CS467655

Improving Electrical Workflows: A Contractor’s Perspective

Tim Vock
Gibson Electric

Learning Objectives

- Establish a framework for organizing drawings and people.
- Learn how to push the power of the Run Schedule.
- Learn how to create drawings automatically.
- Learn how we prefab today and how we will prefab tomorrow.

Description

Most BIM (Building Information Modeling) tools are made for designers or mechanical contractors, leaving electrical construction to blaze trails and share what we’ve learned with one another. Now that we’ve earned the trust of our co-workers to build 3D models, we must step up to an even bigger challenge: efficiently taking coordination models to fabrication. This course consists of lessons learned from past projects, ideas shamelessly stolen from smarter people, and ideas on how to improve in the coming years.

Speaker(s)

Tired of the cubicle life, Tim opted for the tempest that is electrical construction and has been loving it the twelve years since. Tim contributed to many major projects in the Chicago area, including Lurie Children’s Hospital and many showpiece data centers. Working for Gibson Electric, Tim helped push new technologies through the company by implementing and process-testing digital tools. He is always trying to build a better process that supports the long-lasting complicated work without weighing down the smaller-scale projects. Being part of EMCOR allows Tim to contribute on projects for customers across the country, borrowing any better ideas he finds along the way.

Tim Vock
BIM Manager
Gibson Electric & Technology Solutions
An EMCOR Company
Organization

Construction projects are chaos. Our job is to steer that chaos in one direction long enough to complete a safe, quality project in an aggressive schedule. Poor organization is the main culprit when we fail to do so.

Organizing People

*What is the issue and how do we fix it?* Words no one wants to hear from their boss when a job isn't going well.

The issue? Everyone has a different idea of what makes my job successful. The estimator assumed I would put in only the time required to fulfill the contract requirements; the foreman assumed I would have spool drawings for every piece of conduit; the project manager wants me to manage information flow between office and field. Am I supposed to do everything I can with as many hours as it takes? Or keep my hours at a minimum even if it means telling my coworkers and customers that I can't do things for them? Without aligning expectations, I am set up for failure. How do I prevent the problem on Day 1, instead of defending myself and my team on Day 200?

Start with a Coordination Checklist. The idea: if the foreman wants six months’ worth of work from me and there are only 40 BIM hours in the budget, we can resolve that now, with everyone in the room. I’m thinking about changing the name to Coordination Yearly Arrangement, just so I can abbreviate it CYA and everyone would get the idea.

The checklist is a document that must be filled out at the beginning of any large project before we can begin our BIM work to make sure everyone has the same ideas about scope and schedules. Where are the latest files kept? How is everyone alerted when changes come? What submittals do we owe the customer? Are we planning to use prefabrication from the model? How tight is the budget and schedule? Most importantly: Who is responsible for each of these critical tasks?
PART-1: IDENTIFICATION AND START-UP (Internal Kickoff)

Required: estimator, PM, foreman, coordinator, BIM manager

✓ Project Overview – Project scope, customer, schedule, expected challenges for construction (estimator / PM)

✓ Coordination deliverables owed to customer
  a) BIM requirements
  b) List of drawing submittals (signoffs, slab pens, pad drawings, point loads…)
  c) Meeting attendance
  What recourse do we have when customer asks for more? When customer runs coordination poorly, forcing us to go over on hours?

✓ Coordination deliverables for internal use
  a) Install drawings
  b) Takeoffs
  c) Pre-fab
  Do we have enough hours? What happens when expectations change? How are we tracking coordination hours?

✓ Create coordination / sign-off schedule
  a) Earliest dates we could promise at external kickoff meeting
  b) Latest dates we require to make pre-fab / install dates

✓ What other questions / concerns need to be voiced at external kickoff meeting?

✓ Review Document and Information Control Plan (Part 2)

✓ Obtain and review documents/information from the estimating phase that can assist start-up.

✓ Identify any line items or scope that is in the contract but not represented on the drawings or in the specifications. Associated documents that pertain to and added scope need to be marked up to reflect these

Figure 1: Project Coordination Checklist

As we discuss the questions in the document, we make notes on what we agree on. Answers may change, of course. But we have a place to start. Put these notes where ever you keep your project information so everyone can refer to it if they need to know whom to call for questions.

