THE CUTOVER PLAN
CONSTRUCTION STANDARD

Six checks that decide whether the plan you were handed can actually be executed. Work through it line by line with the track lead in the room, not alone at a desk.

> **THE TEST THAT SETTLES MOST ARGUMENTS**
> If a task cannot be handed to a named person who could start it without asking a question, it is not a task. It is a heading with a date attached. Expect to fail 30 to 50 percent of an inherited plan on the first pass.

1. The task schema

Set this before the first task is written. Retrofitting attributes across two thousand lines is a project of its own. Fewer than this and the plan cannot be executed or reported on. More and nobody maintains it.

ATTRIBUTE	WHAT IT HOLDS, AND WHY IT EARNS ITS COLUMN
ID	Stable, never reused, never renumbered when lines are inserted. Renumbering breaks every dependency reference at once.
Task name	Verb first, object second, imperative, under twelve words. Read aloud on a bridge call at 03:00 by someone who is tired.
Phase or wave	Which deployment event this belongs to, as an attribute rather than a section header. A section header cannot be filtered.
Track	Conversion, integrations, security, business, communications. Drives each lead's view and the parallelism in the runbook.
Owner	A named individual. Not a role, not an organization. The most valuable column and the one most often left generic.
Backup	Who answers if the owner is unavailable. Cutover runs outside business hours, so availability gaps are certain.
Duration	Estimated elapsed time, confirmed by the owner. A four-hour load that must run overnight is a twelve-hour elapsed line.
Predecessor	Task IDs that must finish first, with the dependency type stated. Without it there is no critical path and no way to price a delay.
Start condition	What must be true to begin, beyond predecessors finishing: an approval, a file received, an environment available.
Validation	How completion is proven, and by whom. A task marked done on assertion is the most common source of a Day +2 surprise.
Rollback	What undoing this looks like, or explicitly none. Forces the rollback conversation task by task.
Status	Not started, in progress, complete, blocked, skipped. Skipped is a real state and needs a reason field.

2. The line-level quality bar

Run this against a sample of fifty lines before deciding whether the plan is repairable or needs rebuilding. If more than a third fail, rebuild.

Every line

- ☐ Starts with a verb in the imperative
- ☐ Names one individual as owner and one as backup
- ☐ Carries at least one predecessor, or is explicitly marked an entry point
- ☐ States how completion is proven and who proves it
- ☐ Has a duration the owner has personally confirmed
- ☐ Is tagged to a wave, a track and a freeze state

Every section

- ☐ Contains the business tasks, not only the technical ones
- ☐ Contains the decision points where work crosses between organizations
- ☐ Contains the communications that must go out during the window
- ☐ Contains validation tasks, not just execution tasks
- ☐ Has an identified entry gate and exit gate
- ☐ Has a rollback position stated once for the section

Every gate

- ☐ Names the person who makes the call
- ☐ States the evidence required to make it
- ☐ States the time by which it must be made
- ☐ States what happens if the evidence is not there
- ☐ Has been walked through with the person who will make it
- ☐ Appears in the runbook as a hard stop, not as a status update

3. The four dependency types

Most plans use only finish-to-start, which is why they compress badly under pressure. The other three create the parallelism that buys back a window. Record the type explicitly: a predecessor column holding only IDs forces every reader to assume finish-to-start.

TYPE	WHERE IT APPEARS IN CUTOVER	WATCH FOR
Finish to start	The default, and most load sequencing	Overused, where a lag or overlap would be truthful and would recover hours
Start to start with lag	Validation running behind a long load rather than after it. Comms going out as a freeze begins	The lag must be justified by the owner, not guessed by the planner
Finish to finish	Reconciliation that cannot close until every related object has loaded	Often hides a missing task: the thing that actually closes it
External constraint	Third-party file windows, bank cutoffs, statutory deadlines, vendor support hours	The most dangerous category. Confirm in writing and record the date you confirmed it

THE AUDIT TEST

Federal schedule assessment guidance holds that every activity should have at least one predecessor and at least one successor, with only two exceptions, the start and finish milestones, and that any activity missing that logic must be justified in the schedule documentation. Run that test across your plan. The count of failing lines is your rebuild estimate.

4. Decomposition: what earns its own line

Both six hundred lines and two thousand can be correct. What matters is that the rule is stated on the plan and applied consistently, because an inconsistently decomposed plan cannot be reported on.

RULE	GETS ITS OWN LINE WHEN	STAYS INSIDE A PARENT WHEN
Ownership changes	The next action belongs to a different person or organization	One person does the whole thing without handing off
A gate sits between	There is a decision, approval or acceptance in the middle	It runs start to finish without anyone deciding anything
Duration exceeds the reporting interval	It runs longer than the gap between checkpoint calls	It completes inside one reporting interval
It can fail independently	This part can fail while the rest succeeds, with its own response	Failure of any part means the whole thing failed
It is a validation	Always. Proving the work is separate from doing it	Never. Validation is always its own line, different owner
It is a communication	A message goes to an audience outside the cutover team	The message is internal to the bridge call
Volume drives it	Repeated per object, per integration, per department, per file	The set is small enough to name in one line

5. Contested ownership: three questions, in order

When two organizations both claim the accountable letter on the same row, this is not a stalemate to be mediated. Ask these in order. The first clear answer settles it, and you record which one did.

