Economic Desirability and Traceability of Complex Products

Mordechai Ben-Menachem and Ilanit Gavious
Ben-Gurion University, Beer-Sheva, Israel
madaril@bgu.ac.il
quality@acm.org

Abstract: The real values and benefits of Information Technologies are difficult to quantify and frequently even to identify accurately. Existing financial models such as Net Present Value have proven insufficient for complex products, for long-term corporate goals. IS projects and software-rich products are decided upon while ignoring critical financial aspects, as the distance between the corporate product vision and the reality that engineers see may be very large. This paper maps between economics vis-à-vis IS-based product management via an inter-disciplinary approach, looking at the needs and exigencies of corporate management, IS project, products and software engineering.

The basis for the article is a discussion of the difficulties in evaluation of the economic desirability of complex, software-rich products. It presents a dynamic corporate-level model for economic profit evaluation designed to deal with the unique characteristics of such products, over many variants and versions, and the entire lifecycle. Given the extreme uncertainty of costs, benefits, risks and timeframes projections, the model facilitates real time reporting via an information system designed for management of Products, Portfolios and Projects. Whereas existing project management techniques such as Earned Value Management provide a general basis for managing project level activity, our model provides a longer-term view to assess economic affects of corporate strategies over time. This is provided by a dynamic, Management Information System based aggregation of all product information, over an entire product lifecycle, with the objective to provide a knowledge base for corporate dynamic decision-making. Concomitantly, the model fulfils Sarbanes-Oxley Act of 2002 requirements for management assertion traceability of valid and accurate measures.

These aspects co-joined, from Sarbanes-Oxley, back through multiple products, over myriad versions, and through automated requirements, design and testing tools, all combine to form an auditable management feedback loop that can be leveraged at multiple corporate management levels. The paper represents a significant step towards quality product decision-making via a model that is meaningful, while also useful as it is leveraged through an automated tool set.

Keywords: economic profit, information systems, IS management, IS evaluation, product management, project management, traceability

1. Introduction

In the modern economy management seeks enhanced measurement capabilities of economic benefits provided by complex, software-rich, products. This economic benefit is usually derived from costs' savings, revenue increase and/or enhanced resource-usage efficiency. Project managers are required to provide corporate management with a supporting environment for quality decision-making; while products compete for resources within an environment of limited resources.

Extant research examines issues of project efficiency maximization. For example, a recent study by Serich (2005) shows how the combination of Concurrent Engineering with Prototyping enhances project efficiency. However, what is still lacking is the multi-dimensional facet of the economic value of the *product* to the organization. Notably, economic value cannot be limited to the period in a product/system lifecycle devoted to initial development, but needs to be viewed in a longer context.

Corporate environments are characterised by multi-layered product portfolios; wherein each separate product is created, maintained, enhanced and/or expanded via the facility of a project. Basic data concerning these processes begin from the project level but for corporate decision-making, need to be aggregated to product, and frequently portfolio, level. This demands a dynamic decision-making instrument.

One of the most common financial tools used for investment decision making is Net Present Value (NPV). Literature referring to economic modelling for software-rich products is very sparse; whether in computing literature, finance literature or management literature (e.g., Turnbull, 2003; Greiner, 2003; Armour, 2005). Boehm's (1981) well-known book "Software Engineering Economics" deals mostly with software engineering aspects, and not the economic or management aspects. Tockey (2005) is the first book to address this area specifically. His orientation is for the software professional, and is based on an expectation that the usual software practitioner is unfamiliar with economic aspects. This gap between the two professions creates an economic reality in which Information Systems (IS) projects, the basis of every modern organization's economic existence, are being decided upon while ignoring many critical financial aspects.

Reference this paper as:
Ben-Menachem, M. and Gavious, I. "Economic Desirability and Traceability of Complex Products." *The Electronic Journal Information Systems Evaluation* Volume 11 Issue 3 2008, pp. 155 - 166, available online at www.ejise.com

This article presents a dynamic model for economic evaluation of the desirability of complex products. Throughout this article, the term 'complex products' refers to software-rich products, containing in their lifecycle, multiple versions (changes over time) and multiple variants (changes in use). The model is based on analysis of critical differences between complex and more 'traditional' products. These differences are found in costs and expected benefits from the product as well as associated risks and timeframes. Our model applies these differences as measurable information, collected from real objects and events, by an information system.

Existing project management techniques, such as Earned Value Management (EVM), provide a general basis for managing *project* level activity. A project management approach is sufficient in an environment of incremental development or a 'one up' project. Our model provides a longer-term view to assess economic affects of corporate strategies over time. This is provided by dynamic, Management Information System (MIS) based aggregation of all *product* information, over an entire product lifecycle. The objective is to provide a knowledge base for corporate dynamic decision-making. Concomitantly, our model supplies business goals traceability via specific procedures. This is consistent with the Sarbanes-Oxley Act of 2002 (Section 203) requirement for enterprises to implement traceability and change-management, from management assertions through components.

The paper is organized as follows. Section 2 discusses principles of economic evaluation of complex products, focusing on the "traps" in economic modelling. Section 3 presents a multi-dimensional analysis of costs and benefits, with incorporation of risks and timeframes for complex products. Thus providing basis for an economic profit model adapted for the unique characteristics of complex products. Section 4 presents our dynamic model for evaluation of complex products' economic desirability, designed for constant updating via a functioning Management Information System. This dynamism is shown to be crucial in such products because of the extreme uncertainty embedded in projections of costs, benefits, risks and timeframes. Section 5 contains summary and concluding remarks.

