



Build vs. Buy

Ten Lessons Your CIO Should Share About Enterprise EHS & Crisis Management Software

An Executive White Paper

By Jill Barson Gilbert, QEP
President and CEO, Lexicon Systems, LLC



Build vs. Buy

Ten Lessons Your CIO Should Share About Enterprise EHS & Crisis Management Software

Executive Summary

Build vs. buy software decisions often pit business and operations against the Chief Information Officer and the Information Technology (IT) department. Individual agendas and stakeholder needs compete for attention. Users present complex, almost undecipherable requirements, and the project team must deliver a user-friendly solution on time and within budget. If you and your CIO have a good relationship, ask him/her to share their expertise to help you make an informed decision. If not, then use the insider tips in this white paper to help.

What worked in the past does not always work in today's context. Not long ago, organizations of all sizes sported robust IT departments. In the last few years, many have outsourced the IT function altogether, focusing instead on their core business. Others maintain IT staffs in an effort to better control the outcome of IT initiatives at a time when IT is constantly changing. Companies have difficulty tracking, let alone planning and implementing system additions or upgrades.

Business today is global, on-demand, 24/7. Companies must be agile and quick to respond to changing markets and opportunities. While Environment, Health and Safety (EHS) compliance and Crisis Management have been corporate issues for 35 years, today they possess a new sense of urgency due to Sarbanes-Oxley and other corporate governance drivers.

Companies now recognize that EHS and Crisis Management represent critical operational risks. Their ability to help ensure regulatory compliance depends upon having a system to continually collect, process and report the appropriate information. That system should also track ongoing regulatory changes and their business implications. Finally, companies are increasingly aware that they need systems to help them plan for and respond effectively to all kinds of crisis events—from industrial accidents to natural disasters—to ensure business continuity.

If you are thinking about implementing or upgrading an EHS compliance or Crisis Management (EHS/CM) system, you probably know that you can choose from a variety of commercial software packages. But will they do exactly what you need? If you have an internal IT staff, you may wonder if they can build a better solution than you can buy in the marketplace. Hence, the great debate...

Build vs. Buy

Companies recognize that off-the-shelf software has many advantages, among them, standardized solutions that incorporate the latest technical innovations. The benefits of purchasing commercial software applications include:

- Best practices built in; software reflects the experience of a broad range of users
- For mature software packages, proven effectiveness drives time and cost savings
- Fewer bugs, further minimizing costs and potential risk
- Vendor-maintained code, enhancements and help desk
- User organization not distracted from its core business
- Software-as-a-Service (SaaS) minimizes the burden on in-house IT resources.

Yet custom-built software applications remain common today. According to McKinsey & Company analysts, certain business sectors—financial services, pharmaceuticals, high tech, and media—spend more than half of their applications budgets to enhance, support and maintain customized software.¹ Banks build custom applications to support new financial products or to manage risk, and

pharmaceutical companies to build applications for Research & Development and marketing activities. Custom software often makes sense in extremely fast-moving markets where an internal IT team can develop solutions so specialized that the advantages of scale and standardization may not apply. What about businesses that require EHS/CM solutions? What makes sense for YOU—build or buy?

The Hybrid Solution

For many organizations, the short answer is to buy, but this is too simplistic. Every enterprise is different and needs to make its own assessment. A more complete answer, according to many technology users and providers, lies not in developing an all-new system from scratch but rather in adapting existing systems and integrating them where it makes sense. It is generally worth the effort to configure or customize commercial EHS/CM software vs. building your own.



Microsoft, IBM and other major technology vendors promote applications of the future that are hybrids of packaged and custom solutions. IBM believes that next-generation applications will combine tools, middleware, and consulting services. Microsoft envisions solutions that leverage their systems technology, Office applications, and Smart Tags to connect people to the right data at the right time.²

The hybrid approach raises its own questions. How many technology providers must you engage to get the solution you need? How much customization is TOO much—too difficult and/or too expensive to be worthwhile? Who has the expertise to help you design and implement an optimal solution?

There are always SOME cases where it is best to build. If your project is like most, you will have proponents in both the “build” and “buy” camps. Each will articulate their arguments with great conviction and emotion. However, it is important that you make the final decision after careful evaluation of the company’s needs, culture, and capabilities.

