Geometric

Eliminating Waste in Software Project Management using Critical Chain Project Management™

Version 1.0
March 2009

WHITE PAPER

Copyright Notice

Confidentiality Notice

Contents

Common factors and issues encountered in managing software development projects 4

Current Project Environments 5

Parkinson's Law and Student's Syndrome 6

Multi-tasking 6

What is CCPM? 8

Achieving Speed and Reliability 8

Improving Project Throughput 9

Tackling Uncertainties 9

Project Monitoring and Control 9

Conclusion 10

About the Author 10

About Geometric 11

Common factors and issues encountered in managing software development projects

- Original deadlines are not adhered to
- There is significant re-work
- There is too much uncertainty
- Actual work is much more than originally planned
- Resources are not available to work when needed
- Project delivery is rampant with late effort
- Quality is compromised in meeting delivery schedules
- Enough safety buffers are put in our estimates and are yet unable to deliver on time
- The project has been 90% complete for ages

If these points are largely familiar to what you see in your projects, do read on.

The project manager's need in software projects is to ensure delivery on or before time. A mechanism by which we can achieve this is to have aggressive effort estimates for the tasks against the requirements. However, in order to minimize the challenge of uncertainties that we face in a project environment, we introduce safety margin in tasks at each stage. This is a standard conflict that is faced in software project management leading to most of the issue listed above.

"Work expands to fill (and often exceeds) the time allowed" – Parkinson's Law

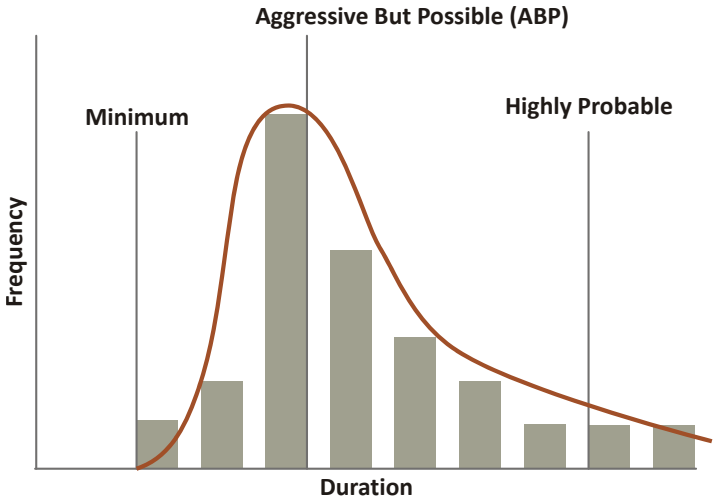
"Whatever can go wrong, will" – Murphy's Law

"There is more work performed close to a deadline" – Student's Syndrome

Current Project Environments

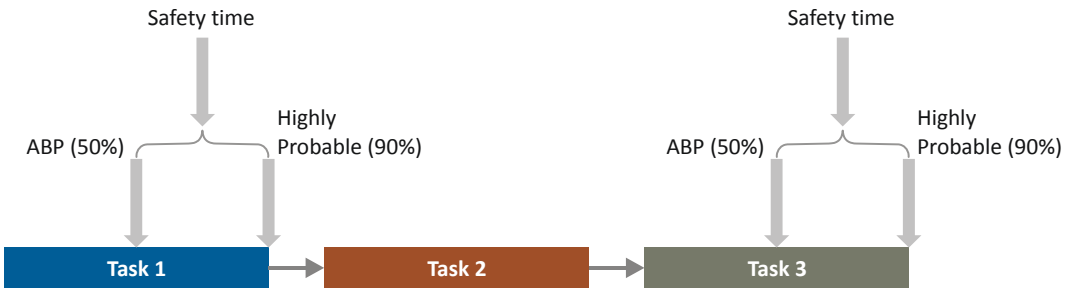
Let us evaluate current planning and execution methods that are practiced in most project environments. If we look at the estimation process - the project deadline date, which is a deterministic deadline, is arrived at by linking together a set of dependent tasks and effort estimates which are probabilistic in nature. The performance of these tasks is monitored against the milestone or deadline they are supposed to be delivered against.

