

Chapter 8

Design of Complex Programs as Sociotechnical Systems

Bryan R. Moser and Ralph T. Wood

Abstract Following the introduction of systems thinking concepts in Chap. 3, we demonstrate here the treatment of complex engineering projects as sociotechnical systems in practical engineering practice. This approach, called Project Design, enables concurrent engineering (CE) teams to foresee the influence of project architecture, behaviors, dependencies, and complexity on emergent performance, thereby reducing the occurrence of unpleasant surprises. We have seen in multiple industrial cases this method as a source of new thinking and practices relevant to CE, with supporting tools and processes. Past assumptions about standard work practices may be tested, including such factors as degree of concurrency, phasing, roles, technology decomposition, system interfaces, and risk and its reduction. If embedded behaviors, in interplay with the total project architecture, lead to surprising negative or positive performance, the design of the engineering project as a sociotechnical system begins with un-learning, then awareness, and then learning of the project approaches more likely to produce positive results. The design of concurrency is specific to the nature of the social and technical elements of the system and its architecture.

Keywords Sociotechnical systems · Project design · Collaborative engineering · Simulation-based planning · Scheduling · Complexity · Teamwork · Learning

B.R. Moser (✉)
Engineering Systems Division, Massachusetts Institute of Technology,
Cambridge, MA, USA
e-mail: bry@mit.edu

B.R. Moser · R.T. Wood
Global Project Design, 75 Arlington Street Suite 500, Boston, MA, USA
e-mail: ralph.wood@gpdesign.com

8.1 Introduction

Leveraging systems thinking for the planning and performance of concurrent engineering (CE), we build project models, simulate to forecast likely outcomes, and lead cross-functional teams to explore and converge across multiple scenarios in planning work-shops. Rather than finding a single, optimal process which assumes stability and absence of real-world uncertainty of cost, schedule, and quality, we capture the characteristics of the project that will allow insights about feasibility, value, and a tradespace of likely outcomes.

We refer to this integrated approach as Project Design—leveraging sociotechnical systems thinking—relying on:

- **representation:** sociotechnical system models
- **analytics:** behavior based simulations
- **workshops:** a social process of engagement, trade-offs, and learning

Together these three enable CE teams to expose deeply embedded assumptions of expected performance, test multiples alternatives project architectures and behaviors, and more rapidly converge on feasible and optimal plans.

This chapter begins with an overview of these three components of the Project De-sign approach. Representation is described as modeling of sociotechnical characteristics include scope, teams, complexity, distribution, coordination as activity, and con-current and mutual dependence. Next, agent-based simulation of the engineering project is described which analyzes realistic outcomes including limits due to technical complexity, human priorities, capacities, interactions, and mistakes. These models and analytics are then shown as they are used in collaborative workshops. Two industrial cases are discussed, followed by a comparison with contemporary approaches and discussion of the benefits of Project Design.

8.2 Representation: Models of Engineering Projects

Our efforts to better represent the sociotechnical characteristics of product development projects began in 1995 at the University of Tokyo [1]. A model represents how a project is structured, products characterize the scope, teams are assigned and prioritize work and coordination, and progress is realized through activities. Progress is constrained not only by the capacity of teams but also by dependencies amongst activities, phases, and products. The total form of the project—including model elements and relationships—constitutes the architecture of the project.

An effective model of an engineering project as a sociotechnical system captures the essence rather than details of product, process, and organization—and especially how they interrelate. Rather than detailed decomposition of tasks, we emphasize a higher level representation of architecture and the interactions across these model elements.




 Product	<p><i>Products</i> are meaningful results of completed work. The product system is most likely the linkage to overall project value. A product is realized through activities, which represent kinds of scope and progress.</p>
 Team	<p><i>Teams</i> are one or more people who make effort to work and coordinate by applying abilities. Teams work on activities through contracts to indicate that the team has some role in that Activity.</p>
 Phase	<p><i>Phases</i> are grouped activities that represent flow of progress over time. These phases are stages of scope and progress which may stretch across multiple products yet are viewed together for governance.</p>

Fig. 8.1 Three fundamental element types cover product, process, and organization

These three project element types are grouped into breakdown structures—outcome centric product breakdown structure (PBS), teams grouped into a project’s organization breakdown structure (OBS), and the phase based groups which often form the topmost level of a work breakdown structure (WBS). Product, phases, and teams are three points of view from which the sociotechnical models are seen and evolve (Fig. 8.1).

