Reactions to Dr. Robert's Lecture

It seems as though the adoption of Extreme Programming may have indeed saved the company in question from certain failure. The end product produced by their former means would have been late, buggy, and would not have met the needs of the customer. According to Dr. Roberts, many in the company were very skeptical of the switch, as would be anyone. However, the results of the switch could be seen very easily in the end. I feel as though the adoption of XP was a very good decision on the part of the company.

The project was in turmoil and quite obviously something needed to change. To reduce the number of errors (“bugs”) in the code produced the adoption of pair programming was a great idea. Two is nearly always better than one. One of the programmers functions as a safety net for syntax and semantics. Also, the introduction of unit testing was perfect in this case. Previously the team had relied nearly solely upon the testers to catch errors, who couldn't possibly catch everything. The unit testing helped to show exactly where the program failed, whereas before it took much much longer to find where the errors were

Along with the quality of the code, many of XP's features helped with the communication of the team. Communication is central to being able to integrate all of the smaller parts of a program and to decide in what direction the program needs to go. Not to mention, it saves time and effort for the programmer if some else is already familiar with something that you need to do and can provide support. Implementing the “bull-pen” and the stand-up meeting facilitated this well. Along with programmer communication it is extremely important to have good communication with the client. This will insure that the program is heading in the right direction and will satisfy the needs of the customer.

Organization and planning was also another need of the company that XP helped to bolster.
Using “stories” facilitated a way to plan the program and also break the problem up into smaller pieces to give to each programmer. Along with this, the “stories” provided a means of work estimation which helped to set deadlines and produce cost analyses. I'm sure that the company had tried to do this earlier, however it's quite obvious by their inability to produce the program on time that the old method was not working.

It seems as though the XP method was absolutely perfect for this company, as it improved the areas that needed the most improvement. Judging from the results, the switch to XP was a great idea for this company. However, the question arises whether this method should be everyone's method of choice. I believe that it may be for some, however I don't think it is something that can be applied to every case. XP may be counter-productive in a very large scale operation. Having bull-pens for 2000 programmers is quite obviously not going to produce the same results as a team of only 10. It seems as though a company's defects should be studied and discussed before the decision is made to switch to any method of programming. What is right for one, may not be right for another.