Our Project Coordination Checklist is a high level document for all. It can also be helpful to create drill-down documents, that go into more specifics. Writing down how we get information from the customer and upload it for the field for example, could prove useful in the case that who ever is in charge of the task ever gets sick for a week or even leaves the company.
Organizing Documents

Document control is perhaps the most difficult aspect of organizing a project. Foremen don’t want to dial into the office network to access submittals, but office personnel resisted using the cloud. Each PM and foreman had their own folder structure they liked to use. What ends up happening? We have duplicate files that most of us have a hard time sorting through, none of which has all the updated information.

Start with a template folder structure, company-wide. Every awarded project begins with a copied version of these folders to ensure things are kept in the same locations.

PART-2: DOCUMENTS AND INFORMATION CONTROL

Required: PM, administrator, foreman, coordinator, BIM manager

If set up properly, this next phase can save you time and many headaches throughout your project. Accurate information that is easily accessible by everyone (including field) is key for all projects. Avoid wasting time searching for information and keep the field focused on installing. Coordinators need to be current with all changes happening.

1. Folder structure:

When job is complete, folders 1 and 2 will be downloaded to the S-drive (no duplicates!) so the entire job can be archived.

2. Who maintains the current set of drawings and how? What about submittals (internal and external)? RFIs?

3. Construction Document Issuances and associated tasks
   a) Document Issuances
      i) Individual or Batch slip sheeting (See example C).
      ii) Overlay drawings can be created to identify all changes, not just clouded changes.
      iii) Coordination team can track quantities for pricing while reviewing documents.
      iv) Tracking of time.
   b) RFIs/Submittal Comments

Figure 2: Document Control Drill-Down List

The folder structure should be organized by who will need access to what. In the figure above, you see 1-Contract Documents at the top. Everyone involved in the project will need read access
to the contract documents, from the people running the job to our sub-contractors. The access is the most open – allowing everyone to read them (only select people can edit).

Next is 2-VCD Documents (Virtual Construction Department). More restrictive. The coordination team may access but the sub-contractors will not. We grow increasingly restrictive until the last folder, 5-Financials. This folder is hidden to all but a few people.

The project member in charge of document control can create a sub-folder called “Bulletin 1” within the 1-Contract Documents main folder. When she does so, all the permissions are inherited by the parent folder. If we are consistent, we spend much less time worrying that people can read or write information they aren't supposed to see, or that people won't be able to access the information they need.

Another advantage of this organization is the ability to split it. We sometimes have the top two folders hosted on the cloud to make them easily accessible and distributable. The bottom folders can stay on the company network drive. Eventually all folders are archived together without fear of duplication.

Bluebeam Studio allows me to house all our files, with proper check-in/check-out control, and set-up permissions without leaning on our IT department.

Figure 3: Using Bluebeam Studio for Document Control Permissions
You may prefer an alternative cloud service, but the principles should be the same. Make sure the service stops people from editing the same file and over-writing one another. And that allows you to change permissions at the sub-folder level, and post any file type, such as Navisworks models for field use. Make sure it meets your security requirements. Look into these aspects before making your decision.

Ideally you want an easy permissions matrix. As soon as you drop someone into the “Coordinators” permission group, they inherit all the correct permissions belonging to that group, as I did in the figure above.

Bluebeam organizes our files into a current set, transferring markups and generating a current drawing log.

PlanGrid and ProCore keep your current file set in a similar way and have the advantage of being much friendlier to anyone accessing these files from an iPad. But Bluebeam includes the most powerful markup tools available in our industry.

Our BIM departments are comfortable enough with cloud solutions that we can help our companies organize projects in all the ways above. But what about the organization within our own house – our models? Let's attempt a first step toward that elusive idea we've been wanting for years: tying the Revit model to the one-line.
Power of the Run Schedule

Problem: BIM modelers don’t know electrical design well enough to correct problems in the model.

