IMPROVEMENT OF AGILE SOFTWARE PRODUCTION MANAGEMENT
USING SYSTEM DYNAMICS MODEL

Pijush Chandra Das¹*, U R Dhar²

¹Department of Business Administration,
Gauhati University, Guwahati-781014
pijush@gmail.com
²Royal School of Business,
RGI, Guwahati-781035
urxdhar@gmail.com

Abstract
Production and operations management for any industry is a highly challenging task, to minimize cost, control overruns, adherence to schedule and improve productivity by optimizing on given resources. Optimizing the production system of agile software products with frequent change of requirements remains a challenge. Through system dynamics model we can predict the dynamic behaviour of the agile software production process including the implications of managerial policies and procedures pertaining to the production of software.

Agile development is one solution to the problem of overly complex methods that has recently been adopted in the field of software production, and has gained considerable popularity with software producing organizations. This research investigates how applying system dynamics can help us analyzing the performance of Agile software production activity and in the long run improves in software production management. This paper using a system dynamics model portrays some of the dynamic forces directly impacting the Quality Assurance (QA) activity. The Feedback loop shows how schedule pressures, which arise when a software project falls behind schedule, can lead to a higher error generation rate. As more errors are committed, a larger chunk of the available manpower is diverted from development work and devoted instead to error correction and rework duties. As this happens, the project's progress rate drops further, leading to even greater schedule pressures thus adding to greater cost of poor quality.

Keywords: Agile software development, software projects, system dynamics, change request.

1 Introduction
The software projects have always been facing immense effort overrun, schedule pressure and excessive increase in rework due to more defect generation injected in systems. This is due to the changing scenario of software development where requirements keep on changing between iterations and within iterations. After functionalities have been developed for iteration the customer on reviewing it may either go for no change or go for changes. Contrary to plan-driven waterfall method for researchers and practitioners agile methodology has evolved as an alternative to overcome the cost, effort overrun, schedule slippage and quality problems (Oorschotet al., 2009). Agile Methods like XP-extreme programming (Beck, 2000), SCRUM (Coplien and Ostergaard, 2009) and Dynamic System Development Methodology have been introduced (Stapleton 1995). These methods have enabled teams in quickly responding to the frequent requirement changes(Paetschet al., 2000). Sinceto accommodate a change at a later stage in the iterations cost more, the cost and project risk reduces to an extent because of the ability of agile project teams to respond fast to changes (Beck,2000). In this paper we probe how a System dynamic subsystem can be used to understand the impact of requirement volatility which causes frequent changes in the project and schedule pressure.

1.1 Literature review
Rapid advances in technology, quickly evolving system requirements demands a flexible methodology like agile to develop software systems. The agile methodology is a widely practiced phenomenon now in IT industry but the effectiveness and appropriateness is not extensively proven by empirical research. A system dynamic model was developed considering the interdependencies of the various methods of agile development. The two important practices like refactoring and pair programming of agile practices are studied by an integrated system dynamic model. This new integrated tool of agile practices have helped in investigating some of the critical aspects of agile
development like customer involvement, refactoring, pair programming, agile planning and control and change management (Cao et al, 2010). It may be difficult to adopt agile methods for mission critical and large projects because lack of appropriate architecture planning, lower test coverage and intense focus on quick result (Boehm, 2002). The other constraints for the applicability of agile practices are experience of project team members, type and size of the project and the knowledge level and commitment of customers (Erickson et al., 2005), (Fitzgerald et al., 2006). The impact of agile methods on the resources involved in the project, the process implemented in the project and the project itself is studied by D. Philips (1998). The most widely used agile methodology is SCRUM and the fastest growing agile methodology is the Lean-Kanban approach. The dynamic behavior of the Kanbanand scrum approach versus the traditional waterfall model is analyzed by Cocco et al. (2011). Agile framework is iterative, incremental, adaptive, self-organizing and emergent.

2 Objective of the study

The objectives of the study are:

1. Impact of requirement Changes in Agile software development Projects.
2. Impact of schedule and other associated variables and their interdependencies in an agile approach.
3. Does Feedback loop play a role agile software project management?
4. Does Causal loop diagram can be used in tasks linking of agile projects?

3 Methodology

3.1 Model structure

Schedule pressure is a main component of our system Dynamic Model. Schedule crunch is very well known phenomenon in application software development as well as in software product development (Perlow, 1999). Due to urgency of releases the team members feels the inadequacy of time in the project. The project effort estimate cannot be done perfectly because in the beginning the number of task associated with a project is not very clear. Software development is not deterministic in nature (Brooks and Fred, 1979), (Boehm and Barry, 1981).

When estimation of the project is done project management has to focus on the less visible components including the highly visible main components. During the project execution more and more task are discovered, which calls forth additional effort and time and thus schedule pressure increases unless scheduled revised or more resource allocation is done.

3.2 Simulation Model

The model of Abdel Hamidet. al., (1991) which has been extensively tested is considered in our experimentation. The enhancement modules of the development project MTR which are of 120 days duration has been taken for our study and divided into multiple parts of Agility level iterations.

![Figure 1 Source Abdel Hamid et al. 1991](image-url)
The agile project iterations have multiple due dates compared to sequential(waterfall) model which causes disruptions in the learning curve of the team and also affects productivity(Seshadri and Shapira, 2000).