QUESTION	WHAT IT SETTLES
1. Is it a contracted deliverable?	Partner-side accountable, client-side consulted. Add a paired row for accepting the deliverable, client-side accountable. If acceptance criteria are missing from the agreement, raise it commercially now, not at acceptance.
2. Does it require a business decision?	Client side without exception. A partner cannot be accountable for a client's decision. Partner-side responsible for the analysis. Name the decision date in the plan, not just the decision.

QUESTION	WHAT IT SETTLES
3. Does it need client staff or access?	Client side. Accountability follows control, not capability. Record the dependency explicitly so the partner is not later held to it.
None of the above	Split the row in two with a handoff between them. If it cannot be split, assign client side and escalate the ambiguity as a commercial gap. Never share the letter, never leave it blank.

THE LINE WORTH ARGUING FOR

Equal partners means no partners. Every row where both organizations are marked accountable is a row where the work has no owner and the contract has no evidence. Make the argument in ownership terms, not collaboration terms, and it is usually won in under a minute.

6. The baseline gate

The baseline is a commitment and re-baselining is expensive in credibility. Do not baseline until every box is ticked, and say so publicly rather than baselining a plan you know is incomplete.

Completeness

- ☐ Every track has been through a working session and a join session
- ☐ Business and department tasks are present, not only technical ones
- ☐ Decision points where work crosses organizations are named tasks
- ☐ Validation tasks exist for every load and every configuration change
- ☐ Communications tasks exist for every audience that must be told something
- ☐ A rollback position is stated for every section

Integrity

- ☐ Every line passes the line-level quality bar
- ☐ No owner cell reads as an organization, a team, or "both"
- ☐ Every duration has been confirmed by the person who owns it
- ☐ The plan calculates its own end date from anchors and predecessors
- ☐ Float is a number that can be stated out loud
- ☐ The delta view for a schedule shift has been built and tested

Operability

- ☐ The tool has been used by real owners on real devices in a rehearsal
- ☐ Status rolls up by track, by wave and by owner without manual work
- ☐ The runbook view is derived from the plan, not written separately
- ☐ Gates appear as hard stops with named decision makers
- ☐ Access has been proven for everyone who needs to update a line
- ☐ A version control and change process is agreed and written down

Appendix. The workstream homework pack

Issue this in the cutover forum with a worked example already completed, and require it back before the per-track session. Response quality roughly doubles when people can see what a good answer looks like. Keep it to two pages or it comes back blank.

SECTION	WHAT THE TRACK FILLS IN	WHY IT IS ASKED
Transactions	Every transaction type the track owns, with rough volume	Input to freeze classification. Tracks always list some the program did not know about
Must continue	Which of those cannot stop, and what breaks if they do	Sets the soft freeze scope and identifies the named key users needed
Cutover activities	What the track must do between the last normal day and the first supported day	Raw material for plan lines. Expect it incomplete and use the session to fill it
Dependencies	What the track needs from others, and what others need from it	Cross-track dependencies surface here more reliably than in any central analysis

SECTION	WHAT THE TRACK FILLS IN	WHY IT IS ASKED
People	Who is available during the window, and who is not	Availability is a planning input and it is always worse than assumed
Decisions needed	Anything the track cannot proceed without a decision on	Feeds the decision list. Frequently the most valuable section
Concerns	Anything that worries them about cutover, free text	Often where the best findings come from

The vocabulary to define once, in the forum

TERM	THE DEFINITION TO GIVE	WHAT PEOPLE ASSUME INSTEAD
Cutover	The whole period from the last normal business day to the first supported day	The go-live weekend
Freeze	Restricted transaction entry, in defined states, per domain	One date when the system switches off
Soft freeze	Named users only, named transaction types only, everything else deferred	A polite request to slow down
Blackout	No access at all while extraction and load run	The same thing as the freeze
Catch-up	Entering everything deferred, after go-live, by named people	Something the program handles centrally
Gate	A stop with named criteria and a named decision maker	A status checkpoint
Cutover task	One action, one owner, one predecessor, one validation	A line in their project plan
Rehearsal	A full dress run of the runbook, not only a data load	The mock conversion they have already done

WHERE THIS CAME FROM

Built from fifteen cutover and deployment decks across public sector, healthcare and global enterprise Workday and ERP programs, plus published audit findings and vendor deployment methodology. Fully anonymized. The long-form versions live at 9nation.com under Insights, in the cutover deep dives.

David Apollon · 9Nation Consulting · Independent ERP and Workday program leadership · 9nation.com