2. Economic evaluation of complex products' desirability

Managers generally base business decisions on evaluation of costs and expenses versus revenues, or other potential benefits. Management needs to perceive which resource investments may produce an optimal result, which should be economically expressible. One possible way to represent this conceived optimum might be via estimation of the expected return on a particular investment, as compared to alternative investments – though a specific project may be chosen even though economic models may not show it the optimal investment (for instance from considerations such as market competitiveness, customer services, quality or other not-readily-quantifiable corporate needs). This is made much more difficult in a software-rich project. Such projects should be measured using standards of performance similar to traditional, tangible asset-based projects. However, while they are expected to stand under the same magnifying glass of economic profit measures, they are not expected to produce the same "kinds" of profit as benefits may frequently be qualitative (e.g., customer satisfaction). Furthermore, looking at the entire lifecycle of a complex product, rather than a single project, complicates the decision-making process.

The use of a financial model to measure product desirability is valuable. A financial model supplies a clear framework for forecasting costs slated for an investment and evaluating its expected benefits. It enforces a well-designed implementation plan from all stakeholders, with viewpoint[1] commonality. A model helps organizations understand and concur on expected results. At the least, such a model is a common tool to show all stakeholders, from the "lowest level" to senior management that an investment has economic value.

However, a significant "trap" in economic modelling is the inherent inaccuracies embedded in forecasting future cash flows and cost of capital over investment lifecycle. This is made much more problematic in software-rich products as development times may be quite long. Moreover, economic value cannot be limited to the period in a lifecycle devoted to initial development, but also needs to account for on-going system evolution (ASD C3I 2002; Ben-Menachem & Marliss 1997; BPM Forum 2004). The temporal gap between the project's *initial* requirements analysis and the concomitant requirements as the system or product matures, then serves to "forcibly grow" the aforementioned inaccuracy. This is more than just requirements drift, which tends to infer changing of agreed upon requirements while in development, and includes new

[1] As defined by IEEE Standard 1471: [3.1.9] *view*: a representation of a whole system from the perspective of a related set of concerns. [3.1.10] *viewpoint*: a specification of the conventions for constructing and using a view. A viewpoint acts as a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis.

requirements resulting from the learning processes inherent to development. The temporal issue has additional effects, as while the product and technology evolve, other effects occur, such as changes in the business environment.

Another trap is that quantifiable methods for estimating intangible results are a major challenge; while frequently, these intangibles may be the most significant (for example, enhanced Information Retrieval ease/speed or an ability to make higher-quality business decisions). Despite these weaknesses, financial modelling is usable even in complex products, provided the model and the modelling techniques are adapted to the product. We make this adaptation in a multi-dimensional costs and benefits analysis, with the incorporation of risks and timeframes, for complex products.

3. Multi-dimensional costs and benefits analysis of complex products

Our model for economic evaluation of complex products' desirability is based upon the traditional NPV method of evaluating the desirability of investments. This section shows the basis for our model and critical points of difference between it and NPV. These differences are in the areas of costs and expected benefits from the product as well as associated risks and timeframes.

3.1 Costs analysis

A deep understanding of costs is critical for good product planning. The Information Technology industry is rife with reports concerning projects' failure rates (Clancy, 1998; Taylor, 2001). Enhanced understanding of costs involved in a project will enhance product managers' ability to effectively plan products, as well as enhance abilities to react to changes in real time. Studies that deal with this issue (e.g., Boehm, 1997) tend to address initial development exclusively, with little or no discussion of on-going costs.

Complex products' characteristic costs divide into those that are typically predictable and those that are known *ex post facto*. The former include cost of hardware (equipment of all kinds), communications networks (infrastructure and connection costs), software licenses, manpower (technical and non-technical personnel, organic to the organization or external "consultants" employed for temporary tasks), training (staff and users) and installation (including data conversion). Although predictable, these costs pose a high degree of uncertainty, temporal and otherwise. For example, whereas consultants may be hired for exactly the times needed, employees' availabilities are not pre-determinable by project needs (only) but also by the availability of the needed personnel; that is, the correct and needed personnel must be available, many of whom may be specialists.

The second type of costs includes aspects of a more complex structure. This complexity causes them frequently to display unpredictably quantifiable aspects. Software (and withal, software-rich products) differs from "traditional" product development projects in four broad areas, Commercial-Off-The-Shelf (COTS) tools and modules, quality assurance activities, development processes and system uncertainty.

COTS tools or modules: It is very rare to develop an entire bespoke system/product. Almost all, whether information or reactive systems, include some vendor-acquired modules and some bespoke. In addition, most work environments (development and evolution) include vendor-developed software tools (these may be deliverable with the project or non-deliverable, used only for development). Some applications may contain both types, e.g., reporting tools and acquired processing modules (the concept of *module* infers that it becomes an active part of application operation; while a *tool* is either passive or external). Packaged modules and tools predetermine a great deal of the design structure of the chosen solution and highly affect all development and evolution processes used to produce and maintain product suitability. A high quality, flexible set may reduce both development and evolution costs. Poor choices will cause many unforeseen and unforeseeable occurrences. Frequently, a great deal of expert and specialized knowledge is needed to utilize these purchased items effectively. This knowledge may also be acquired and not necessarily with the purchased product.

Components' purchase is common in every project – but there is a very significant difference. In traditional projects one purchases a component with the clear intention that it be fundamentally a 'Black-Box'. In software, this is almost never the case. Almost all purchased software components are 'Grey-Boxes' where functionality is acquired and then needs to be studied (and frequently modified) before it can be utilized, or utilized optimally. This difference is a very large cost component, above and beyond the purchase cost that must be accounted for and planned for.

