

Case Study

SpreeRide, Lante and Extreme Programming

Introduction

Extreme Programming (XP) is a culture that promotes collaboration between technology and business teams. In this case study we will describe how XP was used to implement a system with fluctuating business requirements for SpreeRide Corporation. The XP process will then be compared to a hypothetical case in which a traditional software development approach is used. The timelines are illustrated in the included diagram and discussed in the narrative.

The Business Story

SpreeRide developed a well-founded business plan and decided to proceed with development of a system to implement that plan. SpreeRide was interested in approaching the development of the system from a new perspective, realizing that the traditional waterfall cycle was inadequate in its inflexibility, and that a dynamic process suited to the current business environment was needed.

SpreeRide and Lante engaged with a “test first” statement of work, consisting of minimum functionalities structured in the form of tests. This set of tests was structured as a MOSCOW list, (an acronym for Must Have, Should Have, Could Have, and Won’t Have), and this list was ordered according to business priorities at the time.

The only guarantee from a contractual perspective was that the items under the Must Have column were to be implemented during course of the project. The project’s initial release plan consisted of the “Must Have” items. It was commonly understood that once the “Must Haves” were implemented, the client would reprioritize remaining business tasks or present new tasks during the weekly planning game (attended by members of both business and development teams).

A Change in Client Funding

At the fifth week of development SpreeRide encountered a funding issue with one of their venture capitalists. What this meant for SpreeRide was that they could cease development and still have roughly sixty percent of demonstrable functionality. SpreeRide arranged a demonstration of the running system to their venture capitalists. The VC’s were so impressed with the amount of functionality developed that SpreeRide received additional funding and within a week was back on track. The additional funding allowed SpreeRide to continue development with a new fixed-price/fixed-time/variable-scope statement of work based on SpreeRide’s satisfaction and trust of the Lante team.

A Change in Direction

The new statement of work shifted focus from a client-specific system to a self-contained sales-tool to be run on a laptop computer. This change in scope did not hinder the development team, as they were only required to focus on a new set of stories.

A Change in Priorities

SpreeRide's CTO was an integral member of the development team acting as both technical point person for SpreeRide as well as the customer representative. Towards the end of the second phase of the project, the CTO transitioned the story ownership to SpreeRide's COO. This change in leadership meant that business priorities would be shifting, a condition that was par for the entire course of the project anyway. The impact to the team was further minimized by the nature of the weekly iteration, where the client specified the most important functionalities to be developed. Due to the high degree of discipline in applying the XP process, everyone's expectations were managed on a daily, almost hourly basis.

XP Development on the SpreeRide Project

Project Management

Each development day begins with an all-hands team/customer standup meeting. This provides unfiltered feedback to the client, and a transparent view into such things as team dynamics and state of the system.

Each development week began with a planning game. The planning game defines business requirements, technical tasks, and a test suite to define how the team will know that business requirements have been satisfied. During the planning game developers accept responsibility for completed specified tasks by estimating and signing up for each task.

Evaluating the Unknown

Assumptions regarding technical requirements are validated by "spikes" (a time-boxed investigation). If a requirement cannot be estimated with a high level of confidence by the team, then a spike is performed. For example, a spike might consist of evaluating competing COTS packages.

Team Structure

From top to bottom, this hierarchy consists of the Steering Committee, and the XP team. The Committee consists of executive level personnel. The XP team consists of the Coach (providing XP process guidance and enforcement), the client-customer (provides business input and decision making), and the development team.

Team Day

A typical day from the perspective of the XP team starts with the daily standup meeting where outstanding issues and questions are resolved. Then development begins in earnest with members pairing up and selecting tasks, writing tests, and developing code. Code is integrated periodically throughout the day where all regression tests are run.

Team Week

Every two weeks, in preparation for Steering Committee meetings, the team would develop a demonstration of the current system to date.