Since task estimates are probabilistic in nature, they will typically follow a normal distribution curve. This normal distribution curve is usually skewed as shown in the following figure.



Project managers plan duration estimates that they believe they have a high probability of completing the work in, by adding adequate safety time in their task estimates. This is done is because of the need to compensate for uncertainty and the fear that their estimates may get cut during a review process. In addition to this there are other activities which are currently on the person's desk which need urgent attention. So in effect, all estimates that are provided have considerable safety built in. In fact, so much safety that the team should always be in a position to deliver the project on or before time.

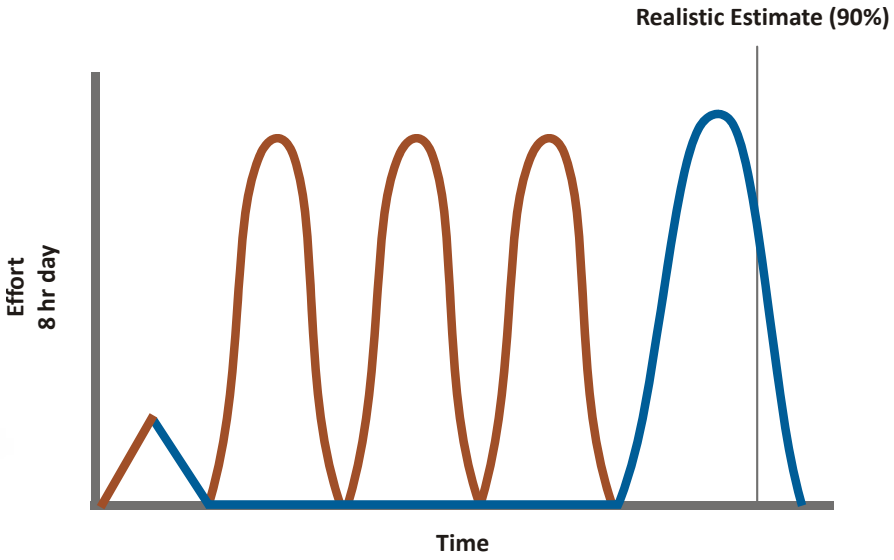
So what does really happen to this safety during project execution?



There are three main reasons as to why the task safety is wasted. These are due to Parkinson's Law, Student's Syndrome and Multi-tasking. Let us look at the effects of each of these parameters.

Parkinson's Law and Student's Syndrome

Once teams have provided a 'safe' task deadline, they tend to plan their tasks around this deadline and focus on delivering against the same. In execution often other 'urgent' work gets the attention rather than the task at hand, whose due date is still out there somewhere. Remember, safety was put in not only to protect against activities that needed urgent attention, but also uncertainties. By starting a task later than originally planned, the team has wasted the safety that was provided for the task, considerably risking the task deadline. This is the effect of Student's Syndrome, where the teams will also put in extra efforts as the deadline approaches.