In contrast to common Gantt and PERT charts which emphasize task-based processes, a sociotechnical model retains the distinction in aggregate between product and process, that is, between results and the flow to complete these results.

Often in industrial use of traditional scheduling the detailed WBS describes a mix of these concepts; some parts of the WBS in a master schedule are product centric while other parts are phase or team (functional) centric. In those cases the mapping of scheduled activities to the product, governing process, or project team organization can be unclear.

A project activity is an effort by teams to generate part of a product during a phase; the activity ties these other elements together (see Fig. 8.2) Activities are described with scope as a unit of progress (e.g. drawings, parts, and tests), abilities

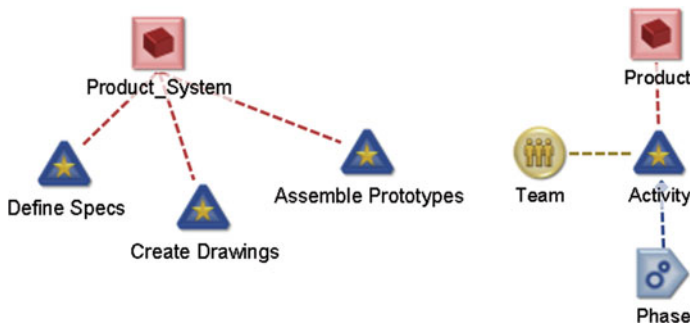


Fig. 8.2 Activity as product scope (left) and leaf of three breakdown structures (right)

required, nominal work effort, and complexity. Any work or coordination related to the activity is carried out by people on teams using resources. Multiple structures overlap and are linked by various relationships as a natural characteristic of the project, yet contrast with idealized process views.

The fact that multiple structures overlap and are linked by various relationships is a natural characteristic of the project, yet contrast with an idealized process task view. Referring to a PERT chart showing an ideal development process, Bucciarelli wrote:

To anyone interested in process, these diagrams shed very little light on how design acts are actually carried out or who is responsible for each of the tasks within various boxes. Nor is it apparent what these participants need to know, what resources they must bring to their task, and, most important, how they must work with others. The lines with arrows hardly represent the negotiation and exchange that go on within designing [2].

To better shed light on real-world activity in engineering projects, the next sections discuss our method's representation of complexity, coordination, and dependencies which improve the integrity of these models.

8.2.1 Complexity in Project Activities

What makes a project activity complex? High complexity activities tend to require more information sharing and rework. Low complexity activities tend to proceed more smoothly, with limited need for coordination and rework. While some view complexity as a purely technical characteristic, when viewing engineering as a sociotechnical system the complexity is also driven by the condition of teams. Some refer to this human derived measure as “complicatedness”, with the phrase “complex” an inherent property of the physical system. In our case, we refer to the complexity of the total sociotechnical system.

Engineering project complexity is defined here as *the cost of uncertainty reduction*—the gap between information as known by humans (tacit and transferable within a practical time horizon) and information of value to the project in the external world. A product system that is fully explored and known is no longer complex, if the knowledge is available at low cost. Complexity is therefore a function of the condition of information in technical and human forms—in current products, standards, engineering systems, and as well as the state of knowledge of the human teams. With over 15 years of industrial application, we continue to find ways to assess, sample, and inquire about the information condition of specific products, processes, and teams, allowing for calibration of complexity measures across an actual project. In turn, for effective Project Design, assessment of complexity is not necessary across all pockets of a program, rather in those which have more significance for systemic outcomes.