Our foremen used to go through the one-lines, then mark up a plan view to tell the modeler: put (2) 3” conduits from this gear to that panel. The modeler would do it, but he wouldn’t understand what those conduits were. If a panel location was changed or the one-line updated, only the foreman would be able to recognize the problem, but he isn’t in the model every day, meaning errors are not found until much later.

Wouldn’t it be great if everyone could hover over any raceway in the model and know what it was and where it was supposed to go?

Solution: Use a Run Schedule in Revit to tie design to model.

First, link the one-lines (pdf or CAD is fine) into your Revit model. Then use a Revit family to “tag” each raceway. These families aren’t actually tags though. In my example below, the red callouts look like tags but are actually generic families with generic annotations inside them. Why can’t they actually BE tags? Because we are going to schedule them.

Figure 4: One-Line

Let’s select one of the families (called FROM-TO) and look at it’s properties.
We gave this FROM-TO family all the information we would want to know about the line it represents. The 1 LINE FDR# corresponds to the engineer’s feeder schedule. It has a start and end location, how many of what size conduits, voltage, type, etc. We should be able to get all this information from the one-line and confirming construction intent with the general foreman.

Most importantly, we give each of them an ID, a unique identifier that will help everyone understand each run. Our IDs are in the format of source panel, then a colon, then a number which goes 1,2,3, from left to right on the diagram. This RunID will now be used throughout the project. When a foreman is marking up changes on a plan view, there is no more “Run (2) 3in conduits from here to here”. It’s simply marked “DP2:3”. Feeder schedules, one-lines, home run logs, cable length takeoff reports… Everywhere the same conduit is marked with the same RunID.
We now have consistency between the one-line and the markups. Next we take it to the model itself.

Create a RunID parameter that applies to all conduits, conduit fittings, and conduit runs. Tab-select a run when you model it, and assign it the proper RunID.

Next, we create a schedule for our Generic Model families in Revit.
With this schedule, modelers have access to all the information they need to make helpful filters, take off material, color code... And they now have the ability to quality check their own work and find any discrepancies among drawings. The BIM Manager can jump into the project at any time and quality check or fill in for someone on their team without missing a beat.

I copied my Run Schedule and filtered for only PVC Type, for a more professional underground submittal drawing:
**Drawback:**
- To add another RunID, you have to insert a FROM TO family with the correct properties
- This works great for feeders, but what about home runs and low voltage runs?

**Solution:**
For runs that don’t belong on the one-line, you can use the “key schedule hack” (see evolvelab’s explanation [here](#)). This allows you to just add rows to your schedule and type in the information there.

**Issues with this approach:**
- You could use the key schedule hack instead of using the one-line family schedule from above, but you would not get the visual representation, which acts as a built-in quality check
- While you can sort and group key schedules, you cannot filter them like we did above to separate overhead and underground runs. You also can’t duplicate them, so management can be a little tricky.

---

### RunID Key Schedule

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Start</td>
<td>Finish</td>
<td>Quantity</td>
<td>Description</td>
<td>Type</td>
<td>Material Length</td>
<td>Diameter Length</td>
<td>Conduit Length</td>
<td>System</td>
<td>Notes</td>
<td>Feed</td>
<td>Circuit</td>
</tr>
<tr>
<td>ATE1</td>
<td>ATE-LPI</td>
<td>TX-LPI</td>
<td>1</td>
<td>1/4&quot;</td>
<td>EMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATE2</td>
<td>ATE-LPI</td>
<td>TX-LPI</td>
<td>1</td>
<td>1/4&quot;</td>
<td>EMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATE3</td>
<td>ATE-LPI</td>
<td>TX-LPI</td>
<td>1</td>
<td>1/4&quot;</td>
<td>EMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATE4</td>
<td>ATE-LPI</td>
<td>TX-LPI</td>
<td>1</td>
<td>1/4&quot;</td>
<td>EMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATE5</td>
<td>ATE-LPI</td>
<td>TX-LPI</td>
<td>1</td>
<td>1/4&quot;</td>
<td>EMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATE6</td>
<td>ATE-LPI</td>
<td>TX-LPI</td>
<td>1</td>
<td>1/4&quot;</td>
<td>EMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 10: RunID Key Schedule*

---

**Running take-offs:**

With all your conduit runs organized by RunID, you can create a conduit run schedule in Revit* for accurate take-offs.
Run schedules only work when you use conduit without fittings. When using native Revit, I recommend always using conduit without fittings for this reason.

