Embracing Digital Transformation to Future-Proof Your Organization

By Brennan Binford, Founding Partner at PDG Consulting

PDG

Digital transformation is here and becoming more prevalent every day. It is the process of integrating digital technology into your organization's operational processes, essentially transforming how you deliver value to stakeholders. Research by PwC found that 60% of C-suite executives see digital transformation as the most important 2022 growth driver in which to invest.

If your organization has not undergone a digital transformation, you are rapidly becoming the minority in the business world. **It is not a matter of if your company will digitally transform but when.**

In this white paper, Brennan Binford, Founding Partner at PDG Consulting, shares why now is the best time to embrace digital transformation and when buying, building, or adopting a hybrid approach makes sense.

Getting Started with Digital Transformation: Asking the Right Questions

There are many reasons to start or continue the digital transformation process, such as integrating your business processes, reducing manual tasks, robotic automation, and providing better decision-making tools.

There are innovative and competitive reasons as well: building platforms to leverage network effect, integrating machine learning into your business processes, driving sales through B2B, increasing margins by selling DTC (direct to consumer), providing superior customer experience, improving customer service, and more.

The question is not whether you need to transform. But how?

- What's the vision and roadmap?
- What's the process?
- What tools should you be evaluating?
- Should you build something, buy something, or is there a hybrid model?
- How can you leverage prior technology investments?
- Can your new applications integrate with your existing enterprise applications?
- Should you make decisions based on your company's current skill set(s)?

If these are the questions you are asking yourself, you are on the right track.

Pre-COVID Y2K as a Digital Transformation Catalyst

At the turn of the millennium, there was a big push by businesses to migrate from Cobol (common business-oriented language) custom applications onto ERP (enterprise resource planning) like SAP. Interestingly, it was not to improve business processes, increase revenue, or create a competitive advantage. It was simply to be able to store a four-digit year (2000) vs. the standard two-digit year (00).

Y2K cost businesses billions of dollars. And some would argue that it moved companies away from custom applications that business users needed onto a generic ERP platform that did 50% of what the organization needed really well, 20% somewhat well, and 30% not at all.

The Emergence of ERPs

Motivated by the 1993 book Reengineering the Corporation by Michael Hammer and James Champy, ERPs emerged to help automate and integrate the new business process flow model. Seeing SAP's success and the opportunity in the enterprise application space, products like Oracle EBS, Microsoft Dynamics, and NetSuite soon emerged to compete in the addressable market of big-cap, mid-cap, and small-cap companies.

There were two interesting developments at the time, for better or for worse, and they had staying power:

- First, after tossing untold billions of dollars at a problem without any quantifiable metric(s) of success, businesses realized that moving forward, return on investment and measurable results were paramount to making technology investment decisions.
- Second, custom application development became a four-letter word. After Y2K, which replaced thousands of custom applications with ERPs, the willingness to write a custom application was greatly reduced. The custom application development long winter had begun.

Build vs. Buy: The Age-Old Debate

For years, the build vs. buy decision was easy. Buy an off-the-shelf (OTS) product, install it, and live with whatever shortcomings that exist. If your business processes do not neatly fit into your ERP package, change how you do business. If your users do not like the user interface and experience, tough.

The Customer Striking Back in the Era of the Customer

But times change. With the adoption of the internet into everyone's everyday lives, people have become more accustomed to great user experience. Technologists can no longer provide functionality alone. The user interface must be intuitive, the experience good, and the integration seamless.

ERP companies struggle to keep up with customer preferences. There are too many lines of old code to be nimble. Upgrade cycles, while improving, are too infrequent and often painful. With SaaS products, what you want might not be a high enough priority for other customers and therefore is not on the near-term roadmap.

Cloud solutions like AWS, Azure, and Google Cloud have eliminated the infrastructure procurement and management barrier completely. The "explore, create, implement" approach is easier and cheaper than ever. Prototypes and POCs (proofs of concept) are in the mainstream.

Software Selection Best Practices

Regardless of whether you decide to build a custom enterprise application, buy an off-the-shelf product, or have not formed an opinion yet, the best way to start the evaluation process is on a solid foundation.

Regardless of your corporate size, if you are going through the software selection build vs. buy process, you are about to invest a lot of money relative to the size of your business.

