SMART, RAPID SOLUTION DESIGN

Guidance and Templates to Fuel Your Next Software Implementation

An E-Book Presented by

DAN PLATO

Onspring

E-BOOK CONTENTS

START WITH SMART DESIGN

When it comes to technology, we once asked ourselves, “Can we do that?” Now, the question has evolved to “***should*** we do that?” There seems to be a technological solution for every problem, but tech is not a magic elixir. Automation will not cure all that ails us—though it certainly makes life easier when employed in a thoughtful way.

Thoughtful. That’s the operative word. Human thought, analysis and planning are crucial elements for any software implementation. When we get our hands on a new piece of technology, we often want to start tinkering right away. And while we may enjoy an immediate sense of gratification, our “just do it” approach can bite us down the road. It may frustrate our natural desire to solve problems as quickly as possible, but taking a few days or weeks to thoughtfully plan our use of new software can dramatically improve the long-term outcomes.

In this E-Book, we’ll walk you through a smart and rapid design process, offering practical tips and templates that you’re welcome to use. We don’t claim to provide a one-size-fits-all method, but we hope you gain some actionable ideas that fuel your next software implementation project.

A BIT OF BACKGROUND

Before we dive into the design process, we’d like to provide some background and context. First, let’s start with introductions:

Dan Plato is a governance, risk management and compliance (GRC) consultant who spent 14 years with Ernst & Young, helping companies to improve their internal control processes. Today, Dan is an independent consultant who works directly with organizational leadership to define their purpose and make intentional decisions that drive practical change.

Sarah Nord is the director of learning services for Onspring, a software company that delivers flexible, cloud-based GRC solutions. Onspring is guided by a singular mission: To empower business people to innovate and solve problems for themselves. Over the years, the Onspring team has led solution implementations for organizations across multiple industries, from shipping, energy and retail to healthcare, financial services and manufacturing.

The guidance you’ll find in this E-Book is not specific to any one technology. Whether you’re implementing HR software, project management technology or a new internal audit solution, the fundamentals of the design process are largely the same.

However, it's important to know that we approach design with a **Process Automation Platform** mindset. In other words, we're talking about technology that can be configured to your needs (without custom code) to support data collection, workflow, collaboration, analysis and reporting. These platforms can be used to solve many business challenges and, as such, have earned a reputation for "doing anything." But just because you ***can*** do anything, doesn't mean you should. When you're dealing with powerful, flexible technology, a thoughtful design process is crucial.

With that context in mind, let's get started!

THE DESIGN PROCESS

When you bring up the concept of a design process, you may be met with groans and eye rolling. After all, many of us have gone through the long and tedious exercise of gathering input and documenting requirements for a project. You'll find plenty of methodologies out there and no shortage of ways to burn months (or years) designing a solution before you ever see the fruits of your labor. That's not what we're talking about.

In our experience, a slow, tedious design process can be a momentum killer. Instead, we focus on rapid iteration and focused effort from individuals who are deeply invested in the outcome. The goal of the design process is to arrive at a **sustainable solution**, not a perfect solution. If you aim for perfection, you may never get there. But if you invest your effort in designing a solution that fits today and can scale with you tomorrow, your time will be well spent.

When Should You Design?

Ideally, the design process should begin before you select your technology. If you can begin your vendor evaluation with requirements, process flows and standardized data in hand, the likelihood of choosing the right platform for your needs is greatly improved.

But let's face it: Reality is rarely ideal. You may be fitting your process into a platform your organization has already licensed. Or you may have budgetary constraints or deadlines that motivate you to select your technology before the design process is complete.

Whatever your situation, we recommend that you give yourself permission to pump the breaks. Take a few days or weeks to gather engaged stakeholders, discuss your goals and needs, and map out your desired outcome.

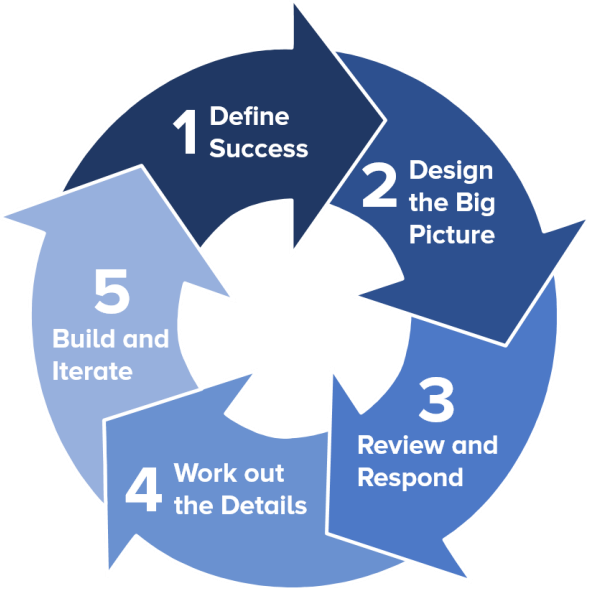
Who Should Be Involved?

