COMPLIANCEQUEST
Complete Quality Transformed

Whitepaper

Build vs. Buy: **What's Better for QHSE Software?**

At ComplianceQuest, we're gearing up to launch our Two-Twenty Whitepaper Series. These whitepapers offer a two-minute brief on the topic, followed by a twenty-minute deep-dive. In this edition of Two-Twenty, we answer the question of Build vs. Buy: What's Better for QHSE Software.

The **Question**:

Decision-makers often find themselves grappling with this issue of build vs. buy, especially when it comes to software purchases. Business leaders sometimes feel that it is better to purpose-build a software application that is completely customized for your business context. They decide to allocate a budget and hire technology experts (or outsource to a partner) and decide to build the application from scratch.

But is this optimal? And, from our perspective of quality and safety management, is this the right way to automate and digitalize QMS and SMS workflows?

Before we get into answering this question, here are some factors to be considered:

1. **Budget:** Do you have the budget and resources (mainly people) to build next-generation software?

2. **Ongoing Expertise:** Do you understand that building software is **not a one-time initiative?** It may be easy to define the scope and build the first version of software you need today. But what about the ongoing development and continuous improvement?

3. **Complexity:** Do you have what it takes to build a next-generation solution that facilitates cross-department collaboration and ease of integration with other business systems?

4. **Best Practices:** You may very well have the technical expertise, but what about building a platform that incorporates industry best practices? Let us say, for instance, you're looking to implement a CRM, you wouldn't think about building one on your own -- simply because a modern CRM like Salesforce is world-class thanks to its ability to bring in industry best practices from many implementations. Why should it be any different for an EQMS? Why not take advantage of a software platform that incorporates learnings from several use cases?

5. **Cutting-Edge Technologies and Platformization:** Do you have the ability to incorporate the latest technologies including AI, data, analytics, visualization, etc.? Isn't this going to be a never-ending effort to build from scratch? Are you building a simple software product or are your "platformizing" the entire solution so more modules and capabilities can be added later?

6 **Future-Proof:** Are you building for today, to automate a current set of processes? Or, are you looking to future-proof your quality management system? How do you add more modules? Let us say you start by implementing document management, audits and complaint management solutions. Isn't it going to be challenging to further add supplier management and, say, a management review solution? Can we get a platform that truly integrates people, software, and systems?

Overall, business leaders will have to lay down clearly the pros and cons of building vs. buying. **The key is to be unbiased** and analyze both options objectively. This is certainly not a decision to be made based on gut. **Talk to peers in your industry**, understand their experience of building or buying an EQMS software, ask deeper questions, **document your findings** and then take a call.

The Insight:

According to an article published in Forbes last year, buying is often better especially when there are proven solutions in the market. According to the article,

> *"In most situations, working with a reputable provider to deploy a proven solution will allow companies to get the technology into the hands of users more quickly and generally ensures that the solution will have more comprehensive support throughout its life cycle."*

The article goes on to add that, from a technical perspective, it is critical that the software solution being chosen has the following capabilities:

- Ease of customization and **configurable** templates available out-of-the-box
- **Intuitive** user experience, so there is widespread adoption across the enterprise
- **Support for SSO,** user authentication
- Highly secure, scalable and flexible
- **Seamless integration** with ERP, CRM, PLM and other business systems
- Offers **data at the core**, so it becomes a single source of truth for data in your domain (in our case all quality and safety data)
- Cutting-edge **AI and analytics** capabilities that'll make the entire platform future-proof

The Solution:

Going back to our original question:

Should an organization design and develop its own custom applications that address its specific quality & safety needs (Build), or should it purchase a pre-made package from an outside software firm (Buy)?

After years of concentrated development from leading ERP and CRM vendors, would you really want to build your own ERP or CRM application? Then why look at building your own quality & safety application?

This paper outlines the advantages and disadvantages of both the Build and the Buy approach, and in the end, concludes that it is possible to **attain the best of both options.**

Buy a flexible and scalable solution, Built on a modern cloud platform with built-in best practices. Take advantage of the latest data and AI capabilities to truly make life easier for quality, health, and safety leaders.

