Table of Contents

Go Back

"Integrative Mechanisms in New Product Development Projects: Effect of Project Complexity on Project Performance A System Dynamics Approach"

Reda M LEBCIR
Imperial College Management School
53 Princes Gate, Exhibition Road, London SW7 2PG, United Kingdom
Phone: ++.44.207.594.9143
Fax: ++.44.207.823.7685
E-mail: Reda.Lebcir@ic.ac.uk

Abstract: The current paper investigates whether the shift to more integration, that is use of concurrent engineering process and cross-functional teams organisation, in new product development (NPD) projects is suitable for all types of projects regardless of their level of "project complexity".

The current paper addresses this research question in two steps. First, a formal framework of "project complexity" is developed for NPD projects in which its contributing factors are determined. Second, a System Dynamics (SD) model is developed to investigate this issue. The model is used to quantify the extent of "project complexity" effects on the relationship between the level of integration in a project and one performance indicator: development cycle time. The model was run for a single phase situation and showed that product complexity and project uncertainty increases project development cycle time and that increasing co-ordination does not offset these negative effects on project schedule performance.

Key Words: New Product Development, Concurrent Engineering, Project Management, Project Complexity, Product Complexity, Product Newness, Uncertainty, System Dynamics.

Introduction

The business environment has changed dramatically in the last two decades creating new challenges for firms in most industries. It has become more global, turbulent, and dynamic making it difficult for firms to respond adequately to the shifting needs of markets and customers. In this context, the development of new products is becoming increasingly central to any strategy for achieving high levels of profitability, market share and, in the long run, gaining a significant leverage of competitive advantage (Ulrich and Eppinger 1999, Wind and Mahajan 1997, Wheelwright and Clark 1992, Clark and Fujimoto 1991, Gupta and Wilemon 1990).

However, developing such new products is a daunting task. Rapid changes in customers' needs have shrunken products life cycle. Similarly, rapid technological changes are making technical processes used to develop new products obsolete more quickly. As a consequence, a bleak image is emerging regarding the performance of new product development (NPD) projects. Research results show that failure is the rule rather than the exception. For example, a significant fraction of new software and computer systems have been marked by cost overruns, late deliveries, poor reliability and users' dissatisfaction (Abdelhamid and Madnick 1991). In other studies, Rodriguez and Bowers (1996) and Page (1993) reported that most of NPD projects registered cost-overruns of 40% to 200% while about 48% of resources are put on projects which are either cancelled before market introduction or fail to yield suitable financial returns

One of the important reasons to explain these failures is the inadequacy of project management procedures used to carry out these innovation projects (Langerak et al 1999, Milson et al 1992). Developers' technical abilities alone have not immune organisations from witnessing the considerable rate of failure mentioned earlier. Project management is the area, which should become the focus of attention if project performance is to be improved. As pointed out by Abdelhamid and Madnick (1991) "the major breakthroughs are now to be made in the management arena".

Project management improvement has become more crucial as companies are facing new challenges to carry out innovation projects. In response to the business environment pressures cited earlier, most firms have re-engineered their NPD project management paradigms from a sequential functional process to a concurrent cross-functional one. Whereas this shift has increased dramatically the dynamic complexity of projects (Ford and Sterman 1998, Rodriguez and Bowers 1996, Ford 1995), the mental models and decision heuristics used by project managers have improved a little to deal with this increased complexity (Diehl and Sterman 1995, Sterman 1994,1992).

1. Evolution of new product development project management paradigms

The traditional process, called the Phased Program Planning (PPP), was the first formal process used to manage NPD projects. Its structure stipulates that the project should pass by checkpoints sequentially to ensure that all the items required at a checkpoint are in good order before the project moves sequentially to the next one (Wheelwright and Clark 1992, Clark and Fujimoto 1991, Smith and Reinertsen 1991). Each phase of the project is performed entirely within a specialised functional structure and the outcomes of the phase are transferred in one "batch" to the subsequent one.

To help in the control of work evolution in such projects, many formal project management techniques, based on network analysis, have been developed. The most famous are Critical Path Method (CPM) and Project Review and Control Techniques (PERT) (Dreger 1992, Moder at al 1983). These techniques are methods for planning, scheduling and controlling projects. They aim to determine the activities which may constitute possible bottlenecks to the progress of projects, help managers determine possible slacks and schedule slippage available in some activities and how trade-offs between project schedule performance and resources can be made (Dreger 1992, Moder at al 1983). The use of CPM/PERT methods was suitable for the management of NPD projects under the old paradigm because they fit well with their underlying assumptions. The development work is sequential. No downstream task can start until all the dependent upstream ones are finished. Blocks of development tasks were represented by activities in the CPM/PERT project network and the precedence relationships were customised to represent the sequential dependencies between development phases.

Despite their usefulness, CPM/PERT suffer from some limitations. Their underlying assumption is that the work in the project will unfold as planned and no changes to its state will occur until the project is finished. The precedence relationships between tasks imply that when a task is finished, there is no possibility that a work can be done again on that task. In addition, CPM/PERT do not address adequately the dynamic interaction between the work structure, that is the map of tasks and their dependencies, and the human environment. Factors such as fatigue, burnout, motivation, and experience, as important as they are, are ignored in these methods.

The new project management process involves a more parallel execution of development phases, a technique known as concurrent engineering (CE), and a cross-functional team organisational structure (Swink et al 1996, Brown and Einsenhardt 1995 Zirger and Hartely 1994, Milson et al 1992, Gupta and Wilemon 1990, Takeuchi and Nonaka 1986). CE includes overlapping activities in product and process development (Swink 1998, Swink et al 1996, Hull et al 1996), simultaneous development of different product components and sub-systems (Liker et al 1999), and integration of decisions regarding product design and manufacturing capabilities (Liker et al 1999, Gerwin and Susman 1996).

The use of CE proved to generate many benefits for companies adopting it in their NPD projects. Merits of CE include greater speed of development execution, increased flexibility in product and process design, enhancement of shared responsibility and co-operation, and stimulation of developers involvement in the project (Swink 1998, Takeuchi and Nonaka 1986). However, the benefits associated with using CE are not priceless. Successful implementation of CE has been found to be problematic in many organisations. Adopting CE means changes in the nature and frequency of information flows between developers, timing of upstream and downstream tasks, and the altitude throughout the organisations towards dealing with uncertain and incomplete information (Krishnan et al 1997, Hauptman and Hirji 1996). As a result, CE has dramatically amplified the dynamic characteristics of projects.

Scheduling and organisation of development tasks in CE contexts were at the origin of the development of a methodology called the Design Structure Matrix (DSM) (Steward 1981). The DSM matrix is based upon the analysis of the logical dependencies between development tasks. This dependency is conceptualised through the information requirements for each task as the previous ones are executed and their output information becomes available. To organise the development work, partition algorithms are used to find the best possible tasks execution sequence. Once the matrix is reorganised, it will indicate the tasks which are sequential, parallel, and those which are coupled, that is the tasks which depends reciprocally on each other to proceed, and, therefore, need to be performed through iterations. However, even if DSM is a useful method to organise task iterations in NPD projects, it remains a static method. Dependencies between tasks and phases are assumed to remain stable over the project life cycle and, consequently, it is not useful to describe the behaviour of a project over time.

Krishnan (1996) and Krishnan et al (1997) developed a formal mathematical model to study the optimal overlapping strategy in sequential concurrent development activities. The model stipulates that when overlapping occurs, the information released from upstream to downstream is not in its final form but it is still being refined. As this refinement process takes place, downstream phase starts its development work with the initial imperfect information and accommodates the consequences of any changes in the information released from upstream through development work iterations. The model represents the refinement of information in the upstream phase and the amount of development work necessary in the downstream phase to accommodate changes in upstream information through the evolution and sensitivity concepts respectively. These concepts are used in a mathematical program to decide about the optimal number of downstream iterations to perform, the starting time of these iterations, and the time at which upstream information should be finalised. The objective function is to minimise the total development cycle time. The problem with this method is that it assumes a one-way directional flow of information and does not indicate which factors affect the evolution and sensitivity concepts in the project.

The adoption of CE was associated with a move from functional organisation to multi-functional teams (Wind and Mahajan 1997, Gerwin and Susman 1996, Carmel 1995, Trygg 1993, Henke et al 1993, Wheelwright and Clark 1992). Two reasons triggered this organisational switch. First, it is well established that NPD projects are inherently multi-functional tasks. The second reason is that in CE environment, development work is carried out simultaneously on different aspects of the product such as concept generation, product design, process design, prototype building, and product testing. As a result, it is necessary to have representatives from different functions to ensure shared control and responsibility of the development work as phases are overlapped (Swink 1998, Swink et al 1996, Henke et al 1993). In fact as stated by Cooper (1994), "the nature of activities (overlapped) virtually forces the use of cross-functional project team approach".

2. Effect of project complexity on NPD project performance

Thus far, we described the operational process and the organisational structure dominating NPD project management under the old and new paradigms. However, it is important to indicate that before the NPD project actually starts, developers and management have to make important decisions regarding the shape and destination of the product to be developed. These decisions include product definition, the number of functions to be performed by the product, the parts and sub-systems to be included, the breadth of the new technologies to be included in the product, the amount of redesign to be performed on the parts, and the market segment to be targeted by the new product. (Tatikonda and Rosenthal

2000a,b, Tatikonda 1999, Griffin 1997 a,b, Olson et al 1995, Zirger and Hartely 1994, Clark and Fujimoto 1991). These choices, which incorporate the new product "strategic choices", are important because they have several consequences over the NPD project life cycle and beyond. (Griffin 1997 a,b,Wheelwright and Clark 1992, Clark and Fujimoto 1991, Smith and Reinertsen 1991)

Projects involving high innovativeness, different interacting new technologies, many inter-related parts and sub-systems, a high fraction of newly designed parts, and directed to a poorly understood market are obviously more difficult to manage and carry high levels of risks (Tatikonda and Montoya-Weiss 2001, Griffin 1997 a,b, Swink et al 1996, Iansiti 1995a,b,c). Non-familiarity with customer needs and requirements, product and process technologies, coupled with a lack of information on how to proceed with development work, diminish developers capability and confidence in executing the project work. One project manager reported in Tatikonda and Rosenthal (2000b) describes well this situation when he said

> "Of course, we know what the big pieces are, but the problem is that we don't know what the small tasks are until we get there in the project, and oftentimes, these small tasks turn out to be big tasks".

The first remedy to more effective handling of strategic issues in NPD projects is to recognise that projects are not all similar and, consequently, they require different managerial approaches. This argument is made even clearer by Tatikonda (1999), who studied two different categories of NPD projects with respect to project complexity, the so-called *platform* and *derivative* projects, and reported that.