8.2.2 Coordination as Activity

The dynamics of contemporary engineering projects are driven by team activities beyond their own direct work, especially coordination. Brooks cites project group failures as stemming from the “man-month” conception, the notion that the number of individuals assigned to a project per month is an appropriate way of determining its dimensions. Consistent with Brooks’ man-month argument, Holt [3] states that coordination is the “greatest common denominator” of group activities and asserts that, despite its importance, coordination is an “odd category” of activity because it has no direct product, it often cannot be performed alone, and much of it is not even performed consciously. Since coordination is “odd” in these respects, it receives less than its due share of management’s attention (although it consumes a large share of organizational resources) and has not as yet obtained adequate research treatment [4].

Complex engineering projects are diverse and uncertain. Distributed teams working with little shared background across boundaries may over a long period develop their own work culture and thus approach efficiencies of a local project team. However, some increase in coordination and uncertainty due to distribution is inherent. Time zones and local holidays do not go away. Distant communication can introduce latency and noise. Shipping and travel take time and budget. In addition, it is typical for team members to simultaneously participate in other local projects. Coordination, driven by dependencies, becomes a significant portion of total activity and demanded in uncertain patterns (that are surprising) to the teams involved. Even if one wanted to invest to adjust structures and behaviors of teams into a common standard, this new standard will not likely make sense by local norms. As Schein reminds us:

Changing an organizations structures and processes is therefore difficult because it involves not only considerations of efficiency and effectiveness vis-à-vis the external task but also the reallocation of internal “property” [5]

8.2.3 Concurrent and Mutual Dependence

As used practically by schedulers in industry, and as captured in common project scheduling tools, dependence amongst activities is represented as precedence relationships; e.g. a task’s start can begin only after another task finishes. As part of creating sequence and duration of a chain of activities, each dependence becomes a sequential relationship between a milestone in one activity with a milestone in another activity [6, 7]. The most frequently used dependencies relate the sequence of start and finish milestones [e.g. Finish to start (FS)]. If a precedent constraint is not precisely the point in time after which one milestone may follow another, a lag or delay can be characterized.

We note that the mathematical relationship describing the sequence and the lag contains very little information. **The precedence representation of dependence captures only the expected schedule consequence rather than any of the other coupled characteristics of the activities.** The underlying driver of dependence—the essential meaning—is not expressed.

Our approach models dependence as a continuous and mutual demand for coordination between teams. The satisfaction of dependencies is real and ongoing activity—coordination that matches the demand for information with the supply of information through coordination by teams. Just as teams have abilities to work, they also have abilities to coordinate, should they allocate attention so. However, an ability to coordinate is not purely determined by their own state, but as a result of their position in the total project architecture with respect to other teams. Poor coordination can trigger reduced quality, exception handling, wait, and rework. As such, the cost of coordination is driven by activity complexity, tacit knowledge, information entropy, and work culture.

8.2.4 Concurrent Dependence During Mutual Progress

As an example of concurrent dependencies, consider the Gantt chart at the top of Fig. 8.3 with two activities, one for design of drawings and the other the development of a prototype. The Gantt chart contains very little information, only the

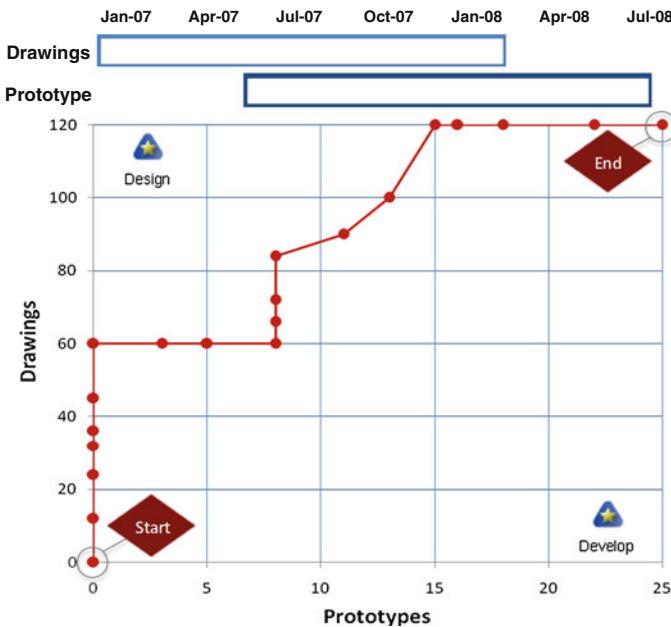


Fig. 8.3 Example mutual progress diagram

expected start and completion dates. While the overlap of the two activities on the calendar is suggestive, there are no clear insights as to any relationship between the two.

