An Agile Project System Dynamics Simulation Model

A. S. White, School of Science and Technology, Middlesex University, London, UK

ABSTRACT

This paper compares established Systems Dynamics (SD) models of software projects with models of agile development. A new minimal SIMULINK™ agile project model was created and compared to a Waterfall model of a NASA project. Results are presented to enable project managers to predict the performance of future agile project processes. The model includes the time to generate the requirements and a function to provide requirements volatility. These models show that for the same productivity and fraction of errors that are satisfactory, the overall development staff costs are similar for agile and waterfall projects and the undiscovered rework is less for the agile project. This model generally supports empirical observations that shorter time-box iterations yield a shorter overall project completion.

Keywords: Agile Project Development, Evo Project Management, SIMULINK, System Dynamics (SD), Waterfall Model

INTRODUCTION

Good project management (Abdel-Hamid & Madnick, 1989) is dependent on technical production processes, time scheduling in a dynamic environment, adequacy of project managers, team members effectiveness and team interaction. Projects may be considered as a system in which demands are made (the requirements) and an internal project organisation, which is controlled to produce the software goals, while being disturbed by the external environment and customers (White, 2013). A history of the development of project management before the agile revolution is given by Garel (2013) exemplifying the standard model.

Lin and Levary (1989) describe computer aided software design using System Dynamics (SD), expert systems and a Knowledge based management system used in the design of a space station. Häberlein (2004) has outlined common structures involved in SD models while Rodrigues and Bowers (1996) have established the criteria for successful use of System Dynamics in project management and Madachy (2008) produced a core text detailing the use of SD methods in software projects.

Operation of the software development process was described by the successful System Dynamics (SD) models of Tarek Abdel-Hamid and Madnick (TAH) (1991), which set up equations relating levels (states) such as the number

DOI: 10.4018/ijitsa.2014010104
of reworked errors and relates them to rates such as the error detection rate or the rework rate. The TAH model was validated at the time against NASA data for a space project and the agreement was very good. SD model structure is highly non-linear with data, usually implemented as table functions, for example about how the errors in the coding are propagated and how behaviour is based on perceived values. These non-linear assumptions (Lyneis & Ford, 2007) do not allow for System Dynamics models to be used to develop scalable rules for managers to make legitimate judgements about projects of different size. In the last thirty years the software development processes used by companies has changed significantly. For example, West and Grant (2010) report up to 35% of software projects now use agile project methods. Any models of software development need therefore, to include the evolutionary project management methods of Gilb (2005) and iterative methods such as SCRUM, Rational Unified Process (RUP), eXtreme Programming (XP) and Rapid Applications Development (RAD).

Research in project management, falls into four distinct phases, according to Hällgren (2012). Of these, gap spotting appears to be dominant. He maintains that challenges to current hypotheses have been neglected and suggests that this should be the function of academic research. Since every project is different it takes many projects to be evaluated before sufficient expertise is generated to contradict a current hypothesis. However models can readily show discrepancies if used properly. In engineering, simulation has enabled considerable progress to be made in recent years in the reduction of development times and in ensuring quality products. In the software industry considerable use is made of power law tools to predict software cost and effort. Since these are not causal and SD techniques are it is likely therefore that simulation of the project development process can be an important aid for managers to help with decision making. The use of SD techniques has allowed the importance of parameters on quite complex systems to be appreciated, when their influence was not clear at first sight, hence its’ wide use in industry. They can also be used to confirm or refute preconceived ideas about how software development proceeds using the scientific method. As will be seen in later sections there are few simulation models of agile processes and many are still quite complex and require a skilled SD practitioner to use them. What is attempted here is to develop a model of the agile project development process, particularly software projects. A model is desired that can be used with little fore-knowledge, particularly with little empirical data, not likely to be known before a project is started. This exercise is an initial effort to determine the minimum features that are needed to describe the observed behaviour of projects using a number of agile practices. It also allows a comparison to be made with existing Waterfall approaches to project development. A predictive tool is the future outcome of this research.

Agile Project Development

What follows is a brief review of agile methods with the purpose of gathering a set of common features that are essential for an agile project model.

The Waterfall software development method was used for many large projects in the 1970s and 80s. Salo and Abrahamsson (2008) claim this method has been largely replaced by agile methods such as Extreme Programming (XP) and Scrum (based on a rugby approach). Larman and Basil (2003) give a comprehensive history of iterative and incremental development. Agile methods are described and compared in detail by Abrahamsson et al. (2002) and Williams (2007). It is quite clear therefore that agile methods have been in use for many years and have been used for large important projects. Phillips (2006) gives comparison data for productivity of RUP teams. Some of the limitations in practice of the methods are described by Salo and Abrahamsson (2007). Qumer and Henderson-Sellers (2008) examine the degree of agility of six agile methods, giving guidance about which method is more appropriate in each circumstance. Dybå and
23 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:

www.igi-global.com/article/an-agile-project-system-dynamics-simulation-model/109090?camid=4v1


www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

On the Study of Complexity in Information Systems
www.igi-global.com/article/study-complexity-information-systems/2532?camid=4v1a

Human Capital Architecture and its Utilization in Accounting
www.igi-global.com/chapter/human-capital-architecture-its-utilization/29885?camid=4v1a

Food Security Policy Analysis Using System Dynamics: The Case of Uganda
www.igi-global.com/article/food-security-policy-analysis-using-system-dynamics/193593?camid=4v1a
Test Path Prioritization from UML Activity Diagram Using a Hybridized Approach
www.igi-global.com/article/test-path-prioritization-from-uml-activity-diagram-using-a-hybridized-approach/190604?camid=4v1a