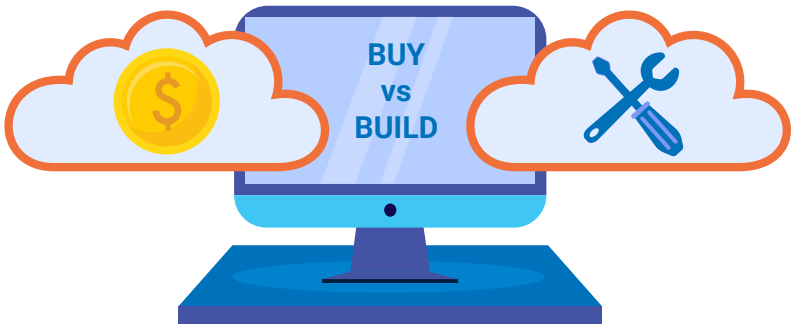
The point is software development is **not about "re-inventing the wheel,"** but about implementing a proven solution that solves a business problem.

At ComplianceQuest, we offer a next-generation QHSE solution built on our five guiding design principles aimed to deliver Quality and Safety Excellence for your enterprise:

The **Pros** and **Cons** of Building

Both providers offer a comprehensive range of quality modules to accommodate all quality processes.

- What is the nature of the problem and the complexity of the application to be built?
- Does the organization have the expert resources to build and support the application over time?
- Is there time to build and support the application?



One can choose to build in the following cases:

- For limited, ad hoc applications
- If the problem is perceived as unique or highly unusual
- To solve a “stand-alone” problem that does not affect any other area of the business
- For a one-time solution with a relatively short life span
- If the organization maintains a large and talented IT department that is dedicated to remaining with the company for a long time

Advantages of building

- Complete control
- Tailored to unique business needs
- Ownership of the software code

Drawbacks of building

Development Time

One of the primary drawbacks to developing software internally is the time it requires from an organization's human resources. For small, non-mission critical applications, time may be less of an issue, but if the organization is developing a complex solution for multiple users or will potentially need to expand the usage to more divisions or sites, countless hours may be spent collecting and harmonizing business requirements and configuring or writing code.

In addition, the organization also must look at the **integration needs** of the application either between processes being built or integration to other existing infrastructure systems.

After coding or the technology configuration is complete, **more time in testing, debugging, and verifying the system** is required until the product and any integration will function according to expectations. Once done, the organization needs to be prepared for further changes and user requests down the line as the initial "configuration" and development of the application may not be "quite right" the first time.

In many cases, optimization for automating manual processes may not have been considered, requiring **more thought in workflow design**. Such re-design and re-testing further entail costs that may not be realized upfront in the initial project implementation.

Training, Support, and Adoption

No software system (no matter how good the design) will be effective without proper training. Often in-house software developers do not have the advantage of specialized training services to ensure that company employees have the knowledge they need to effectively work with the new system. Some organizations may take the effort in the initial rollout training, but what about **ongoing training as employees change positions or new employees come on board?** Organizations must consider the resources needed and the cost for ongoing training efforts to maintain the built application's effectiveness.

The question of support is a serious one. Unfortunately, things can and do go wrong with any software package. Users of the system will require support and the organization must have the time and resources to offer support when it is needed over the life of the built application – not just the initial implementation.

Organizations must include the overall cost of building their own internal help desk or specialized IT department and look at how to maintain and sustain the application when key IT personnel or champions leave, which may impact the overall system integrity.

Staying Current

Technology and business needs change over time. The software application

designed to meet a need today may be out of date in just a few years.

Software applications need to be flexible **to meet change and adapt quickly** to new technology. Here are some of the challenges organizations may face after developing an in-house software package:

- ***New Versions of Infrastructure:*** Software vendors maintain software development teams dedicated to updating their products to be compatible with the latest technology – an advantage most organizations will not have. A software package built to work with the current version of a cloud platform's SDKs or existing hardware and backend application can easily be rendered obsolete with the next version. Some major retooling of the application and a lengthy data migration process may be required. This is especially true if the usage and setup of the built application database are not standardized ("normalized"), extending the time and resource effort needed for change control in upgrades/updates or even process revisions.

- ***Changing Business Needs:*** Software systems must be adaptable. Business practices can change overnight, and the software should have enough flexibility to quickly adjust.

