



Return on Agility: Financial Perspectives on Agile Development

Published in Cutter IT Journal, Vol. 20, No. 10

Agile software development approaches provide organizations with realizable benefits that are applied in the financial services community every day. By understanding the underlying principles, agile practitioners can better collaborate with business leaders who place decisions into a financial context. This article provides real-world examples of agile practices that provide three financially oriented benefits:

1. Efficient use of capital
2. Creation of exercisable options, which can also be thought of in terms of liquidity
3. Risk management and control

Achieving Innovation Through Integration

The financial services industry has used IT innovation to enhance revenue and contain costs across all of its main product lines, to the point where each step of the value chain is now heavily automated. Financial information systems have moved from a back-office role of capturing and recording transactions to playing a vital role in the front office in decision support, risk management, deal initiation, and — in the case of algorithmic or “black box” trading — actually initiating and executing transactions.

Increasingly, the frontier for innovation comes from efforts that cut across product and service lines — consolidating common processes as services, integrating siloed processes, and creating an enterprise view of

counterparty activities, investment risk, and opportunity. Together, these efforts provide business leaders with the tools to intervene across the boundaries of traditional IT applications. Consider the following recent initiatives:

- A major brokerage seeks to offer its hedge fund and asset management clients accurate information about their transactions and holdings within hours of the close of their local exchange. The brokerage also wants to enable its clients to flexibly group activity by account and accurately calculate portfolio performance (including tax effects) across all types of investments.
- A global commodities trading company needs to understand the net difference between its pending purchases and sales of each commodity by each of its operating subsidiaries so that it can most effectively hedge its risk through the futures markets.
- A multinational insurance underwriter needs to be able to assess premium income and loss exposure to geopolitical and climate risks across its many operating units, as well as to track net premium income and claims.

All of these companies face the same dilemma. Their dominant position in their respective markets is due to their effective implementation of business and IT strategies that were focused on the demands of individual markets and product niches. This decentralized approach has resulted in a balkanized IT environment, which makes it especially difficult to pursue an innovation strategy based on information integration and “loop closing.”

To pursue innovation through integration, IT leaders need the participation and buy-in of stakeholders across the entire enterprise. Project budgets often run to the tens of millions of dollars, and multiyear development cycles are typical. In addition to being costly, these efforts often need to displace mature, siloed applications that handle enormous volumes of data and conform to very high standards of reliability, availability, security, and response time. Their implementation involves extremely high levels of business risk.

Projects of this scope require especially high levels of management approval. Traditionally, this has meant the preparation of a business case that illustrates the investments that must be made, the specific features that will be delivered, when they will be delivered, and a schedule of paybacks (i.e., the cost savings and revenue realizations that will result). Once approved, these projects are expected to report their progress against the plan, using traditional project metrics such as earned value and cost and schedule variance to

gauge their success. Success itself is considered the delivery of expected benefits according to budget - a result that is seldom attained.

Agile Methods: A Dynamic Approach To IT Investment

Even when they are meticulously planned and executed, command-and-control “waterfall” projects often end up failing simply because they require strict adherence to plan and can’t adapt to a changing business environment¹. In contrast, when agile practices are applied to enterprise-wide efforts, it becomes possible to reduce both the risk and duration of technology investments while increasing their returns by allowing businesses to capture unforeseen opportunities.

The paradox is that, outside of technology, most companies already employ agility as they pursue a business strategy through adaptive tactics. Financial companies, in particular, commit their capital toward strategies and reward their managers for net results, not execution according to plan. They use a dynamic view of investments that controls risk not by demanding a fixed return or a specific set of buys and sells, but by employing a strategy to make considered “bets” and controlling overall risk through diversification and the use of “options,” hedges, and other techniques that limit the cost of the losers and maximize gains from the winners.

These same concepts — the use of options to hedge risk, or to inexpensively purchase “upside” advantage, as well as the ability to redirect capital to obtain the highest net return — can be applied to IT projects. The cases that follow, based on actual experience at our client companies, show how financial companies have been able to do this using agile methods.

Case 1: Using Agility To React To Changing ROI

In our first case study, the use of agile methods provided the ability to divert development staff to other important priorities while the business reacted to a potential change in the ROI of a major project. Freeing up capital (in this case, development capacity) and investing it at a higher rate is the underlying financial benefit that resulted from agile practices. Traditional methods, which would have left the capacity idle, can be viewed as having an option that could not be exercised, meaning that the value could not be realized.