> [Platform and derivative projects represent ***different*** types of product development projects. Different types of projects can have fundamentally different characteristics and, in turn, may require different product development project planning and execution approaches. This is, a one shot, singular project management approach may not be appropriate for each product in the family series. Use of a similar approach may lead to reduced overall product development effectiveness]

Unfortunately, firms still carry out projects with significant differences in project complexity levels in the same way. Meyer at al (1997) found that "firms make little if any explicit differentiation between more fundamental platform effort and the development of derivative products". Similarly, Shenhar (1998) indicates that there is no distinction, in practice, among different types of projects in terms of project complexity. This situation prevails even within the academic community as it was well described by Pinto and Covin (1989):

> "The prevailing tendency among the majority of academics has been to characterise all projects as fundamentally similar…. The implicit view of many academics could be represented by the axiom: "a project is a project is a project".

In this context, the target of the current paper is to shed more light on this issue and investigate whether the new NPD project management paradigms are suitable for all projects regardless of their project complexity level.

This issue warrant more attention as the empirical results so far show no-conclusive, indeed contradictory, results. For example, in some studies the use of cross-functional teams was found to reduce cycle time for highly innovative projects (Liker et al 1999, Griffin 1997a,b, Olson et al 1995). Other studies reported that their use had no effect on development time regardless of the level of innovation in the project (Tatikonda and Rosenthal 2000b, Hull et al 1996, Larson and Gobelli 1989). Similraly, CE was found to have a positive impact on highly innovative projects (Swink 2000, Detoni et al 1999, Hanfield 1994). But, others reported completely opposite conclusions (Swink et al 1996, Einsenhardt and Tabrizi 1995). Similar conflicting conclusions have been drawn regarding the effects of this interaction on project quality and costs (Swink 1999,2000, Hull et al 1996, Larson and Gobelli 1989). In summary, it is far from clear how the interaction between strategic and operational characteristics in NPD projects affects their performance.

3. Definition of project complexity

Many project managers are using the term “complex projects” in describing the projects they manage, yet it is not clear what are the factors contributing to this complexity. It is until recently that a review within the project management literature has shed more light on this issue (Williams 1997,1999a).

The first factor contributing to project complexity is related to the underlying structure of the project and known as “structural complexity”. This factor was introduced by Baccarini (1996) who defined project complexity, in a broader sense, as “consisting of many varied interrelated parts”. This factor was breakdown into two dimensions. The first is *differentiation*, that is the number of varied components in the project (tasks, specialists, sub-systems, parts). The second is *interdependence* or *connectivity,* that is the degree of inter-linkages between these components.

However, “structural complexity” does not account for the level of difficulty to carry out project’s tasks. Williams (1997,1999a) argues that this “difficulty” is, in fact, the second factor contributing to project complexity and define it as “uncertainty”. This factor means that, contrary to the widespread belief, project goals and execution methods are not always known and well defined at the beginning of the project execution (Turner and Cohrane 1993). In many projects, a great deal of uncertainty about project’s goals and execution methods remains after the project work has been underway. This uncertainty causes the project work to become difficult and its outcome unpredictable, hence increasing the overall level of project complexity.

The “uncertainty” factor has been also broken down into two dimensions: *uncertainty in methods* and *uncertainty in goals.* Uncertainty in methods refers to the lack of knowledge on how to proceed to achieve project goals. It represents situations in which the tasks to be performed and the ways to perform them are not well known and defined at the beginning of the project execution. This increases project complexity because the managerial elements of the project in the form of project breakdown structures cannot be defined with certainty. Uncertainty in goals refers to situations in which the project targets are ill-defined at the beginning of the project This adds to project complexity because as the work proceeds, requirements will have to be changed and refined many times causing changes in the product components, layout, interfaces and architecture. In such situations, the basic project management activities such as planning, scheduling, monitoring, and control becomes ineffective as the structure of the product to be developed keeps changing over the project life cycle.

4. The New Product Development perspective of project complexity

NPD projects are inherently complex because they involve development of products which carry some degree of novelty. However, if there is an implicit acknowledgement among practitioners and academics that NPD projects are complex, there is a great deal of confusion about the drivers of this complexity (Ulrich and Eppinger 1999, Wheelwright and Clark 1992, Smith and Reinertsen 1991, Clark and Fujimoto 1991). Thus far, there has not been a single comprehensive framework which includes and integrate all the aspects of project complexity in the context of NPD projects. Much attention has been devoted to the technological novelty factor. The "structural complexity" factor, as important as it is, has been relegated to a secondary level of importance (Tatikonda and Rosenthal 2001a, Swink et al 1996, Ulrich 1995, Griffin 1997a,b, Zirger and Hartley 1994,1996)

The first factor contributing to project complexity in NPD projects is the level of "product complexity". The effects of product complexity on overall project complexity were first highlighted by Clark and Fujimoto (1991) observed that if a new product contains many interrelated parts, it becomes problematic to fit them together in a coherent whole. Similarly, if a product consists of many parts, the number of possible interface combinations between them increases exponentially (Murmann 1994). These studies recognised implicitly that product complexity, in terms of the number of parts in the product (differentiation) and their inter-linkages (interconnectivity), is a powerful driver of overall project complexity. This definition of "product complexity" is, in fact, similar to the concept of "structural complexity" described by Baccarini (1993) and Williams (1997,1999a) within the project management literature.

The second factor driving "project complexity" in NPD projects is related to their degree of "innovation" (see figure 1). "Innovation" may originate from new designs incorporated in the product, new product technologies which improve the translation of customer requirements into design parameters, or a new process technologies which ensure compatibility between design specifications and process capabilities (Swink 1999, Souder and Moneart 1992).

The "innovation" factor can be broken down into two dimensions: "product newness" and "project uncertainty". "Product newness" is defined as the portion of the new product which has to be redesigned from previous generations of the same product (Griffin 1997 a,b, Wheelwright and Clark 1992). Product newness affects project complexity because if its level of important, it will lead to an exponential increase in the number of tasks to be performed to finish the project (Griffin1997a, Clark and Fujimoto 1991, Clark 1989).

Significant problems of interfaces and fitness between the new parts are likely to arise leading developers to consider more design possibilities and alternatives (Swink 1999). High levels of product newness require also important amounts of knowledge creation, transfer, and synthesis (Zirger and Hartley 1994). Such projects are associated with an intensive use of highly skilled labour, market knowledge, process ability, and considerable transfer of information among the organisation (McDermott 1999).

The second dimension of innovation is “project uncertainty”. This dimension is inherent in NPD projects since each project includes a certain jump into the unknown. The suitable means, methods, and capabilities to be deployed in a project are rarely known to at the start of the NPD project execution phase.

“Project uncertainty” occurs whenever there is a gap between “ the amount of information required to perform the task (in this case the NPD project) and the amount of information already possessed by the organisation (in this case the development team)” (Galbraith 1977). “Project uncertainty” includes many categories such as market, technological, and resource uncertainties. Market uncertainty indicates the uncertainty about the market segment targeted by the new product, and the definition and articulation of customers’ needs (Tatikonda and Montoya-Weiss 2001, Souder et al 1998, Olson et al 1995, Souder and Moneart 1992). Technological uncertainty relates to the uncertainty about the best technologies to be used in the product and/or process, and the degree of familiarity of the team with the technologies involved in the project (Swink 2000, Tatikonda and Rosenthal 2000 a,b, McDermott 1999, Souder et al 1998, Swink et al 1996, Adler 1995, Olson et al 1995, Souder and Moneart 1992). Resources uncertainty reflects the uncertainty about the quantity, quality, and mix of resources to be put in the project (Swink et al 1996, Souder and Moneart 1992). Project uncertainty increases the level of “project complexity” because the wider is the gap between the required information to perform the project and the available information within the organisation, more it becomes difficult and lengthy to perform project tasks as the learning curve is slow, problems solving methods inaccurate, and the set of possible solutions large. Prior experience is not very helpful in these situations because developers face new challenges which they have never tackled before (Olson et al 1995).

5. The NPD Project System Dynamics Model

The model focuses on the description of development process of a single project although, in practice, the dynamics observed within a project is linked to the dynamics of a portfolio of projects. However, a clear understanding of multi-project dynamics cannot be achieved unless the issues involved in single projects are deeply investigated. It is assumed that the organisational structures and the technological capabilities are defined at the beginning of the project and will not change over the project life cycle. Organisational stability ensures that co-ordination mechanisms, planning and monitoring procedures, decision-making rules, and relationships between developers and project management are defined and cleared at the beginning of the project. The technology to be used in the project is available at its starting point and developers are not offered the luxury of incorporating newer

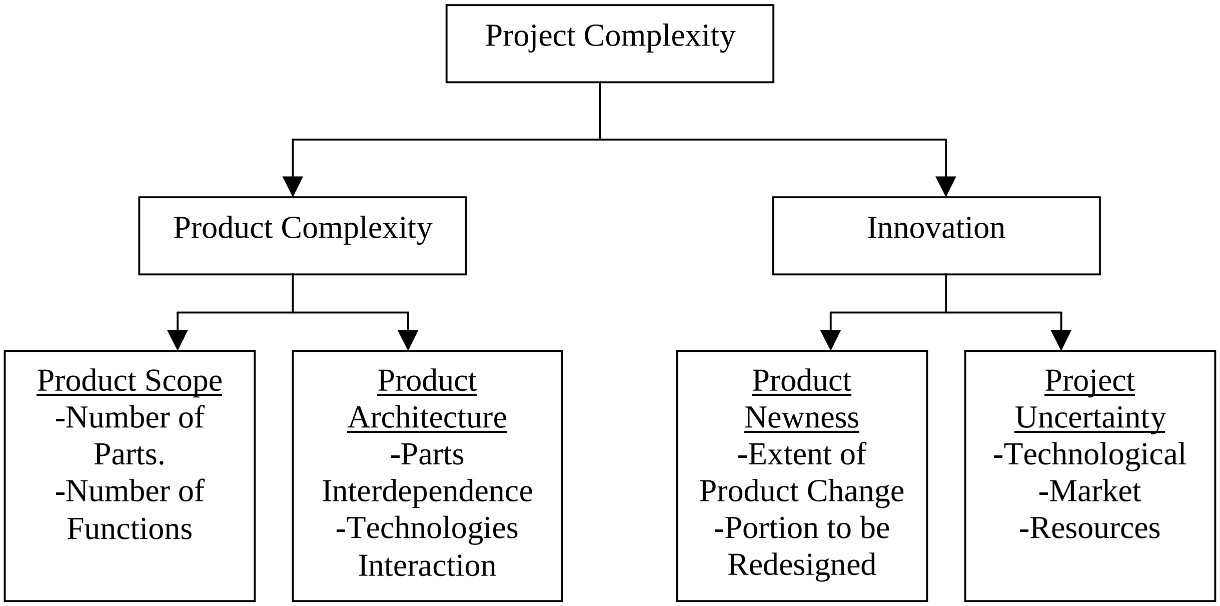


Figure 1: Project complexity factors in new product development projects

technologies once the project get underway (Gupta and Wilemon 1990). This assumption is important because many technologies related parameters in the model such as average activities duration, average delays in human resource management, and reference values for developers effectiveness are assumed constant in the model.