Quality Assurance activities: Large portions of the resources of software-rich projects are used to correct errors produced by the project (typically, fifty percent of overall expenditures (Ben-Menachem & Marliss, 1997)). As part of, and in parallel with, the quality assurance process, every project contains many verification and validation processes, both to capture errors and to allow timely correction after discovery. Generally, software quality is not easily quantifiable (despite many attempts). Notwithstanding best efforts, every software system contains some unknown quantity of defects. Moreover, defects also occur in parts of a system that were previously satisfactory, as requirements and environments change and evolve. Many costs of non-quality can be computed, but only in retrospect.

Software development processes: Theoretically, discussions of software development process may not seem significantly different from many other process-oriented industries - for example, the highly published processes in the automotive industry (e.g., “common platforms use”). However, when designing an automobile, the concept is to design it once and then manufacture as many “copies” as possible (even if the copies are not identical, they are technologically non-dissimilar; differing usually in superficial aspects such as colour schemes). Software does not have manufacturing. It is always designed and produced ‘one-up’, then constantly changed and evolved afterwards. Automobiles are also not designed to be updated while in use, while software must be constantly updated to remain viable (Lehman & Ramil, 2001).

System uncertainty: A most prevalent attribute of software-rich projects is high uncertainty; degrees of uncertainty ensure unforeseen costs. Lehman (1990), in his seminal article concerning computer system uncertainty shows that: “*In the real world, the outcome of software system operation is inherently uncertain with the precise area of uncertainty also not knowable*”. Additionally, Ben-Menachem & Marliss (2004) showed that this uncertainty principle extends beyond programs to systems and systems-of-systems (e.g., ERP) and more than that, is not static and is a natural aspect of the ways that systems evolve. To whit, *“…all change increases system uncertainty, which grows exponentially as the configuration item view extends toward systems and systems-of-systems; anomalies propagate, creating unforeseen states, and inherent system uncertainty explodes*”.

Unforeseen costs refers to costs which, while the experienced project manager knows that such will occur and may even be able to “guess-timate” their approximate quantities, what these will be, when they will occur and exactly for what will the money actually be spent, is impossible to define ex-ante.

3.2 Benefits analysis

Potential benefits of complex products may be quite broad. Occasionally, the most significant benefits will be strategic. Strategic and tactical benefits should be differentiated, as the former frequently are not directly measurable while the latter may be straightforward (see Tockey, 2005). Strategic might be overall “change the company” concepts such as ERP implementation, while tactical might be enhancement of a corporate function (e.g., customer management).

Many systems surprise the producers, bringing benefits other than those originally defined, some of whom may actually prove of greater value than the original plan. Not all benefits are, or can be, tangibly defined. For example, if the new system enables more efficient report production, using less manpower and/or less time, than the direct benefit is clear and can be quantitatively measured. However, indirect benefits may be more difficult to measure - staff that previously produced this report may now be free to work on enhanced services. These are not as easily measured.

Additionally, benefits are not actualized at once. Projects may take significant time to implement (three to five years is not uncommon) while many other factors may affect the corporate environment during the period. Direct cause-effect analysis is impracticable, as it may be difficult to isolate a project’s benefits from other causes. However, this does not preclude indirect analysis and/or use of simulations and projections. Pre-project evaluation of benefits and costs should be based on projection of the (proposed) situation with and without the project, and not before and after the project. This is to isolate (proposed) changes to the organization caused by the project (see also, Tockey, 2005, pp. 305-306) and even more so, by the product that results from the project (or projects) creating and evolving it.

Whereas benefits analysis of an individual project is difficult and complex, it is significantly more so for an entire product lifecycle, at least as a product may typically be developed via many individual projects over its lifecycle.

3.3 Risks of complex products and their effect on the investment

Longstaff, Chittister, Pethia & Haimes,(2000, pp. 45) state: "We cannot simultaneously measure the risks associated with software and information assurance when no protective actions are taken and measure the efficacy of deploying risk assessment and management on the system because the system has fundamentally changed." This extends Heisenberg's Uncertainty Principle, to software, and has quite profound and far-reaching effects on economic modelling of software-rich systems ("…since the measuring device has been constructed by the observer, and we have to remember that what we observe, and what we observe is not nature itself but nature exposed to our method of question." "…when searching for harmony in life one must never forget that in the drama of existence we are ourselves both players and spectators.", Heisenberg, 1991). What seems the most obvious characteristics of software, to both practitioners and users, are the lack of wear-down and the ease of changing it. Both perceptions are incorrect and are a major source of risk (Broekman & Notenboom, 2003). The mis-perception results from viewing software via an inapplicable paradigm.

The past few centuries used an industrial paradigm for filtering worldview. The industrial paradigm assumes glacial change; so slow and gradual as to be barely recognizable except over relatively long periods of time; in hindsight. Today's reality differs. "*We live in a world where the only certainty is change. Changes occur so quickly that in most situations we must resign ourselves to them without first being able to assess what their risks or benefits might be*" (Asuaga, 2001). Changes have levels of risk associated with them (Broekman & Notenboom, 2003), and as they occur so quickly, how can they be economically modelled, effectively? If software always has threats, hazards and risks, what are the roles of the project, and project management, in identifying, preventing, mitigating and controlling them?

