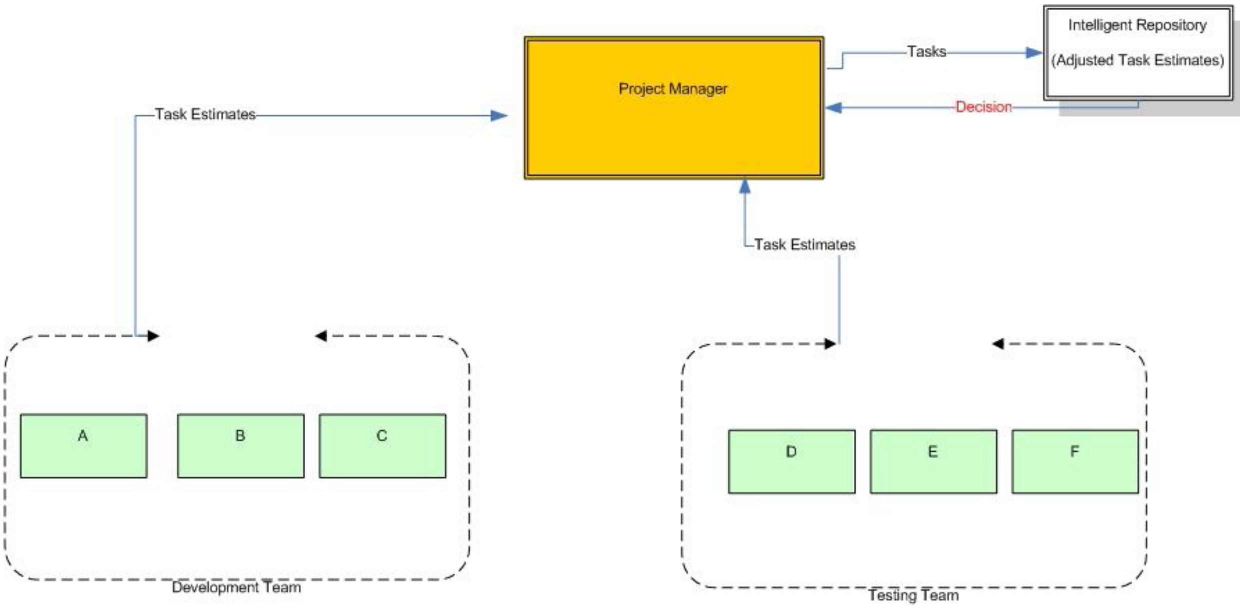


Fig. 7 Planning based on reasoning system

Historical repository stores the estimated task work hours against actual time to perform the tasks. This helps in calculating the confidence of a given estimate by a developer/team. Estimates are adjusted based on historical information found in repository and the error tolerance for a given task. Error tolerance includes external and internal dependencies.

2.2.1. AI in Project Planning

There is now some focus on the development of new software that shows how computers can support decision-makers. Such software will have the capability to engage in and lead routine tasks, and to assist when the unexpected happens. To focus the research on real problems and to ensure the software meets requirements such as privacy, security, and trust.

This lead to research on how Artificial Intelligence learning techniques (Abstraction, Search, Use of knowledge), and other different AI algorithms can be used to build a knowledge center for an Agile project based on historical statistics. This intelligent knowledge center can be useful in project planning, task estimations, and risk analysis [Abdomerovic, Muhamed. 2010].

2.2.2. Business process modeling

One of these problems in project management is modeling of business processes in a company. In this case, AI can be used with the help of examples like the use of generalized networks [Atanassov, K., Hadjiski, M. 2008] in defining the main process groups in the project [Vasilev, Ivaylo. 2008].

The idea of presenting the business processes in a dimensional generalized network [Atanassov, K., Hadjiski, M. 2008.] presents a lot of freedom and opportunities [Vasilev, Ivaylo. 2008]. Thus, easy and clear visualization of all business processes in an organization would vastly improve the possibilities of their improvement and evaluation.

Some of the benefits of this approach are:

- very easy adding and removing of processes and documents which allows easy testing of different variants of the business processes in the company;
- positioning internal hierarchy at each level. In the standard model of Project Management, such a hierarchy is usually put only between the processes, but often such a connection exists between the documents too – which could be presented very easily in this model;
- easy preparation of work procedures; once prepared the business processes could easily be transformed to work procedures which would greatly contribute to the ISO standardization of the company;
- easy analysis; one of the main tasks of every Project Manager is to observe, analyze and improve the business processes in a company, and with the help of this method of presentation, this would be much easier [Vasilev, Ivaylo. 2008].

In [Vasilev, Ivaylo. 2008] we presented an innovative method of analysis of the main business processes in the Project Management. Main characteristic of this method is that it uses the apparatus of generalized networks [Atanassov, K., Hadjiski, M. 2008] combined with comfortable dimensional situation of the basic components of Project Management.

2.2.3. Data Mining

There are many quantitative approaches currently used in project management in order to access project, project status and to gain knowledge from historical data. Data mining is the process of extracting patterns from data. Data mining, or knowledge discovery, is the computer-assisted process of digging through and analyzing enormous sets of data and then extracting the meaning of the data [Gorakavi, Pavan, Kumar. 2010].

Implementation of data mining techniques in project management yields effective results. This is discussed in [Gorakavi, Pavan, Kumar. 2010], and it studies data mining classification techniques and how it can be used to build a knowledge center for an Agile project based on historical statistics. This data mining application can be used in project planning, task estimations, and risk analysis.

Data mining is also widely used for estimation. The purpose of project data mining is the generalization of project management experience for successful implementation of projects during their planning and execution.

The quantitative nature of the project data implies that the known methods of the theory of experiments can be applied for their processing. But the traditional methods of the theory of experiment have limited applications in the case of project data, since these data are not the results of planned experiments. In fact, these data are the results of a random collection of information about the projects.

