Project Workflow Management[1]

The WBS and Preliminary Project Planning[2]

By Dan Epstein

Introduction

This is the second article in the series on Project Workflow Management. The first article describes an overall Project Planning Process for developing plans for executing and controlling all project groups of processes called frames and detailed processes within each frame. The described here process is one of twelve detailed processes within the high level Project Planning process.

This article is based on the book *Project Workflow Management: A Business Process Approach* by Dan Epstein and Rich Maltzman, published by J Ross Publishing in 2014. The book describes PM Workflow® framework, the step-by-step workflow guiding approach using project management methods, practical techniques, examples, tools, templates, checklists and tips, teaching readers the detailed and necessary knowledge required to manage project "hands-on" from scratch, instructing what to do, when to do and how to do it up to delivering the completed and tested product or service to your client.

The project workflow framework is the result of the author's research into the subject, having the following objectives:

1. Create a virtually error-free project management environment to ensure significant reduction of project costs

2. Reduce demands for highly qualified project managers using the step-by-step workflow guiding approach.

While PM Workflow® is the continuous multi-threaded process, where all PM processes are integrated together, this article will attempt to describe the Preliminary Project Planning as a stand-alone process that can be used independently outside of PM Workflow® framework. It will be difficult in this article not to venture into processes outside of the scope of this article, such as planning, quality, communications and other management processes, so they will be just mentioned. However, to get full benefit and the error free project management environment, the complete implementation of PM Workflow® is required.

---

[1] *This series of articles is based on the book Project Workflow Management: A Business Process Approach by Dan Epstein and Rich Maltzman, published by J Ross Publishing in 2014. The book describes the PM Workflow® framework, a step-by-step approach using project management methods, practical techniques, examples, tools, templates, checklists and tips, teaching readers how to manage a project "hands-on" from scratch, including what to do, when and how to do it up to delivering a completed and tested product or service to a client.*

[2] How to cite this paper: Epstein, D. (2018). The WBS and Preliminary Project Planning, *PM World Journal*, Volume VII, Issue VI - June.

In order to understand how PM Workflow® ensures this environment, I strongly recommend reading my article Project Workflow Framework – An Error Free Project Management Environment in the PMI affiliated projectmanagement.com (https://www.projectmanagement.com/articles/330037/Project-Workflow-Framework--An-Error-Free-Project-Management-Environment)

The article above provides the overview and explanation of how the project workflow framework works and achieves the established objectives.

For more information, please visit my website www.pm-workflow.com

Purpose

The purpose of the Work Breakdown Structure is to establish plans and methods for implementing and managing projects. The WBS in the context of this book is a tool for developing or updating the schedule data, such as milestones, deliverables, dependencies, risks, work products and resource requirements. Changes to the project scope any time throughout the project will bring the project flow back to this process. The WBS is a foundation upon which task estimates, task dependencies, resource allocation, risk management, quality management and project planning build into the project schedule.

WBS Activity Decomposition

The WBS is a deliverable-oriented multilevel decomposition of the entire project scope into sets of smaller, more manageable and controllable tasks. The WBS is, in effect, a hierarchical tree, which may be presented in several different forms:

- WBS Decomposition Diagram (hierarchical)
- Gantt Chart (adds element of time)
- Network Diagram (focused on sequencing and dependencies)

WBS decomposition rules are different from process decomposition rules, which were described in the Requirements Frame section of the book. Instead of decomposing high level business activities to elementary business activities during the process decomposition, WBS activities are decomposed down to the level of the smallest executable project elements with the identified deliverables, called a task. Deliverables do not have to be a client deliverable; they may be an intermediate deliverables, which are required to create a client deliverable. There are two important rules of WBS decompositions:

- No task should be longer than 40 hours (since any larger would indicate that further decomposition is possible)
- No task should be shorter than 16 hours (since this could cause an excessive number of small tasks to track).

The 40 hour rule allows the project manager to have better control over the project. Experienced project managers know, when they ask team members about status of not yet completed tasks, the answer is often 80% or 90% completion. Unfortunately, the remaining 10% takes three times

longer than the previous 90%. In fact, it is impossible to know the real status of the task until it is 100% complete and the deliverable is available for review. Therefore, the worst delay in task completion that may happen in this case is 40 hours, after which the corrective measures can be taken. If the task is eight weeks long, it will become obvious eight weeks down the road that the task is far from completion.

The correctly designed WBS of the medium size project has between several hundreds and one thousand tasks. If many tasks are shorter than 16 hours, then the WBS of the same project may have many thousands of tasks, which makes the project less manageable. There is an exception for mini projects with duration of several days or weeks, which may have shorter tasks. WBS creation should involve working sessions with all key technical personnel.

