IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e-ISSN: 2278-1684,p-ISSN: 2320-334X, Volume 17, Issue 3 Ser. III (May - June 2020), PP 38-47
www.iosrjournals.org

Design of Engineering Project Planning Software: A Case Study

[1]*Paul Tamaragaibi ELIJAH, [2]Emmanuel Ekiome ELIJAH and
[3]Ojong Elias OJONG

Department of Mechanical Engineering, Faculty of Engineering, Nigeria Maritime University, Okerenkoko, Delta State, Nigeria.

Abstract: *Development of an engineering software that is very suitable for project planning was done and tested with GAP international limited. The work involved a project planning tool that can maintain multiple projects and can be used by anyone (such as small independent firms, public municipalities, and government organizations) for their engineering projects. The research objective was to provide a suitable engineering project planning software that is cheap to afford, easy to use as well as being powerful enough to plan long-term projects and flexible enough to adapt to changes, constraints and clients needs. The critical path method (CPM) was employed in achieving it. This paper also looked at the use of powerful planning tools like Gantt charts and task lists to schedule engineering projects, assign tasks and make milestones to stay motivated in order to achieve meeting-up with project completion date. The developed software is used for determining critical paths, shortest project duration, and project completion date. When compared with some available planning software, it showed that the results on the critical paths were the same but the developed software from this work was more user friendly (much easier to use) thus, does not require high level of expertise to use. Having completed the analyses and given the importance of each criterion (robustness of software, ease of application, time, computational cost and reliability and accuracy of results) the paper therefore recommends that the developed engineering planning software is a better planning model for the case study in particular, and for engineering projects in general, thereby enhancing grassroots reduction of scarce resources and increase productivity in the industry.*

Keywords: *Design, Engineering, Software, Project, Planning, Critical path.*

Date of Submission: 06-06-2020 Date of Acceptance: 22-06-2020

I. Introduction

One of the innovative project planning software available to engineering firms is the Microsoft project (MPS) and it makes use of the Gantt chart (a horizontal bar chart developed as a construction control tool in 1917 by Henry L. Grantt, an American engineer and social scientist (Grant and Taylor 1917). Others includes; Primavera and the PERT chart (a project evaluation and review technique otherwise known as the program evaluation and review technique developed in 1958 by Booz Allen Hamilton) (Platie and Gayer, 1994). Both the Gantt charts and PERT chart have the simple versions created on graph paper and the more complex automated version created using project management applications such as Microsoft project or excel (Scacchi, 1984).

The most common mathematical programming approaches are linear programming (LP) models, of which a wide variety exists (Kolisch and Draxel, 1996). In most of these models, capacity is modeled as a fixed upper bound on the total amount of a resource that can be consumed in a time period (Hans and Royld, 2002). The classic algorithms also called exact algorithms are groups of algorithms that separates the environment into different sections. These are enhanced version of the preliminary motion planning algorithms bug, potential field (Mac et al., 2016). Mathematical programming and the geometry-based algorithm are widely used for this kind of research work (Ge and Cui, 2000; DeBoer, 1998). A vast amount of literature has been dedicated to project scheduling and includes: Demeulemeester and Herroelen, 1992; Ozdamar and Ulusoy, 1995; Kolisch and Drexel, 1996; Elijah and Etebu, 2019). To develop a Linear Programming model with both work in progress (WIP) and finished goods inventories that were modeled distinctly, under most reasonable assumptions on inventory holding costs (specifically, that inventory holding costs are non-decreasing in the amount of processing that has be completed on the project) (Mendes et al, 20009).

De Boer and Schutten (1999) propose several heuristic algorithms for RCCP problem in which they try to minimize the total number of hours of non-regular capacity that is used, that is $C_{kt} = 1$ ($\forall$ K, t) and conducted similar studies for the multi-project-scheduling situation, and their conclusion was that the shortest activity for shortest project (SASP) and maximum total work contents (MAXTWK) were found to be superior to the other scheduling rules; Some early pioneers include Sir James Watt, who built the first stem engine and Charles Babbage (from the United states of America) one of those who developed the modern day computers. Mohring,

(1984) utilized linear programming in his model formulation. He commented that while there is no difficulty in problem formulation, there are a large number of variables and constrains even for small-sized problems and as such, the model has very little practical application. He concluded that in order to be certain of achieving the 'best' solution, it might be necessary to try different heuristics and to select the best, which may give the optimal or near-optimal result.

Akpan and Chinzea, (2000); De Boer and Schutten (1999) proposed on problem (P) that one or more precedence relations are violated in intervals (r_i, d_i) and (r_j, d_j) overlap and the precedence relation between job J_i and job J_j is "repaired" by specifying a week T_{ij} ($r_i + p_i \le T_{ij} \le d_i - p_j + 1$) before which job J_i must be finished and before which job J_j cannot start; De Boer and Schutten (1999) suggest a number of heuristics based on problem (P), that is, based on problem (Ps) with set S such that $S_j = r_j$ and $C_j = d_j$ (Hans and Rroyld, 2002), and also consider the so-called resource loading problem.