In turn, this means that from the standpoint of the theory of experiments, these projects as an experiment are carried out in a variety of conditions. Consequently, the results of such experiments are impossible and inappropriate compared with each other.

Under these conditions, the successful processing of project data is mainly related to how well these projects can be divided into groups, using different project similarity criteria. Such a partitioning of project data into groups has only one goal. This goal is to convert projects as

experiments, performed in different conditions, into groups of experiments carried out in almost identical conditions. This enables the application of the classical theory of experiments, in particular, regression analysis techniques, to project data.

In [Barseghyan, Pavel. 2010] a new methodology of project data mining is discussed, through which data is presented in the form of the families of curves, reflecting the fundamental relationships between the parameters of projects. This enables to improve the accuracy of estimates and predictions of project parameters.

2.2.4. Cloud Computing and Project Management

Cloud Computing can be best described as a highly automated, readily scalable, on-demand computing platform of virtually unlimited processing, storage and ubiquitous connectivity, always available to carry out a task of any size and charged based on usage [Asava, Raj, and Mzee, Hussein. 2010].

In this regard, project management is the application of knowledge, skills, tools and techniques to activities that meet project requirements and is accomplished through the use of the time tested PMI's five phase approach: initiation, planning, execution, controlling, and close-out. Collectively, these phases are known as the project life cycle. Project managers use the life cycle to deliver projects while balancing the constraints of scope, schedule, quality and resources, along with customer expectations and risk management [Asava, Raj, and Mzee, Hussein. 2010].

With the advent of cloud computing, there is an added need and urgency for a formalized project management methodology across organizations. From lost time to inconsistency, the lack of process & tools for managing projects will mean poor performance and an inability to harvest the true benefits of the cloud. The key to successful project management in the cloud era is to maintain the standardized project management framework that embeds best practices into how one manages projects, inside and outside the cloud [Asava, Raj, and Mzee, Hussein. 2010].

It is expected that cloud computing will:

- Usher in large complex projects, which in the past were not feasible due to limited compute, storage or network capacities within an organization;
- Enable the parsing of multiple transactions in a highly distributed environment made up of multiple providers, to be processed in tandem and subsequently aggregated;
- Provide real-time collaboration between globally dispersed teams;
- Allow rapid staging, set-up and take-down of a variety of compute environments as needed to test/validate an application;
- Require real-time project management software with a rich set of web-based tools [Asava, Raj, and Mzee, Hussein. 2010]

2.2.5. Artificial Neural Networks (ANN)

A critical issue in project management is the accurate estimation of size, effort, resources, cost, and time spent in the development process. Underestimates may lead to time pressures that may compromise the full functional development and the testing process. Likewise, overestimates can result in noncompetitive budgets. In [Fabiana, Iris, Demísio, José, and Sant'Anna, Nilson. 2008], artificial neural network and stepwise regression based predictive models are investigated, aiming at offering alternative methods for those who do not believe in estimation models. The results presented in that paper compare the performance of both methods and indicate that these techniques are competitive with the APF, SLIM, and COCOMO methods.

Another study [Iranmanesh, Seyed, Hossein, and Zarezadeh, Mansoureh. 2008] presents an application of Artificial Neural Network (ANN) to forecast actual cost of a project based on the earned value management system (EVMS). For this purpose, some projects were randomly selected based on the standard data set, with the necessary progress data such as actual cost, actual percent complete, baseline cost and percent complete for five periods of the projects. Then, an ANN with five inputs and five outputs and one hidden layer is trained to produce forecasted actual costs. The comparison between real and forecasted data show better performance based on the Mean Absolute Percentage Error (MAPE) criterion. That approach could be applicable to

better forecasting the project cost and result in decreasing the risk of project cost overrun, and therefore it is beneficial for planning preventive actions.

ANN is also used for predicting a project performance; successful project delivery depends on many factors. Various approaches have been advanced to facilitate tender award decisions. Essentially, this type of decision involves the prediction of a performance based on information available at the tender stage. A neural network based prediction model was developed and presented in [Cheunga, Sai, On, and Wonga, Peter, Shek, Pui, Ada, Fungb, S.Y., and Coffeyb, W.V. 2005]. It was found that the networks for the prediction of performance scores for Works gave the highest hit rate.

2.2.6. Fuzzy Logic

Managers of business organizations often use decision-making models to reduce the potential for poor decisions. In situations where decision variables have fluctuating values, Bayesian models based on probability theory have been used to accommodate the inherent uncertainty and ambiguity. The problem with using Bayesian probabilities in decision-making situations is that the probabilities of a future event are not always precisely known. Particularly, in the case of new projects, no statistical information is available upon which to base the development or assumption of a probability distribution. Even where statistical information is available, the circumstances in the future may be such as to make the development of probabilities difficult.

Whenever it is necessary to estimate probabilities in the absence of information about relative frequencies, the calculus of fuzzy set theory offers a useful alternative to the traditional approach. In [Shipley, Margaret, F., De Korvin, Andre, and Omer, Khurshehd. 1996] a new methodology is rooted in the general concepts of fuzzy logic theory with specific emphasis on belief functions and extension principles, and fuzzy probability distributions.

There is continuing interest by academics and practitioners alike in measuring and coping with project schedule uncertainty. Fuzzy logic has been proposed as an alternate approach to

probability theory for quantifying uncertain, related to activity duration. However, the fuzzy logic approach is not widely understood, and generally accepted computational approaches are not available. In [Matthew, J. 2002] the differences between the probabilistic and fuzzy approaches and the advantages of the latter are described; a straightforward approach for applying fuzzy logic is also illustrated.