The subject of this case study, Company A, handles a high volume of financial transactions between institutions and individuals. As part of the services it provides, the company produces printed statements

and enables the viewing of information online. Company A formulated a business case that determined that shifting these functions to a lower-cost outsourcer would yield significant payback.

Shifting printing out of house was a major investment decision that required executive approval and oversight. Funding for development changes, process modifications, and elimination of a printing facility was secured and the project was initiated. It turned out, however, that changes to Company A's software were required in addition to development of software by the outsourcer. Partway into the project, the executive in charge was informed that the outsourcing shop could not meet the promised system or capacity commitments.

Because the outsourcer could not provide reliable estimates on dates or deliverables for several weeks, the executive faced a difficult dilemma. Under a traditional development methodology, she would either have to request that the project be cancelled and then restarted or wait until the estimates were provided and recast the plan. Because the outsourcer could not commit to capabilities, capacity, or time frames, the return on investment was completely unclear. Fortunately, Company A was a fairly mature agile shop.

As the executive navigated the uncertain waters of the changing financial possibilities, the development team adapted to the change. The first course of action was to package the existing stories from the project such that the features neither disrupted the existing print processes nor caused wasted development effort. The second path was classic agile: continue adding business value as the landscape evolves. Developers were assigned the story backlog of requested features in other systems. These stories were "burned down" from the backlog as decision making on the outsourcing project was clarified. By gradually delivering the functionality that was "the next most important" to the business, there was no disruption in the flow of development, and the business realized sustained value while the major project was reviewed.

Once the outsourcer reevaluated the situation, it determined that it could not meet its initial promises and recast its plan. Only a portion of the functionality was outsourced, so the story list for the project was subsequently reduced. The result was a scaled-back project that still yielded ROI and a consistent flow of high-value features from development. This level of efficiency was only possible through the use of agile practices.

Case 2: Using Agility To Capture Opportunity

In this case study, the use of agile methods allowed the firm in question to pursue a consistent IT plan while adjusting strategy to take advantage of a new market opportunity. The ability to exercise an option on a new opportunity and trade out of a position with a gain were the underlying financial principles applied.

As a financial company that is part of a very large institution with both consumer and institutional touchpoints, Company B processes a large number of financial transactions and deals with numerous counterparties. The organization operates in a regulated environment, and it was building compliance features into its core software platform when a large, market-driven opportunity arose. This opportunity forced a choice between compliance and potential revenue — a difficult but familiar decision.

As a growth business, Company B had a keen interest in seizing the opportunity. There were both established and emerging competitors in the market who were aggressively fighting for market share across multiple product lines. These competitors announced their product intentions early, and the market was awaiting a response from the firm.

Given sufficient time and internal capacity, Company B would have pursued the new revenue source as an incremental investment. However, ramp-up time and an increasingly shrinking window of opportunity did not allow for this luxury. Normally, the situation would have presented a loss in one form or another.

Under a traditional governance structure for software development, there would be two major courses of action. First, the existing compliance release could be completed, tested, and promoted into production. The new opportunity would then be put through the process. In this scenario, the potential for squandering the opportunity would be significant. The second alternative would be to unwind the partial compliance efforts and build the new functionality. The compliance code would then be merged into the new codeline. The waste from rework or merging code would be costly.

Company B's adoption of agile practices afforded it a better course of action. The existing compliance features could be moved into production with manual processes instituted to achieve compliance. The new opportunity could then be immediately pursued, and the option to complete a fully automated compliance solution could be exercised at a future date.

This is exactly the path Company B chose. While the market opportunity was being more completely defined, final compliance iterations were completed. By the time pricing, release dates, bundling and

packaging, and all of the initial marketing activities were completed, a compliance release was moved into production. The development team was then free to prioritize, develop, and release new features.

By utilizing agile practices, Company B was able to realize benefit from the investment in the existing development stream while trading into another, more attractive investment. It still retained an option to complete compliance development at a future date. The ability to simultaneously trade out of one investment and into another was only possible because of agile practices.

Case 3: Using Agility to Control Risk

This last case provides an illustration of how an agile approach can be used to control risk — hedging against potential adversity by “buying put options” (i.e., small investments that ensure against much larger losses that might occur).