Making Your Decision

The following pages present ten lessons learned over the course of numerous EHS management information system (EMIS) projects and ongoing dialogs with senior management in software companies and user enterprises. We trust that these lessons will help you make a sound decision.

Lesson 1: Conduct an objective assessment.....	3
Lesson 2: Your issues are not unique	3
Lesson 3: One size does NOT fit all.....	4
Lesson 4: Follow IT Best Practices	5
Lesson 5: Measure what matters.....	5
Lesson 6: Software development is not for everyone.....	6
Lesson 7: Software is risky business.....	6
Lesson 8: Don’t fall for speed traps.....	7
Lesson 9: Beware of hidden costs.....	8
Lesson 10: Know Thyself.....	9
Conclusions.....	9

Lesson 1: Conduct an objective assessment

The Needs Analysis or Needs Assessment is fundamental to any software initiative. It sets the project boundaries and provides the foundation for project strategy, plans and implementation. Before you can make a “build or buy” decision, you must clearly and objectively identify the needs of the various stakeholders.

A good assessment is hard to find. Internal assessments are seldom objective because:

- they favor internal biases and knowledge
- they focus too much on detailed subject matter needs versus the big picture
- stakeholders with competing interests find it difficult to reach a consensus on needs
- stakeholders may resist change
- the company has IT staff available and needs to assign them to projects
- the organization rarely has the combination of business, subject matter and IT expertise

Stay focused. During the assessment, it is vital that you:³

- understand the problem to be solved
- involve stakeholders and document their needs
- select a solution with well-defined business logic
- look for fit with near-term and future needs
- evaluate alignment with the organization’s strategy
- consider the organization’s information technology strategy and standards
- catalog existing EHS and information and technology throughout the enterprise
- find a translator that can speak to EHS, business and IT needs

Know what forms a capable system and what to seek when buying a package.⁴ Find a system that is:

- proven – achieving strong measurable results in the field for many clients over many years
- best of breed – the most effective, fully integrated blend of features and functionality
- scalable and flexible – offering modular solutions to the broadest variety of potential applications; available in multiple delivery options (onsite, hosted, subscription) to fit your IT environment

If you cannot conduct an objective assessment, find a third party with the expertise and experience to provide an honest and credible recommendation.

Lesson 2: Your issues are not unique

What about MY needs? Your organization has decided to implement an EHS/CM information system. Since the organization has never used software for this purpose, someone says that your IT team must build it from scratch. You surf the Internet and see off-the-shelf software applications that claim to “meet the unique needs” of your organization. What is fact, and what is fiction?

No matter your sector—industry, government, nonprofit, or academia — your objective is to automate one or more common business processes, such as document, compliance, task, or incident management. While your organization may have a unique culture and perspectives, chances are that another organization has faced the same issues that you face, just with a different twist.

Your IT group may require that all software must sit on company servers, behind company firewalls. Or, your IT group may allow software that sits in a vendor’s secure data center, with delivery over the Internet. You can find commercial EHS/CM software delivered by one or both of these methods.

Learn from others. See how other organizations manage the business processes you plan to automate, e.g., task calendar, Crisis Management, emissions management or document management. Take advantage of lessons that others learned when they developed software in-house. Did they get the desired features and functionality on time and within budget? What challenges did they face, and

how did they meet these challenges? Did the organization involve end-users and others during the development process, and do the users embrace the software today?

Find out how software packages on the market address your challenges. Find a vendor that has clearly drawn upon the experiences of a broad user community and used their input to develop systems that facilitate proven best practices in information management. Do not reinvent the wheel.

Lesson 3: One size does NOT fit all

Adapt the software or adapt the business. Organizations often shy away from commercial EHS/CM packages because they believe the software will not fit the way they do business. They do not want to change the way they do business to fit the software. Worse, the software may require too much customization, making it difficult to support. The reality is, in both build and buy situations, the organization must try to streamline its EHS and Crisis Management processes. There is no benefit to automating poor business processes.