Using a plug-in to push the Run Schedule further

Problem: Revit doesn’t like to automatically push information through conduit runs. For example, you drew a conduit and labeled it ELP-1:1. Later you go back to cut in a box and draw more of the run. The new conduit does not have the RunID. So you have to tab-select and input it again each time you make changes. What a pain! The only way to get around this in Revit is by using a type parameter. But making a new conduit type for each RunID would take forever.

Solution: Use a plug-in. Many third party plugins will push parameter information through your conduit runs automatically. Examples include: Wireworks, Bendworks, Sysque, and eVolve.
Some people create Dynamo graphs to do this, since Dynamo doesn’t cost extra money and graphs can be easily shared with your team. Creating good graphs requires some visual programming knowledge, though.

At Gibson, we use eVolve (from Applied Software) who announced a partnership with Autodesk and GTP (makers of Wireworks and Stratus) this summer.

This is how eVolve’s interface looks:

![eVolve Conduit Run Schedule](image)

Figure 12: eVolve Conduit Run Schedule

You can type in a new RunID information or import the information from an Excel file.

eVolve pushes RunID information throughout the run, including the Start, Finish, Notes, etc. It also tracks the total length of conduit modeled with that ID. Export this schedule any time to give your project manager updated lengths!

Having a plug-in like this gives you other advantages. There is a quality check that will color code any conduit drawn the wrong size, type, or with incorrect run information.
Pro-tip: If you can get your foreman (or whoever keeps your run schedules up to date) to use an Excel spreadsheet compatible with the headings above, the BIM modelers never have to re-type any information. They simply import the latest from the foreman and run the built-in quality check, just in case the foreman changed anything and forgot to tell you about it.

The program will also color your conduits by system without using filters (yay!), and push material information throughout. Why is pushing material information important? I was tired of looking at Navisworks models like this:
If you push material information through the conduit runs, your Navisworks looks more like this:

Figure 14: See of White

Figure 15: Conduits in Color
And this does NOT rely on anyone to run an appearance profiler in Navisworks!

*Tangent: Can we get color into a Navisworks model without using a plugin?*

Yes. Create a multi-category schedule, filtered to only pick up conduits and conduit fittings. I grouped my schedule by system.

---

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>eE_ConduitRun_System</td>
<td>Conduit Material</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

208/120V Power
480/277V Power  eVolve_Pink
Access Control  eVolve_Cyan
Data            eVolve_Purple
EM Lighting     eVolve_Mustard
Fire Alarm      eVolve_Red
Ground          eVolve_Blue
Low Voltage     eVolve_Blue
Sound Masking   eVolve_Blue

*Figure 16: Conduit Color Schedule*
The schedule includes a project parameter I created called Conduit Material:

![Conduit Material Parameter](image)

**Figure 17: Conduit Material Parameter**

Note that Conduit Material is an instance parameter. Revit won’t let you push material into conduit runs by type.

Now you can click on the Conduit Material next to 208/120V Power in the Color Conduit Schedule and pick from the materials loaded into your project. And when I say “select material” I really mean, “select color”, since that’s all I’m using materials to do in this case.

The downsides to the native Revit method:

- You have to do this any time changes are made to the system before exporting to Navisworks.
- If you are worksharing, you take control of all conduits of that system when you push the material this way. So you may find yourself constantly telling your coworkers to sync unless you push the materials through conduit runs after hours.
Create Drawings Automatically

I haven’t found a great way to auto-generate drawings using native Revit. You can build sheets into your template, but each project will have a different number of floors and different key plans. You can use Dynamo to pull information from your project and create views and sheets. If that interests you, learn Dynamo and play with some graphs that experienced users share to get you started but I won’t be covering Dynamo here.

I automate sheet creation using the eVolve software plugin, and I will explain uses for the functionality. For more detailed how-to instructions, see eVolve’s help page here.

