Towards Supporting Agile Practice Within The Libre Software Paradigm

Paul Adams and Cornelia Boldyreff (Supervisor)
University of Lincoln
Lincoln, England
{padams, cboldyreff}@hemswell.lincoln.ac.uk

I. INTRODUCTION

Individual agile methods have never been practiced as defined, in the same way that Royce's waterfall [1] model never reflected actual practice. Instead, practitioners adapted the core principles of these processes in order to suit their needs. Understanding this is key to appreciating the agile mindset. What does exist is a set of principles[1] which, when followed loosely, form the agile practices.

It is an important part of the agile mentality that the individuals within a project are more important that the process they follow. However, the individual methods do have their own identifying features that make them unique; for example testing must be performed before coding within eXtreme Programming (XP) [2]. However, if practitioners were to apply XP, exactly as Beck describes it, then they are probably not "doing agile" as they may not be following the process that suits their needs best.

One of the interesting features of the XP method is its requirement of a collocated team. This requirement is never particularly specified, but can be inferred from XP's lightweight nature coupled with its requirement of tight collaboration, such as pair programming. This requirement of a collocated team can limit the usefulness of XP in certain contexts, especially as it may not always be possible to have a customer always available on site for the development team. This is not acceptable as, within the agile mindset, the customer is just as much part of the development team as the programmers.

This research is concerned with the effective distribution of agile practice and, in particular, the XP method. This abstract addresses a particular case study of this research, applying distributed XP within the libre software paradigm. The work described here will only play one small part of a larger programme of research, investigating the effective distribution of agile practice.

II. MOTIVATION

Due to its lightweight nature and its focus on certain communication principles, we can infer from the XP agile method a requirement of collocation. This can be very restrictive and limit the usefulness of XP.

Within the agile methods, the customer is just as much part of the "team" as the developers. However, to always have a representative of the customer on site, as XP requires, can be difficult. In the perfect world the development team would uproot itself to be with the customer; in the past, great things have come from this [3]. In reality, this is rarely possible and the customer has to uproot to join the developers. This, although a key motivation for this research, is not too relevant to this case study.

Processes within the libre software paradigm are inherently distributed. Each project, as with classic software engineering, has its own process. As can be expected, some of these processes are more agile than others. The purpose of this case study is to experiment and evaluate the use of distributed agile methods, in particular XP, within the libre paradigm.

III. BEYOND THE LIBRE AND AGILE PARADIGMS

This project will investigate a new paradigm, the "Liberal" paradigm. It would not be sufficient to create a process that allows XP practitioners to fit into the libre paradigm, nor would it be sufficient to create a process allowing libre practitioners to fit the agile paradigm. Figure 1 shows how the "Liberal" paradigm is derived from both libre and agile paradigms. This new paradigm will provide an process framework for allowing agile practitioners to follow a libre process and vice versa.

The "Liberal" paradigm shall be formed on principles and practices shared by the libre and agile paradigms. This will create an environment in which distributed eXtreme Programming will flourish, but still be acceptable as a libre software process.

The distribution of XP is already a well researched area [4,5,6]. Although not as deeply researched, the application of agile methods within the libre software context has also been explored [7]. Despite all of this research, there has yet to be a process developed that supports agile and libre practices within both paradigms. Most of the research conducted so far has, however, produced useful (but not ideal) tools for supporting these practices.

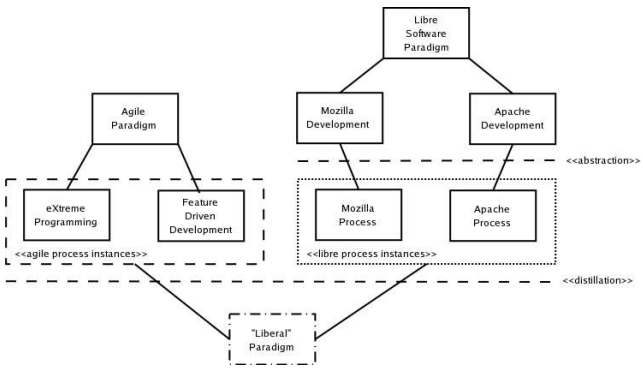

Fig. 1. The transition from agile and libre paradigms to the "Liberal" paradigm.