A Decomposition Diagram for a children's birthday party is presented in Fig 7-1. The diagram is just a sample and many essential tasks are not included for the sake of keeping it simple. All activities and tasks are shown as rectangles. The top level activity Children's Birthday Party is decomposed into four activities:

- Invite Guests
- Get Supplies
- Conduct Party
- Clean Up the Mess

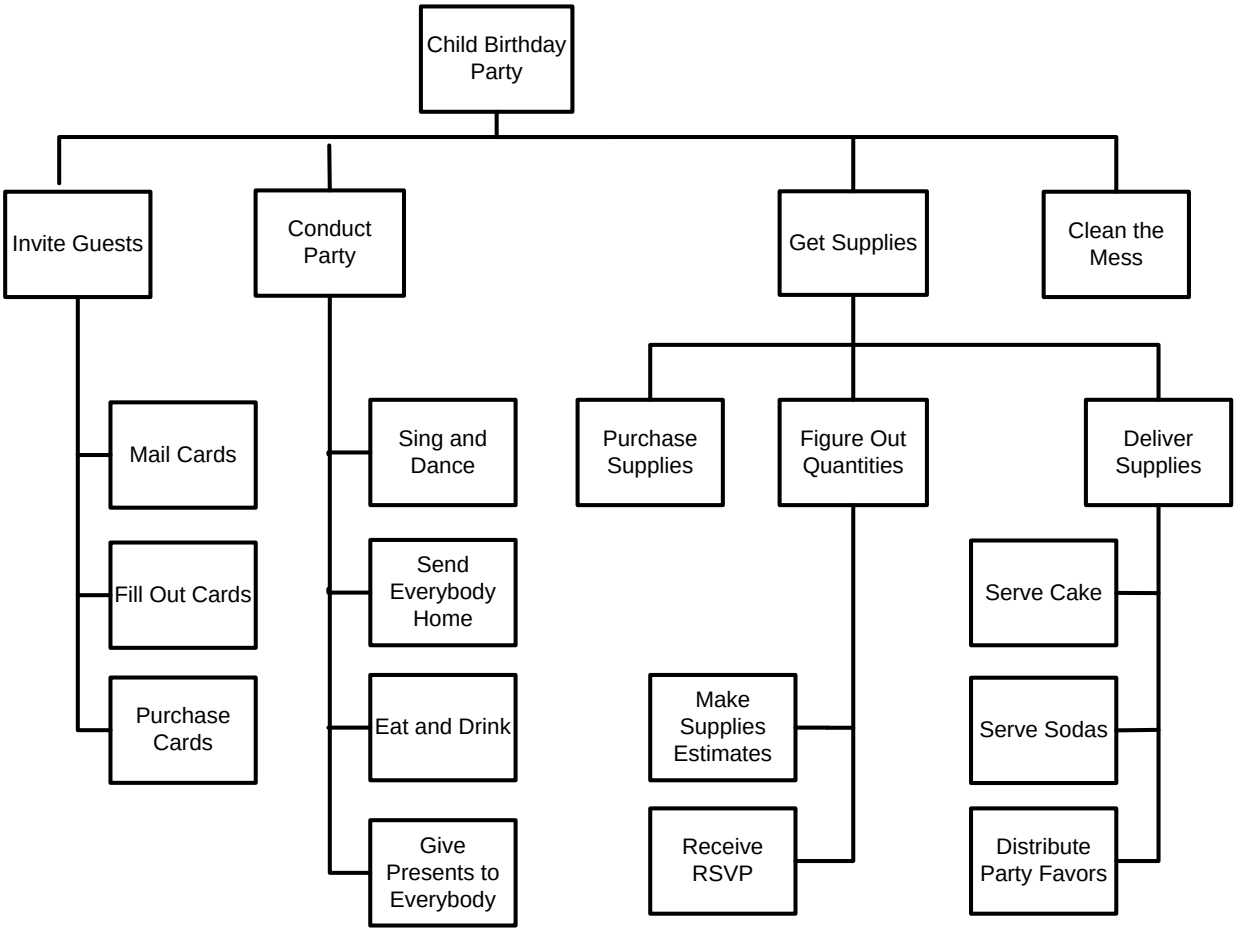


Fig 7-1

A decomposition diagram is not a chronological list of tasks. At this point, sequencing and duration of tasks is not really considered. The focus of the WBS at this point is clearly delineating what work is in the project. The WBS is a complete list of tasks required to deliver the entire project scope with no regard to their sequence, duration or assigned resources yet. Each task must point out to the specific element of the project scope. While there may be more than one task needed to implement one element of the project scope, there should not be duplication of efforts, when two tasks do the same work.

The activity decomposition diagram on Fig 7-1 represents a list of tasks required to deliver the mini-project. If the WBS has several hundred tasks, it is impractical to draw this type of diagram.