> *"If you don't know where you're going, you might not get there."*
>
> *— Yogi Berra*

We know that what we're about to recommend may seem heavy for some software selection build vs. buy analysis efforts. Trust us when we say that this can be done efficiently and quickly and will save you a lot of time on the backend when trying to finalize the decision and get approval to move forward.

1. Treat the software selection/discovery process like the start of a traditional SDLC (software development lifecycle) project.

- Create a governance structure that identifies who the stakeholders are, who is on the team, and most importantly, who makes the decision(s) and the criteria for making the decisions.
- Define a communication strategy, steering committee, and meeting/status reporting cadence.
- For larger initiatives, create a business case.
- Document the budget and timeline for the software selection process and the overall project.
- Determine the decision-making criteria. A good approach is a list of metrics evaluated using Harvey balls. Harvey balls are a visual aid to indicate 0%, 25%, 50%, 75%, and 100% rating for each criterion.

2. Gather customer needs, use cases, and target specs.

- Identify the OKRs (objectives and key results) for the enterprise application.
- Define the functional (customer needs, use cases) and non-functional requirements (response time, concurrent users) in terms of target specifications.
- Create a mission (or vision) statement for the enterprise application and expand this into the top-line goals for the enterprise application.
- While going through the customer needs assessment, keep your evaluation criteria in mind.

- Assign relative importance to customer needs. If possible, identify any latent needs.
- Organize the customer needs statements into common themes. Look for patterns.
- At this point of the concept-generation process, don't think in terms of solutions.

3. Identify the problems you are trying to solve and evaluate solutions.

- Decompose the customer needs into problems and subproblems.
- Research existing technology solutions. At the beginning of the explore phase, cast a pretty wide net (e.g. ERPs, CRMs, open-source, APIs, third-party solutions, SaaS).
- Evaluate both external technologies and technologies that you already have within your company (including the ones you've built in-house).
- Search with a specific problem or subproblem in mind.
- Create a matrix (e.g. spreadsheet) of the problem/subproblem and the possible solutions to that problem. Remember, you are trying to solve as many of these problems and subproblems as possible in order to address customer needs.
- If you have a software selection team, ask each team member to conduct their own research and create a matrix of their proposed solutions (for some or all of the problems per individual).

4. Evaluate enterprise application software.

There are many available tools for conducting technology solution research.

Gartner Magic Quadrants

As a starting point for enterprise application software, if you understand the category that you're researching (e.g. data visualization tools, CRM), start by reviewing the Gartner magic quadrants. While you need a subscription to Gartner for direct access to these reports (which can be expensive), software companies that are named in a research report publish them online for free as marketing material.

You will need to provide contact information and should expect to be contacted by these companies (possibly repeatedly), but these are solid research and a good place to start. Gartner provides high-level research and a rating based on interviews of existing customers conducted by its team of researchers. This can be very valuable as it is unlikely that you will get direct access to the same number of existing customers to solicit input.

In addition, the Gartner report may evaluate the enterprise application based on criteria that you have not considered. Think about adding these criteria to your success factors. Evaluate several years of reports to get a sense of how the company is trending. Is their overall rating improving? Are the customer satisfaction criteria improving? And so on.

Depending on your customer needs and how much time you have to conduct this selection, you may use this Gartner research to short-list candidates for future consideration. For example:

- If you're searching for an enterprise application for a critical business need and/or your customer is a Fortune 500 company, focus on the Leader and Visionary quadrants.
- If the client is price-sensitive and/or the use cases are more niche, consider applications from the other quadrants. However, document why you're including these solutions and do more research to prove out your inclusion criteria.

Google

Once you have a list of the commercial enterprise applications you'd like to research further, the next step is to scour the internet for information about each application. Use Google to search for specific applications and comparisons between applications. People have likely written about the comparisons of technologies.

If there is a lot of information available from Google searches, it is likely that the product has a large ecosystem. Go to the company website. Look for white papers, watch videos, read the technical and marketing documentation, find the feature sets that are available, and investigate pricing (which is likely a decision criterion).

Don't be afraid to contact the company and ask to speak to a sales representative. They are often knowledgeable and very eager to help. Inform the sales representative that you are conducting a software selection and evaluating their software and some of their competitors (don't specify which ones). If you are ready to see a demo, either request one or plant the seed that you may be asking for one in the future.