Notice that we call out ***engaged*** stakeholders very specifically. You do not need everyone in the room for your design sessions. You need the right people in the room, and that's typically 1–2 individuals from any business function with a vested interest in the project.

Members of the design team should be "in the trenches," so to speak. They should be interested in the process, open to change, and committed to sharing ideas and providing feedback. If you have people in the room who are constantly checking messages and stepping out for other meetings, they can bring down the mood of the entire room. Don't invite them. Focus instead on the people who have valuable knowledge to share and who want to be part of the process.

What Does the Design Process Look Like?

We approach solution design as a 5-step process, described below. Note that steps 1–4 occur on paper. By "on paper," we mean whiteboards, notebooks and basic tools like Word, Excel and Visio. Nothing fancy.



If you've already selected your technology, you can begin your design session with a brief demo to show people the "art of the possible." Then get out of the tech and return to paper. If you spend too much time exploring the technology, you can easily become distracted by all the available features. You may not even need all those features—at least not right away. By keeping your design process on paper, you can be practical and focus on what you really need the technology to do for you.

1. Define Success

Renowned author and speaker Simon Sinek published a book called *Start with Why* in which he argues that great leaders and organizations don't tell you **what** they do. They tell you **why** they do it, and they invite you to be part of the outcome they are working to achieve.

The same is true in the solution design process. When you assemble your team for the first time, don't merely focus on what you want to accomplish (i.e., replacing a legacy system, automating a labor-intensive process, etc.) Remember to ask why. For example:

> ***Why is your current system or process not working?***
> ***Why do you need to introduce automation?***
> ***Why do you need more visibility or collaboration?***
> ***Why should you invest time and energy to do things differently?***

This is very different from asking, "What do we want this software to do?" The software can do lots of things, but why do you need it to do those things? The answer may be, "you don't." Your platform may have many great features that you're eager to implement, but if you focus on ***why*** before ***what***, you can ensure that you're implementing the right features for your needs.

When defining success, it's also important to take a long-term view. You may have a pain point that you want to solve as quickly as possible—but be wary of band-aid solutions. You need an implementation that's sustainable over time with minimal rework. Think in terms of long-term, measurable goals.

For example:

- We want to reduce the issue response time by 25%.
- We want to standardize our control library and eliminate duplicate controls.
- We want our Legal and Vendor teams processing contracts out of the same solution so business owners have a real-time view of contract status.

When you start the design process with a clear picture of success, you can keep your team engaged and focused on the things that matter.

2. Design the Big Picture

Once you've defined your goals, you can begin to sketch out the solution design, first with broad brushstrokes, then with finer detail. Whenever possible, we recommend that you start the design process with basic process flows in hand. It's much easier for people to react to something they can see than to start with a blank sheet of paper. But one note of caution: Be sure participants understand that your initial process flows are not set in stone. They're simply a starting point for discussion.

So, what you should you produce in your design sessions? We recommend the following:

Design Notes: As design participants discuss, make suggestions, negotiate and come to agreement on how your solution should function, be sure to take detailed notes. It's easy to forget decisions or open questions as soon as you leave the room, but your design notes will help to keep everyone aligned. See **Appendix A: Design Session Summary** for a template.

Detailed Process Flows: If you started your design sessions with basic process flows in hand, great! But be prepared for lots of changes and refinements. You need to consider as many of the if-then scenarios as possible. For example:

- If we make minor changes to a control that has already been approved, does it need to go through the entire review process again? And how do we define "minor"?
- If a business owner submits a contract for a vendor that has previously been terminated, how should we handle that?
- If only one sub-process has a high-risk rating, should we automatically treat the parent process as high-risk?
- If we need to escalate the response to an issue, what should that workflow look like?

You may not design all possible scenarios into your solution if they are rare occurrences, but you should note them and make a conscious decision to "handle these cases offline."

During your design sessions, you can sketch out your process flows on the whiteboard for the sake of speed and visibility. Then afterward, you can digitize them using MS Visio, PowerPoint, Draw.io (a free app for Google Docs) or your diagramming tool of choice. See **Appendix B: Process Flow Design** for an example.

