An Introduction to Working with Pivotal Labs

How Working with Us is Different

When we take on a project, we like to hit the ground running with an experienced team, ready to begin executing immediately on your product vision. As the vision grows and changes, our team adapts. As you hire your own development team, we’ll weave them into our team, teaching them the code base and all our techniques as well. We want to make sure that when you’re done working with us, your team can move into the next phase of development with confidence and keep things running smoothly. Once your team is self-sufficient, we can weave our developers back out and move on to the next engagement. You’re always welcome back, for that next new feature initiative, to add a little extra horsepower, or to help out with some thorny issue down the line.
Introduction to Our Process

Flexibility
One thing that’s special about us is that we can vary the team size on demand. During the early stages of development, projects often have large fluctuations in workload. We can ramp the team size up or down depending on development objectives and financing constraints. We can develop at full speed for a week, a month, or a year, and then stop on a dime—and stop burning capital—for as long as you need to close a funding round, get more traction in the marketplace, or decide the next round of features.

Our clients are done with us when they decide they’re done. Your business goals and product needs are always in the driver’s seat.

We are a strategic partner for our clients. We are not an outsourcing firm but instead strive to make our clients self-sufficient. We believe our approach offers a unique balance between the short-term need for execution and the long-term need for sustainability.

Agile Development and Extreme Programming
Pivotal has been using and improving agile development methodologies since the very beginning. The particular flavor of agile process Pivotal favors is called Extreme Programming, or XP. It combines Test-Driven Development, Continuous Integration, short iterations, and Pair Programming to radically improve software quality and flexibility while reducing cost and time to market.

Test-Driven Development
When we build applications, we start by writing tests for every feature we implement. Only after the tests are in place do we write the feature code. The discipline of writing a failing test and then writing the feature code to make it pass ensures complete test coverage and a more reliable product.

Continuous Integration and Continuous Deployment
When we check in our code, a Continuous Integration server checks the code out again and runs all the tests to make sure that the code will work correctly in production. When tests fail in the Continuous Integration environment (i.e. when the build breaks) the focus of the team shifts to fixing the build, ensuring that defects don’t creep into the codebase.

We also continually deploy new features to our demo environment. The customer can see the new features in a realistic setting and immediately make sure the features are implemented as intended and that the actual behavior is desirable.

Short Iterations
We keep our development cycle to one-week iterations. This keeps features from drifting out of control and gives quick feedback on each new feature being developed. Larger efforts are always broken down into week-long pieces, increasing focus and minimizing risk.

Pair Programming
One of the first things you’ll notice about coding with us is that we pair program. This means two developers sit down at one computer and write code together. Perhaps surprisingly, two developers can produce more and better code by pair programming than by working individually. There are a number of reasons why this seemingly counterintuitive approach works. For one, pair programmers spend more time doing productive
work. They are constantly refining each other’s ideas, discarding bad ideas quickly and saving projects from poor choices that can cost weeks of work down the road. In addition, with enough eyes all problems are shallow. If we look where a typical developer spends their time, much of it is spent stuck on relatively trivial problems, often ones that same developer would breeze through on a different day. When two developers collaborate, one can press on when the other gets stuck and the flow is never broken; progress is made consistently and rapidly. We’ve found it the best way to work and use it on all of our projects, particularly the ones we pay for ourselves. The quality of the code produced is superior and the time spent to produce it is shorter.

**Agile Inception**

We believe the best architectures, requirements, and designs emerge from self-organizing teams, not from months of requirements gathering. When we start a project, we organize the team with an Agile Inception session. The goal of the two day session is to air all the ideas and assumptions about every aspect of the project so that the entire team starts off on the same page. Agile Inceptions are a powerful, ultra-fast way for the team to internalize your vision and the structured discovery process enables the developers to begin coding as soon as the Inception is finished.

The two-day session explores goals, risks, users, roles, use cases, as well as a few other key topics. These sessions are intensive workshops facilitated by Pivotal and are based on our vast project experience and collaborations. Product features are prioritized immediately and a concrete list of next steps are imported into Pivotal Tracker, allowing the entire team to begin delivering core product functionality.

