Agile EAI Methods: Minimizing Risk, Maximizing ROI

Agile development methods have been shown to help companies realize business value early and often on software projects of all sizes. Today, practitioners of agile are quickly discovering that the approach is also well suited to enterprise application integration projects.

By Bob Hunter, Director, Insurance Practice Group
Martin Fowler, Chief Scientist, and
Gregor Hohpe, Senior Architect
ThoughtWorks, Inc.

More and more insurance companies are looking to Enterprise Application Integration (EAI) as a way to leverage existing IT assets to improve profitability and overall competitiveness. But many are doing nothing more than looking because insurance IT leaders recognize that EAI strategies involve a serious commitment of precious resources.

And therein lies the root of the problem with most enterprise-wide initiatives like EAI: How can companies minimize risk and maximize ROI on large and highly complex IT projects? It's a tough question that becomes even more difficult considering the fact that virtually any insurance carrier's IT environment is subject to frequent change. Not only do IT leaders have to deliver solutions that provide improved time-to-market and profitability—they have to do so in a way that simultaneously accounts for mergers and acquisitions, an ever-changing regulatory climate, and ongoing internal and external requests for new functionality.

Challenging? Yes. But very realistic, thanks to the recent emergence of a set of practices that make the prospect of tackling EAI initiatives more manageable. These practices—known as agile development methods—have been shown to help companies realize business value early and often on software projects of all sizes. Today, practitioners of agile are quickly discovering that the approach is also well suited to EAI projects.

Here's how an agile approach to an EAI strategy allows insurance carriers to pursue enterprise integration in a way that harnesses the power of change, delivers end-user satisfaction, and favorably impacts ROI.

EAI: More Ups than Downs

In the most basic sense, EAI uses messaging (a method of moving information around) to allow existing systems and applications to exchange data with new applications. The big advantage of EAI is that it can save companies a great deal of time and money by leveraging core IT assets. We'll delve into the challenges associated with EAI a little later.

In the insurance industry, EAI has become increasingly important. Today's crucial business functionality is spread across a wide range of custom and packaged applications. In addition, internal and external customers are increasingly demanding a single, customer-centric view into the business—which often challenges a company's existing systems.

EAI strategies center on the selection and integration of software called "middleware." EAI uses middleware to link diverse systems through a common communications interface. This integration technique allows insurers to improve customer service capabilities and more quickly capitalize on new market opportunities.

One reason EAI is gaining in popularity is because it extends the functionality of legacy systems without requiring the organization to replace existing systems or retool major components. It's also generating excitement because new business functionality can be added directly into a middleware layer. That means business processes can be automated without the need for major overhauls of legacy systems.

The real beauty of EAI is that existing systems can continue to do what they do best. Yet at the same time, new technology can be added to take the enterprise to the next

level. What's more, EAI vendors have moved far beyond simple messaging solutions. Some EAI tool suites, for example, provide the ability to manage workflow, manage documents and even create portals. In addition, EAI solutions now make it easier to adapt to standard protocols, such as Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP).

Of course, EAI is not a technological panacea. EAI implementations tend to be very large initiatives that require the integration of a host of applications and legacy systems across multiple platforms. As such, it's not unusual for companies to spend hundreds of thousands of dollars, if not millions, to implement an EAI strategy. In addition, larger EAI projects frequently take a year or more to complete. All the while, EAI strategies involve infrastructure work that is virtually transparent to business users, making it easy for some business leaders to lose sight of the overall mission.

Bob Hunter

Martin Fowler

Gregor Hohpe

Traditionally Speaking

The traditional approach in an EAI implementation is to plan most, if not all, of the process and business requirements up front before the integration work proceeds. This predictive approach also tends to create the mindset that it's critical to stick to the established plan at all costs as the project moves forward.

Predictive approaches grew out of the need to bring order to the early and often chaotic days of custom software development. Today, predictive methods continue to have a place, which is usually when technical and user requirements remain fairly static. However, business and technology is rarely, if ever, static. That's especially the case in the insurance industry, where constant change is driven by some of the factors referenced earlier (ever-changing regulations, mergers and acquisitions, etc.). Additionally, the pace of change has increased dramatically, and the ability to leverage technology to secure a strategic advantage is paramount. These points are especially worth noting when tackling complex EAI projects.

Let's say, for example, that a large carrier with multiple billing and policy administration systems uses an EAI strategy to implement a packaged CRM solution. Let's also say the project is estimated to take 13 months to complete. Using the traditional approach *(see Figure 1)*, the carrier might spend more than five months in the analysis and design stage. From there, the work to install new software, create a new Web-based front end, add adapters, and integrate the system to communicate and work in concert with the legacy environment occurs in the latter phases of the project. The new system is then deployed at the end of the 13-month period.

EAI extends the functionality of legacy systems without requiring the organization to replace existing systems or retool major components. It also allows new business functionality to be added directly into a middleware layer.

