

Chapter 1

Why the Web Industry Needs XP

*We are all in the gutter, but some of us
are looking at the stars.*
—Oscar Wilde

Web development is an adolescent, unique in its requirements and unparalleled in its potential. Like most adolescents it wants to be accepted as an adult as it tries to pull away from its parents. If it is going to reach its full potential, it must take a few lessons from the more seasoned world of software development.

In the bad old days, the Web industry was experiencing growing pains in its progress toward maturity. Technologies were changing under our feet, and we still hadn't developed best practices for completing projects. It seemed as though too many customers were paying for the industry to learn how to do things right. Not surprisingly, there were a lot of unhappy customers, and it is always the unhappy ones who seem to talk. As a result Web development

was developing a bad reputation. What were we doing wrong? Quite a few things, it appears:

- ✧ We tried to be all things to all customers with insufficient expertise.
- ✧ We often failed to deliver on time or on budget.
- ✧ We tended to develop adversarial customer relationships.
- ✧ Our projects didn't always meet customer expectations.

Trying to Be All Things to All Customers

Most Web firms evolved from a software or an advertising background, tacking on Web development as an additional service to customers. We have found that the industry is made of both strong



Wearing too many hats, trying to be all things to all customers, leads to bad projects.

technical shops that lack a solid understanding of branding and marketing issues, and strong design firms that are weak in technology and user interface. A 2000 Forrester Research report¹ on e-commerce integrators mirrors our findings, even in the largest Web development companies. It graded 150 Web developers and found that in the categories of marketing, strategy, design, technology, and business practices not one scored higher than 70 percent across the board and most scored much lower.

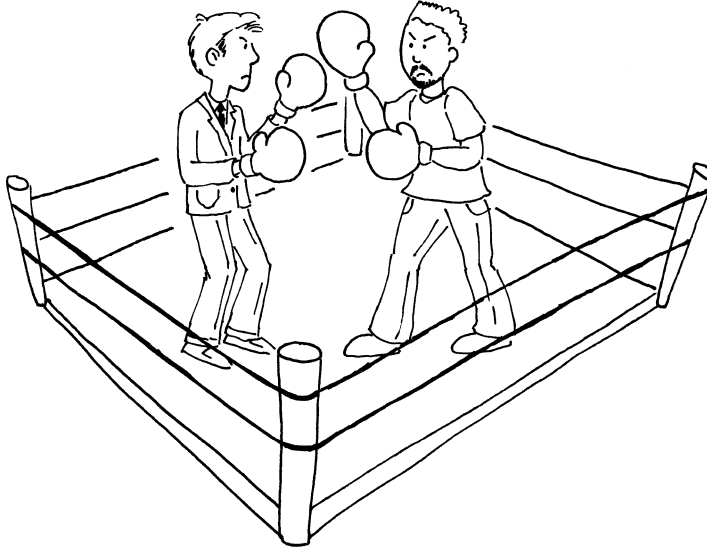
Projects Not Delivered on Time or on Budget

The Forrester report showed that project management was an area of overall weakness, with 80 percent of respondents indicating that their projects were delivered late and over budget. Finding good project managers was the bane of our existence for years. It seemed to be the hardest position to fill, with few candidates having relevant experience in project management or Web development or having strong customer skills. As you can imagine, the chances of launching a site on time became a matter of luck. To date, few developers have put into practice a project management methodology that accurately tracks and adjusts projects and takes into account the multi-disciplined nature of the industry.

Adversarial Customer/Developer Relationships

Traditionally projects were (in most cases still are) a fixed-price, fixed-deliverable agreement based on an initial estimate of the overall

1. Averby, Christine Spivey, and Paul Sonderegger, with Harley Manning, Julie Meriger, Aaron Hardisty, and Sadaf Roshon, *eCommerce Integrators Exposed*. Forrester Research, June 2000.



Customers and developers may come to “blows”
about expected deliverables.

work involved. However, the chances of developers being able to guess the costs of a project accurately were minimal, and the challenge was to make the project fit the estimate. This could only result in an adversarial relationship, in which the customer consistently took the widest interpretation of what was included and the developer responded with the narrowest, with both sides taking contrary positions on the expected deliverables. The inevitable changes to the project often caused bitter fights.

Unsuccessful Projects

Web companies consistently sought projects with larger budgets under the mistaken assumption that this would automatically boost

their profits. But without defining the metrics for determining project success, they couldn't define the requirements and functionality that would signal project completion. Simply delivering a Web site on time was not enough. Moreover, unless it solved the customer's business challenges and/or provided a means to measure the impact of the site on the customer's business, it was not a successful project. Unfortunately, as projects made it to production the developers lost sight of the original objectives and wasted many expensive man hours on unnecessary or unwanted tasks because they forgot why they had been hired in the first place.