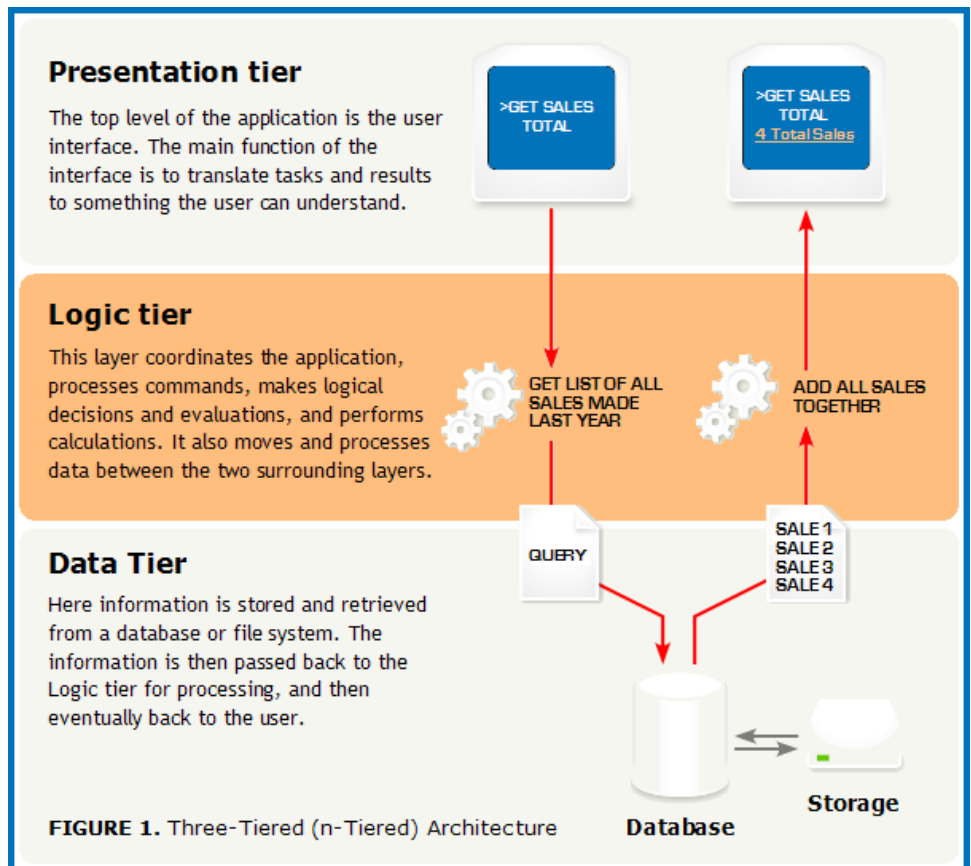
Configure vs. customize.

The trick is often in understanding what kinds of alterations or tailoring your system might require. What makes sense for your organization?

Today's software typically has three layers—the underlying database structure, a middle layer with process logic (business rules), and a presentation tier (user interface) (Figure 1). You can configure the user interface or write company-specific reports and not customize the underlying data tier. Vendor upgrades typically support configuration—similar to switching on the cruise control in your car or setting options in your office software—but not user customizations to the underlying software code.

Today's options are tomorrow's standard features.

Certain unique characteristics of your organization justify configuration of commercial software to let you keep a competitive edge. Other characteristics justify customization to help keep you lean and mean. If your organization develops custom features that reflect best practices, then the software vendor should be willing to incorporate these features—especially data entry forms, reports, and data migration tools—into the commercial product. Think about the car you drive today and your first car. Chances are the car you drive today has many features that were options or custom add-ons years ago. When custom features become mainstream, they raise the bar for the entire software user community.



Lesson 4: Follow IT Best Practices

Follow software industry best practices whether you build or buy. IBM Rational (formerly Rational Software) articulates six Business-Driven Development principles for the creation, deployment, and evolution of software-intensive systems:⁵