### 3.3 Constructive cost model (COCOMO) Estimation for one of the enhancements of MTR Project

The enhancement module of the MTR project consists of around 18000 LOC each and the schedule duration of them is 120 days. Using the COCOMO equations to estimate the perceived project effort in Person-days from the number of KLOC:

Planned person days for project

\[
= 2.73 \times 19 \\
\times \text{perceived project size in LOC} \\
/1000)^{1.05}
\]

\[
Project Size = \text{Actual Project size in LOC} \\
\times \text{uncertainty fraction} \\
= 18000 \times 0.65 \\
= 11700 \text{ LOC}
\]

Planned person days for Project

\[
= 2.73 \times 19 \left(\frac{11700}{1000}\right)^{1.05} \\
= 687 \text{ person} \times \text{days}
\]

The number of resources is taken as 8. As the number of task is now finally known 18000 LOC then 1078 person days are required. So with a team of 8 it would take 134 days. Therefore with 8 team members the schedule pressure will increase and overtime work will be required.

### 4 Context of Study

Two scenarios that is one is no change requirement from the customers between iterations and the other is requirement change given by customers. The length of the iteration is taken as a parameter to investigate its impact on schedule through the various Enhancementmodules of the development of MTR project. Each of the Enhancements had to be delivered to the client within 120 days. The Level of Agility(iterations) means the modules are divided in to how many parts of equal sizes is done. The data for the no change requirement is simulated from the model and the data for the requirement change is from the enhancement of the Live MTR project (primary data).

### 4.1 Data Analysis and Result

#### Table 1 Simulated data and data of MTR Project enhanced module of no change and requirement change scenario.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Iteration Duration (Days)</th>
<th>Defect leakage (no change)</th>
<th>Defect Leakage (requirement changes)</th>
<th>Effort (Person Days) for NO CHANGE</th>
<th>Effort (Person Days) FOR CHANGES</th>
<th>Actual time (in days)</th>
<th>Actual time (in days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration 1</td>
<td>12</td>
<td>217</td>
<td></td>
<td>1026</td>
<td>1026</td>
<td>129</td>
<td>220</td>
</tr>
<tr>
<td>Iteration 2</td>
<td>60</td>
<td>201</td>
<td></td>
<td>998</td>
<td>998</td>
<td>126</td>
<td>215</td>
</tr>
<tr>
<td>Iteration 3</td>
<td>40</td>
<td>192</td>
<td></td>
<td>987</td>
<td>987</td>
<td>125</td>
<td>208</td>
</tr>
<tr>
<td>Iteration 4</td>
<td>30</td>
<td>180</td>
<td></td>
<td>990</td>
<td>990</td>
<td>122</td>
<td>194</td>
</tr>
<tr>
<td>Iteration 5</td>
<td>24</td>
<td>164</td>
<td></td>
<td>988</td>
<td>988</td>
<td>122</td>
<td>178</td>
</tr>
<tr>
<td>Iteration 6</td>
<td>20</td>
<td>160</td>
<td></td>
<td>977</td>
<td>977</td>
<td>121</td>
<td>180</td>
</tr>
<tr>
<td>Iteration 7</td>
<td>17</td>
<td>172</td>
<td></td>
<td>995</td>
<td>995</td>
<td>123</td>
<td>199</td>
</tr>
<tr>
<td>Iteration 8</td>
<td>15</td>
<td>207</td>
<td></td>
<td>1000</td>
<td>1000</td>
<td>123</td>
<td>204</td>
</tr>
<tr>
<td>Iteration 9</td>
<td>13</td>
<td>230</td>
<td></td>
<td>1015</td>
<td>1015</td>
<td>124</td>
<td>254</td>
</tr>
<tr>
<td>Iteration 10</td>
<td>12</td>
<td>249</td>
<td></td>
<td>1032</td>
<td>1032</td>
<td>125</td>
<td>320</td>
</tr>
</tbody>
</table>

The defect count is the least in-case of iteration 6. Hence the Schedule pressure will be less with reduced rework. The exhaustion in the project will reduce productivity gradually increasing with decreasing turn over(Moore,2000),(Oliva and Sterman, 2001).

Impact of Iteration duration on defect count

The defect count is the least in-case of iteration 6. Hence the Schedule pressure will be less with reduced rework. The exhaustion in the project will reduce productivity gradually increasing with decreasing turn over(Moore,2000),(Oliva and Sterman, 2001).
IMPROVEMENT OF AGILE SOFTWARE PRODUCTION MANAGEMENT USING SYSTEM DYNAMICS MODEL

5 Conclusion

This research very well establishes that the number of iterations in agile project play a role in impacting the effort and defect count which in turn impacts the cost of quality. Applying system dynamic model in case of agile software projects and obtaining improved result shows that there is good scope for the usage of SD in Agile methodology and the model needs to be implemented in case of multiple projects with agile approach across multiple domains. If the consistency of result is visible across domains and technology which itself will throw open a vast area of research. Beyond this commercialized software Project management tool which uses system dynamics and caters to agile methodology can be other value addition to software production management.

References


J. Coplien and J. Ostergaard, (2009), SCRUM: It’s all about common sense, Scrum Training Institute, Scrum Alliance, Inc.


Boehm, B. (2002), Get Ready for Agile Methods, with Care, Computer, pp. 64-69.


Oliva, Rogelio, and John D. Sterman(2001), Cutting Corners and Working Overtime: Quality Erosion in the...
L. Cao, B. Ramesh, and T. Abdel-Hamid (2010), Modeling dynamics in agile software development, ACM Trans. Manage. Inf. Syst. 1, 1, Article.