Based on the same schedule, a mutual progress diagram is shown in the lower part of the figure, with progress of Design as the upstream activity (shown on the Y axis) and progress of Develop (shown on the X axis). Rather than a shared calendar axis, we rotate the upstream activity onto a Y axis, so that we can see the mutual progress of the two activities independent of calendar. This creates a diagram which captures the mutual progress of the two activities. Units of progress—drawings and prototypes—are the measures for each axis respectively.

The expected schedule, from project start to end, appears as a red line. Each point on the line is of equal calendar time apart, in this case 1 week. Points that are closer together imply that, for that portion of the schedule, more calendar time has passed or that mutual progress has slowed. Visually one can quickly see concurrency (weeks 12–15) as the slope of the line indicates both activities making progress in the same weeks. In contrast to the Gantt chart, this diagram shows the relative and ongoing progress of the two activities. Again, even though the diagram is very suggestive (why during the first 12 weeks is there a back and forth pattern?), there is still no explicit knowledge of the dependency in the diagram.

Design information flow simulation (DiFS) was developed by Christian et al. [8] to characterize the design process as the generation and flow of information. In Christian’s approach, the dependencies are depicted as areas of ongoing constraint on progress. In Fig. 8.4 on the left is a traditional precedence dependency, sometimes referred to as a FS dependence, as depicted as a concurrent dependence: the blue area completely constrains all progress until all upstream drawings have been complete. Progress in the downstream activity—triggered by its own start milestone—must wait for the full information of the upstream design activity. Importantly, one can now see beyond precedence, that the dependence is not just the relationship between two milestones, but a relationship between all information upstream with the activity downstream.

In the middle one can see a concurrent dependency that shows a strong constraint early in the downstream progress, yet as the two activities approach completion the

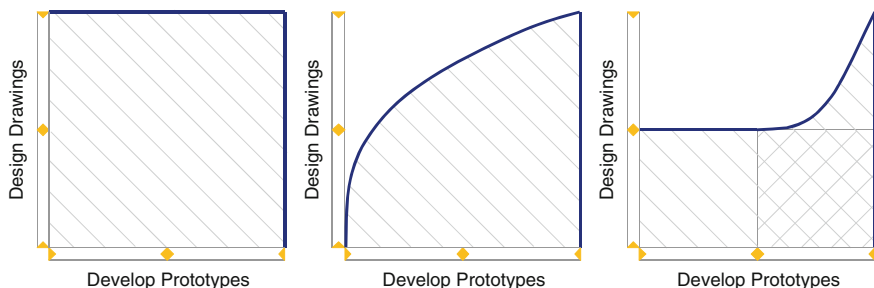


Fig. 8.4 Concurrent dependence—FS (left), early (middle), and stages (right)

constraint is softened to allow a pacing progress. In our method we refer to such a dependency shape as an “early” dependence: the activity is more dependent early on.

Using named milestones as markers of transition from one stage to another, a concurrent dependence can be divided (“carved”) into sections with different patterns of constraint. The figure on the right shows an example of a two staged concurrent dependence: until the first upstream milestones (halfway) the dependence acts as a FS for half of the downstream scope. After that point there is very little dependence until “late” in their mutual progress.