The project is divided into phases. A phase is a sub-structure of the project which accommodates similar set of activities (Cooper 1994). The conceptualisation of NPD projects as multi-phase processes is important within the context of this research. It enables to determine the boundaries between different project phases and to draw the line between the internal and external influences on a project phase.

To model the development project progress, it is assumed that each phase consists on a certain amount of "development tasks". When these tasks are performed, they generate the information which constitute the output of the development phase. In the current research, a development task is defined as "an atomic unit of development work" (Ford and Sterman 1998, Ford 1995). It is important to emphasise that the selection of "development tasks" in a project is phase related. For example, if the model focuses on a new car development project, "development tasks" in the product definition phase of the project would be the set of product concepts and specification for each part of the car. However, development tasks for the product design of the same project would be the set of dimensions and geometrical forms of the car components. Development tasks are important because they represent the unit of measure for project progress. In the current model, project progress is measured by the fraction of "development tasks" released from the development phase. They also serve as the base of many decisions in the project. For example, information about the number of "tasks completed", "tasks in rework", "tasks remaining", and "tasks waiting approval" are triggers of most decisions regarding human resource hiring and allocation, project targets alteration, co-ordination mechanisms, integration levels, and so on.

The last assumption in the model is related to the “project complexity” level in the project. As described earlier, project complexity consists of three factors: product complexity, product newness, and project uncertainty. The assumption here is that these three factors have a fixed level for a given project (Tatikonda and Rosenthal 2000a, Griffin 1997a). The reason is that in a project, the level of “project complexity” is set by strategic choices taken prior to the start of the development work. Such choices cannot be subject to alteration once the project work gets underway.

The framework on which the SD model presented in this research is grounded combines elements drawn from NDP theoretical frameworks briefly described earlier and from previous SD models. Theoretical frameworks provide the main variables and concepts affecting NPD performance whereas SD models provide some of the established inner feedback structures which link some of the variables found to be pertinent with respect to project performance. A review of the NPD theoretical frameworks and previous SD models indicates that NPD project performance is affected by the following:

- Development process structure: This includes the different development activities involved in a project, the level of concurrency among phases, and the structure of the development work within each phase.
- Resources and team characteristics: This represents the level and mix of resources committed to the project, the level of developers experience and training, the number of projects each developer is involved in, the proportion of full and part time developers in the project, and developers effectiveness in executing development tasks.
- Tasks Characteristics: This deals with the qualitative issues regarding development tasks. It includes the level of tasks difficulty perceived by developers, the amount of information available to carry out development tasks, and the proportion of new tasks in the project.
- Project size: This indicates the number of tasks needed to execute the project, the amount of rework due to errors, and the work to be added to the project due to initial underestimation of the real project size.
- Project objectives: This includes the setting of the initial project targets in terms of development cycle time, project cost, and project quality. It also determine the mechanisms to alter project goals, that is the project management level at which the decisions about project targets revisions are made (development team, project manager, or senior management), and the power hold by the team relatively to senior management.
- Top management support: This represents the interaction between senior management and project development team. It includes the priority given to the

project in terms of getting resources, the effects of senior managers' support on developer productivity, and the decision mechanisms related to goal alteration.

In the simulation model, each phase of the development project is represented by the generic structure presented in figure 2. In this structure, project performance in each phase is effected by the interaction of development process structure, project resources, project tasks characteristics, project size, project objectives, and the level of top management support. Development process simulates the different activities involved in the project, the nature of concurrency relationships among development phases, internal progress structures imposed by information and physical availability of work within the phase, and the generation, inheritance, and correction of errors and changes. Resources simulates the hiring and training of personnel, allocation of personnel labour among development activities, fluctuations of developers productivity in response to project conditions, learning and experience accumulation, and overtime work. Tasks characteristics simulates the effects of task difficulty and the fraction of new tasks in the project. Project size simulates the effects of changes to initial phase scope (number of tasks), the effects of rework, and the mechanisms of late work discovery and inclusion in the project. Top management support simulates the effects of senior management involvement in project objective alteration, approval of engineering changes and priority given to the project. Objectives simulate the level of initial goals and how they are changed in response to project conditions.

The project complexity factors (that is product complexity, product newness, and project uncertainty) interact with the previous operational characteristics to drive phase performance. Product complexity, which is the parameter indicating the number of tasks and their interdependence, affects development process, resources, and tasks characteristics. Product newness, which is the fraction of new tasks in the development project, affects development process, task characteristics, resources, and top management support. Project uncertainty, which represents information gaps and risks due to market, technical, environmental, and resource uncertainty have effects which span over all the operational characteristics in the project phase.

Because the model represents a multi-phase project, the interactions between operational characteristics and project complexity factors have ramifications at two levels. The phase level, which focuses on management practices taking place within a single phase of the project (intra-phase level), and project level, which focuses on those practices governing the interaction among different phases (inter phase level). These interactions are described in the following:

Intra-phase level

- A phase includes planning and execution. Development work execution does not start until sufficient information about the methods to carry out the tasks is released from the planning process.
- Some tasks cannot be planned until the phase has registered a certain progress in its work. Such tasks remain in planning until enough information from work execution is available.
- Development work execution is constrained by task planning information and task physical execution availability.
- Once a task is completed, it undergoes a testing procedure. If the tests are satisfactory, the task is approved. Otherwise, the task will have to be reworked.
- Once the tasks are approved, they are not released immediately to downstream phase. They are hold within the phase until there is sufficient information to be released.
- Changes occurring later in the phase may make some of the work already done and approved redundant. In such cases, some of the finished work will have to be done again.
- The extra rework due to late changes in a phase will have to be approved by project managers and co-ordinated with other developers before it is reworked.
- Each time a task is reworked, it undertakes the testing procedure before it is approved and released again.
- Decisions about human resource management are made within the phase according to the status of the project. The status is determined by the level of discrepancy between the phase real and planned targets.

Inter-phase level

- Development work released from an upstream phase to its downstream dependent ones determines the availability of work in downstream phase.
- Changes which escape testing procedures in an upstream phase are released to downstream phases and corrupt the work within these phases.
- If a downstream phase discover an inherited change form upstream phases, it return it to the phase which generate the change. The change is co-ordinated before it is worked again.
- Released tasks, which become redundant as a result of tasks interconnections, are recalled from downstream phases and co-ordinated before being reworked.

The simulation sectors presented in this paper are related to the development process structure part of the model. The feedback loops representing this part of the model are shown in figure3. The detailed description of the stock and flow structures (Sterman 2000, Richardson and Pugh 1981, Forrester 1961) of the development process is presented in the next section. It includes the planning sector and the development sector.

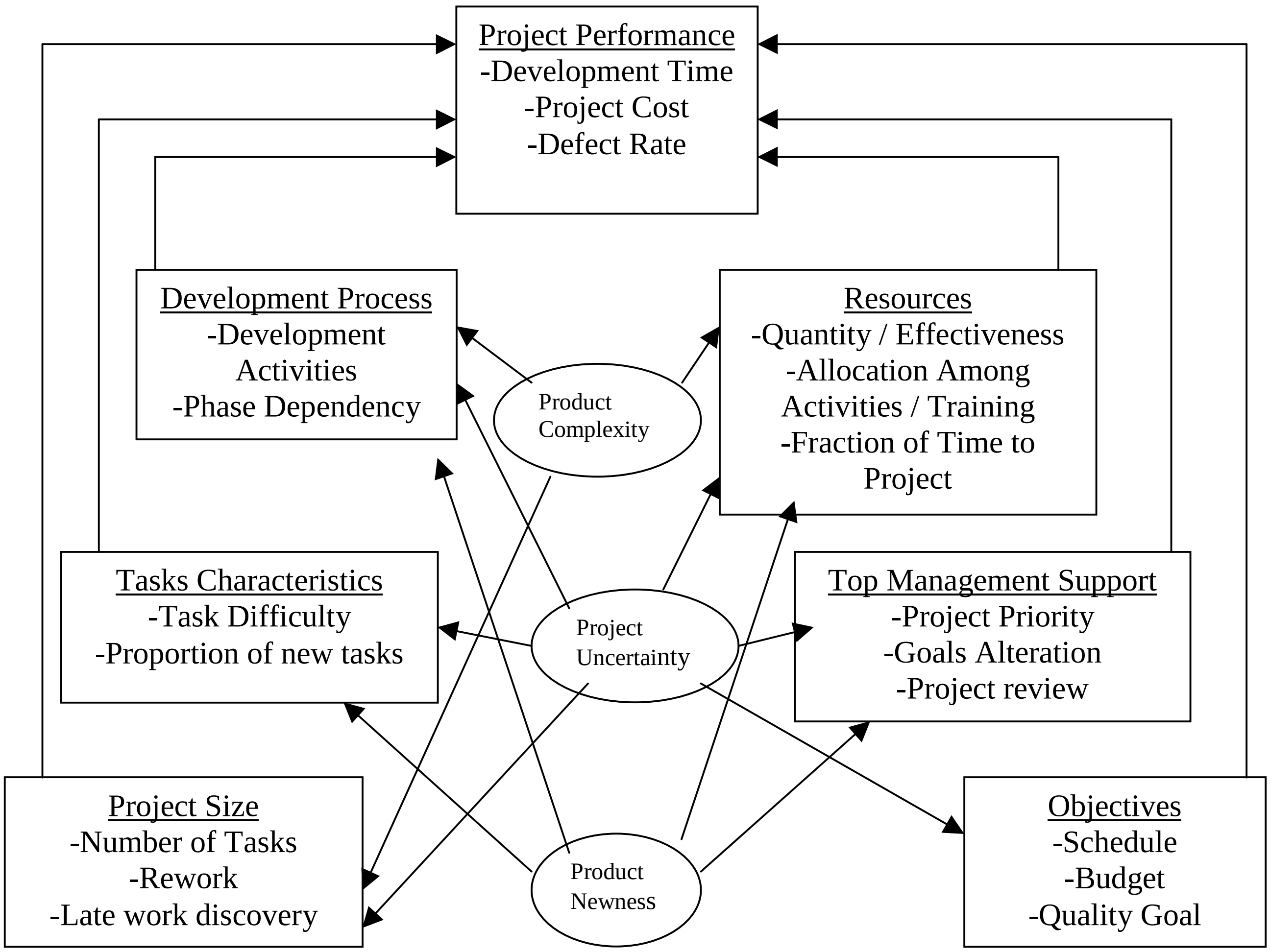


Figure 2: Generic Structure for Each NPD project phase

5.1 The Planning sector

This sector simulates planning activities, that is the activities taking place before the actual development execution starts. Planning activities are important because they set the framework within which the development execution will take place (Iansiti 1995 c). Effective project planning phase improves developers problem solving capabilities and identify potential sources of problems early in the project (Thomke and Fujimoto 2000, Khurana and Rosenthal 1998). The outcome of the project planning phase is a set of product requirements, project objectives, technology choices (Tatikonda and Rosenthal 2000 a), design and manufacturing capabilities, skills, procedures, and project structures (Adler 1995)

