An Adaptive Process Model to Support Product Development Project Management

Viktor Lévárdy and Tyson R. Browning

***Abstract*—Projects are temporary allocations of resources commissioned to achieve a desired result. Since each project is unique, the landscape between the current state (the start of the project) and the desired state (the successful end of the project) is often dynamic, uncertain, and ambiguous. Conventional project plans define a set of related activities (a work breakdown structure and activity network) with the assumptions that this set is necessary and sufficient to reach the project's desired result. Popular models for project planning (scheduling, budgeting, etc.) and control are also based on a set of project activities that are specified and scheduled *a priori*. However, these assumptions often do not hold, because, as an attempt to do something novel, the actual path to a project's desired result is often revealed only by the additional light provided once the work is underway. In this paper, we model a product development process as a complex adaptive system. Rather than prespecifying which activities will be done and when, we set up: 1) a superset of general classes of activities, each with modes that vary in terms of inputs, duration, cost, and expected benefits; and 2) simple rules for activity mode combination. Thus, instead of rigidly dictating a specific project schedule *a priori*, we provide a "primordial soup" of activities and simple rules through which the activities can self-organize. Instead of attempting to prescribe an optimal process, we simulate thousands of adaptive cases and let the highest-value process emerge. Analyzing these cases leads to insights regarding the most likely paths (processes) across the project landscape, the patterns of iteration along the paths, and the paths' costs, durations, risks, and values. The model also provides a decision support capability for managers. For researchers, this way of viewing projects and the modeling framework provide a new basis for future studies of agile and adaptive processes.**

***Index Terms*—Adaptive processes, agile project management, process modeling, product development, project management.**

I. Introduction

A PROJECT is "a temporary endeavor undertaken to create a unique product, service, or result" [86]. Thus, it represents an attempt to do something that has never been done before, at least not in a particular set of circumstances. A product development (PD) project consists of a myriad of multifunctional activities, all (hopefully) working together to produce the information that will reduce the risk of the outcome being something other than what the project's stakeholders desire [12]. Since doing something complex, novel, and expensive is challenging and risky, a plethora of models has emerged to improve our understanding of PD projects, processes, and the management thereof.[1]

However, most traditional project models, plans, and tools make a critical assumption about the goal of a PD project that is quite limiting in practice, even when the goal is known and fixed. These models assume that *the path to reach the goal* (i.e., a predetermined set of activities and dependencies) is known and will be sufficiently efficient and effective. Yet, this is rarely the reality in PD: the planned set of activities may be both insufficient and partly unnecessary. When the path to the destination is unclear, it is no wonder that so many PD projects are "challenged" or fail. For example, in a survey of over 250 000 small software PD projects, the Standish Group [108] reports that only 28% succeeded, while 23% failed and 49% were "challenged," meaning they were either late, over budget, or had fewer features or functions than originally specified.

The unclear path to a project's goal has been discussed in the literature in terms of process uncertainty and ambiguity. Both stem from a lack of knowledge about a problem at the time of making a decision affecting its solution. Researchers have essentially categorized uncertainty as "knowing what you don't know" and ambiguity as "not knowing what you don't know" [97], [99] and furthermore into a spectrum from variation, to foreseen uncertainty, to unforeseen uncertainty (ambiguity), to chaos [30], [68], [84]. Since greater amounts of ambiguity characterize PD projects, the traditional methods and tools of project management provide less value. Moreover, project managers' and participants' attitudes and aptitudes must change [31], [68], and presumably also their models, methods, and tools. While ambiguity is clearly a source of risk for PD projects, it can also bring opportunities to organizations capable of effectively sensing the endogenous and exogenous changes and responding to them efficiently by adapting to the changed conditions [43]. Indeed, PD projects capable of coevolving with their environments and dynamic stakeholder needs can profit from the accelerating pace of change in market needs [34], [37].

In response to the realities of process and goal uncertainty, ambiguity, and instability in projects, some advocate nontraditional approaches to project management such as *extreme* [4], *adaptive* [48], *flexible* [69], *response-able* [35], *lean* [88], *agile* [47], etc. These have gained particular traction in software PD [89], although they may not apply as readily to complex

Manuscript received March 28, 2008; revised July 7, 2008. Current version published October 21, 2009. This work was supported in part by the European Community for the SysTest project "G1RD-CT-2002-00683" at the 5th Framework Programme (GROWTH) and by the Texas Christian University Research and Creative Activities Fund under Grant 60306. Review of this manuscript was arranged by Department Editor C. Terwiesch.

V. Lévárdy is with the 3D Systems Engineering GmbH, München 80335, Germany (e-mail: v.levardy@3dse.de).

T. R. Browning is with the Neeley School of Business, Texas Christian University, Fort Worth, TX 76129 USA (e-mail: t.browning@tcu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2009.2033144

[1]See [18] for a review of PD process modeling paper. See [74], [86] for overviews of project management knowledge. See [23], [117] for overviews of PD issues and methods. See [68] for an overview of managing projects under uncertainty.

hardware PD and large, safety-critical software PD [6]. Nevertheless, it seems clear enough that the assumptions underpinning the conventional models and tools for PD project planning and control do not always hold [59], [123], and that these models and tools could extend their capabilities by accommodating more dynamism and flexibility.

In this paper, we propose a new modeling framework that views the PD process as a complex adaptive system. Rather than prespecifying which activities will be done and when, we set up: 1) general classes of activities with multiple modes that vary in terms of inputs, duration, cost, and expected benefits; and 2) simple rules for activity interaction and combination. That is, instead of dictating a specific project schedule *a priori*, we provide a "primordial soup" of activities and rules through which the activities can self-organize and adapt to the changing state of a project. This *adaptive PD process* (APDP) modeling framework treats project control as a decision-making process, where each decision aims to maximize the expected value of the overall project in light of its current state and environment. In contrast to conventional approaches that separate project planning and control, we consider planning and control more holistically, combining the recent ideas of search and selection (e.g., [68]) and exploration and exploitation (e.g., [5]) with the traditional ones of coordination and transformation. By assuming from the outset—during the initial project planning stage—that the process will adapt, it is possible to improve project planning by understanding the "design space" of potential paths to success (rather than focusing only on a single one), orient the workforce toward self-organization, and form the capability to cope with many unanticipated situations—in effect, to be more agile and flexible. Furthermore, instead of merely exploring artificial landscapes for purely theoretical insights, the APDP model provides a decision support tool for actual projects.

This paper's primary contribution is *a new way to consider and model a PD project's process*. We motivate and support the new approach and model with theory, describe its formulation, and begin to validate it with a real application. We devote roughly equal space to each of these areas, organizing the paper as follows. Section II discusses the theoretical background motivating our approach, after which Section III introduces the APDP modeling framework. Section IV applies the model in an industry setting. Section V concludes the paper.

II. Theoretical Background and Research Motivations

The APDP model is motivated by several theoretical bases, including PD process modeling, product design cycles or iterations, measuring product design performance and project value, and project adaptation. In this section, we provide an overview of the theory and distill pertinent motivations from each area.

A. Product Development Process Modeling

To enable the division of labor, the innovative, problem-solving efforts in a PD project are decomposed into smaller, interacting activities [119]. The interactions have also been referred to as dependencies, precedence relationships, interfaces, work products, inputs, outputs, information flow, or interim

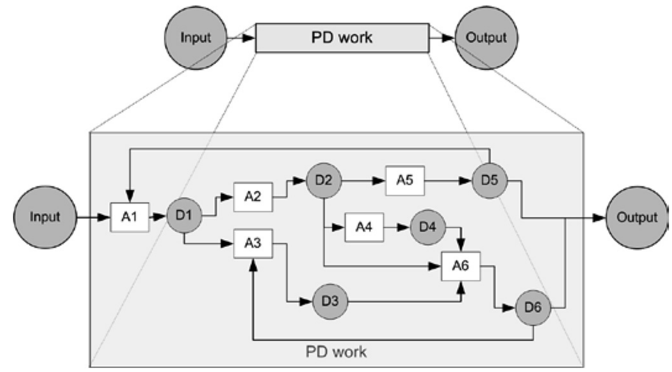


Fig. 1. Decomposing PD work into activities and deliverables to yield a process architecture.

deliverables [18]. We adopt the last term. Product development processes have been modeled with a number of frameworks, including activity networks, Petri nets, etc.; see [16] for a review. We refer to the network of activities and deliverables as the *process architecture* (see Fig. 1).[2] Alternative process architectures—i.e., alternative sets of activities, deliverables, and/or sequences of them—can vary in their performance [13], often due to the number of assumptions required in certain activity sequences[3] and the consequential iteration and rework when those assumptions prove inadequate.

Product development projects, like projects in general, seek to do something new, once, rather than repeat a known process exactly [16]. A distinct challenge in modeling such a situation is anticipating which activities and deliverables will actually be needed. Process modeling can facilitate the description and exploration of the *process space* [22] (the set of feasible process architectures) and foster the definition, sizing, and planning of the activities and deliverables. It can provide a useful description of a PD system's behavior that improves understanding, which is a prerequisite to effective project planning and improvement [121]. In relatively well-understood projects, Austin *et al.* [2] found that over 90% of design activities and deliverables could be anticipated *a priori*. Even in the relatively ambiguous conceptual design stage (the "fuzzy front end") of PD, Austin *et al.* [3] showed the efficacy of efforts to identify activities and deliverables. Thus, even when projects are less well understood, modeling can structure the act of sorting out what is known and unknown [16] and prompt the discovery of the "unknown unknowns" ("unk unks" for short in common industry parlance). It often turns out that many of the "unk unks" that perturb projects were actually known to *someone*, but they nevertheless surprise project management because there was no way to get this knowledge into the project plans. Building a

[2]In terms of a physical product, the IEEE has defined architecture as "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [51]. The definition of a *process architecture* is analogous, where the system is a process and the components are its constituent actions.

[3]When a deliverable is information only, an assumption can often act as its surrogate. Many PD activities, therefore, actually have a variety of options for when they will occur, depending on the assumptions developers are willing to make.

process model can help expose such latent information. Thus, the best purpose of PD process modeling would not seem to be to develop an infallible project plan, but rather to prompt a fuller exploration of the project landscape and the potential paths between the current state and the desired state(s)—to better anticipate the roadblocks. (Former U.S. President and General Eisenhower famously stated: "The plan is nothing. Planning is everything.") In contrast, in traditional project management, process modeling has often focused on a single path (e.g., the "critical path").

Motivation 1: A key purpose of PD process modeling can be to help a project manager understand the feasible "design space" for his or her project, a set of process architectures called the process space.

B. Iteration

Fundamentally, PD is a nonlinear, iterative process [57], [95]. Any process of solving the complex problems interdicting the production of a unique result will be fraught with some amount of trial, discovery, and redirection. Many common models acknowledge this cyclical process: e.g., plan-do-check-act (PDCA, the Shewhart–Deming cycle) [32], define-measure-analyze-improve-control (DMAIC, the Six Sigma cycle) (e.g., [44]), observe-orient-decide-act (Boyd's OODA loop) [8], design-build-test (e.g., [23]), and the experimentation cycle [114]. Furthermore, in PD, many impending design failures cannot be discovered immediately. This increased "rework discovery time" and the consequential elongation of the feedback loops in the process have a significant effect on project duration and cost [26], [41]. While some design iterations are deliberate (e.g., spiral development [7]), rework cycles usually are not. Researchers have proposed various classification schemes for iteration, including planned and unplanned iterations.[4] We refer to all cycles in the PD process as iterations.

In terms of the deliverables that relate activities, we compile the following causes of iteration.

1) *Poor activity sequencing*: creating deliverables at the wrong time (often too late), which forces other activities to wait or make assumptions [67].
2) *Missing activities*: not creating all of the needed deliverables.
3) *Poor communication*: not transmitting a deliverable clearly, promptly, or appropriately.
4) *Input changes*: undermining the deliverables (or proxy assumptions) used by activities to do their work and, in turn, create further deliverables (e.g., requirements changes.
5) *Mistakes*: inadvertently creating defective deliverables.

In general, we note that

Motivation 2: Iteration occurs when the cumulative output deliverables of prior activities, plus the assumptions that can be reasonably made at the time, are insufficient to enable subsequent activities to add appropriate value to the project.

Researchers have proposed a variety of models to explore iteration in PD projects. System dynamics models (e.g., [41]) have looked at the overall "work to be done," some portion of which will have to be redone. However, this aggregation can prevent the identification and management of the specific activities and deliverables with the greatest leverage in a process. Hence, most PD process models involve networks of distinct activities. Yet, conventional network models such as program evaluation and review technique (PERT) and critical path method (CPM) assume a stable, acyclical network. Some network modeling frameworks, such as the graphical evaluation and review technique (GERT) (e.g., [76]) and the *design structure matrix* (DSM) [11], account for iteration explicitly.

The DSM has been instrumental in a number of PD process models, since one of its key strengths is highlighting iterative loops. A DSM is a square matrix representation of a directed graph (digraph), with the nodes (activities) represented by the cells along the diagonal (e.g., Fig. 7). A mark in an off-diagonal cell represents one or more deliverables flowing from one activity to another. One reads across a row of the matrix to see where the activity in that row sends its outputs, and one reads down a column to see from where the activity in that column receives its inputs.[5] For example, in Fig. 7, activity 1 may receive one or more inputs from activity 7 and provide one or more outputs to each of activities 3, 4, and 8. One can analyze a binary DSM with the simple objective of minimizing feedback loops (i.e., seeking an order of the activities to upper-triangularize the matrix), although a variety of objective functions can be used [73]. More realistic, numerical DSMs become more challenging, especially since the optimal process architecture may not be the one with minimum feedback [13]. Recent DSM-based models employ discrete event simulation and account for a number of important PD process characteristics, including activity learning curves and the risks of second- and higher-order iterative loops [13], [20]. However, all of these network models assume that: 1) one can estimate all PD activities, dependencies, and iteration probabilities *a priori*; and 2) iteration implies the repetition of the same, previously completed activities.

In addition to the *presence* of iteration, its *implications* are also challenging to model. An entire activity or set of activities may not have to be redone, and a repeated activity may go faster due to nonrecurring set-up time and/or learning. All of this may vary based on the exact nature of the problem and the means to its correction. For example, Thomke and Bell [113] showed how experimentation strategies contribute to design iteration efficiency and effectiveness and found that both tended to increase over successive cycles. Also, the cost of design iteration could be relatively low for virtual prototypes and extremely high for full-fidelity physical prototypes, while physical prototypes often had to be built again for retesting (e.g., automotive crash tests), computer models required only slight modification for a rerun of the simulation. Most PD activities will produce results that increase knowledge of the product recipe, and doing any activity will expend resources, so the project state after any activity will differ from the prior state. This new state may imply different

[4] See further categorizations in [118] and [25, Ch. 3].