- **Adapt the process.** Sometimes, less is more. Smaller projects with co-located teams and known technology require less formal processes. Larger projects with distributed teams, more complex technology and many stakeholders require processes that are more formal. Right-size the process and continuously improve it.
- **Balance competing stakeholder priorities.** Software initiatives often have conflicting business and stakeholder needs. Understand and prioritize business and stakeholder needs, but to design the software with the end-user in mind. Leverage legacy systems where practical while considering that reusing these systems may not meet all end-user needs.
- **Collaborate across teams.** Software initiatives involve business, subject matter, IT, and operations team members. Give members the authority to make decisions, have them commit to deliverables, and motivate them to perform. Encourage cross-functional collaboration and provide an environment that facilitates automated status reporting and metrics and fewer meetings.
- **Demonstrate value iteratively.** The new paradigm is to provide value to stakeholders early and often. This approach, also known as Agile Development⁶ or Rapid Development, provides deliverables that, with user feedback at each iteration, get one step closer to the final solution. The iterative approach also helps to reduce risks early in the project life cycle—early and continuous user feedback lets the development team know if they are on the right track, and if the software meets the requirements.
- **Elevate the level of abstraction.** Design the high-level building blocks and the most important components early in the project. This constitutes the framework for future modules, code, capabilities or users.
- **Focus continually on quality.** Quality goes beyond meeting software requirements. The entire team owns a continuous, repeatable process. Analysts document business requirements and ensure that others can test the requirements; developers design applications with end-use in mind, and test their code before releasing it; IT and business teams test the application and provide feedback; and management ensures that test plans and resources are in place. The iterative process results in incremental development of a software system, allowing the developer to take advantage of lessons learned during the development of earlier deliverables.⁷

You might find it easier to purchase commercial software because software development is a vendor's core business. The goal of the entire organization is to facilitate and support IT best practices. Those who develop EHS/CM software internally do so in an organization whose goals focus on the core business, not on IT. In a recent survey, nearly half of CIOs said that their organization views IT as a support or staff function, not a strategic element.⁸ IT departments may find it difficult to apply IT best practices consistently because of other priorities.

Good software design calls for a multitasking team beyond analysts and programmers—designers, graphical user interface experts, testers, report writers, trainers, etc. EHS subject matter experts do not have these skills. You may have to compete for these IT resources because different functions throughout the enterprise have added IT applications and services over the past few years.

Lesson 5: Measure what matters

Whether you build or buy, an effective EHS or Crisis Management information system includes a performance management system, including Key Performance Indicators (KPIs). *Key Performance Indicators* are

Traditionally, EHS business intelligence involved historical information flowing one way—from operations to EHS professionals to management. Well-designed KPIs can alert EHS and business managers to trends using forward-looking metrics.

financial and non-financial metrics that compare the organization's performance against its strategies and objectives. KPIs align with an organization's strategy (for example, using techniques such as the Balanced Scorecard) and differ from one organization to the next.⁹

Many companies already collect the data necessary to generate KPIs. In its recently updated guidelines on environmental key performance indicators, the UK identifies three principles for Environmental KPIs. They must be:¹⁰

1. **Quantitative:** make each KPI measurable and use standard units of measure. This allows enterprises to set targets that they can act upon, and to confirm impacts on the environment.
2. **Relevant:** for each KPI, take into account all relevant information and comparators, and clearly describe progress made for the reporting period.
3. **Comparable:** express KPIs in absolute terms for the reporting period, e.g., tons/year, to allow comparisons with other companies.



A few commercial EHS and Crisis Management software packages marry KPIs with data analysis tools. The resulting dashboards quickly allow the User to view the state of the organization. Just like your car's dashboard—a cluster of digital or analog displays that provide a quick overview of your car's status—a software dashboard displays EHS and Crisis Management information. Dashboards offer one-stop shopping for EHS/CM information, displaying information from different sources, independent of the technology platform. Users view information and metrics relating to their job roles, as well as links to key resources such as regulatory databases, agency Web sites, or material safety data sheet (MSDS) repositories.

Lesson 6: Software development is not for everyone

Sometimes CIOs are afraid to admit this. After all, development is only part of the picture. You need to be prepared to manage your internally built software over its entire lifetime, from concept to development to installation and long-term support. Moreover, in-house development requires a certain amount of focus and structure, and may require retooling IT governance.

Needs drive development, not vice-versa. Your IT organization may have talented business analysts and programmers, but do they understand EHS/CM needs, and can they translate needs into clear and concise requirements? If you lack good EHS business analysts, find a translator who understands EHS, business and IT issues. Otherwise, consider a commercial application from a vendor that has the combined skill sets. Remember to focus on what your organization does well!

Change happens. Most EHS/CM software initiatives span months or years. Over the project lifetime, people move on. Do not allow the project to rely too heavily on one or two individuals. Be sure that your team includes redundant expertise, whether on staff or outside the company.