**The Effect of Team Size**

Developers are most effective when they collaborate full-time, bringing more perspectives to bear on the problem at hand. To this end, we always deploy programmers in pairs. How many pairs is up to you, the customer, but we provide some guidance on the decision.

One pair is the smallest team you can choose. Typically this is the right team size at the beginning of a project when the team is just getting up to speed on the particulars of your application. One pair is enough to do solid work, but as the project moves forward it is useful to change up who pairs with whom. With one pair, there is only one pairing, but with two pairs, there are six possible pairings. Fresh pairings keep the project lively, and communication overhead for a two-pair team is negligible. Doubling team size from one to two pairs typically doubles productivity outright.

Three-pair teams offer even more pairings, and typically produce just under three times as much work as a single pair team, but three-pair teams can start to suffer from the inability to parallelize tasks. This depends on the scope of the particular project and how well the work can be segmented.

With more than three pairs communication overhead starts to become a cost, but when time to market is the key consideration, teams as large as five pairs can make sense, particularly for short stretches when the needs of the project are sufficiently segmented. At five pairs, you should expect to add one additional full-time person in a leadership role to coordinate the efforts of the team.

**Pivotal Tracker**

Our main planning tool is Pivotal Tracker, an award-winning web application we developed in-house that embodies many of our XP principles. We use it to keep track of priorities and estimates and to track releases and deadlines. Both developers and customers use it to keep track of where things are in development, and to keep development focused on the highest priority features at all times.
Story Breakdown and Task Estimation

It is the development team’s job to work with the customer to break down features into small, user visible units of work called stories. These stories are the key to keeping focus on visible, tangible, customer value. Once the breakdown is complete for the next set of features, the development team can estimate the effort involved in completing each story.

Velocity

We use a simple, point-based system for story estimation, where the points are measures of relative complexity. Developers are much better at estimating the complexity of a problem than the time it will take to solve that problem.

Velocity is a measure of how many points a given team completes in a given week. Experience has shown that a given development team will work through a fairly consistent number of points on a weekly basis, and this strong central tendency means the velocity for a team becomes very predictable after the first two or three weeks of development. This will give you a very early indication of how long a given set of features will take, and give you the visibility to make informed choices about what features you choose to implement.

Your Role in the Process

As the customer, you play the key role in the development process. Your involvement with the team shapes what gets done, and when. You determine the backlog of features to be developed, and you determine their relative priority. You make those decisions informed by the relative cost of the features in your backlog, since you can see at any moment the cost in terms of points. Pivotal Tracker allows you to rearrange stories as you see fit, immediately displaying the impact each choice has on your schedule, and by extension, on cost.

One thing that may surprise you is how our development teams respond to changes in priorities. They respond by doing whatever it is you ask them to do in whatever order you choose. They are confident that they’ve exposed the cost to you, and they trust you to make informed choices about the highest priority. For many of our customers this is a revelatory experience. For us, it’s just the sign of a healthy relationship between the development and product teams.

Done Means Done/The Life of a Story

Developers use Pivotal Tracker to log the state of all stories as they work on them. When they complete the work on a story, and all the tests are green, they check the code into source control and mark the story finished. Several times a week the team will push the current code to a demo server for story acceptance.

When developers have pushed a story to the demo server, it’s ready for you to accept or reject. We don’t consider a story done until you have seen it working in a live application and Pivotal Tracker will not credit the points for a story to the current iteration’s velocity until you have accepted it. And, since we only deploy when all the tests are green, you know you can deploy what you’ve seen into production at any time.

Moving On With Confidence

The engagement will eventually end, but your time with Pivotal ensures your team can excel on their own. Your developers have been pairing with our developers throughout the process and know the code. Our highly effective XP process, as well as our tools, are now yours to use. We’ve set you up in Pivotal Tracker and have your Continuous Integration running on a server you take with you at the end of the engagement. Our team has passed along their best technical and process knowledge and your team is ready to continue the excellence. Pivotal will always be here for quick questions or return engagements, but we sincerely hope we’ve given you everything you need to make your vision succeed.