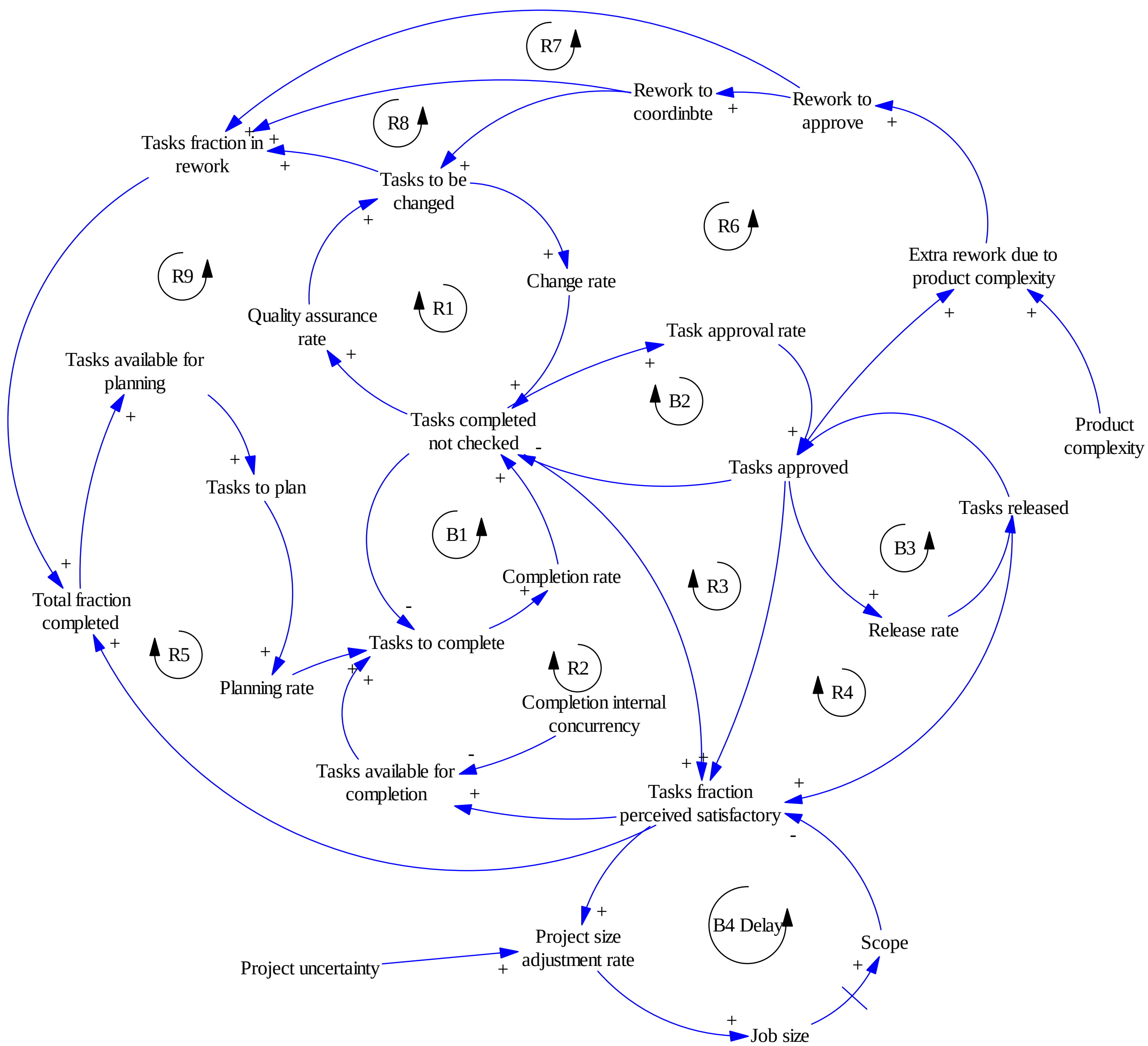


Figure 3: The feedback structure for the development process

The stock and flow diagram which simulates the project planning process is presented in figure 4. The stock of tasks to be planned T_{TP} whose initial value is equal to the phase scope (number of development tasks in the development phase), is depleted at a certain rate per unit time determined by the planning rate:

$$\frac{d}{dt}(T_{TP}) = -a \qquad (1)$$

The planning rate (a) is determined by the minimum value allowed by the planning development structure process (the number of tasks available for planning at any given time) and the planning labour which represents the availability of labour to perform the planning activities.

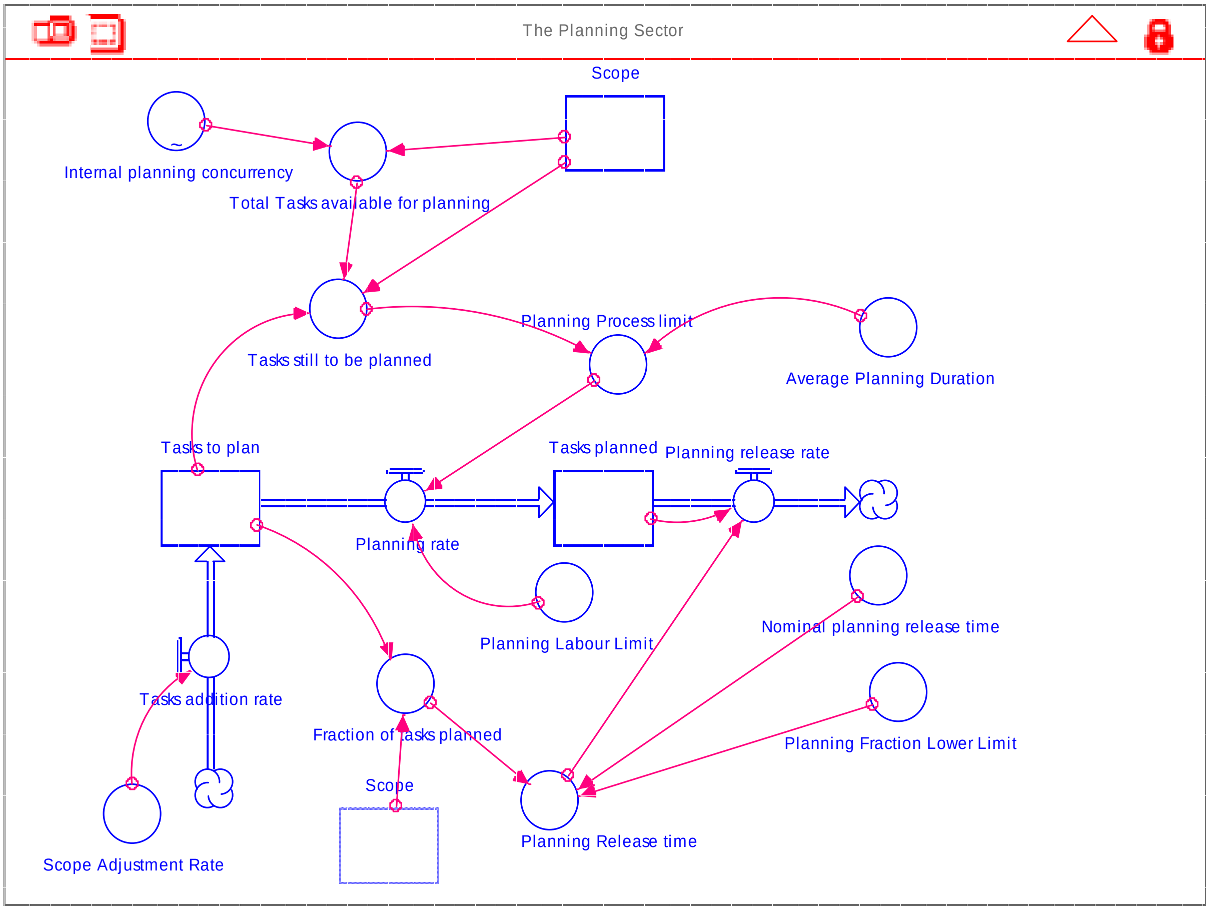


Figure 4: The planning sector stock and flow diagram

$$a = \text{Min}\ (P_{PL}, P_{LA}) \qquad (2)$$

P_{PL} is equal to the tasks still to be planned (T_{SP}) divided by the average planning duration (τ_{PL}).

$$P_{PL} = T_{SP} / \tau_{PL} \qquad (3)$$

The variable T_{SP} has been taken here instead of the stock of tasks to plan T_{TP} because at the beginning of the development phase, not all tasks can be planned. It is virtually impossible to specify all tasks completely before project execution begins (Iansiti 1995c). Information on how to proceed to plan some tasks may not be available until late into the development phase. This characteristic of the planning activity is known as the "isolation and absorption" principle (Laufer et al 1996). In some situations, it is necessary to isolate some tasks which are impossible to plan at the phase start, plan and execute other tasks in the meantime, and then use the generated information by the development execution process to plan the isolated tasks.