8.2.5 *Dependence as Need and Demand for Coordination*

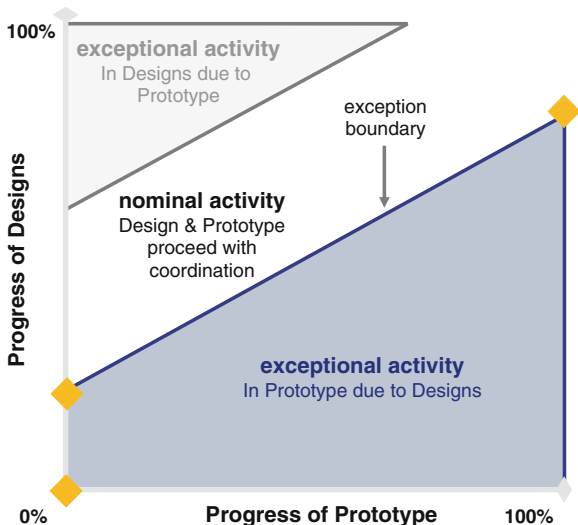
Our approach emphasizes the broad definition of *dependence as need* in contrast to precedence constraints common to existing planning methods. It is argued here that the meaning of “dependence” should go beyond a measure of sequence and be tied to an activity’s own purpose for which the dependency is relevant. Thus, in our project models, **dependence is defined as need for interaction that matters—a demand for coordination—so that an activity’s outcome is successful**. The characterization by Christian et al. [8] of dependence as need for information is extended in three ways:

- **Information, Results, and Attention:** a result, shared resource, or other item that is needed from the other activity, yet does not necessarily contain “information”.
- **Mutual Dependence:** In complex engineering projects the activities can also be mutually dependent, and therefore not clear which activity is upstream and which downstream. Progress of the two activities is tightly coupled, with need for shared information in both directions.
- **Unsatisfied dependence triggers exception:** a violation of the constraint (a dependency not being satisfied) triggers exceptional activity, which can lead to not only wait but also errors, quality activity, exception handling and propagating consequences.

Not all coordination is perfect and timely. Realistic schedules, as well as scope, cost, and risk dimensions of project plans, should reflect not only perfection, but the chance that some things change and mistakes can be made. In turn, an organization’s need to respond to oversight, decisions, change, and rework trigger new demands for coordination and work. This representation—of dependence as need—allows consideration of outcomes should the demanded coordination not be satisfied. From contingency theory, organizations follow patterns of behavior within limited rationality related to factors of organization structure and environment. Such contingency theory has been included in simulation from Cohen’s 1972 Garbage Can Model to more recent *Organizational Consultant*, a rule-based expert system [9].

Another established view from organization theory is the information and communication processing role of organizations. In the 1970s Galbraith described

Fig. 8.5 Concurrent, mutual dependency extended beyond information flow



the exception handling behavior of organizations with established channels of communication and limits in information processing capacity. Cohen et al. have applied the exception handling behaviors in an agent based simulation called Virtual Design Team [10].

Taken together, these attributes of dependence describe a continuous need for information, results, and attention due to the progress of activities, triggering exceptional activity if the constraint boundaries are violated, and able to be depicted in both directions of dependence. Figure 8.5 shows these characteristics in one diagram. If the state of progress is in an unconstrained area, and dependence up to that point has been satisfied through coordination, then the two activities at that moment are effectively independent. The dependency is represented in both directions, from Design activity to Prototype activity and from Prototype to Designs [11]. The shaded areas indicate the general zone of exceptional activity. Thus the open area—when the two tasks can operate independently—is a zone of nominal activity. Crossing the exception boundary triggers exception handling behavior.

8.3 Analytics: Forecasts of Engineering Activity

Given a meaningful model of an engineering project’s integrated product, process, and organization, we then evaluate the project through simulation. Our simulations of CE projects trace the behavior of team participants as agents. These software agents (or actors) are modeled with both work and coordination behaviors. Work behavior enables the agent to complete the skill-based scope within an agent’s own domain (i.e. task). Coordination behavior allows the agent to respond and interact with other agents.

The simulator event loop begins with each agent observing the state of the project. This awareness of the environment is itself a behavior. Based on the observation, the agent selects an action to take, attempts the act which in turn impacts both the internal state of the agent and the outside environment, the project.

Participation in the project implies that there are demands on the agent, both in response to direct contract work for a task and through dependencies with work in other tasks. While specific models characterize types of demand differently, demand to work, to communicate, and to transfer results are most common. The agent makes a selection amongst these demands based on priority and availability of its own supply. The selected demand is attempted. If the selection is work which is dependent on work, information, or resources from a different agent, additional effort may be required within the event loop to coordinate across that dependency.