Business Requirements: When documenting business requirements, you need to define how the solution should address ***specific*** needs or challenges. For example:

Req. ID	App	Description	Complexity
Risk-01	Risk Register	The Risk Register must enable references to strategic objectives, processes, related risks and supporting controls. When entering a new risk in the risk register, the user must be able to keyword search into these related apps and select appropriate content.	Easy
Risk-02	Risk Register	When new risks are documented, they should enter a work queue for the Enterprise Risk Management team. All risks must be reviewed and approved by ERM before they are published in the Risk Register.	Moderate

Your business requirements are an itemized list of solution behaviors or capabilities. These requirements, coupled with your process flows, enable you to get all design participants on the same page about the expected outcomes of the project.

As we mentioned earlier, it's best to define requirements before selecting your technology. But if that's not the case, keep moving forward. It's simply important to capture your requirements before exiting the "Big Picture" phase. You'll reference them throughout the design process, particularly when it comes to testing.

See **Appendix C: Business Requirements** for a template.

3. Review and Respond

Depending on the complexity of your software implementation, you may need multiple design sessions to capture all necessary requirements and process flows. At the conclusion of each session, we recommend that you send the deliverables to all participants as quickly as possible. If you can send them out within 48 hours, that's ideal.

Why the hurry? If you wait a week or two to initiate the review process, or if you wait until all design sessions are complete, you'll hear a lot of, "I don't remember that" or "When did we

decide this?" By contrast, if you send the deliverables quickly, the conversations will be fresh in your participants' minds. A rapid turnaround helps to maintain alignment and momentum.

You should also require a quick turnaround on the review process. Give participants 3 business days (max) to respond with comments or questions. Again, this helps to keep the design process moving forward.

4. Work Out the Details

Once you reach agreement on business requirements and process flows within your design team (and from higher management, if necessary), it's time to go a level deeper and define your configuration requirements. This is where you connect your business requirements directly to the capabilities of your chosen technology.

In this phase, you need to define:

- Data capture requirements (field name, description, type, values, required/optional, etc.)
- Reporting requirements (data to include, text or chart presentation, filters, etc.)
- Messaging requirements (email alerts, intended audience, message, trigger event, etc.)
- Security requirements (user types, data to view, data to edit, etc.)

Here's the good news: You don't have to do all this work yourself! You can provide design participants with a configuration template, and they can fill in the details for a specific set of business requirements. This is a great way to keep people engaged and improve their familiarity with the technology. They don't need to be logged in to document configuration requirements, but they should understand the basic capabilities of your platform so they know what's possible.

If you do "farm out" the configuration design work, be sure to have a central point of review. You'll want to ensure consistency and weed out any points of confusion before moving forward to the next phase of work. See **Appendix D: Configuration Requirements** for a template.

5. Build and Iterate

This is where the rubber meets the road. You've done the heavy lifting in the design process, and now it's time to see your design come to life. Whether you're configuring the solution in-house or working with consultants, you can hand over a design workbook with process flows and requirements, and they will be well-equipped to get to work.

This is not to say that the build process will be completely smooth sailing. You may discover that some of your design decisions don't work well in practice. It's OK to make design refinements as you go, but before you change directions, go back to your design notes and recall why you made a particular decision in the first place. You may find that small adjustments get you to your goal.

While configuration work is in process, we recommend giving your design team occasional "sneak previews." Even though it can be intimidating to show a work in progress, there's really no need to wait until all the build work is complete before you show your team the fruits of their labor. Depending on their level of familiarity with the technology, you can even allow them to log in and start exploring with the caveat that the solution is still a work in progress. Whether you provide demos or hands-on access really depends on your team.

When the build work is complete, you can engage your design participants again to test the solution, and here's where your business and configuration requirements come in very handy. You don't want to test the solution just to test it. You want to test for specific behaviors and outcomes, which are clearly defined in your requirements. Use those requirements as a checklist, and verify that every item is configured with the level of precision you need.

POTENTIAL OBSTACLES

Going through a smart, rapid solution design process can prevent a multitude of mishaps when implementing new technology. However, you should be prepared for a few bumps in the road along the way. That's just the nature of this type of work.

Here are a few obstacles you may encounter:

Data Inconsistencies: If you're gathering information from disparate systems or multiple lines of business, inconsistencies are bound to surface. For example, you may have a Category field that's required in your solution design, but when you start gathering data from across the organization, you find that only a subset includes a Category value.

