



Onboarding (Software Setup)

Why This Matters

The book is blunt about this: software issues are rarely caused by the software itself, they come from the setup. "It is not the software. It is leadership." A rushed or shortcut implementation just moves your existing problems into a new system instead of fixing them. This guide walks through what a builder actually needs to do before, during, and after bringing on new software.

Before You Onboard

- ☐ Define your goals and metrics: efficiency, accuracy, and visibility. Have every team member lay out their own process first
- ☐ Audit existing workflows across sales, estimating, purchasing, production, warranty, and accounting. Document how work really flows today, not how it's supposed to flow
- ☐ Identify pain points and bottlenecks before you shop for a fix
- ☐ Build a cross-functional implementation team. Include IT, but don't let IT own the process. Operations should lead
- ☐ Assign a project champion, one internal owner accountable for progress, decisions, and communication
- ☐ Evaluate scalability and vendor track record, not just whether it solves today's problem

Ask This Before You Switch

- ☐ Is the software the actual problem, or is it your SOP? Audit process before blaming the platform
- ☐ Are you using most of the system you already have? If not, is it a training gap, the wrong people, or a poor setup?
- ☐ Do you have the right people to adapt to this? Software exposes weaknesses in process and personnel

Setting Up the System

- ☐ Build a detailed implementation timeline with milestones and dependencies
- ☐ Schedule onboarding and training that respects everyone's actual workload
- ☐ Build consistent data standards: naming conventions, SKUs, and cost codes (use NAHB cost codes if you don't already have your own)
- ☐ Set up a sandbox environment for safe testing before anything goes live



- ☐ Run department-specific training with real-world use cases, not generic demos
- ☐ Document every configuration and key decision so it survives staff turnover

Going Live

- ☐ Run parallel systems to verify accuracy before fully switching over
- ☐ Launch with internal support ready, a designated power user in each department
- ☐ Monitor data integrity and reporting outputs daily
- ☐ Make sure leadership can actually navigate the dashboards and see what matters
- ☐ Gather feedback early and act on quick wins

After Go-Live

- ☐ Schedule ongoing cross-training every quarter so the system doesn't live in one person's head
- ☐ Build department-level SOPs for the new workflows
- ☐ Keep an internal FAQ or help guide with key contacts
- ☐ Run annual audits of system use and data health
- ☐ Revisit your original goals and ask honestly if the software delivered

Practical Truths Worth Remembering

- ☐ Buy for the workflow, not the bells and whistles. A flashy demo means nothing if the field still has to enter data twice
- ☐ Never skip the implementation phase. The system will expose your gaps, not fix them, if you rush setup
- ☐ Keep cost codes simple, consistent, and sortable. A messy code list slows everything down
- ☐ Don't force the field to fit an office-designed workflow. If the field won't use it, you're back to email and sticky notes
- ☐ Watch for "clerk and recorder syndrome," if the system needs a full-time admin just reentering updates, it's costing more than it saves
- ☐ Use reports for insight, not micromanagement. One clear snapshot beats fifty pages of detail
- ☐ Be ready to evolve. Review phases, payables, and approvals every quarter as the company grows