

[print](#) | [close](#)

# Which Came First: The Innovation Process Or The Software Tool?

[Electronic Design](#)

[Noel Sobelman](#)

Noel Sobelman, Kalypso

Thu, 2014-05-22 12:28

“Do I design improved innovation processes before I implement enterprise software tools, or do I start out implementing software tools that claim to have best practice innovation processes built in?” This is one of the most frequently asked questions I get from companies working on innovation improvement.

Related

[Model-based Design Process Tool Handles Variants](#)

[Trying Out Simulation With Open Virtual Platforms](#)

## The Process Geek Argument

Buying a software system and expecting to improve your innovation performance is akin to buying an exercise bicycle and expecting to lose weight. You need to do the hard work to establish an appropriate development process structure, set up a governance approach with project investment and portfolio decision-making ground rules, ensure reliable data and rigorous data analysis, identify cross-functional interdependencies, and establish high-performance project teams. Without these critical innovation elements working in a synchronized, systematic way, software tools have limited utility.

Too often, companies looking for a quick fix are dazzled by software claims of improved R&D productivity and faster development cycle times. In reality, the roadside is littered with false starts, failed implementations, and seemingly endless spending to “get the software tool right.” And when innovation productivity shows little or no improvement, the software immediately gets the blame, when in reality, the fault was due to a botched implementation.

Attempting to implement software tools before the basic elements of your innovation process are in place will only lead to the ability to execute an inefficient process faster.

Waiting until your basic innovation processes are designed and, more importantly, operating at a consistent, advanced level of maturity allows a company to work out the kinks and show measurable results ahead of the software. As a company’s processes mature and the company grows, software can be layered in as an enabler of those processes. At that time, the benefits to enterprise software solutions become more obvious.

## The Software Tool Geek Argument

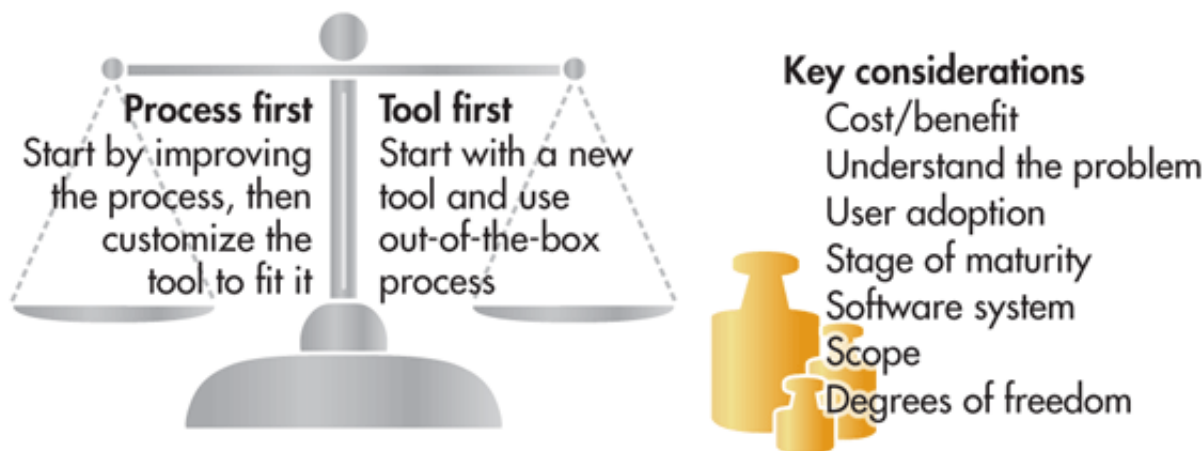
The innovation process benefits greatly from the implementation of software tools. These tools can accelerate development activities, drive timely decision making, improve data analysis, improve communications across functions and business units, and enable seamless collaboration with development partners.

Waiting to design and implement an innovation process before implementing software tools will delay the benefits of the tools while increasing the likelihood of extensive tool configuration and customization. The tool configuration will need to be adapted to the existing process design, extending time to value while adding significant tool implementation cost.

You are better off going with the pre-configured “out-of-the-box” solution (workflows, document templates, data models, etc.) and using the software implementation as a catalyst for making necessary process changes along the way. Most software solutions are configurable and provide flexibility to tailor the tool to specific process needs. But if tool customization is required, vendor support may be compromised and future upgrades may be unnecessarily complex.