[5] Some DSM paper uses the opposite convention, showing feedback above the diagonal, which is the transpose of the matrix. The two conventions convey equivalent information.

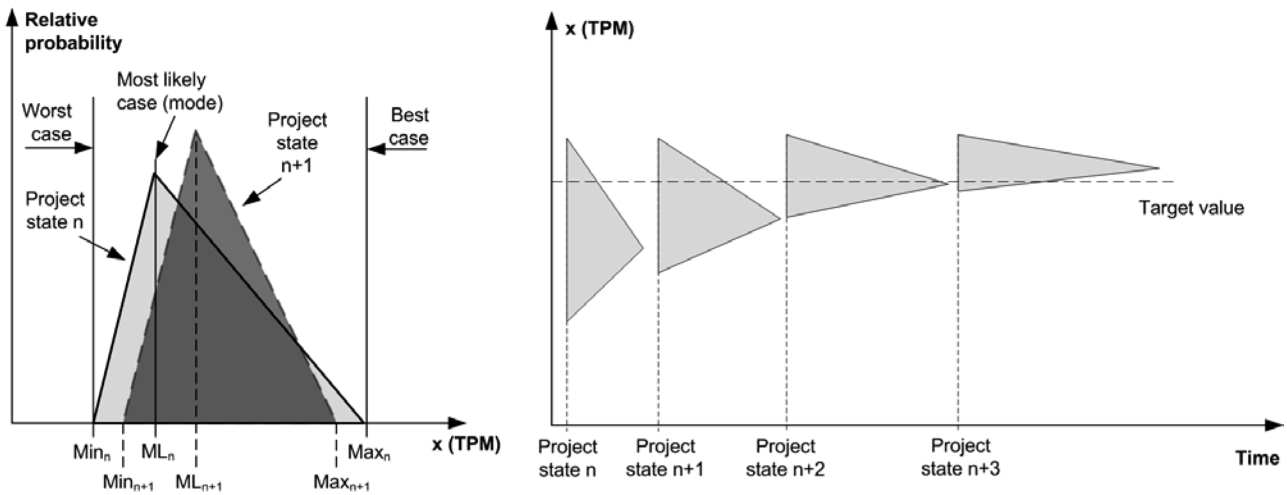


Fig. 2. Examples of the evolution of two TPMs.

project needs and actions rather than the exact repetition of the activities deemed appropriate for prior needs.

Thus, in a real project, a particular iteration may not be a foregone conclusion. The managerial decision about whether or not to iterate depends on a comparison of the actual and desired states of the project, in terms of time, cost, technical performance, and risk. Most PD process models focus on time and cost variables and resort to assumptions about the probabilities and impacts of iterations. Because experts could tell which activities had been the main triggers for iteration in prior projects (e.g., design reviews and tests), and because sometimes they could estimate statistical probabilities for rework, probabilistic iteration modeling became a popular technique (used by all of the aforementioned models). Such probabilities have to be inferred *a priori* based on typical situations, but they do not address the entire range of potential managerial options.

Motivation 3: Iteration is a managerial control option and it will be exercised when it provides the path of greatest added value to a project.

C. A Product Design's Technical Performance and a PD Project's Value

The desired outcome of a PD project is a "recipe" for a successful new product [91]—i.e., a product design including a list of ingredients (bill of materials) and instructions for their acquisition (supply chain), preparation (production process), support (customer service), and even disposal. Perhaps the most important aspect of the recipe is confidence in its ability to satisfy project stakeholders, who desire certain levels of product performance (which includes a product's features, functions, aesthetics, reliability, producibility, cost, etc.). Hence, much of the development effort is spent on experimentation to confirm hypotheses about the product design [114]. The value added by such experimentation comes in the form of information that increases confidence and decreases uncertainty and risk [12]. However, the actual performance level of an evolving product recipe can be difficult to verify immediately, and prior PD process models have not accounted for it [18]. Meanwhile, two nonprocess models by Paquin *et al*. [81] and Browning *et al.* [12] capture the typical influences of individual activities on project-level requirements and use these maps to anticipate the likelihood that a set of activities will lead to desired design performance levels. These models draw upon the convention in the marketing literature to treat a product as a vector of attributes important to customers (e.g., [38], [61], [94]). One can measure each attribute using one or more *technical performance measures* (TPMs) [85]. Thus, one can specify the technical goals of a PD project in terms of the attributes that matter to its stakeholders, with at least one TPM per attribute [12].

Motivation 4: Product performance can be represented as a vector of attributes, each measured by one or more TPMs, which collectively define the technical performance level of a project, which can be seen as one aspect of its overall value.

To account for uncertainty in a product recipe's technical capabilities, Browning *et al.* [12] used TPMs represented by random variables with a probability density function (PDF) or distribution. For example, the left side of Fig. 2 shows two triangle PDFs that represent a "larger is better" (LIB) TPM at two project states, n and $n+1$, before and after a set of activities. Here, the information produced by the activities serves to improve all three estimates (worst, most likely [mode], and best) of the TPM, thereby shifting its triangle PDF to the right. The right side of Fig. 2 exhibits the evolution of a TPM through successive states. Due to high technical uncertainty at the beginning of the project, the TPM initially has a high variance in potential outcomes, and the most likely outcome falls well below the target. Yet, in this example, the information created by the project's activities improves the outcome distribution relative to the target and also shows increased confidence through the reduction in the variance and range of the TPM.

A product recipe's level of technical performance is one aspect of a PD project's value. TPM profiles can capture the evolving technical performance levels. The resources (such as time and money) used to attain these technical performance levels is another aspect of project value. The amount of time and money required by a project can be similarly represented by outcome distributions, and these also tend to narrow over the course of a project [13]. For cost, duration, and technical attributes of a project, a project manager can weight each potential outcome by its probability and consequence—e.g., in terms of its effect on expected sales revenue—and, thereby, calculate a project's expected value and value at risk, thus translating the consequences of these strengths and weaknesses into monetary terms to support decision-making [15]. From this point of view, project value increases as the portion of that value which is at risk decreases. The value of a project at any state (start, interim, or end) is a function of the current performance of the product recipe (including the risks to that), the time and money spent, and the time and money deemed necessary to get the product recipe to the point of achieving the targets with sufficient confidence (indicated by a sufficiently low amount of technical value at risk). During project execution, a set of performance measures can represent the project's state relative to its goals, and thus its current value. Mapping the expected effects of individual PD activities on particular TPMs provides the connection between product success and project planning and control.

Motivation 5: The execution of activities: 1) uses resources; 2) creates deliverables that can revise TPMs; and 3) thereby adjusts the state (and value) of a project.

D. Project Process Adaptation

In this paper, we view the *process* of accomplishing a PD project as a kind of complex adaptive system (CAS). A CAS is a system composed of independent but connected agents that collectively adapt and self-organize, causing the overall behavior of the system to emerge over time [49]. For example, the cars on a highway interact to cause traffic patterns, and competing firms cause business patterns. Researchers have used CAS theory to explore many managerial topics including supply chain management [21], [65], [82], organization change [33], [72], [87], [101], [111], invention [39], innovation [9], [19], [52], strategy [65], [107], and PD projects [70]. While CAS theory has had a strong influence on organization science, so far it has been explored much less in the context of project processes. Here, we consider activities as the *agents*, deliverables as their *connections*, simple rules for activity selection and deliverable flow as the basis for *self-organization* and *adaptation*, the *emergence* of a process path, and its implications for project *fitness* or value. Project state and process value thus *coevolve* over the course of traversing the project territory or *landscape*, which can be dynamic and rugged.

Several researchers have modeled aspects of PD *process* adaptation, both at the strategic and tactical levels. At the macro level—using generic, undifferentiated activities—Pich *et al.* [84] characterized a project's process in terms of its information structure (knowledge about the *state of the project* and the world) and contingency plans, which managers can compare in order to dynamically redetermine appropriate actions. Huchzermeier and Loch [50] measured flexibility in terms of managerial options to delay, abandon, contract, expand, switch, or improve a project. Sommer and Loch [105] mentioned the combined challenge of unforeseeable uncertainty (the inability to recognize influence variables and their relationships) and high complexity (a large number of variables and interactions) and noted two approaches in this context: trial and error learning (continued, flexible adjustment of considered actions and targets) and selectionism (pursuing several, independent approaches and choosing the best one *ex post*). While these macro models provide strategic advice, they provide less tactical advice to project controllers. These studies also used artificial project landscapes.

At a level of differentiated activities, but for general business processes, several researchers (e.g., [55], [90], [92], [93]) have investigated and developed frameworks for managing adaptive workflow systems. However, these systems tend to treat exceptions or deviations from plans as opportunities for correction rather than desirable process behaviors, and they do not lend themselves as well to the dynamic, iterative nature of PD processes.

At the micro level of individual designers who must collaborate, Danesh and Jin [28] used an agent-based model of an ongoing set of decisions about which activities to do, coordinated to convergence through common policies. For engineering design projects, the *signposting* method [24], [78], [125] dynamically selects activities during process execution based on the confidence level of a potential activity's inputs. Policies based on the state of the information inputs and the capabilities of the activities govern activity selection and timing. A pre-evaluation selects the appropriate version of an activity and a postevaluation step determines if iteration is needed. Recently, a similar approach has been proposed by Karniel and Reich [54]. Similarly, Chung *et al.* [22] advocated a "grammatical approach"[6] to process specification and defined a *process space* of all possible activities and their arrangements. Indeed, the analogy of process to grammar and language seems powerful: building emergent processes from standard activities mirrors the way people can write a variety of creative literature using a commonly accepted language [102]. *The standardization actually enables the creativity.* Finally, Pall [80] used a "network of commitments" framework to design adaptable processes, where a key is to predefine acceptable ranges of interactions (instead of only point values) so that robust commitments can be made.[7] We assimilate these insights as follows:

Motivation 6: A PD process may be modeled as a CAS, where activities are agents, deliverables imply their connections, and a process path emerges from the exercise of simple rules for activity selection and deliverable flow. The fitness of (value provided by) this process will depend on the dynamic state of the project (duration, cost, and technical performance) and its environment (represented by project goals).

An adaptive process is amenable to alteration in the event of either a change in the goal or a rejection of the hypothesized (planned) path to achieving the goal. These changes may

[6]"As the grammar for a language describes all possible sentences, a process grammar describes all possible arrangements of tasks in a design process. Rather than focusing on a particular process, a grammatical approach draws attention to the set of alternatives" [22]. For further background on process grammars, see [83].

[7]This process engineering concept is related to the product engineering concept of set-based design [104]. Robustness is similarly relates to the concept of *sensitivity* in activity overlapping [60] in that a robust process is insensitive to perturbations.

involve new activities (to produce new deliverables), different versions of old activities (to produce different deliverables), iteration of old activities (to improve the maturity of deliverables), or elimination of old activities (because certain deliverables are not needed). These changes are facilitated and accelerated when potential and likely adjustments to the activity and deliverable sets have been anticipated—i.e., when potential "new" activities and deliverables have already received some consideration. It also helps to discern the potential process patterns that may emerge from the planned actions and interactions. Duimering *et al.* [36] argued that one can understand the changes to PD activities caused by ambiguity in terms of a finite set of predictable and potentially manageable patterns. This understanding and anticipation comes from additional investments in project planning, which involves systematic learning about the territory ahead—what is known and unknown, what is certain and uncertain [29], [30]. When one drives across a major city, for instance, it is highly useful to have not only a planned route, but also foreknowledge of the best alternative routes should an unexpected roadblock arise. Also, when composing a piece of creative writing, it helps immensely to possess fluency in the language used: instead of having just one way to express an idea, a gifted writer can select from many options to convey just the right nuance. Similarly, project managers can invest in better understanding of the project landscape. Often this is called "planning," but it is not just determining a supposedly optimal process architecture (what Eisenhower would have called "the plan"). Planning provides the basis for quick adaptability and agile project control. In other words, while some equate agility and adaptability with a lack of process structure, somewhat the opposite seems appropriate: *process structure gives project participants a framework for quick analysis* of what is likely to solve a pop-up problem, whereas a completely unstructured environment forces project participants to spend a lot of effort simply finding the information with which to make a decision [17]. Spear and Bowen [106] described how the Toyota Production System, paradoxically, was both rigidly specified and highly adaptable. However, enhanced adaptability has costs, largely reflected in a project's investment in planning and learning about the territory ahead.[8] Thus, we propose that

Motivation 7: Adaptability in PD projects is facilitated by advance knowledge of the potential activities and their relationships (planning) and their rules for combination (work policy), because this enables the activities to be quickly and effectively re-evaluated and reorganized over the course of a project.

According to Meredith and Mantel [74], the fundamental objectives of project control are (1) the regulation of results by altering alternatives and (2) the stewardship of organizational assets. Hence, project control requires continuously updated project information and decision alternatives [1]. To exercise adept control, managers require extensive information-sensing and -filtering systems. To support rapid decision-making, project managers benefit from having alternative courses of action conceived and analyzed. The most mainstream project control model, the earned value management system (EVMS) (e.g., [40], [86]), compares the planned and actual cost and schedule status of a project. An EVMS can be a helpful managerial tool, but it has at least four key shortcomings: (1) it takes a linear view of a process (ignoring iterations), (2) it does not account for a product's technical performance or quality, (3) it assumes a pre-specified set of activities will lead to project success, and (4) it does not close the control loop—i.e., it cannot recommend helpful courses of action contingent upon various project states.

Motivation 8: Systematic project control entails: 1) the synchronization of internal and external data regarding the state of the project; and 2) the use of those data in making decisions on project changes.

III. Model Constructs and Simulation Methodology

Our proposed APDP modeling framework stems from the preceding theoretical motivations. In this section, we describe the model's formulation and simulation.

A. Project State

We model a project's evolution through a series of states. Several variables characterize each state: cumulative time, cumulative cost, remaining time, remaining budget, and technical performance risk. While others have modeled projects as a progression through a series of states (e.g., using Petri nets), the APDP model is the first to do so in terms of the broad set of variables that encompasses cost, schedule, quality, and risk. In the basic model, we assume a project's goals are given: a deadline, a budget, and a set of technical performance targets. We also presume a single-attribute utility function for each goal. The utility (market response) functions indicate the expected change in project value (e.g., in terms of expected unit sales or revenue) for any change in cost, duration, or technical performance levels. Using the approach that Browning [14], [15] outlined, we use these inputs to calculate the expected value of a project and (as described in Section II-C) the risks of failing to provide this value given the current distributions of the project's state variables.