The WBS is now used as the indented task list on the left-hand side of a typical Gantt chart. This list, as we continue working with the WBS, gradually evolves into the schedule with the addition of task duration and sequencing information. The Gantt chart is used in all project scheduling tools, such as Microsoft Project, Clarity, Primavera and others. The lowest level executable activity is a task. All others are called activities and presented in the Gantt chart mostly for clarity and context. Higher-level (or "rolled-up" activities are displayed in bold font, while the subservient individual task names are in regular font, as shown on Fig 7-2.

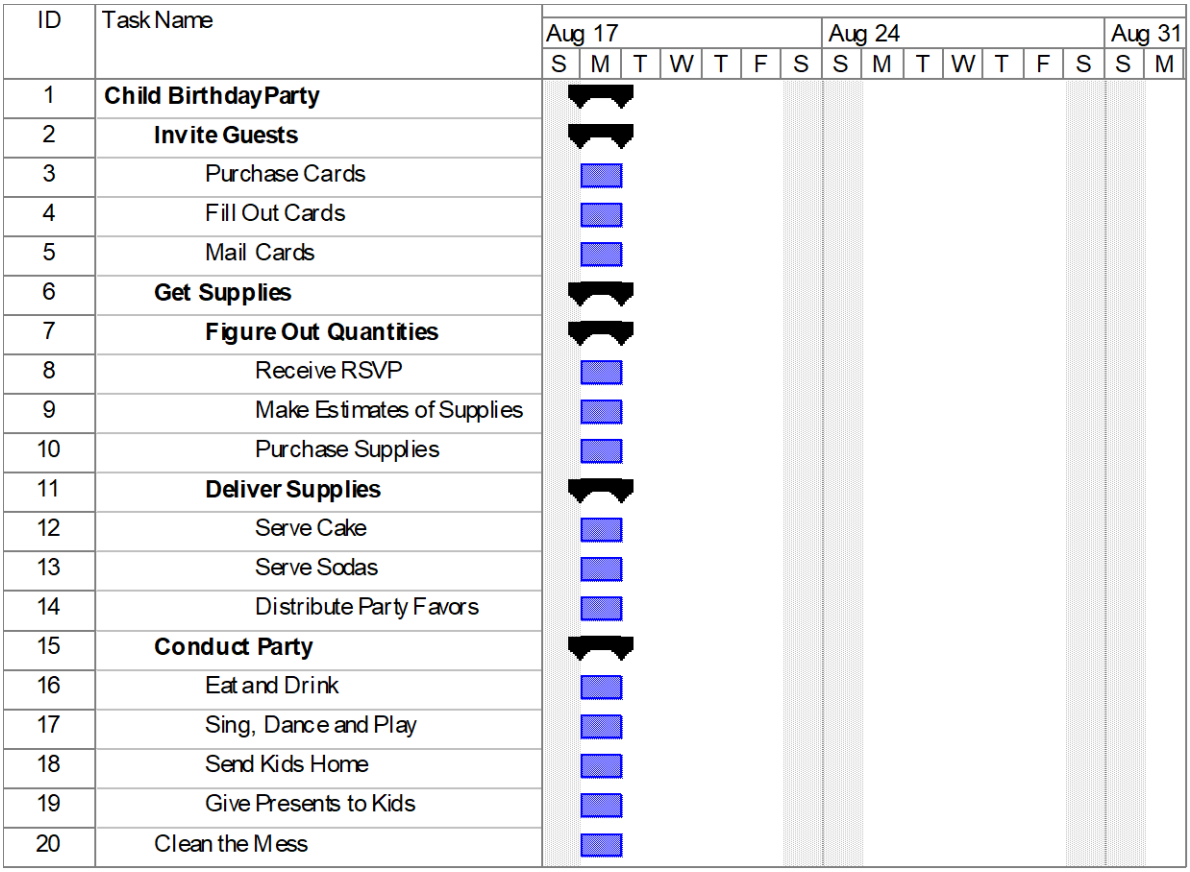


Fig 7-2

Building a Preliminary Project Schedule

The indented task list is a first step in building project schedule. To develop it further, the following steps are required:

1. Enter task dependencies, also called task precedence
2. Estimate all tasks
3. Assign Resources

Building Task Dependencies

Task dependencies indicate relationships between tasks. For example, you can't mail out cards (task) before you purchase them (task). Also, sometimes a delay is required between the end of one task and beginning of another, as in the case of mailing cards and receiving RSVPs. The dependency relationship is shown on Fig 7-3.Tasks may be taken on in one of the following relationships:

- Finish to Start (Task starts at the time when a predecessor task ends)
- Start to Start (Task starts at the time when a predecessor task starts)
- Finish to Finish (Task ends at the time when a predecessor task ends)

- Start to Finish (Task ends at the time when a predecessor task starts)

By far, the most commonly-encountered sequence of tasks is Finish to Start, where one task starts after the previous task ends – this is illustrated in our example of the cards for the children's birthday party. In each of the above relationships, a lag between each task and its predecessor is zero by default, but may be defined differently. In our case, a lag of 5 days is a waiting time required between the finish of task Mail Cards and the start of task Receive RSVP.