Specific applications of fuzzy logic in project management are relatively few in comparison to other application areas. The criteria of project cost, project time, and project quality may be considered as project management internal measures of efficiency. The objective of a research done in [Dweiria, F.T., and Kablan, M.M. 2006] is to present an approach that employs fuzzy decision making (FDM) to combine these three measures into one measure, namely the project management internal efficiency (PMIE), which should represent an overall estimate of how well the project was managed and executed. The proposed approach for the evaluation of PMIE is illustrated on a case study. A fuzzy decision making system is designed and implemented using the MATLAB software for the evaluation of the PMIE. The methodology and procedure proposed in that research may be easily implemented by project management organizations. The evaluation of PMIE can serve for project managers and for project organizations as an indicator for the level of achievement of the project management internal objectives. PMIE may help in the evaluation of the performance of project teams [Dweiria, F.T., and Kablan, M.M. 2006].

2.2.7. Genetic Algorithms

GAs are robust general-purpose search programs based on the mechanism of natural selection and natural genetics [Karova, Milena, Petkova, Julka, and Smarkov, Vassil. 2008]. They are stochastical methods that can be used to solve a very broad class of optimization problems. They are known to solve problems in a heuristic way under consideration of the problem’s environment. The surrounding conditions of business processes, especially knowledge based business processes, can be seen as the problem’s environment with respect to the application of GAs. Therefore, it is useful to apply Genetic Algorithms to improve and manage business processes as well as knowledge based business processes.

The scheduling of tasks and the allocation of resource in medium to large-scale development projects is an extremely hard problem and is one of the principal challenges of project management due to its sheer complexity. As projects evolve, any solutions, either optimal or near optimal, must be continuously scrutinized in order to adjust to changing conditions. Brute force exhaustive or branch-and-bound search methods cannot cope with the complexity inherent in finding satisfactory solutions to assist project managers. Most existing project management (PM) techniques, commercial PM tools, and research prototypes fall short in their computational capabilities and only provide passive project tracking and reporting aids. Project managers must make all major decisions based on their individual insights and experience, must build the project database to record such decisions and represent them as project nets, then use the tools to track progress, perform simple consistency checks, analyze the project net for critical paths, etc., and produce reports in various formats such as Gantt or Pert charts.

In [Chang, Carl, K., Christensen, Mark, J. and Zhang3, Tao. 2001] a research was made to develop a new technique based on genetic algorithms (GA) that automatically determines, using a programmable goal function, a near-optimal allocation of resources and resulting schedule that satisfies a given task structure and resource pool. It was assumed that the estimated effort for each task is known a priori and can be obtained from any known estimation method such as COCOMO. Based on the results of these algorithms, the manager will be able to assign tasks to staff in an optimal manner and predict the corresponding future status of the project, including an extensive analysis on the time-and-cost variations in the solution space. The new GA algorithms can operate on much more complex scheduling networks involving multiple projects. It can also deal with more realistic programmatic and organizational assumptions. The results of the GA algorithm were evaluated using exhaustive search for five test cases. In these tests, the GA showed strong scalability and simplicity. Its orthogonal genetic form and modularized heuristic functions are well suited for complex conditional optimization problems, of which project management is a typical example [Chang, Carl, K., Christensen, Mark, J. and Zhang3, Tao. 2001].

[Karova, Milena, Petkova, Julka, and Smarkov, Vassil. 2008] presents implementation of Genetic Algorithms in Project Planning and Project Scheduling Problem. There are strict requirements to timeframes and budget cost for projects. Project management causes the problem of efficient resource assignment, activity, time constraints and relationships between activities. Genetic Algorithm uses random presentation of chromosome and two types of crossover: uniform and arithmetic. The approach is tested on a set of randomly generated problems.

2. 3. Problem Specification

Duration of the human actions has a random character and nonlinear dependency on the degree of difficulty of these actions and the productivity of people. Both the random nature of the actions of people and their nonlinearity have a direct connection to the analysis of the scheduling risks of human activities. The best mean for studying these risks are the distribution functions of the duration of human actions. From the adequacy of these distribution functions depends largely the correctness of the provisional estimates of risk, and in general, the meaningfulness of the whole methodology of the scheduling risk analysis [Barseghyan, Pavel. 2010].

For project planning, the role of the project manager is very critical in the development of project plans. Early involvement of the project manager in the project planning process can improve the project plan and save the project time and money [Abdomerovic, Muhamed. 2010].

One of the major disciplines of project management is project planning since it is:

- A scientific tool for integrating project scope, time and cost.
- An analytical coordination tool for project work.
- The knowledge-based and provable approach to get things done as required [Abdomerovic, Muhamed. 2010]

Therefore, it is obvious that the project plan can only be as good as the degree to which the project manager understands the project planning process and how much he or she contributes to the development of the project plan [Abdomerovic, Muhamed. 2010].

In the next chapter we will propose our approach that addresses the problem of time estimation in project time management.

CHAPTER 3

TIME ESTIMATION USING FUZZY LOGIC

Project management is an artistic skill of getting things done using qualitative and quantitative methodologies. There are many quantitative approaches currently used as we saw in project management in order to access project, project status and to gain knowledge from historical data.

In approaching his role in the development of a project plan, the project manager must first assess:

- project work breakdown structure and durations for upper-level elements of the structure;
- the length of project time;
- the probability that project will be completed without delay.

Although there are different approaches to project planning, all available planning processes are intended to facilitate project management functioning. However, a lack of understanding of those processes and/ or weak managerial support for project planning makes all this effort a real problem.

The key reasons that inhibit the project manager from giving his full support to the project planning team are insufficient procedures for project planning, diversified project planning experience, and project team members' lack of project management knowledge. (These are factors that can affect accurate time estimation as we will discuss later, and they can be considered comparison variables when needed).

The involvement of the project manager in project plan development may vary significantly from project to project. If a project is considered a big one, then the project manager should employ an experienced system analyst/ project planner that understands the project and is capable of leading the planning effort.

The project planning specialist should ask for adequate inputs from functional managers and prospective contractors and then compile an integrated project plan.