However, depending on the timeline of your software selection, how serious you are about this product, and how many products you'll be evaluating, it may be too soon in the cycle to make this request. The customer sales representative will likely request enough of your time and information to cater the demos to your company's customer needs and use cases, which can be time-consuming to support.

When Does It Make Sense to Build or Buy?

Considerations When Buying

Off-the-shelf enterprise software applications have emerged to enable many core business processes (e.g. ERP, CRM). Most commercial products offer APIs (application programming interfaces) or SDKs (software development kits) that allow a certain level of customization to fit customers' specific needs. They may not allow you to cover all your needs, but you may cover a sizable chunk.

The risk with too much customization is maintainability and support. In addition, the risk of too much customization is that you may be constraining yourself to what's allowed to be customized vs. meeting business requirements (the square peg in the round hole). Also, the teams who normally work with commercial products are not typically well-versed in custom development best practices.

Considerations When Building

What if some but not all of your business processes (or a key business process) are addressed with off-the-shelf enterprise software? What if there's a gap in the middle of the process? This is likely the most complex part of your business process, the part of the business that is either very industry-specific or provides your business with its competitive advantage.

If you were able to automate the most critical parts of your business, your business may have a leg up on the competition or it may be able to focus capital and expenses elsewhere. If this is the case, you will want to explore building complementary applications that integrate into your core enterprise application or you may decide to build completely from scratch. If you cannot find a vendor-managed product that meets the majority of your use cases, you'll want to think more seriously about the build option.

Building Your Custom Application

If you decide to consider building your application, you don't have to start from scratch and write every line of code yourself. Open-source applications, third-party APIs, and managed services (like AWS) are a great place to start exploring.

Using Open-Source Applications

Open-source applications may provide a foundation for your application while allowing you to customize and add features that are important to your business. An open-source application often has a more limited feature set but may be easier to extend with custom features. Open-source applications, like JBoss, tend to have limited or no production support available.

Determine if the open-source application has the option to upgrade to a more robust product with additional features and support options. This often comes with additional fees. If you intend to extend the capabilities of any application, become familiar with:

- The SDK that's available
- How easy it is to implement your own features
- If there are limitations on how and how much you can extend the product without nullifying any production support that is provided

Determine how robust the ecosystem is and how many resources are available (and at what price) to extend and support the application.

Integrating Third-Party Frameworks

Identify third-party libraries or frameworks that can be integrated into the open-source platform to accelerate the delivery of features that aren't available in the core open-source application. Some APIs are free while others can be licensed via subscriptions or pay-as-you-go services.

Determine if you can assemble other libraries and open-source frameworks so that you don't have to write everything from scratch. While the idea of integrating a group of frameworks and libraries into an enterprise application may seem risky, a big advantage is that you will own the IP (intellectual property).

But is the platform going to be maintainable years from now? How do you know if developers will continue to enhance and contribute to the open-source frameworks and libraries that you've chosen? If there are several options to choose from, how do you know which platform to choose as your starting point?

Unlike vendor-managed applications that are rated by Gartner and Forrester, open-source applications lack a central authority to provide customer feedback, evaluation criteria, and rankings. You'll need to rely on the development community to provide comparative information. A great place to start is GitHub, which provides independent ratings for open-source projects from people who use the project and contribute to the community. Pay attention to things like:

- How recent were the last code commits?
- How many people are committing code?
- Are the projects still active?

If the last commit is old, the project has limited stars, and few people are committing code to the project, you can deduce that the project isn't in wide use. If a project has hundreds or thousands of stars, the last commit date is recent, and many people are contributing, you know the project is active and there's a robust ecosystem.

Once you have a shortlist of open-source projects, Google them by name and see how many people are blogging about them to determine the level of interest online. Lots of blogs means community engagement and can help you identify trends in the tech community. From your shortlist, Google [this] vs. [that] and see what people are saying.

Exploring Managed Services

Another area for exploration when considering a custom application is managed services like AWS, Azure, Google Cloud, and Oracle Cloud. These companies are constantly innovating and simplifying the custom build process, reducing support costs, and eliminating the need for certain customer resources like database administrators and DevOps. Using RDSs (relational database services), you no longer need a database administrator full time on the payroll or need to worry about certain operating systems or database patching plans.