Basic comprehension of risks' centrality is critical to economic modelling (Armour, 2005). In order to comprehend the difficulty inherent to the task of risk management and its concomitant affects on economic modelling, one must understand that this is not limited to "the expected" – that is, to the risks inherent in original product development and deployment (see section 3.1 "System Uncertainty"). Brooks (1995) added to this the inherent instability of computer systems: *"Fixing a defect has a substantial (20 to 50 percent) chance of introducing another"*. From this is derived the concept that after a complexity threshold is exceeded, fixing one flaw tends to create new flaws.

Risks in complex products exist on at least two levels, product oriented risks and risks associated with the projects that create and maintain the product. These are separate and not a linked hierarchy of inter-related risks.

3.4 Timeframes

The timeframes in which a product's desirability is evaluated changes as a function of the type of project(s) utilised, the type of product/system that the project(s) are intended to develop, the product's defined (and undefined) lifecycle and the changeability of the external environment over the time that the product is in development. Many researchers have shown that software projects almost always take more time than was initially expected (Brooks, 1995; Clancy, 1998; Taylor, 2001). The project's original definition frequently changes over time, sometimes without management knowledge. In addition, Verhoef (2002) shows that project costs and risks (and withal, concomitant product risks) *"…dramatically increase when the development schedule is compressed…"*. Thus, despite the basic management desirability to shorten individual projects which compose a product development (and/or evolution), as suggested by Clancy (1998), the utility of this technique has a 'lower bound' when this increases risk and defects.

To summarize, an economic evaluation of investments – particularly for items of questionable tangibility – should examine issues both quantitatively and qualitatively. It is not always simple to acquire the data needed for a proper quantitative analysis; or at least, not a sufficient population of such data as to ensure an analysis of reasonable value. In such a case, qualitative data can prove relevant and significant. This is particularly poignant for complex products where fuzzy values are prevalent. Quantitative tools tend to be more appropriate to economic valuation of traditional projects based upon "hard" tangible assets while, qualitative (or a combination of quantitative and qualitative) tools tend to be more appropriately used for "soft" or intangible assets. As such, our model uses a combination of quantitative and qualitative tools.

Thus, this combination of quantitative, qualitative and risk together with opportunities creates a much more complex situation than that for which NPV was designed. The organization, after a complex product implementation (many complex products consist of a system of systems), can arguably be a quite different organization than it was before. This “degree of impact” on the enterprise implies that the optimal model will supply more than just a numeric measurement; but rather, it will show a strategic view of product value to the enterprise, including qualitative values, information states, etc.

4. The model

We present a model for economic evaluation of complex products' desirability based on estimation of their incremental contribution to corporate value. Standard financial models, such as NPV and ROI, are static, have difficulty mapping between quantitative and qualitative data, are not oriented towards a view of systems (or systems of systems) and not designed for constant updating via a functioning MIS. A model needed for complex products must be dynamic. The large extent of risks and system uncertainty compels the corporation to base decisions upon verifiable, event-related data, as supplied by both Financial and Inventory Management Information Systems (see section 4.1.1, below), mapped with strategic Business Goals and Product Plans. Notably, this complexity is layered on top of the Enterprise Projects Management system. This basis, upon multiple management systems, allows application of the model to products and to portfolios of products, at enterprise level.

Product planning generates a project or series of inter-related projects. In the framework of our model, at product initiation, an initial estimation of economic profit from the product is performed based on projections of costs, benefits, risks and schedules. All these projections are product-level; as a product management process and not a part of the development (technical) process. This will usually be a part of the approval process. Each year or period (whatever periodicity management chooses as optimal) a comparison needs to be performed between projected and realized schedules, costs and benefits. Projections for the future are then updated via new information from this period and new profit computations are executed. An intelligent application of future projections adjustment demands cooperation between two parties; the firm’s financial officers and software engineers. This collaboration is important for a thorough analysis from both perspectives, finance and engineering, needed to complement each other.

During the evolutionary process of complex products there will be many versions and releases (Lehman & Ramil, 2001). The computation as per our model, is intended to be realized from “any point in time” until “any point in time” – inclusive of any chosen timeframe.

The model is a corporate-level management instrument, for viewing a product (or portfolio). One of the commonly used *project* management tools is Earned Value Management (EVM) (Boehm & Li, 2003; Ernst, 2006; DI-MGMT-81466, 2004; Lipke, 2002). EVM views tasks’ schedule, resources, requirements and costs. Each and all together, are addressed as managerial aspects of the task that can be itemised. It is a bottom-up project view and does not address overall product separating the project that creates the product or a new version of it from an overall corporate view.

We now present the model; depicted by two flow graphs (Figure 1 and Figure 3). Figure 1 presents the basic economic profit computation, with its information sources and processes. Figure 3 presents the iteration process of updating profit computations, with relationship to business goals and overall product planning. The graphical language used is standard notation, from Data Flow Diagramming techniques.[2]

4.1 The basic economic profit computation

Figure 1 presents a flow graph, divided into three sections. The left section deals with costs’ acquisitions (actual and projected), the middle section deals with acquisition and analysis of benefits (realized and projected) and the right section deals with the economic profit computation, albeit with the updated and dynamic inputs allowing needed model sophistication.

4.1.1 Costs’ acquisitions

Costs acquisition divides into two processes; quantitative cost elements and conversion of the costs into monetary terms.