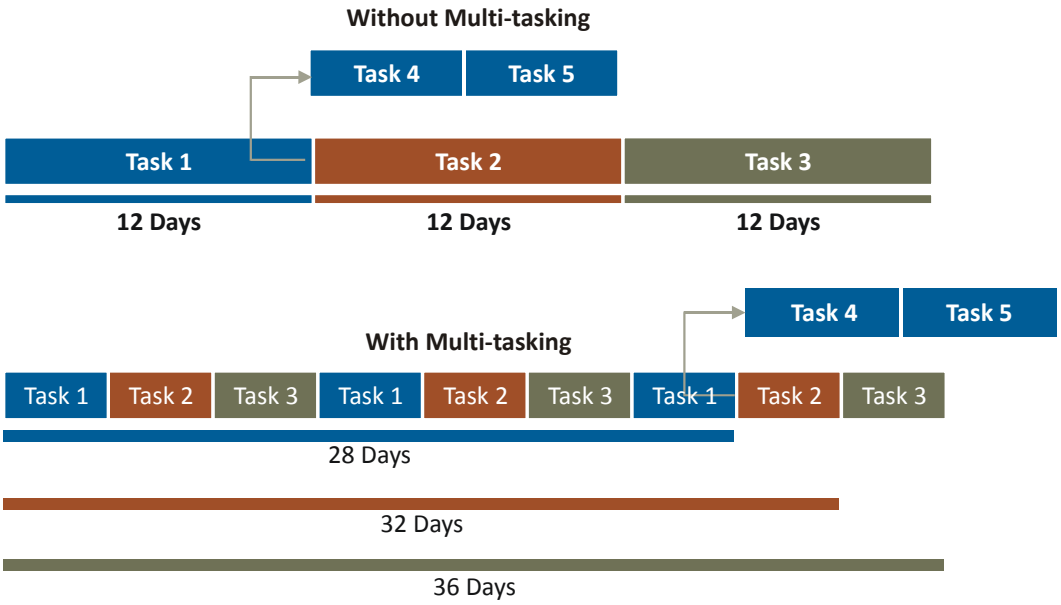


If, a resource does manage to complete and deliver his task earlier than the schedule because uncertainty did not hit the task, he will receive kudos. Achieve this twice in a row and it is likely that the resources will be suspected of over estimating the task. In such a scenario, even if the resource finishes the task early, the person does not report the early finish. In addition to this, the team may still not be able to start the next task due to a resource dependency working on some other task.

This delay propagates across tasks and its effect is clearly seen at an integration point within the project. So in essence, time gained is not reported whereas time lost propagates and affects the project schedule. Typical affects of the Parkinson's Law.

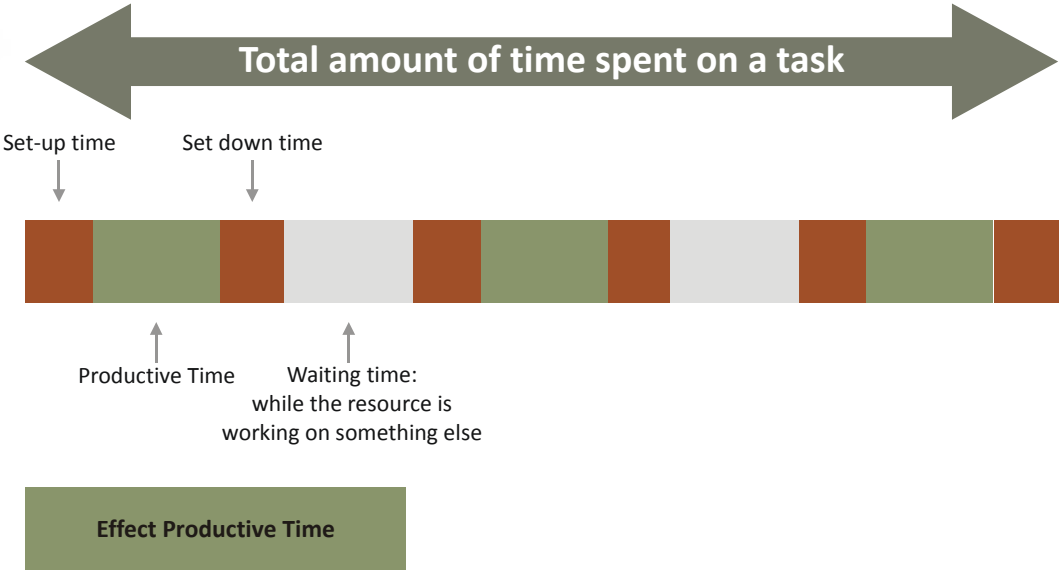
Multi-tasking

One of the other large causes of project delay is multi-tasking. Multi-tasking has two major negative impacts on a project deadline.



Consider three tasks as in the figure above. Without multi-tasking, Task 1 would be finished in 12 days, Task 2 in 24 days and Task 3 in 36 days cumulatively. With multi-tasking, Task 1 will finish in 28 days, Task 2 in 32 days and Task 3 in 36 days. Hence there is a 16 day delay in Task 1 and an 8 day delay in delivering Task 2. If any of these two tasks has dependencies from other tasks in the project plan, the team will lose any advantage of delivering projects early.

