J. Software Engineering & Applications, **2010, 3, 852-857**
doi:10.4236/jsea.2010.39099 Published Online September 2010 (http://www.SciRP.org/journal/jsea)

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 [1-3] 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 preformed 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 use 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 quantitive 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

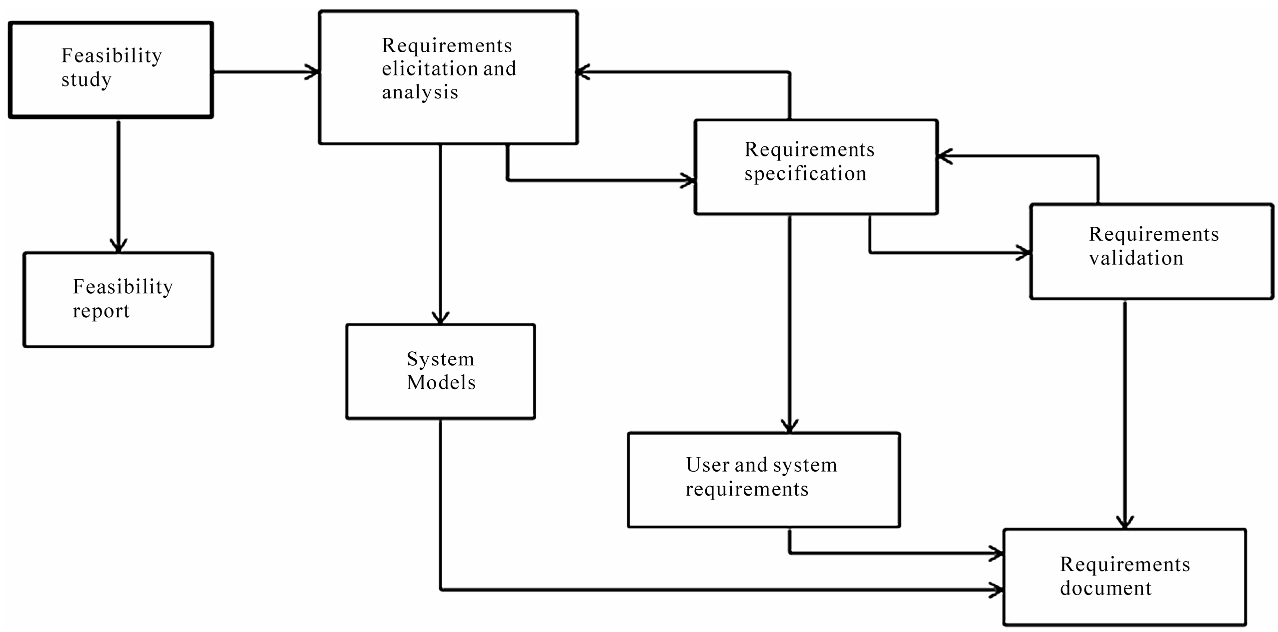

Figure 1. Requirements engineering (from Kotanya & Sommerville).

Table 1. System Dynamics a subject summary from Wolstenholme [14].

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

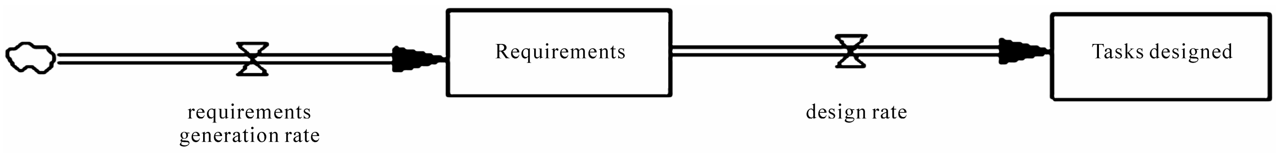

Figure 2. Raymond Madachy's model.