### Which Approach Is Right For My Company?

The answer to the process-tool design question is not as black and white as these arguments imply. In fact, it’s helpful to think about a spectrum of possible approaches (*Fig. 1*). At one end, you design and pilot innovation process elements and wait until you reach a comfortable level of process maturity and discipline before starting to enable those processes with enterprise software tools. At the other end, you use the software tool out-of-the-box and modify the innovation processes to accommodate the tool.



Chances are, the best approach for your company lies somewhere in between. Approaches closer to the “process first” end may start implementing a configurable software solution part of the way along the process maturity journey. Those closer to the “tool first” end heavily leverage out-of-the-box tool work flows, maturing processes and software together, over time, using more of a parallel path approach. Developers should take stock of seven considerations before deciding on an approach.

#### 1. Cost/Benefit

Leading companies view ongoing operational improvement as a normal course of business and a sustainable source of competitive advantage. They have built continuous process improvement into the fabric of their operations. But when it’s time to enable their business processes with additional software tools, they ask how much it will cost, why now, and what the benefits will be.

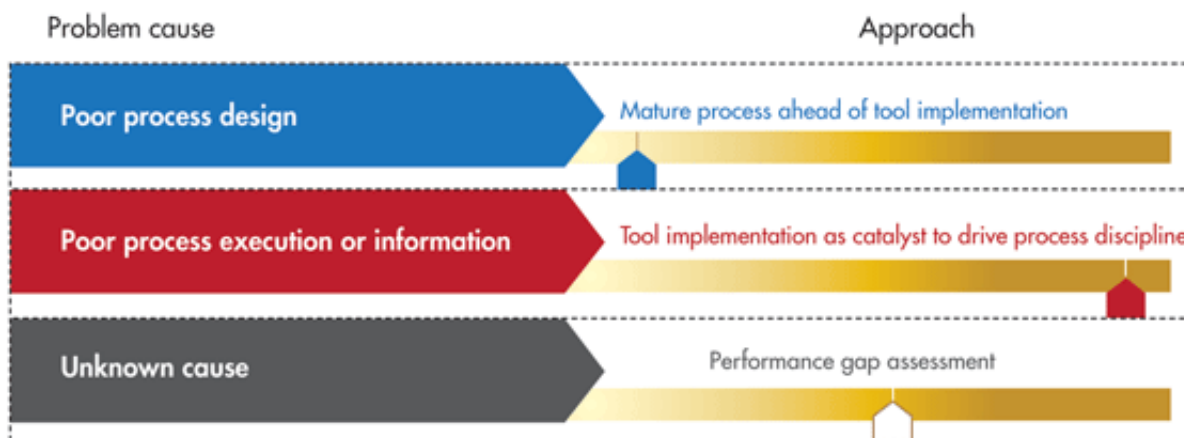
CFOs and IT leaders with many projects and scant resources demand a return on investment, which the initiative champion strives to produce. The fun really starts when business model assumptions are challenged with questions like, “How do I know how much of those assumed benefits are coming from the software tool versus the process improvements we would have gained anyway?”

How does one decouple the process and tool benefits when one enables the other? When faced with this stalemate, companies tend to focus on the process side, eking out incremental process improvements. However, there will come a point when a company outgrows its ability to gain significant operational performance improvement from process improvement alone. At this point, the benefit from a software investment becomes more obvious and easier to quantify.

In the past, the relatively high cost of enterprise software systems and long, drawn-out system implementations caused companies to lean more heavily on the process end of the spectrum. Yet as software moves to cloud-based, software-as-a-service (SAAS) models, the startup cost for on-demand versions gets significantly lower. As the software cost barrier trends lower, there will be less financial risk in designing both process and software together.

## 2. Understand The Problem

One way to figure out where you should be along the process-tool spectrum is to ask yourself if the problem you are facing is being caused by a poor process design, poor process execution, or poor information needed to make good decisions (Fig. 2).



If the answer is poor process design, a new software system is not necessarily the answer. New systems that model a flawed process will not solve the problem. Correcting or improving the flawed process requires analysis (where and why you are performing poorly) and action (change in process or improved process discipline).