Regulatory-driven processes, especially EHS, continually change. You should ensure that your EHS/CM software has frequent updates to ensure that regulatory information remains accurate. Look for software that links with a leading regulatory reference service and sends automatic updates that help you quickly adjust your business processes.

Ensure that you develop strategies for system upgrades and for system end-of life. If you plan to build a system, invest enough resources to handle not only the initial build, but also to keep the application current when technology changes.

Lesson 7: Software is risky business

Risk is inherent in any software effort, whether you build or buy. Simply defined, a *risk* is any potential problem that could cause some loss or threaten the success of our project by increasing the cost of the system, delaying its delivery, or reducing its benefits. Risks can adversely affect the project cost, schedule, or technical success, software quality or project team morale. A *critical risk* is a risk that can jeopardize the successful delivery and/or operation of the system.¹¹

To ensure your EHS/CM software initiative's success, you must identify and manage these risks. One risk identification approach puts risks into five categories, as follows.¹²

- **Dependencies.** Many project risks stem from dependencies on outside agencies or factors that we cannot control.
- **Requirements Issues.** The project team must clarify requirements early to avoid building the wrong product, or building the right product badly. Be aware of these risk factors.
- **Management Issues.** Management issues can derail a project. Use defined project management processes and clear roles and responsibilities to mitigate some of these risks.
- **Lack of Knowledge.** The project team may not have the skills required to execute the project, especially with technology changing so quickly. These factors might apply to your team.
- **Other Areas.** Other risks include unavailability of development or testing equipment and facilities; inability to acquire resources with critical skills; turnover of essential personnel; unachievable performance requirements; problems with language translations and product internationalization; and technical approaches that may not work.

In making your Build vs. Buy decision, after you identify risks, you must evaluate and prioritize them considering how one approach or another impacts them. Watch for opportunities to mitigate your risk by, for example, buying commercial software from a vendor that has proven its ability to avoid pitfalls while successfully implementing truly global EHS systems. And, before moving forward with your EHS/CM software initiative, know how you will address each risk and what contingencies you have in place when risk becomes reality.

Lesson 8: Watch for speed traps

Speed kills. When you start an EHS/CM software initiative—a commercial application, an in-house effort or a hybrid project—you might feel pressure to deploy the system “yesterday.” Do not give in to the urge to accept unrealistic deadlines

Speed bumps can slow the project, such as unclear requirements, roles, and responsibilities. Whether you build or buy a system, be on the lookout for:

- **Unclear requirements, roles, and responsibilities.** Get the right people involved early, and get agreement on the project strategy, scope, schedule, budget and responsibilities.
- **Overanalysis.** Avoid excessive second-guessing, which leads to scope creep, unnecessary redesign and rework. At some point, you need to make decisions and move on.
- **Resistance to change.** Refusal to accept new business processes, new interfaces, outside help and experience can undermine a software project.
- **Adopting an overly complex user interface.** A simple, easy-to-use software interface reduces intimidation and increases the speed with which users adopt the software throughout the company.
- **Assuming that an internal team can deliver the application sooner** than a software vendor or implementation firm can. In different instances, each can outperform the other.
- **Skipping milestone reviews.** Project milestone reviews have a purpose, to allow the project to proceed or not, based upon predefined acceptance criteria.
- **Inviting experts in to educate your internal development team.** Why not hire the experts in the first place?



Lesson 9: Beware of hidden costs

Do the math. When organizations evaluate whether to build or buy software, they often make a math error—"We didn't write a check to a vendor, so the software did not cost that much." Do you have a realistic understanding of your software project costs? If you purchase commercial software, the total cost is greater than the license fees. If you develop software in-house, then the total cost to your organization is much greater than salary, benefits and overhead for the IT team.

Use estimation tools. Software project costs can be difficult to estimate, and no estimate is certain until the project is complete. However, many articles and books on software estimation and software project management can help identify the cost components.