There is one area of logic that is used in artificial intelligence, and it is called Fuzzy Logic. In this chapter we will explain this fuzzy logic and how I used it to estimate project time based on similarities with other projects. I will start by explaining the fuzzy logic basics and with each concept I will explain how it is used for time estimation. The software that I developed will learn by interacting with and being advised by its users, will handle a broad range of interrelated decision-making tasks that have been resistant to automation in the past. This methodology can reduce the issue of over or under estimations of tasks duration and drive the project with an accurate planning.

The main idea behind this methodology is to use historical project statistics for better project planning. The most successful projects are almost always characterized by having a well-considered plan developed by an outstanding and committed team, not just by the project manager, project planner, or some independent consultant. However, the project manager's past experience may be significant for project plan development and project success. We are trying to help the project manager making better estimation based on how other projects are performing. And for this our approach will rely on fuzzy logic rules instead of only on the PM estimation.

3.1 Fuzzy Logic

Fuzzy logic is basically a logic with multiple values, which allows values between the conventional evaluations of the precise logic 1 and 0. It also includes operations for 'and', 'or', 'not' and 'if-then'. Hence, fuzzy logic extends conventional Boolean logic to handle the concept of the partial truth – the values falling between "totally true" and "totally false". These values are dealt with using degree of membership of an element to a set. The degree of membership can take any real value in the interval [0, 1]. Fuzzy logic makes it possible to imitate the behavior of human logic, which tends to work with "fuzzy" concepts of truth.

3.1.1 Elements of Fuzzy Sets

In classical or crisp sets, an element in the universe has a well defined membership or non membership to a given set. Membership to a crisp set F can be defined through a membership function defined for every element *x* of the universe as:

$$\mu_F(x) = \begin{cases} 1 & x \in F \\ 0 & x \notin F \end{cases}$$

For an element in a universe with fuzzy sets, the membership transition can be gradual. Therefore, the membership function can take any value between 0 and 1. This transition among various degrees of membership can be thought of as conforming to the fact that the boundaries of the fuzzy sets are vague and ambiguous. The membership of an element is measured by a function that attempts to describe vagueness and ambiguity. In fuzzy logic, linguistic variables take on linguistic values which are words (linguistic terms) with associated degrees of membership in the set. Thus, instead of a variable "numberOfYearsExperience" assuming a numerical value of 2 years, it is treated as a linguistic variable that may assume, for example, linguistic values of “small” with a degree of membership of 0.92, "normal" with a degree of 0.6, or "very experienced" with a degree of 0.2

Each linguistic term is associated with a fuzzy set, each of which has a defined membership function (MF). Formally, a fuzzy set is defined as a set of pairs where each element in the universe U has a degree of membership associated with it:

$F = \{(x, \mu F(x)) \mid x \in U, \mu F(x) \in [0, 1]\}$

The value $\mu F(x)$ is the degree of membership of object x to the fuzzy set F where $\mu F(x) = 0$ means that x does not belong at all to the set, while $\mu F(x) = 1$ means that the element is totally within the set.

As an example, let us consider the number of lines of code in a software project with the concepts of small and big. The membership functions in the classical logic for this example are given by

$$\mu_{Big}(x) = \begin{cases} 1 & \text{if } x \geq \text{10,000 lines of code} \\ 0 & \text{if } x < \text{10,000 lines of code} \end{cases}$$

$$\mu_{Small}(x) = 1 - \mu_{Big}(x) \begin{cases} 1 & \text{if } x \leq \text{10,000 lines of code} \\ 0 & \text{if } x > \text{10,000 lines of code} \end{cases}$$

The 10,000 lines of code boundary in this example is arbitrary. Independently of this boundary value, classical logic cannot interpret intermediate values. On the other hand, fuzzy logic solves the crisp problem with membership functions.

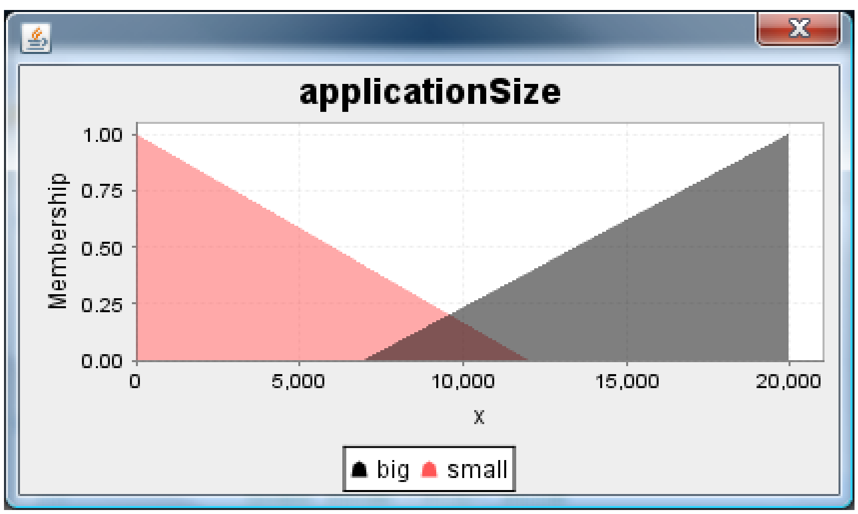


Fig. 8 An example of fuzzy set

In the above figure, we see a representation of a fuzzy set using these membership functions.

In this set, a 10,000 lines of code project belongs 18% to small and 24% to big. Further refinements, such as introducing an intermediate "average" size, can be made.

In any case, the crisp nature is always present in classical sets, while fuzzy sets allow gradual membership to better adapt to our subjective criteria. The membership function describes the degree of membership of the different elements of the fuzzy set in the universe. The selection of the form of membership function is subjective and depends on the context. However, for practical reasons, triangular, trapezoidal and bell shape functions are the most commonly used.