The second effect is due to productivity losses during multi-tasking.



Each task, no matter how simple or complex, has a setup time and a set down time. A lot of productivity loss is also due to the other activities that a resource would dwell into when the person is actually supposed to focus on completing the planned task. The typical manifestation of

this problem happens when a resource working on a feature implementation, suddenly has a bug delivered on the table to fix. This changed focus leads to losses of efficiency during the tasks.

How do we resolve the problems posed by the above factors for project planning and project management? The answer probably lies in using Critical Chain Project Management ™ (CCPM).

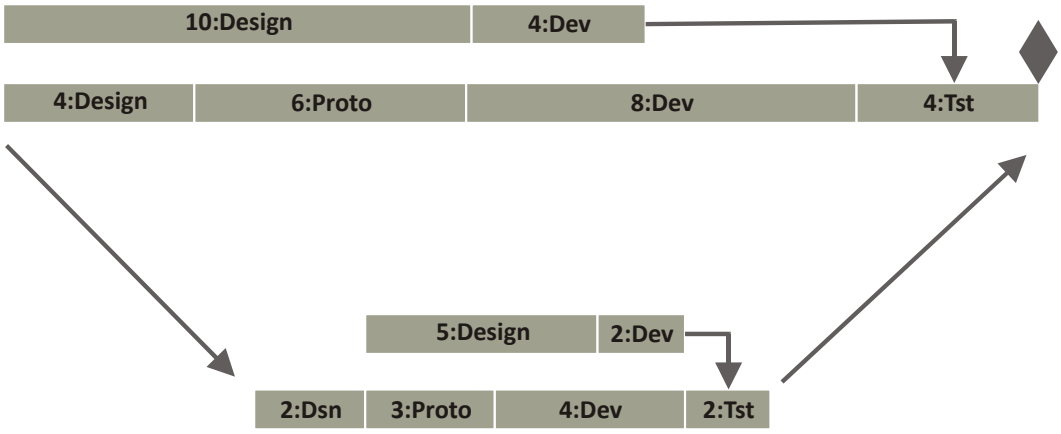
What is CCPM?

Developed by Eliyahu M Goldratt, CCPM is based on methods and algorithms derived from the principles of Theory of Constraint (TOC). Geometric has successfully adopted these methods to software project management. It delivers projects 10%-50% faster compared to traditional methods, and achieves the discussed improvements through a process of 'waste reduction'. Critical Chain is a combination of Critical Tasks and Critical Resources in the project.

Using the CCPM technique, we address the challenges by making modifications to two key processes - Project Planning and Project Monitoring & Control.

Achieving Speed and Reliability

In Project Planning, what really matters is completion of the project on time and not completion of individual tasks on time. We can eliminate Student's Syndrome and Parkinson's Law by taking safety allowance out from tasks and putting it at the strategic points of the project. The strategic points are at Integration points, which would be at the end of the project and at intermediate points where integrations occur. The allowance at the end of the project is called Project Buffer while that at intermediate integration points is called Feeding Buffer. Typically, the ABP level (Aggressive but Possible) in an estimate is 50% of the Highly Probable level in a skewed normal distribution curve. So we can remove task buffers by cutting the total task estimate in half.



The implementation of CCPM requires a change in management behavior as well. One step that will enable success is that resources should not be measured on the success of delivering tasks on time. This behaviour if not changed will lead to an implementation failure.

The method helps leverage the benefits of tasks finished early by identifying the resources working on critical tasks and ensuring their availability, when the preceding task is done. For this to be successful, resources have to provide regular alerts on how much of their current task is pending. Hence, this process requires adequate and due behavioral acceptance by the management and the teams on projects.

