

Empirical Investigation of the Impact of Extreme Programming Practices on Software Projects

Lucas Layman
North Carolina State University
900 Main Campus Dr., Rm. 197
Raleigh, NC 27695
+1-919-513-5082
lmlayma2@ncsu.edu

ABSTRACT

Extreme Programming (XP) is an agile software development methodology composed of several practices that purportedly yield high quality and high customer satisfaction. However, there has been little formal investigation of these claims. We conduct empirical, industrial case studies to evaluate XP. Results from two case studies are presented.

Categories and Subject Descriptors

K.6.3 [Management of Computing and Information Systems]: Software Management — *software process*; and D.2.9 [Software Engineering]: Management — *life cycle, programming teams*

General Terms

Management, Measurement, Experimentation, Human Factors

Keywords

Agile software development, extreme programming, case studies

1. PROBLEM AND MOTIVATION

The introduction of Extreme Programming (XP) [2] into mainstream software development has been met with both enthusiasm and skepticism. Reports both extol the virtues and question the shortcomings of XP. Most often, these reports take the form of anecdotal success stories or lessons-learned from organizations that have adopted and/or modified some or all XP practices for a project. However, many organizations remain skeptical regarding XP's value.

For decision-makers, an empirical, quantitative investigation is beneficial for demonstrating XP's efficacy. Realistic, methodologically-defensible case studies and experiments are an effective means for conducting research that can be effectively communicated to the software development community. Case studies are valuable because they involve factors that staged experiments generally do not exhibit, such as scale, complexity, unpredictability, and dynamism [4]. However, Zelkowitz and Wallace [6] reported that less than 10% of papers in the respected software engineering journals and conferences they examined involved a case study.

2. APPROACH

For our case studies, we desired to examine business-oriented results of adopting XP practices, such as changes in productivity, quality, and customer satisfaction. Multiple case studies will be required to provide sufficient results before we can draw any conclusions about XP's efficacy. However, these early results add to the weight of evidence in support or in refutation of these propositions. Under the guidelines of the Goal Question Metric paradigm [1], our Goal is to build theories about whether the business-related results of a team change when XP practices are used. This goal was refined into Questions about: 1) pre-release quality; 2) post-release quality; 3) programmer productivity; 4) customer satisfaction; 5) team morale.

To guide and facilitate our research, we have constructed the Extreme Programming Evaluation Framework (XP-EF) [5]. This benchmark has been created for expressing the XP practices an organization has selected to adopt and/or modify and the outcome thereof. The XP-EF is comprised of three parts: XP Context Factors (XP-cf), XP Adherence Metrics (XP-am) and XP Outcome Measures (XP-om), as shown in Figure 1, and was constructed using the GQM approach [1].

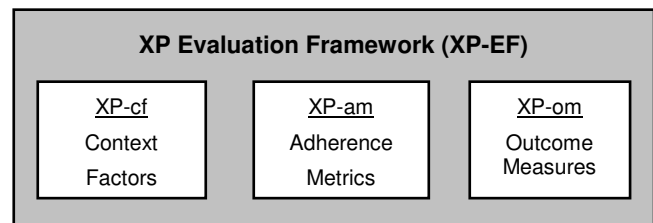


Figure 1. XP-EF structure

In the XP-EF, researchers and practitioners record essential context information of their project via the XP Context Factors (XP-cf). Recording context factors such as team size, project size, criticality, and staff experience can help explain differences in the results of applying the methodology. The second part of the XP-EF is the XP Adherence Metrics (XP-am). The XP-am use objective and subjective metrics to express concretely and comparatively the extent to which a team utilizes the XP practices. When researchers examine multiple XP-EF case studies, the XP-am also allow researchers to investigate the interactions and dependencies between the XP practices and the

extent to which the practices can be separated or eliminated. Part three of the XP-EF is the XP Outcome Measures (XP-om), which enable one to assess the business-related results (productivity, quality, etc.) of using a full or partial set of XP practices. Furthermore, we conduct interviews with team members and customers to help understand the team's adoption of XP and the customer's satisfaction with the project.