A list of possible membership functions is shown in the next figure:

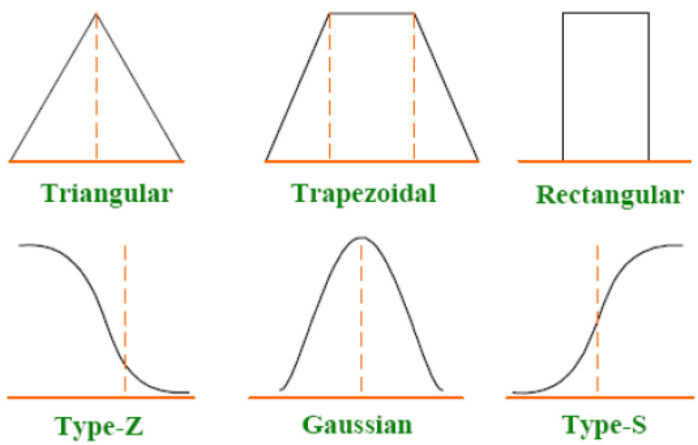


Fig. 10 Different membership functions

3.1.2 Fuzzy Logic

Lotfi Zadeh from UC-Berkeley introduced Fuzzy Logic in the 60's as a means to model uncertainty in natural language. Fuzzy logic extends conventional Boolean logic to handle the concept of the partial truth – with values really between "totally true" and "totally false" truths, making it easier to imitate the behavior of the human reasoning.

The extension of crisp logic to fuzzy logic is made by replacing the bivalent membership functions of crisp logic with fuzzy membership functions. Fuzzy values are assigned to evaluate the truth of propositions, and operations with these values are applied to evaluate composite propositions.

The meaning of a fuzzy proposition such as “x is A” is defined by the membership function that represents fuzzy set A. It is necessary to define the interpretation for the linguistic connectives “and”, “or” and for the operator “not”. The logical connectives of negation, disjunction, conjunction, and implication are also defined for fuzzy logic.

Modus Ponens is a rule of inference pertaining to the *if/ then* operator. Modus Ponens states that if the antecedent of a conditional is true, then the consequent must also be true.

Modus Tollens also is a rule of inference pertaining to the *if /then* operator. Modus Tollens states that if the consequent of a conditional is false, then the antecedent must also be false.

Fuzzy inference refers to computational procedures used for evaluating fuzzy linguistic descriptions. There are two important inference procedures: Generalized Modus Ponens (GMP) and Generalized Modus Tollens (GMT).

3.1.3 Fuzzy Logic Systems

A FLS receives a crisp input and may deliver either a fuzzy set or a crisp value. The basic FLS contains four components: a rule set, a fuzzifier, an inference engine, and a defuzzifier. Rules may be provided by experts or can be extracted from numerical data.

In either case, the engineering rules are expressed as a collection of IF-THEN statements. These statements are related to fuzzy sets associated with linguistic variables.

The fuzzifier maps the input crisp numbers into the fuzzy sets to obtain degrees of membership. It is needed in order to activate rules, which are in terms of the linguistic variables. The inference engine of the FLS maps the antecedent fuzzy (IF part) sets into consequent fuzzy sets (THEN part). This engine handles the way in which the rules are combined.

In most applications, crisp numbers must be obtained at the output of an FLS, and in our case we have two numbers: the estimated time and the estimated error. Defuzzifier maps output fuzzy sets into a crisp number, which becomes the output of the FLS.

3.1.4. Fuzzification

The first step in fuzzy logic processing involves a domain transformation called fuzzification. Crisp inputs are transformed into fuzzy inputs. To transform crisp input into fuzzy input, membership functions must first be defined for each input.

Once membership functions are defined, fuzzification takes a real time input value, such as years of experience, and compares it with the stored membership function information to produce fuzzy input values.

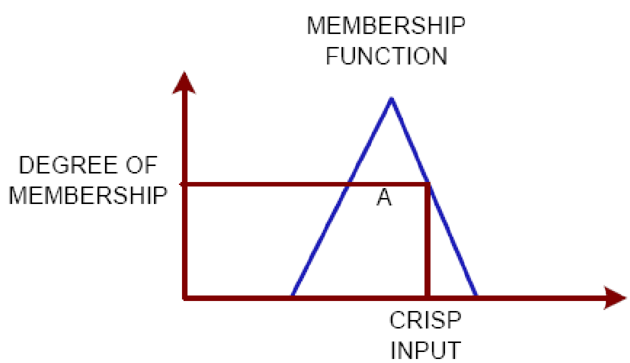


Fig. 11 Fuzzification of crisp values

The first step in fuzzification is to assign fuzzy labels in the Universe of discourse of each of the crisp inputs.

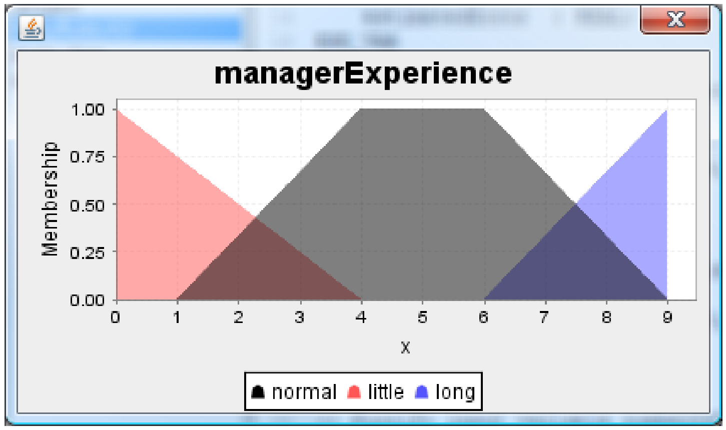


Fig. 12 Labels for a fuzzy set

Each crisp input into a fuzzy system can have multiple labels assigned to it. In our case, the years of experience of project manager have three labels: little experience, normal experience, long experience. In general, the greater the number of labels assigned to describe an input variable, the higher the resolution of the resultant fuzzy control system, resulting in a smoother response.

Next, membership functions are defined to give numerical meaning to each label. Each membership function identifies the range of inputs values that corresponds to a label.

