A Review of the Impact of Requirements on Software Project Development Using a Control Theoretic Model

Anthony White
School of Engineering and Information Sciences, Middlesex University, the Burroughs, Hendon, London, UK.
Email: a.white@mdx.ac.uk

ABSTRACT

Software projects have a low success rate in terms of reliability, meeting due dates and working within assigned budgets with only 16% of projects being considered fully successful while Capers Jones has estimated that such projects only have a success rate of 65%. Many of these failures can be attributed to changes in requirements as the project progresses. This paper reviews several System Dynamics models from the literature and analyses the model of Andersson and Karlsson, showing that this model is uncontrollable and unobservable. This leads to a number of issues that need to be addressed in requirements acquisition.

Keywords: Requirements Models, System Dynamics, Control Systems, Observability, Controllability

1. Introduction

Software projects have a low success rate in terms of reliability, meeting due dates and working within assigned budgets with only 16% of projects being considered fully successful while Capers Jones has estimated that such projects only have a success rate of 65%. The American “Standish Group” has been involved for 10 years with research into ICT. In their research, they aim to determine and change success and failure factors regarding such projects. Their study, which has been appropriately baptised “Chaos” [4,5], appears every two years. This study also shows that in 2003 only 34% were successful, 51% did not go according to plan but ultimately did lead to some result and 15% of the projects fail completely.

Despite these failures significant progress has been made in the use of System Dynamics methods to describe the development of software projects. The models of operation of the software development process were described by the successful System Dynamics (SD) models based on the work of Abdel-Hamid & Madnick [6], which set up equations relating levels such as the number of perceived errors, or the number of reworked errors and relates them to rates such as the error detection rate or the rework rate, significant features of these models included the decision processes. These models were validated against NASA project data for a medium size project and the agreement is strikingly good.

Many of these failures can be attributed to changes in requirements as the project progresses. Capers–Jones [7] states that as the project gets larger the probability of requirements creep becomes more likely, typically 1-2% per month and as high as 10% in a single month. Lorin May [8] talks about poorly established guidelines that determine when requirements should be added, removed and implemented. Deifel and Salzmann [9] describe a view of “requirements dynamics” relating to the process of changing requirements. They go on to develop a strategy to deal with the regime in which some requirements are invariant and some migrate.

Coulin et al. [10] state that “the elicitation of requirements for software systems is one of the most critical and complex activities within the development cycle” and that “this is performed after project initiation and preliminary planning but before system conception and design.” This would not be strictly true if evolutionary or iterative methods were used. The later the requirements in the cycle of development change, the more costly is that revision (Boehm & Pappacio [11]). It is certainly the case as Hoorn et al. [12] report that owing to many shifts in focus and priorities, stakeholders become inconsistent about what they actually want to accomplish with the system. If we are to improve the requirements process then proper models of a process are needed. Kotanya &
Sommerville [13] outlines the requirements engineering process as shown in Figure 1. Although there is feedback between requirements validation and specification and in the elicitation and specification as will be shown this is not represented in the current models. It is not clear in any of the texts on the subject whether the involvement of the user is mandated at these stages.

The whole purpose of this paper is to present simple control system models of the project development process including requirements, as in inventory analysis, and demonstrate rules for stability.

2. System Dynamics

Wolstenholme [14] describes System Dynamics as:

“A rigorous method for qualitative description, exploration and analysis of complex systems in terms of their processes, information, organizational structure and strategies; which facilitates simulation modelling and quantitative analysis for the design of system structure and control”.

This definition is expanded in Table 1 taken from Wolstenholme.

The SD model structure is highly non-linear with a number of theoretical assumptions, for example about how the errors in the coding are propagated.

These structural assumptions do not allow for System Dynamics models to enable any general rules to be developed by academics for managers to make sound judgments based on good analysis. The distinction with models of inventory processes, which are related, is the rationale for this research program. Early SD inventory models developed by Forrester [15] were also non-linear and contained a number of factors, such as employment rate, that made the problem too complex for simple rules to be developed.

