Summary of *Extreme Programming*

By Marc Novakouski

Description
Extreme Programming (also known as “XP”) is a popular software development process which grew out of the growing movement towards Agile processes[1]. The basic idea behind Extreme Programming is to strip out virtually all of the elements of the traditional software process to get to streamline development[1]. As a result, an XP project usually results in a software project in which software development is begun immediately, virtually no documentation artifacts are created, other than “user stories” which are written on index cards, and the project proceeds in an iterative fashion in which prototypes are created with the direct input (and sometimes help) of the stakeholders on a daily basis until the desired effect is created[1]. Finally, a key principle of XP is to expect and even “embrace” change, and XP does this by encouraging “refactoring,” or restructuring of the code on the fly on an as-needed basis.

Extreme Programming is essentially a conglomeration of a number of specific “Agile” software development practices and concepts[2] in an attempt to remove the non-essential parts of the development process[3]. There are 12 “core” practices that make up XP, which are separated into 4 key areas: Planning, Design, Coding, and Testing[3][4]. Some practices are considered to be “best practices” of the software industry, such as Continuous Integration and Test Driven Development[4]. However, several practices such as Pair Programming and the System Metaphor are more controversial, and are often excluded in practice[5][7].

Extreme Programming is considered to be a development method suitable for only certain types of projects, such as small projects, research projects, and projects dealing with new technology[4]. As a research design process, Extreme Programming can be a very effective tool for research based upon software, as stakeholder interaction is often very high, since the researcher is very involved in the process. Additionally, the iterative and documentation-light approach of XP allows very quick development cycles, fuelling further research.

Background
The seminal work on Extreme Programming was written by Kent Beck in 1999[6]. The practices upon which XP is built have existed for some time within the Agile process movement[2]. Interestingly, the project in which Kent Beck codified XP actually failed, but Beck was undeterred by the failure[4]. Many software projects use XP today, including the ActiveAd project by Connextra[7] and the EJB project by CQG, Inc[8].

References