

# A SOFTWARE STORY



**Vornel Walker,**  
**Intergraph CADWorx &**  
**Analysis Solutions Inc.,**  
**USA,** discusses the best  
approach to selecting  
plant software.

**T**here are many valid reasons for buying plant design and engineering software, and most companies probably share similar drivers for doing so:

- Code industry requirements and standards.
- Clients' needs/requirements.
- Reduce project time.
- Reduce mistakes.
- Improve quality.
- Increased efficiencies.
- Improve competitiveness.
- Better leverage personnel skills.
- Increase market share.

None of these are bad reasons, and in fact these, and probably many other reasons, should be considered when looking for the ideal solution. The thing is that complying with this list would more than likely ensure that one does not get the solution that will best meet one's needs. In fact this list is so generic that all it will

make sure of is that what one purchases is no worse than what anyone else has purchased. Every company's needs are unique and it is important that the unique challenges that drive a business are what buyers should focus on.

## 'I was sold a lemon'

Over the years, the author has often been in sales situations where the buyer has gotten it wrong.

However, when the buyer gets it right typically both buyers and customers arrive at a long term relationship that benefits both parties.

But what about the seller, do they not have a responsibility? Can they not also be at fault? Well, yes of course that is often the case, and a substantial factor in customers settling on the wrong solution. Ultimately though 'buyer beware' defines the final responsibility, so it is the buyer that needs to ensure that they get what they really need. But how is that best done?

## Features, features, features

Buyers should not get into the habit of selecting software based purely on features, or worse still the number of features listed. It may be a surprise but most users deploy only a small fraction of what a software solution can deliver. So it would be best for the buyer to establish, as early as possible, what that fraction actually is and if they can also leverage some of the other capabilities that are so often overlooked.

So, do features not matter? Well they do because they are a means to an end. But if buyers want something that is going to hold them in good stead for years to come they ought not look at just features, but on the value of what the tool can do for them.

So how can value be measured? It is all very subjective, right? Well, here is a novel concept, how about using \$, £, €, ¥ to measure value? Return on investment (ROI) calculations are a good way of doing this, but if one really wants to get a good feel of the day to day benefit of a proposed solution, why not start with work processes?

## Workflows

The bottom line is that software, in and of itself, is no cure for an inefficient or inadequate workflow. Plugging a plant design or engineering solution into a flawed work process will not ensure that one will deliver optimal results.

In this process of picking, not only a solution to do a job, but also a multi year partnership, one is much better off formalising work processes and practices before starting to look at software. Work processes are the best way to work out where one is (one's starting point) and to identify where one's best opportunities for improvement are (one's destination).

So if one is reading this and saying, 'Of course all companies should have this!' and most would agree, but the sad fact is that the majority of companies seldom have anything that they can point to that codifies how they work and what they need to do to get their work done.

A simplified workflow for the creation of plant design deliverables is shown in Figure 1.

In plant design and engineering most workflow examples follow something close to this and, truth be told, most people working in engineering firms would be able to come up with something similar and in a relatively short time. But for design and engineering companies or workgroups, who are attempting to define what may be needed from a software solution, it is not what is currently done that should drive the buying decision, but what needs to be done, or produced, that should drive it.

## The start of an optimised workflow

If a 'destination' is truly to be identified then something else needs to be quantified, and it is so simple that it is surprising that more people do not do it. It is the procedure of defining stakeholder requirements and deliverables at each stage of the process by asking:

- What has to be produced?
- Who is the creator of the required deliverable?
- What the creator needs before creating deliverables?
- How is that deliverable created?
- Who it is to be consumer of what is produced?
- What information the consumer needs from what is produced?
- In what format is that info best delivered?

By identifying the above one is trying to define the input that each stakeholder needs to create the deliverables that are required of them by those downstream of them. By doing this for every stakeholder, in a chain of influence, one can begin to see exactly how things measure up at each stage of the process. The idea here is to not only layout a complete chain of requirements and deliverables for immediate areas of responsibility, but to identify how what others require can be best optimised for their use.