In this context, at any given time, the total number tasks available for planning (TT_{AP}) is not equal to the phase scope S, but to a fraction of it. The value of TT_{AP} is equal to:

$$TT_{AP} = S \times P_{CC} \qquad (4)$$

Planning concurrency P_{CC} is a non-linear function, which represents the fraction of tasks available for planning with respect to the fraction of tasks which has been already completed at least once. The latter fraction is equal to the fraction of tasks perceived satisfactory (F_{TPS}) and the fraction of tasks in rework (F_{TR}) (the variables F_{TPS} and F_{TR} will be described in more details in the process development sector). Therefore:

$$P_{CC} = f\,(F_{TPS} + F_{TRW}) \qquad (5)$$

The number of tasks still to be planned T_{SP} is equal to the total tasks available for planning TT_{AP} minus the number of tasks which have been already planned. The latter number of tasks is equal to the difference between the phase scope S and the stock of tasks to be planned $T_{TP.}$ This lead to the following equation:

$$T_{SP} = \mathrm{Max}\,(0, TT_{AP} - (S - T_{TP})) \qquad (6)$$

The tasks which have been planned are not released immediately for completion. The information generated by the planning process is kept for a while until a sufficient amount is accumulated to allow a smooth starting of the tasks completion process. Between the instants that tasks are planned and released fore completion, tasks are accumulated in a stock of tasks planned $T_{PLAN.}$ This stock change over time at the following rate:

$$d/dt\,(T_{PLAN}) = a-b \qquad (7)$$

The planning release rate depends on how much of the tasks have been already planned. To ensure that planned tasks are not released immediately, a minimum fraction of tasks has to be planned. Once this threshold value is reached, planning information starts to flow to development execution. The threshold value is set in the model through the parameter lower planning limit $L_{PLLIMIT}$ which is chosen in practice by the modeller. Therefore:

$$b = \begin{cases} T_{PLAN} / \tau_{PLRLT} & \text{If } FT_{PLAN} \geq L_{PLLIMIT} \\ 0 & \text{Otherwise} \end{cases} \qquad (8)$$

The fraction of tasks planned is given by the number of tasks planned to date divided by the phase scope. The number of tasks already planned is equal to the phase scope minus the stock of tasks to plan (Equation 6). Consequently, the fraction FT_{PLAN} is given by the equation:

$$FT_{PLAN} = (S - T_{TP}) / S \qquad (9)$$

5.2 The Development Sector

This sector simulates the development process within a development phase. It includes the mechanisms which alter development tasks states after tasks have undertaken a planning phase. In the current model, each possible state of a development task is represented by a stock. The movement of tasks among stocks (states) is dictated by the rates at which the development activities are performed. The stock and flow diagram representing the development sector is shown in figure 5.

The development sector includes four stocks which determine the completion and release of development tasks. These stocks are "Tasks to complete T_{TC}", "Tasks completed not checked T_{CNC}", "Tasks approved T_{APPRV}", and "Tasks released $T_{RELEASE}$".

The tasks to complete stock T_{TC} receives the tasks moving from the tasks planned stock T_{PLAN}. Once the threshold planning fraction $L_{PLLIMIT}$ is reached, tasks starts to move to the stock of tasks to complete T_{TC} signalling the start of the development execution phase. Tasks are executed and move to the stock of tasks completed not checked T_{CNC}. The rate in which the tasks move between these two stocks is dictated by the base-work activity. Base-work activity is defined as the execution of a development task for the first time.

Once a task is completed for the first time, it undertakes a verification process to make sure that it is not flawed. If the task pass this checkpoint successfully, it is approved. Otherwise, the task will have to be corrected (reworked) another time to address the flaw found in the task. From a stock and flow perspective, this means that there are two outflows from the stock of tasks completed not checked T_{CNC}. The first outflow moves tasks to the stock of approved tasks T_{APPRV}. The second outflow moves tasks to the stock of tasks to be reworked T_{REWORK}. The rate of these two outflows is determined by the quality assurance (QA) activity, which represents the verification process occurring within a development phase. The tasks which move to the stock of tasks to be reworked T_{REWORK} move again to the stock of tasks completed not checked T_{CNC} where they are checked again for possible flaws. The rate at which the tasks move from the former to the latter stock is determined by the rework activity, which indicates the work carried out on a task for correction purposes.

The tasks found correct after the quality assurance activity are hold in the stock of tasks approved T_{APPRV} until they are released to subsequent phases. This process is represented by the movement of tasks from the stock of approved tasks T_{APPRV} to the stock of tasks released $T_{RELEASE}$. Task release means that the information generated through development work within a phase is transferred to the subsequent phases. For example, tasks released from a product definition phase in a new car development project would be a set of the car's parts specifications.

The equations which represent the previous stock changes over time are in the following:

$$d/dt\ T_{TC} = b-c \qquad (10)$$
$$d/dt\ T_{CNCH} = c+g-d-e \qquad (11)$$
$$d/dt\ T_{APPRV} = e-f-h \qquad (12)$$
$$d/dt\ T_{RELEASE} = f \qquad (13)$$

The base-work completion rate c is equal to the minimum value between the base-work process limit and the base-work labour limit. The base-work process limit, which is the rate of base-work completion allowed by the process structure, depends on base-work availability. In this context, base-work availability BSW_{AV} is the minimum value between the base-work available due to concurrency BAV_{CC} and base-work available due to planning BAV_{PL}.

$$BSW_{AV} = \mathrm{Min}\ (BAV_{CC}, BAV_{PL}) \qquad (14)$$

The value of BAV_{PL} represents the information constraints put on the availability of development work due to lack of tasks released from the planning activity. BAV_{CC} represents the physical constraints put on the availability of the current work given the work already done. The latter constraint is similar the internal gates presented in previous SD models (Notzon 2000, Ford and Sterman 1998,1999, Ford 1995, Homer et al 1993). It represents how much work is currently available given the work which has been already finished.

The value of base-work available due to planning BAV_{PL} is simply equal to the stock of tasks to complete T_{TC}. The value of BAV_{CC} is much more complicated to determine. In a phase, not all the development work is available for completion once the planning stage is finished. Some development work remains unavailable for completion unless some pre-required tasks have been performed. For example, if a new car is being developed, the design of some components cannot start until parameters and geometrical form of the car body has been determined.

To model this type of constraints, we will use a base-work process concurrency BS_{CC} function which determines the fraction of the phase scope S available for base-work completion given the fraction of tasks perceived satisfactory F_{TPS} by developers. The latter value is the total tasks perceived satisfactory TT_{PS}, which is equal to the sum of tasks completed not checked T_{CNCH}, tasks approved T_{APPRV}, and tasks released $T_{RELEASE}$. Therefore, at any given time, the total tasks available for base-work completion TT_{BSWC} is equal to the product of the phase scope S and the base-work process concurrency BS_{CC}.

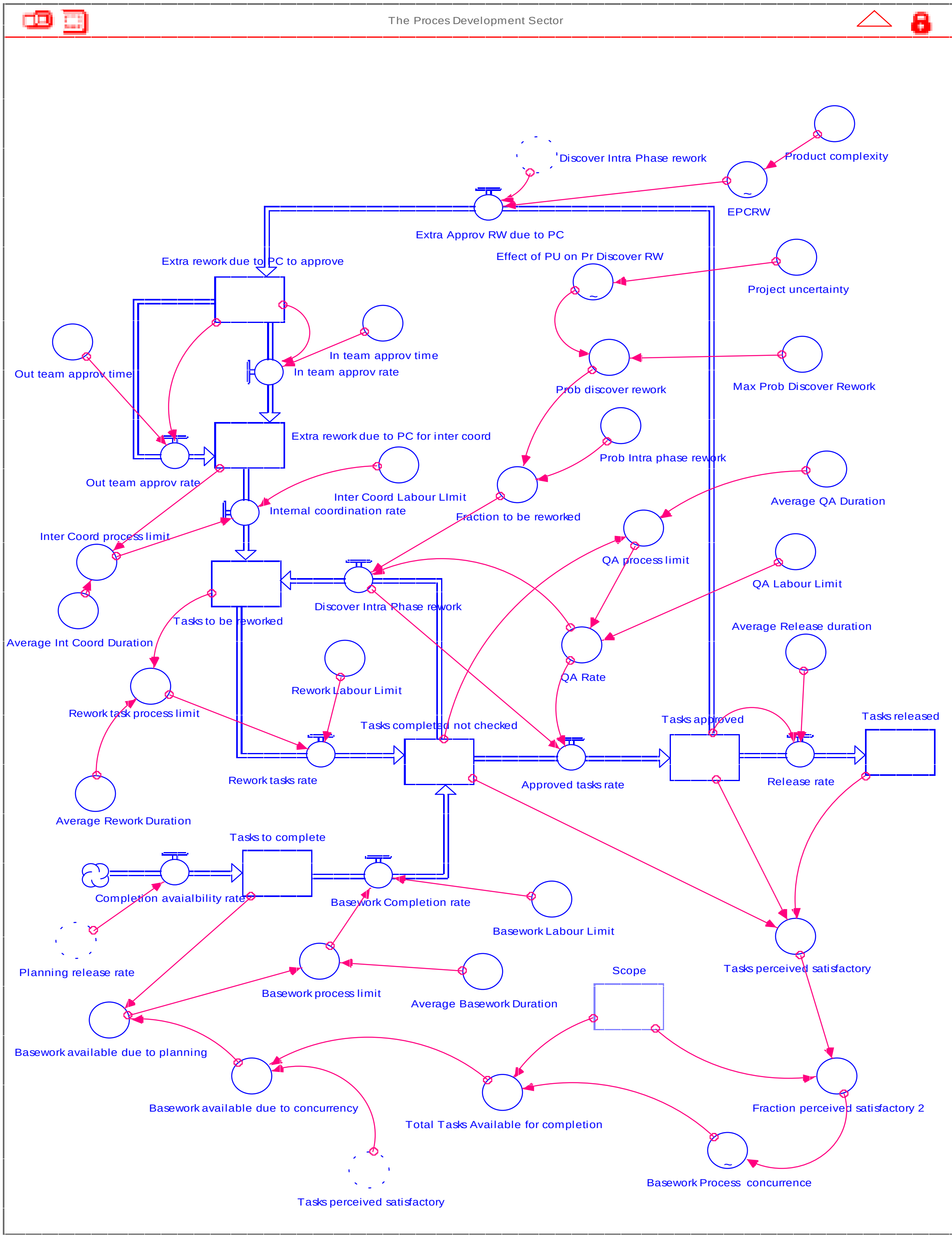


Figure 5: The development sector stock and flow diagram

$$TT_{PS}= T_{CNCH} + T_{APPRV}+ T_{RELEASE} \quad (15)$$
$$F_{TPS}= TT_{PS} / S \quad (16)$$
$$BS_{CC}= f(F_{TPS}) \quad (17)$$
$$TT_{BSWC}=S \times BS_{CC} \quad (18)$$

Total tasks available for base-work completion TT_{BSWC} includes the tasks which have been already executed at least once and the tasks which are still to be completed. Therefore, the number of tasks corresponding to base-work available due to concurrency BAV_{CC} is equal to the total tasks available for base-work completion TT_{BSWC} minus the tasks which have already completed at least once. The latter number of tasks is equal to the phase scope S minus the total tasks perceived satisfactory TT_{PS} and the total tasks in rework $TT_{RW.}$

$$BAV_{CC}= \mathrm{Max}\ (0, TT_{BSWC}- (S-TT_{PS}-TT_{RW})) \quad (19)$$

The maximum condition is put here to avoid situations when an important number of tasks are still in planning phase. Such situation means that the values of TT_{PS}, TT_{RW}, and TT_{BSWC} are low and therefore BAV_{CC} may become negative, a meaningless situation. In practice, this reflects the situation when most of the tasks are still in planning and not enough information is being released to start completion work and, therefore, physical constraints on base-work evolution have not yet become effective.