  Access to a large client base gives software vendors the distinct in developing flexible software to meet a diverse set of needs.

- ***Integration with Other Applications:*** A good organization will have the foresight to develop some level of integration with other critical business applications (ERP systems for example). Establishing those business integrations in tandem with a custom-built application increases testing and implementation cycles. Further evaluation is needed when new software systems, such as CRM, MES/MOM, LIMS, are added after the initial rollout of the custom-built application. Even more development and long-term analysis are required for each integration data point, particularly if these applications themselves go through new version releases.

Competitive Functionality

Software users are becoming ever more sophisticated, and they maintain high expectations from the software systems they use. To be successful, software packages must win the confidence of the user. In this area, in-house packages face tremendous competition from established software vendors.

Application vendors create software applications with a depth of functionality and ease of use that is hard to match. With software development as their primary business, they have time to research the problem, collect input from the industry and customers, and develop "best-of-breed" applications.

Software vendors maintain development teams made up of the most talented and highly trained IT professionals available. They are experts in the latest development techniques that are required to produce a reliable, top-quality software product

quickly and efficiently.

For small, non-critical applications, an organization's internal team might be more flexible and efficient than these software vendors. That changes, however, when mission-critical or enterprise solutions are required. Most in-house developers fail to produce a superior product, in less time, and at a lower cost than rivals.

Robust Testing

One of the most critical elements in any software implementation is the testing and verification effort. When a system is built in-house from the ground up, the associated testing scripts need to be developed as well.

A multitude of documentation needs to be developed to enable detailed testing of the software. Examples include

- Functional requirements specifications
- Functional test scripts
- Test matrix – functions vs. test scripts for development of the test plan
- Issues tracking list
- Assumptions and scope of the test cases

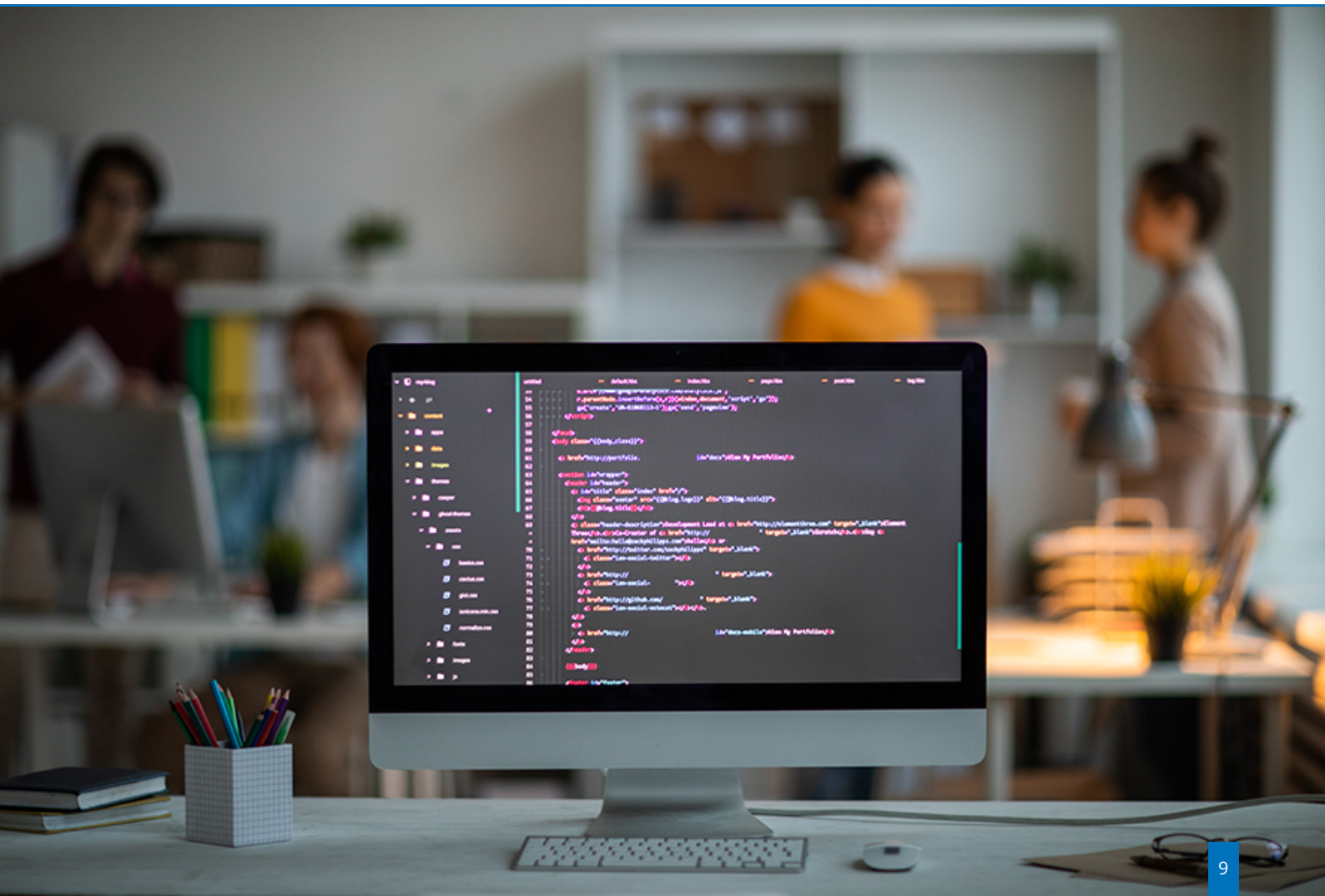
Significant investment is required in terms of time and effort spent in developing the test scripts. Again, depending on the complexity of the system built, the scripts would need to be equally detailed and complex. A typical "build it yourself" testing option would lay a vast burden on internal resources to write the scripts and then test the scripts and execute the scripts. Add to that the cost of executing the scripts and hiring personnel who are experienced in writing test scripts and familiar with the industry's regulations electronic signatures and security. The other option is to "outsource" the development and execution of the test scripts. This option typically could take more than three to six months and the cost can be up to three times higher than the cost of the entire software system.