## Workflows: Do not go it alone

A few months ago, the author and six others got together to define the workflow of someone doing a form of pipe stress analysis, using Intergraph® CAESER II® that incorporated the inclusion of finite element analysis (FEA) to determine the true stiffness (stress intensification factor (SIF)) of branches using Paulin Research Group® FEATools™. Each of those in the group had piping design, pipe stress analysis, or industry experience, and some over 40 years. If this assignment had been given separately to each individual then each person may have finished the task in 30 - 60 mins.

The surprising thing is that, at the beginning, none of those in attendance agreed on either the stakeholder requirements, or the basic workflow, or even the software workflow. It eventually took five hours to work out the stakeholder's requirements, the workflow without the above software and the workflow including the above software, and how that final workflow could improve a company's efficiencies and quality of output.

The point here is that having the 'brightest person in the room' work all of this out is OK, as long as there are other bright persons in the room that are engaged in validating any assumptions any individual may make. When one thinks about it it makes sense as buying decisions, that may involve millions

of dollars, will be based on what assumptions are established here.

## Creating an optimal workflow

During this process it may quickly become apparent that one is developing a workflow that may not align with current work practices, it may even seem like a fantasy, but that is OK. If one is going to invest in a plant design or engineering software solution, would it not be best if, instead of the navel gazing and picking apart everything that is currently done, one comes up with a list of things that one wishes it to achieve?

For that a workflow needs to be identified that will give the best chance of selecting the right tool. That workflow will likely not be the one that is currently employed, but one that would be an optimal workflow for the workgroup(s). Software will not automatically make up for bad practices, but deploying the right solution can make good practices better, even if those practices are not even part of current work processes!

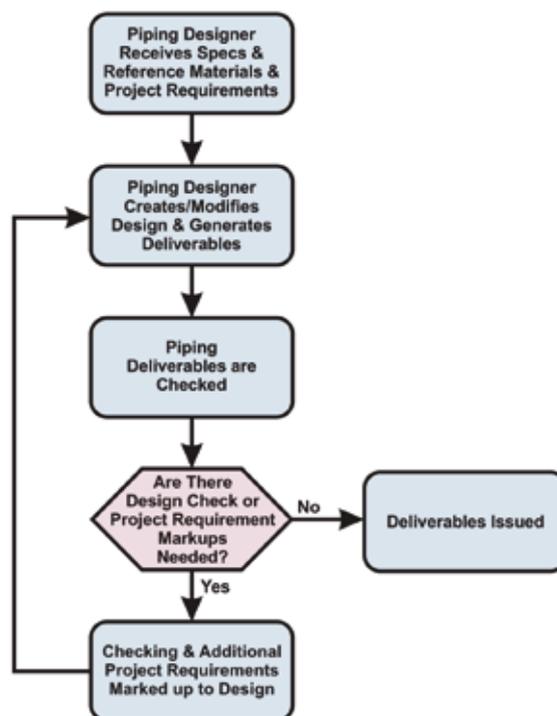
If after the above exercise one discovers that best practices are not currently employed and workflow is less than optimal, one may feel that this method of defining what one wants from a solution may be too risky. If that is so please consider that defining what success will look like will put evaluators well ahead of the majority of people that are looking to optimise their plant design and engineering processes. It is easy to fall into the trap of just wanting things to be better than they currently are, but really is that enough? If that is the case, it really does not matter which solution one settles on, because the truth is that very few solutions are so bad that they would not give you some degree of improved performance.

## Perspective, reality and fear

It will not be a surprise to hear that, given unlimited funds, and time, software can be made to do almost anything that may be required, but the reality is that neither resource is unlimited, and every company needs to determine what they absolutely must have, what would be nice to have, and what they can develop workarounds for and, yes, what they can do without.

