Introduction

Welcome

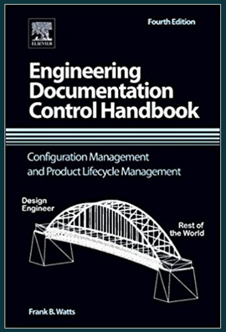
Welcome. This is the Awkward Engineer's Guide to Making at Scale. In short, it's a guide covering much of the hardware related project management skills I learned the so-called "hard way". Skills that were once in the domain of manufacturing engineers are increasingly important for a small team to take on themselves.

Training needs to catch up with accessibility.

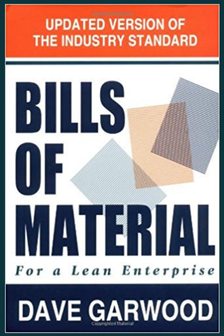
As proof of the importance of these skills, take a look at all the sorts of creative, entrepreneurial individuals who can make one of something, but struggle to make 100's (or more) of something. (Look how many hardware Kickstarter's never ship!)

Best practice knowledge was developed for large companies.

The good news is that there is an entire body of manufacturing engineering knowledge dedicated to ERP/MRP/PLM, or other three letter manufacturing/data management system of your choice. *The challenge is adapting from something that was developed by billion dollar car makers to a small team with a spreadsheet*.


Fourth Edition
Engineering Documentation Control Handbook
Configuration Management and Product Lifecycle Management
Design Engineer
Rest of the World
Frank B. Watts

https://www.amazon.com/Engineering-Documentation-Control-Handbook-Fourth/dp/1455778605/



https://www.amazon.com/Bills-Material-Lean-Enterprise-Garwood/dp/0962111848

Goals

The intent of this document is to share techniques I've found useful for managing small batch hardware product builds, as well as teaching some of the foundational knowledge necessary for using them.

Who is this for?

Tiers of Manufacturing

There are three main "tiers" of manufacturers. This is aimed at tier 1 companies, those that make end products, turning components into assemblies.

Ore into steel	Steel into screws	Screws into assemblies
Tier 3: Processes raw material	Tier 2: Process raw material into components	Tier 1: Components into assemblies

Aimed at transfering to a contract manufacturer (C.M.)

More specifically, this document is aimed at companies in the Tier 1 category who plan on transferring their designs to a contract manufacturer. The techniques in this book focus on developing a Bill of Materials internally for someone else to make.

Driving Philosophy: Computer Aided Designing for Manufacture

In short, model everything

The underlying philosophy behind all the material in this document can be summarized in a sentence or two. "Accurately and completely capturing a product design with *design tools* is essential for the fastest, most efficient generation of good Bills of Materials. Account for *everything* in CAD."

This may sound a little obvious, but in practice, it requires good discipline and knowledge of your design toolchain. At the root, much of the challenge comes from the what I call the "CAD/BOM sync problem".

BOM will be the ultimate source of truth

In short, the problem is that when going to build something, there should only be one source of "truth" for what to build, and that can and should be the BOM. The problem then is that design changes then need to be updated in two places, the BOM *and* the design files.

The sync issue and why manual scrubbing has problems

Some people maintain good BOM discipline by maintaining a spreadsheet with all the pertinent information and laboriously scrubbing it line by line by hand, making sure every piece of data is correct and up to date.

The problem with that approach is that in a startup or development environment, the design changes way too often to have any reasonable expectation of having anything close to error free synchronization between the design files on the BOM.

Automatic BOM generation resolves the sync problem

My philosophy is that if you're going to spend that time and effort scrubbing the BOM, you may as well do it in your CAD environment, which can repeatedly and reliably generate BOMs. For each build, I export a top level Bill of Materials from my top level CAD model. The BOM is the source of specification truth for each build, but it flows from the design files.

The CAD / BOM sync problem is resolved by making updates in one place, in CAD, from which the BoM is automatically "synced" for each batch release.

This is for small, fast teams

This process isn't for everyone everywhere, in particular, heavy, established companies with existing product lines and processes. If you're small, fast, nimble, iterating often, and working on a single product, this is making the most of your design files to prepare for manufacture.

Caveat: A BOM alone is not Planning for Manufacture

Limits of a good BOM

Much of the content in this document is aimed at setting up CAD models, along with a corresponding system of organization for developing and eventually releasing product specifications to a C.M.

While development of a good BOM is essential to being able to ramp production, it actual won't arrive in precise detail until fairly late in the development process. If you can’t plan ahead for what your Cost of Goods (COGs) and other development costs will be, you’ll be in trouble.

Planning for Manufacture

Planning for manufacture also requires a sense of business models and profit margins, development costs and Non-recurring Engineering, a sensitivity to schedule dependencies like design freezes, tooling releases, and regulatory compliance testing.

One (of many) reasons for hardware failures

This explains why a few college students with a detailed list of parts purchase from McMaster might be incredibly successful at building one of something, and yet struggle to scale. There are not-so obvious costs (I won't say hidden), that need to be understood and managed properly.

This document is focused on technical execution

This document focuses more on the technical aspects of BOM development and build management, rather than the business aspects of planning for product development.

Guiding Principles

Recognize the challenges startups face

Startups are constantly iterating, improving, and changing, all on a shoestring budget. They don't have time for inefficient, heavy, version control, PLM, or ERP/MRP systems that slow them down and get in the way.

Still need some control

Nevertheless, I still believe a degree of control is needed over an evolving design in order to be able to transfer the design to a C.M. to achieve scale, and also to understand the costs involved. At a minimum

I believe companies should know

- *What they've built*
- *What the current state of their design is.*

If it’s too hard to use, people won’t use it.

In my experience, engineers are notoriously impatient with drudgery, paperwork, and systems that get in the way of doing actual engineering. In short:

Until you have a full time purchasing manager whose job includes running an MRP system, you should stick to Spreadsheets and Dropbox

Version Control, Release Management, and Change Management become blurred

My system recognizes that in a rapidly iterating development cycle, version control, release management, and change management all become blurred together. Every iteration is a change, a revision, and a release, all at the same time.