First thing we do is set up Kit Areas. Kit Areas are 3D boxes that work like a cross between Rooms or Spaces and Scope and Section Boxes. I set up a Kit Area for each key plan area. If I have a Kit Area for Level 2 – Area B, it will push that location information into all content inside it. Rooms/spaces do the same thing except they only work with families that have Room Calculation Points. Spaces work great for light fixtures, but not so much for conduits and hangers, which is usually the takeoff information my team needs me to provide.

![Figure 18: Kit Areas](image)
With the Kit Areas created, open the Kit Manager.

![Kit Manager](image)

**Figure 19: Kit Manager**

I check the kit areas for which I want the program to create drawings, then pick the proper title block. In this case, I want it to make a hanger drawing for second floor west area. If I wanted, I could have generated sheets for all areas at once.

The result:
The program created a view of hangers in Level 2 – West per my hanger view template, and put it on a sheet. It also duplicated my hanger schedules and cut lists, filtered these schedules to show only the content that appears in Level 2 – West, and put them on the sheet!

I also have schedules in my projects that don’t go onto the sheet. For example, I have a schedule made to export to our label maker. The Kit Manager created one of these schedules filtered by L02-West, and left it off the sheet, ready to go.

Now all I have to do is dimension and tag my drawing, and export my tag schedule to be printed.
How did the program know how I wanted my hanger drawings to look?

It requires a bit of set up on the front end. But like any good template, you spend the time and energy making it great – once, then use it to save time a thousand times after that.

The bundled eVolve title block gives me a good start. I open it up, do a Save As to create my hanger title block.

This title block has a bunch of windows in it. Click the first window, and enter the View Template to tie it to. In this case, G-HNGR. Stretch the window to the correct size and location on the page. Then click one of the schedule windows (bottom of pic below). In the properties, enter the schedule to be copied. Move this window to where it should appear on the sheet. Do this with as many windows as you want generated. Done!

![Figure 22: Kit Titleblock](image)

Load this title block into your project template and use it any time you need hanger drawings generated for you. Want to take it a step further? Make another copy of the title block for overhead conduit, another for equipment layout, and anything else you want to automate.

**Pro-tip:** Above we discuss making drawings per key plan. But what if I want to create drawings for Electric Room 1, which is partially inside L2-West and partially in L2-East. Can I do that?

Yes! You can create a Kit Area inside another Kit Area and the sheet generation will still work. But a word of caution: content can only have one Kit information pushed through to them at a
time. So if I run the Assign Kit ID for my electric room, all content inside will have that Kit Name associated but no longer the L2-West Kit Name. As long as you run the Assign Kit ID for the area you want to print just before doing so, it works like a charm, allowing you to auto-create all your electrical room drawings. Just plan accordingly!
How we Prefab

Revit does a nice job allowing us to prefab families through the use of parameters and schedules. But it’s terrible at perhaps the biggest model-based prefab opportunity we have: conduit bends. Revit sees an offset as (5) distinct pieces – straight, bend, straight, bend, straight. It’s hard to even dimension to the center of bends in Revit on a straight offset, much less when it’s rolled.

There is a great video on how to spool plumbing runs using native Revit from Synergis Engineering [here](#). Using Synergis’ method, we created conduit bend families to make dimensioning possible, as shown in the rolled offset below.

![Figure 23: Spooled Conduit in Native Revit](#)

**Drawbacks:**

- You have to make these modifications to every conduit bend family
- Adding lengths to make a 10’-0” piece of conduit is still manual
- Changes are difficult to manage
- Requires MANY drawings: spool drawings, assembly drawings, install drawings

**Prefab next steps (Get a plugin!)**

Plug-ins exist that aide the process of getting the correct bend radius, mark values, and even do some auto-dimensioning of spool drawings. These include BendWorks, Sysque, Stratus, and eVolve. My workflow uses eVolve.
eVolve works by combining the (5) pieces into a single family, which allows it to have its geometry driven by schedulable parameters. In practice, this means adding a 30° offset of 14” to my model and I don’t have to make spool drawings at all. Instead, I print a schedule and a legend for my prefab shop.