For example, consider a simple PD project with the following goals: a budget of \$30 000; a duration of three months; and only one technical performance attribute, mass, for which the target is 3.2 g or less. If these goals are met, the project is expected to be worth, say, \$100 000, whereas a product with a greater mass, and/or a project with a cost or schedule overrun, may reduce that value according to the pertinent utility function(s). At some interim state, suppose that the project is on schedule, within budget, and has technical performance estimates (for mass) of, in the best case, 3.0 g; most likely case, 3.5 g; and worst case, 3.9 g. Using these three estimates as the basis for the outcome distribution (in the form of a triangle PDF) normalized to unity, we weight the large percentage of outcomes that fail to achieve the target by their impact (in terms of lost project value) to arrive at an estimate of the project's "value at risk."

[8] Of course, it is possible for a project to invest too much in adaptability, if the need to utilize that capability is unlikely. So, the question of how much to invest in planning remains an important one for further research. The appropriate combination of planning and "doing" also relates to balancing exploration and exploitation (e.g., [5]).

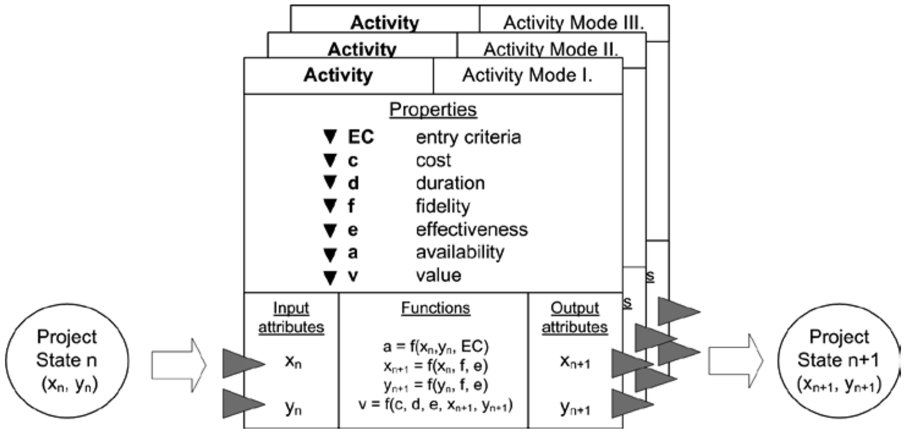


Fig. 3. Activity mode objects and attributes.

Suppose that the utility function is fairly flat until 3.7 g (such that outcomes in the 3.2–3.7 g range do not have much negative impact) but then increases rapidly above that point, so that the current estimates of the outcome distribution place, say, \$40 000 of the project's value at risk (thus giving it an expected net value of \$60 000). (For fuller details of the calculations, see [15].) Hence, at this point in the project, cost and schedule look fine, while technical performance is problematic, and the value at risk is \$40 000. In this state, appropriate project control actions call for an increased focus on technical performance and the activities (such as design changes, investigations of alternative materials, iterations of strength and stress analyses, etc.) with such an emphasis.

Next, suppose at some later state of the project, it is 5 days behind schedule, with a cost of delay (a linear utility function) of \$2000/day; it is \$5000 over budget; and mass estimates are 2.9 g (best), 3.1 g (most likely), and 3.4 g (worst). In this state, suppose the project's value at risk is (5 days) (\$2,000/day) + \$5000 + \$5000 = \$20 000, where the last \$5000 is due to the technical risk from the (now lesser proportion of) outcomes that exceed the 3.2 g target. In this state, project control calls for an increased emphasis on speed and efficiency.

Thus, we model each project state as a function of time, cost, technical performance, the goals for each of these, and stakeholder utilities. We can combine these to arrive at a scalar for project value at each state, although we will instead use the converse, the expected amount of that value at risk, represented by the *risk index* $\mathscr{R}$. This allows us to measure project progress in terms of a single index of added value (or risk reduction, $\Delta\mathscr{R}$). If the utilities are expressed in monetary terms, then $\mathscr{R}$ is also in monetary units, which is most intuitive. Even if $\mathscr{R}$ is in nonmonetary terms, it can still be useful to note its relative change, positive or negative, from one state to the next. It can also be helpful to normalize $\mathscr{R}$ over the continuous range [0,1], where zero indicates that none of the project's value is at risk and one indicates that all of it is (smaller numbers are better). This way, project managers can categorize $\mathscr{R}$ into selected ranges of low, moderate, and high risk, such as [0, 0.2), [0.2, 0.5), and [0.5, 1], respectively.

Here we use normalized risk indices for cost (C), duration or schedule (S), and technical performance (T) risks and combine them via a simple weighting into an overall project risk index $\mathscr{R}$

$$\mathscr{R} = w_S\mathscr{R}_S + w_C\mathscr{R}_C + w_T\mathscr{R}_T \quad (1)$$

where w_S, w_C, and w_T are the weights, which sum to one. These weights represent the relative importance of the project's goals. Of course, more sophisticated functions could be used in lieu of (1); each has advantages and disadvantages. Here we adopt the weighted average primarily for its simplicity.[9] Each individual risk factor, $\mathscr{R}_\varphi$, is evaluated as

$$\mathscr{R}_\varphi = \kappa_\varphi \int_{-\infty}^{G_\varphi} \tilde{P}_\varphi(x)\,[U_\varphi\,(G_\varphi) - U_\varphi(x)]dx \quad (2)$$

where $\varphi \in \{C, S,$ or any of the TPMs comprising $T\}$; x is an outcome; $\tilde{P}_\varphi(x)$ is the PDF; G_φ is the goal or requirement; $U_\varphi(\cdot)$ is the utility function; $U_\varphi(G_\varphi) - U_\varphi(x)$ is the impact (in terms of lost utility) of an outcome that fails to meet the goal; and κ_φ is a normalization constant [12]. The overall project value is highest when the portion of the project's value at risk is minimal—i.e., when $\mathscr{R} = 0$.

B. Activity Modes

Project state transitions, which we will discuss further in Section III-E, are affected by the accomplishment of activities. To model discrete activities, we adopt an object-oriented framework [16] based on classical input-process-output diagrams (e.g., [110]) and the information processing view of project activities (e.g., [42], [115]). Fig. 3 provides an overview of the activity mode objects and their attributes, which are defined in the followings.[10] Before going further, we note that the APDP model is independent of the approach to its representation and

[9]For a discussion of alternative approaches (see [10, Ch. 7]). It is important to note that one common disadvantage of the weighted average—the possibility of very poor performance in a low-weighted area being obscured by good performance in a high-weighted area—can be overcome by nonlinear utility functions that place a large penalty on poor performance and marginal returns on great performance.

[10]The state variables x and y, the input and output attributes, and the functions shown in Fig. 3 are stylized for example purposes only. The full explanations of these items follow in Section III-B.

simulation. We could construct this type of model using any of several potential frameworks, including IDEF0, Petri nets, GERT, DSM, etc. A model such as this has not previously been built using any of these frameworks, although it could have been. We adopt the DSM as a representation technique because of its conciseness, but the APDP model is not DSM-dependent.

An *activity mode* is a version of an activity, a particular way of getting a desired output. Each mode has a similar purpose, in general, but different characteristics. For example, there may be several ways to conduct a product test: evaluating a simulation model, testing a quickly fabricated prototype, or testing an actual piece of hardware. Each of these alternatives, or activity modes, may have different inputs, entry criteria, costs, durations, and benefits. Hence, an activity's mode of execution influences all of its other attributes. An activity can also have one or more specific iteration or rework modes, which may vary depending on the type of input change that triggered the rework. Varying an activity's mode provides a more intricate means of evaluating possibilities for crashing (faster execution at varied expense) and the use of alternative technologies. The activity mode is a kind of agent and is the atomic building block of the APDP model.

Entry criteria (EC) indicate the minimally acceptable level of technical performance or the maturity required of the inputs in order to perform the activity mode.[11] Executing an activity without satisfying its EC would compromise its effectiveness. Thus, the APDP model currently prohibits the execution of activities whose EC are unsatisfied unless no other option is available. For example, the EC for a test activity might be the number of lines of code implemented in software, the expected level of performance of an aircraft propulsion system, or the precision with which the aerodynamic drag of an aircraft can be qualified.[12] EC are specified during the calibration of an activity mode for a certain purpose in a project by considering historical data and expected results. The EC act as "mini toll gates" and enforce a nominal degree of precedence constraint on a process. Here, they provide one of the "simple rules" of activity interaction and self-organization in the adaptive process. Note that the use of EC allows activity overlapping. Krishnan *et al.* [60] argued that not all kinds of information can be exchanged in a preliminary form, so they recommend the identification of adequate information types and the definition of packages of information with increasing maturity. Here, overlapping occurs by breaking down activities into smaller segments with varied EC. The use of appropriate EC can also mitigate undesired iterative design oscillations ("design churn"—e.g., [71], [75], [126]).

Availability (*a*) is a Boolean value, "TRUE" if the project can perform the activity mode given the project's current state (i.e., if the activity's EC are satisfied) and "FALSE" otherwise.

Cost (*c*) and *duration* (*d*) are random variables, each with a distribution of possible outcomes. In the absence of further information, we use three estimates (optimistic, most likely, and pessimistic) to generate a triangle PDF of potential *c* and *d* outcomes, as discussed in [13].

Fidelity (*f*) represents the TPMs expected to be influenced by the results of the activity mode and the degree to which this influence occurs. For example, a high-fidelity activity mode might entail producing a detailed model or prototype that should yield a great deal of information in a number of areas about the hypothesized product design, whereas a low-fidelity activity mode might produce quick but approximate feedback on the feasibility of just one aspect of a design [114, p. 101]. Thus, the fidelity attribute records a list of the TPMs typically affected directly by the activity, the type of effect on each, and the typical magnitude of the effect in general terms ("high," "medium," or "low") in each case. As described in Section II-C, an activity may affect a TPM distribution by (1) shifting the most likely value and/or (2) shifting the best and worst values. Since each effect can be positive, negative, or unknown in its direction of change, this yields nine possible types of effects, which may be envisioned as a 3×3 table, where the rows represent the improvement, unknown change, or worsening of the most likely value and the columns represent the reduction, unknown change, or growth in the uncertainty bounds [12].

Effectiveness (*e*) is a dynamic attribute that depends on the state of a project. An activity that could be highly effective at one point in a project may be less beneficial at another point. Effectiveness is determined by using the fixed information regarding fidelity and calculating an expected benefit of the activity, given the project's current TPM distributions. Thus, an activity mode that provides a great amount of uncertainty reduction for a particular TPM will be more effective when the uncertainty in that TPM is high and less effective otherwise. For example, a product design simulation may be much more valuable early in a PD project and much less later on when hardware tests become more beneficial. For another example, many companies have standard procedures to correct typical design failures, including activity modes with a special focus on failure correction. The value of such activity modes or procedures is usually low, until the special kind of failure is detected. If the failure is found, then this activity mode becomes highly beneficial because it was developed specifically for the situation. In our current model, we determine *e* as the expected reduction in overall *technical* risk, which is a function of *J* TPMs (i.e., it does not include cost or schedule)

$$e = E[\mathscr{R}_T] = \sum_{\varphi=1}^{J} w_\varphi \left(\Delta\mathscr{R}_\varphi\right) \qquad (3)$$

where $E[\mathscr{R}_T]$ is the expected overall technical risk reduction provided by the activity, $\Delta\mathscr{R}_\varphi$ is the expected risk reduction in TPM_φ after executing the activity mode, w_φ is the relative importance of TPM_φ, and the weights again sum to one. $\Delta\mathscr{R}_\varphi$ will be zero for most φ for a particular activity mode, as most activity modes will only affect one or a few TPMs directly.

The *expected value* ($E[v]$) of an activity mode balances its benefits (*e*) with its costs (*c* and *d*) and is calculated using a

[11] EC are similar to the parameter confidence levels required to perform an activity in the signposting approach [24], [78].

[12] Process models of this type may also include exit criteria, which are implicit in our model since we assume they can be met given enough internal iterations of the activity.

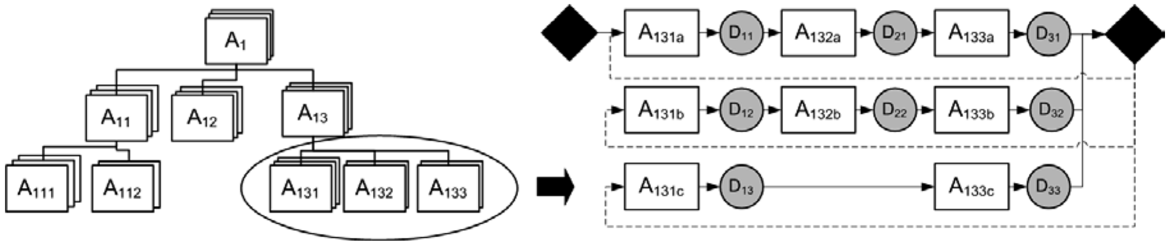


Fig. 4. Example of generating an initial superset of activity modes.

form of (1):

$$E[v] = w_S(\Delta\mathscr{R}_S) + w_C(\Delta\mathscr{R}_C) + w_T(\Delta\mathscr{R}_T) \quad (4)$$

where $\Delta\mathscr{R}_\varphi$ is the expected risk reduction provided by the activity mode, given its resulting c, d, and e. While an activity mode will almost always provide some positive expected benefit (e), this will vary depending on the project state, and it will often be offset by the concomitant increase in project cost and schedule risk ($\Delta\mathscr{R}_C < 0$ and/or $\Delta\mathscr{R}_S < 0$) caused by c and d. Thus, v can be positive or negative. It is important to emphasize that v is a random variable prior to the actual execution of the activity mode, at which point actual amounts for c, d, e—and, therefore, v—become known. A decision about whether or not to attempt an activity mode must depend on $E[v]$ since the actual v is as yet unknown.

C. Establishing the Initial Superset of Activity Modes

Fig. 4 exemplifies the generation of an initial superset of activity modes. The left-hand side of Fig. 4 depicts a *work breakdown structure* (WBS) with the various activity modes represented as layers. The right-hand side of the figure shows the relationships between the modes of three of the activities, where the additional subscripts "*a*," "*b*," and "*c*" signify the activity modes: "*a*" is the conventional activity mode, while "*b*" and "*c*" are rework modes with special purposes, typically to be conducted only if the deliverables from the "*a*" mode are found to be inadequate upon subsequent design review.

While some classes of product development projects are so novel that it seems impossible to nominate even a superset of potential activities, the initial activity superset should include all of the plausible activities that the planners can anticipate. As they discover additional activities, they can add these to the mix. However, as noted in Section II-A, most projects will be able to anticipate a majority of the activities they *might* have to do. Thus, while assuming some prior knowledge of a project's potential activities, the APDP model considerably relaxes the presumptions of knowledge about the content and form of a project's network vis-à-vis conventional methods for project planning and scheduling. Nevertheless, the planning effort required to define the initial superset of activity modes is likely to be somewhat greater than that required to define any single project plan (that contains a smaller set of activities).

