PROJECT SCHEDULING, TIME LINE CHART AND SOFTWARE QUALITY METRICS

Principles of project scheduling:

Compartmentalization: Project must be compartmentalized into a number of manageable activities, actions, and tasks; both the product and the process are decomposed.

Interdependency: The interdependency of each compartmentalized activity, action, or task must be determined. Some tasks occur in sequence while others can occur in parallel.

Time allocation: Each task to be scheduled must be allocated some number of work units. In addition each task must be assigned a start date and a completion date that is a function of their interdependencies.

Effort validation and defined responsibilities: Every task that is scheduled should be assigned a specific team member.

Defined outcomes and milestones: Every task scheduled should have a defined outcome. Every task or group of tasks should be associated with a project milestone.

Effort applied vs. Delivery time: The nonlinear relationship between effort applied and delivery time can be easily found in our project where effort increased rapidly when delivery time is reduced.

We tried hard to achieve 40-20-40 distribution of effort rule.

Software quality metrics:

We calculate the effort and time taken using the cocomo model with respect to system design. (The total number of lines in the code is 705. We remove 250 lines that are either comments or just blank lines)

We consider the model to be advanced model (as design is over).

$$E=a*(S^b)*F$$

Here a=3.6, b=1.20, F=1, S=0.455 (KDSI). So effort is 1.39.

$$T=a*(E^b)$$

Here a=2.5, b=0.32, E=1.39. So time taken is 2.77months when calculated using this method.

Timeline Chart:

All project tasks are listed in the far left column. The next few columns lists the following for each task: projected start date, projected stop date, projected duration, actual start date, actual stop date, actual duration and task inter-dependencies. To the far right are columns representing dates on a calendar. The length of the horizontal bar on the calendar indicates the duration of the tasks. When multiple bars occur at the same time interval on the calendar, it implies task concurrency. A diamond represents a milestone and has time duration zero.

Timeline Chart given below:

Task #	Task name	Duration (Days)	Actual start date	Actual end date	Predecessor	Feb-Mar	Mar-Apr	Apr-May	May
A	Inception	4	2/26	2/29	None	▬			
B	Elicitation and elaboration	8	3/1	3/8	A		▬		
C	Negotiation and specification	7	3/9	3/15	B		▬		
D	Planning, modelling and documentation	16	3/16	3/31	C		▬		
E	System design	13	4/1	4/13	D			▬ ◆	
F	Interface design	12	4/4	4/15	D			▬ ▬	
G	Back end design	11	4/21	5/1	E,F			▬	
H	Documentation	6	4/27	5/2	G			▬ ▬	
I	Testing and test report	2	5/1	5/2	G			▬ ▬	
J	Completion	1	5/2	5/2	H,I				▬

The corresponding task network is attached as separate file. We find that the task network provides a critical path of 2.2 months. Both cocomo model and the task network provide nearly equal answers.