The resource loading problem can be seen as an RCCP problem with simple precedence constraints, where the project network is a chain. They present a hybrid model of the resource-driven and the capacity-driven variants; they extended the precedence relations as in our RCCP problem description (Hans and Rroyld, 2002). Kolisch and Kelly (1995) were more interested in those heuristics that give the shortest project duration.

However, the above planning software requires a high level of training, and cannot be used by everyone. It is for this setback that the intent of this paper is to develop an engineering project planning software (a computer program with a user friendly interface) in engineering projects scheduling used for determining critical paths, shortest project duration, and project completion date in engineering projects. To achieve this objective, a simplified project capacity planning process that encourages the engineer in the use of computational approach to engineering project plan is employed (Maevis, 1980; De Waar, 1999; Davis, 1975).

II. Material and Methods

2.1Materials

Study Location

GAP international limited located in Delta State, Nigeria was used as the case study in this research work. It was contracted by an oil and gas company to design, construct and install a fire protection system along side with the piping facilities that carry its petroleum products at some point in the facility and at some other point within the same facility was specifically based on reconstruction of the existing terminal's fire protection system, changing the pipe from epoxy coated carbon steel to heavy wall carbon steel

Data Collection

Project planning in the case study was investigated through the civil engineering. Project executed are broken down into activities prior to execution.

Data for the project of reconstructing the terminal's fire protection system was collected from the department of civil engineering and project planning unit of GAP international limited and tabulated as shown in Table 1.

Table 1 Showing activities and resource data for the case study.

Activities	Nodes		Duration (weeks)	No. of shifts
	Starting	Ending		
Mobilization	1	2	1	1
Engineering Design	1	3	2	1
Scarification & draining pipelines	1	4	4	1
Setting out	2	5	8	1
Cracking of road surface & concrete walls	3	5	9	1
Laying of pipeline	5	7	7	1
Casting and laying of Kreb & slabs	3	6	3	1
Installation and coating of pipelines	4	6	6	1
Charging line with foam seal chemical & Testing	5	7	5	1
Clean up and Demobilization	7	8	10	1

2.2 Methods

Data Analysis

The project is broken into a number of rather large work packages, called job. The network diagram analysis is used to establish the critical path, which is the optimum solution.

The models for the earliest finish time, latest finish time, total float time and others are developed as shown below.

$C_j = \min (d_j, \min k_{1J}; - J_k\, r_k - 1)$ Equation no. (1)

To determine earliest finish time:

EFT=EST + Duration (2)

To determine late start time:

LST = LFT – Duration (3)

To determine total float time:

TF = LET – EST- Duration (4)

Determination of the critical path or activities

TF= LFT – EST – Duration = 0 (5)

Determination of the critical cost

Sum up the duration of the activities on the critical path. Time is money (resource) as such referred to as cost. Other important formulae or Equations are follows:

$$\text{(P): min } \sum_{t=1}^{T}\sum_{k=1}^{K} C_{kt}\ U_{kt}, \tag{6}$$

Subject to

$$\sum_{t=r_j}^{d_j} x_{jt} = 1 \qquad \forall j, \tag{7}$$

$$x_{jt} \le \frac{1}{p_j} \qquad \forall j,\ t, \tag{8}$$

$$U_{kt} \ge \sum_{j=1}^{n} q_{kj}\, x_{jt} - Q_{kt} \ \ \forall k,\ t, \tag{9}$$

$$x_{jt}, U_{kt} \ \ge\ 0 \qquad \forall j,\ k,\ t, \tag{10}$$

$$x_{jt} \le \frac{s_{jt}}{p_j} \qquad \forall j,\ t, \tag{11}$$

Where: parameter S_{jt} indicates whether processing of job J_j is allowed in week t:

$$s_{jt} = \begin{cases} 1 & if\ S_j \le\ t\ \le\ C_{j,} \\ 0 & otherwise \end{cases} \tag{12}$$

$$\sum_{\tau=r_j}^{t-1} \chi_{iT}\ \ \ge y_{jt} \quad \forall (i,j) \in \rho, \quad r_j \le t \le d_{i,} \tag{13}$$