As mentioned earlier, the activity associated with task verification is defined here as quality assurance (QA). The rate at which this activity is carried out is determined by the minimum value between the quality assurance process limit QA_{PL} and the quality assurance labour limit QA_{LL}

$$QA = \mathrm{Min}\ (QA_{PL}, QA_{LL}) \quad (20)$$

The quality assurance process limit QA_{PL} is equal to the stock of tasks completed not checked T_{CNCH} divided by the average quality assurance duration τ_{QA}

$$QA_{PL}= T_{CNCH} / \tau_{QA} \quad (21)$$

Once flawed tasks are discovered, they move from the stock of tasks completed not checked T_{CNCH} to the stock of tasks to be reworked T_{REWORK}. The rate d at which tasks move between these two stocks depends on the quality assurance rate QA and the fraction of tasks which needs to be reworked FN_{REWORK}. The latter fraction is the product of the probability that a task needs rework (task flawed) PN_{REWORK} and the effectiveness of developers to detect flawed tasks. Developers' effectiveness to detect flawed tasks is quantified through the probability of discovering a rework $PR_{DSREWORK}$.

$$d= QA \times FN_{REWORK} \quad (22)$$
$$FN_{REWORK}=PN_{REWORK} \times PR_{DSREWORK} \quad (23)$$

The tasks which are not detected to need a changed are approved and move from the stock of tasks completed not checked T_{CNCH} to the stock of tasks approved T_{APPRV}. The tasks approval rate e is simply the quality assurance rate QA minus the rate at which changes are discovered d.

$$e = QA - d \qquad (24)$$

The probability of discovering a rework $PR_{DSREWORK}$ is affected by the project uncertainty level PU. Uncertainty means that there is insufficient information about the task, so chances are high that a flawed task is considered correct and, therefore, is mistakenly approved. Therefore, the probability of discovering a rework $PR_{DSREWORK}$ is equal to the product of a maximal probability to discover a rework $MAXPR_{DSREWORK}$ and the effects of project uncertainty $EPU_{DSREWORK}$. The latter effect is a non-linear function with respects to project uncertainty PU.

$$PR_{DSREWORK} = MAXPR_{DSREWORK} \times EPU_{DSREWORK} \qquad (25)$$

$$EPU_{DSREWORK} = f(PU) \qquad (26)$$

The flawed tasks due to developers' work errors are not the only tasks to be reworked. In fact, product complexity PC, which is a parameter capturing the interconnections among tasks, affects also the number of tasks to be reworked. One of the most important consequences of tasks connectivity is that if some late tasks are found flawed, some other tasks which have been previously approved and released will have to be reworked again. <u>This is the situation in which a change in a task, such as the design of a component, generates more work than the original one(Williams 1999b).</u>

The process of changes due to product complexity is modelled by the introduction of two new stocks "Extra rework due to PC to approve $RW_{EXTAPPROV}$" and "Extra rework due to PC for internal co-ordination $RW_{EXTINTCOOR}$". Tasks which become redundant as a result of product complexity will move form the stock of tasks approved T_{APPRV} to the stock of Extra rework due to PC to approve $RW_{EXTAPPROV}$. This approval is necessary because these changes have not occurred because of developers mishandling of development work, but due to task interconnections. These changes go generally through an approval stage before they are reworked again (Loch and Terwiesch 1999, Williams 1999b). Once these changes are given the "go-ahead" decision for rework, <u>they will have to be co-ordinated within the development phase</u> before the actual rework takes place.

The movement of changes due to work made redundant as a result of product complexity is determined by the following rates: approved rework rate due to product complexity h, in-team approval rate k, out-team approval rate l, and internal co-ordination rate m. The changes over time of the stocks representing detection and correction of flawed tasks and changes due to product complexity are:

$$d/dt\ T_{REWORK} = d+m-g \qquad (28)$$

$$d/dt\ RW_{EXTAPPROV}. = h-k-l \qquad (29)$$

$$d/dt\ RW_{EXTINTCOOR} = k+l-m \qquad (30)$$

The rate h depends on the level of product complexity PC. The higher the value of this parameter, the higher are the interconnections among tasks, and hence it is more likely that a flawed task makes more already accomplished tasks redundant. This relationship is captured by a non-linear function which determines the effect of product complexity on rework EPC_{RW} with respect to the parameter product complexity PC.

$$EPC_{RW} = f\,(PC) \qquad (32)$$

The second parameter determining this rate is the discover rework rate d. In fact, redundant tasks due to product complexity become so because they are linked to tasks which are being found flawed through quality assurance activity. Therefore, rate h depends on the rate d. More flawed tasks discovered, more interconnected tasks to them become redundant. Consequently, the equation for h is

$$h = Max\,(0, FT_{APPRV} \times d \times EPC_{RW}) \qquad (33)$$

These tasks have to be approved by project managers before they are reworked. To model this process, I used a quantification of team approval power similar to the one presented in Zirger and Hartely (1996). They determined it as “the fraction of decisions outside the team control”. This fraction is modelled here as a probability that a task needs an out-team approval $PR_{OUTTEAM}$. Therefore, the rates of changes approval k and l, that is in-team and out-team approval rates respectively, are given by

$$k = (RW_{EXTAPPROV} / \tau_{INTEAM}) \times (1 - PR_{OUTTEAM}) \qquad (35)$$
$$l = (RW_{EXTAPPROV} / \tau_{OUTTEAM}) \times PR_{OUTTEAM} \qquad (36)$$

The average in-team approval duration τ_{INTEAM} is in fact equal to dt, the time unit, because a change which does not require project management approval moves instantaneously to the stock of extra rework due to PC for internal co-ordination $RW_{EXTINTCOOR}$

Internal co-ordination activity takes place to address integration concerns between developers working within the same development phase but on different components of the product. Internal co-ordination takes place in high product complexity environments because developers work generally on different product “sub-systems” and “components” simultaneously (Clark and Fujimoto 1991)

The rate m at which this activity is performed is determined by the minimum value between the internal co-ordination process limit INC_{PL} and the internal co-ordination labour limit INC_{LL}

$$m = Min\,(INC_{PL}, INC_{LL}) \qquad (37)$$

The internal co-ordination process limit INC_{PL} is determined by the stock of extra rework due to PC for internal co-ordination $RW_{EXTINTCOOR}$ and the average internal co-ordination duration $\tau_{INTCOORD}$.

$$INC_{PL} = RW_{EXTINTCOOR} / \tau_{INTCOORD} \qquad (38)$$

Before the description of the development sector come to conclusion, it is necessary to recall that in the planning sector described previously, two quantities have been mentioned without explanations: Fraction of tasks perceived satisfactory F_{TPS} and fraction of tasks in rework F_{TRW} (Equation 5). F_{TPS} is presented in equation (16) above. Fraction of tasks in rework F_{TRW} is equal to the total tasks in rework TT_{RW}, that is the number of tasks in rework or waiting to be changed, divided by the phase scope S. The former value is equal to the number of tasks in the stock of extra rework due to PC to approve $RW_{EXTAPPROV}$, extra rework due to PC for internal co-ordination $RW_{EXTINTCOOR}$, and tasks to be reworked T_{REWORK}

$$F_{TRW} = TT_{RW} / S \qquad (39)$$

$$TT_{RW} = T_{REWORK} + RW_{EXTAPPROV} + RW_{EXTINTCOOR} \qquad (40)$$

Finally, activities duration depends on the level of innovation in the project. As described earlier, innovation has two components: product newness PN and Project uncertainty PU. The former refers to the size of change taking place in the new product whereas the latter indicates the level of difficulty associated with these changes. Clearly, both parameters influence activities duration time. A sizeable change will take longer because there are more new tasks to perform as the number of alternatives to consider is important (Murmann 1994). A difficult task requires more time because of lack of information on how to proceed with it (Emmanuelides 1993, Iansiti 1995b,c). Therefore the average activity duration time is equal to the product of the minimum activity duration $\tau_{Min,ACT}$ and the effects of product newness EPN_{TM} and project uncertainty EPU_{TM}. The two effects are non-linear function depending on the product newness PN and project uncertainty PU levels.

$$\tau_{ACT} = \tau_{Min,ACT} \times EPN_{TM} \times EPU_{TM} \qquad (41)$$

$$EPN_{TM} = f(PN) \qquad (42)$$

$$EPU_{TM} = f(PU) \qquad (43)$$

ACT = {Planning, base-work, quality assurance, rework, internal co-ordination}

6. Simulation results for single phase model

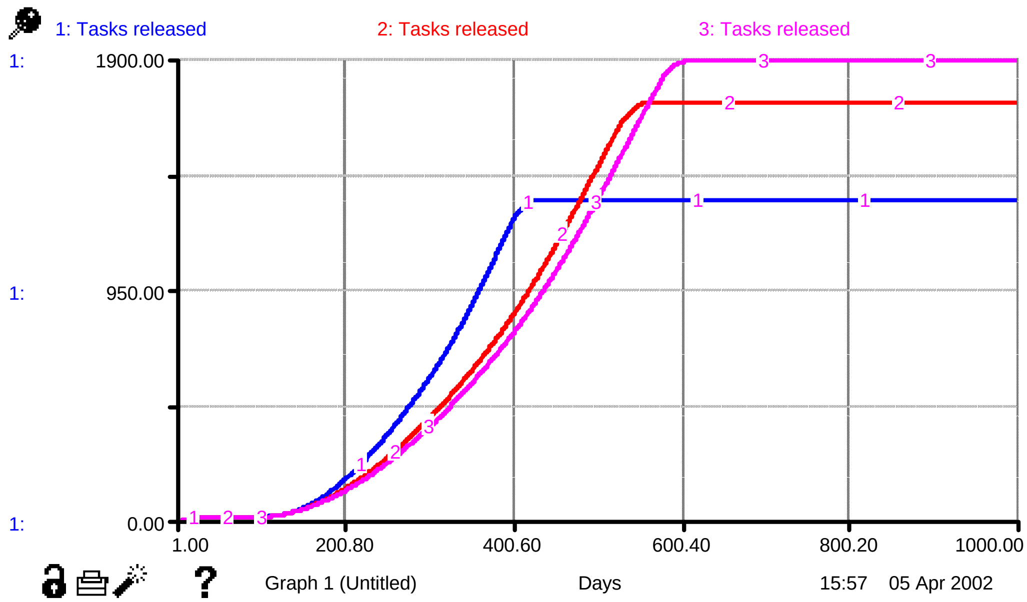
In order to get some early results, the single-phase simulation model was run to investigate how the integration within a development phase affects its performance. Although, the model generates the behaviour of three performance indicators: development cycle time, project quality, and project cost, the discussion here is limited to the behaviour of the development cycle time. Discussions related to the multi-phase model and other performance indicators will be the subject of a separate paper. The parameters values used in these runs are not from a real project, but from some of the previous SD models. Formal parameterisation is being undertaken currently on a project within the Aerospace industry.