Data gathering and cleanup can be one of the most time-intensive parts of the design process. It's best to prepare for it early on. As soon as you have your field requirements documented, go ahead and start collecting data so you can see where you stand. Data collection should happen before or during the Build and Iterate phase. This will give you time to standardize your data or adjust your solution to accommodate non-standard data.

Short Memories: It never fails. You get your design team aligned on goals, process flows and requirements, and you move forward with the build phase. Then memories start to grow fuzzy, and people begin to forget design decisions or the reasons behind them. They may start to question those decisions and slow down the process.

To prevent or resolve "design amnesia," we recommend the following strategies:

- When you deliver process flows and requirements for review during the design process, ask participants to acknowledge that they have seen them. Then if questions

come up later, you can point them back to the design documents and say, “See, this is what we decided together. This is how we agreed that this should work.”

- Recognize that making some changes during the build phase is normal. Your design documents should get you about 80% of the way to your desired outcome, and then you’ll tweak and refine during the build process to get that last 20%. You don’t need to be completely rigid with the design documents, but be wary of making changes for the sake of change.
- If any significant change is needed during the build phase, get back in the design room with your team. Talk it through and come to consensus as rapidly as possible. This will give people a renewed sense of involvement and ownership.

SUSTAINING THE SOLUTION

Circling back to the beginning of the design process, we focused on defining success in terms of long-term, measurable goals. It would be easy to document those goals and stash them away, but we encourage you to keep them front and center. Make your goals the focal point of ongoing discussions with project stakeholders. This is a great way to keep people focused on sustaining and improving the solution rather than just stamping it complete and moving on.

Here are a few additional strategies you may consider for sustaining your solution:

Value Scorecard: Take your definitions of success and determine specific measures for tracking progress. On a quarterly basis, present those success measures to your design team on a simple 1-page slide. This allows the team to see how the new solution is performing, based on the goals you collectively defined. The value scorecard also keeps people in the mindset of, “We’re not finished. We’re going to continuously improve.”

Here’s an example:

What Is Success?	Measures	Status
Reducing effort (real hours) associated with managing controls, testing and issue management	15 – 20% reduction in effort to manage controls and issues	On Target
	10% reduction in the number of key controls	On Target
Issue and control owners better understand their responsibilities and are more engaged	75% response rate in control owner certification	Making Progress
	Less than 15% of issues are past due	Corrective Action Needed

Stakeholder Surveys: Also on a quarterly basis, send a survey to your design team, asking for their feedback and enhancement ideas. (If your platform offers built-in survey functionality, why not use it?) We recommend that you survey those individuals who had a hand in the design process. This is a great way to gather new ideas and identify issues that may be very easily corrected.

User Group Sessions: When rolling out new solution enhancements, give people an opportunity to join a live session to preview what's coming and provide their feedback. You can extend these sessions to anyone who is actively using the solution. Again, this is a great way to keep people engaged and gather insights from those who work with the technology day-to-day.

Solution Champions: Over time, your stakeholders will change. New people will bring fresh ideas and different perspectives. Because your team will evolve, it's essential to have a role for one or more "solution champions" who will be responsible for enhancements. These are the people who will continue to gather feedback, vet new ideas, and spearhead future design and implementation processes.

It may take some trial and error to determine the right methods for sustaining your solution and keeping users engaged over time. The point is this: Never stop measuring and improving. Keep your design goals on the front burner, and maintain dialog with your users. Based on our experience, we believe you'll be very happy with the results.

THE GOOD STUFF: TEMPLATES AND SAMPLES

Throughout this E-Book, we've referenced templates and samples that you're welcome to use in your own design process. If you'd prefer Excel versions of the Business and Configuration Requirements templates shown below, please contact us at dan@danplato.com or sarah@onspring.com.