$$\sum_{\tau=r_j}^{t} \chi_{jt}\ \ \le\ y_{jt} \qquad \forall j,t, \tag{14}$$

$$y_{jt}\ \ \in \{0,\ 1\}. \tag{15}$$

Design the Software

For the purpose of clarity, some of the codes used in designing the software are shown below:

```
Public Class Form3
    Dim act, desc As String
    Dim dur, start, en As Integer
    Dim ES, LS, EF, LF, TTF As Integer
    Private Sub btnLoadInput_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnLoadInput.Click
```

```
        act = Me.txtActivities.Text.ToString()
        desc = Me.txtDescription.Text.ToString.ToString()
        start = Val(Me.txtStartingPoint.Text.ToString())
        en = Val(Me.txtEndpoint.Text.ToString())
        dur = Val(Me.txtDuration.Text.ToString())
        Me.DataGridView1.Rows.Add(act, desc, dur, start, en)
        Me.txtActivities.Text = ""
        Me.txtDescription.Text = ""
        Me.txtDuration.Text = ""
        Me.txtEndpoint.Text = ""
        Me.txtStartingPoint.Text = ""
        Me.txtActivities.Focus()
    End Sub

    Private Sub btnCompute_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnCompute.Click
        Dim I, J, K, N, MIX As Integer
        N = Me.DataGridView1.Rows.Count - 1
        K = 0
        Do
            For I = 0 To N
                J = Me.DataGridView1.Rows(K).Cells(4).Value
                If Me.DataGridView1.Rows(I).Cells(3).Value = 1 Then
                    Me.DataGridView1.Rows(I).Cells(5).Value = 0
                    Me.DataGridView1.Rows(K).Cells(6).Value = Me.DataGridView1.Rows(K).Cells(5).Value +
Me.DataGridView1.Rows(K).Cells(2).Value
                ElseIf Me.DataGridView1.Rows(K).Cells(4).Value = Me.DataGridView1.Rows(I).Cells(3).Value
Then
                    Me.DataGridView1.Rows(I).Cells(5).Value = Me.DataGridView1.Rows(K).Cells(6).Value
                    Me.DataGridView1.Rows(I).Cells(6).Value = Me.DataGridView1.Rows(I).Cells(5).Value +
Me.DataGridView1.Rows(I).Cells(2).Value

                ElseIf I <> J And J = Me DataGridView1.Rows(I).Cells(4).Value Then
                    MIX = Math.Max(Me.DataGridView1.Rows(K).Cells(6).Value,
Me.DataGridView1.Rows(I).Cells(6).Value)
                    Me.DataGridView1.Rows(I).Cells(5).Value = MIX - Me.DataGridView1.Rows(I).Cells(2).Value
                    Me.DataGridView1.Rows(I).Cells(5).Value = Me.DataGridView1.Rows(I).Cells(6).Value -
Me.DataGridView1.Rows(I).Cells(2).Value
                    'Me.DataGridView1.Rows(I).Cells(6).Value = Me.DataGridView1.Rows(I).Cells(5).Value -
Me.DataGridView1.Rows(I).Cells(2).Value
                    'Return
                End If
            Next
            K = K + 1
        Loop Until K = N
    End Sub
End Class
```