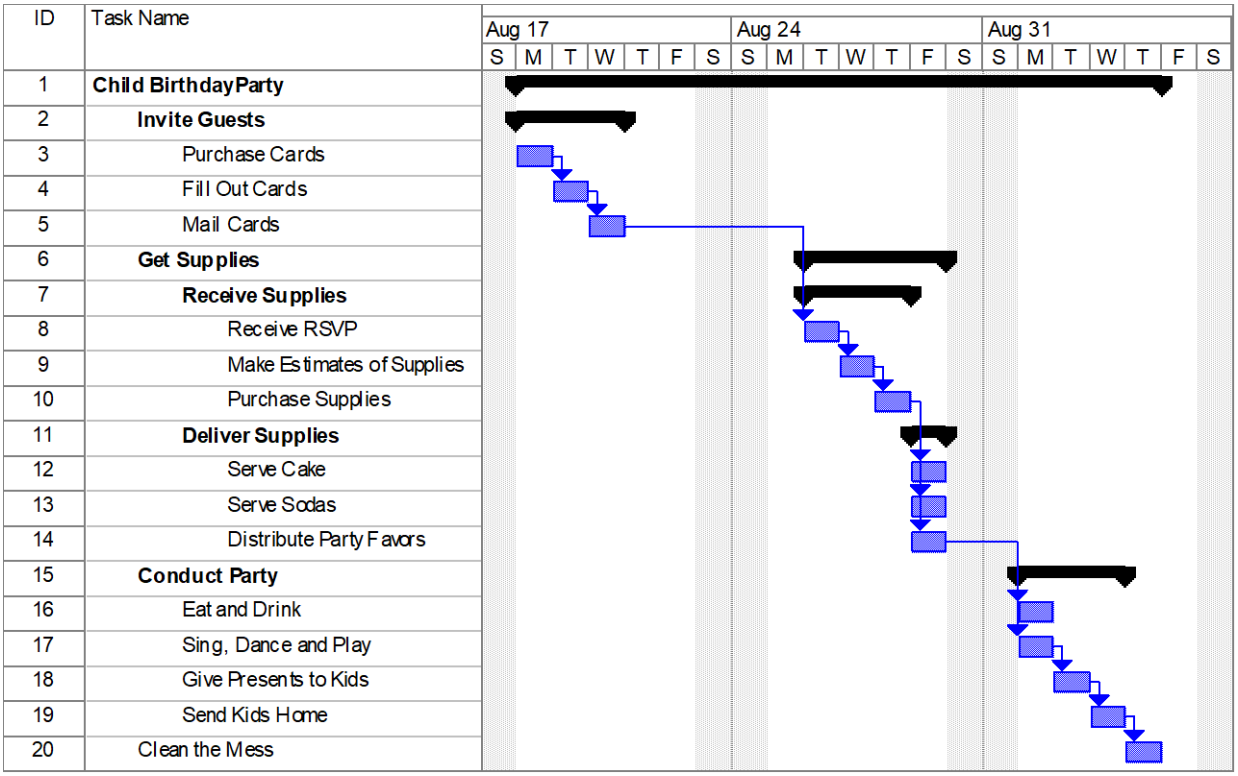


Fig 7-3

Entering Task Constraints Type

Another important parameter that should be taken into consideration is Constraint Type, which describes the period when the task should start or finish. The most used Constraint Type is As Soon As Possible, which means the next task should start immediately, unless the lag is defined. There are eight Constraint Types, which will be explained later, when the Critical Path Analysis is covered:

1. As Soon As Possible
2. As Late As Possible
3. Finish No Earlier Than
4. Finish No Later Than
5. Must Finish On
6. Must Start On
7. Start No Earlier Than

8. Start No Later Than

Constraint types 3 to 8 are useful, but must be used with extreme caution, because they are tied up to specific dates. If the schedule is later modified, tasks with Constraint types 3 through 8 won't move, which creates confusion in the project scheduling and error messages generated by scheduling tools. Inexperienced project managers make that mistake often, using Must-Start-On constraint, later unable to update the schedule, unless constraints are modified. In a large project of 1000 tasks and more, it takes a while to fix this problem.

The **Must-Start-On** constraint should be used for tasks like a scheduled training with many participants, which are difficult to reschedule. The predecessor task is supposed to be completed a while ago, so if the schedule changes, there is still enough time to start the task at the earlier scheduled date.

Must-Finish-On constraint is used when the task must be finished on the specific date.

Entering Tasks Estimates

The next step in building a schedule is estimating all tasks. Each task has two measurements:

- Effort
- Duration

Effort is the number of labor hours required to accomplish a given task. If the resource is assigned to work 100% to the task, which is called 100% availability or 100% load, then the task duration will be equal to the task effort. However, if resource availability is 50%, then it will require twice as many workdays to accomplish the same task. For example, if the task effort is estimated 20 hours and the resource availability is 50%, then the task duration will be 40 hours.