Reporting

When software is homegrown, the software developers need to build the data models by connecting the different database tables and developing the views. Sometimes, multiple tables might need to be connected and the right views developed in order to be able to extract certain data. This requires having people with the right technical skills – for database management and report development. The report developers would need to be very familiar with concepts of RDBMS such as primary/foreign key relationships, right types of database connections, using the right data storage models, etc. However, since the system is custom-built, business users can get the exact reports that they are used to looking at daily.

Other Risks

- **Turnover:** If the organization's software developer leaves the company, who supports the application?
- **Malicious Code:** While the chances of an internal developer writing malicious code is unlikely, it can and does happen.
- **The Back Door:** A more plausible scenario involves the creation of a back door to the software. In this case, the developer sets himself up as a "super user," meaning that he can do literally anything in the application, including removing system administrators. If the developer is also the application manager, this may not be a problem; if not, then you have just given an outsider complete access to the data you are managing with the software.



Consider the ROI

Does it make good business sense for an organization to custom build a software application? The following chart provides a basis for what to expect in terms of expense. Before building an application or using tools that help you build an application (technology, templates, etc.), an organization should consider the nature of the application it is trying to build and confer with the IT department to estimate the time involved, then complete the chart and see what it will cost.

Task	*Estimated Time*	*Cost = Hours x Loaded Labor Rate*
Define the problem ● Identify entities and relationships ● Map out the business process, including inputs, outputs, all steps, and all participants		
Design system components		
Design and test algorithms/-configurations		
Design and test input screens		
Design and test output screens and reports		
Design and test integration points		
Design and test deployment options for infrastructure and other networking considerations such as SSO		
Build Help Files		
Build Testing Scripts		

Subtotal, Application Development Cost: _____

Total Cost: _____

Note: The above-identified matrix does not consider, the time and resources required to maintain and upgrade the application built by the company during its useful life.

It is best to show a Total Cost of Ownership (TCO) over a 5-7 year period of any software application package.

It may also be wise to recognize the opportunity costs: the value of whatever the IT resources would have been directed at rather than building and maintaining the software.

The benefits can be estimated using a similar method:

Benefit	*Estimated Value per Unit*	*Benefit = Qty x Estimated Value*
Incremental productive activities		
Reduction of non-value added activities		

Total Value: _____

Then calculate the difference: Return = Total Value – Total Cost ROI = Return/Cost

Note: Remember to consider the time value of money. If you are contemplating a lengthy development and implementation cycle, the benefits will be postponed and therefore, would be valued less in today's dollars.