The author has gone through many sales cycles, as part of a purchase team, in sales support and also as a salesperson, and one thing that is the worst thing that can happen, at any point of the sales cycle, is when fear, on both sides, sets in. Do not get the author wrong, a good amount of skepticism on the buyer's side is healthy, and concern on the seller's side that they are hitting all of the buyers main points, is also to be expected. Even so, fear on the buyers side often comes from not being sure that they are giving more than they will be getting and that the solution will not measure up to the salesperson's promises.

As fear enters the equation the focus devolves into the mitigation of risk and concentration on features. And with that, gone are thoughts of the great things that the software can do for the buyer and how it can make the buyer's life better. The thing is that the process does not have to be fearful if the buyer can, at any point in the process, be sure that they are getting a good deal, that they will see value in, not only the software, but the long term relationship that the



**Figure 1.** Simplified piping design workflow.

implementation of the tool will entail. So as long as value is assured, and is the continual focus on both sides, fear can be replaced with focus.

## Defining value

If one has read this far one may feel that something is missing, and that this article, that seems to be dismissive of current work practices and workflow, to help determine what a client wants the software to do, is foolish or, at best, just plain naïve. But there is actually great value in establishing current workflow and it plays a large part in calculating the value of implementing an ideal system, and removing the fear of moving forward.

As mentioned earlier, current workflow should not be the main factor in deciding what a client wants to achieve with a software solution. But when used in conjunction with a specific optimised workflow it is a great way to help determine the following:

- Inefficient work practices.
- Where the software can help.
- Where the software cannot help.
- Team skill level.
- Training needs.
- Implementation requirements.

Value is the delta between where one currently is with one's current workflow, practices and tools and when compared with where one wants to be with the solution installed, and with a specific optimised workflow in place. ROI, on the other hand, is the difference between the deployment costs and that value.

So should the cost of the software be ignored? Well no, not at all, it would be deleterious to go into this process

without having some idea as to what it may cost if one decides to move forward. What is being proposed is that one needs to spend the majority of one's time focusing on the benefits, and value that the software can deliver before taking the 'deep dive' into determining the final cost of implementation and final ROI. It is better to look at total cost in light of what one will get for the outlay instead of looking at what you can get for a certain spend.

## Surviving the demo

It is not a mistake that the subject of the software/solution demo has been left to the end of this piece, and that is because that is where it belongs. Often when a software solution is being looked at one of the first things that is requested is a demo, and often based on the 'performance' of the person doing the demo most of the buying decisions are made. This is not the best approach.

In basing a decision on what is known in the industry as a 'dog and pony show' i.e. a circus act, is not the best basis for the selection of a software solution, and especially something as complicated and enterprise impacting as a plant design and engineering solution. Please do not misunderstand, a client will need a demo and it should not only be thorough, but it should help answer questions as to 'Can it deliver?' That means that having someone demonstrate the solution until one sees something that one likes, or even dislikes, is not productive. A solution may be rejected because the demonstrator was just not on that day. The author has been that demonstrator, and is certain because of that misstep the prospect in that case ended up with the wrong solutions.

The better way to use the demo is to have the software company demonstrate that they can do all that has been agreed can be done. In some cases, with more complicated solutions, which may need customisation, have them demonstrate how what is needed can be arrived at, and how that will be done.

Remember, eyes on the prize, the idea is for them to demonstrate the capability of the software to do what is needed, not to catch them out, or for them to provide entertainment.

## Conclusion

There is a lot of good software out there, created by teams of dedicated individuals that passionately care about what they are doing and how they will effect the lives of those that put their trust in what they are creating. Take the time to evaluate as many vendors as possible and also take the opportunity to find out what others are doing and why they have made the decision to select one vendor's offering over another. The reading of success stories can be particularly enlightening as they can often offer insight into an end users situation, the problems that they faced, how they solved those problems and what their pay off, or value was from deploying a solution.

All the above may seem complicated compared to someone coming in and doing a demo, but if the steps above are followed, one will be more assured of success than if more 'traditional' ways of selecting a solution were used. One will know what one wants the software to do and will be less surprised when it does not perform as slickly as when 'that guy did the demo.' 