In this book milestones are mentioned many times. As a reminder, a milestone is a zero duration task, which serves as a flag or indication that certain parts of the project are complete. Some project managers like to think of milestones as "anchors" because they help to tie down the project schedule and help connect it to the outside world more directly. In our example, we have the following milestones:

- Cards Mailed
- Ready to have fun
- Thanks, it is over

After estimating task length, our Gantt chart looks as shown on Fig 7-4

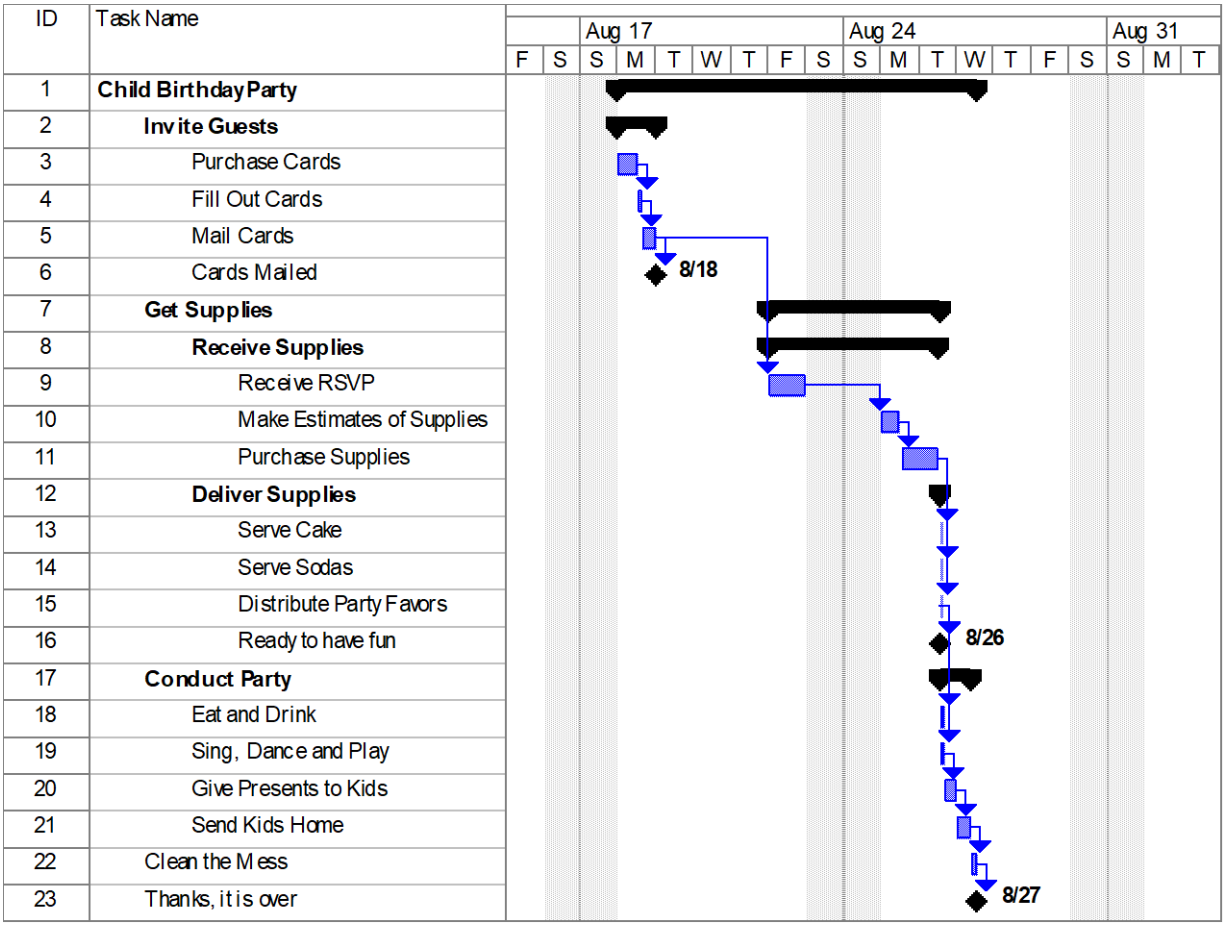


Fig 7-4

Resource Assignment

The next step is to assign resources to the project. In this example, John's availability is 25%, so the required effort of 2 hours to do Purchase Supplies has a total duration of 8 hours. Also, tasks 18 and 19, which are performed by resource Children, should take place at the same time. If that resource had 100% availability, the scheduling tool would automatically put it one after another, because a resource cannot be scheduled for 200% availability. In order to prevent this from happening, their availability must be assigned to 50%. In total, the duration of each task is 4 hours, performed at the same time, while the effort of each task is 2 hours. The resource assignment is displayed on Fig 7-5