The XP Solution

After analyzing the mistakes the industry was making, we needed to come up with practical, cost-effective solutions. Clearly, Web development teams needed to adopt new development methodologies. These methodologies needed to stress customer involvement in the day-to-day decisions within the project, include incremental milestones of weeks rather than months, and allow for ongoing refinement of estimates based on actual progress. Project management would have to be redefined, not as software or advertising but as a new entity consisting of multidisciplined teams with close coupling to customer stakeholders.

We knew of XP in association with software development and wondered if it could solve the problems in Web development. Given the Web industry's essential differences from the software industry, could XP be as good a fit?

Web Development versus Software Development

XP methodology could potentially deal with a number of Web development problems. XP involves the customer, integrates teams, generates maintainable code, and, most important, divides the responsibility for project success between the developer and the customer

and so creates an atmosphere of trust. But would XP map directly to the needs of Web projects? Extreme Programming was developed to resolve problems in software, not Web, development. There are a number of key differences between the two.

Teams

Web projects involve multidisciplinary teams made up of graphic designers, copywriters, Flash programmers, server-side programmers, interface programmers, testers, and project managers. Team members can't be isolated because decisions made by one affect the others, and the intersection and overlapping of skills make it impossible to set strict boundaries of responsibility. A series of questions and answers illustrates this:

Question: Who is responsible for how the page displays?

Answer: The designer and the interface programmer.

Question: Who is responsible for how a function works between the server and the customer?

Answer: The server-side programmer and the interface programmer.

Question: How does the Flash element pull a weather forecast from a third-party application?

Answer: Better ask the designer, the Flash developer, the interface programmer and the server-side programmer. Oh, and you'd better ask the customer about everything!

XP is very good at getting programmers to communicate among themselves and with customers. Web projects require a myriad of new disciplines. How does XP cope with this decision-making interconnectedness?

Support for Multiple User Environments

Most software projects create deliverables for one platform at a time. Different software versions are developed to handle different user environments. For example, the latest release of an application will have a version for Windows, a version for UNIX, and perhaps a version for Mac, depending on its audience. Unfortunately, in Web we don't have the luxury of separating out the product. There is only one version of the Web site, and it must be able to support multiple browsers on multiple operating systems simultaneously, running multiple screen color and resolution settings, not to mention modem speeds. Worse, not only do the system requirements vary depending on the Web site audience but they can change from the beginning of a project to the end.

How does Extreme Programming allow for the varied support needed in one product?

Testing

Web projects require unique testing practices to account for multiple customers, and they have an emphasis on how things look that is often unheard of in software projects. Testers have to be able to test interfaces in totally new ways. For example, page layout, design, screen colors, and screen resolution are all requirements that can be tested only by eye.

How does XP deal with the need for sense-based testing?

Rapid Deployment

Software developers have the luxury of large releases. The cost of a new version of the software forces the customer to plan releases carefully and to make them substantial revisions. Web projects can be deployed as often as the customer wants. Web XP projects need to harmonize continuous integration with new release practices.

How does XP accommodate the need for frequent deployment?

.....

Customers

Software development needs a customer to set priorities, define the problem domain, and make key decisions, because it is the customer who understands the process that the software is trying to emulate. Web projects are generally the development of a corporate Web site or system and are often marketing vehicles new to the organization. Thus, it is hard to find an expert to consult. Web customers look to their developers for far more guidance than XP allows.

How can Extreme Programming help educate customers and increase customer satisfaction?

Quality

Many original Web sites were low-quality, brittle masses of code. Few Web developers used an object-oriented approach to development. Most used procedural languages, which made refactoring code or making changes to it much more difficult—often sites were simply redeveloped each time enough changes were requested. Over time sites get worse, not better. New design patterns pioneered for software have to be created for Web sites.

How can XP help to improve quality?

XP Web Development

In providing satisfactory answers to the questions just asked, XP has had a great influence on the way we have developed projects over the last two years. We have developed a number of mutations that serve our projects well and that make the use of XP for creating Web sites second nature to us. In many cases we have simply extended an XP practice to nonprogrammers; in others we have invented new practices.

In general we have found XP methods to be especially helpful in three key areas:

- ✧ Project estimating
- ✧ Customer relationships
- ✧ Release planning

Rather than define how Web projects should use XP, we will attempt to explain what has worked for us and to open the dialog between the Web development and XP communities.