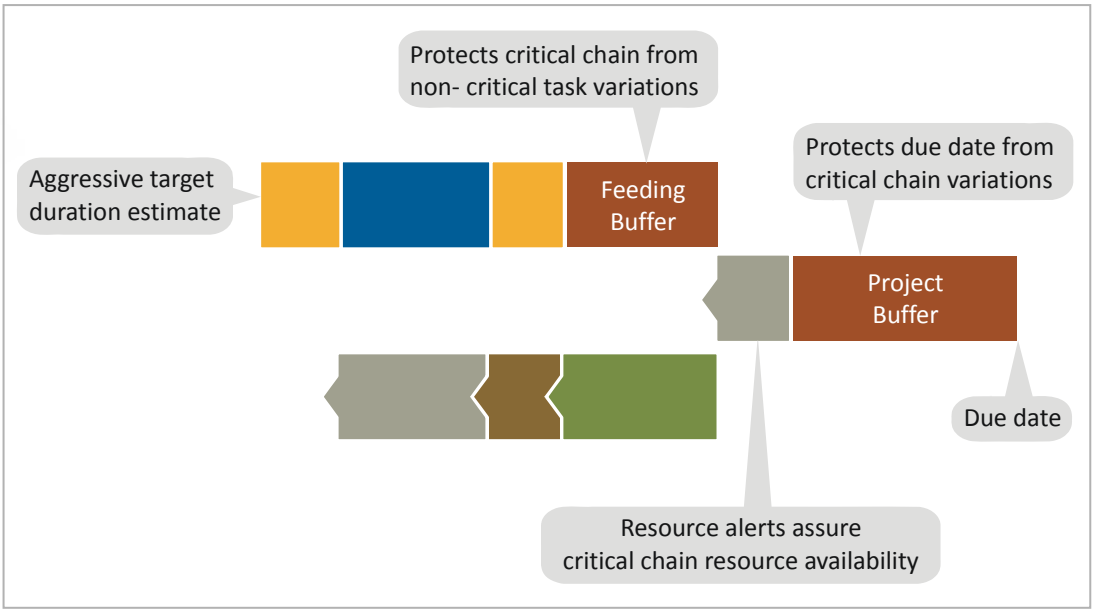
Improving Project Throughput

Every project will have resources that act as bottlenecks and prevent improved throughput. We will have to identify the bottlenecked resource in the system and ensure elimination of multi-tasking on the bottlenecked resource. This can be done by reducing the number of Work-In-Process (WIP) or open work fronts based on the constrained resource in the project.

Tackling Uncertainties

Although we have established a tight schedule supported by resource alerts to ensure resource availability as needed, it will still not offer us protection against uncertainties. We protect against uncertainties by adding 50% of the total task buffer to the strategic points mentioned earlier.

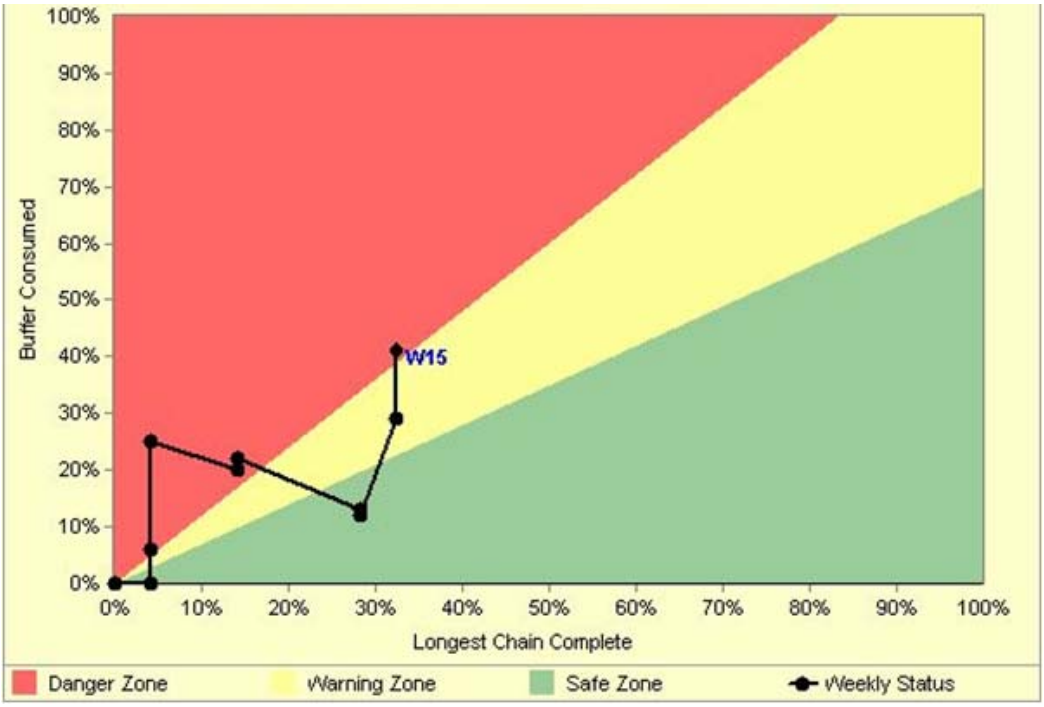
This will lead to a Benefit of Aggregation, which ensures the same amount of safety to give a higher degree of protection when aggregated, leading to a major reliability benefit in the execution of project plans.