There's no question that a traditional approach to implementing this theoretical CRM solution can and does work. However, it can lead to serious headaches—not to mention budget overruns and lost opportunities.

"Big-Bang" Challenges

One of the main challenges with the traditional, "big-bang" approach is that it's very difficult to determine whether progress is being made throughout the project, and more importantly, whether or not things are going well. Of course, you will have your answer when you flip the switch the end of those 13 months. But in an age when ROI is king, not many insurers can afford to wait a year or more to learn whether or not their precious resources were put to good use.

Another difficulty with big-bang approaches involves business customer and end-user satisfaction. On just about any significant IT project, it's a safe bet that new internal and external customers will emerge mid-project, demanding business functionality that could not have been considered during the initial analysis and design phase. The risk is that the 13-month EAI project *as designed* might not fit the needs of the organization and users six months or more down the road.

There's also a tendency to over-engineer IT infrastructures up front because people tend to think of all of the things they might need to do with the new technology at some future date. The result is an overly complicated infrastructure that could very well be out of sync with the actual needs of the applications to be added.

Finally, EAI is often designed to radically transform an enterprise for

Figure 1

Traditional Development Approach

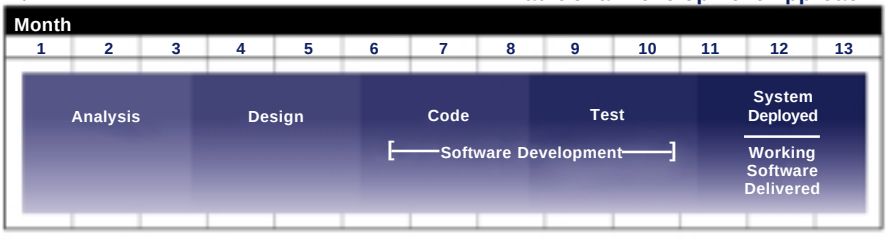
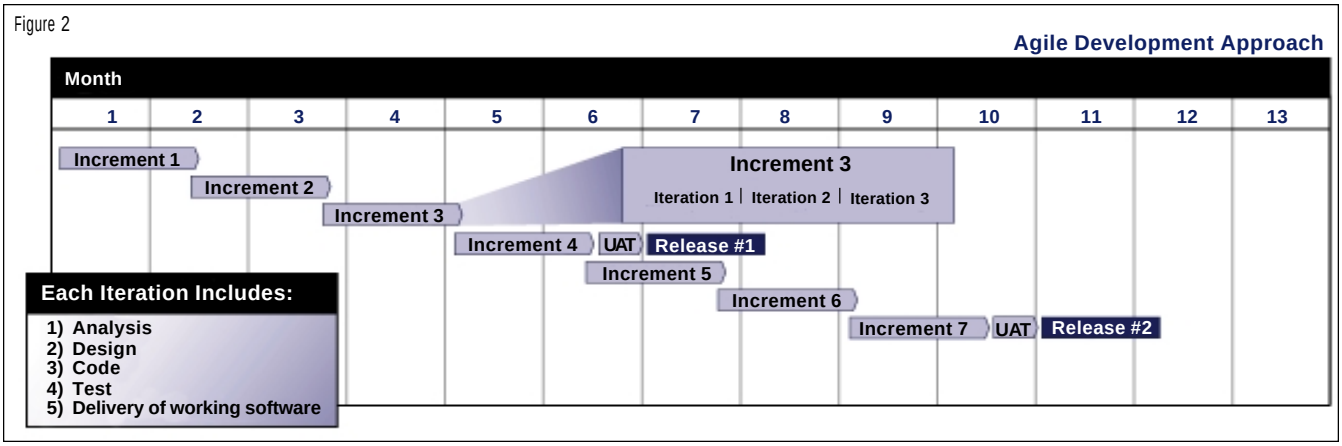


Figure 2



the better. However, any radical transformation that occurs at the flip of a switch can have an unexpected impact on an organization if not thoroughly tested, such as the introduction of an automated process that leads to added costs rather than savings. For example, automating tasks in one area of the business might increase transactions in a separate part of the system. As such, it could inadvertently extend the nightly cycle and throw it off schedule.

But none of this takes away from the fact that EAI can prove to be an enterprise-transforming initiative that delivers true business value. All that's needed is a smart way to get it done, which is why it makes good business sense to borrow a few pages from the agile approach to software development.

Agile and EAI Go Together

Interest in and the use of agile methods has increased dramatically in recent years. Today, more global leaders than ever in a variety of industries are using it to move forward intelligently and effectively on complex IT initiatives. (They're also achieving impressive results, but that's another story.)

The term "agile" actually describes a number of software development methodologies. These include Extreme Programming (XP), SCRUM, Crystal, and others. The methodologies share common characteristics that lend themselves to an "agile" process (i.e., lightweight, easy to adapt). While based on highly disciplined processes, agile methods are inherently *adaptive* (vs. predictive). This enables development teams to embrace changing business and end-user requirements. Aside from this basic premise, agile practices that contribute to the success of EAI initiatives include iterative development, ease of design, business/user partnering, and rigorous and frequent testing.