Evaluate the Total Cost of Ownership. Tables 1 and 2 summarize the cost components for packaged software applications and internally built applications, respectively. Project management cost elements are very similar for build vs. buy. Maintenance and support costs can be much lower for packaged than for custom applications, especially with the SaaS delivery model. Packaged software fees for licenses, maintenance, implementation, etc. are a tradeoff for internally built software development fees. Internally built software adds cost elements for product management functions. Also beware of commercial packages that are priced very low but require you to build out so much of the basic functionality you need that your total cost of ownership is far higher than expected. Both packaged and internally built software have certain opportunity costs—lost business, employees that could have worked on other issues, and schedule delay costs.

TABLE 1. Cost Components: Packaged Software

<p>Project management</p> <ul style="list-style-type: none"> Determine customer needs Gather requirements Budget approval Software evaluation Software gap analysis Proof of concept modeling Data collection Training Evaluation 	<p>Fees</p> <ul style="list-style-type: none"> License Fees Implementation fees Configuration fees Integration fees Customization fees Annual maintenance fees 	<p>Maintenance and support</p> <ul style="list-style-type: none"> Ongoing maintenance* Software upgrades* Auto-update tools* Help desk Issue tracking tools
		<p>Opportunity costs</p> <ul style="list-style-type: none"> Schedule delays

*If installed inside Company firewall. Greatly reduced or absent with SaaS model.

TABLE 2. Cost Components: Internally Built Software

<p>Project management</p> <ul style="list-style-type: none"> Research Data collection Model development Methodology development Validation Training Evaluation 	<p>Software development</p> <ul style="list-style-type: none"> Software model creation Software designer tool creation Software module creation Web development Country-specific versions Data localization Data mapping Data migration Testing Documentation Rollout 	<p>Maintenance and support</p> <ul style="list-style-type: none"> Ongoing maintenance Software upgrades Auto-update tools Help desk Issue tracking tools
<p>Product management</p> <ul style="list-style-type: none"> Pricing Product changes License monitoring 		<p>Opportunity costs</p> <ul style="list-style-type: none"> Business Employees Schedule delays

Budget for the future. Allocate budget not only for the initial software build, but also for ongoing maintenance, upgrades, and support. Remember to budget for technology upgrades to keep the application current. Otherwise, you risk shortening the usable life of the software, which is costly.

Lesson 10: Know Thyself

How Agile is your organization? Agile organizations have a competitive edge. Executives in a range of industries, governmental and nonprofit organizations feel that their businesses must be agile to enhance their performance. Businesses must be able to shift strategic direction and quickly execute plans to meet organizational objectives. Nearly 90 percent say that agility is “extremely important” or “very important” to business performance. Unfortunately, 31 percent of executives say that their companies are “much less” or “somewhat less” agile, and slower than the competition.¹³

Properly applied, IT can help organizations to become quick and agile. It can increase efficiency, decrease reaction time, improve communications, and yield positive financial benefits. Conversely, IT applied haphazardly can torpedo an organization’s competitive edge.

Are you receptive to new technologies? Organizations must recognize where they fit on the technology adoption curve (Figure 2).¹⁴ This relates to the ability to embrace new information management systems and optimize their use.