2.3 Flow Chart

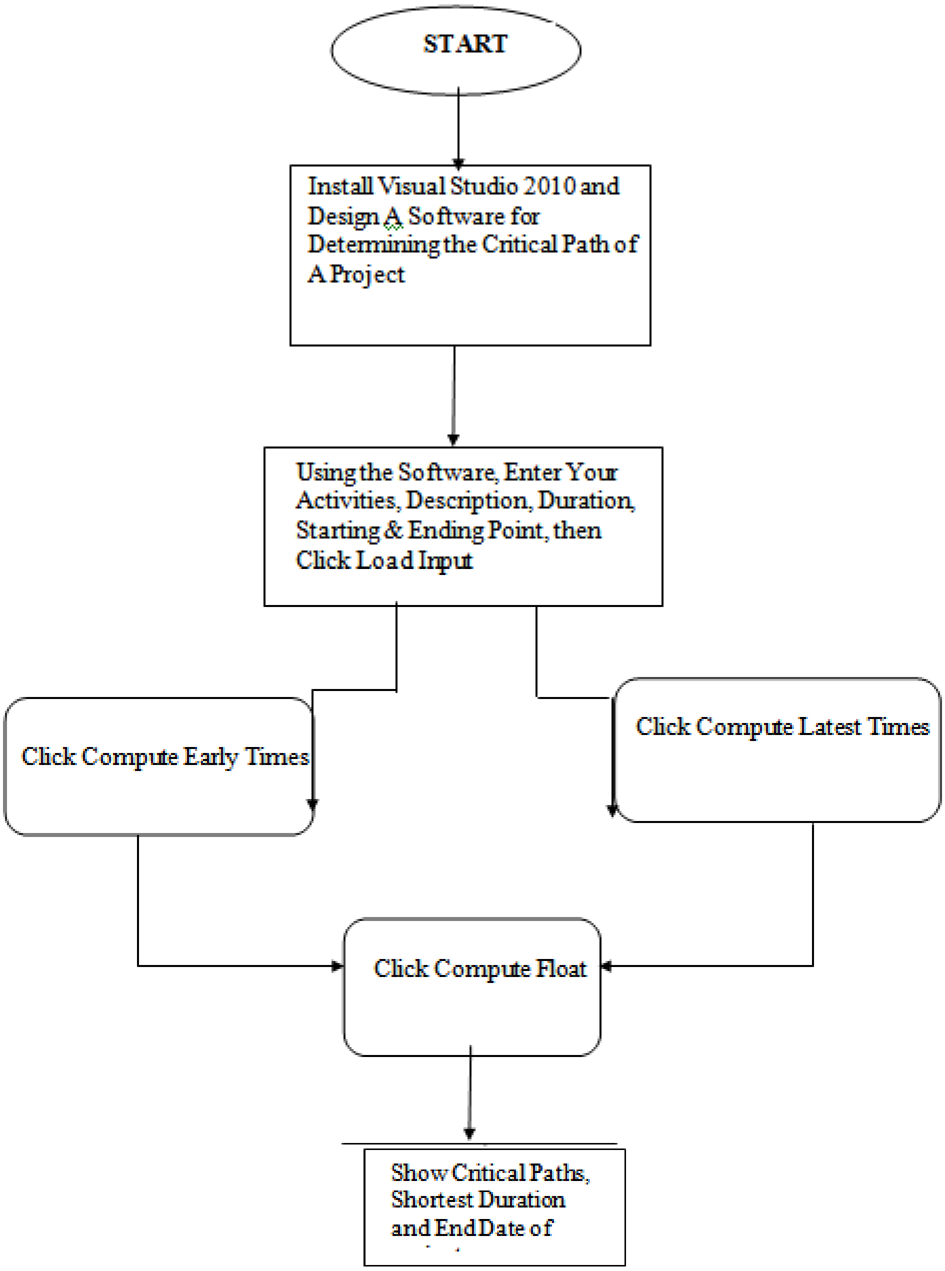

START
Install Visual Studio 2010 and Design A Software for Determining the Critical Path of A Project
Using the Software, Enter Your Activities, Description, Duration, Starting & Ending Point, then Click Load Input
Click Compute Early Times
Click Compute Latest Times
Click Compute Float
Show Critical Paths, Shortest Duration and End Date of

Figure 1: Flow chart used in designing the software.

2.4 Algorithm

An algorithm is a special case of an iterative method, which generally need not converge in a finite number of steps. Instead, an iterative method produces a sequence of iterates from which some subsequence converges to a solution. An algorithm terminates in a finite number of steps with a solution. In using algorithm to solve planning problems, three (3) basic ingredients are considered: objective function which is required to be either minimized or maximized**;** set of factors of variables which affects the valve of the objective function**;** and set of constraints that allow the variables to take on certain values but exclude others.

III. Results and Discussion

3.1 Results

The network diagram for the case study is shown below.

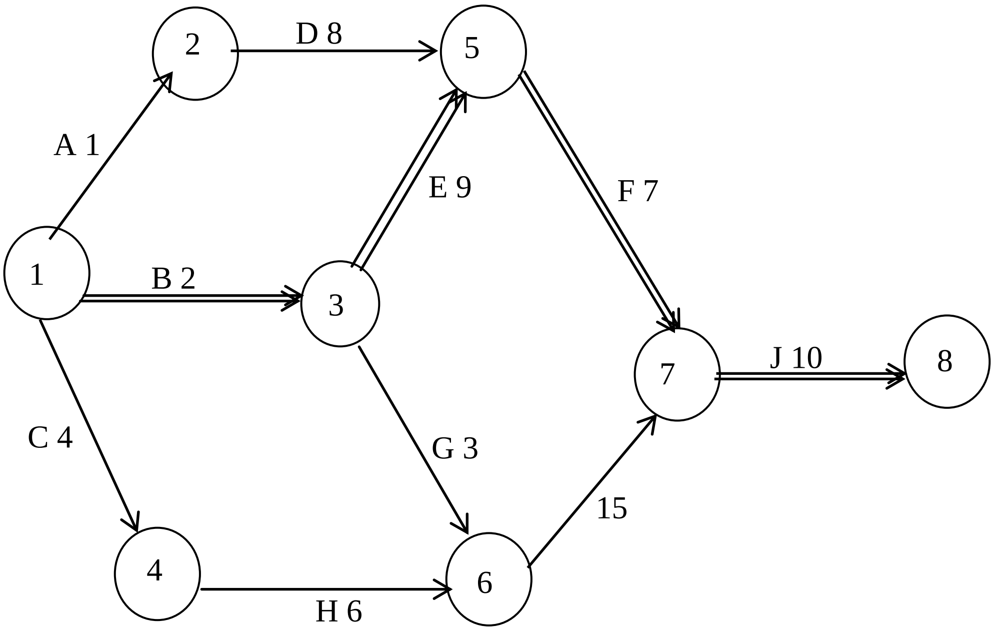
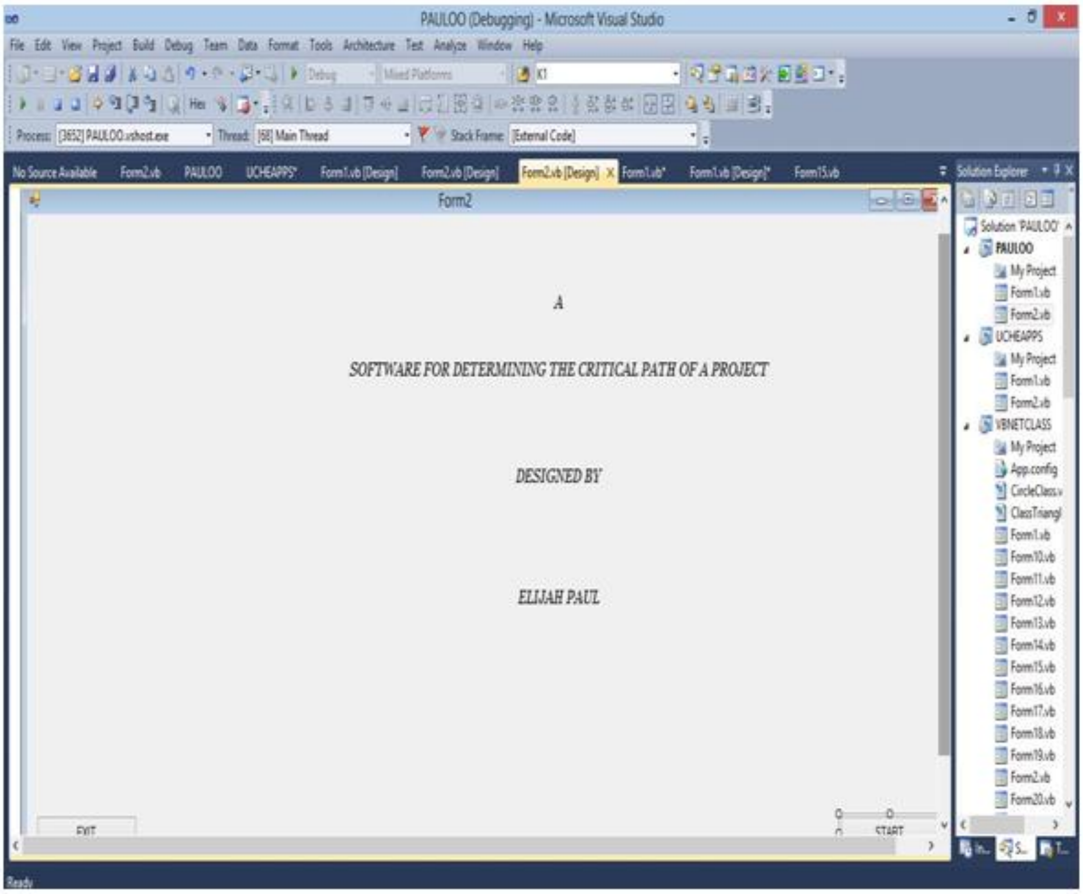