Managed services like AWS also make integration much easier and provides assurances that the solutions you use will be supported well into the future. You can build applications that maximize and optimize certain features of the cloud infrastructure, providing even more benefits to the organization. Throttling up and down power with techniques like autoscaling can provide horsepower at peak times while reducing costs during off-peak times. With managed services, computing power is nearly limitless.

Creating a Proof of Concept

To IT executives, the idea of assembling a bunch of open-source frameworks, libraries, managed services, and third-party APIs can seem risky. If you do choose to go in this direction, a formal proof of concept (POC) phase is highly recommended.

With managed services like AWS, Azure, Google Cloud, and Oracle Cloud, the barrier to entry of building a POC has been greatly reduced. Creating a development or POC environment can be done in hours instead of weeks or months. If you want to start a custom application development project and create a proof of concept, you can be up and running in less than a day. Off-the-shelf source control and repositories can be connected to your infrastructure in no time.

If the POC doesn't go well, you can blow it away and retire the cloud servers almost immediately, eliminating any infrastructure costs.

Build vs. Buy: Not an Either-or Scenario

For complex business environments and requirements, don't limit yourself to only one approach. Building a complementary application to go with your core ERP or CRM, and integrating the applications together should also be considered. Complements make both products more valuable without reinventing the wheel.

Once you've evaluated how you will solve your problems with all the tools available and have drafted a system-level design, it's time to ask yourself some serious questions.

- Will your users be happy with the final product?
- Can your users articulate what they want? Would you benefit from decades of customer needs assessments and hundreds of thousands of users providing input to improve the enterprise application?
- Is time-to-market more important than any specific features?
- If a SaaS product is in the mix, are there features missing that you need? Are your features on the roadmap? Will the SaaS provider share your vision and priorities for how the product should be enhanced?
- Given how fast the capabilities of technology are improving, should you build your own tech or let a team with a full product development organization?
- Does your organization have the experience, skill, and organizational (methodology) skills to implement a successful custom application development project?
- Are you capable of managing an ERP?
- Are your choices limited by the skills of your existing resources, and is that the right approach?

If you spent the time to define your selection criteria in advance and used a well-defined methodology to run the selection process, you've put yourself on solid footing to choose the right technology architecture and are one step closer to a successful digital transformation.

Championing Digital Transformation with PDG Consulting (PDG)

Digital transformation future-proofs your organization and enables your teams to keep up with rapidly evolving customer demands—it's no longer a nice-to-have but a requirement for business growth. Because it involves changing your processes, systems, workflows, and even the company's culture, we've found that implementing a cohesive strategy that coordinates all efforts across the organization is a critical ingredient for success.

PDG's digital transformation approach takes into account your diverse needs and spans a range of services tailored to your unique requirements:

- Custom enterprise software development
- Data management solutions
- Enterprise mobile software development
- Innovative digital technology

Clients we have helped find success with digital transformation include CBS, Sony Pictures, SESAC, 21st Century Fox, JAFRA, and DIRECTV. To learn more, connect with us today.

About the Author

Brennan Binford, Founding Partner, has been implementing financial, forecasting, business intelligence, and ERP solutions since 1998. He specializes in revenue and margin forecasting, tax, and learning industry specific practices.

Prior to focusing on business intelligence / forecasting, Brennan built and delivered financial and ERP solutions across a wide range of industries. He has a bachelor's degree in Engineering and Applied Mathematics from the University of Pennsylvania and is a Certified Scrum Master (CSM).

ABOUT PDG CONSULTING

PDG Consulting is a Los Angeles-based technology consulting firm specializing in enterprise software development, business intelligence, and digital transformation. Our team of technologists and innovators are experts at translating complex business requirements into application design. Customers like Warner Bros. Discovery, Sony Pictures, DIRECTV, CBS, 21st Century Fox, and Amazon trust PDG to increase productivity, drive business growth, and maximize opportunities.

PDG

PDG Consulting | sales@pdgc.com | +1 (323) 347-4640

714 S. Hill Street, M1 Los Angeles, CA 90014