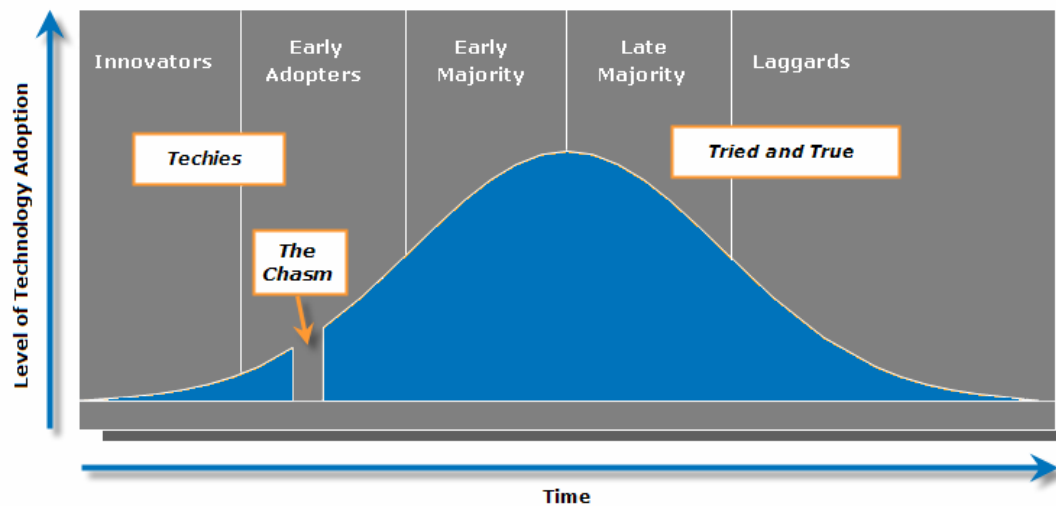


Figure 2. The Technology Adoption Life Cycle

As shown here, innovator and early adopter organizations feel they gain a competitive edge by adopting new technology early. They are willing to be the “alpha customers” and work out the bugs in new software. Laggards resist adopting technology, and enter the game so late that they miss market opportunities.

The great majority of organizations adopt new technology conservatively. They prefer systems that others have “road tested” and are on their third or fourth version. They want the security of knowing that their EHS/CM system will be reliable and will deliver results. This is one of the biggest reasons that organizations choose to buy proven commercial software rather than build their own.

Conclusions

Build and Buy are appropriate for different organizations for different reasons—depending on the complexity of the user needs, the organization’s culture, skill sets, etc. Good arguments exist in both the “Build” and “Buy” camps (Figure 3). In our experience, purchasing a commercial software package from a proven vendor—and then configuring and customizing it as necessary—is the best option for most organizations. A commercial EHS/CM software application that truly meets the organization’s needs is usually preferable to a one-off, internally built solution.

More and more organizations buy commercial EHS and Crisis Management software for several reasons. Enterprise software has matured over the years, boasts integrated modules and is feature-packed. The market is at a tipping point, and Software-as-a-Service makes commercial applications attractive to a pool of prospective buyers. Configuration and customization are reality, but today's software is easier to configure than ever and should limit the amount of customization needed.

To determine if build, buy, or a hybrid solution is best for your organization, you need an honest assessment. In addition, you need to understand what characteristics to look for when buying a package. While each organization has its idiosyncrasies, you can benefit from the best practices built into many commercial software applications. If you find that the solution you need does not exist at all, then custom software development may be your only option.

Figure 3. Build vs. Buy Matrix^{15,16}

	Benefits	Drawbacks
Build	<ul style="list-style-type: none"> <input type="checkbox"/> Tailored to unique needs <input type="checkbox"/> Understand organization's business processes and culture <input type="checkbox"/> Designed with current and future needs in mind <input type="checkbox"/> Control your own fate <input type="checkbox"/> Leverage internal staff <input type="checkbox"/> Fosters innovation <input type="checkbox"/> Open source tools <input type="checkbox"/> Service-oriented architecture 	<ul style="list-style-type: none"> <input type="checkbox"/> Software is not "core business" <input type="checkbox"/> Difficult to estimate total cost <input type="checkbox"/> Incomplete skill sets, staff turnover <input type="checkbox"/> New technology learning curve <input type="checkbox"/> System requirements change as features develop <input type="checkbox"/> Less credibility with the client <input type="checkbox"/> Often non user-friendly <input type="checkbox"/> Assume all of project risk
Buy	<ul style="list-style-type: none"> <input type="checkbox"/> Software is "core business" <input type="checkbox"/> Objective, independent third-party <input type="checkbox"/> Adopts industry best practices <input type="checkbox"/> Typically user-friendly <input type="checkbox"/> Configurable, customizable, scalable <input type="checkbox"/> Integrated modules <input type="checkbox"/> Can integrate with other systems <input type="checkbox"/> Choice of delivery—"inside the firewall" or "software-as-a-service" <input type="checkbox"/> Vendor maintains IT infrastructure, handles software maintenance, upgrades, help desk (SaaS) <input type="checkbox"/> Shift much of project risk to vendor 	<ul style="list-style-type: none"> <input type="checkbox"/> Software may lack key features <input type="checkbox"/> Time to market for future features <input type="checkbox"/> High expense if significant customization or integration required <input type="checkbox"/> IT infrastructure upgrades required as technology changes ("inside the firewall" installation) <input type="checkbox"/> Software maintenance, upgrades, help desk ("inside the firewall")

If you take these ten lessons, to heart, then you will be prepared to make an informed build or buy decision. Consult your CIO or other IT professionals. Talk to peers who have been through an EMIS implementation and learn from them. Always consider the total cost of ownership, and remember your core business!