A key feature of the simulation is that coordination—activity to satisfy dependencies—is explicitly analyzed. *Coordination activity is real work and must be considered if realistic schedule and cost forecasts are to be generated.* Many large complex global projects have 35–45 % of real work associated with collaboration/communication. Simulation generated forecasts include the demands, feasibility, and value of coordination to overall performance. Risk due to coordination misallocation is exposed.

Typical simulation results are shown in Fig. 8.6 including “smart” Gantt charts (the chart can be queried to determine the pattern over time of an amount of work, communication, wait, etc. that is associated with team, e.g. engineers, designers, etc.). Future uncertainty is considered, including patterns of allocation attention in an unconstrained environment as well as a Monte Carlo algorithm delivering variances on key activity characteristics and team abilities. The schedule and cost produced by the simulation show a range of most likely values for that design.

In contrast to the limited information for each task in a traditional Gantt chart, the progress and exceptions in activities and the utilization of teams are visualized—not only the start and finish of these tasks and teams (Figs. 8.7, 8.8, and 8.9).

During a workshop, teams explore how each project scenario implies the project is likely to unfold. Starting exploration from a high-level architectural view of the total project, the teams then “drill down” on particular parts of the project: a product, phase, or their own combinations of work and coordination. The pace and quality of the workshop is fostered by the team’s iterative ability to generate scenarios and rapidly explore their forecasts.

The continuous progress and utilization forecasts shown below provide a much more realistic insight into expected outcomes, stimulating dialogue. Teams are able to see the systemic impacts from day to day overlap of various kinds of activities, rather than idealized fixed duration and error-free tasks. They can also see combinations of their own activities with the activities of other teams, and how that combination impacts performance and systemic risk.

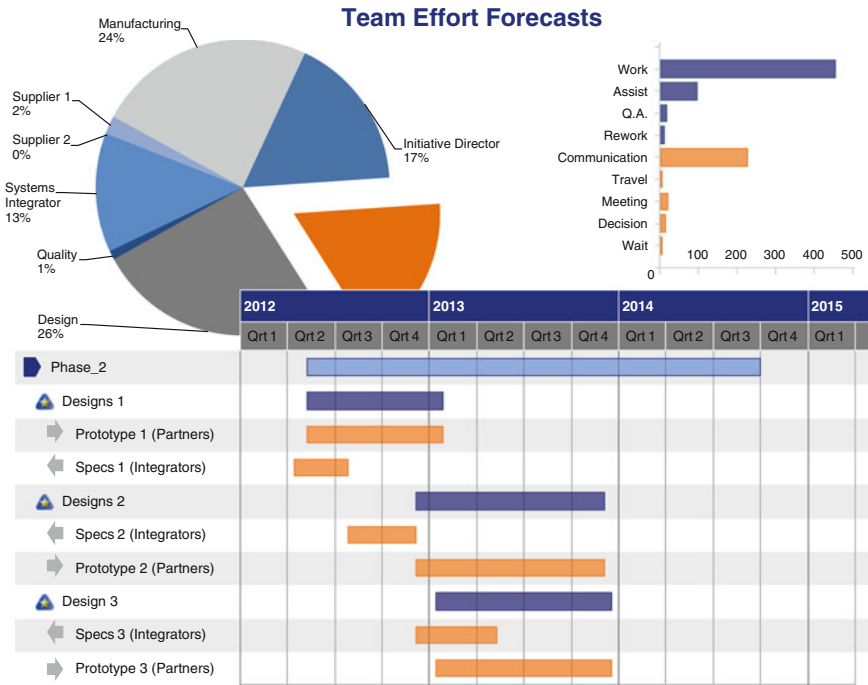


Fig. 8.6 The figure shows a typical result of a forecast. The pie chart denotes the effort in man-hours of the various teams. The pie slice representing the engineers’ effort is further queried (to the right of the pie chart) as to the distribution of activity across work, assistance, QA, rework, communications, etc. A snapshot of a *Gantt* shows one team’s work and coordination on the schedule

8.4 Workshops: Project Design in Collaborative Practice

Since teams joining a complex project bring embedded assumptions and practices, a critical need is to enable these teams to foresee the consequences of their own behaviors and in turn make adjustments. Models of integrated product, process, and organization combined with an analytic capability to forecast emergent outcomes sets up an engaging “humans in the loop” optimization process. This process is deployed in workshops for the cross functional team.