[1]The Agile Manifesto. http://www.agilemanifesto.org

IV. RESEARCH METHOD

It has been shown [8] that there is great difference between the libre software processes and, therefore, some are going to be more "agile" than others. At first, it appears that these differences are greater than those found between the agile processes. The first area of research within this project will be to develop a set of libre software process features, abstracted from current libre software projects. This layer of abstraction will provide one input set for the distillation, that will eventually produce the "Liberal" paradigm. It will be impossible to create an abstraction over all libre software project processes. Instead, a sample of projects shall be chosen according to their size (in number of active participants). Some large projects, such as Apache and Mozilla, have already been studied and this research can easily be incorporated into this project. Similar will be carried out on medium and small-sized projects to complete the abstraction. This project shall not be as concerned with an abstraction of agile processes as this is already provided, in the form of the "Agile Manifesto".

The "Liberal" paradigm will be generated as a distillation of the abstractions of the agile and libre paradigms; the resultant paradigm utilising the important and relevant features of the parent paradigms. At this stage, it is possible to make some predictions as to what the characteristics of the "Liberal" paradigm will be.

From the libre software paradigm the "Liberal" paradigm will inherit the low-level freedoms that libre practitioners are afforded. These are freedoms such as location, tools and to an extent, time. From the agile paradigm the "Liberal" paradigm will inherit some of the high-level principles of the Agile Manifesto, but with a resolution of conflict. For example, customer collaboration and code production are both important, but we cannot allow one to stand in the way of the other.

Any tools developed within the larger research of this project shall be used in order to help define a process within the "Liberal" paradigm. However, there shall also be top-down approach to this research, allowing for the tool's development to be affected by any hypothetical process.

The ideal method for evaluating this project would be based on groups of experienced eXtreme Programmers using the final system. It would be possible to evaluate the performance of any tools developed to support the "Liberal" paradigm in a goal-question-metric [9] environment. XP lends itself well to metric-driven evaluation because of aspects such as project velocity.

To thoroughly evaluate any new processes or tools produced by this case study, experimentation and metric analysis will take place within different contexts. Firstly, it will be required to see that the process and tools can support the use of XP within an open source project; the case study described in this abstract. It is planned for this work to be performed in conjunction with the EU FP6 project, "CALIBRE"[2]. It will also be important to see if software engineers, used to collocation can perform distributed XP effectively. It is planned for this part of the evaluation to be a collaboration with an industrial partner.

[2]http://www.calibre.ie

V. CONCLUSION

This case study is aimed at developing the "Liberal" paradigm, encompassing features of both the open source and agile paradigms. Further work will also help define processes that fit into this paradigm. These processes will allow for the effective distribution of agile practice in a manner that would allow libre software projects to use them. Processes within the "Liberal" paradigm may, in particular, facilitate the contributions of industrial contributors to open source; especially those organisations that already use agile practices.

VI. REFERENCES

[1] W.W. Royce, "Managing the development of large software systems," in *Proceedings of the 1970 IEEE WESCON*.

[2] K.Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2000.

[3] R.X.Cringely, *Accidental Empires: How The Boys Of Silicon Valley Make Their Millions, Battle Foreign Competition And Still Can't Get A Date,"* Penguin Books, 1996.

[4] M.Kircher, P.Jain, A.Corsaro and D.Levine, "Distributed extreme programming", in *Proceedings of XP2001*.

[5] F.Maurer, B.Dellen, F.Bendeck, S.Goldman, H.Holtz, B.Kötting and M.Schaaf, "Merging project planning and web-enabled dynamic workflow technologies," in *IEEE Internet Computing*, May 2004.

[6] R.Jota and A.Rito-Silva, "Supporting distributed extreme programming with adaptive workflow," in *Proceedings of the Workshop on Cooperative Support for Distributed Software Engineering Processes*, 2004

[7] M.Kircher and D.Levine, "The XP of TAO: eXtreme Programming od Large, Open-source Frameworks", in *Extreme Programming Examined*, Addison-Wesley, 2000.

[8] A.Mockus, R.T.Fielding and J.D.Herbsleb, "Two Case Studies of Open Source Development: Apache and Mozilla", in *ACM Transactions on Software Engineering and Methodology*, July 2002.

[9] R.van Solingen, *The goal/question/metric method: a practical guide for quality improvement of software development*, McGraw-Hill, 1999.