An input membership function is created by specifying a number, that is, the degree of membership for each possible input value for a given label. The y-axis {μ} values refer to the degree to which the crisp input value (years) applies to each of the membership function labels (little, normal, long). Input values can belong to more than one fuzzy set.

3.1.5 Fuzzy Inference

Fuzzy logic based systems use RULES to represent the relationship between observations and actions. These rules consist of a precondition (IF-part) and a consequence (THEN-part). The precondition can consist of multiple conditions linked together with AND or OR conjunctions. Conditions may be negated with a NOT.

The computation of fuzzy rules is called Fuzzy Inference. Fuzzy rule inference consists of two steps:

- Inferencing - which determines the fuzzy subset of each output variable for each rule. Usually only MIN or PRODUCT are used as inference rules.
- Composition - which combines the fuzzy subsets for each output variable into a single fuzzy subset. Usually MAX or SUM are used.

Fuzzy rules trades the precise representation of the values that variables must assume with much more intuitive fuzzy representations. In binary logic, the consequent is either true or false. In fuzzy logic, partial truths are allowed so the consequent is as partially true as the antecedent allows it to be.

In general, a rule by itself does not do much. What is needed are a set of rules that can play off one another. The fuzzy inference methodology allows "fair" competition between these rules to produce sophisticated answers using seemingly simple premises.

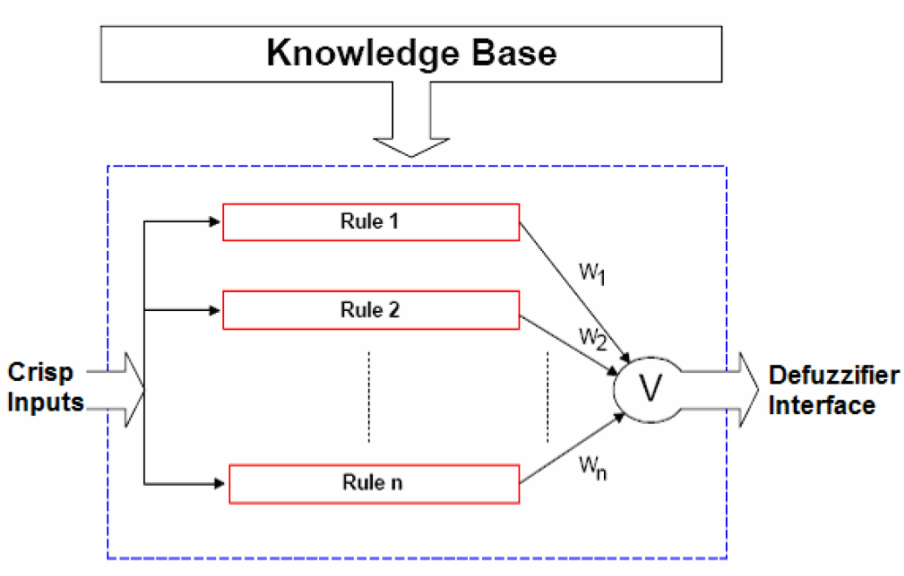


Fig. 13 Fuzzy inference procedure

3.1.6 Defuzzification

This stage is used to convert the fuzzy output set to a crisp number. Two of the more common techniques are: the Centroid and Maximum methods. In the Centroid method, the crisp value of the output variable is computed by finding the value of the center of gravity of the membership function. In the Maximum method, the crisp value of the output variable is the maximum truth-value (membership weight) of the fuzzy subset.

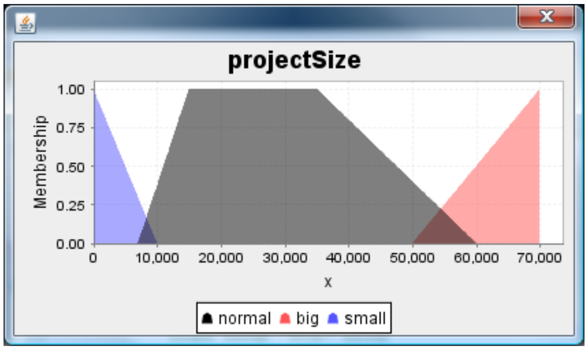
In our approach we used the centroid method. We explained each step of this approach, and now we put the steps all together and will illustrate the complete procedure of how our new system works.

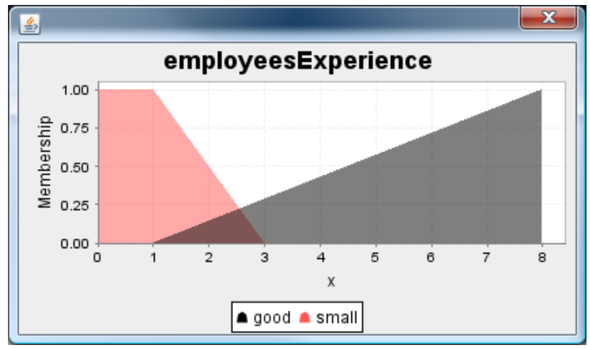
As we stated earlier, we want to estimate a time (it can be the project time, or an activity duration), and we also want to estimate errors of estimated time compared to what the project manager estimates. For example, given a project and the necessary inputs, it can either give an estimated time for the activity, or it can give a percentage number and it will be added to the first estimated time by the project manager.

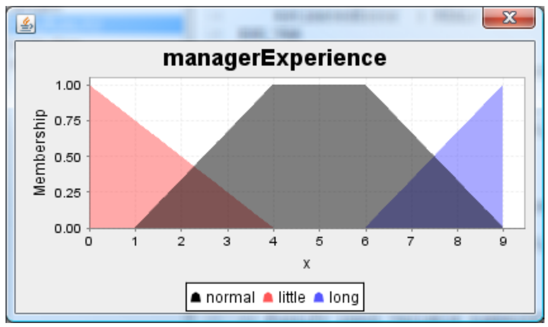
Therefore, we must identify how a project can resemble another project, and for this, we must first define the comparison variables on which we will compare the similarity between a project and another. And since comparison with more projects can give better accuracy than comparing with just one project, then the comparison will be based on factors affecting the time estimation.