D. Simulating the Model

We explore the APDP model with a discrete-event simulation that transverses project states. Simulating project execution entails selecting activity modes according to simple rules—i.e., policies for choosing among alternative state transitions. At each state, the simulation determines the feasible next steps (the activity modes with satisfied EC) and the expected value of each. These activity modes may reside in one or more independent decision sets. (The independent decision sets allow for the concurrent execution of independent activities.) In each decision set, the activity mode with the highest expected value is selected, after which its actual attributes are sampled via Monte Carlo techniques. When the shortest-duration activity mode finishes (of all of the activity modes working concurrently), then that activity mode's duration and benefits, and all concurrent activity modes' expended costs, are accounted for to determine the project's new state. Run thousands of times, the simulation enumerates a variety of potential paths through the project and the properties of each, thereby illuminating the process space. We discuss these steps further in the following subsections.

E. State Transition Decisions

The path from the beginning to the end of a project consists of a set of states, punctuated by activity modes. The states attained will vary depending on the activity modes chosen. Selecting an activity mode is a three-step procedure. First, the eligible activity modes are determined by comparing their EC to the current project state.[13] Second, the eligible activity modes are sorted into independent decision sets. Each decision set represents the set of options available on a particular process thread. Third, the activity mode with the highest expected value is chosen from each decision set. This approach follows the decision-based design perspective [45], [116] in PD and uses the insight from agent-based systems that dynamic, weighted relationships (edges) permit a network to exhibit and encode learned knowledge and adaptive states [58].

Fig. 5 demonstrates the decision framework for state transition and project adaptation. Suppose that at a given project state, there is one decision set consisting of three eligible activity modes: A2, B1, and B2. Each has a distribution of potential

[13]The model accounts for activity mode "self-iteration" by including its effects within the stochastic activity attributes. Also, the basic model assumes resource availability, but a step could be added to disqualify activities for which adequate resources are not available. We save exploration of the effects of resource constraints for future research.

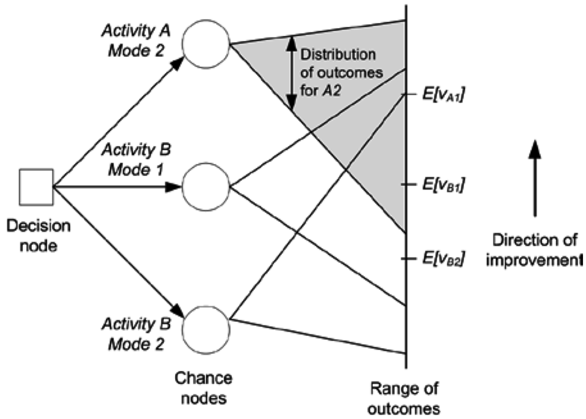


Fig. 5. Example of the decision framework for state transition under uncertainty.

outcomes for v (shown stylistically in the figure as a triangle distribution emanating from each chance node) and thus an $E[v]$. A2 is selected because $E[v_{\mathrm{A2}}] > E[v_{\mathrm{B1}}] > E[v_{\mathrm{B2}}]$. However, when the actual v_{A2}, $A[v_{\mathrm{A2}}]$, is sampled, it is, of course, possible for it to be less than this expectation—i.e., $A[v_{\mathrm{A2}}] < E[v_{\mathrm{B1}}]$ is a possibility. Thus, a good decision could have a bad outcome, and a bad decision might have had a good outcome. The basic model assumes a rational, risk-neutral decision maker. The state transition decisions allow for a contingent model of iteration and rework, where activities will tend to be iterated when weighted technical risk is greater than weighted cost and schedule risk—i.e., when $w_T \mathscr{R}_T > w_S \mathscr{R}_S > w_C \mathscr{R}_C$ in (1). (If the EC of none of the downstream activity modes are satisfied, then iterative rework is the only way to continue the project.) In the basic model, a simulation run ends when a nominal subset of activity modes has been completed and either technical performance risk is "low" (e.g., below 0.2) or cost or schedule risk is "too high" (e.g., above 0.5), although other suitable stopping criteria could be adopted.

F. Stylized Example

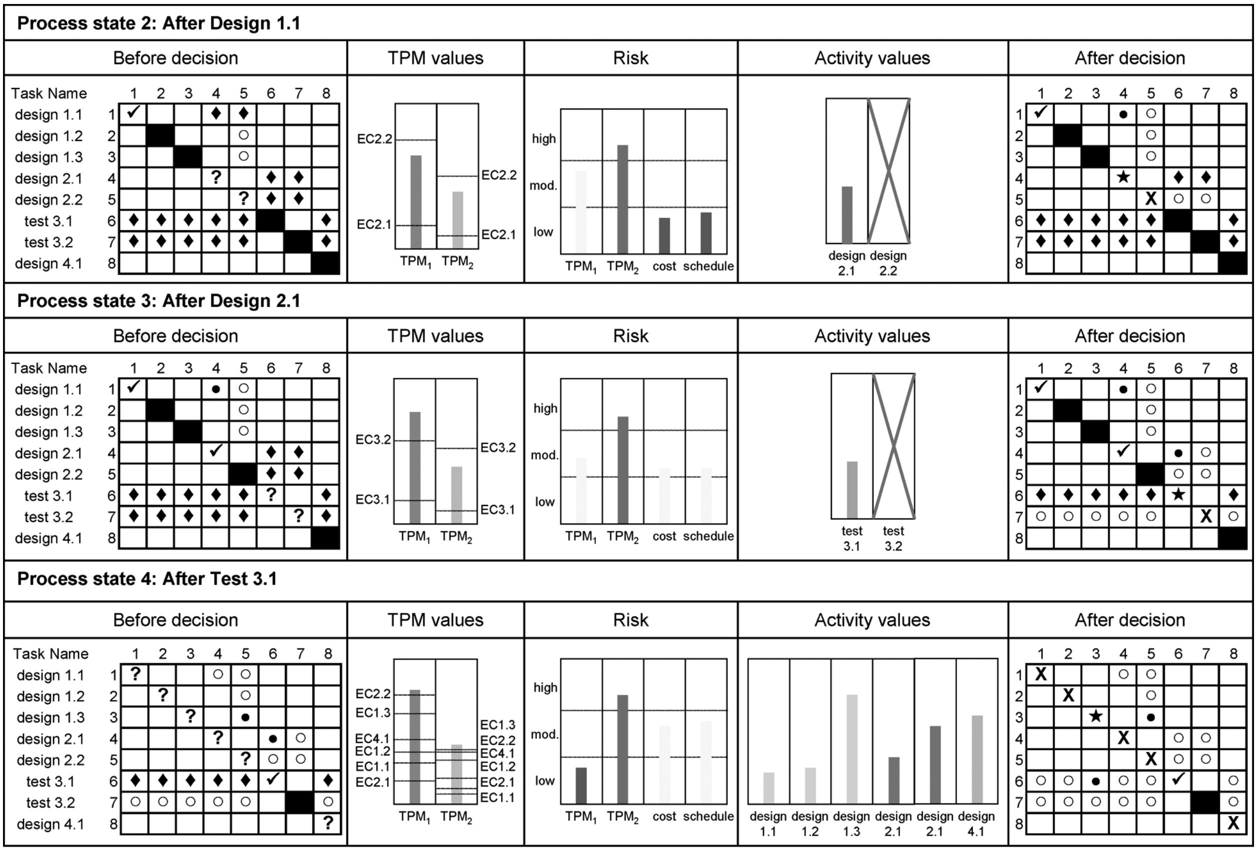
We now demonstrate the APDP simulation logic on a simple project with two TPMs and a superset of eight activity modes. In its main rows, Fig. 6 shows three project states along an emerging process path. Each row pertains to a particular process state transition, and each row begins with a DSM representation of the process (as described in Section II-B). For example, the DSM in the upper-left corner shows potential outputs from the activity mode "design 1.1" flowing to either activity mode "design 2.1" or "design 2.2." The DSMs in Fig. 6 use the following notation. On the diagonal, *checks* (✓) denote completed activity modes; *question marks* (?) show potential next activity modes; filled black cells indicate unavailable activity modes; *stars* ($\star$) represent chosen activity modes; and "X"s mark unselected activity modes. Off the diagonal, *solid circles* (●) depict active relations, *empty circles* (○) inactive relations (paths not chosen), and *solid diamonds* (♦) yet-to-be-determined, contingent relations. The activity superset in this example includes both "broad scope" activity modes (with x.1 numbering, such as "design 1.1"), which can be considered as general design and test activities, and rework modes (with x.2 or greater numbering) focused on correcting specific failures. Rework modes tend to have lower fidelity, cost, and duration due to their reduced scopes. Furthermore, since they are special activities to correct failures, the EC for rework modes are more stringent: a certain level of technical performance is required in order to perform them. (Otherwise, iteration can occur by reworking the primary mode.)

In the first row of Fig. 6, activity mode 1 ("design 1.1") has just finished, as indicated by the ✓ in the upper-left cell of the left DSM. Reading across row 1 in this DSM shows a flow from activity 1 to either activity 4 or 5 (as indicated by the ♦s). Hence, activities 4 and 5 ("design 2.1" and "design 2.2," respectively) are marked with "?". The next items in the first row of Fig. 6 depict the decision variables used to determine which activity will be chosen as the next step in the emergent process. To the immediate right of the DSM, the current level of design performance, estimated using the expected value of each TPM distribution, is shown to satisfy the EC for continuing on to "design 2.1" but not the more stringent EC for "design 2.2." Therefore, activity 4 is, in fact, eligible while activity 5 is not. The next diagram compares the current $\mathscr{R}_\varphi$ values for each TPM along with $\mathscr{R}_C$ and $\mathscr{R}_S$. Next, the "Activity Values" diagram shows $E[v_{\text{design 2.1}}]$ and that the EC for "design 2.2" have not been met. Thus, "design 2.1" is the only choice, and the far-right DSM shows the contingent flows (from activity 1 to activities 4 and 5 in the left DSM) now replaced by an active flow from activity 1 to activity 4 (the ●) and an inactive flow from activity 1 to activity 5 (the ○). Furthermore, activity 4 is now marked with a "$\star$" and activity 5 with an "X".

In the second row of Fig. 6, "design 2.1" has now completed, and the next options are tests 3.1 or 3.2. Again, the EC are satisfied only for the first of these two activity modes, so activity 6 is chosen. In the third row, "test 3.1" is finished and a choice must be made to continue on to "design 4.1" or to iterate one or more of the previous activity modes. (Activities 1 and 4, while done previously, now re-enter into consideration.) The EC are satisfied for all of these options, but the highest expected value comes from activity 3 ("design 1.3"), signaling that the first pass through these activities did not provide adequate risk reduction in the TPMs. In fact, TPM_2 remains at high risk, and design 1.3 is a rework mode targeted mainly at this TPM. While not shown with additional rows in Fig. 6, note that the next step in the process will be to do activity 5 and then either activity 6 or 7, as one can see by tracing the potential flow paths in the final DSM. Iterations will continue until technical risk is reduced to an acceptable level, or until the resource expenditures make the cost and schedule risk more problematic.

IV. Model Application, Results, and Managerial Insights

While the APDP model has strong face validity from its firm theoretical grounding, it is important to increase its validity further through application to an industrial project. In this section, we report on such an application, present some of the results, and



● - active relation; ○ - inactive relation; ♦ - contingent relation; ✓ – finished activity; ? – potential activity; ★ - selected activity; X – inactive activity;
EC – entry criteria, TPM – technical performance measure

Fig. 6. Three example states in an APDP simulation.

discuss some of their insights for managers. Since our primary goal in this initial application was to validate the feasibility of the model, we started with a fairly high level, sequential project representation and a single TPM.

A. Project Description and Model Inputs

TetraPak Carton Ambient (TPCA), a business unit of TetraPak, is the world leader in the development and production of packaging systems for liquid food products that can be stored at room temperature. TPCA operates in six countries with 2100 employees, with major PD and production facilities in Modena, Italy, and Lund, Sweden. We applied the model to a project at TPCA S.p.A. in Modena that dealt with designing a new production process in which semi-manufactured goods would feed into a generic transformation process that released a finished product. The project's main objectives were to evaluate the correlated effects of raw materials, the transformation process, and the normal variation in the noises (i.e., variability in product and process characteristics) versus defects in appearance, the nominal geometric dimensions, and the nominal tare weight of the container. The overall goal was to design and develop a robust and stable process by reducing variability in the parameter values and thereby increase confidence in the characteristics of the final product.