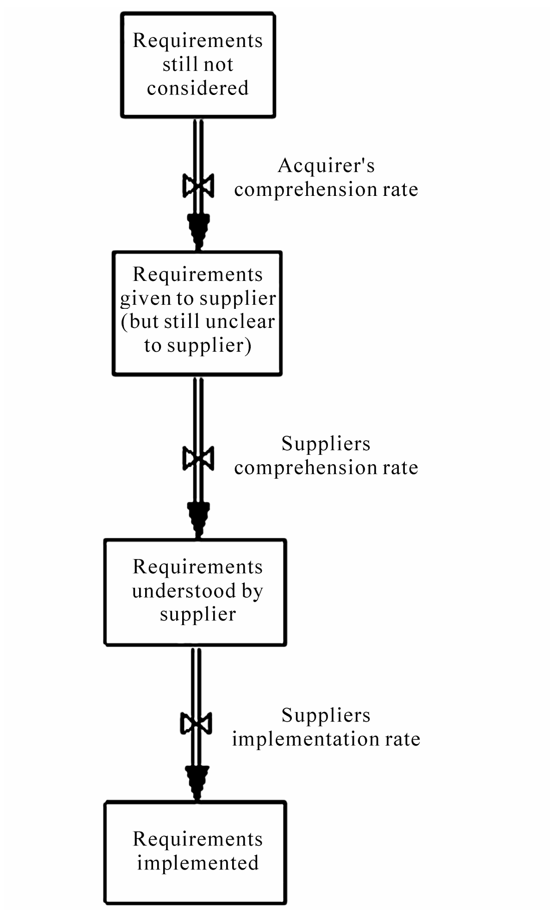

Figure 3. Requirements as a total process in comparison to Abdel-Hamids' task based mod.

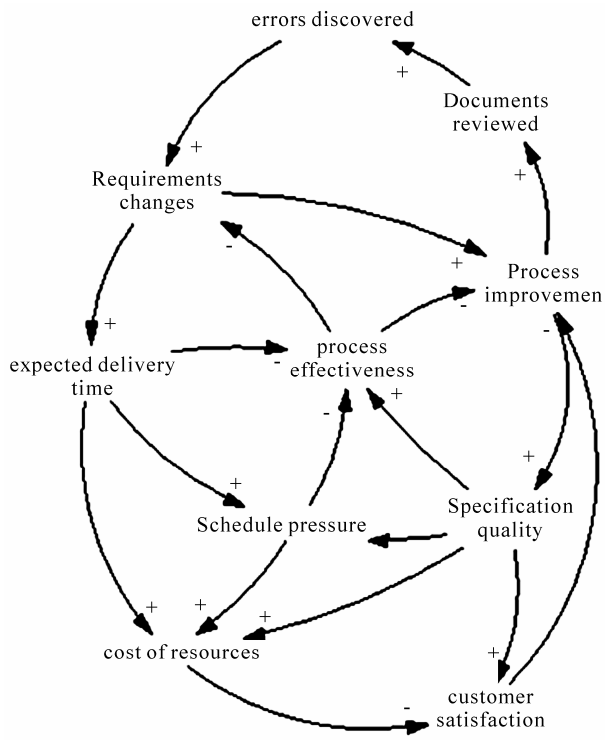

Figure 4. Requirements model of Williams [17].

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).