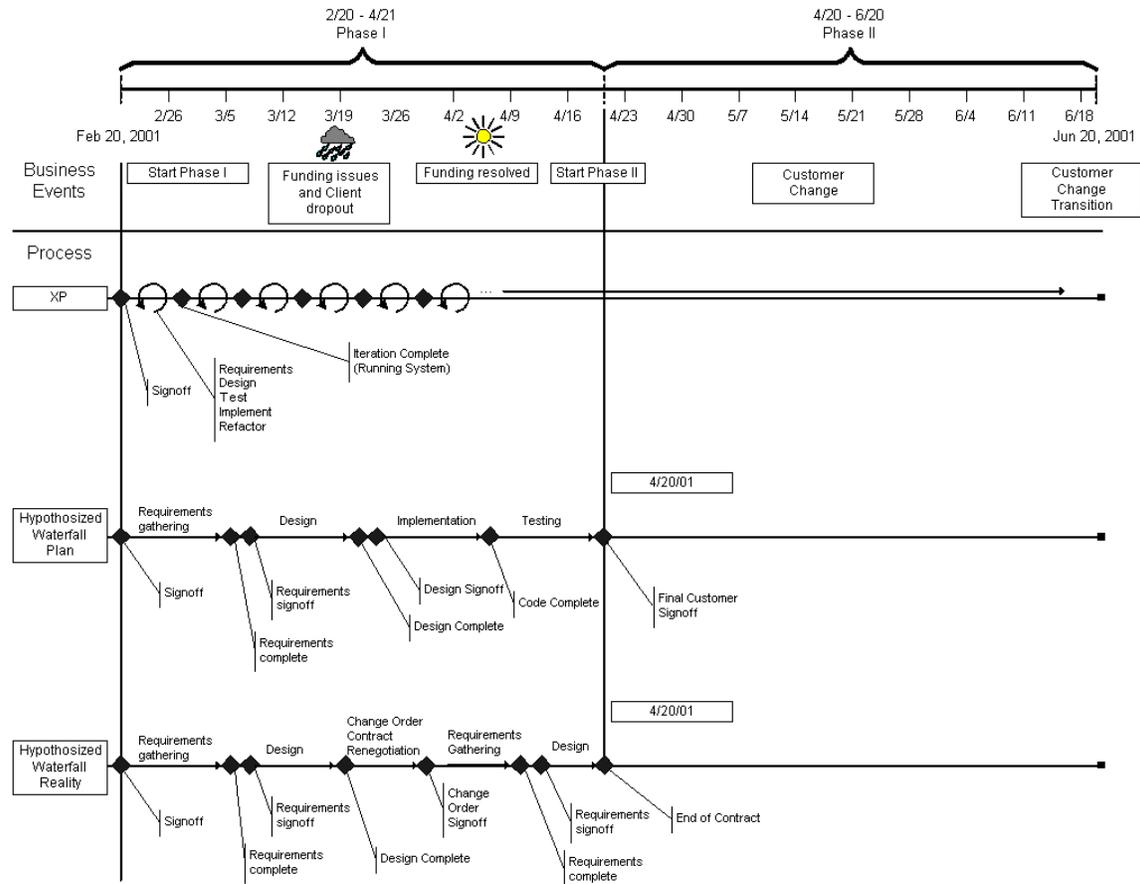
Team Impact Based on Business Events

Events, which might adversely affect the team such as a hiccup in funding or a change in managerial direction, have minimal impact on an XP team. In the case of a project coming to a grinding halt midway through a development effort, the client walks away with one hundred percent implementation of the most important features, while the development team remains satisfied with their craftsmanship in that they delivered a high-quality product.

A Traditional Development Methodology

Typically, a traditional waterfall approach would demand a team structure that fluctuates per phase, is more hierarchical, and requires more management due to the layering and coordination of various competencies. A project would move through a requirements phase, followed by design, implementation, and finally testing phases.

In our hypothetical scenario, the requirements team would attempt to gather all requirement details within two to three weeks of starting the project. The client would then be asked to validate or approve that all requirements had been gathered. The client might blindly approve that the requirements are satisfied, thereby potentially jeopardizing the foundation of the project, or may take an inordinate amount of time sifting through details, effectively stalling the project. Once requirements have been approved, the design process can commence. Substantial changes to requirements has a domino effect on the project, triggering a change request process and cutting into the time budgeted for design. For example, had SpreeRide's funding event occurred during the design phase, change orders would have been filed and timelines reestablished, a process taking several days to weeks. At this point in our story, the project would most likely have ended, because SpreeRide would have been unable to demonstrate a working system to their financiers. Emerging from this train wreck, SpreeRide would walk away carrying no more than a stack of requirements documentation and any design documentation completed up to that point.



Conclusion

Although the business scenario presented here might be considered unusual or extreme, today's business environment is nevertheless always changing, and demands a software development practice that changes with it. One of the reasons XP succeeds is that it fosters a collaborative rather than combative dynamic between organizations. In XP, expectation management is a built in feature, while traditional techniques require a dedicated effort to manage customer expectations. On the SpreeRide project, the client saw business value implemented on a daily basis. Had the project used the traditional approach, SpreeRide would have held a fixed set of expectations of a completed system that would have been shattered with the loss of funding. With XP, the team builds business value every day, thereby justifying their existence. XP hands over the reins to the client every week, allowing them to re-scope and change requirements. To build anything, requirements must remain fixed for a certain amount of time. With the waterfall process requirements are fixed at the end of the requirements phase, remaining so during the course of the project pending no further changes. In XP, requirements are fixed for a week's time, allowing the project to move at the same velocity as the business. Due to a disciplined team rigorously applying the practices of XP, SpreeRide was able to weather a storm and successfully develop their system.