In the set of simulation runs described here, I analysed the effect of interaction between two project complexity factors, that is product complexity (PC) and project uncertainty (PU), and integration represented by the internal co-ordination activity. PC is assigned two values: low and high whereas PU is assigned three values: low, reference, and high. Integration is evaluated through two levels: low and high. Development cycle time is represented by the time needed to release all the phase tasks. For each level of integration and product complexity, a sensitivity analysis was conducted with respect to project uncertainty. The behaviour over time of the variable “tasks released” is presented in graphs 1 to 4.

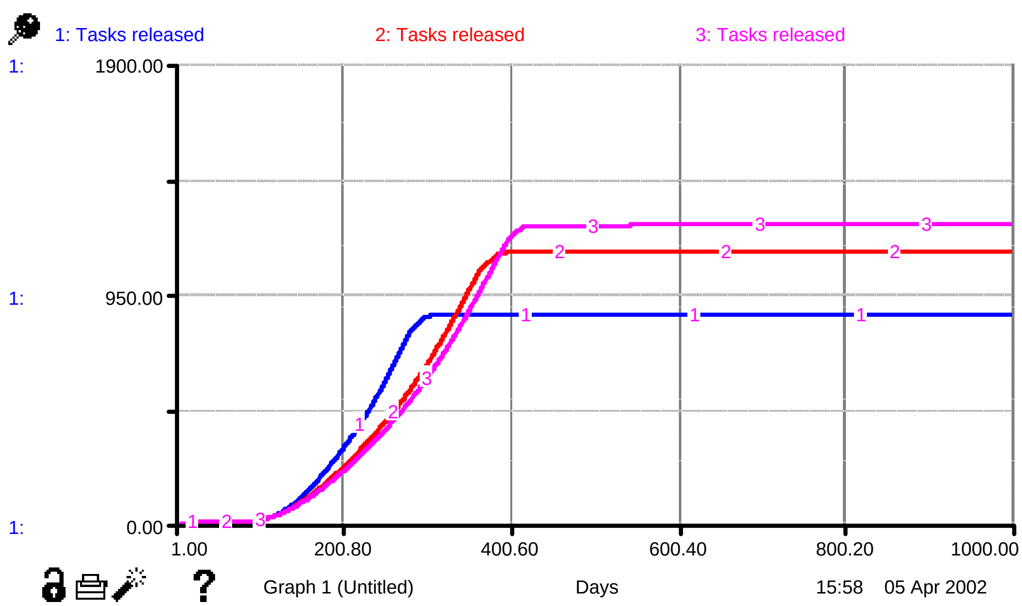
The graphs show that development cycle time increases as the level of project uncertainty goes up from low to high. This behaviour holds for both situations of low and high PC. However, development cycle time increase is more dramatic when PU moves from its low to reference value. The jump is not equally important when the value of PU moves from reference to high value. This indicates that there is a decreasing marginal effect of PU on the development cycle time. Although the effect of PU on cycle time is dramatic, once PU reaches a certain level, its increasing effect is no longer strong. This shows that even moderate levels of project uncertainty may have considerable effects on cycle time.

From the product complexity perspective, a similar picture is emerging. If PC increases from low to high level, development cycle time increases for the three levels of PU. This also indicates that PC is an important driver of development cycle time. However, in this situation, the development cycle time increase is similar for the three levels of PU. This support the view that firms should be careful when they decide about the architecture of the new product to be developed, an aspect of project complexity which has been often relegated to a secondary level of importance in NPD projects.

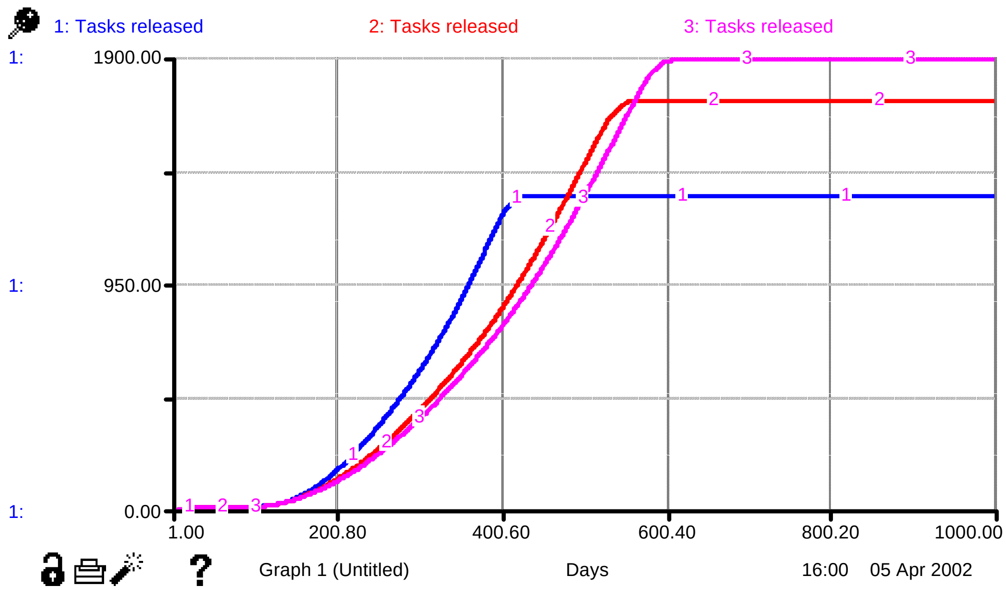
It appears also from the results that integration is not very helpful in compensating the effects of project complexity. No significant improvement in performance has been made when integration moves from low to high level. However, this may be understandable in the single-phase model. Internal co-ordination is associated only with the redundant work due to product complexity. I suspect that even if this activity is given a high priority, it cannot offset the negative consequences of high PC and PU. This indicates that it may be more beneficial to concentrate the efforts on other execution activities such as planning and base-work to avoid generating errors in the first place. In any case, this is just a preliminary analysis regarding the effects of integration. A more detailed and in-depth analysis will be presented in a coming paper describing the multi-phase model.



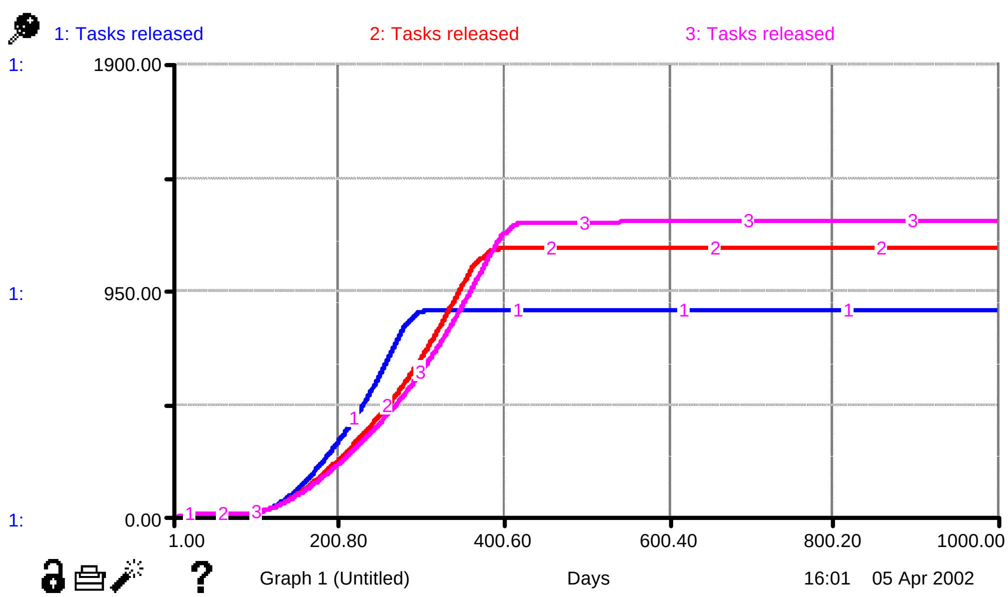
Graph 1: Development cycle time (high integration, high product complexity)



Graph 2: Development cycle time (high integration, low product complexity)



Graph 3: Development cycle time (low integration, high product complexity)



Graph 4: Development cycle time (low integration, low product complexity)

References

Abdelhamid TK and Madnick S.1991. "*Software project management dynamics: an integrated approach*". Prentice Hall, New Jersey.

Adler PS.1995. "Interdepartmental interdependence and co-ordination: the case of the design/manufacturing interface". *Organization Science*, **6**(2), 147-167.

Baccarini D. 1996. "The concept of project complexity-a review". *International Journal of Project Management*, **14**(4), 201-204.

Brown SL and Eisenhardt KM.1995. "Product development: Pat research, present findings, and future directions". *Academy of Management Review*, **20**(2), 343-378.

Carmel E.1995. "Cycle time in packaged software firms". *Journal of Product Innovation Management*, **12**(1), 1-14.

Clark KB.1989. "Project scope and project performance: The effect of part strategy and supplier involvement on product development". *Management Science* **35**, 1247- 1263.

Clark KB and Fujimoto T.1991. "*Product development performance: strategy, organization and management in the world auto industry*". Harvard Business School Press, Cambridge, MA.

Cooper RG.1994. "Perspective: Third generation new product process". *Journal of Product Innovation Management*, **11**, 3-14.

De Toni A, Nassimbeni G, and Tonchia S.1999. "Innovation in product development within the electronics industry". *Technovation,* **19**, 71-80.