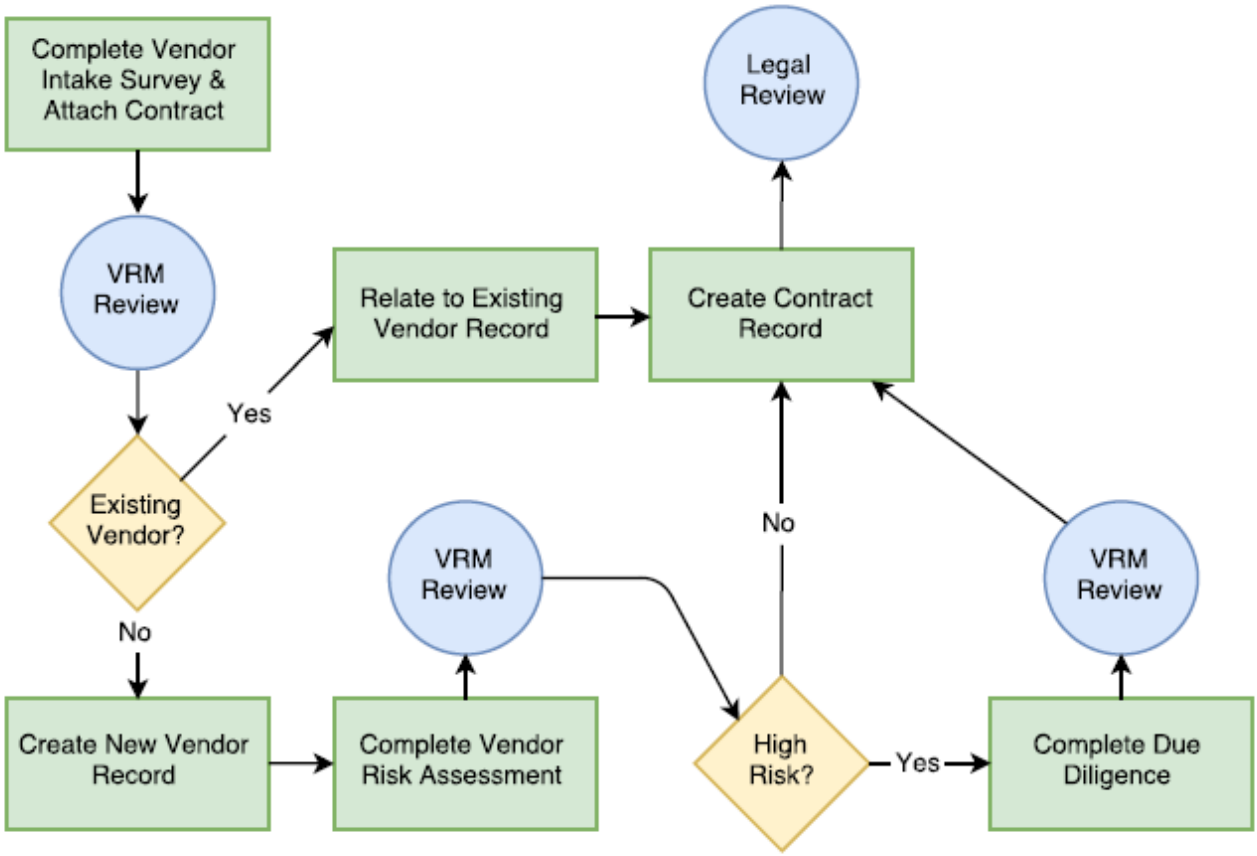
Appendix A: Design Session Summary

The template below enables you to summarize the decisions made in your design sessions. These decisions will help you define specific requirements and action items to complete prior to configuration.

[Process Name] Design Session
Date/Time:
Attendees:
Action Items and Owners: • ___ • ___ • ___
Notes and Design Decisions: • ___ • ___ • ___
Open Questions: • ___ • ___ • ___

Appendix B: Process Flow Design

This process flow is for example purposes only. We've included it in this E-Book to illustrate the level of detail you might want to include in your own process flows. Note the if-then statements, shown in the yellow diamonds. These indicate points in the process where the path can diverge, depending on the data provided.



Appendix C: Business Requirements

The template below will help you define how the solution should address specific business needs or challenges.

Req. ID	App	Description	Complexity
Unique identifier	*Name of the app or form where the requirement will be addressed*	*Detailed description of the expected solution behavior*	*Easy, Moderate or Complex*

Appendix D: Configuration Requirements

The templates below will help you define specific requirements for data collection (fields), reporting, messaging and content security.

Field Requirements

App	Field Name	Description	Field Type	Values	Required?	Location of Existing Data
Name of the app where the field will exist	*Short, descriptive name*	*Purpose of the field*	*Text, number, date/time, values list*	*Values from which the user may select*	*Should users be required to enter data in the field?*	*File name and specific field or column where the data exists*

Reporting Requirements

Report Name	Description	Intended Audience	Dashboard
Short, descriptive name	*Purpose of the report and data to be displayed*	*User roles or groups*	*Dashboard where the report should be displayed to users*

Email Messaging Requirements

Message Name	Email Body	Intended Audience	Trigger
Short, descriptive name	*Message to be displayed to the email recipient*	*User roles or groups*	*Condition that causes the email to be sent*

Security Requirements

User Type	Content to View	Content to Edit
For example, Audit Staff	*Specific types of data in the solution (for example, audit plans)*	*Specific types of data in the solution (for example, audit workpapers)*