Agile Software Development: An Executive Overview

Most business critical decisions within the enterprise are laden with both opportunity and risk. Software development is no exception.

The management is usually far removed from the technology, tools and culture of software development, which creates the necessity of a knowledgeable human interface between those two worlds: At the core of software development, the way of communication and vocabulary is often very different from what C-level executives can understand and want to hear.

Launching a software project begins with traditional obvious screening steps. Know-how, experience, track record, success stories and references qualify or disqualify a developer and any experienced manager can work through this process easily. However, the next level is much more critical and requires a thorough understanding of the business of software development.

Agile Development

The most effective software developers today employ a core set of principles and methodology under the general label of “Agile.”

Agile is not an exotic approach anymore. It is widely recognized by cutting edge software development houses and market research firms. For example, Michael Blechar, Vice President of Gartner’s Applications Architecture Research division, recently wrote in a research note that Agile has become a “mainstream, mature and proven set of development methods.”

Agile caters directly to the unique challenges of every project in a rapidly evolving technology environment. Under disciplined and experienced management guidance, an agile programming team organizes its workflow to accomplish the optimum deployment of programming resources to match the client needs and goals.

The agile approach is based on a variety of select, structured processes that provide maximum flexibility and adaptability to the development cycle. Agile employs iterative practices aimed at the earliest possible delivery of system components, which minimizes the chance of problems at final delivery. To

achieve this goal, the development relies on a team-focused effort that cycles tasks rapidly through iterations within the team structure. As a result, a functional version of the software is available very early in the development process and potential problems are flushed out continuously.

The agile approach has several natural advantages.

• It encourages adaptive product management throughout the development process, yielding the client a more powerful and versatile product than originally envisioned. Traditional standard software project management methodologies, referred to as “waterfall” approaches, are driven by initial specifications and are limited by those.

• Agile yields less, yet more targeted code. It is often 75% more compact than software produced in more traditional ways. This means fewer chances for bugs and system failures, easier manageability of the code and programs that are vastly easier to maintain and extend.

• Agile methodology removes the guesswork from a project as it requires ongoing direct client involvement. There is continuous knowledge exchange and transfer and vastly improved odds that the final product will be exactly what the client wanted. Customers can expect much more visibility into the progress of the project, which in effect results in a product that reflects customer and business needs much more than a traditional waterfall development.

• Improved efficiency on both the developer and the client side directly result in cost and time savings.

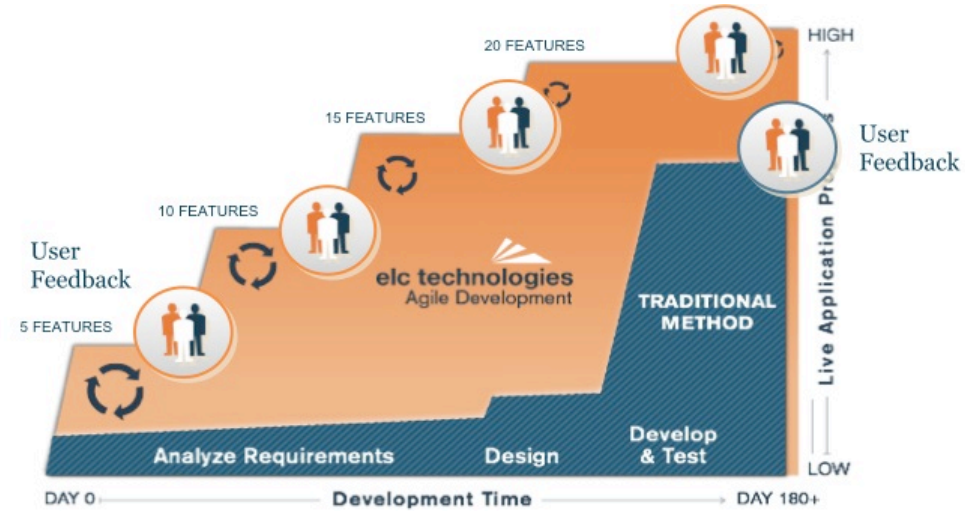
ELC – An Example

ELC, a Portland, OR based software firm, has embraced a custom recipe of Agile methodologies, based on their experience of building enterprise software. Dylan Stamat, head of development, notes that their core investment in Agile practices is key to their success.

A typical software project at ELC begins with the development working with the client to organize the tasks to be completed, prioritize them, and agree on a pace for deliverables.

This foundation provides the client with working software to enable an evaluating of the progress on a regular, weekly basis (or anther interval specified by the client.) As the demand for additional features arise, the client prioritizes them in

order to understand the tradeoffs in the workflow. The client then becomes the active leader of the project roadmap, and, as priorities shift, the changes in the direction are owned by the client rather than the developer. Modifications and improvements are driven by the client's needs, not the developer's way of working.



Early satisfaction as well as early success, are outcomes of this approach. ELC remains an active guide, advising on the tradeoffs of changing original plans and priorities. The client can tap into more than 100 years of combined development expertise from the team of ELC professionals, who maintain a weekly iteration of review processes to avoid any discontinuity in the overall workflow.

Traditional (Waterfall) methodologies cascade from one discrete phase to the next. They require an inflexible set of specifications to begin with. Each phase must be completed before the next one is launched. This rigidity all too often leads to a final product that doesn't deliver what the client initially asked for.

ELC's agile approach offers a shorter from concept to product cycle than any other form. The project can quickly modify paths, shift directions and can make midstream alterations to the specs with little impact on the process and little or no disruption to the full development cycle.

ELC's approach to Agile is based on the following principles:

- Quick startup time for projects
- Fast delivery: speed and iteration keep the pulse of the project healthy as well as deliver the product quickly
- Agile project managers who have been practicing Agile for a long time
- Belief in core principles: success rates are directly proportional to the involvement of those who believe in it to the core

- Build a team based on those core principles
- Developers are assigned to a project along with project manager from the beginning through to launch.

What makes ELC's agile process different is the fact that developers are not rigidly focused on doing things one way. The agile process is extended to additional levels to adapt to different business environments and cultures. ELC is equally adept at working with large corporations and small startups as size is not a determinant of success.