Figure 2: Network diagram for the case study project.

A snap-shot of the welcome page of the software is shown in figure 3.
At the bottom right hand corner is located the start command button, when clicked, the windows views pops up as shown in the figure (3, 4, & 5) below. The bottom left hand corner of the welcome page has the exit button, and when clicked, it exits the program.


PAULOO (Debugging) - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Format Tools Architecture Test Analyze Window Help
Process: [3652] PAULOO.vshost.exe
Thread: [68] Main Thread
Stack Frame: [External Code]
Form2
Solution Explorer
EXIT
START
Ready

A

SOFTWARE FOR DETERMINING THE CRITICAL PATH OF A PROJECT

DESIGNED BY

ELIJAH PAUL

Figure 3: A snap-shot of the software welcome page

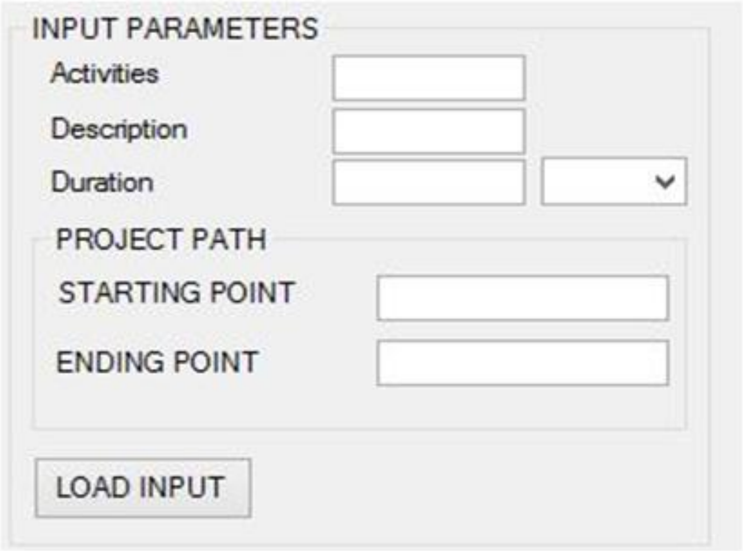
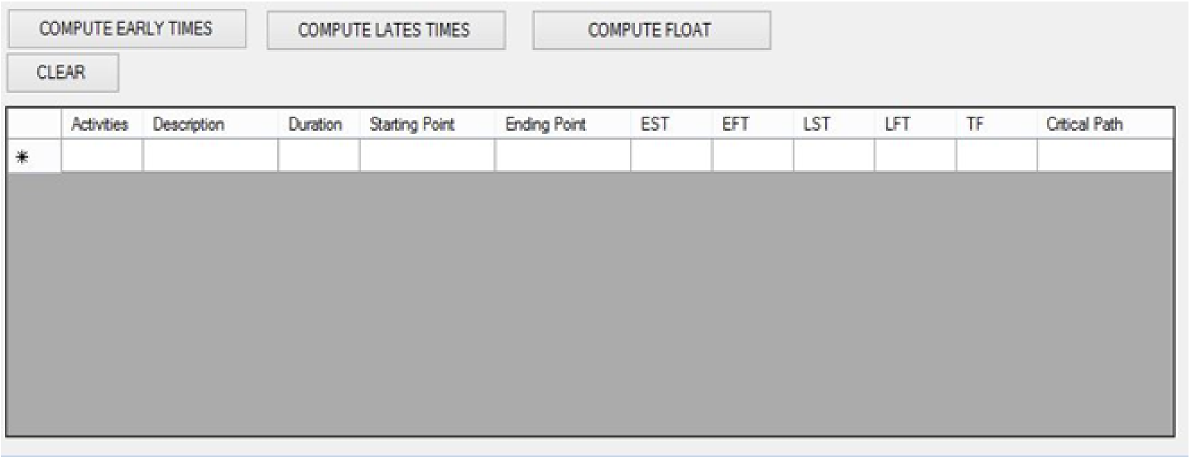
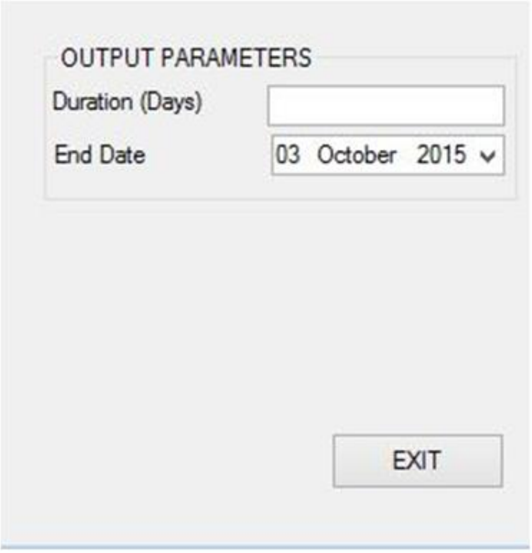