If the answer is poor process execution or poor information, consider implementing something to give key stakeholders more accurate, meaningful, and faster information to make decisions or to respond more quickly. This is where software systems excel by providing the right information to the user, manager, or executive at the right time, helping them make faster and better portfolio and project decisions. Adverse event reporting in the medical device world and ingredient traceability in the food & beverage industry are two examples where fast, reliable information is critical.

If you don't know the answer, it may be time to step back and thoroughly assess your process, comparing current practices to industry leaders and identifying performance gaps and root causes. Once everyone agrees on a prioritized list of immediate and longer-term improvement opportunities, you can ask the original question again and use the answer as an

input to an innovation improvement plan.

### 3. User Adoption

Pay close attention your organization's readiness and ability to adopt a software tool. Implementing software too early in your innovation process maturity lifecycle can backfire if user adoption is not considered early and often. We have all heard the horror stories of failed enterprise software implementations. Companies spend millions of dollars buying and implementing software only to have a fraction of the functionality take hold.

Be prepared for a revolt if you force innovation stakeholders to change their current state process to accommodate an off-the-shelf software solution without involving them in every step of the decision process from system selection to system design and rollout. When you do implement it, make sure to understand the gaps between the desired "future state" process design and tool functionality. Work with key stakeholders to gain buy-in as you evaluate tool configuration and process design tradeoffs. Without a well-executed change management plan, the software will be the first thing to get blamed if something goes wrong.

### 4. Stage Of Maturity

All companies vary in their stage of innovation maturity. Early-stage companies may benefit by patterning their business process after those already defined and embedded in a software system. On the surface, this may seem like a "fast track" approach to bypass a long, drawn-out process improvement initiative and jump right to a software solution with leading practices built in. Those considering this path need to be prepared.

First, make sure the basic architecture of your innovation process is agreed upon and functioning ahead of software implementation. In addition, be sure to drive buy-in and adoption of the pre-configured process workflows, templates, and other tool features. Regardless of the amount of pre-configuration available, you will most likely need to tailor the software to the unique wants and needs of key stakeholders across functions and business units and tailor the process design for your particular environment.

Driving stakeholder alignment is not easy. It often requires strong facilitation from someone who understands both leading practice processes and the inner workings of the software. This person will need to drive agreement every step of the way and on every element of the tool, all the way down to the look and feel of the user interface.

Most large companies that have been around for a while typically have some level of innovation process already in place, and they face a different set of challenges. Altering well-ingrained processes to match the out-of-the-box software may not be practical and can negatively impact adoption.

Some would argue that early-stage companies have less to gain from the benefits that enterprise software solutions provide. Early-stage companies are not as large. They also have less data to analyze and far fewer, less complex cross-functional communication linkages to manage. Often, they can get by just fine with Microsoft Office tools like Excel, PowerPoint, and Word and using document sharing solutions like SharePoint.

But at some point in its life cycle, a company will outgrow the capabilities of these ubiquitous tools. There is a point where managing complex, data-heavy business processes on disconnected spreadsheets or manually routing Word documents becomes inefficient.

"We had a phase gate process in place for years and more recently started implementing improved Excel-based portfolio management processes in a few business units. The early excellent results we saw from those portfolio management efforts

created pull for change,” said Cor Bosselaar, Kimberly-Clark’s director of global innovation capabilities and processes.

“Once it became clear that the ‘Excel hell’ did not work, as people found out the hard way, we gained some traction to look at an enterprise tool. At the same time we also started our journey to become a global company and it became clear that the ‘hell’ would only get worse if we continued with Excel,” Bosselaar explained.

## **5. Software System**

It is critical to carefully map current and future state innovation processes to the capabilities in the software solution you are considering. Ideally, software implementations should leverage out-of-the-box functionality with minimal configuration changes to avoid the need for customization.

Choosing a software solution that closely replicates your process without the need for excess configuration and customization is tricky. Anything is possible in software with enough time and money. The trick is finding out how much configuration and customization complexity is required to enable your business process and balancing that with your must-have and nice-to-have process design elements. Start simple and layer in advanced software functionality as your process matures.

## **6. Scope**

Software tools range from narrowly focused point solutions designed to enable one element of your innovation methodology (e.g., requirements management, idea management, document management) to broad enterprise-wide tools with modules that cover a broad spectrum of innovation capability. Some product lifecycle management (PLM) solutions have functionality that supports everything from ideation through product launch, including idea management, portfolio management, resource management, requirements management, document management, configuration management, supplier collaboration, parts management, CAD data management, and more.