The investment banking unit of Company C, a large financial institution, sought to develop an Internet portal for its capital markets clients. The portal would draw on many sources of financial data, research, and market data and present customers with a highly personalized and flexible view. The project had an extremely aggressive time frame; the investment bankers who sponsored the project insisted that a working site be deployed within eight months, with enhancements added on a regular basis for at least the following year.

The sponsors’ outlook — which involved a fixed funding commitment in pursuit of flexible objectives that maximized delivered value in a short time frame — was highly conducive to an iterative development approach. The project team chose to employ a process based on the Rational Unified Process (RUP), using timeboxed development phases. While this approach was quite structured by the standards of the investment bank, it was at odds with the traditional approach followed by the central IT group, which was responsible for hosting all Internet applications as well as ensuring the security and integrity of the entire network.

Before the central IT group would agree to host any Web application, its process called for a complete security review of the application’s security architecture as well as all of its connections to other systems. This would ordinarily take place at the end of the design phase. This requirement posed a number of risks to the project. The first was that if the team waited to hold the review until the elaboration (or design) phase of the final iteration, it could not meet the target date. A second risk was that, whenever the review itself was

held, it would extend the duration of the design phase, delaying construction. Finally, should the review raise any issues with the design, more delay would result.

To address these risks, the development team chose to make the security architecture a priority in the first iteration and to address any feedback from the review as requirements to the second iteration. This enabled the review to take place early and in parallel with other development.

Developers worked with the business to define a story that included all of the functions of user access, authentication, and authorization, as well as the retrieval of application data from a single point on the internal network. This story was completely abstracted from the content or function of what was displayed. An application architecture was identified that supported these functions. It was this “design” that was reviewed about two months into the project.

Following the review and approval, the infrastructure group proceeded to purchase and configure servers and set up a portal with a functional Internet presence, albeit without any functional content or commercial users. This early deployment provided an additional benefit — basic monitoring processes revealed serious reliability problems with the overall Web infrastructure. During months seven and eight of the business project, the central IT group undertook a crash program of upgrades to stabilize the entire Internet infrastructure, and the investment bank portal was able to debut, on schedule, in a stable environment.

Key to the success of this approach was the scope of the initial user story, which included the full set of end-to-end interactions envisioned by the business sponsor. This approach was “deep” in terms of systems layers affected, but narrow in terms of apparent business value. By purchasing this story over one that delivered a functional welcome page, the sponsor effectively purchased a put option, thus protecting his large portal investment against delays in implementing a stable and secure Web infrastructure.

In the year after the portal’s launch, content and data from many bank systems were added to the portal, with all access coming through the network pathway derived from the initial story. No additional security reviews or penetration studies were needed because the new connections did not change the overall network security model. Hence, the early investment in a robust, generalizable, initial story also served as a “call option,” enabling the business to make future investments at less than their market price.

Conclusion

Viewed from a financial perspective, agile methods are applicable to enterprise-class projects for a number of key reasons:

Efficient use of capital: Agility allows project capital, which is overwhelmingly represented by development costs, to be applied efficiently throughout the lifecycle of the project. The ability to adjust the use of capital midstream to maximize return is the essence of agility.

Creation of exercisable options: Agility provides the business with options that can be exercised without significant time lag or transaction costs. Unwinding investments in development under traditional methods often involves a complete write-off of the effort. Agile development provides realizable value throughout the course of the project, and this leads to options that have value.

Risk control: Frequent iterations and early deliverables contain risk. The ability to understand and mitigate risks based on early releases has a real value to the business².

When making the case for using agile methods for an enterprise project, IT leaders have an opportunity to express the value of an agile approach by recognizing these financial impacts in the business case. Seen as an investment, agile projects have short durations, more certain return on investment, and, typically, less “value at risk” than conventionally run projects³. As with any approach, agile methods should be applied where appropriate in terms of the problem at hand as well as the organization’s ability to execute using the methods. Finally, while agile development proponents may face cultural challenges within IT, especially with regard to infrastructure, they may find unexpected allies among their business sponsors.

¹ Ambler, Scott W. “The Dire Consequences of Fixed-Price IT Projects.” *Dr. Dobbs Agile Newsletter*, 16 April 2007.

² Bailey, Duff. “A User’s View of Risk.” *Cutter IT E-Mail Advisor*, 22 August 2007.

³ Waters, Kelly. “10 Good Reasons to Do Agile Development.” *AllAboutAgile.com*