Brady Field

Extreme Programming and its Development Practices

**Important points**

“In the core of the XP practices are programming activities, with strong emphasis on oral communications, automated tests, pair programming, storytelling culture and collective code-ownership at any time in the XP project.” [97]

“In short, XP promises to reduce project risks, achieve adoption of constantly changing business requirements/needs and improve productivity throughout the life of the system - all at the same time.” [98]

“The four XP values and those derived principles are a basis for building a discipline of software development practices.” [99]

“The Development strategy is a radically transformed format of the traditional view of the development process. Its motto is that in XP all activities are centered around programming, i.e. ‘everything you do in XP looks like you are doing programming’.” [100]

“XP encourages code production before having an overall investigation of all major system's use cases and domain models. They are all needed in order to generate conceptual design and overall implementation solution.” [101]

“Communications at any level/phase of the XP project seems to be the backbone of all XP practices. You are required to program in pairs with your colleague in order to approach unit tests and code production more dynamically and share someone else’s expertise/knowledge.” [102]

“The answer to question (a) above should then probably be negative if we think about the quality of software production in small projects where XP is highly applicable and possibly more efficient than any other approach. These are cases where we do not think about the architectural issues very much and could conform to and accept a very wise wording of Planning game practice under a Metaphor of a system‘ model, instead of using the word “architecture“, as expected.” [102]

“RUP is a generic process framework that can be specialized for any software system including different application domains, types of organizations and project sizes. Its component-based approach to software development is captured in use case driven, architecture centric and iterative/incremental processes.” [103]

“The strongest resemblance with XP could be in a highly iterative and incremental RUP approach adopted when delivering releases for any of the phases of its lifecycle.” [103]

“This paper has attempted to raise serious concerns regarding XP’s courageous approach to the software development process and its impact on software architecture solutions. However, the aim of the paper has not been to dispute any values that XP might contribute towards SE practices due to their relative immaturity, insufficient number of sources where one could learn how to use XP, and few examples where XP is put into practice.” [104]

**Disagreements**

“A software development process, which guarantees to build a software product or enhance an existing one is expected to offer effective guidelines and capture the best practices of software engineering (SE) disciplines in order to deliver an adequate solution, reduce risks and increase the predictability of software intensive systems.” [97]

A software product does not guarantee to a build a software product or enhance an existing one. In fact, it cannot guarantee this. A software development process is used in an attempt to maximize the chances of delivering an appropriate solution, but delivery or enhancement cannot be guaranteed.

“XP came to light within the last two years as an answer to all the problems we still experience when delivering software solutions.” [97]

There are two problems with this claim. First, there is nothing new under the sun. Second, there is no “silver bullet” in software development. This cannot be an answer to all the problems present in software development.

**Questions**

I understood everything in the article.