Figure 4: A display of the analysis window for input parameter



	Activities	Description	Duration	Starting Point	Ending Point	EST	EFT	LST	LFT	TF	Critical Path
*											

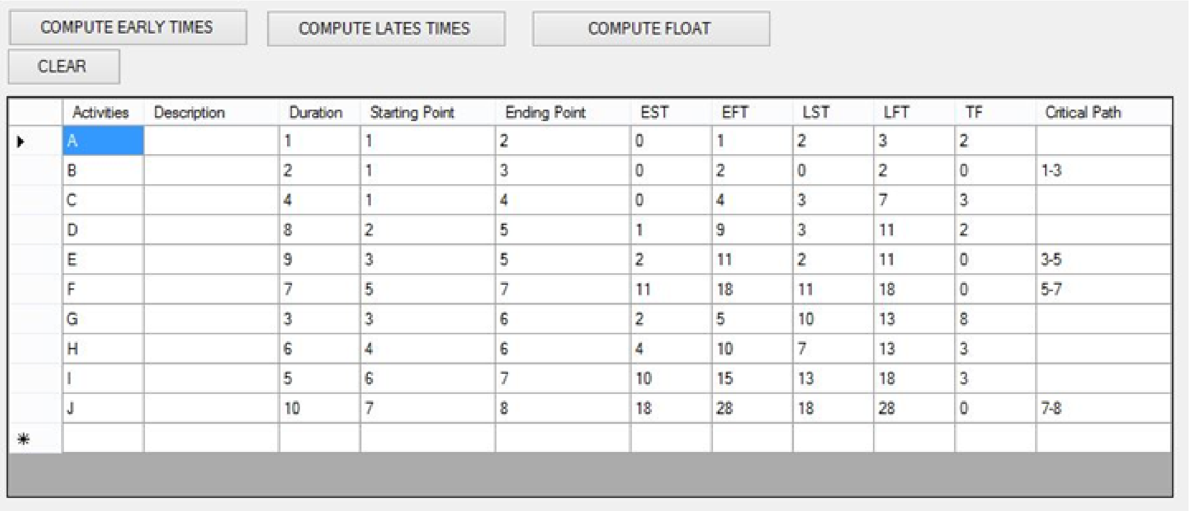
Figure 5: A display of the analysis window for the critical path before the input parameters are transferred into it.



End Date 03 October 2015

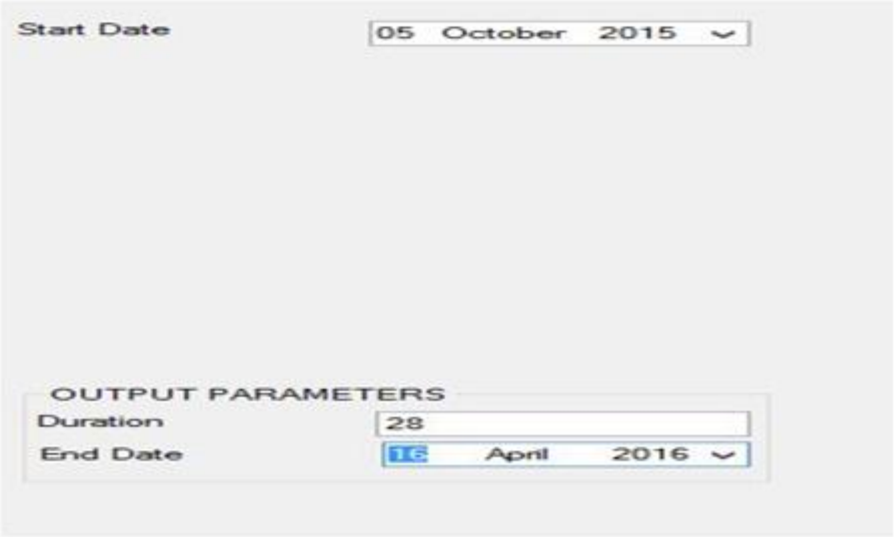
Figure 6: A display of the analysis window for output parameters.