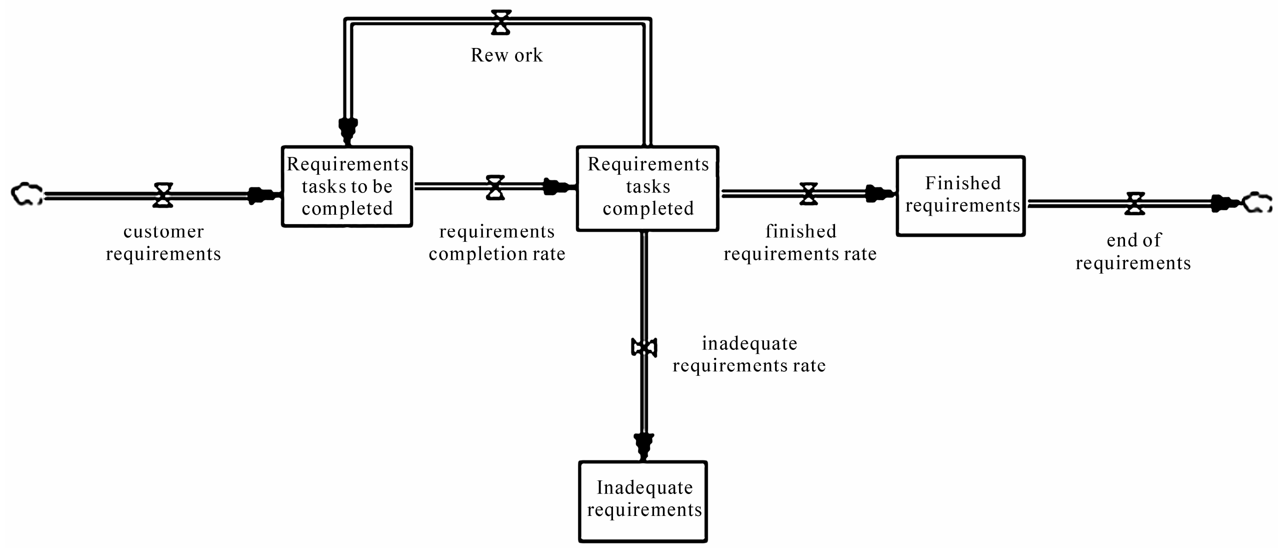

Figure 5. The model of Andersson and Karlsson [18].

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:

$$\frac{drttbc}{dt} = crr - rcr + rw \tag{1}$$

$$\frac{drtc}{dt} = rcr - frr - rw - irr \tag{2}$$

$$\frac{dir}{dt} = irr \tag{3}$$

$$\frac{dfr}{dt} = frr \tag{4}$$

The linearized auxiliary SD equations are:

$$crr = fi\delta(t) \tag{5}$$

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

$$rcr = rprod \tag{6}$$

$$irr = \left(\frac{rtt}{rp}\right) rtc \tag{7}$$

$$rw = rwp(rtc) \tag{8}$$

$$frr = \left(1 - rwp - \frac{rtt}{rp}\right) rtc \tag{9}$$

These equations can be represented by a state-space equation

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}\dot{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{B}'\boldsymbol{v} \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{aligned} \tag{10}$$

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

$$\boldsymbol{A} = \begin{bmatrix} 0 & rwp & 0 & 0 \\ 0 & -\left(\left(1 - rwp - \frac{rtt}{rp}\right) + rwp + \frac{rtt}{rp}\right) & 0 & 0 \\ 0 & \frac{rtt}{rp} & 0 & 0 \\ 0 & 1 - rwp - \frac{rtt}{rp} & 0 & 0 \end{bmatrix} \tag{11}$$

$$\boldsymbol{B} = \begin{bmatrix} fi \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

$$\boldsymbol{B}' = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{13}$$

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$\boldsymbol{D} = 0 \quad (15)$$

$$\boldsymbol{x} = \begin{bmatrix} rttbc \\ rtc \\ ir \\ fr \end{bmatrix} \quad (16)$$

where $\boldsymbol{u}$ = pulse function and $\boldsymbol{v} = rprod$. In this configuration $\boldsymbol{v}$ acts as a disturbance.

State-space theory can be used to see if this system is either controllable or observable.

We can define two matrices that will allow a measure of these properties if they are both full rank. The control stability is defined by the four eigenvalues two zero and two damped complex conjugates. The system is neutrally stable at best.

$$\boldsymbol{Cm} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{A}^3\boldsymbol{B} \end{bmatrix} \quad (17)$$

The rank of $\boldsymbol{Cm}$ is 1! The observability is given by:

$$\boldsymbol{Om} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \boldsymbol{CA}^3 \end{bmatrix} \quad (18)$$

The rank of this matrix is also 1. This means that the system described by the linearized state equations is uncontrollable and unobservable! The principle reason for this is that no corrective forces exist to alter the rate of production of requirements and that the rework and inadequate requirements cannot be altered independently of each other. Although a set of parameters will allow the requirements to be produced, once set in train no process exists to vary that process. No variation in workforce for example is set up in this model. No simple solutions allow this model to be put into a controllable form, although it can be made observable.