The first process recognizes quantitative cost elements and has two data sources. The first source is an on-line inventory system for software assets that shows enterprise software artefacts with their history and

[2] A “bubble” is a process (always begins with a verb); a rectangle is a “source or sink” entity; a parallel with arced ends represents a data stores and arrows represent flow (with heavy arrows depicting main flow path).

complexity, and the accumulated costs incurred (Ben-Menachem & Marliss, 2004). The second is a Products Management MIS (see Figure 2). This system back-ends to MS-Projects for task tracking, providing data relating to business goals, lifecycle events, present and projected resources, and business and product risks. Note the absence of time as a controlling element. Time is provided for by any standard project management tool (e.g., Primavera or MS-Projects). The Inventory system stores values concerning realized costs, while the Products MIS shows projected costs.

The second process converts all costs (such as data stored in terms of ‘hours’, or other non-financial terminology) into monetary terms for use in the profit computation. It is aided by a financial reporting system (Ben-Menachem & Gavious, 2007).

4.1.2 Benefits’ acquisition and analysis

Costs are actual or projected values that are fairly straightforward to collect and simple to understand. Benefits are much more complex, whether actualized already or projected (see section 3.2). The first process is to capture system benefits. The main source for understanding benefits the definers of the system intended is from the list of system goals. However, this will cover only those intended benefits defined as primary goals of the system. Many systems have additional benefits that are indirect results of their development. In many cases, these may even be difficult to discern. We recommend that the system be examined, in a recursive relationship, to discover additional, indirect benefits.

These benefits are then critically examined and quantitative benefits are separated from qualitative. Quantitative benefits are converted into monetary terms, via information from the financial reporting system and supported by Expert opinion; while in parallel, the qualitative benefits are converted to metric values via the Goals-Question-Metric Paradigm (Mashiko & Basili, 1997) or some equivalent tool. This analysis provides additional value to the model as a management support for tracing system benefits’ development and for auditing of systems’ value to the organization.

4.1.3 Computation of economic profit

The Economic Profit computation is executed based upon updated projected cash flows of costs and benefits. Realized costs and benefits are relevant for the adjustment of projected costs and benefits, however irrelevant for decision-making; i.e., they are no longer relevant for evaluating the viability of continuing with the project. To account for risk and the time value of money, a risk-adjusted discount rate needs to be intelligently estimated, with the required differences per industry or business environment, by the firm's financial officers in collaboration with software engineers. This collaboration is needed for a risk analysis from both perspectives of finance and engineering, resulting in a best estimation of a risk premium to be accounted for in the cash flows' discount rate (Tockey, 2005).

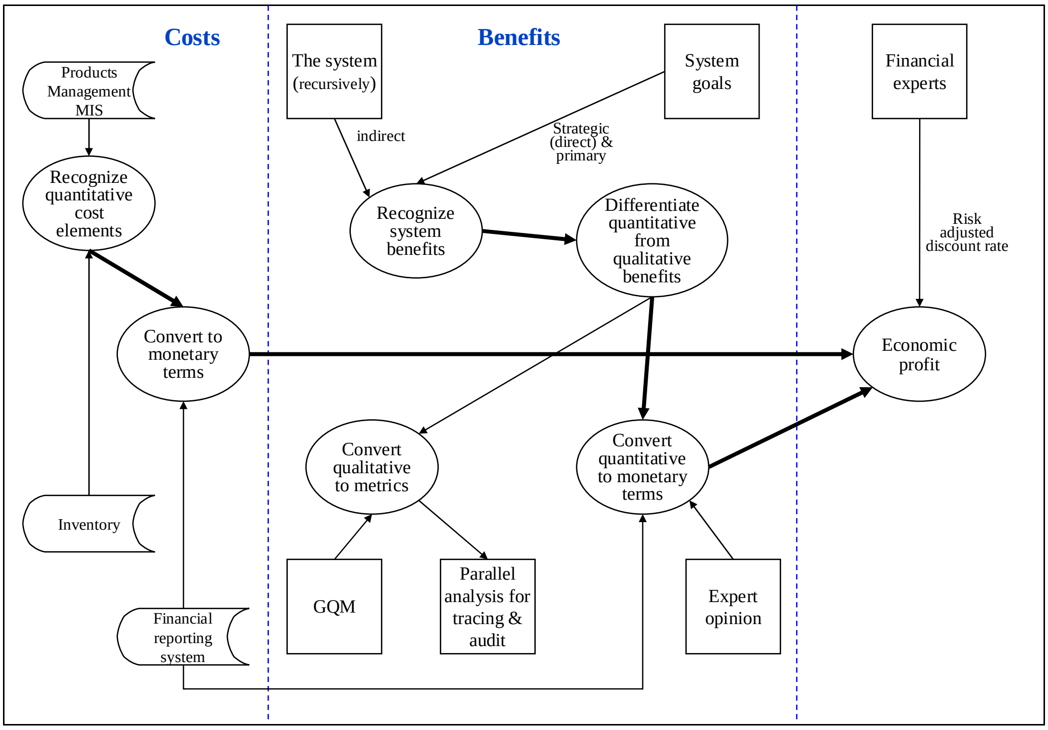

Figure 1: The basic economic profit computation

The processes described above need to be based upon sophisticated information processing tools. Until recently, tools to provide corporate management with realistic, real-time data that maps both existing systems and projects in progress were lacking. Recent development of a discipline called the Paradigm of Change (henceforth 'PofC') (Ben-Menachem & Marliss, 2004) provides an integrated toolset designed for data collection from projects, information management concerning projects (of all types, i.e., projects that create new systems as well as those that update existing systems) and knowledge management for decision support concerning complex products. Figure 2 shows the Products Management MIS focus of the PofC (one of several possible foci that PofC provides).

