

VersionOne provides software and services to companies transitioning to or scaling iterative and agile development efforts. VersionOne's application, *V1: Agile Enterprise* is available as a hosted or locally installed solution and simplifies iterative project planning, tracking, collaboration, and reporting. Lightweight requirements management, release planning, iteration planning, project tracking, and task management integrate to provide visibility into today's rapidly changing development projects.

Introduction

Anyone who has ever been responsible for managing software development projects knows that *software is not easy*. Successfully coordinating and dealing with project sponsors, customers, team members, technology issues, and changing requirements challenges even the most experienced project leader. Leading projects in an environment that embraces both rapid delivery and change can prove even more daunting. Yet individuals with the desire to change the fundamental rules of the software game and accept the empirical nature of software development are faced with numerous opportunities. In order to capitalize on the evolutionary nature of agile development, today's leadership community must also focus on five key aspects associated with of agile development: *Courage, Context, Course, Cadence, and Cost*.

Courage

Also one of the four primary values identified in Kent Beck's *Extreme Programming Explained*, courage takes on a much broader and more strategic meaning outside the boundaries of programming. Software development requires many interfaces - customers, other project teams, customer support, professional services, external stakeholders, human resources, and many more - and the confidence to step up to the plate and be willing to enact positive change in the face of tradition can be a risky, but ultimately very rewarding experience.

Agile development is definitely not for the faint of heart. This does not mean that project management won't ultimately be simplified, but rather that any new way of doing business requires practice and hard work. Agile leaders cannot be afraid to fail, especially during early iterations. With agile development, at least failures are typically limited in scope to a few weeks, at which point you can reevaluate the situation and adapt accordingly. These early iterations should be used to learn, adjust, and stabilize.

Teams, and people, generally get good at what they practice. In agile development, planning, estimating, testing, and delivery, occur every few weeks as opposed to once every year. As a result, teams quickly develop a rhythm. By focusing on removing obstacles that get in the way and allowing this rhythm to emerge, agile leaders set the stage for improvement, predictability, and success. While still in the heat of software delivery battle, a vast majority of agile teams look back after six months, see the improvement, and wonder how they ever could have successfully delivered software any other way.

Context

With much of the fundamental project infrastructure (scope, priorities, estimates, schedules, risks, etc.) in a state of flux, it has never been more important to steer and manage actions and decisions within an overall business context. While functional value can oftentimes drive the details of a project, business values need to help drive project goals. Force hard decisions about the business and project context as early as possible. Get as simple as possible answers to key questions such as:

- What is the vision for the project?
- What are the primary goals and business drivers of the project?
- What are the values that should drive key project and product decisions?
- What are the expectations of project sponsors and stakeholders?

The answers to these questions serve as the basis for future decision-making, as well as a thread that can prevent a project from spiraling out of control. Additionally, the degree of visibility afforded these priorities can serve as a foundation for managing conflict and stakeholder negotiations. Many teams document answers to the above on a single page or project web page, and keep them visible throughout the life of the project. These answers often serve as a bar to which you will be held accountable.

Course

Context and course are complementary ideals. Whereas context defines overall circumstances, course defines direction and progress. Just because some agile approaches reference not looking in detail beyond one to two iterations, do not be lured into believing that longer range planning is not necessary or valuable. Longer-term software plans serve as a roadmap for important interim business and project decisions. Iteration, milestone, and release plans remain critical components in planning and measuring progress. Keep in mind that as you plan further out, confidence levels diminish, but a minimum three- to six-month roadmap serves as a continuous reality check in the face of a changing environment.

Make a concerted effort at least every iteration to revisit objectives, reconcile overall direction, review time-frames and deadlines, and communicate variances. Reality has proven that rarely does time get made up on software projects. This is especially true in the world of agile development. In agile development, change is accepted and even fostered for business reasons, so in addition to delivering early and often, be prepared to communicate early and often. Unlike traditional projects environments where the feedback loop can span quarters or even years, many more successful agile project teams communicate directly with key business and management stakeholders as often as each iteration.

Cadence

I would like to take credit for first using the term in this context, but its initial utterance came from Gary Evans of Evanetics, Inc., back in 2003 in a conversation we were having about the virtues of agile development. At VersionOne, we have consistently used the concept of cadence, or rhythm, to communicate one of the most beneficial effects of agile development. Most successful agile teams get into a powerful groove that can significantly benefit team confidence as well as overall project reliability.

Similarly, teams' deep grasp of iterative and incremental delivery in its complete sense can serve as significant risk mitigator. Instead of delivering once a year or more, agile teams deliver working, tested, installable software every iteration. In this type of environment, unexpected loose ends greatly diminish, defects are much easier to manage, integrated builds are second nature, and production deployment becomes much more streamlined and foolproof. This is not to say that some problems will not arise, but those that do will be much more manageable.

To early practitioners, this team rhythm is a goal to strive for as early as possible. To experienced practitioners, this rhythm greatly simplifies the task of leading agile development efforts. Many agile teams we have worked with achieve a much higher degree of predictability than with traditional methods. Simply having a quantifiable, accurate history of estimation and delivery can tremendously simplify planning and improves reliability.

Cost

Even agile development cannot escape the associated financial implications of software development. The challenge, and advantage, of agile development is the opportunity to justify the cost with business value and customer benefit throughout the process, as opposed to once at the very end. In addition, instead of managing projects as cost centers, agile development also provides the opportunity to view software development as an investment, with a goal being to achieve the highest possible returns earliest in the development cycle.

While you may not have the same direct influence over business benefit that you do over cost, the opportunity to allow the business to extract value sooner rather than later, continuously align business and technology goals, and adjust to changing business dynamics is of tremendous value. It is because of this aptitude for change that financial considerations must continue to remain front and center. While the cost model for most agile development projects is fairly straightforward, "cost per iteration" is not something that many organizations have either experienced or understand. Fortunately, many consulting companies and software organizations have made the transition to a simplified project costing model,

multiplying the approximate cost per person for the entire team by the length of an iteration to derive on-going financial impact (i.e., \$80,000 per iteration). While not as sophisticated as traditional staffing and resource allocation models, this type of analysis and justification can be just as accurate.

Conclusions

There are so many things that today's project leaders can pay attention to during software development projects. The key is to focus on the fewest things that matter, simplifying and streamlining those aspects of software development [outside of working software which should be a given] that deliver the greatest value to the business and the team. Especially in an arena of high-change, rapid delivery, and continuous feedback, the need for leadership has never been greater. While the physical time required for actually planning and tracking a project may lessen, the time commitment associated with coordinating, communicating, and guiding in a rapidly evolving environment can be amplified exponentially. By maintaining a laser-focus on the incremental delivery of business value and at the same time giving appropriate attention to the five areas referenced in this article, hopefully the process of leading in an agile world can actually be simplified.

About the Author

Robert Holler is president and CEO of VersionOne (<http://www.versionone.net/>), a provider of integrated agile life cycle management tools for companies adopting agile development processes. Robert has more than seventeen years of enterprise software experience in the CRM, ERP, HR/Payroll, and the ALM markets. His career in the software industry includes working as a consultant, developer, tester, technical writer, manager, and executive. You can access his blog at <http://www.agilechronicles.com/>.