The factors that can affect time estimations can be divided in two domains: project specific factors (in a software project these can be databases used, programming language used, etc.) and general factors (such as the experience of a project manager, experience of employees, project size, etc.).

The next step after we chose the comparison factors is to convert them to fuzzy sets. For example, if we want to base the estimation on these factors: manager experience, employees experience, project size, then their fuzzy can be like:

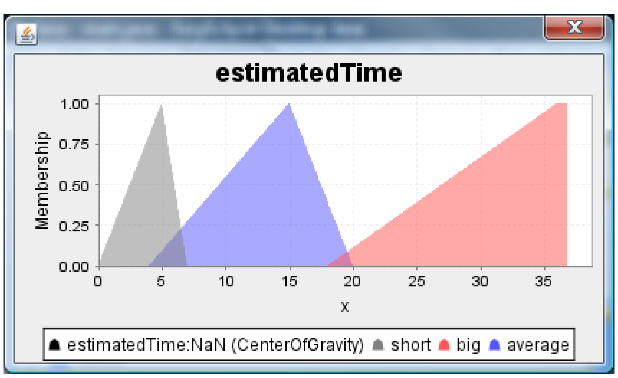


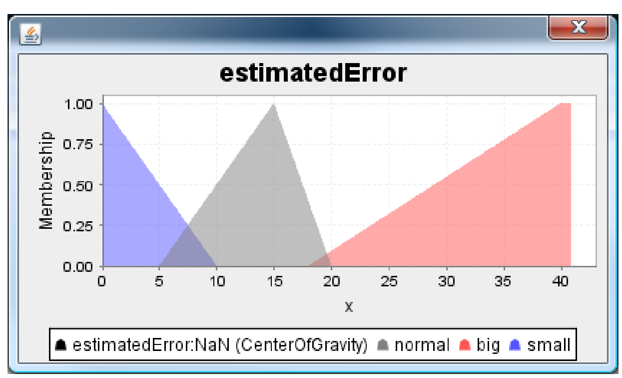




The values we choose to define the labels for a fuzzy set can be either collected from surveys, or they can be collected automatically if the proper tool was installed on the computer on which each project has its data.

Then we must choose the output variables which will be the output we are interested in. In our case, we have two variables: estimatedTime and estimatedError.





After we chose the labels for each fuzzy set, we must write the inference rules. An example of such rules that are added to the rule base is shown below:

RULE 1 : IF managerExperience IS little AND employeesExperience IS small
THEN estimatedError IS big;

RULE 2 : IF projectSize IS big
THEN estimatedError IS big ;

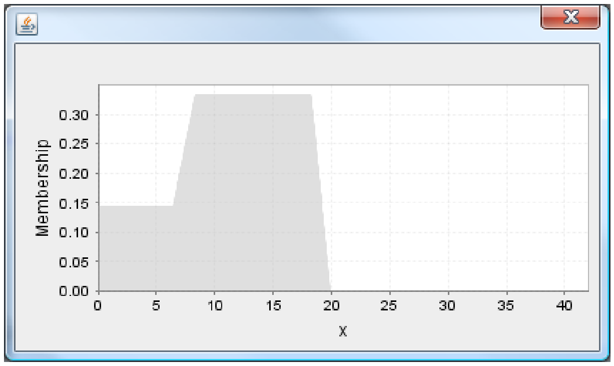
RULE 3 : IF projectSize IS big AND managerExperience IS long AND
employeesExperience IS good
THEN estimatedError IS small ;

RULE 4 : IF projectSize IS normal
THEN estimatedTime IS average ;

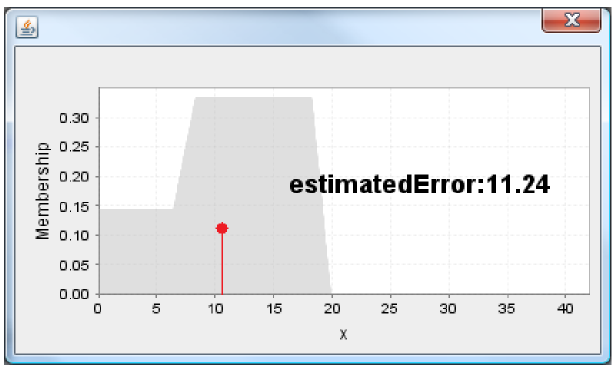
and of course we add all the rules that we discovered using a survey or automatically as we stated earlier.

Then we enter the crisp values of the input variables if, for example, we have these input values: 7 years of experience for the project manager, 65,000 lines of code for the project size and 2 years average experience for the employees.

After that, we must fuzzify our input values to get membership values according to the related fuzzy set, then we start the inference process and get a fuzzy shape as an output, as shown below:



The final step is the defuzzifying step. We used the center of gravity method and gave us the following result:



We got a number that represents the estimated error for the project related to the initial estimation. In our example, the project will last 11.24% longer than what was initially estimated.

Conclusion

Using Artificial Intelligence in project management helps managers/ organizations/ Scrum masters to take more relevant decisions. The credibility of the system basically relies on how well the historical information is persisted and how relevantly it is used in project planning. In this thesis I presented a new mathematical model that can be used to determine the resemblance degree between a project and other projects depending on the comparison variables (which they should be factors affecting the time estimation ability). This new model uses Fuzzy Logic as its basis. The goal lies in the concept of time management, specifically time estimation. The more facts and resources are used early on for development of the project plan, the more likely it will be that the project is completed on time and within the budget.

The result of our work is the development of a new way for dealing with the area of time estimations in project development, where the probability and impact of a project delay can determine the attractiveness of the project. Yet, our methodology can also be generalized and therefore has the capability of being used in other domains of project management in different kinds of industries.