To build the initial model, the project leader provided alternative plans and strategies for various project scenarios. We used these to define an initial superset of activity modes across four project phases, as shown in Fig. 7. This superset included all of the feasible activity options identified and a rework mode for each. As for TPMs, TPCA wanted to focus initially on a single, major characteristic of the end product—the *tare weight* of the liquid food container. Extremely high production volumes make tare weight a key cost driver in the food packaging industry. Hence, potential variability in this parameter was the major source of technical risk in the project. Other important TPMs included appearance defects and geometrical dimensions, but we base our initial results only on the tare weight TPM. Similar historical projects and worker experiences provided the basis for the data concerning the typical, direct effects of the activity modes on this TPM. According to the test experts interviewed, the dispersion of outcomes defined by the coefficient of variation is 10–15% in real life experiments and 5% in laboratory tests. Therefore, we use these values as upper bounds on the effectiveness of an activity mode on a TPM; the lower bound is zero (no effect). For simulation purposes, the "actual" effectiveness of an executed activity mode is randomly sampled from a uniform distribution across this range. For the weights in (1), TPCA used $w_C = 0.25$, $w_S = 0.25$, and $w_T = 0.5$ at the beginning of the project and thereafter allowed the weights to vary dynamically, as follows. If an attribute had "high" risk (i.e., $\mathscr{R}_\varphi \geq 0.5$), it

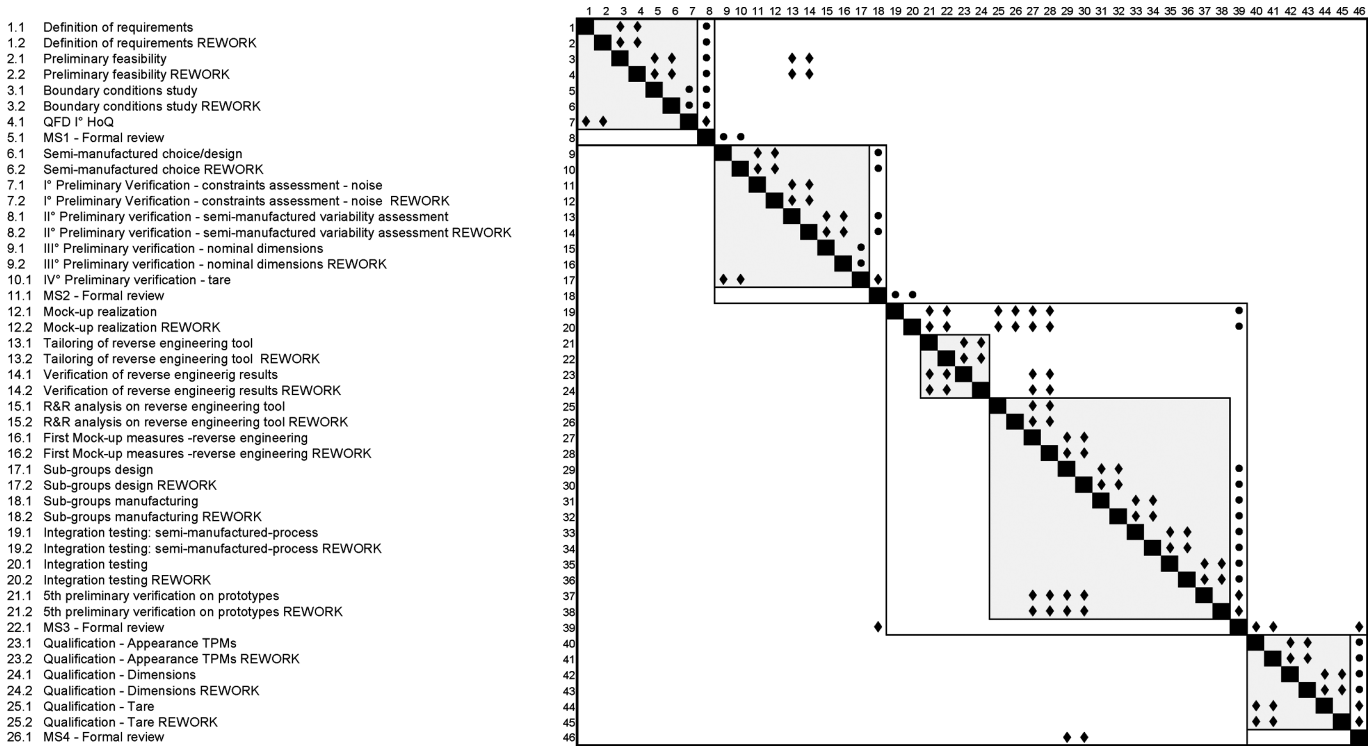


Fig. 7. DSM showing the activity mode superset and the potential, deliverable flow paths for the TPCA project.

received a preliminary weight of 3; $0.2 \leq \mathscr{R}_\varphi < .05$ implied a preliminary weight of 2; and $\mathscr{R}_\varphi \geq 0.2$ implied a preliminary weight of 1. These preliminary weights were then normalized to determine w_C, w_S, and w_T.

Fig. 8 shows the names of the four project phases and some characteristic results from a single simulation run. For each of four interim states of the project (after each of the four phases), a PDF represents the relative likelihood of the TPM outcomes. The vertical lines with each PDF show the target at the end of each phase. TPCA changed the target for each phase by continually lowering it (i.e., making it more difficult to achieve). The region of each PDF to the right of the target represents the outcomes that fail to achieve the performance objective. The consequence of each of these unsatisfactory outcomes is weighted by its impact (an inverse utility function) to determine $\mathscr{R}_T$, as in (3). Note that TPCA also changed the utility functions for each phase to reflect the dynamic project environment.

We evaluated cost risk using the two time-invariant functions depicted in Fig. 9. We used these functions to translate the difference between planned and actual cost at each point during the project to a probability and an impact of an eventual cost overrun, respectively. In the example given in Fig. 9 example, the actual cost is higher than expected at a point in the project, and we take this difference to imply a 60% chance that the budget will ultimately overrun, but we deem such an outcome to have a fairly low impact (0.12) on the project's value. We multiplied these probability and impact numbers to get an estimate of $\mathscr{R}_C$, where $\mathscr{R}_C \in [0, 1]$. We used a similar approach to evaluate $\mathscr{R}_S$. Of course, we could have used a more sophisticated approach to estimating $\mathscr{R}_C$ and $\mathscr{R}_S$, but, working with TPCA, we deemed this approach sufficient for this project.

Based on these inputs to (1), TPCA's PD project had an initial risk index of 0.21, where $\mathscr{R} \in [0, 1]$, meaning that about 21% of the project's value was at risk. TPCA sought to accomplish a project that would serve to reduce this risk and thus increase value.

B. Results and Managerial Insights

Many PD process models reported in the literature do not explicitly note verification or validation efforts. Nevertheless, as recommended by Sargent [98], we used several verification and validation techniques during the model and simulation's development and application. Conceptual model validity hinged on acceptance of the theoretical motivations described in Section II and the basic constructs described in Section III. Computerized model verification focused on ensuring the correctness of the programming and implementation of the conceptual model. Operational and data validation occurred through review by managers and project experts at TPCA, who also confirmed the plausibility of the results vis-à-vis company models and historical experiences. We settled on analyzing a batch of 4000 simulation runs, because additional runs did not make a significant difference in the results. We also used event, face, and internal validation techniques, as described by Sargent [98]. For example, we ran two batches of 4000 runs each and compared them to ensure internal validity and consistency. The model and simulation also met the validation critieria outlined by Johnson [53]. Overall, the validity of the model compares favorably with that of other PD process models reported in [103].

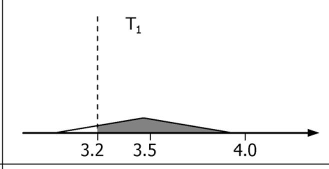
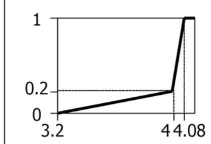
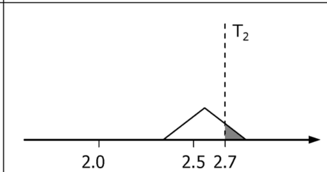
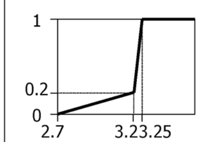
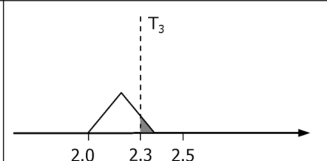
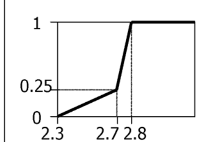
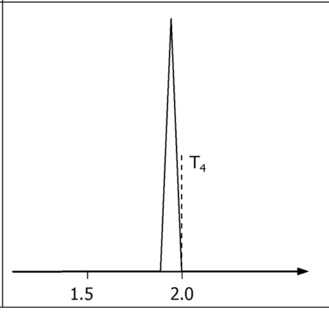
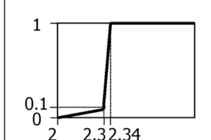
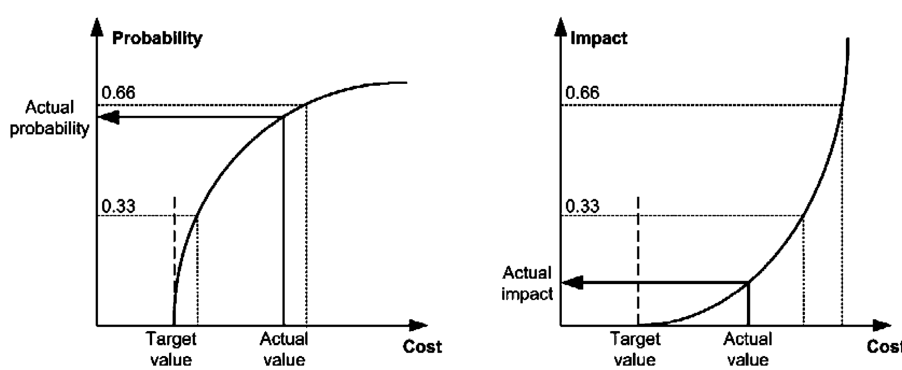
Phase	Product weight (g)	Impact Function	Target (g)	Risk (0.0-1.0)
Project Definition			3.2	0.15 low
Concept Development			2.7	0.08 low
Prototype Development			2.3	0.05 low
Product Qualification			2.0	0.04 low

Fig. 8. Behavior of the tare weight TPM over four phases of the TPCA project.



(a) (b)

Fig. 9. APDP risk calculation inputs for project cost. (a) Probability of failing the final target depending on the actual value at time point X. (b) Impact of failing the final target depending on the actual value at time point X.

After gaining confidence in the data, the APDP model, and its computerized simulation, we explored the process space based on the initial 4000 simulation runs. We will present five results and associated insights.

Project-Specific Result 1: Fig. 10 shows the frequencies of different process instances (emergent paths through the project landscape). The 4000 simulation runs yielded 2550 potential paths, most of which occurred only once. The single most common path occurred in only 153 of the 4000 instances (3.8%).

Managerial Insight 1a: Since many process models and conventional project planning tools assume a single process option (and focus even further on its critical path), these techniques would seem to have a high probability of yielding misleading results. A subset of project management literature (e.g., on GERT and DSM) has explored probabilistic branching in *predefined* networks and come to this conclusion. The APDP model takes the exploration of a process space to the next level and underscores the need for methods of planning with the

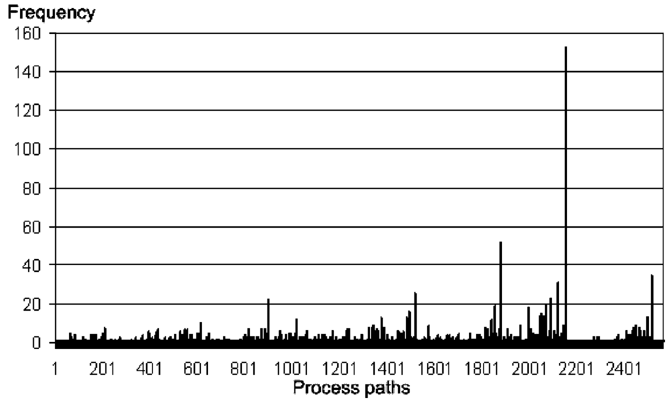


Fig. 10. Frequency of occurrence for each of the 2550 unique paths (counts sum to 4000).

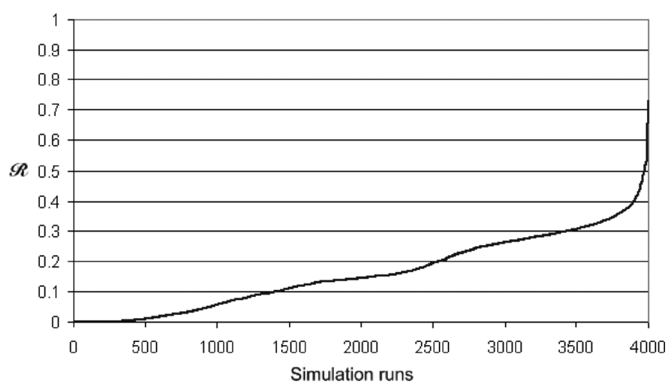


Fig. 11. Final risk indices for the 4000 outcomes, sorted from best to worst.

capabilities to account for process changes and evaluate a variety of potential outcomes.

Managerial Insight 1b: The ability to combine activities into a greater variety of process options increases managerial flexibility and value. Starr [109] noted how modular production capabilities allow for a greater variety of outputs. According to options theory, greater variety in outcomes can lead to increased value [50]. The APDP model demonstrates the concept of modular PD process capabilities and holds out the possibility of similar benefits for project managers.

Project-Specific Result 2: Fig. 11 plots the 4000 process paths, ranked by their final, overall risk level. Two thousand five hundred and eighty-eight (64.7%) of the paths were considered successful in that they reduced the overall project risk to a "low" level, where "low" was specified as $\mathscr{R} < 0.2$.

Managerial Insight 2: In the APDP model, the simple rules guiding process adaptation aim to maximize a project's expected value by minimizing the portion of that value at risk. However, all alternative processes do not provide equivalent benefit in this regard. A large percentage of unsuccessful paths (ones that do not sufficiently reduce $\mathscr{R}$) could foreshadow an especially challenging (or ill-advised) project.[14] In this case, if a "low" risk outcome is necessary to justify the project (e.g., in terms of

[14]The threshold for this percentage depends on the risk attitudes of the project's stakeholders.

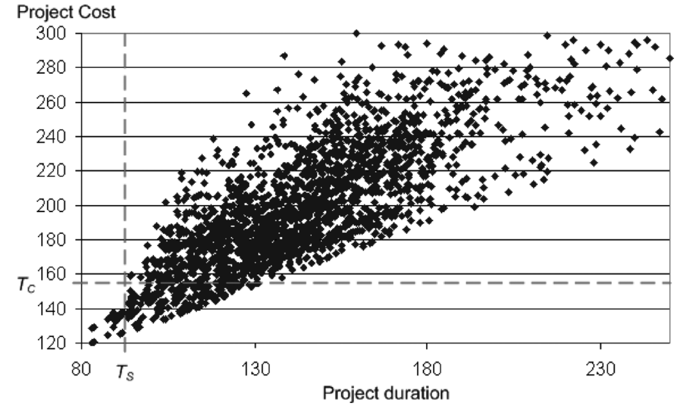


Fig. 12. Scatter plot of 4000 project cost and duration outcomes.

expected return on investment), then the project should be abandoned in 35.3% of the cases (since the other 64.7% of cases were successful). By calculating the expected value of the project as the average of the final values of the 4000 outcomes, one could show how exercising the option of abandonment in more than a third of these cases would increase this expected value. Moreover, identifying the specific paths that portend project failure could provide project managers with leading indicators of problems (e.g., if they find the project to indeed be on such a path). These indicators could help forecast if a project should be abandoned prior to actually reaching the state where all of the potential next steps have indeed negative expected values. In addition, the model can show the risk reduction provided by setting easier targets, and this can be traded off against the corresponding reduction in the baseline value of the project (when $\mathscr{R} = 0$).

Project-Specific Result 3a: A majority of the simulated outcomes exceed the project cost and duration targets (i.e., the budget and the deadline—T_C and T_S, respectively), which are shown overlaying the scatter plot in Fig. 12 at 154 cost units and 87 time units. (Actual numbers have been disguised to protect company data.) Managers agreed in hindsight that the cost and schedule goals for this project were set too optimistically. The probable number of iterations had been underestimated; correcting even small technical problems would have caused cost or schedule overruns.

Project-Specific Result 3b: Iteration patterns can be observed by examining the structures of individual process paths, such as the two of special interest in Fig. 13. The most likely process path, shown on the left side of Fig. 13, reduced risk significantly in the early phases of the project via "cheap" iterations using virtual prototyping methods. On the other hand, the worst path (leading to the least project value) on the right side of Fig. 13 contained many late, expensive iterations. Since TPCA usually does product qualification (the final project phase) at a customer's site using the actual product (e.g., milk or fruit juice), any design failure found in this phase (that causes rework or retesting) delays the start of liquid food production and tremendously decreases project value.

