

Case Study: Recovering Changeover Hours on a Multi-Line Converting Operation

FirstPass Systems — Production-Grade AI for Manufacturing & Operations. Anthony Ford, founder. Delivered as an internal improvement project at the global manufacturer where he works as a manufacturing engineer — built, deployed, and adopted on a live production floor. Details below are masked to protect the plant's process knowledge: part numbers are fictional, and a proprietary process attribute is shown only as a color code.

The Challenge

Six converting lines ran from a schedule built by hand every morning: a scheduler pulled work orders from the ERP, arranged them in a spreadsheet, and released jobs to the floor once a day. The floor's standing complaint: jobs for the *same part* kept arriving a day apart — the line would finish a run, tear down the setup, and set it right back up tomorrow for an identical order that had been sitting in the ERP the whole time. With changeovers costing roughly an hour each, every unnecessary teardown was an hour of capacity gone.

Nobody was doing anything wrong. The scheduler had no tool that could see, across hundreds of open work orders, which pending jobs matched a setup already running.

The Approach

FirstPass built a sequencing engine that reads the same files the plant already produces every day — the ERP work-order exports and the schedule workbook. No integration project, no new system to learn, nothing installed on the ERP: a folder of exports in, an optimized schedule out.

The engine encodes the floor team's own changeover physics as a setup hierarchy:

Setup change	Relative cost	Handling
COLOR group (masked process attribute)	Highest — the hour	All jobs of one color run as a single block
Material family	Medium	Families run together inside each color block
Slit width	Lowest	Width changes sequence together inside each family

Within every group, the hottest due date runs first, so batching never comes at the expense of the order that has to ship. Flagged priority customers pull their entire setup block to the front — served first without paying a single extra changeover. Every run also projects a finish date for every job against line capacity and marks the ones trending late.

Reliability was engineered in from the first version, production-line style: inputs saved before today trigger a "check your exports" warning naming the stale file; a malformed row is flagged for attention, never silently dropped; job counts are verified so nothing can vanish between input and output. Deployment is a self-configuring one-click launcher on a standard plant PC — it finds Python, installs its own dependency, and runs entirely on the local machine. No cloud, no network calls: plant data never leaves the building.

The floor kept its habits. Operators still work from printed run sheets marked off with a pen — the sheets are just generated now, one per line, with the changeovers announced in advance and a checkbox column where the crew expects it.

The Results

- **Roughly 20% of planned changeover time eliminated** on the first live schedule — nearly six line-hours returned from a single snapshot of the wall, before any process change on the floor.
- **One color changeover per line** in the optimized sequence, where the manual schedule had lines flipping back and forth between setup groups.
- **Eleven pending ERP work orders flagged on day one** that matched setups already running — the exact "same part, released tomorrow" failure the floor had been reporting, now caught automatically every morning.
- **A data-quality finding leadership did not know it had:** dozens of open jobs carried due dates already in the past — surfaced, quantified, and routed to the scheduling conversation instead of silently distorting priorities.
- **Idea to deployed on the plant floor: two days.** The production team was requesting refinements within the first day — the changeover hierarchy in the engine is theirs — and plant leadership installed it on their own machines.

Why This Matters for Your Plant

The engine here is not a scheduling product — it is a pattern: read the messy files your operation already produces, apply the rules your own experts already know, verify everything, and hand back a decision-ready document. The same pattern prices quotes from RFQ documents, stages customer orders for ERP entry, mines downtime logs for repeat offenders, and keeps audit evidence assembled.

It works because it was built the way plants build product: assumptions declared and verified with the floor, failures flagged instead of hidden, and adoption won by fitting the crew's existing paper workflow instead of replacing it. That is what FirstPass means by production-grade.

Next step: a free 30-minute working session. Two or three specific ideas for your operation and a live demo of a working system — no pitch deck, no commitment. If one idea is worth proving, a fixed-scope two-week pilot puts a working tool on your real exports, and you keep it either way.

Anthony Ford · FirstPass Systems · Thorntown, Indiana · anthony.ford@firstpasssystems.com