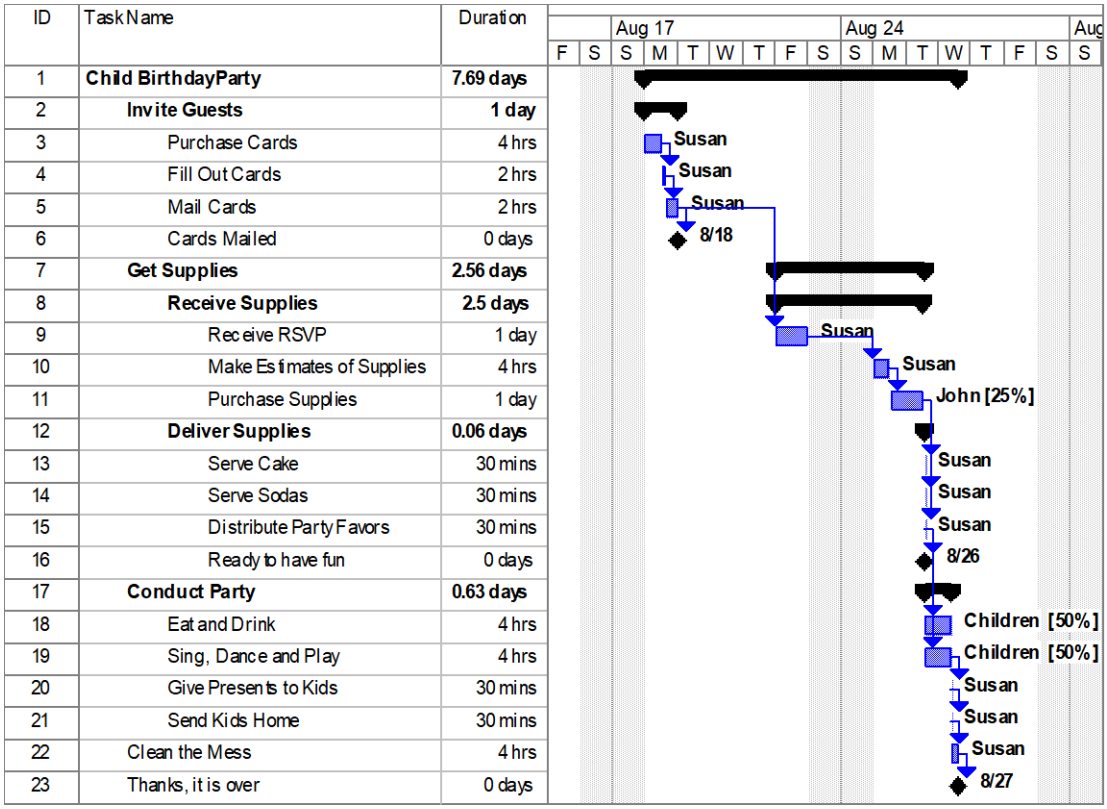


Fig 7-5

Modifying the Project Completion Date

It appears that the project will be completed on 8/27, which is when the party takes place. But what if the birthday in on 8/25 and you must have the party on 8/25? One of many ways to accomplish it is by setting a Must-Start-On constraint on the first task, Purchase Cards, moving the end of project back two working days to 8/25, as displayed on Fig 7-6. There are additional methods that may be used to accomplish this schedule compression. Those methods will be reviewed in the following section.

ID	Task Name	Duration
1	**Child BirthdayParty**	**7.69 days**
2	**Invite Guests**	**1 day**
3	Purchase Cards	4 hrs
4	Fill Out Cards	2 hrs
5	Mail Cards	2 hrs
6	Cards Mailed	0 days
7	**Get Supplies**	**2.56 days**
8	**Receive Supplies**	**2.5 days**
9	Receive RSVP	1 day
10	Make Estimates of Supplies	4 hrs
11	Purchase Supplies	1 day
12	**Deliver Supplies**	**0.06 days**
13	Serve Cake	30 mins
14	Serve Sodas	30 mins
15	Distribute Party Favors	30 mins
16	Ready to have fun	0 days
17	**Conduct Party**	**0.63 days**
18	Eat and Drink	4 hrs
19	Sing, Dance and Play	4 hrs
20	Give Presents to Kids	30 mins
21	Send Kids Home	30 mins
22	Clean the Mess	4 hrs
23	Thanks, it is over	0 days