When compared with traditional methods, it's safe to say that agile takes the planning process to a higher, more flexible level. With Agile EAI, business and user requirements are broken down into small increments that can be developed in short periods of time (or iterations). This iterative approach allows planners to estimate the amount of time and effort required to successfully complete each segment of a project.

Earlier, we looked at how a traditional approach to EAI would have a company tackle a 13-month EAI project. If the same project were tackled using Agile EAI, development teams would do a complete cycle of the process in increments. Each cycle includes analysis, design, code, and testing (*see Figure 2).* Bits of working software are then released in a controlled setting before the full system is deployed.

The key benefit to this approach is visibility. By moving forward incrementally, decision-makers can determine whether progress is actually being made. If, for example, specific functionality cannot be completed within the second or third month, it's a very good sign that schedules were overly optimistic. But rather than struggling for a way out of the situation, the Agile EAI approach anticipates change and enables plans to be modified.

Along these same lines, ease of design supports iterative development. Rather than creating a complete infrastructure at the beginning of a project, focusing on small, incremental deliveries early, including limited business functions, allows the system to be tested throughout the development cycle. Additionally, the infrastructure can be refined while it is still practical. Plus, the concept allows the infrastructure and the applications to evolve together so that the architecture is aligned more closely with the needs of the enterprise.

Connecting Business and Technical Functions

Agile EAI also stresses close partnerships with business leaders and end users throughout the project. Essentially, agile says it's not a good idea for business people to come up with a list of requirements and go away for months or even years. Instead, close collaboration allows for short and rapid feedback cycles. In turn, course corrections can be made based on inevitably new requirements.

Once again, the process enables business leaders and the development team to go back to the original plan and determine the consequences of any particular change derived from new requirements. From there, a collective decision can be reached about whether to make any changes. The result is that both the enterprise and users get what is needed—rather than what was originally wanted. At the same time, the rapid feedback cycle ensures the project is not overbuilt. In other words, change works in favor of the organization, not against it.

Meanwhile, the ability to deliver rapid iterations without introducing defects dictates the need for rigorous and frequent testing. To do so, Agile EAI techniques often involve a continuous integration process that incorporates automated testing suites. The suites execute all unit and acceptance tests multiple times a day in an automated fashion, making regression testing fully automated.

Testing means that IT teams have a very good chance of detecting and

fixing mistakes that are inadvertently injected into the code base. Of course, correcting mistakes on an evolving infrastructure is a challenge. With Agile EAI, however, any mistakes are diagnosed and corrected much earlier in the process—often the same day—thereby minimizing the chances that costly errors will go undetected.

The long and short of it is that Agile EAI prevents many of the breakdowns that are common to EAI projects that use a traditional, predictive approach. Ultimately, Agile EAI blends discipline with adaptability, thereby enabling companies to reduce risk and deliver ROI sooner on major integration projects.

Making a Good Thing Better

The importance of a well-defined enterprise integration strategy and the right selection of EAI tool sets cannot be overstated. The best bet is to define an initial strategy that incorporates steps from vendor selection all the way through solution deployment. Tapping into outside systems integration expertise, along with deep insurance domain experience, can help address the need to:

- Clearly define common business objects
- Semantically and structurally map metadata
- Implement custom adapters
- Develop solutions that bridge multiple EAI vendors' tools

Agile EAI can be a powerful ingredient in a successful enterprise integration strategy. The method offers leading insurance companies the chance to improve specific processes incrementally, and thereby, more rapidly realize significant business value. It also lays the foundation for additional systems to be rapidly integrated in the future—even as business and customer requirements change, new front-office applications are added, new businesses are formed or acquired, new relationships are forged with external partners, and new products and services introduced.

As effective as EAI can be, taking an agile approach to its delivery makes a good thing even better. ❑

About the Authors:

Bob Hunter leads ThoughtWorks' Insurance Practice. Before joining ThoughtWorks in 2000, Hunter served as a director for Cap Gemini Ernst & Young U.S.'s Insurance Global Market Unit. Before CGE&Y, Hunter served in IT leadership roles at Great American Life Insurance, Progressive Insurance, and Continuum Systems, Inc.

ThoughtWorks Chief Scientist Martin Fowler is a leading authority on Agile Methods such as Extreme Programming. He has been a featured speaker at dozens of software development conferences and seminars worldwide and is particularly known for his work in patterns, UML, agile methodologies, and refactoring. He has written four books: Analysis Patterns, Refactoring, UML Distilled, and Planning Extreme Programming.

Gregor Hohpe, a senior architect with ThoughtWorks, has helped a variety of Global 1000 companies deploy enterprise integration solutions. Hohpe has also authored a number of articles and papers on application development and systems integration, and frequently speaks at technical conferences.