3. CASE STUDY RESULTS

We have completed the analysis of two case studies structured using the XP-EF. For the outcome results listed below, measurements are normalized with the old releases to protect private information. Both customer satisfaction and morale are assessed using subjective means (customer interview and developer survey respectively). These measures may be subject to influencing factors beyond the XP practices, though the interviews and the survey questions were structured in order to reduce the impact of these factors as much as possible. Our post-release quality results are limited by the lack of a measure of customer usage of the product during the defect collection periods.

3.1 IBM Case Study

A year-long case study was performed with a small team (7-11 team members) at IBM to assess the effects of adopting XP practices [5]. The team develops Servlet/XML applications for a toolkit that other IBM teams utilize to create products for external customers. This case study analyzed two consecutive releases of the same product: one release completed using a traditional, waterfall-like approach (old release) and the other completed using XP (new release). Through these two releases, this team transitioned and stabilized its use of a subset of XP practices. The use of a "safe subset" of the XP practices was necessitated by corporate culture, project characteristics, and team makeup. The team's XP-om results are outlined in Table 1. We note that the new release was approximately half the size (in KLOEC) than the old release, and that smaller projects are generally considered less complex.

Table 1. IBM Outcome Measures

XP Outcome Measure	Old	New
Pre-release Quality (test defects/KLOEC)	1.0	0.50
Post-release Quality (released defects/KLOEC 6 months after release)	1.0	0.61
Productivity		
User stories / Person month	1.0	1.34
KLOEC / Person month	1.0	1.7
Putnam Productivity Parameter	1.0	1.92
Customer Satisfaction	N/A	High
Morale (via survey)	1.0	1.11

3.2 Sabre Airlines

The second case study was conducted at Sabre Airline Solutions [3]. The Sabre team (Sabre-A) was also small (6-10 team members) and co-located, and had been using XP for approximately two years. The team develops a scriptable GUI

environment for external customers to develop customized end user and business software. Again, two releases of the same product were compared: the old release completed three years ago using a waterfall-based process and the new release completed recently using XP. The Sabre-A team's outcome measures are documented in Table 2. Both releases were similar in size, but the old release was developed over a span of 18 months while the new release was developed over a span of 3.5 months. The product was continuously tested during development, thus, the old release had a significantly longer pre-release defect collection period. Furthermore, the only the KLOEC/person month measure of productivity was available for this case study, which alone may not be a sufficient measure of productivity.

Table 2. Sabre-A Outcome Measures

XP Outcome Measures	Old	New
Pre-release Quality (test defects/KLOEC)	1.0	0.35
Post-release Quality (released defects/KLOEC of 4 months after release)	1.0	0.64
Productivity		
KLOEC / person month	1.0	1.46
Customer Satisfaction (approx)	N/A	High
Morale (via survey)	N/A	68.1%

In our case study comparisons, we observed that both teams increased their productivity and improved pre-release and post-release quality when using XP compared to the plan-driven approaches. We remind the reader that these results are based on two case studies in two particular contexts, and do not generalize beyond the realm of small, co-located teams. Our results can be used to build up the weight of evidence about XP, but do not yet offer any conclusions about the efficacy of XP practices. Two additional XP-EF case studies are pending analysis, and several other case studies are underway in Europe and the United States.

4. REFERENCES

- [1] Basili, V., G. Caldiera, and D. H. Rombach, The Goal Question Metric Paradigm, in Encyclopedia of Software Engineering, 1994, John Wiley and Sons, Inc. p. 528-532.
- [2] Beck, Kent, Extreme Programming Explained: Embrace Change. 2000, Addison-Wesley: Reading, Massachusetts.
- [3] Layman, L., L. Williams, and L. Cunningham, "Exploring Extreme Programming in Context: An Industrial Case Study," 2nd IEEE Agile Development Conference, Salt Lake City, UT, February 29, 2004, pp. in press.
- [4] Potts, C., "Software Engineering Research Revisited," IEEE Software, vol. No. pp. 19-28.
- [5] Williams, L., W. Krebs, L. Layman, and A. Antón, "Toward a Framework for Evaluating Extreme Programming," 8th International Conference on Empirical Assessment in Software Engineering (EASE 04), May 2004, pp. 11-20.
- [6] Zelkowitz, M.V. and D.R. Wallace, "Experimental Models for Validating Technology," IEEE Computer, vol. 31, No. 5, pp. 23-31, May 1998.