About the Author

Ms. Jill Barson Gilbert advises senior management in industry, as well as software, investment, and consulting firms. Her career includes EHS positions of increasing responsibility in industry, many years as an EHS management consultant, and ten years in EHS software product management, strategic and implementation roles. She has a rare perspective, and is a thought leader on EHS management information systems. Jill is a past Vice President of the Air & Waste Management Association and writes the bimonthly "IT Insight" column for *EM*, an A&WMA publication. Ms. Gilbert earned an M.S. in Environmental Management from the University of San Francisco and is a Qualified Environmental Professional. For information, visit www.Lexicon-Systems.com.

About ESS

ESS is the leading provider of Operational Risk Management software and services for EHS and Crisis Management. The company has provided information management solutions to over 17,000 organizations worldwide, including nearly half of the Fortune 500. Its *Essential Suite*[™] and *Compliance Suite*[™] software portfolios are used by business leaders from a host of industries including oil and gas, chemical, utilities, manufacturing, technology, healthcare, education, and more—as well as federal, state and local government agencies and military institutions. ESS software helps users comply with international, federal, state and local regulations and industry standards. ESS' fully integrated product suites transform EHS and Crisis Management data into the information needed to support regulatory reporting and performance metrics so you can drive operational expenses and risk out of business. For more information, visit www.ess-home.com or call 800-289-6116.

¹ Marwaha, Sam, Samir Patil and Ranjit Tinaikar, "The Next Generation of In-House Software Development." *McKinsey Quarterly*, Web Exclusive, February 2006, www.mckinseyquarterly.com/article_page.aspx?ar=1747&L2=13&L3=11.

² Greenbaum, Joshua, Build vs. Buy In the 21st Century, *Intelligent Enterprise*, April 22, 2003, www.intelligententerprise.com/030422/607feat2_1.hjhtml.

³ Gilbert, J.B., Agile Compliance Management Systems for Environmental, Health & Safety, April 2005.

⁴ Gilbert, J.B., Agile Compliance Management Systems for Environmental, Health & Safety, April 2005.

⁵ Kroll, Per and Walker Royce, *Key principles for business-driven development*, IBM Rational, IBM, 15 October 2005, www-128.ibm.com/developerworks/rational/library/oct05/kroll/.

⁶ Miller, Randy, The Dynamics of Agile Software Processes, Part I: Characteristics, Borland Developer Network, Revised July 15, 2003.

⁷ Wikipedia, Iterative and Incremental Development, Software Development Process Series, <http://en.wikipedia.org>.

⁸ Alter, Allan E., "August 2006 IT Organization Survey, IT's Future is Brighter than You Think," *CIO Insight*, September 6, 2006. See <http://www.cioinsight.com/article2/0,1540,2012840,00.asp>.

⁹ Wikipedia, Key Performance Indicators, <http://en.wikipedia.org>.

¹⁰ Department for the Environment, Food and Rural Affairs (DEFRA), Environmental Key Performance Indicators, Guidelines for UK Business, 2006. See <http://www.defra.gov.uk/ENVIRONMENT/business/envrnp/envkpi-guidelines.pdf#search=%22environmental%20KPI%22>.

¹¹ Definition borrowed and modified from Mosaic, Inc., *Defect Management Process*, <http://www.mosaicinc.com/defectmanagement/risk.htm>.

¹² Wiegars, Karl, "Know Your Enemy: Software Risk Management," *Software Development*, October 1998, reprinted (with modifications by author), <http://www.processimpact.com>.

¹³ McKinsey & Company, *Building a nimble organization: A McKinsey Global Survey*, Web exclusive, June 2006. See www.mckinseyquarterly.com/article_page.aspx?ar=1808&L2=18&L3=30.

¹⁴ Moore, Geoffrey A. and Regis McKenna, *Crossing the Chasm*. Harper Business, 1991 (Revised 2002).

¹⁵ Keller, Erik, The Death of Packaged Apps? Wapiti LLC, May 26, 2006, www.sandhill.com/opinion/editorial.php?id=83.

¹⁶ Reasons to Buy Rather than Build Internally, CIOview, www.cioview.com/solutions/buy_vs_build.html.