Project-Specific Result 3c: The results indicate a high likelihood of design iterations. Only 2.5% of the emergent processes

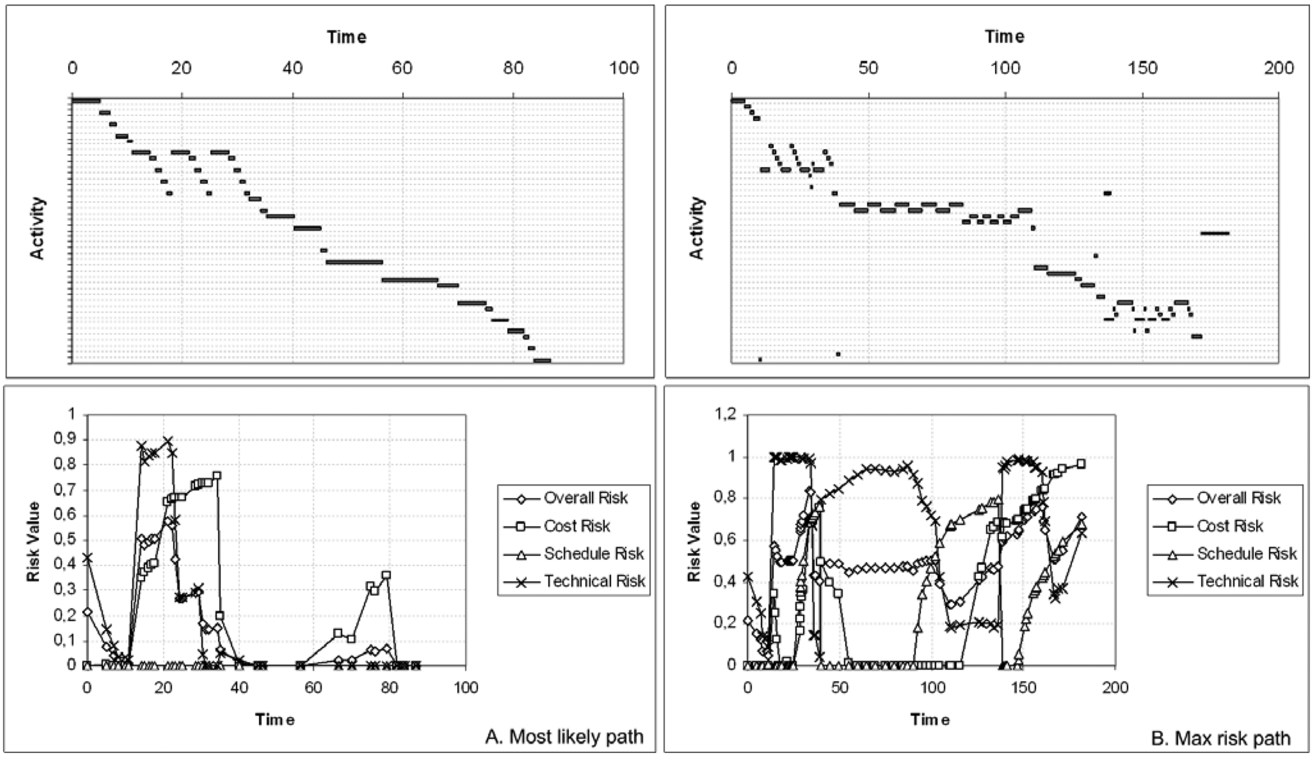


Fig. 13. Two interesting process paths out of the 4000.

fulfilled all project goals without iterations. Interestingly, these were *not* the paths providing the highest overall project value. Rather, in these cases, technical performance risk remained "low" (yet nonzero) throughout the project, which led the simulation to decide against iteration at the state transitions. Hence, an amount of technical risk was allowed to persist until the end of the project rather than being driven out. Meanwhile, 97.5% of the paths included one or more iterations. While the high-risk process paths had many iterations (e.g., the right-hand side of Fig. 13), the results indicated that even one to two iterations in the late phases could have catastrophic consequences. However, a variety of high-value paths had —two to three iterations in the early phases.

Managerial Insight 3: These results confirm some previous findings regarding the significance of iteration and rework as primary drivers of PD project cost and duration (e.g., [26], [79], [95]). Furthermore, the APDP model provides a deeper insight into desirable and undesirable iterations. First, we confirm that *the amount of iteration does not matter as much as its timing and scope* [13]. In fact, in many cases, iteration is quite helpful. More specifically, the results support the benefits of additional, early iterations and the detriments of additional, late iterations. The early iterations also seem to be much smaller in scope (as there is less other completed work for them to impact). For example, we find support for the frontloading of activity modes that discover design failures early in the project and allow for their correction in short, inexpensive iterations [112], [114]. It is, therefore, important to plan for this and allocate adequate resources to the early phases. However, the requisite activity modes to support quick and inexpensive—yet appropriately effective—experiments early in the design process must be included in the initial superset of activity modes. When they are, they may often be selected. Exploring the exact conditions under which frontloading certain activity modes makes the most sense is an area for continued research. In such studies, researchers could use the APDP model to explore the specific ways in which front-loading would best occur. Instead of merely arguing for a general allocation of additional resources early in a project, planners could use the model to help identify the leverage points (specific new or enhanced activity modes) where these resources could provide the greatest expected benefit.

Project-Specific Result 4: Fig. 14 shows the frequency of each activity mode (on the diagonal, as a count) in the 4000 simulation runs. (Iterations may cause an activity mode to occur more than once per run, which is why many of the entries on the diagonal exceed 4000.) Interestingly, some activity modes were very popular, whereas others were rarely picked. For example, modes 21 and 23 were highly popular, whereas modes 25 and 26 were hardly used. In the first phase, we note that rework occurred more often through the conventional activity modes (1, 3, and 5) than through the rework modes (2, 4, and 6). In phase three, one pair of (conventional and rework) modes (21 and 22) was conducted a total of 23 047 times, while another pair (25 and 26) occurred only 45 times.

Managerial Insight 4: Interestingly, our results suggest that *high-value PD processes require not fewer activity options (as suggested by lean) but more*. Activity mode frequencies may provide an indication of their current value to the project. Since this project included two modes for most activities, it is

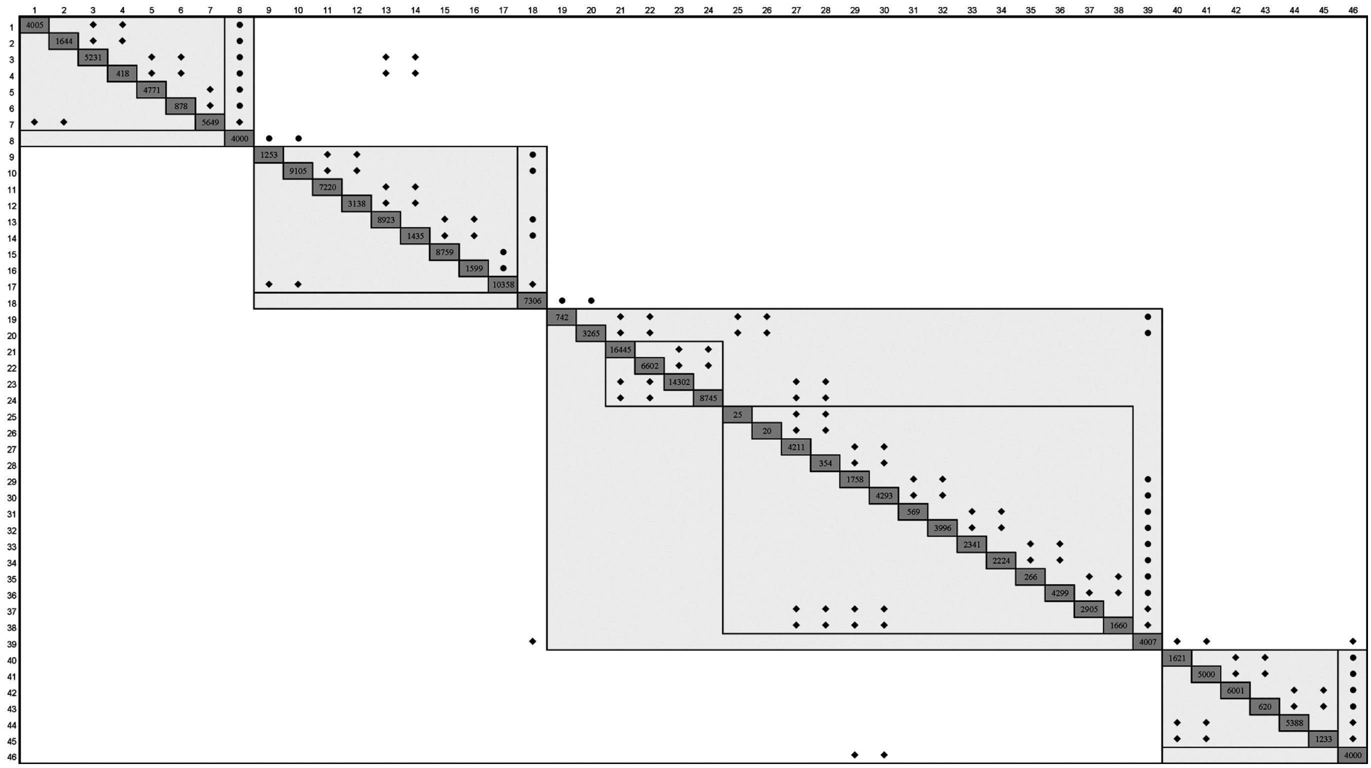


Fig. 14. Frequencies of activities and flow paths in 4000 simulation runs.

interesting to note where either the regular mode (e.g., 19, 31, and 35) or the rework mode (4, 28, 43) was selected infrequently. In such cases, it may be prudent to redesign the less popular modes by prescribing different technologies, methods, and/or personnel that change the mode's attributes. Or, perhaps a different form of task decomposition would be more appropriate [36], [119]. Hence, one can pinpoint the "weakest links" in a process space for strengthening, or perhaps even removal. Examining a process space offers a more sophisticated avenue toward process improvement than the approach advocated in the paper on lean or value stream mapping (e.g., [124]), for example, where individual activities are determined to be wasteful (or not) according to their internal characteristics only, without regard to their various modes or the timing of their occurrence. Our results show that "waste" turns out to be a dynamic attribute. Because an otherwise value-adding activity will be worthless if it is done with incorrect inputs, it is important to consider activity value from a system perspective in the context of an overall process [14]. Here, that context is further extended to a process space containing a large number and variety of process options. The APDP model considers the potential value of an activity mode as a contingency or option. This information can help managers decide where to invest in developing new modes (perhaps through new technologies and tools) and where such investments are unlikely to show much return on a project (because the modes are unlikely to be used). This way of investigating the prospective value of an activity accounts for many factors, including its timing (in the process), EC, and other attributes. Thus, the model and its results may be used by managers to explore where and how potential additions and subtractions to the activity mode superset might benefit the project.

Detailed specification of these capabilities provides another area for future research. In this realm, one could explore ties between the use of specific activity modes and successful (simulated) project outcomes. An area of particular interest could be to determine where certain activity modes, while unpopular, might nevertheless prove beneficial in the highest-value paths. Meanwhile, other modes might be quite popular, yet primarily in the low-value paths. The activity mode usage frequencies might also inform multiproject planners' resource allocation decisions. For example, a testing facility that is highly unlikely to be used by a project could be prioritized for another project.

Project-Specific Result 5: While sometimes the simulation would select a conventional activity mode for rework, sometimes it would chose a rework mode for initial execution, despite the rework mode's more stringent EC. Since the simulation picks the eligible mode with the greatest expected value, process states with high technical risk often led to the selection of conventional modes for rework, and states with low technical risk often pointed to a lower fidelity version of the activity.

Managerial Insight 5: What will turn out to be the best-choice activity mode is not always obvious prior to the project. This result supports the admonition to project planners to delay activity mode selection until what Smith [102] called the "last responsible moment"—ideally, right before the activity should begin. This enables project managers to react to unforeseen situations using a set of "process building blocks." This insight aligns with findings on set-based design from the Toyota PD System [77], [104], [120] but applies them in the context of process design rather than product design. In situations in which delaying an activity mode decision seems problematic (e.g., because of long lead-time inputs or the need to presecure resources), the costs of delay could be traded off against the benefits, as illuminated by simulation of the APDP model and its accounting for specific activities and deliverables in the context of the entire process space. While this type of analysis would also be likely to highlight the inefficiencies attributable to resource inflexibilities and long lead-time items from external organizations such as suppliers, it would also help managers quantify the specific benefits of improving the most problematic situations.

C. Reactions From TPCA

The APDP simulation results convinced TPCA that its original project plan, defined using conventional methods, was too optimistic. The project manager had not anticipated the major risk posed by iteration. The simulation reduced the project's ambiguity by highlighting the critical iteration loops (previously unknown unknowns, caused by interactions among known elements) and prompting strategies to deal with them. The results also advanced frontloading as an effective strategy for efficiently reducing technical risk in an early project stage. Defining alternative modes for each activity also proved valuable for TPCA. Adding just a single rework mode to each activity increased the flexibility of the project plans and showed that some of the modes should be sized differently to better suit the project objectives. APDP modeling also provided the project manager with new process architecture and activity mode options; these pointed to unexpected project states and prompted the proactive consideration of appropriate contingencies. Overall, the application of the APDP model at TPCA demonstrated that thorough planning and the application of adaptive process simulation could reduce ambiguity and uncertainty at the outset of a project by shedding additional light on the territory between a project's start and its desired end state. Hence, TPCA recognized the importance of defining activity modes and other contingencies early and strategically.

V. Conclusion

This paper presents a new, theoretically grounded modeling framework for the process used to accomplish a PD project. Rather than charting a single course through uncertain terrain, the APDP model generates a *process space* (set of likely paths) based on the available *options* (activity modes) for stepping through the landscape, where the availability of certain steps depends on one's current position (i.e., the state of the project). Unlike prior models of project states, the APDP model accounts for not only the time and cost but also performance, risk, and value. Depending on the "terrain" encountered, a subset of the activity modes will combine (according to simple rules) to form a path (process) through the landscape. Rather than presuming that a particular set of activities and interactions is necessary and sufficient to achieve a project's goal, the model accounts more broadly for a superset of *potentially relevant* activity modes and interactions. From this "primordial soup," we explore what types of processes emerge and their comparative *fitness* (or value, in terms of risk reduction) in achieving the goals. The approach recasts project planning from detailing a single plan to gaining insight from a process space, via simulated project execution without extreme prejudice.

