

Tool Support for the Effective Distribution of Agile Practice^{*} (Extended Abstract)

Paul Adams and Cornelia Boldyreff^{**}

Department of Computing and Informatics, University of Lincoln

1 Introduction

Agile methods are quickly gaining notoriety amongst software engineers. Having been developed over the past decade, they now present a mature, lightweight alternative to the “classic” approaches to software engineering. Although agile methods have solved some of the problems of established software engineering practice, they have created some problems of their own. Most importantly, we can infer a, potentially problematic, requirement of collocation.

The usefulness of the agile methods is restricted by this requirement of collocation and by the requirement of small development teams. If these requirements can be loosened then it would be possible to apply agile methods to a larger arena of software development. This research intends to extend the usefulness of agile methods by defining a new paradigm for software engineering practice, the “Liberal” paradigm and providing tool support for processes within this paradigm.

2 Tool Support for the “Liberal” Paradigm

This new paradigm is based on the principles of agility combined with the experience of libre software process distribution. This paradigm offers important advantages as it will encompass all important features of both agile and libre software engineering practice in order to facilitate the distribution of effective agile practice. This paradigm will provide a new set of processes that allow the distribution of agile practice within the libre software paradigm, but also potentially improve the performance of collocated agile teams.

In this research the intention is to develop a distributed software engineering support system that will allow the effective distribution of agile practice within the “Liberal” paradigm.

Tool support for the “Liberal” environment will be provided in the form of a plug-in for the Eclipse IDE. Existing support for distributed agile practice is often formed as a naïve, ad hoc composition of existing tools. Environments such as this can offer effective solutions to this problem but are restricted in that they have not been specifically developed for this purpose. Unlike other research

^{*} A complete edition of this paper can be found at <http://eprints.lincoln.ac.uk/48/>

^{**} Project Supervisor

where process has been the focus, the principle contribution of this research shall be a tool set for supporting processes within the “Liberal” paradigm. This tool set must be rich enough to support distributed agile practice, but flexible enough to allow libre software practitioners the low-level freedoms they are familiar.

There are three types of tool that will be developed within this project: awareness tools, communication tools and task support tools. All of these shall be encapsulated within an Eclipse plug-in environment. The tasks initially identified for support include distributed story cards, virtual daily meetings, component integration and most importantly, pair programming.

3 Research Method

This research is largely based on empirical process. After thoroughly researching the requirements of agile programmers and distributed software engineering, I intend to iteratively implement and evaluate features for the distributed agile environment. Once the entire system is complete a thorough evaluation of the system through experimentation is planned followed by improvement (where required) and then further experimentation etc.

This research is focused on a bottom-up approach, that is, the development of a tool for supporting distributed agile practice; the development of a process for this tool and ensuring this process fits within the “Liberal” paradigm are secondary. However, there is an element of top-down approach, in that it has been possible to form some high-level descriptions of the “Liberal” paradigm and processes within it.

From the agile paradigm the “Liberal” paradigm inherits some of the high-level principles of the Agile Manifesto. However, these processes need to be more adaptable than the current agile processes to ensure that the principles do not conflict. For example, it may not be desirable to allow customer collaboration to restrict code production. From the libre paradigm the “Liberal” paradigm inherits the low-level freedoms that libre practitioners are afforded. It is also important for the “Liberal” paradigm to inherit scalability from the libre software paradigm.

4 Conclusions

We can infer from many agile methods that communication must take place in a collocated manner. This project aims to allow the distribution of effective agile practice by providing tool support that fits the “Liberal” paradigm without constraining its usage within specific software processes.

The main focus of this research is the creation of a tool that aids the distribution of effective agile practice. This tool must not only support agile practice but also offer support for the communication and awareness overheads that distribution causes. It is intended that the tool developed within this research will aid the distribution of effective agile practice in a manner that is relevant to both industrial practitioners and libre software practitioners and thus broaden the usefulness of the agile methods.