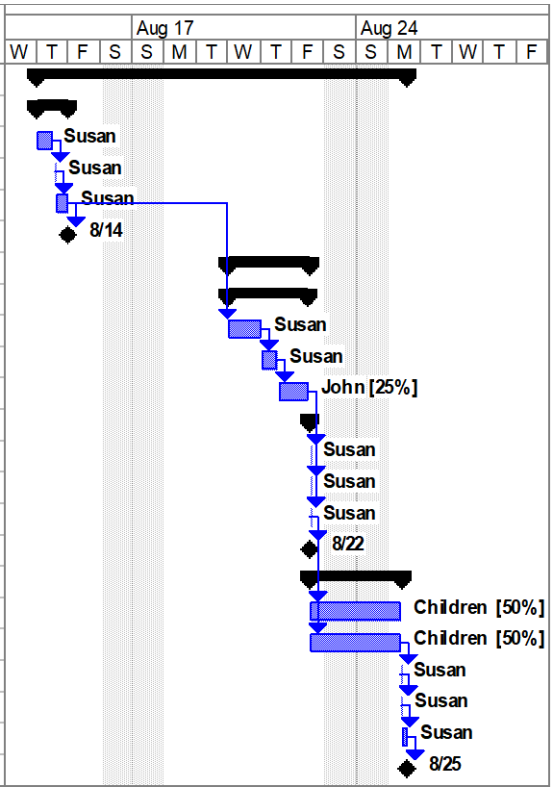


Fig 7-6

Critical Path Analysis

Critical Path Analysis (CPA) is a method of identifying tasks which must be completed on time in order to have the whole project completed on time. Those tasks are called critical tasks. Note that critical tasks are not necessarily any more "important" than other tasks. What makes them "critical" is that any slip in any critical path activities, will, by definition, cause a slip in the overall project's end date.

That means that non-critical tasks may have a calculated maximum allowable delay in their completion without delaying the project completion. CPA will calculate the delay between completion of one task and the beginning of the next one. CPA allows project managers to calculate project duration and to improve resource utilization.

The Network Diagram or PERT chart presents all tasks as small rectangles or circles. It shows tasks, their dependencies and duration, just as Gantt chart, but also helps to identify the longest path of the project, which is called critical path. The Birthday Party project is too simple to show a meaningful Network diagram; therefore a generic and slightly more complex project is used in Fig 7-7 to illustrate the CPA concept. This generic project has 15 tasks identified by numbers 1 through 15. The duration of each task in days is indicated on top of each rectangle.

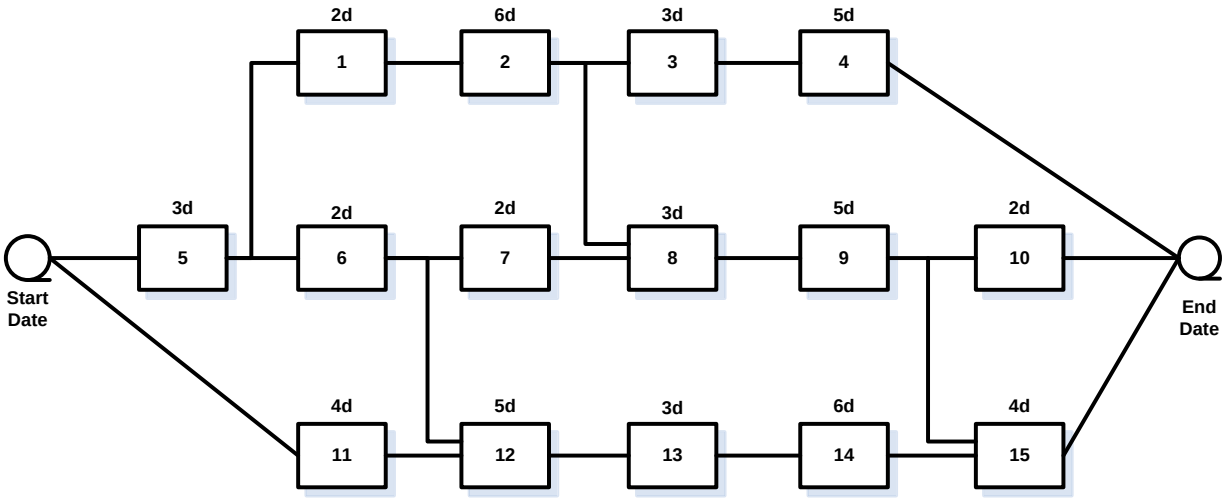


Fig 7-7

The project path is a contiguous chain of tasks from the start to the end of a project. Note that technically, 'start' and 'end' are zero-duration tasks. There are seven paths in the Network Diagram on Fig 7-7:

Path 1:	Start, 5, 1, 2, 3, 4, End	Total duration: 19 days
Path 2:	Start, 5, 1, 2, 8, 9, 10, End	Total duration: 21 days
Path 3:	Start, 5, 1, 2, 8, 9, 15, End	Total duration: 23 days
Path 4:	Start, 5, 6, 7, 8, 9, 10, End	Total duration: 19 days
Path 5:	**Start, 5, 6, 12, 13, 14, 15, End**	**Total duration: 25 days** (longest path)
Path 6:	Start, 5, 6, 7, 8, 9, 15, End	Total duration: 21 days
Path 7:	Start, 11, 12, 13, 14, 15, End	Total duration: 22 days

Note that it is not the number of tasks in a path that is important, but rather, the accumulated (added) duration in the path. Path 5 in the above example is the critical path, because it is the longest path, which, in turn establishes the project duration of 25 days. If the project start day is in the morning of 01/01, then the project end will be the end of day on 01/25, assuming that Saturdays and Sundays are workdays. If all tasks have the constraint "As Soon As Possible", then all tasks in path 1 start and finish on following dates, as shown in Table 7-1.