The model is intended to apply to any kind of PD project, although, like any model, it will run into difficulties at the frontiers of project novelty and ambiguity. However, it pushes further toward that frontier than other approaches, because it accounts for more unrealized interactions between known and potential elements. It also applies to less-novel PD projects: with greater knowledge and past experience from highly similar projects, it is possible to define a much richer initial superset of activity modes.

Through an initial application to validate the model, we found and confirmed several insights for researchers and managers. No single path was found to dominate the process space, which calls into question the management strategy of planning around a single process and critical path. Also, many of the potential paths that led to a project of unsatisfactory value exhibited patterns that might prompt managers to abandon the project. One of the most problematic patterns, late iterations to address lingering technical risk, occurred much later than most managers would usually consider abandoning a project. Hence, gaining greater foresight into such situations becomes critical. Meanwhile, paths that provided high value often exhibited the characteristics of frontloaded PD, with many short iterations occurring early in the project and driving out technical risk more quickly. On another front, the model provided a basis for exploring the dynamic value of individual activity modes by illuminating the popular and unpopular modes. Unlike the literature on lean, which advocates paring down a process to the bare minimum of activities, we find that the availability of contingent activity modes can lead to higher value. Thus, the best-value processes may emerge from building up activities rather than paring them down. Finally, the model reinforced the benefits of waiting until the last minute to choose the best activity mode, as this choice could vary depending on the dynamic characteristics of the project state. We make no claim that these insights are fully generalizable, yet they provide a strong motivation for further investigation. In particular, the model should be further validated in different project and industry environments, with more sophisticated processes (such as processes exhibiting greater concurrency), and with more TPMs.

Despite their prominence in the management science literature, "insight models" have limited value; decision support tools are also important [100]. Rather than being just an insight model, the APDP model supports a variety of managerial decisions on actual projects, including the choice and negotiation of realistic budgets, schedules, and technical requirements—and, crucially—where best to deploy available resources. The model supports these decisions in the contexts of dynamism, uncertainty, and ambiguity that characterize PD. Of course, since there is no way to guarantee that even a superset of *known* activities and modes will be sufficient to achieve project success, working with a process space does not provide a complete solution for project ambiguity and unforeseen uncertainty. However, since De Meyer *et al.* [30] noted that unforeseen uncertainty often arises from the unanticipated interaction of *foreseen* events, the APDP model nevertheless provides an important step forward in addressing it: by accounting for more of what is known about a project (than is typically included in a baseline plan)—*and* a larger variety of interactions among those known elements—the APDP model gives project managers a much larger portfolio of ready options for avoiding surprises and dealing with them when they occur. For example, APDP simulation can tell a project manager not only that a plan involves risk (e.g., due to a high likelihood of iteration) but also which iterative loops have the greatest possibility and how those iterations would likely affect project value. While it is true that design cycles provide a key avenue for learning, and thus PD will always include them, it is important to understand in which phases and to what extent iterations are assets and where they are liabilities. The APDP results help clarify more exactly which iterations should be encouraged or ameliorated, thereby taking the natural next step beyond the macrolevel insights of conceptual and analytical models. Finally, it is worth noting that the results of the APDP model (e.g., the high-value paths) can provide an input to conventional project scheduling techniques (c.f., [46]).

Corroborating findings at Toyota, where rigid specification paradoxically accompanied high flexibility [106], the APDP model similarly demonstrates how increased activity specification and project control can nevertheless lead to greater adaptability and higher overall project value. Organizing work into standardized packages with known inputs and EC empowers lower level managers to self-organize rapidly in the face of change. Potentially, with the correct set of rules, executives and managers may be able to *guide* the faster emergence of an optimal process without having to micromanage each state transition decision, since workers will be empowered with the information and decision policies to self-organize the work. Just as a common language gives people a platform for both quicker and more creative communication, using standard process building blocks (activity modes) as the basis for evaluation allows managers to replan and reassimilate work more quickly in a dynamic environment. After all, to manage in a context of unforeseen uncertainty, management must "build in the ability to add a set of new tasks to the decision tree" [30]. In this, the APDP model provides a basis for organizational learning and knowledge management at both the activity and process architecture levels, which can provide a basis for competitive advantage [13], [96].

Some of the APDP model's main limitations include the following. First, the basic model does not account for resource constraints. While extending the model to account for resource constraints would not be difficult, extending the decision logic of the simulation to make appropriate choices under limited resources is not as simple. When two eligible activities cannot proceed because both need the same resource, it is not clear which activity choice is best. This classic quandary has been the subject of a great deal of research.[15] Second, use of the model requires relatively more up-front planning than traditional approaches such as the critical path method. However, in this sense the model provides "a stone in the shoe for better data" [66] by prompting investigation and learning about the ambiguous aspects of a project. Third, like any model, the APDP model is not a replacement for a good project leader who maintains the global perspective of the project's (and its parent organization's) overall goals.

The APDP model presents still other opportunities for further research. While Section IV-B mentioned several such areas, those represent only a few of many interesting areas of inquiry. Beyond these, the activity mode sets, the resulting process spaces, and their potential roles in project knowledge management and organizational learning, seem worthy of further study. Second, an optimal value path for a given set of goals may not be robust to providing value across a variety of goals; hence, value could be considered even more broadly in the context of dynamic goals. This would link the model of the project to its environment and allow study of an emergent project process in a dynamic environment [122]. Similarly, can a better overall path be found without necessarily choosing the activity mode with the highest expected value at each state transition? Could doing so trap the simulation in local optima? Third, it would be interesting to explore the organizational structure that best supports an adaptive PD process. In some cases, it may be better to divide a project temporarily into two processes to achieve higher long-run performance [101]. Fourth, it is also important to understand how people would self-organize and operate within the usage of the APDP framework for collaborative design, when product performance is a rugged landscape without a single optimum design upon which to converge [56]. Fifth, the various patterns (such as iterations) that emerge in processes with different levels of value might be amenable to further classification and understanding [27].

[15]The APDP model's value-based decision logic might provide an interesting new priority rule heuristic to investigate in this regard.

ACKNOWLEDGMENT

The authors would like to thank R. Haberfellner, Q. Hommes, V. Krishnan, C. Leardi, R. Ramasesh, and two anonymous reviewers for their helpful comments on earlier versions of the manuscript, which extends two conference papers [62], [63], and part of a dissertation [64].

REFERENCES

[1] E. G. Anderson and N. R. Joglekar, "A hierarchical product development planning framework," *Prod. Oper. Manage.*, vol. 14, pp. 344–361, 2005.

[2] S. Austin, A. Baldwin, B. Li, and P. Waskett, "Integrating design in the project process," *Proc. ICE: Civil Eng.*, vol. 138, pp. 177–182, 2000.

[3] S. Austin, J. Steele, S. Macmillan, P. Kirby, and R. Spence, "Mapping the conceptual design activity of interdisciplinary teams," *Des. Stud.*, vol. 22, pp. 211–232, 2001.

[4] K. Beck, *Extreme Programming Explained*. Reading, MA: Addison-Wesley, 1999.

[5] M. J. Benner and M. L. Tushman, "Exploitation, exploration, and process management: The productivity dilemma revisited," *Acad. Manage. Rev.*, vol. 28, pp. 238–256, 2003.

[6] B. Boehm and R. Turner, *Balancing Agility and Discipline*. New York: Addison-Wesley, 2003.

[7] B. W. Boehm, "A spiral model of software development and enhancement," *IEEE Comput.*, vol. 21, no. 5, pp. 61–72, May 1988.

[8] J. L. Bower and T. M. Hout, "Fast-cycle capability for competitive power," *Harv. Bus. Rev.*, vol. 66, pp. 110–118, 1988.

[9] S. L. Brown and K. M. Eisenhardt, *Competing on the Edge*. Boston, MA: Harvard Business School Press, 1998.
[10] T. R. Browning, "Modeling and analyzing cost, schedule, and performance in complex system product development," in *TMP*. Cambridge, MA: Massachusetts Institute of Technology, 1998.
[11] T. R. Browning, "Applying the design structure matrix to system decomposition and integration problems: A review and new directions," *IEEE Trans. Eng. Manag.*, vol. 48, no. 3, pp. 292–306, Aug. 2001.
[12] T. R. Browning, J. J. Deyst, S. D. Eppinger, and D. E. Whitney, "Adding value in product development by creating information and reducing risk," *IEEE Trans. Eng. Manag.*, vol. 49, no. 4, pp. 443–458, Nov. 2002.
[13] T. R. Browning and S. D. Eppinger, "Modeling impacts of process architecture on cost and schedule risk in product development," *IEEE Trans. Eng. Manag.*, vol. 49, no. 4, pp. 428–442, Nov. 2002.
[14] T. R. Browning, "On customer value and improvement in product development processes," *Syst. Eng.*, vol. 6, pp. 49–61, 2003.
[15] T. R. Browning, "Technical risk management," in *The Risk Management Universe: A Guided Tour*, D. Hillson, Ed. London: BSI, 2006, pp. 292–320.
[16] T. R. Browning, E. Fricke, and H. Negele, "Key concepts in modeling product development processes," *Syst. Eng.*, vol. 9, pp. 104–128, 2006.
[17] T. R. Browning, "Program Architecture and Adaptation," presented at the Symp. Complex Syst. Eng., Santa Monica, CA, 2007.
[18] T. R. Browning and R. V. Ramasesh, "A survey of activity network-nased process models for managing product development projects," *Prod. Oper. Manage.*, vol. 16, pp. 217–240, 2007.
[19] Y.-T. Cheng and A. H. Van de Ven, "Learning the innovation journey: Order out of chaos?" *Org. Sci.*, vol. 7, pp. 593–614, 1996.
[20] S.-H. Cho and S. D. Eppinger, "A simulation-based process model for managing complex design projects," *IEEE Trans. Eng. Manag.*, vol. 52, no. 3, pp. 316–328, Aug. 2005.
[21] T. Y. Choi, K. J. Dooley, and M. Rungtusanatham, "Supply networks and complex adaptive systems: Control versus emergence," *J. Oper. Manage.*, vol. 19, pp. 351–366, 2001.
[22] M. J. Chung, P. Kwon, and B. T. Pentland, "Making process visible: A grammatical approach to managing design processes," *J. Mech. Des.*, vol. 124, pp. 364–374, 2002.
[23] K. B. Clark and S. C. Wheelwright, *Managing New Product and Process Development*. New York: Free Press, 1993.
[24] P. J. Clarkson and J. R. Hamilton, "'Signposting', A parameter-driven task-based model of the design process," *Res. Eng. Des.*, vol. 12, pp. 18–38, 2000.
[25] D. Clausing, *Total Quality Development*. New York: ASME Press, 1994.
[26] K. G. Cooper, "The rework cycle: Benchmarks for the project manager," *Proj. Manage. J.*, vol. 24, pp. 17–21, 1993.
[27] J. O. Coplien and N. B. Harrison, *Organizational Patterns of Agile Software Development*. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.
[28] M. R. Danesh and Y. Jin, "An agent-based decision network for concurrent engineering design," *Concurr. Eng.: Res. Appl.*, vol. 9, pp. 37–47, 2001.
[29] A. P. de Geus, "Planning as learning," *Harv. Bus. Rev.*, vol. 66, pp. 70–74, 1988.
[30] A. De Meyer, C. H. Loch, and M. T. Pich, "Managing project uncertainty: From variation to chaos," *MIT Sloan Manage. Rev.*, vol. 43, pp. 60–67, 2002.
[31] R. de Neufville, "Uncertainty Management for Engineering Systems Planning and Design," presented at the Massachusetts Inst. Technol., Eng. Syst. Div., Internal Symp., Cambridge, MA, 2004.
[32] W. E. Deming, *Out of the Crisis*. Cambridge, MA: MIT Press, 1982.
[33] K. J. Dooley, "A complex adaptive systems model of organization change," *Nonlinear Dyn., Psychol., Life Sci.*, vol. 1, pp. 69–97, 1997.
[34] D. Dougherty, "Reimagining the differentiation and integration of work for sustained product innovation," *Org. Sci.*, vol. 12, pp. 612–631, 2001.
[35] R. Dove, *Response Ability*. New York: Wiley, 2001.
[36] P. R. Duimering, B. Ran, N. Derbentseva, and C. Poile, "The effects of ambiguity on project task structure in new product development," *Knowl. Process Manage.*, vol. 13, pp. 239–251, 2006.
[37] K. M. Eisenhardt and B. N. Tabrizi, "Accelerating adaptive processes: Product innovation in the global computer industry," *Admin. Sci. Q.*, vol. 40, pp. 84–110, 1995.
[38] M. Fishbein, "Attitude and the prediction of behavior," in *Readings in Attitude Theory and Measurement*, M. Fishbein, Ed. New York: Wiley, 1967, pp. 477–492.
[39] L. Fleming and O. Sorenson, "Technology as a complex adaptive system: Evidence from patent data," *Res. Policy*, vol. 30, pp. 1019–1039, 2001.
[40] Q. W. Fleming and J. M. Koppelman, *Earned Value Project Management*, 2nd ed. Upper Darby, PA: Project Management Institute, 2000.
[41] D. N. Ford and J. D. Sterman, "Overcoming the 90% syndrome: Iteration management in concurrent development projects," *Concurrent Eng.: Res. Appl.*, vol. 11, pp. 177–186, 2003.
[42] J. R. Galbraith, *Designing Complex Organizations*. Reading, MA: Addison-Wesley, 1973.
[43] S. H. Haeckel, *Adaptive Enterprise*. Boston, MA: Harvard Business School Press, 1999.
[44] M. Harry and R. Schroeder, *Six Sigma*. New York: Currency, 1999.
[45] G. A. Hazelrigg, "A framework for decision-based engineering design," *J. Mech. Design*, vol. 120, pp. 653–658, 1998.
[46] W. S. Herroelen, "Project scheduling—Theory and practice," *Prod. Oper. Manage.*, vol. 14, pp. 413–432, 2005.
[47] J. Highsmith, *Agile Project Management*. Boston, MA: Addison-Wesley, 2004.
[48] J. A. Highsmith, *Adaptive Software Development*. New York: Dorset House, 2000.
[49] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press, 1992.
[50] A. Huchzermeier and C. H. Loch, "Project management under risk: Using the real options approach to evaluate flexibility in R&D," *Manage. Sci.*, vol. 47, pp. 85–101, 2001.
[51] IEEE, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, Institute of Electrical and Electronics Engineers Standards Association, IEEE Standard 1471–2000, Sep. 21, 2000.
[52] S. Jayanthi and K. K. Sinha, "Innovation implementation in high technology manufacturing: Chaos-theoretic empirical analysis," *J. Oper. Manage.*, vol. 16, pp. 471–494, 1998.
[53] J. Johnson, "The 'Can you trust it?' Problem of simulation science in the design of socio-technical systems," *Complexity*, vol. 6, pp. 34–40, 2001.
[54] A. Karniel and Y. Reich, "Managing dynamic new product development processes," presented at the 17th Annu. Int. Symp. INCOSE, San Diego, CA, 2007.
[55] M. Klein and C. Dellarocas, "A knowledge-based approach to handling exceptions in workflow systems," *Comput. Supported Cooperative Work*, vol. 9, pp. 399–412, 2000.
[56] M. Klein, H. Sayama, P. Faratin, and Y. Bar-Yam, "The dynamics of collaborative design: Insights from complex systems and negotiation research," *Concurrent Eng.: Res. Appl.*, vol. 11, pp. 201–209, 2003.
[57] S. J. Kline, "Innovation is not a linear process," *Res. Manage.*, vol. 28, pp. 36–45, 1985.
[58] C. Koch and I. Segev, "The role of single neurons in information processing," *Nature Neurosci.*, vol. 3, pp. 1171–1177, 2000.
[59] L. Koskela and G. A. Howell, "The underlying theory of project management is obsolete," presented at the IGLC-10, Gramado, Brazil, 2002.
[60] V. Krishnan, S. D. Eppinger, and D. E. Whitney, "A model-based framework to overlap product development activities," *Manage. Sci.*, vol. 43, pp. 437–451, 1997.
[61] K. J. Lancaster, "A new approach to consumer theory," *J. Pol. Econ.*, vol. 74, pp. 132–157, 1966.
[62] V. Lévárdy, T. R. Browning, and M. Hoppe, "Adaptive test process—An integrated modeling approach for test and design activities in the product development process," presented at the ASME Int. Design Eng. Tech. Conf. (Design Theory Methodol. Conf.), Salt Lake City, UT, 2004.
[63] V. Lévárdy and T. R. Browning, "Adaptive test process—Designing a project plan that adapts to the state of a project," presented at the 15th Annu. Int. Symp. INCOSE, Rochester, NY, 2005.
[64] V. Lévárdy, "Model-based framework for the adaptive development of engineering systems," in *Lehrstuhl für Raumfahrttechnik*. Munich, Germany: Technische Universität München, 2006.
[65] D. Levy, "Chaos theory and strategy: Theory, application, and managerial implications," *Strateg. Manage. J.*, vol. 15, pp. 167–178, 1994.
[66] J. D. C. Little, "Models and managers: The concept of a decision calculus," *Manage. Sci.*, vol. 16, pp. B466–B485, 1970.
[67] C. H. Loch and C. Terwiesch, "Rush and be wrong or wait and be late? A model of information in collaborative processes," *Prod. Oper. Manage.*, vol. 14, pp. 331–343, 2005.
[68] C. H. Loch, A. DeMeyer, and M. T. Pich, *Managing the Unknown*. New York: Wiley, 2006.
[69] A. D. MacCormack, R. Verganti, and M. Iansiti, "Developing products on "Internet Time": The anatomy of a flexible development process," *Manage. Sci.*, vol. 47, pp. 133–150, 2001.