"ComplianceQuest is amazingly robust and easy to use. We first started using CQ for Document Control but today we use it for Training, CAPA, Complaints, Change Control, Supplier Management, and Audit Management.

Our experience has been very positive throughout – right from selection to implementation. The CQ solution is extremely powerful and has a variety of applications and modules. ***Configurable reporting and dashboarding*** *features allow for* ***ease of communication*** *and* ***transparency of data*** *within the company. All processes are electronically linked, adding both ease and* ***traceability*** *to interrelated QMS processes (I can launch a CAPA from my Complaint form!) (I can launch an Engineering Change Order from my CAPA form!). Configuration can be as simple or as complex as you make it."*

– Manager, Quality Assurance, Experic

REQUEST A DEMO

The **Pros** and **Cons** of Buying

While an organization may understand its business needs better than anyone else does, very few problems are truly unique. Purchasing software from a "proven" vendor provides a base of expertise for solving business issues. Rather than "reinventing the wheel," an organization can take advantage of the lessons learned from other companies within your industry that faced similar challenges.

The question of support and maintenance must also be addressed. A custom, self-built system may solve today's problems, but what happens when there is a problem? When technology, business, and regulatory needs change, will the application be able to adapt? Software vendors provide a place to turn for implementation, training, and technical support; they are also better equipped to evaluate the marketplace and incorporate the latest technologies to address change.

Disadvantages of Buying

- Vendor retains rights to the code
- Product functionality determined by vendor
- Reliance on vendor's technical support to resolve issues

Advantages of Buying

- Ready-made solution
- Thousands of hours of research and development
- Fewer “bugs”
- Expert support and training
- Flexibility/adaptability
- Functionality continuously enhanced through customer input

Be Careful While Buying Applications - Not All are Alike

There are two types of software vendors, each offering unique capabilities in helping enterprises optimize and automate their business processes – tool kit technology vendors and off-the-shelf pre-configured application vendors.

While all software vendors spend time developing their technology that is the foundation of the solution they bring to the market, some of these software firms (the “tool kit” vendors) do not go beyond the development of the technology into “tools” and templates that help organizations build a custom application. Through marketing their technology, they have increased their knowledge about an organization’s business need for Quality & Safety, and successfully convert these messages into gifted demonstrations.

However, although “pre-configured” templates and demos can be used as a kickstart or accelerator to any application development effort, the burden still resides on the organization to have the skill sets and resources to complete and test the “end” Quality & Safety application and to train and support the end-users over its lifetime.

Through a careful evaluation process, an organization needs to understand how the vendor will provide its solution not only initially, but for the long term. A pre-configured, fully developed, out-of-the-box application will provide you not only the best practices and domain built into the software but the long-term care of the application itself.

Reporting

Any data in a database is useless if it cannot be analyzed. Any software which provides the ability to store data needs to provide powerful analytical capabilities which, in turn, would enable valuable real-time reports and trends on key performance indicators. This ultimately enables better and faster decision-making.

With a system that is **COTS (Commercial Off-the-Shelf)**, the application’s database models are developed by the vendor. This enables the vendor to provide out-of-the-box or canned reports. The disadvantage is that it is possible business users might not care for the format of a report provided by the vendor. People are used to looking at their data in a certain format and structure and they would like to

continue doing so. This basically means that the reports provided out of the box would have to be modified to meet the exact reporting requirements. However, the advantage is that the reports provided by the vendor can be used as a starting point and then simply modified.

Additionally, vendors with COTS systems based on best business practices typically provide very robust data models. These data models may already contain the requisite views and table connections required to pull the appropriate data.

Multiple Site or Local Organization Deployment

From an implementation perspective, the vendor would need to ensure that its application can provide the ability to meet the requirements for a single site as well as global implementations. A good application vendor would ensure that the system eliminates multiple testing requirements by providing a centralized system with a single application instance, data model and repository. Additionally, the system should provide for global harmonization of processes by allowing for policies and rules to be defined at the corporate organization and driven through all the local sites. At the same time, however, the systems must provide the flexibility for process independence between the sites to incorporate unique business requirements.