Project Monitoring and Control

We use the principle of Buffer Management to monitor project progress and expedite corrective steps. Remember, we now have aggressive task schedules, which are protected by buffers added to strategic points in the critical chain.

It is very likely that tasks on the critical chain will not achieve their ABP delivery times and will eat into the project buffer. We will monitor the progress of a project by measuring the progress along the critical chain and the total project buffer that has been consumed.



In above tracking plot, when the project is in red, we need to immediately have expediting actions in place, when it is in yellow we need to plan for expediting and when it is in green we take no action.

Conclusion

Critical Chain Project Management provides measurable benefits of between 10%-50% reduction in the overall cycle time of a project, without impacting project costs. It also shows significant benefits in providing the right controls to effectively manage projects, and allowing actions to be expedited as and when they occur in a project cycle. What is needed in order to successfully implement CCPM is a change management process, and a dedicated team that will help in institutionalizing the changes required.

About the Author

Venkatesh Jagannath

Venkatesh Jagannath is a Vice President and the head of the Software Product Engineering (SPE) business unit at Geometric. He has been part of Geometric for over 14 years, during which he handled various roles and responsibilities. Venkatesh started his career at Godrej and Boyce in

1993, as part of the GCAM product team. For over ten years, he played a key role in Geometric's IP development and spearheaded the desktop products development. In July 2003, he moved to head the quality initiative and was responsible for Geometric's successful assessment at CMMI Level 5 in February 2004. For a short span in 2004, he was the delivery head for Geometric's software OEM customers and responsible for quality, delivery and customer satisfaction. In 2006, Venkatesh pioneered the organization productivity initiative to drive a Theory of Constraints (ToC) and Critical Chain Project Management (CCPM) based program aimed at improving team productivity levels. Since April 2007, he has been responsible for Geometric's SPE business which targets the Outsourced Product Development (OPD) markets. Venkatesh holds a Mechanical Engineering degree from the University of Mumbai and an M.Tech. from the Indian Institute of Technology, Delhi. Venkatesh is a keen tennis player and likes to read and listen to music in his spare time.

About Geometric

Geometric is a specialist in the domain of engineering solutions, services and technologies. Its portfolio of Global Engineering services and Digital Technology solutions for Product Lifecycle Management (PLM) enables companies to formulate, implement, and execute global engineering and manufacturing strategies aimed at achieving greater efficiencies in the product realization lifecycle.

Headquartered in Mumbai, India, Geometric was incorporated in 1994 and is listed on the Bombay and National Stock Exchanges. The company recorded consolidated revenues of Rupees 4.86 billion (US Dollars 121.6 million) for the year ended March 2008. It employs close to 3000 people across 10 global delivery locations in the US, France, Romania, India, and China. Geometric is assessed at SEI CMMI Level 5 for its software services and ISO 9001:2000 certified for engineering operations. For further details, please visit www.geometricglobal.com