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 implications 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 productivity is fixed and the number of staff is decided beforehand then the final outcome is proscribed. They can with some manipulation be made stabilizable.

REFERENCES

[1] J. Smith, "The 40 Root Causes of Troubled IT Projects," *Computing and Control Journals*, June 2002, pp.109-112.

[2] K. T. Yeo, "Critical Failure Factors in Information System Projects," *International Journals of Project Management*, Vol. 20, 2002, pp. 241-246.

[3] Royal Academy of Engineering, "*The Challenges of Complex IT projects*," Report of working group of RAE and BCS, 2004.

[4] The Standish Group International Inc., Standard Group CHAOS Report. 1998

[5] The Standish Group International Inc., Standard Group CHAOS Report, 25 March 2003

[6] T. Abdel-Hamid and S. E. Madnick, "Software Project Dynamics: An Integrated Approach," Englewood Cliffs, Prentice Hall, New York, 1991.

[7] C. Jones, "Large Software System Failures and Successes," *American Programmer*, April 1996, pp.3-9.

[8] L. J. May, "Major Causes of Software Project Failures," *Crosstalk*, July 1998.

[9] B. Deifel and C. Salzmann, "Requirements and Conditions for Dynamics in Evolutionary Software Systems," *Proceedings of the International Workshop on the Principles of Software Evolution, IWPSE*99, Fukuoka, 1999.

[10] C. Coulin, D. Zowghi and A. Sahraoui, "A Situational Method Engineering Approach to Requirements Elicitation Workshops in the Software Development Process," *Software Process Improvement and Practice*, Vol. 11, No. 5, 2006, pp.451-465.

[11] B. Boehm and P. N. Pappacio, "Understanding and Controlling Software Costs," *IEEE Transactions on software engineering*, Vol. 14, 1988, pp.1462-1477.

[12] J. F. Hoorn, M. E. Breuker and E. Kok, "Shifts in Foci and Priorities. Different Relevance of Requirements to Changing Goals Yields Conflicting Prioritizations and Its Viewpoint," *Software Process Improvement and Practice*, Vol. 11, No. 5, 2006, pp. 465-485.

[13] G. Kotonya and I. Sommerville, "Requirements Engineering Processes and Techniques," Wiley, 1988.

[14] E. Wolstenholme, "A Current Overview of System Dynamics," *Transactions on Institute MC*, Vol. 114, No. 4, 1989, pp. 171-179.

[15] J. Forrester, "Industrial Dynamics," MIT press, Boston, 1961.

[16] R. Madachy and B. Khoshnevis, "Dynamic Simulation Modeling of an Inspection-Based Software Lifecycle Process," *Simulation*, Vol. 69, No.1, 1997, pp.35-47.

[17] T. Häberlein, "Common Structures in System Dynamics Models of Software Acquisition Projects," *Software Process Improvement and Practice*, Vol. 9, No. 2, 2004, pp. 67-80.

[18] D. Williams, "Challenges of System Dynamics to Deliver Requirements Engineering Projects: Faster, Better, Cheaper," 21st *System Dynamics Conference*, New York, 2003.

[19] C. Andersson and L. Karlsson, "A System Dynamics Simulation Study of a Software Development Process," Lund Institute of Technology Report, March, 2001.

[20] J. W. Cangussu, R. A. DeCarlo and A. P Mathur, A Formal Model for the Software Test Process, *IEEE Transactions on Software Engineering*, vol. 28, no.8, August 2002, pp.782-796.

[21] A. S. White, "Control Engineering Analysis of Software project Management," *BCS SQM Conference*, Stafford, Section 3, April 2007.

Symbols

C*rr*	***Customer requirements rate***
fi	***Initial value of requirements assumed***
fr	***finished requirements***
frr	***finished requirements rate***
ir	***Inadequate requirements***
irr	***inadequate requirements rate***
rtc	***Requirement Tasks Completed***
rcr	***requirements completed rate***
rp	***requirement part***
rprod	***requirement productivity***
rtt	***fraction of tasks inadequate***
rttbc	***Requirement tasks to be completed***
Rw	***rework rate***
Rwp	***rework fraction of RTC***