References

1. Abdomerovic, Muhamed. (2010) *Project Planning Manager's Contribution*, PM World Today, April.
2. Addison, T. and S. Vallabh. (2002) *Controlling Software Project Risks - An Empirical Study of Methods used by Experienced Project Manager*. SAICSIT
3. Aggarwal, Naveen, Nupur Prakash, and Sanjeev Sofat. (2008) *Content Management System Effort Estimation, Model based on Object Point Analysis*, International Journal of Computer Science and Engineering
4. Angel, George. (2010) *PMP Certification. A Beginner's Guide*. Mc GrawHill
5. Asava, Raj, and Mzee, Hussein. (2010) *Cloud Computing meets Project Management*, PM World Today, June, 12(6).
6. Atanassov, K., Hadjiski, M. (2008) *Generalized Nets as Tools for Modelling of Intelligent Systems*, 4th International IEEE Conference "Intelligent Systems".
7. Barseghyan, Pavel. (2010) *Probabilistic Analysis of the Duration of Human Actions (Analytical Scheduling Risk Analysis)*, PM World Today, April. 12(4)
8. Barseghyan, Pavel. (2010) *Project Data Mining and Project Estimation Top-down Methodology with TRANSCALE Tool*, PM World Today, June, 12(6).
9. Bergeron, F. and J.-Y. St-Arnaud. (1992) *Estimation of Information Systems Development Efforts: A Pilot Study*, Information and Management, 22: 239 - 254.
10. Cantor, Murray. (1998). *Object-Oriented Project Management with UML*. Wiley Computer Publishing
11. Chang, Carl, K., Christensen, Mark, J. and Zhang3, Tao. (2001) *Genetic Algorithms for Project Management*, Journal Annals of Software Engineering, Springer Netherlands, 11(1): 107-139.
12. Cheunga, Sai, On, and Wonga, Peter, Shek, Pui, Ada, Fungb, S.Y., and Coffeyb, W.V. (2005) *Predicting project performance through neural networks*, Department of Building and Construction, City University of Hong Kong, available online, 19 September.
13. Dweiria, F.T., and Kablan, M.M. (2006) *Using fuzzy decision making for the evaluation of the project management internal efficiency*, Decision Support Systems archive, November, 42(2): 712 – 726.
14. Fabiana, Iris, Demísio, José, and Sant'Anna, Nilson. (2008) *An investigation of artificial neural networks based prediction systems in software project management*, Journal of Systems and Software archive, March, 81(3).
15. Flynn, T.A., P.E. (2007) *Integration of the Project Management Life Cycle (PMLC) and the Systems Development Life Cycle (SDLC) in Accelerated Project Efforts, Adapting Project Management Best Practices to Unreasonable Requests*. Advanced Management Services, Inc. (AMS)
16. Gorakavi, Pavan, Kumar. (2010) *Application of Data mining techniques in Agile Project Management - Emerging trend in Project Management*, PM World Today, June, 12 (6).

17. Gorakavi, Pavan Kumar. (2010) *Usage of Predicate Calculus to represent and reason about Resource Planning, Artificial Intelligence Techniques in Project Planning*, PM World Today, April. 1 (4).
18. Heemstra, F. J. (1992) *Software cost estimation*, Information and Software Technology. 34(10): 627 - 639.
19. Heemstra, F.J., and R.J. Kusters. (1989) *Controlling software development costs: a field study*. International Conference on Organisation and Information Systems
20. Iranmanesh, Seyed, Hossein, and Zarezadeh, Mansoureh. (2008) *Application of Artificial Neural Network to Forecast Actual Cost of a Project to Improve Earned Value Management System*, World Academy of Science, Engineering and Technology.
21. Karova, Milena, Petkova, Julka, and Smarkov, Vassil. (2008) *A Genetic Algorithm for Project Planning Problem*, International Scientific Conference Computer Science
22. Kumanan, S., and O.V. Krishnaiah Chetty. (2005) *Estimating product development cycle time using Petri nets*, Springer-Verlag London Limited.
23. Loeser, Anne. (2006) *Project Management and Scrum – A Side by Side Comparison*
24. Matthew, J. (2002) *Project schedule uncertainty analysis using fuzzy logic*, Project Management Journal, December 1.
25. Moores, T.T. and J.S. Edwards. (1992) *Could Large UK Corporations and Computing Companies Use Software Cost Estimating Tools? - A Survey*, European Journal of Information Systems. 1(5): 311 - 319.
26. Paliés, Odile, and Mayer, Jacques. (2004) *Economics reasoning by econometrics or artificial intelligence*, Theory and Decision, Springer Netherlands, Monday, November 08.
27. Phan, D. (1990) *Information Systems Project Management: An Integrated Resource Planning Perspective Model*. Department of Management and Information Systems
28. Phan, D., D. Vogel, and Nunamaker. (1988) *The Search for Perfect Project Management*, in Computerworld: 95 - 100.
29. Shankar, N. Ravi, K. Surya Narayana Rao, and V. Sireesha. (2010) *Estimating the Mean and Variance of Activity Duration in PERT*. International Mathematical Forum, 5(18): 861 – 868
30. Shipley, Margaret, F., De Korvin, Andre, and Omer, Khurshehd. (1996) *A Fuzzy Logic Approach for Determining Expected Values: A Project Management Application*, University of Houston-Downtown, Texas.
31. Suri, P. K., Bharat Bhushan, and Ashish Jolly. (2009) *Time Estimation for Project Management Life Cycle: A Simulation Approach*, IJCSNS International Journal of Computer Science and Network Security, 9(5)
32. Vasilev, Ivaylo. (2008) *Project Management and Artificial Intelligence*, International Scientific Conference Computer Science, Sofia, Bulgaria.
33. Wydenbach, G. and J. Paynter. (1995) *Software Project Estimation: A Survey of Practices in New Zealand*, New Zealand Journal of Computing. 6(1B): 317 - 327.