When your initiative scope is broad, waiting to mature all or even most of your individual process elements before enabling them with a broad-based software solution will only delay time to value. You also risk initiative fatigue when using a “boil the ocean” approach.

An incremental approach to enabling process with software tools allows you to mature your process first, one element at a time, enable the process element with a properly scoped solution, demonstrate measurable results, and demonstrate a “win” before moving on to the next critical process element. Done well, the immediate business improvement payback can help justify or even pay for the next step in your transformation journey. Even the most conservative CFO will find it hard to turn down a “self-funding” initiative.

When using an incremental approach to build your innovation capabilities, though, be careful to avoid ending up with multiple, disconnected point software solutions. There is tremendous value in broad-based enterprise solutions that integrate interdependent innovation elements. Apply the integration value test every step of the way and keep an eye on longer-term system architecture objectives as you make decisions on point solution versus broad-based enterprise software tools.

## **7. Degrees Of Freedom**

Starting with process improvement allows more creativity around the best resolutions for innovation issues. After all, tool implementation is simply an effort to automate some agreed-upon process. If the tool’s capability dictates the process, you are not necessarily optimizing your solution from a value creation perspective. The limitations of the tool should not limit

process design thinking.

Tool limitations can be overcome, but this can be a long, drawn-out, and expensive endeavor. It's easier and less expensive to change a process than to change a tool. A better way to hedge the risk of heavy tool modification is to have someone familiar with the tool and its capabilities participate in the initial process design to make sure key process design elements are possible within the software environment being considered. This will provide limited value if you haven't selected the tool in advance of the process design session, though.

So, can you effectively select the tool without the process determined? One way to resolve this "chicken and egg" dilemma is to have a good upfront awareness of the kinds of tools and tool capabilities available while developing your process. Design the process architecture with business objectives in mind, but optimize detailed activities with specific tool capabilities in mind.

### **Leading Practices For A Successful Software Tool Implementation**

- Leverage out-of-the-box functionality with minimal configuration where possible, and avoid software customization.
- Involve tool and process experts early on in process design. Otherwise, there is a high risk for process rework or extensive tool customization.
- Design with the "end in mind" and map out your process-tool journey. Understand the overall goals and business objectives.
- Pilot new tool functionality first and adjust your next steps as you learn.
- Capture baseline metrics to measure and communicate the success of your implementation.
- Use a phased approach allowing for real-time feedback on process and tool design and demonstrated results that create "pull" for change.
- Recognize this is a journey. Get started and get better. Layer in advanced capabilities over time.

### **Getting Started**

Innovation process design, pilot, and rollout initiatives can take months to see initial benefits, years to see step function performance improvement, and multiple years to become institutionalized as world class. You will need to decide when along your overall improvement journey (timing), how much (scope), and which processes (highest impact) to enable with software tools as you travel down the innovation improvement path.

Regardless of where along the process-software spectrum you choose to begin your journey, it is important that you understand both the process and tool sides of the equation, and if you don't, get help. To achieve real results from your initiative, look past software system components and take a more strategic approach.

Being strategic does not mean attempting to "boil the ocean" or driving the best new tool that has been mandated from above. Marry a comprehensive knowledge of your desired future-state process, people, and culture to ensure that business objectives are met and the software is adopted as an enabler for improved business performance.

Create an innovation improvement roadmap that maps both process and tool pathways toward your desired future-state. But don't stop there. Identify critical milestones where software can enable the next level of performance. The most

successful companies define a transformation vision and take an incremental approach to implementing software as an enabler for their journey with an unrelenting focus on business results.

*Noel Sobelman leads Kalypso's New Product Development practice. He has assisted clients in transforming their innovation capabilities to deliver lasting results. Noel has worked extensively in the areas of innovation strategy, product development, portfolio management, product commercialization, and the software systems that enable innovation. His industry background includes experience with high technology, life sciences, consumer packaged goods, industrial, and renewable energy companies. He is a frequent speaker, researcher, and writer on innovation effectiveness, disruptive innovation, and time-to-market reduction. He can be reached at [noel.sobelman@kalypso.com](mailto:noel.sobelman@kalypso.com).*

**Source URL:** <http://electronicdesign.com/embedded/which-came-first-innovation-process-or-software-tool>