David Lambertson

**Extreme Programming and its Development Practices Summary**

**Important Points from Article**

* “Extreme Programming (XP) has attracted our attention through its fierce denial of many of our well-accepted software engineering practices which we consider today as a sound approach to the development of intensive software systems. XP has been declared to be a new way of software development: a lightweight methodology, which is efficient, low-risk, flexible, predictable, scientific, and distinguishable from any other methodology.” [97]
* “XP is declared to be a new way of developing software: a lightweight methodology which IS efficient, low-risk, flexible, predictable, [and] scientific... in brief distinguishable from any other methodology.” [98]
* “It is designed for smaller projects and teams of two to ten programmers, which result in efficient testing and running of given solutions in a fraction of a day.” [98]
* “XP addresses risks at all levels of the software development process, which requires communication of the XP discipline to programmers, managers and customers. XP project looks at problems/risks of the development process itself and derives solutions that dictate a set of XP activities.” [98]
* “XP uses values as the main criteria for a successful software solution. In other words, the XP values serve as a guarantee, which shows that an XP’s set of practices is taking the right direction.” [98]
* “The four XP values and those derived principles are a basis for building a discipline of software development practices. However, before practices are identified, XP gives a list of activities, which are derived from the XP principles.” [99]
* “The implementation of XP should be based on the strategies of the XP project management where almost all XP values, principles and practices are employed in order to deal with a particular strategy For example, the Management strategy that emerges from evaluation/use of Accepted responsibility, Quality work. Incremental change, Local adoption. Travel light and Honest measurement principles will guide us towards decentralised decision making and leave managers with Planning game practice, using metrics as the basic XP management tool.” [99-100]
* “XP strategies are put into practice through a lifecycle of an “ideal” XP project. It consists of a short initial development phase followed by a long-term simultaneous production support and refinement.” [100]
* “We have learned throughout the last 4 decades that the best SE practices are based on the up-front analysis and clearly specified conceptual models prior to any code production. XP practices directly violate this principle i.e. it almost looks like XP denies all the values of high level analysis and design activities which were prevalent in structured systems development lifecycles as defined in (DeMarco, 1979) and some 00 methodologies (Jacobson et al., 1999), (Rumbaugh et al., 199 1)” [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]

**Things I Didn't Agree With**

“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]

I agree with this statement because I think not everything should look like you are programming. When we are designing, the programmers could be writing code, but they don’t need to mask it to actually look like they are writing code. Design is a crucial aspect of software development. Without a design, it is hard to actually implement the code.

**Things I Did Not Understand**

I did not understand how the author was trying to compare Extreme Programming to the other life cycle models.