Once the load input button is clicked, it transfers the input parameters to the designed table waiting for computation of the early times, latest times, and float (which determine critical activities and hence critical path). The table of data from the information (input parameters) is entered and computed as shown in Figure 7.



Activities	Description	Duration	Starting Point	Ending Point	EST	EFT	LST	LFT	TF	Critical Path
A		1	1	2	0	1	2	3	2	
B		2	1	3	0	2	0	2	0	1-3
C		4	1	4	0	4	3	7	3	
D		8	2	5	1	9	3	11	2	
E		9	3	5	2	11	2	11	0	3-5
F		7	5	7	11	18	11	18	0	5-7
G		3	3	6	2	5	10	13	8	
H		6	4	6	4	10	7	13	3	
I		5	6	7	10	15	13	18	3	
J		10	7	8	18	28	18	28	0	7-8

Figure 7: The display computed values of total float and critical path for different activities.



Start Date 05 October 2015

OUTPUT PARAMETERS

Duration 28

End Date 16 April 2016

Figure 8: The display computed values of Project Duration and End Date

Table 2: Computer output of schedule data and the critical path for the fire protection system renewal project

Activities	Duration (Weeks)	Starting point	Ending point	Earliest start time	Earliest finish time	Latest start time	Latest finish time	Total float	Critical path
A	1	1	2	0	1	2	3	2	
B	2	1	3	0	2	0	2	0	(1-3)
C	4	1	4	0	4	3	7	3	
D	8	2	5	1	9	3	11	2	
E	9	3	5	2	11	2	11	0	(3-5)
F	7	5	7	11	18	11	18	0	(5-7)
G	3	3	6	2	5	10	13	8	
H	6	4	6	4	10	7	13	3	
I	5	6	7	10	15	13	18	3	
J	10	7	8	18	28	18	28	0	(7-8)

3.2 Discussion

In the design, construction and installation of the fire protection system as well as the renewal project carried out by the case study, the first activity is the mobilization of all the necessary resource needed for the project. The resources include materials (such as pipes, new flanges, values, red oxide coating, and so on), machinery (such as trolleys, pumps, vice, and so on), and man (expatriate, quantity survey, fitters, and so on).

Based on Table 2, the critical path for the case study used in this research work is (1-3), (3-5), (5-7), (7-8), or (B-E), (E-F), (F-3).

The developed software is used to automatically computer early times, latest times, total float, and critical activities, requiring as input the activities, duration (their performance time requirement), and the precedence relationships established (i.e starting and ending nodes). The software's output are the duration for the whole project (which is the possible shortest completion time), and the end date of the project.

3.2.1 Study of Sensitivity Analysis

The project duration was 28 weeks using the developed software and it corresponded to the result obtained when solved manually. The calculation of total float which determine the critical paths, are based on the two extremes of the earliest and latest starting times. Actually, there are over five (5) different paths from start to finish through the network, but the shortest path is that of 1 – 3 – 5 – 7 –8. This route gives the best option to the completion of the fire protection system renewal project. The total cost of this path is 28weeks. For projects networks with multiple paths (such as this case study), one could enumerate all the alternative paths/routes to find the longest path, but it has no relevance and there is no benefit in doing so. This is because clients normally subscribe to the shortest delivery date as long as the quality of the completed project is not compromised.

3.2.2 Advantage of the Developed Software over Existing one used by the Case Study

The project planning unit of GAP international limited uses a specific software as a tool in managing their project. This software determines the critical path/activities of any project when the activities start nodes, end nodes, and duration are inputted. Figure 7 shows a display of the computed values the critical path for the project activities using the developed engineering project planning software. The result was the same with what was obtained using the existing planning software of the case study. However, using the existing planning software requires a high level of' expertise and therefore requires the training of personnel that needs it for planning and control/monitoring. For some oil and gas multinationals, training before now, requires sending personnel abroad and of course, it involves a lot of money (transportation, lodging, feeding, and cost of training). Nowadays, training is relatively cheaper due to competition (as some cost about $555 per personnel) as it is now done in Nigeria.

This cost of training personnel can be saved by employing the software developed from this research work. Aside other staff in the planning unit, the case study has five (5) personnel’s working directly with the existing planning software. So, if the existing planning software is replaced with the developed one in this work, it will help to save as much as (5 x $555) spent on training alone. Also, the cost of purchasing most of the licensed project planning software (like the ones used in this research) are very high when compared with the developed engineering project planning software.

IV. Conclusion

Development of an engineering software that is very suitable for project planning was done and tested with GAP international limited. Most of the commercial planning software available are incorporated with the different models which the mathematical principles and basic assumptions associated with them are not clearly known. Hence, they make the study of such phenomenon not to be precise and reliable. Therefore, the developed engineering planning software employed the critical path method (CPM) in its design. The work critically evaluated and compared the developed software with some available ones against a set of five broad criteria which include robustness, reliability and accuracy of results, time, ease of application and cost. Having completed the analyses, the findings therefore provide better insight into advantages of the developed engineering planning software.