Design is an iterative and social process—the evaluation of choices and outcomes early-on, before committing to a course of action. By rapidly exploring possibilities—through dialogue, analysis, and prototyping—awareness is built and better results are achieved. As things change a good design is easily adjusted. Project Design is this forward-looking capability applied to engineering projects themselves. Much like design practices and tools revolutionized product development (e.g. 3D visualization, parametric modeling, QFD, CAD, CAE), Project Design can transform awareness, speed, and performance of teams on complex engineering projects.

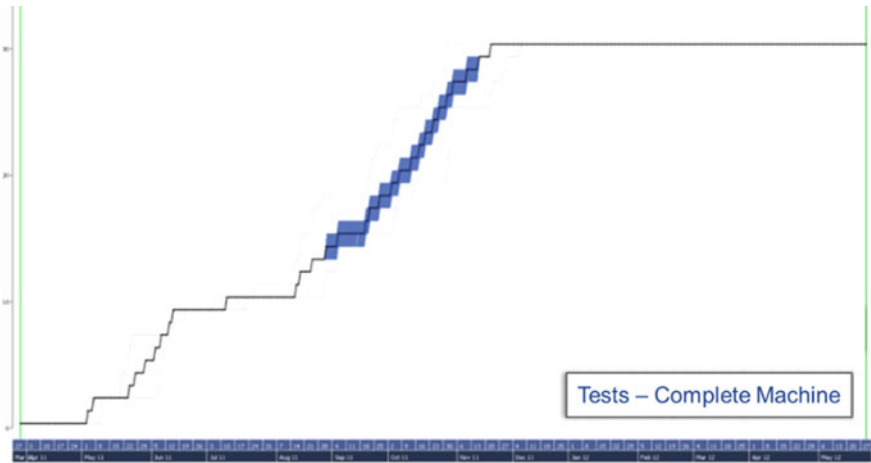


Fig. 8.7 The figure shows a **scope progress** forecast of a metric across an entire phase—in this example the tests completion for a complete machine. Other sets of tests and their most likely completion can also be compared, providing foresight into the reduction of risk based on the most critical validation. These progress timeline forecasts can be seen at various product, phase, and activity layers in the project

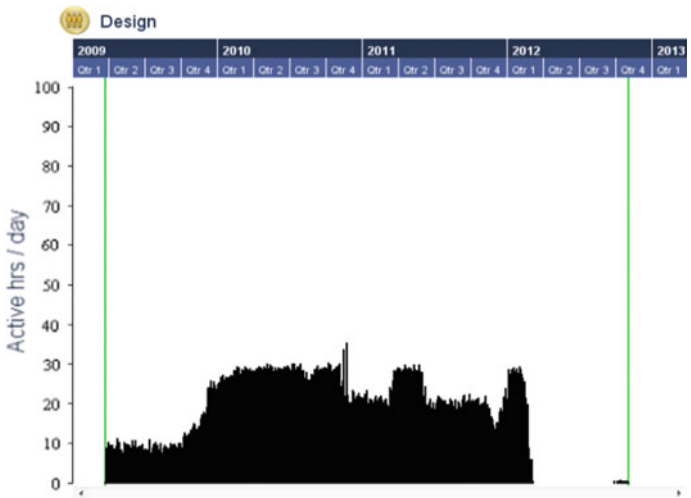


Fig. 8.8 The figure shows **total activity** for the Design team in this example project. Rather than simply an allocation assignment or capacity calendar, this forecast of expected utilization is analyzed with the interplay between various demands on this team and their skills, capacities, and priorities. Just because a team is available and assigned, does not mean it is well utilized



Fig. 8.9 The figure shows **communication** as forecast for the Design team, a subset of the total activity shown in the figure above. The demand for communication is driven by dependencies, meetings, and other factors. Still, the design team has limited capacity to respond, or perhaps other priorities. Very often, not all communication demand is satisfied, which can have delay and quality consequences

We have built a platform called *TeamPort* to support visual modeling, agent based simulation, and collaboration in Project Design (Fig. 8.10). The platform enables collaborative visual capture of the product, process, and team architectures for a proposed project. These architectures emerge similarly to traditional work breakdown structures, integrated master plans, and workflow. They differentiate themselves by a high degree of consistency and completeness that can be achieved at an agreed to and meaningful level of granularity. The platform further allows tracking of the evolution of projects, revealing the impact of any changes in key assumptions, requirements, scope, progress, uncertainty, skills, utilization, and processes.