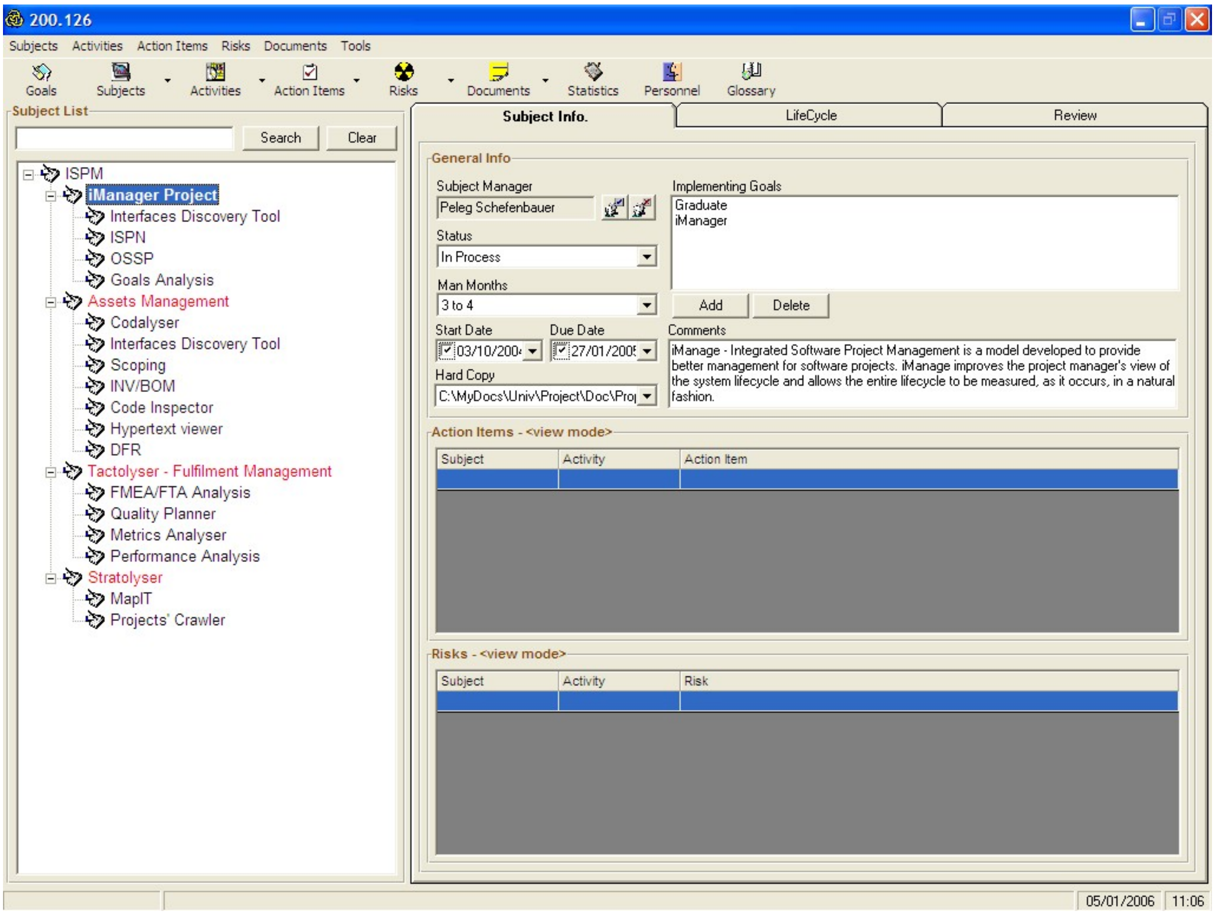

Figure 2: Screen shot of products management MIS

Figure 1 includes a series of data-models (Products Management MIS and Inventory) drawn directly from PofC and from financial reporting system which has information derived from PofC. In addition to these, the Analysis, GQM and System functions base their operations on PofC-derived knowledge.

4.2 The iteration process

Figure 3 presents a superset of the economic profit computations presented in Figure 1. It shows the evolutionary profit computations process, automatically updated with aid of the information system; hence making it a useful enterprise management tool.

The Product Plan and the Business Goals from which the product plan is derived, are general inputs to this process. The product plan has a defined scope and objective. Scope is critical to comprehend economic profit computation, vis-à-vis the goals defined by the business. Project plans are created by engineering staff (engineering management) in terms of performance, time (milestones), risks management and budget. The 'Projected Costs' and 'Projected Benefits' in Figure 3 are projected at product initiation, but in realistic practice today, seldom revisited categorically for continual feedback as to product viability.

The basic, 'initial' economic profit (I) computation is performed at 'initiating project' (the project that initiates the product) start, usually as part of the process of product approval. Each period (as chosen by management) a new computation is performed based on adjusted Cash Flow (CF) projections (economic profit (Sn)). As shown, the process of performing these computations is now much simplified by the fact that the input data for the computations are all based upon the PofC Management Information System. The succeeding computations are performed 'iteratively' per period. Note the two feedback loops in Figure 3. The inner loop (dashed line) is internal to the model process. It shows that the computed profit is based upon constantly updated data/information from the MIS and that each new computation is, itself, also stored in the MIS for future use. Thus, the accumulation of computed profits becomes a knowledge base with which upper management can deal for constant updating of business goals processing.

Notably, this becomes a positive feedback loop (as opposed to passive feedback) because it represents Past-Present-Future value of model use. Accounting systems, for instance, show a picture of the present based upon accumulated data from the past. However, such normative management information is augmented by showing the combined processes of profit evaluations at every action-point during product evolution. Additionally, these action-points can be compared one with others, across projects and across products, so that decision-points and action-points can be aggressively compared for enhancing the processes of making improved business decisions. This leads to Future, where the Past and the Present combine positively to enhance both the business model itself and the processes of improving the modelling technique.