The successful completion of any project is a function of appropriate firms to perform key project activities (critical activities), clarity of project assignment. Making provision for networks communication system within the project, making use of performance milestone and giving adequate financial incentives for all groups involved all leads to a successful completion of projects on time. If time is spent up front planning the project (in all ramifications) and subsequently working the plan through proper execution and control as can be achieved based on the performance of a planning software is it obvious that project are likely to be completed within budget, specification and on schedule. A project would therefore run accordingly if and only if these principles and techniques are applied as and at when due by the project manager through a team while making sure that the owners specification are always met.

Nomenclature

Symbol	**Meaning**	**Units**
EST	Earliest Start time	s
EFT	Earliest Finish time	s
LST	Earliest Start time	s
LFT	Latest Finish time	s
TF	Total Float	s
FF	Free Flout	s
Jn	Set of n-Jobs	-
Rk	Plans on k-resources	-
T	Time Horizon	s
Qkt	Capacity of Resources in week, t	tons
$X_{jk}t$	Fraction of job J_j performed on Resource, R_k in week, t	%
$^{I}/_{Pj}$	Maximum fraction done per week	%
C_{kt} U_{kt}	Cost of U_{kt} hours of non-regular capacity of resources, R_k in week, t	$
S	Set allow to work (ATW) window	-
A	Activities	-

References

[1]. Akpan, E.O.P. and Chinzea, E.F., Project management: Theory and practice, Nigeria (2000), Koby Press Ltd: Owerri

[2]. Davis, E.W., Project network summary measures constrained-resource scheduling, AIIE Transactions, Vol.7, Issue 2 (1975), pp.132-142.

[3]. De Boer, R. and Schutten, J. M., Multi-project rough-cut capacity planning in Flexible Automation and Intelligent Manufacturing, Ashayeri, J, Sullivan, W.G. and Ahmad, M.M. (eds.), (1999), pp. 631-644, Begell House, Wallingford: UK.

[4]. De Boer, R., Resource-constrained multi-project management, (1998), PhD thesis: University of Twente, Enschede, (Netherlands).

[5]. De Waar, A. J., Re engineering non flow large scale maintenance organizations (1999), PhD thesis: University of Twente, Enschede, (Netherlands).

[6]. Demeulemeester, E.L. and Herroelen, W.S., A branch-and-bound procedure for the multiple resource-constrained project scheduling problems, Management Science, Vol. 38 (1992), pp.1803- 1818.

[7]. Elijah, P. T. and Etebu, O. M. O., Project planning using heuristics approach: a case study of gap international limited, World Journal of Engineering Research and Technology (WJERT), Vol. 5, Issue 5 (2019), pp.154-163.

[8]. Gantt, H. L. and Taylor, F. W., The Gantt Chart, Management Institute (1917), AMA: New York

[9]. Ge, S.S. and Cui, Y.J., New potential functions for mobile robot path planning, IEEE Transactions on Robotics and Automation, Vol.16, Issue 5 (2000), pp. 615-620.

[10]. Hans, E. W. and Royld, F. P. T., Resource loading by branch-and-price techniques: models and algorithms, Technical report BETA-WP87, (2002). The Netherlands: University of Twente, Enschede.

[11]. Kolisch, R. and Drexel, A., Adaptive search for solving hard project scheduling problems, Naval Research Logistics, Vol. 43 (1996), pp.23-40.

[12]. Kolisch, R. and Kelly, J. E., Characterization and generation of a general class of resource-constrained project scheduling problems, Management Science, Vol.41 (1995), pp.1693-1703.

[13]. Mac, T., Copot, C., Twing, D.T. and Keyser, R. O., Heuristic approaches in robot path planning: A survey, 2016.

[14]. Maevis, A. C., Construction Cost Control by the Owners, ASCE Journal of the Construction Division, Vol. 106 (1980), pp. 435-446.

[15]. Mendes, J. J. M., Goncalves, J. F. and Kesende, M .G. C., A random key based genetic algorithm for the resource constrained project scheduling problem, Computers and Operations Research, Vol.36, Issue 1(2009), pp. 92-109.

[16]. Mohring, R. H., Minimizing costs of resource requirements in project networks subject to a fixed completion time, Operations Research, Vol.32 (1984), pp. 89 120.

[17]. Ozdamar, L. and Ulusoy, G., A survey on the resource-constrained project scheduling problem, HE Transactions, Vol. 27 (1995), pp.574-586.

[18]. Platje, A and Gayer, R., Project portfolio planning cycle: project based management for the multi-project challenge, International Journal of Project Management, Vol. 12, No. 2 (1994), pp. 100 – 106.

[19]. Scacchi, W., Software engineering: A social analysis study, IEEE Transactions on Software Engineering, Vol. 2, No. 1 (1984), pp.45-60.

[20]. Thukwo, D., Computer software and programming, Journal of Computer Science, Vol.19, Issue 3 (2005), pp. 1121-1136.

Paul Tamaragaibi ELIJAH, et. al. “Numerical Study of the Seismic Behavior of a Historic Stone MasonryTower.” *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, 17(3), 2020, pp. 38-47.