The simplest expression of representation of requirements in SD models is that use by Madachy [16], shown in Figure 2. In this case requirements are added to by a rate of generation, usually constant. The time taken to acquire the whole requirements is dictated by the acquisition rate. Häberlein [17] proposed a different structure for the development of the whole project. In his model (Figure 3) the rate of generation of requirements is split into several phases depending on the comprehension of the supplier and how this is influenced. This model could show considerable promise but no equations are presented. The model of Williams [18] (Figure 4) could not be evaluated further at this time due to incomplete equations. The structure indicated shows dependence on quantities such as customer satisfaction that are not readily measured during the process. The model of Andersson and Karlsson [19] (Figure 5) is the most complete and useful model out in the literature. Not only are all the equations given, with data, but the results are of a project in industry. This model shows that the process of gaining requirements is split into a phase where the level of requirements tasks to be completed is gained via an input pulse function. The required tasks to be completed are fed from the previous state by a constant requirements completion rate. Rework is discovered in these requirements and this is fed back at a constant rate to the first level. Inadequate requirements are discarded at a rate that

![Figure 1. Requirements engineering (from Kotanya & Sommerville).](image-url)
Table 1. System Dynamics a subject summary from Wolstenholme [14].

<table>
<thead>
<tr>
<th>Qualitative system dynamics</th>
<th>Quantitative system dynamics</th>
</tr>
</thead>
<tbody>
<tr>
<td>(diagram construction and analysis phase)</td>
<td>(Simulation phase)</td>
</tr>
<tr>
<td>Stage 1</td>
<td>Stage 2</td>
</tr>
<tr>
<td>1. of existing/proposed systems</td>
<td>1. To examine the behavior of all system</td>
</tr>
<tr>
<td>2. To create and examine feedback loop structure</td>
<td>variables over time.</td>
</tr>
<tr>
<td>3. To provide a qualitative assessment of the relationship between system process</td>
<td>2. To examine the validity and sensitivity</td>
</tr>
<tr>
<td>structure, information structure, delays organizational structure and strategy</td>
<td>of the model to changes in</td>
</tr>
<tr>
<td></td>
<td>• Information structure</td>
</tr>
<tr>
<td></td>
<td>• Strategies</td>
</tr>
<tr>
<td></td>
<td>• Delays and uncertainties</td>
</tr>
<tr>
<td>1. To examine alternative system structures and control strategies based on</td>
<td>1. To examine alternative system structures</td>
</tr>
<tr>
<td>• Intuitive ideas</td>
<td>and control strategies based on</td>
</tr>
<tr>
<td>• Control theory analogies</td>
<td>• Control theory analogies</td>
</tr>
<tr>
<td>• Control theory algorithms: in terms of non-optimizing robust policy design</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. Raymond Madachy’s model.

Figure 3. Requirements as a total process in comparison to Abdel-Hamids’ task based mod.

Figure 4. Requirements model of Williams [17].

is also a constant’. The final finished requirements are fed by a finished requirements rate. A number of non-linear “constants” are embedded into the system. No proper validation is made of this model or any of the models given here (this is normally very difficult).
Do any or all of these models match the published material on requirements engineering? In the broadest sense, yes, they do match what is contained in books such as Sommerville. To make further progress let us assume that the Anderson and Karlsson model is correct. This non-linear SD model has been linearised and analysed using control theory to see any general lessons can be learned.

3. Control Analysis

Part of the simplification of the Project Model is being tackled in the USA by the newer control system models of software testing (Cangussu et al. [20]) and the approach to control of software development by White [21].

In this case the model of Andersson and Karlsson was linearized and the following state equations obtained:

$$\frac{drttb}{dt} = crr - rcr + rw$$  \hspace{1cm} (1)

$$\frac{drtc}{dt} = rcr - frr - rw - irr$$  \hspace{1cm} (2)

$$\frac{dir}{dt} = irr$$  \hspace{1cm} (3)

$$\frac{dfr}{dt} = frr$$  \hspace{1cm} (4)

The linearized auxiliary SD equations are:

$$crr = fio(t)$$  \hspace{1cm} (5)

(where this is a pulse of height $fi$, the initial estimate of the number of requirements).