![Diagram](image)

<table>
<thead>
<tr>
<th>eE_Conduit Bend Offset - Labeled</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Label</strong></td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>Greenlee 881</td>
</tr>
<tr>
<td>4in EMT Conduit</td>
</tr>
<tr>
<td>O-001</td>
</tr>
<tr>
<td>O-002</td>
</tr>
<tr>
<td>O-003</td>
</tr>
<tr>
<td>O-004</td>
</tr>
<tr>
<td>O-005</td>
</tr>
<tr>
<td>O-006</td>
</tr>
<tr>
<td>O-007</td>
</tr>
<tr>
<td>O-008</td>
</tr>
<tr>
<td>O-009</td>
</tr>
</tbody>
</table>

*Figure 24: Hanger Prefab Schedule*

Prefab has all the information to make each bend, knows how many of each to make, and how to tag them. I tag them the same way on my install drawing:
Once I dimension where to place the prefabricated bends, I'm finished.

I've worked on projects with a dozen detailers working overtime for weeks making drawings to go from coordinated model to prefab shop to field installation. We can now do much of that work in a fraction of the time. All drawings for the electrical room below were made and checked in a couple man-days (model and photo overlaid):
Tomorrow's Pre-fab

The better we do with BIM, the more powerful it becomes as a tool, and the more pronounced the problem it has: The entire project depends on a Revit model that only a few of us know how to even query, much less manipulate.

Figure 26: Prefabbed Electric Room

Stakeholders

Figure 27: BIM Bottleneck

The next step of BIM and pre-fab is not drones or machine learning; it's the bottleneck we have right in front of us.
Revit cloud worksharing alleviates part of the bottleneck by allowing us to bring collaborators into our projects at the click of a button. If I need help from a colleague across the country or a trusted resource outside my company, I give them permissions to my model so they can help resolve my issue. If you aren’t using Revit cloud worksharing, please look into it.

We also train some of our more tech-savvy foremen to use Revit. They won’t do any heavy lifting in the model, but they can move light fixtures while sitting in a coordination meeting. They can add annotations and check circuit information. They can create views for ordering wire spools.

We need to also train our estimators. They are expected to take design models from large clients and take off all devices and conduit lengths from the model. With a little knowledge of Revit schedules, they could save themselves days or weeks of time.

The pre-fabrication shop also needs a part. They should be able to change the status of a Revit assembly from “spooled” to “fabbed” to “shipped” to “installed”. The BIM modeler can walk into a coordination meeting knowing what can still be changed and what cannot. Stratus from GTP is one solution creating a way for the shop to push this information into the model without allowing them to mess up anything they shouldn’t.
Conclusion

BIM is at the center of all the technologies changing our industry now and going forward. It can make our break our companies, but we have the ability to thrive if we are prepared.

First, organize. All the fancy Revit tricks in the world won’t mean much unless every member of your company’s team knows what you can do, what you need them to do to make your job possible, and build the trust that you can rely on each other.

Once people are organized, make sure you do the same with the files. Everyone relies on the drawings, submittals, specs, feeder schedules, and panel schedules. The team needs to understand where to get the most updated information and know whom to call if there is an issue. They need the right permissions (view, edit) to do their jobs.

Next, get your BIM models aligned with the schedules and markups. Using a Run Schedule to make sure all information is accurate and mistakes can quickly be found and eliminated.

Automation only makes sense when the whole team can first trust in all the steps above. Break your process down to find repetitive tasks that can be made easier, such as drawing creation. Automation helps not only with speed, but consistency.

Rework is killer in our industry. That’s why we BIM. It’s why we need to eliminate mistakes through organization. Projected labor shortages and shortening of schedules require us to prefabricate better in every project we do. Start somewhere. Find the right solution for day 1. Once comfortable with the initial steps, move on to the more difficult aspects of pre-fab such as conduit bending. Always looking at how to do our best today and wondering what our next steps will be in the next project we begin. These possibilities make our roles so exciting, so enjoy the ride!

I hope you got something from this handout and presentation at Autodesk University 2020. I would love to hear from you if you have questions, or if you have solutions that you could teach me. Or just reach out to stay in touch.