

Briefing

The critical path you're not watching

Why CP2 and CP3 often deserve more attention than CP1.

Synopsis. Most program teams focus on CP1 and miss the near-critical paths behind it. This briefing uses a real critical-path snapshot to show how buffer-burn rate flags which path becomes the next constraint, and why CP2 through CP5 all need to be shortened in step.

Most program teams stare at one number — the longest critical path — and treat the rest as background. That is half right. CP1 is the schedule constraint by definition, so it earns close attention. The other half is dangerous. The paths sitting behind CP1 quietly consume their buffer faster than CP1 is being pulled in, and one morning they wake up as the new CP1, with a thin pull-in plan and weeks of slip already baked in.

Critical Path Analysis: Top critical activity
Project schedule, weekly snapshot

CP	Last task in critical path	Owner	Gap <i>Early/Late</i>	9 Week Trend <i>Change Pull-in/Slip</i>	9 Week Average Trend	Buffer	Target: 6/30/2027	Pull-in Actions	
1	Critical milestone alpha	Owner A	(65)		0	Pulled-in 2 days over the last 7 weeks	0	9/3/2027 <small>Apr 22 - Confirm supplier estimate of earlier delivery, Owner: Lead 1, Due: 06/05/2026 Apr 22 - Align on engagement to shorten supplier turnaround, Owner: Lead 2, Due: 06/05/2026 Apr 01 - Discuss mitigation options for upstream process, Owner: Lead 3, Due: 4/3</small>	
2	Test program completed	Owner B	(52)		(18)	Slipped 59 days over the last 7 weeks	13	8/21/2027	
3	Write functional test spec	Owner C	(40)		(12)	Slipped 40 days over the last 7 weeks	25	8/9/2027	
4	Package tester setup	Owner A	(34)		0	Slipped 86 days over the last 7 weeks	31	8/3/2027	Apr 29 - Task is cancelled. Look for alternative tester capacity, e.g. renting tester time externally.
5	Revise and release plan (Final)	Owner D	(30)		0	Pulled-in 16 days over the last 7 weeks	35	7/30/2027	Apr 15 - Push process owners to faster decisions, Owner: Lead 4, Due: 27/04/2026
6	Supplier commitment	Owner D	(19)		0	No change over the last 1 weeks	46	7/13/2027	
7	Develop parametric test program	Owner C	36		(7)	Slipped 42 days over the last 7 weeks	101	5/25/2027	
7	Develop characterization plan	Owner C	36		(5)	Slipped 84 days over the last 7 weeks	101	5/25/2027	
8	Validation milestone	Owner E	40		(7)	Slipped 9 days over the last 5 weeks	105	5/21/2027	

Critical path snapshot

Figure 1. Critical path snapshot showing the nine top activities ranked by gap to the 6/30/2027 target. Anonymized; structure preserved from a real program review.

What the data actually says

The headline reads as a CP1 problem. The top activity is 65 days late, has zero buffer remaining, and lands on 9/3/2027 against a 6/30/2027 target. Three pull-in actions are open. The 9-week sparkline saw-tooths; over the last seven weeks the team has pulled in 2 days. That activity is being managed.

The story below the headline is the one to worry about. CP2 has slipped 59 days over seven weeks, with 18 of them in the two weeks since 04/22. Buffer remaining: 13 days. CP3 has slipped 40 days over seven weeks, 12 of them since 04/22. Buffer remaining: 25 days. Both sparklines climb monotonically — no resistance in the system. The Pull-in Actions column for CP2 and CP3 is empty.

Run the math at the recent rate. CP2 is burning buffer at about 9 days per week, so its 13 days disappears in roughly 10 days. CP3 is burning at about 6 days per week, so its 25 days lasts about 4 weeks. If nothing changes, CP2 becomes the new CP1 inside two weeks; CP3 follows within a month.

What makes this a red flag rather than a maybe: CP1's recent change is 0 (it has stabilized), while CP2 and CP3 are still slipping at their fastest recent rate. Pull-in attention is concentrated where it has already worked. The paths that need it next are invisible to the team.

Buffer-burn rate is the leading indicator. Current buffer tells you where you stand. Buffer-burn rate — slip days per week over the last two or three weeks — tells you when each path becomes the next constraint. Sort the top five paths by buffer divided by recent burn rate and you have a ranked list of which path will hurt you next, and how soon.

What to do about it

Three moves keep a program ahead of this pattern. Broaden the watch list to the top three to five critical paths every week, not just CP1. Score each path by buffer-burn rate, not just current gap; a small slip with a high burn rate is a bigger risk than a large slip that has already stabilized. When CP1 stabilizes, redirect that pull-in capacity to CP2, CP3, and out to CP4 and CP5 if the team has the bandwidth to spread it that wide. Pull-in work is cheapest while there is still buffer to defend; once a path becomes CP1 the options narrow and the cost rises.

Assume CP1 is getting shorter, not longer. Most teams plan around the implicit assumption that CP1 will get longer — schedules usually slip, and the planning reflex accommodates that. When CP1 lengthens, every path behind gains buffer, and those owners quietly absorb the extra room. They take it. They start working to the later date. Then they slip too. That is the mechanism behind most multi-month program slips.

The discipline that breaks the pattern is to assume the opposite, and to shorten CP2 through CP5 in step with CP1. When every path is accelerating with the same intent, slack does not accumulate, owners do not relax, and the program holds its target. In the snapshot above, CP1 has stopped slipping. The team should treat that stabilization as freed-up capacity and push it immediately at the next four paths, preferably in the same week the data says so.

CP Analysis is one of the standard functions in **fastProjectAI**. It surfaces the top critical paths, computes buffer-burn rates, ranks the paths by time-to-overtake, and flags cases where pull-in actions are concentrated on a stabilized path while the next path is unattended. Used in a weekly schedule review, it turns the silent slippers into Monday's conversation, not the one a month later when the new CP1 is locked in.