Best Practices

Programs, initiatives, or activities that are considered leading-edge are exceptional models for others to follow. Tried and true management practices and work processes can lead to world-class, superior performance.

> "A best practice is a technique or methodology that, through experience and research, has proven to reliably lead to the desired result."

A commitment to using the best practices in any industry is a commitment to using all the knowledge and technology at one's disposal to ensure success. However, demonstrated best practices can be slow to permeate within an organization. The three main barriers to the adoption of a best practice are a lack of knowledge about current best practices, a lack of motivation to make changes involved in their adoption, and a lack of knowledge and skills required to do so.

Some technology vendors who rely on delivering tools rather than fully functional applications may suggest that best practices mean an inflexible solution. But the notion of 'Best Practices' does not commit people or companies to one inflexible, unchanging practice. Instead, 'Best Practices' is a philosophical approach based around continuous learning and continual improvement.

The philosophy of "best practices" is all about not "re-inventing the wheel," but **learning from others and implementing what has been shown to work.**

Why wouldn't an organization want to leverage the success of some of its leading peers and competitors?

Leading software houses such as SAP, Oracle, Salesforce, and others understood this need and have built their fully functional applications based on years of best practices. Today's organizations typically do not build ERP or CRM solutions anymore but look to leverage the domain built into these applications.

It should be no different in the area of quality and safety management.

The **"BUILD"** Option

Customers may have unique business requirements which sometimes cannot be met by standard off-the-shelf systems. Business process requirements dictate that the organizations develop custom software in-house.

The advantage of building software in-house? The ability of the product to meet 100% of the organization's requirements. You define the requirements; you build the software and you maintain it.

The disadvantages? The software is developed in-house and the domain knowledge that goes into defining the requirements is limited to the expertise and experience of a few people within the organization. If a few key individuals within the application development team happen to leave the organization, a big chunk of knowledge walks out with them.

A collective knowledge base from experts across the spectrum of the market segment is always better than a few individuals within an organization or having to source such experience in the market.

Furthermore, as organizations move from a paper-based, manual system to an electronic system, people tend to define their existing processes as requirements for an automated system without much thought on how the process can be improved upon. Without the expertise or skill sets, the end application may lead to an ineffective process being automated. Deficiencies in such a process may quickly surface through automation, causing an organization to potentially re-design the application incurring more time and cost.

The **"BUY"** Option

A good COTS application typically is developed by a vendor after studying the market requirements across a variety of industries or from a specific industry segment. Vendors who focus on different market segments tend to make an effort to better understand the requirements by involving their existing customers as design partners in a good product development strategy. Since their business is software development, they understand cutting-edge development and the new trends in technology, and would be more likely to include the new trends as part of their continuous improvement strategy.

Since vendors tend to understand the requirements – from a business user perspective as well as the technology side – the vendor's application includes all the business processes and data models which are required for customers to be successful in their implementation. So, the organization receives processes or workflows (capabilities), along with business logic and data models that have been tried, tested, and improved upon by different organizations. The vendor does all the work in terms of gathering the business requirements relating to how and what data should be collected, what steps should be completed for a particular process, and builds this depth of domain (best practices) into its application.

Regulatory and technology trends are always moving targets. A vendor focused on certain market segments would maintain appropriate personnel with the required and most current domain expertise to gather required data and incorporate it into the system.

Furthermore, from an **integrated compliance perspective**, multiple systems would need to be integrated to provide a single global view of the state of the organization's compliance and to ensure that there is a closed-loop process in place.

This may include the need to determine the number of people to be trained on the documents that are being released. It also may include the need to escalate audit findings to be resolved through a CAPA process, address needs when an incoming material inspection failure may trigger a product disposition, etc.

The best value to a customer comes from a vendor who can provide many of these integrated process models out of the box. Best practices dictate that a good COTS system should intend to meet a **customer's requirements 90-95% out of the box** while providing them the **flexibility to tailor the system to meet their unique requirements.**

The **Journey** of a Customer

SPR Therapeutics, a fast-growing medical device manufacturing company specializing in neurostimulation technology, offers a revolutionary approach to drug-free pain management. Founded in 2010 and headquartered in Cleveland, Ohio, SPR Therapeutics is a privately held company dedicated to providing a new standard of care. Their flagship product, SPRINT PNS, offers safe and effective pain management options to treat both acute and chronic pain without the use of opioids or invasive surgeries.