These equations can be represented by a state-space equation

$$x = Ax + Bu + B'v$$  \hspace{1cm} (10)

where $A$, $B$ and $B'$ are given by:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 - rwp & 1 - rwp \\ 0 & 0 & 1 - rwp \end{bmatrix}$$  \hspace{1cm} (11)

$$B = \begin{bmatrix} fi \\ 0 \\ 0 \end{bmatrix}$$  \hspace{1cm} (12)

$$B' = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$  \hspace{1cm} (13)
neutrally stable since there are no feedback mechanisms
viewpoint. This analysis shows that such models are
and Karlsson has been analysed from a control system
The most comprehensive model cited, due to Andersson
complexity. If this is generally true it has severe implica-
only reluctantly, taking no account of project size or
productivity. The number of staff in the cases cited
the customer to specifications, depending strictly on staff
constant rate of conversion of requirement wishes from
All the SD models illustrated here would appear to use a
4. Conclusions

All the SD models illustrated here would appear to use a
constant rate of conversion of requirement wishes from
the customer to specifications, depending strictly on staff
productivity. The number of staff in the cases cited
appears to be fixed at the start of the process and altered
only reluctantly, taking no account of project size or
complexity. If this is generally true it has severe implica-
tions for the later analysis and development of the project.
The most comprehensive model cited, due to Andersson
and Karlsson has been analysed from a control system
viewpoint. This analysis shows that such models are
neutrally stable since there are no feedback mechanisms
to establish when all the requirements are obtained, and
they are neither controllable nor observable. The problem
is that only the group of states \(fr, ir\) and \(rtc\) together are
specified, one of them cannot be separately described or
made to achieve a particular trajectory. If the staff pro-
ductivity is fixed and the number of staff is decided be-
forehand then the final outcome is proscribed. They can
with some manipulation be made stabilizable.

REFERENCES

[2] K. T. Yeo, “Critical Failure Factors in Information Sys-
tem Projects,” International Journals of Project Man-
Complex IT projects,” Report of working group of RAE
CHAOS Report. 1998
CHAOS Report, 25 March 2003
Dynamics: An Integrated Approach,” Englewood Cliffs,
[7] C. Jones, “Large Software System Failures and Suc-
tions for Dynamics in Evolutionary Software Systems,”
Proceedings of the International Workshop on the Prin-
Method Engineering Approach to Requirements Elicita-
tion Workshops in the Software Development Process,”
Software Process Improvement and Practice, Vol. 11, No.
trolling Software Costs,” IEEE Transactions on software
and Priorities. Different Relevance of Requirements to
Changing Goals Yields Conflicting Prioritizations and Its
Viewpoint,” Software Process Improvement and Practice,
[14] E. Wolstenholme, “A Current Overview of System Dyna-
mics,” Transactions on Institute MC, Vol. 114, No. 4,
1961.
A Review of the Impact of Requirements on Software Project Development Using a Control Theoretic Model


Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crr</td>
<td>Customer requirements rate</td>
</tr>
<tr>
<td>fi</td>
<td>Initial value of requirements assumed</td>
</tr>
<tr>
<td>fr</td>
<td>finished requirements</td>
</tr>
<tr>
<td>frr</td>
<td>finished requirements rate</td>
</tr>
<tr>
<td>ir</td>
<td>Inadequate requirements</td>
</tr>
<tr>
<td>irr</td>
<td>inadequate requirements rate</td>
</tr>
<tr>
<td>rtc</td>
<td>Requirement Tasks Completed</td>
</tr>
<tr>
<td>rcr</td>
<td>requirements completed rate</td>
</tr>
<tr>
<td>rp</td>
<td>requirement part</td>
</tr>
<tr>
<td>rprod</td>
<td>requirement productivity</td>
</tr>
<tr>
<td>rtt</td>
<td>fraction of tasks inadequate</td>
</tr>
<tr>
<td>rtbc</td>
<td>Requirement tasks to be completed</td>
</tr>
<tr>
<td>Rw</td>
<td>rework rate</td>
</tr>
<tr>
<td>Rwp</td>
<td>rework fraction of RTC</td>
</tr>
</tbody>
</table>