Task	Start Date (morning)	Finish Date (end of the day)
5	01/01	01/03
1	01/04	01/05
2	01/06	01/11
3	01/12	01/14
4	01/15	01/19

Table 7-1

Let's look at task 4. The earliest possible date it can start is 01/15, which is called Early Start. Subsequently, the earliest possible finish date is 01/19, which is called Early Finish. If the start day is delayed for 2 days, the task will be finished on 01/21 without impact on the project

schedule, still several days before the scheduled end of the project on 1/25. The latest date the task may finish without causing delay of the project is 01/25, which is called Late Finish. Subsequently, the latest date when the task can start is 01/21, which is called Late Start. The difference between Late Start and Early Start (or Late Finish and Early Finish) is 5 days, which is called Slack or Float. This is the time by which the task can be delayed without impact to the project schedule.

Compressing the schedule

During the course of a project we often must complete the project or some tasks earlier than scheduled, due to new imposed dates or other significant issues. If we speed up implementation of tasks not on the critical path, the project duration in our example still will be 25 days. It really does not make any difference if the duration of path 1 is reduced to 10 days, because the project duration will still be 25 days. The only way to reduce project duration (i.e. "schedule compression") without changing project scope is via the following methods:

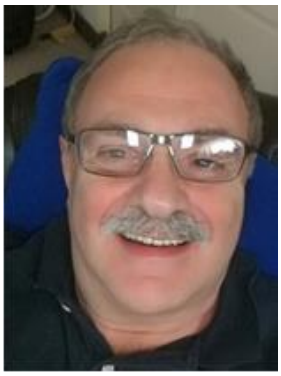
- **Fast tracking**. This is scheduling of tasks on the critical path in parallel, whenever possible without violating task dependency rules.

- **Crashing**. This is adding extra resources to tasks on the critical path. However, implementing the same task by two resources at the same time makes it uncertain who is doing what, reducing accountability of resources and their productivity. If it is possible to split a task to two or more independent parallel tasks and assign them for implementation to two different resources, then crashing may be used effectively.

- **Extending working hours.** This method is used quite often, but caution must be exercised not to do it for extended periods of time, because this may wear the staff out, reducing their productivity.

The schedule is dynamic – and so is the critical path. Even if the duration of the critical path 5 is reduced to 20 days, the project duration won't be reduced to 20 days. As soon as the duration becomes less than 23 days, path 3 will become the new critical path with duration of 23 days. Any subsequent reduction of the duration of path 5 (which is no longer 'critical') won't give any benefits to the project schedule and the project duration will remain at 23 days.

About the Author



Dan Epstein

New York, USA

Dan Epstein combines over 25 years of experience in the project management field and the best practices area, working for several major Canadian and U.S. corporations, as well as 4 years teaching university students project management and several software engineering subjects. He received a master's degree in electrical engineering from the LITMO University in Leningrad (today St. Petersburg, Russia) in 1970, was certified as a Professional Engineer in 1983 by the Canadian Association of Professional Engineers – Ontario, and earned a master's certificate in project management from George Washington University in 2000 and the Project Management Professional (PMP®) certification from the Project Management Institute (PMI®) in 2001.

Throughout his career, Dan managed multiple complex interdependent projects and programs, traveling extensively worldwide. He possesses multi-industry business analysis, process reengineering, best practices, professional training development and technical background in a wide array of technologies. In 2004 Dan was a keynote speaker and educator at the PMI-sponsored International Project Management Symposium in Central Asia. He published several articles and gave published interviews on several occasions. In the summer of 2008 he published "Methodology for Project Managers Education" in a university journal. His book, *Project Workflow Management - The Business Process Approach*, written in cooperation with Rich Maltzman, was published in 2014 by J. Ross Publishing.

Dan first started development of the Project Management Workflow in 2003, and it was used in a project management training course. Later this early version of the methodology was used for teaching project management classes at universities in the 2003–2005 school years. Later on, working in the best practices area, the author entertained the idea of presenting project management as a single multithreaded business workflow. In 2007–2008 the idea was further refined when teaching the project management class at a university. Since 2009, Dan has continued working full time in Project Management. Dan can be contacted at dan@pm-workflow.com.