In 2016, SPR Therapeutics received FDA clearance for its SPRINT PNS product. With the goal of increased market penetration, they pivoted their focus from solely an R&D company to a blend of research and product sales. SPR Therapeutics' customers include hospitals, surgery centres, and pain management practitioners.

The company was using a **combination of manual processes and legacy software** to manage the quality management workflow. According to the company's Director of Regulatory Affairs and Quality Systems, this proved out to be a major bottleneck. It was adding complexities and inefficiencies into not only the quality management workflow but the overall product lifecycle.

The company clearly needed a QMS upgrade. It could have built a solution, but that would inefficient and a waste of time.

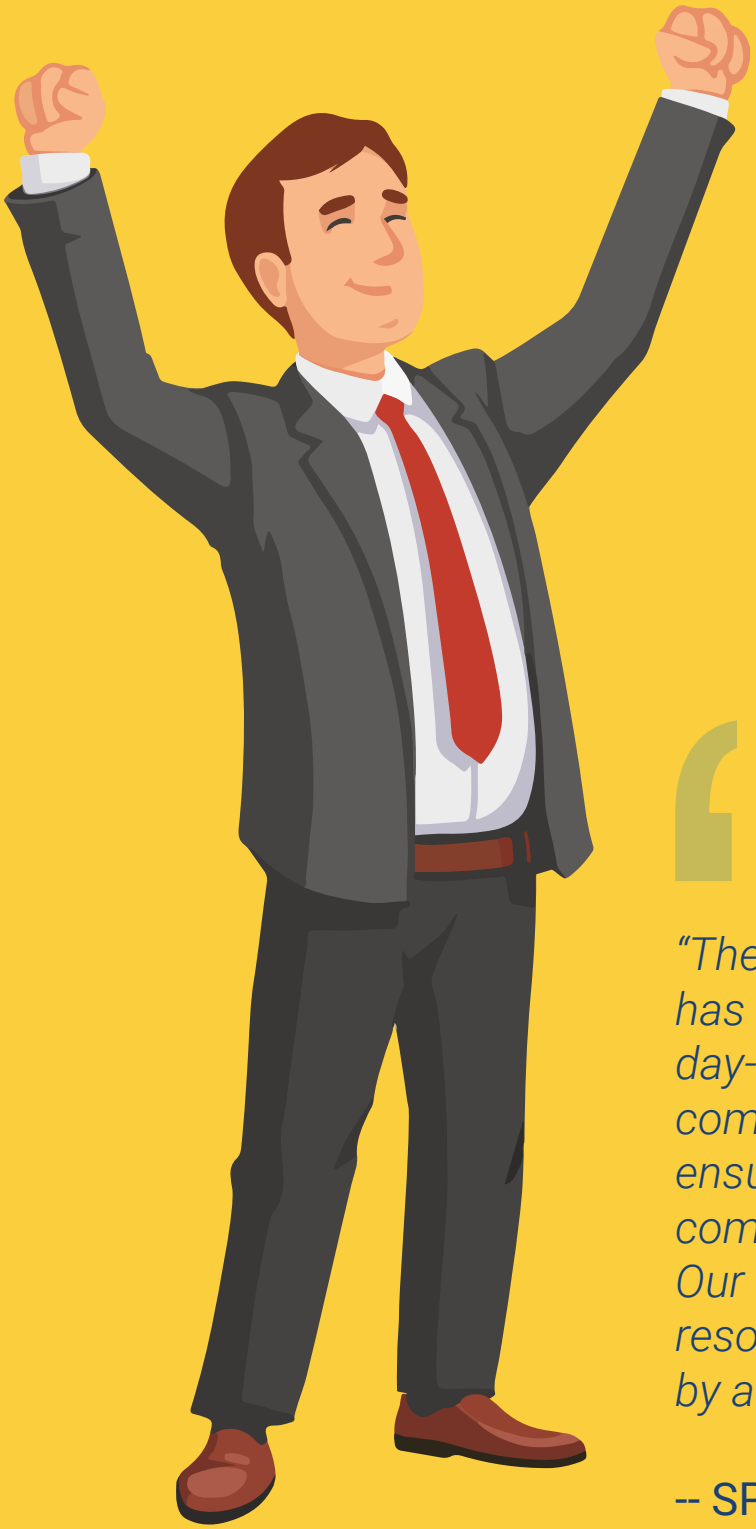
The company's leadership soon realized that the only way to do this right would be to **migrate to a next-generation, cloud-based EQMS** for the following reasons:

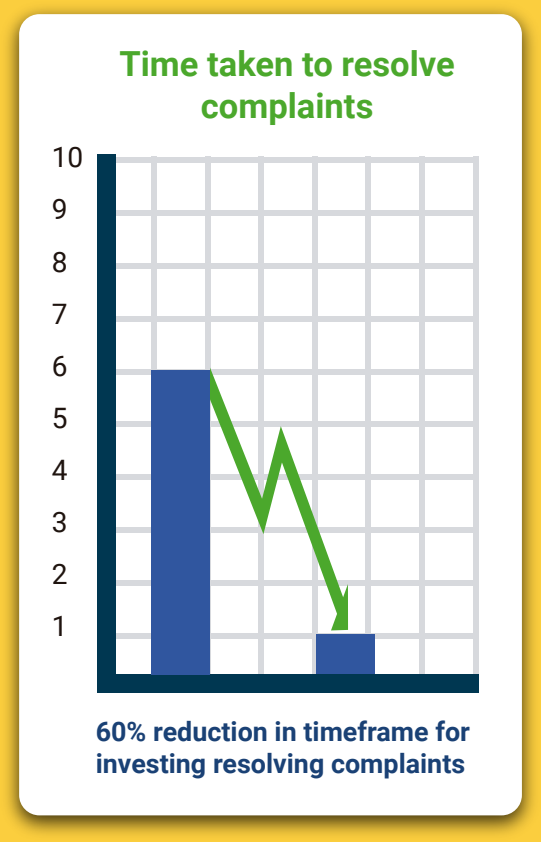
1. The company was scaling fast and it was critical to embrace digital transformation and automation to drive **operational excellence**.
2. It needed a next-generation EQMS that would seamlessly **integrate with its ERP and CRM**. (SPR Therapeutics decided to go with ComplianceQuest for EQMS, Salesforce for CRM, and Rootstock for ERP).
3. There was a crying need for an integrated system to **automate** complaints handling, document control, calibration, inspection, and training.
4. Quality analytics with world-class **dashboards and data visibility** across the product lifecycle would drive continuous improvement of quality management processes.

5. **FDA compliance** and all other regulatory requirements had to be automated.

6. An integrated system connecting the EQMS, ERP, and CRM was key to **handling complaints with agility**.

For SPR Therapeutics, migrating to a next-generation cloud-based solution was critical to improving both quality and compliance. It also ensured that the quality team and other stakeholders saved valuable time and became more productive.





"The Complaints Management System has made a huge impact on our day-to-day operations. We can track complaints from receipt to closure ensuring customer satisfaction and compliance with industry regulations. Our timeframe for investigating and resolving complaints has decreased by an average of 60%"

-- SPR Therapeutics

In today's world of digital-led growth, it is a no-brainer to move your QMS to the cloud. And, it is critical to do it fast.

Make your move now. Don't waste time in building, when you can buy a world-class, truly next-generation solution from ComplianceQuest.

Request for a demo here:

https://www.compliancequest.com/online-demo/

About ComplianceQuest

Transform to a fully connected business with a **next-generation AI-Powered Quality and Safety management platform, built on Salesforce**. Our connected suite of solutions helps businesses of all sizes increase quality, safety and efficiency as they bring their products from concept to customer success. Our intelligent data-driven platform comes with best-in-class integrated processes to mitigate risks, protecting your employees, suppliers and brand reputation, and to increase innovation, compliance, profit and customer loyalty. ComplianceQuest is pre-validated and easy to implement, use, and maintain, allowing for streamlined communication and collaboration across the product value chain.

For more information, or to request a demo with a ComplianceQuest expert, contact ComplianceQuest today:

- Visit **www.compliancequest.com**
- Email us at **marketing@compliancequest.com**
- Call us at **408-458-8343**