Finally, as shown in Figure 3, the model contains an element called 'defects'. Every technology-rich project creates defects while in production. This is the state-of-the-art and the inherent nature of all technology-rich projects. Their quantity can be limited by very careful development processes, and their effects can be minimized if technology management so desires. These are choices with associated short-term costs; however it is generally accepted that the long-term corporate benefits justify them. Product defects may interfere with an ability to satisfy critical requirements or key functionality and always negatively affect customer satisfaction. Defects also negatively affect schedules and future product cash flows. All this negativity must be reflected in projections.

Some project management methods can deal with estimations of defects and resources consumed, such as parametric estimating, backfiring or several function point estimation techniques. This relates to specifics of software coding and is outside the scope of this article.

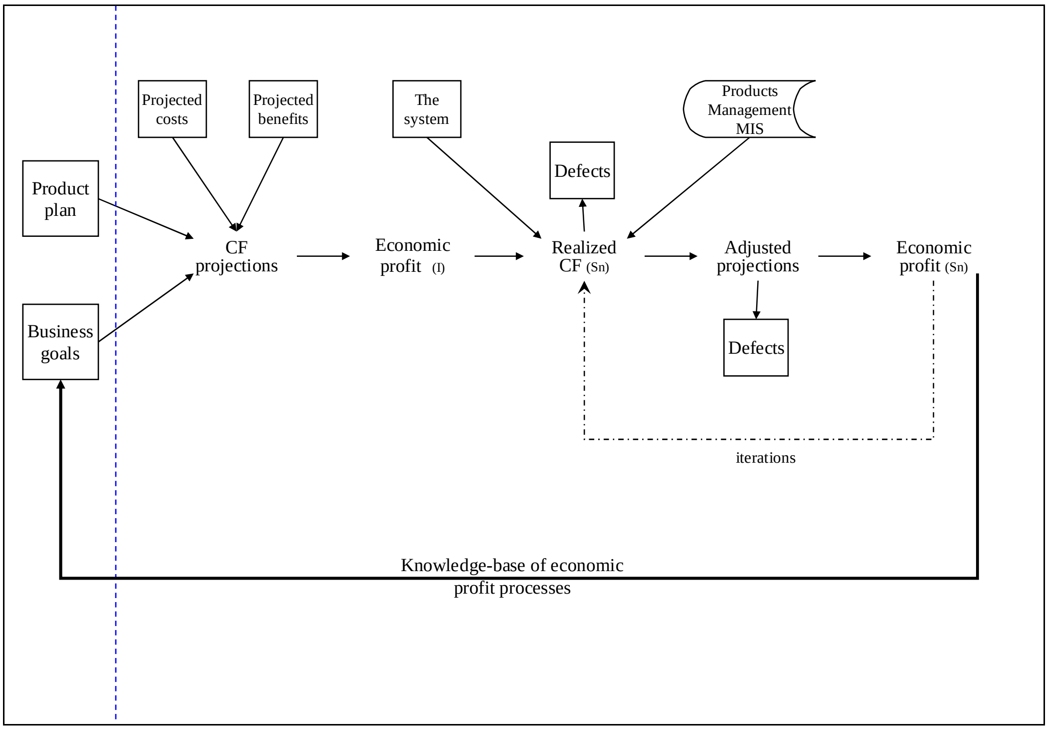

Figure 3: The iteration process

The suggested model helps learn from history ("mistakes") in estimating product economic desirability, where costs are typically underestimated and benefits are typically overestimated (Tockey, 2005). Experience improves firm's costs and benefits analysis and hence its investment decisions. Beyond management's need to perceive which resource investments may produce an optimal result is now added a regulatory requirement for lifecycle traceability. Sarbanes-Oxley Act of 2002 (Section 203) requires that IT be made transparent to the business and fit within mainstream business management. *"Specific assertions … have a general relationship to assertions about internal controls for both manual and information processing systems"*. *"An assertion can either be explicit or implicit"* and based upon *"specific procedures that can be tested and used to evaluate the assertions made by management."* Furthermore, *"Assertions made regarding disclosure/internal controls can be evaluated by testing stated or measurable criteria."* The criteria are based upon *"specific assurance objectives"* such as business goals that are traced via change management, mapped from management assertions to products and product components.

To accomplish these regulatory requirements, IT and business goals need to align and this alignment must be traceable, throughout the entire product lifecycle – that is, from the first requirements statement until the product is finally retired from all use. Use of automated requirements gathering and analysis tools, together with design and testing traceability, allows auditability of this alignment, both immediately and long-term. Economic evaluation, performed periodically via auditable information, from a traceable resource (MIS) and recorded for continual re-evaluation, endows corporate management with verifiable tools to affirm this regulatory requirement. Figure 1 shows this feedback and how this 'loop' is utilised for weaving the regulatory requirement into the corporate fabric – requirements, audits, economic evaluation, new product planning, etc.

5. Conclusion

Products and product portfolios should be economically valued, using economic terminology and standard financial computations. Economic models, common to the financial-world, apply to information systems (software-rich products); for they can be made to apply with adjustments. The application of economic models inherently enhances a corporation's long-term viewing capabilities. Software-rich products need to be managed with a view extending significantly farther than competitors conceived. Notably, many analysts have claimed that the primary driving force to market power is a long-term view.

