

Building and Linking a Metaphor: Finding Value!

Frank Keenan¹, David Bustard², Namgyal Damdul¹, David Connolly¹

¹Department of Computing and Mathematics, Dundalk Institute of Technology, Ireland
{frank.keenan, namgyal.damdul, david.connolly}@dkit.ie.

²School of Information and Software Engineering, University of Ulster, Coleraine,
BT52 1SA, UK, dw.bustard@ulster.ac.uk

Abstract. Initially System Metaphor was included as one of twelve practices of Extreme Programming (XP) to facilitate participants' common understanding and description of the problem under consideration. However, the practice was removed in the second version of XP as practitioners found it difficult to understand, devise and link it to implementation activities. This poster describes how a metaphor can be constructed and linked to development by using practices associated with Soft Systems Methodology (SSM). Initial, positive, feedback from an educational exercise is presented.

Keywords: Metaphor, Soft Systems Methodology (SSM).

1 Introduction

XP evolved from a fixed set of twelve practices now described as a “process of experimentation and improvement” [1]. Despite its removal as an explicit XP practice this poster describes how describing a *metaphor* and linking it to development provides benefit. Here, parts of SSM [2], a well-proven method for analyzing problems including the development of software is combined with XP. Traditionally, SSM has been described as a seven-stage process. Problem situations are usually captured diagrammatically as *rich pictures*. Although subjective, with no rules defined for drawing them, they help achieve a shared understanding of a situation among interested parties. Models of ‘relevant systems’ are then developed, based on that knowledge and are expressed as *root definitions* and *conceptual models*. A root definition is a short textual statement that defines the important elements of the ‘relevant system’ providing a particular perspective on the system under investigation. A conceptual model is derived from the root definition by identifying the activities present or implied and their inter-relationship. These models provide a basis for further debate on the activities involved. Next, change recommendations are derived from the results and action to improve the situation undertaken is recommended.

This approach used *rich picture analysis* to build a metaphor and then detailed examination to develop a conceptual model. The activities of this model were then used to create user stories.

2 Evaluation

An evaluation study was conducted with two final year groups of Degree students. All had previous knowledge of agile methods, particularly XP, and were introduced to SSM and the combined approach. One team used only XP and the other SSM-XP. Both had access to a customer throughout development. In the first SSM-XP session the customer outlined the problem. After a discussion, drawing of a rich picture was facilitated and the various issues raised were easily and quickly clarified by the customer. This session was interactive with members offering suggestions, raising questions and amending the diagram. Finally, the rich picture was photographed and transcribed. Thus, the metaphor was created. The main challenge appeared to be reaching agreement on the problem rather than drawing the diagram. As the conceptual model developed more focused discussion took place. Essentially, one or two related activities formed an XP release. User stories were agreed and estimated facilitating the link between metaphor and development. Project scope was not a difficulty at this stage—a problem that did arise for the XP team. In subsequent sessions the team used the recorded pictures to recall objectives and stay focused.

The SSM-XP team implemented more user stories which seemed to be of higher quality. Also, despite extra initial effort, after five sessions they reached the same stage as the XP team. Feedback indicated that the Rich Picture was “very helpful” and “effective” in understanding the problem. The conceptual model helped “prioritize the requirements” from the Rich Picture and helped “clarify stories” for implementation.

3 Conclusion

This poster has proposed that there is still benefit in developing a metaphor for XP as originally indicated. The new version of XP allows inclusion of other practices so there is no reason why metaphor should not be implemented. The key is to provide guidance on how one can be constructed and linked to development. This has been achieved through the combination of parts of SSM with XP. Student exercises, although limited, have provided positive feedback indicating that this combination is beneficial without detracting from the overall development time. Our future work includes the examination and development of tools to build a metaphor while enabling communication within a distributed team.

References

1. Beck K., Andres C., *Extreme Programming Explained*, 2nd Edition, Addison Wesley, 2005.
2. Checkland, P., *Systems Thinking, Systems Practice* (with 30-year retrospective), John Wiley & Sons, 1999