Diehl E and Sterman JD.1995. "Effects of feedback complexity on dynamic decision making". Organizational Behavior and Human Decision Processes, V 62(2), 198-215.

Dreger JB.1992. "*Project management*". Van Nostrand Reinhold, New York.

Emmanuelides PA.1993. "Towards an integrative framework of performance in product development projects". *Journal of Engineering and Technology Management*, **10** 363-392.

Eisenhardt KM and Tabrizi B.1995. "Accelerating adaptive processes: product innovation in the global computer industry". *Administrative Science Quarterly*, **40**, 84-110.

Ford ND.1995. "The dynamics of project management: an investigation of the impacts of project process and co-ordination on performance". Unpublished PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Ford ND and Sterman JD.1998. "Dynamic modelling of product development processes". *System Dynamics Review* **14**(1), pp 31-68.

Ford ND and Sterman JD .1999."Overcoming the 90% syndrome: Iteration management in concurrent development projects". Unpublished working paper, System dynamics group, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.

Forrester JW (1961). "*Industrial dynamics*", The MIT press, Cambridge, MA.

Galbraith JR (1977). "*Organization Design*". Reading MA, Addison-Wesley

Gerwin D and Susman G.1996. "Special issue on concurrent engineering". *IEEE Transactions on Engineering Management*, **43**(2), 118-123.

Griffin A.1997a. "The effects of project and process characteristics on product development cycle time". *Journal of Marketing Research* **34**(1), 24-35.

Griffin A 1997b. "Modelling and measuring product development cycle time across industries". *Journal of Engineering and Technology Management*, **14**, 1-24

Gupta AK and Wilemon DL 1990. “Accelerating the development of technology-based new products”. *California Management Review*, **Winter 1990**, 24-44.

Hanfield RB.1994. “Effects of concurrent engineering on make to order products”.*IEEEtransactions on Engineering Management,* **41**, 384-393.

Hauptman O and Hirji KK.1996. “The Influence of process concurrency on project outcomes in product development: An empirical study of cross-functional teams”. *IEEE Transactions on Engineering Management,* **43**(2), 153-164

Henke JW, Krachenberg RA and Lyons TF.1993. “Perspective: cross-functional teams, good concept, poor implementation”. *Journal of Product Innovation Management* **10** 216-229.

Homer JB, Sterman JD, Greenwood B and Perkola M.1993. “Delivery time reduction in pump and paper mill construction projects” Proceedings of the 1993 International System Dynamics Conference, Cancun, Mexico. Monterey Institute of Technology.

Hull FM, Collins PD, and Liker JK.1996. “Composite forms of organisation as a strategy for concurrent engineering effectiveness”. *IEEE Transactions on engineering Management*, **43**(2),133-142.

Iansiti M.1995a. “Science based product development: An empirical study of the mainframe computer industry”. *Production and Operations Management,* **4** 353-59.

Iansiti M.1995b. “Technology development and integration: An empirical study of the interaction between applied science and product development”. *IEEE Transactions on Engineering Management,* **42**(3), 259-269.

Iansiti M.1995c. “Technology integration: Managing technological evolution in a complex environment”. *Research Policy,* **24**, 521-542.

Khurana A and Rosenthal SR.1998. “Towards holistic “Front Ends” in new product development”. *Journal of Product Innovation Management,* **15**, 57-74.

Krishnan V.1996. “Managing the simultaneous execution of coupled phases in concurrent product development”. *IEEE Transactions on Engineering Management,* **43**(2), 210- 217

Krishnan V, Eppinger SD and Whitney DE.1997. “A model based framework to overlap product development activities”. *Management Science,* **43**(4), 437-451.

Langerak F, Peelen E and Nijssen E.1999. “A laddering approach to the use of methods and techniques to reduce the cycle time of the new to the firm products”. *Journal of Product Innovation Management,* **16**, 173-182.

Larson EW and Gobeli DH.1989. “Significance of project management structure on development success”. *IEEE Transactions on Engineering Management,* **36**(2), 119-125.

Laufer A, Denker GR, and Shenhar AJ.1996. “Simultaneous management: the key to excellence in capital projects”. *International Journal of Project Management,* **14**(4), 189-199.

Liker JK, Collins PD, and Hull FM.1999. “Flexibility and standardisation: Test of a contingency model of product design-manufacturing integration”. *Journal of Product Innovation Management*, **16**, 248-267.

Loch HL and Terwiesch C .1999. “Accelerating the process of engineering change orders: Capacity and congestion effects”. *Journal of Product Innovation Management*, **16**, 145-159.
McDermott CM.1999. “Managing radical product development in large manufacturing firms: a longitudinal study”. *Journal of Operations Management*, **17**, 631-644.
Meyer MH, Tertzakian O and Utterback JM.1997. “Metrics for managing research and development in the context of the product family”. *Management Science*, **43**(1), 38-111.
Milson MR, Raj RP and Wilemon D.1992. “A survey of major approaches for accelerating new product development”. *Journal of Product Innovation Management*, **9**, 53-69.
Moder JJ, Philips CR and Davis EW.1983. “*Project management with CPM, PERT and precedence diagramming*”. Van Nostrand Reinhold, New York.
Murmann PA.1994. “Expected development time reductions in the German mechanical engineering industry”. *Journal of Product Innovation Management*, **11**, 236-252.
Notzon I.2000. “Effective design of phase overlapped product development processes- a system dynamics based analysis”. Unpublished Mphil thesis, University of Bergen.
Olson EM, Walker OC and Reukert RW.1995. “Effective new product development: the moderating role of product innovativeness” *Journal of Marketing*, **59**, 48-62..
Page AL.1993. “Assessing new product development practices and performance: establishing crucial norms”. *Journal of Product Innovation Management*, **10**, 273-290.
Pinto JK, and Covin JG.1989. “Critical factors in project implementation: A comparison of construction and R&D projects”. *Technovation*, **9**, 49-62.
Richardson GP and Pugh III AL .1981. “*Introduction to system dynamics with DYNAMO*”. The MIT press, Cambridge, MA
Rodriguez AG and Bowers J.1996. “System dynamics in project management: a comparative analysis with traditional methods”. *System Dynamics Review* **12**(2), 121-139.
Shenhar AJ.1998. “From theory to practice: towards a typology of project management styles”. *IEEE Transactions on Engineering Management*, **45**(1), 33-48.
Smith PG and Reinertsen DG.1991. “*Developing products in half the time*”. Van Nostrand Reinhold, New York.
Souder WE and Moenaert RK.1992. “Integrating marketing and R&D project personnel within innovation projects: An information uncertainty model”. *Journal of Management Studies*, **29**, 485-512.
Souder WE, Sherman JD, and Davis-Cooper R.1998. “Environmental uncertainty, organisational integration, and new product development effectiveness: a test of contingency theory” *Journal of Product Innovation Management*, **15**, 520-533.
Sterman JD.1992. “System dynamics modeling for project management”. Unpublished working paper, System dynamics group, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA.
Sterman JD.1994.”Learning in and about complex systems”. *System Dynamics Review* **10**(2-3), 291-330.

Sterman JD.2000. "*Business dynamics. Systems thinking and modeling for a complex world*". Mc-Graw Hill/Irwine, International edition.
Steward DV.1981. "The design structure system: a method for manging the design of complex systems". *IEEE Transactions in Engineering Management*, **EM-28**,3 ,71-74.
Swink ML.1998. "A tutorial on implementing concurrent engineering in new product development". *Journal of Operations Management*, **16**, 103-116.
Swink ML.1999. "Threat to new product manufacturability and the effects of development team integration processes". *Journal of Operations Management*, **17**, 691-709.
Swink ML.2000. "Technological innovativeness as a moderator of new product development design integration and top management support". *Journal of Product Innovation Management*, **17**, 208-220.
Swink ML, Sandvig CJ and Pearson AW.1996. "Customizing concurrent engineering: Five case studies". *Journal of Product Innovation Management*, **13**(3), 229-244.
Takeuchi H and Nonaka I .1986. "The new new product development game". *Harvard Business Review*, **January-February**, 137-146.
Tatikonda MV.1999. "An empirical study of platform and derivative product development projects". *Journal of Product Innovation Management*, **16**, 3-26.
Tatikonda MV and Rosenthal SR.2000a. "Technology novelty, project complexity, and product development project execution success: A deeper look at task uncertainty in product innovation". *IEEE Transactions on Engineering Management*, **47**(1), 74-87.
Tatikonda MV and Rosenthal SR.2000b. "Succesful execution of product development projects: Balancing firmness and flexibility in the innovation process". *Journal of Operation Management*, **18**, 401-425.
Tatikonda MV and Montoya-Weiss MM.2001. "Integrating operations and marketing perspectives of product innovation: The influence of organizational process factors and capabilities on development performance". *Management Science*, **47**(1), 151-172.
Thomke S and Fujimoto T.2000. "The effect of "Front-loading" problem solving on product development performance". *Journal of Product Innovation Management*, **17**, 128-142.
Trygg L.1993. "Concurrent engineering practices in selected Swedish companies: a movement or an activity of the few?". *Journal of Product Innovation Management*, **10**(5), 403-415.
Turner JR and Cochrane RA.1993. "Goals and methods matrix: coping with projects with ill defined goals and/or methods for achieving them". *International Journal of Project Management* **11** (2), 93-102.
Ulrich TK.1995. "The role of product architecture in the manufacturing firm". *Research Policy* **24**, 689-710.
Ulrich TK and Eppinger SD.1999. "*Product design and development*". Mc-Graw Hill/Irwine, International edition.
Wheelwright SC and Clark KB.1992. "*Revolutionizing product development: Quantum leaps in speed, efficiency and quality*". The Free Press, New York.
Williams TM.1997. "The need for new paradigms for complex projects". In Williams TM (Eds) "*Managing and Modelling Complex Projects*". Kluwer Academic Publishers.

Williams TM.1999a. “The need for new paradigms for complex projects”. *International Journal of Project Management* **17**(5), 269-273.
Williams TM.1999b. “Seeking optimum project duration extension”, *Journal of the Operational Research Society*, **50**, 460-467.
Wind J and Mahajan V.1997. “Issues and opportunities in new product development: an introduction to the special issue”. *Journal of Marketing Research* **34**(1), 1-12.
Zirger BJ and Hartley JL .1994. “A conceptual model of product development cycle time”. *Journal of Engineering and Technology Management*, **11**, 229-251.
Zirger BJ and Hartley JL.1996. “The effect of acceleration techniques on product development time”. *IEEE Transactions on Engineering Management*, **43**(2), 143-152.

Back to the Top