Organizations need a long-term knowledge base, to attain a long-term view. The on-going management process, with all its decision-points, increases the necessity for long-term product information. Moreover, today's regulatory environment (e.g., Sarbanes-Oxley Act of 2002) necessitates the availability of such information.

Withal, how can this long-term view be consistently achieved, even in corporate environments that have not been previously built upon such concepts? Or, what can a corporation do, when it perceives that this kind of product view is not yet its corporate strength, and wishes to enhance capabilities in this area?

This article discusses these issues and provides a dynamic model for economic evaluation of complex products' desirability, via an inter-disciplinary approach, by mapping products, with the needs and exigencies of corporate management, product managers and software engineers. The model facilitates real time reporting via an information system designed for management of Products, Portfolios and Projects. This dynamism is shown to be management crucial because of the extreme uncertainty embedded in projections of costs, benefits, risks and timeframes associated with software-rich, complex products. As an additional significant benefit, the model fulfils Sarbanes-Oxley Act of 2002 requirements for management assertion traceability of valid and accurate measures, with auditability.

The model is designed to be based upon quality information, from automated data-gathering utilities, that is, the information is both of high quality and highly auditable, allowing corporate management to leverage this information to maximum benefit. What is no less significant is that this information can be used consistently, over related products, and to make long-term comparisons between product lines, while taking all their aspects properly into account.

These aspects co-joined, from Sarbanes-Oxley, back through multiple products, over myriad versions, and through automated requirements, design and testing tools, all combine to form an auditable management feedback loop that can be leveraged at multiple corporate management levels. The paper represents a significant step towards quality product decision-making via a model that is meaningful, while also useful as it is leveraged through an automated tool set.

References

Armour, P. G. (2005) "Project Portfolios: Organizational Management of Risk", *Communications of the ACM,* Vol 48, pp. 17-20.

Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (ASD C3I). (2002). Repairing Latent Year 2000 Defects Caused by Date Windowing. Investment and Acquisition Directorate, US Department of Defense, Version 1.0, April.

Asuaga, A. (2001) "Engineering a New Society", *IEEE Computing*, June, pp. 102-104.

Ben-Menachem M., & Gavious, I. (2007) "Accounting Software Assets: A Valuation Model for Software", *Journal of Information Systems of the American Accounting Association*, Vol 21, No.2, pp. 117-132.

Ben-Menachem, M., & Marliss, G. (1997) *Software Quality: Producing Practical, Consistent Software*, Int'l Thomson Computer Press.

Ben-Menachem, M., & Marliss, G. (2004) "Inventorying Information Technology Systems: Supporting the "Paradigm of Change"", *IEEE Software*, Vol 21, pp. 34-43.
Boehm, B. (1981*) Software Engineering Economics*, Englewood Cliffs, NJ: Prentice Hall PTR.
Boehm, B. (1997) COCOMO II Model Definition Manual. University of Southern California, Version 1.4.
Boehm, B. & Li, G. H. (2003) "Value-Based Software Engineering: A Case Study", *Computer*, Vol 36, No.3, pp. 33-41.
BPM Forum. (2004) Software Drain or Business Gain: Assessing Application Value, Relevance and Cost to Your Company. http://www.bpmforum.org/PDF/Software_Drain.pdf; December.
Broekman, B., & Notenboom, E. (2003) *Testing Embedded Software*, Addison Wesley.
Brooks, F. (1995) *The Mythical Man-Month*, Addison Wesley.
Clancy, T. (1998) *CHAOS*, Standish Group.
DI-MGMT-81466. (2004) Contract Performance Report. Department of Defense, December 2004.
Ernst, K. D. (2006) Earned Value Implementation Guide. Department of Defense, October 2006.
Greiner, L. (2003) "Analytics Unlock the Secrets to ROI", *Computing Canada*, Vol 29, pp. 16-17.
Heisenberg, W. (1991) “The Copenhagen Interpretation of Quantum Theory”; as reproduced in *The Treasury of Physics, Astronomy and Mathematics*, ed. Timothy Ferris, Little, Brown and Company, pp. 86-96.
Lehman, M. M. (1990) "Uncertainty in Computer Applications", *Communications of the ACM*, Technical Correspondence, V33N5, pp. 584-586.
Lehman, M. M., & Ramil, J. F. (2001) "Rules and Tools for Software Evolution Planning and Management", *Annals of Software Engineering*, Vol 11.
Lipke, W. H. (2002) EVM and Software Project Management - Our Story. Cross Talk.
Longstaff, T. A., Chittister, C., Pethia, R., & Haimes, Y. Y. (2000) "Are we forgetting the risks of information technology?", *Computer,* December, pp. 43-51.
Mashiko, Y., & Basili, V. (1997) "Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement", *Journal of Systems Software*, Vol 36, pp. 17-32.
Sarbanes-Oxley Act of 2002. Available at:
http://fl1.findlaw.com/news.findlaw.com/hdocs/docs/gwbush/sarbanesoxley072302.pdf
Serich, S. (2005) "Prototype Stopping Rules in Software Development Projects", *IEEE Transactions on Engineering Management*, Vol 52, No.4, pp. 478-485.
Taylor, A. (2001) IT Projects Sink or Swim. British Computer Society Review, September.
Tockey, S. (2005) *Return on Software*, Addison Wesley.
Turnbull, I. (2003) "Finding the Return on HR Technology Investments", *Canadian HR Reporter*, Vol 16, pp. 17-18.
Verhoef, C. (2002) "Quantitative IT Portfolio Management", *Science of Computer Programming*, Vol 45, pp. 1-96.