[70] I. P. McCarthy, C. Tsinopoulos, P. Allen, and C. Rose-Anderssen, "New product development as a complex adaptive system of decisions," *J. Prod. Innov. Manage.*, vol. 23, pp. 437–456, 2006.

[71] C. D. McDaniel, "A linear systems framework for analyzing the automotive appearance design process," in *Mgmt./EE*. Cambridge, MA: MIT Press, 1996, p. 141.

[72] B. McKelvey, "Quasi-natural organization science," *Org. Sci.*, vol. 8, pp. 352–380, 1997.

[73] C. Meier, A. A. Yassine, and T. R. Browning, "Design process sequencing with competent genetic algorithms," *J. Mech. Design*, vol. 129, pp. 566–585, 2007.

[74] J. R. Meredith and S. J. Mantel, *Project Management*, 6th ed. New York: Wiley, 2006.

[75] J. Mihm, C. Loch, and A. Huchzermeier, "Problem-solving oscillations in complex engineering projects," *Manage. Sci.*, vol. 49, pp. 733–750, 2003.

[76] L. J. Moore and E. R. Clayton, *GERT Modeling and Simulation*. New York: Petrocelli/Charter, 1976.

[77] J. M. Morgan and J. K. Liker, *The Toyota Product Development System*. New York: Productivity Press, 2006.

[78] B. O'Donovan, C. Eckert, and P. J. Clarkson, "Simulating design processes to assist design process planning," presented at the ASME Int. Design Eng. Tech. Conf. (Design Theory Methodol. Conf.), Salt Lake City, UT, 2004.

[79] S. M. Osborne, "Product development cycle time characterization through modeling of process iteration," Master's thesis (Mgmt./Eng.), MIT Press, Cambridge, MA, 1993, 83 pp.

[80] G. A. Pall, *The Process-Centered Enterprise: The Power of Commitments*. New York: St. Lucie Press, 1999.

[81] J. P. Paquin, J. Couillard, and D. J. Ferrand, "Assessing and controlling the quality of a project end product: The earned quality method," *IEEE Trans. Eng. Manage.*, vol. 47, no. 1, pp. 88–97, Feb. 2000.

[82] S. D. Pathak, J. M. Day, A. Nair, W. J. Sawaya, and M. M. Kristal, "Complexity and adaptivity in supply networks: Building supply network theory using a complex adaptive systems perspective," *Decis. Sci.*, vol. 38, pp. 547–580, 2007.

[83] B. T. Pentland, "Grammatical models of organizational processes," *Org. Sci.*, vol. 6, pp. 541–556, 1995.

[84] M. T. Pich, C. H. Loch, and A. D. Meyer, "On uncertainty, ambiguity and complexity in project management," *Manage. Sci.*, vol. 48, pp. 1008–1023, 2002.

[85] N. D. Pisano. (1996). Technical performance measurement, earned value, and risk management: An integrated diagnostic tool for program management. [Online]. Available: http://www.acq.osd.mil/pm/paperpres/paperpres.html#pisano

[86] PMI, *A Guide to the Project Management Body of Knowledge*, 3rd ed. Newtown Square, PA: Project Management Institute, 2004.

[87] M. Poole, A. V. d. Ven, and K. Dooley, *Methods for Studying Organizational Change and Development*. New York: Oxford Univ. Press, 1999.

[88] M. Poppendieck and T. Poppendieck, *Lean Software Development*. Upper Saddle River, NJ: Addison-Wesley, 2003.

[89] E. S. Raymond, *The Cathedral and the Bazaar*, Revised ed. Sebastopol, CA: O'Reilly, 2001.

[90] M. Reichert, S. Rinderle, U. Kreher, and P. Dadam, "Adaptive process management with ADEPT2," presented at the 21st Int. Conf. Data Eng. (ICDE 2005), Tokyo, 2005.

[91] D. Reinertsen, "Lean thinking isn't so simple," *Electron. Design*, vol. 47, no. 10, p. 48, 1999.

[92] S. Rinderle, M. Reichert, and P. Dadam, "Flexible support of team processes by adaptive workflow systems," *Distrib. Parallel Databases*, vol. 16, pp. 91–116, 2004.

[93] S. Rinderle, M. Reichert, and P. Dadam, "Correctness criteria for dynamic changes in workflow systems—A survey," *Data Knowl. Eng.*, vol. 50, pp. 9–34, 2004.

[94] J. H. Roberts and G. L. Urban, "Modeling multiattribute utility, risk, and belief dynamics for new consumer durable brand choice," *Manage. Sci.*, vol. 34, pp. 167–185, 1988.

[95] M. J. Safoutin, "A methodology for empirical measurement of iteration in engineering design processes," Ph.D. thesis, Univ. Washington, Seattle, WA, 2003.

[96] R. Sanchez and J. T. Mahoney, "Modularity, flexibility, and knowledge management in product and organization design," *Strategic Manage. J.*, vol. 17, pp. 63–76, 1996.

[97] S. D. Sarbacker and K. Ishii, "A framework for evaluating risk in innovative product development," presented at the ASME Int. Design Eng. Tech. Conf. (Design Theory Methodol. Conf.), Sacramento, CA, 1997.

[98] R. G. Sargent, "Validation and verification of simulation models," presented at the Winter Simulation Conf., Phoenix, AZ, 1999.

[99] S. Schrader, W. M. Riggs, and R. P. Smith, "Choice over uncertainty and ambiguity in technical problem solving," *J. Eng. Technol. Manage.*, vol. 10, pp. 73–99, 1993.

[100] S. A. Shane and K. T. Ulrich, "Technological innovation, product development, and entrepreneurship in management science," *Manage. Sci.*, vol. 50, pp. 133–144, 2004.

[101] N. Siggelkow and D. A. Levinthal, "Temporarily divide to conquer: Centralized, decentralized, and reintegrated organizational approaches to exploration and adaptation," *Org. Sci.*, vol. 14, pp. 650–669, 2003.

[102] P. G. Smith, *Flexible Product Development*. San Francisco, CA: Jossey-Bass, 2007.

[103] R. P. Smith and J. A. Morrow, "Product development process modeling," *Design Stud.*, vol. 20, pp. 237–261, 1999.

[104] D. K. Sobek, A. C. Ward, and J. K. Liker, "Toyota's principles of set-based concurrent engineering," *MIT Sloan Manage. Rev.*, vol. 40, pp. 67–83, 1999.

[105] S. C. Sommer and C. H. Loch, "Selectionism and learning in projects with complexity and unforeseeable uncertainty," *Manage. Sci.*, vol. 50, pp. 1334–1347, 2004.

[106] S. Spear and H. K. Bowen, "Decoding the DNA of the toyota production system," *Harvard Bus. Rev.*, vol. 77, pp. 97–106, 1999.

[107] R. D. Stacey, *Managing the Unknowable*. San Francisco, CA: Jossey-Bass, 1992.

[108] Standish, "Extreme CHAOS," The Standish Group International, Inc., Boston, MA, Rep. 2001.

[109] M. K. Starr, "Modular production—A new concept," *Harvard Bus. Rev.*, vol. 43, pp. 131–142, 1965.

[110] I. D. Steiner, *Group Process and Productivity*. New York: Academic, 1972.

[111] R. A. Thiétart and B. Forgues, "Chaos theory and organization," *Org. Sci.*, vol. 6, pp. 19–31, 1995.

[112] S. Thomke and T. Fujimoto, "The effect of 'front-loading' problem-solving on product development performance," *J. Prod. Innov. Manage.*, vol. 17, pp. 128–142, 2000.

[113] S. Thomke and D. E. Bell, "Sequential testing in product development," *Manage. Sci.*, vol. 47, pp. 308–323, 2001.

[114] S. H. Thomke, *Experimentation Matters*. Boston, MA: Harvard Business School Press, 2003.

[115] M. L. Tushman and D. A. Nadler, "Information processing as an integrating concept in organizational design," *Acad. Manage. Rev.*, vol. 3, pp. 613–624, 1978.

[116] D. G. Ullman, D. Herling, and B. D'Ambrosio, "What to do next: using problem status to determine the course of action," *Res. Eng. Design*, vol. 9, pp. 214–227, 1997.

[117] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*, 3rd ed. New York: McGraw-Hill, 2004.

[118] D. Unger and S. D. Eppinger, "Product development process design: Planning design iterations for effective product development," MIT Center for Innovation in Product Development, Cambridge, MA, Working Paper, Nov. 2002.

[119] E. von Hippel, "Task partitioning: An innovation process variable," *Res. Policy*, vol. 19, pp. 407–418, 1990.

[120] A. Ward, J. K. Liker, J. J. Cristiano, and D. K. Sobek, "The second toyota paradox: How delaying decisions can make better cars faster," *MIT Sloan Manage. Rev.*, vol. 36, pp. 43–61, 1995.

[121] D. E. Whitney, "Designing the design process," *Res. Eng. Design*, vol. 2, pp. 3–13, 1990.

[122] T. Williams, "Assessing and moving on from the dominant project management discourse in the light of project overruns," *IEEE Trans. Eng. Manage.*, vol. 52, no. 4, pp. 497–508, Nov. 2005.

[123] M. Winter, C. Smith, P. Morris, and S. Cicmil, "Directions for future research in project management: The main findings of a UK government-funded research network," *Int J. Project Manage.*, vol. 24, pp. 638–649, 2006.

[124] J. P. Womack and D. T. Jones, *Lean Thinking*. New York: Free Press, 2003.

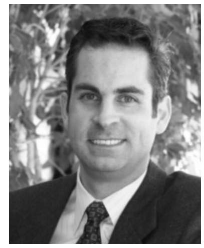
[125] D. C. Wynn, C. M. Eckert, and P. J. Clarkson, "Applied signposting: A modeling framework to support design process improvement," presented at the ASME Int. Design Eng. Tech. Conf., Philadelphia, PA, 2006.

[126] A. A. Yassine, N. Joglekar, D. Braha, S. Eppinger, and D. Whitney, "Information hiding in product development: The design churn effect," *Res. Eng. Design*, vol. 14, pp. 145–161, 2003.

Viktor Lévárdy received the M.Sc. degree in air transportation engineering from the Budapest University of Technology and Economics, Budapest, Hungary, and the Ph.D. (Dr.-Ing.) degree in systems engineering from the Technische Universität München, München, Germany.

He is currently a Senior Consultant with the 3D Systems Engineering GmbH (3DSE), München, Germany, a consulting company for product development. His research interests include systems engineering, lean and agile product development, and product strategy issues.

Dr. Lévárdy is a Member of the International Council on Systems Engineering. He is a laureate of the German Doctoral Award for Project Management (Gesellschaft fuer Projektmanagment—International Project Management Association, German Society for Project Management).



Tyson R. Browning received the B.S. degree in engineering physics from Abilene Christian University, Abilene, TX, and two Master's degrees, and the Ph.D. degree from Massachusetts Institute of Technology, Cambridge, MA.

He was a Consultant for several organizations including General Motors, Lockheed Martin, Northrop Grumman, Seagate, Southern California Edison, and the U.S. Navy. He was also with Lockheed Martin, Fort Worth, TX; the Lean Aerospace Initiative, Cambridge; Honeywell Space Systems, Irving; and Los Alamos National Laboratory, Los Alamos, NM. He is currently an Associate Professor of operations management at the Neeley School of Business, Texas Christian University, Fort Worth, TX, where he teaches operations management and project management in the MBA program and conducts research on managing complex enterprises, projects, and processes—the intersection of systems engineering and project management. His research papers have been published in various journals such as the *Journal of Mechanical Design*, IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, *Journal of Operations Management*, *Production & Operations Management*, *Project Management Journal*, and *Systems Engineering*.