In a workshop spanning several days, the major stakeholders and team leaders bring their skills, experience, and knowledge of the project to the table. Consistent with the fundamental practices of CE, these teams collaborate across functional boundaries to prototype their own project. The total project architecture, scope as mapped to requirements, roles and responsibilities, dependencies, and other knowledge are captured through dialogue, simulation, and iteration. The search for a desirable and feasible project plan takes time, attention, and learning. Similar to other high performance team practices, such as rehearsals in the performing arts, field exercises in the military, and practices by sports teams, the teams are iteratively exposing hidden assumptions, gaps, and converging on shared actions with situational awareness.

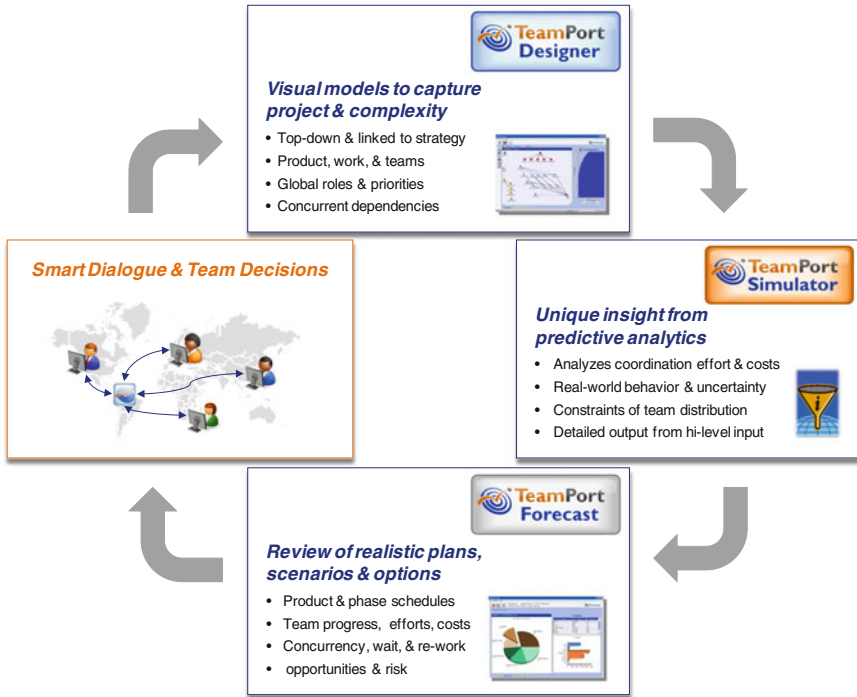


Fig. 8.10 Project design as a collaborative, iterative process

We have seen that automatic generation of an idealized schedule is not helpful if the teams are not confronted—and then agree—with the adjustments to their existing behaviors necessary to make such schedules (and cost and quality) feasible. A tension is exposed between what is hoped for and what is likely. Visual, diagnostic hints are offered during the workshops to help guide towards a Pareto “feasibility frontier”, yet the real trade-offs of cost, schedule, and scope requires a social process to parallel the analytic. In that sense, the project design workshops are “Human in the Loop” optimizations, a search for root cause and correction in response to systemic and emergent characteristics of the complex project at hand. Therefore, the pace of the iteration must match and respond to the capacity of the teams to learn and adjust.

Assumptions about standard work practices may be tested, including such factors as degree of concurrency, phasing, roles, technology decomposition, system interfaces, and risk and its reduction. If embedded behaviors, in interplay with the total project architecture, lead to surprising negative or positive performance, the design of the engineering project as a sociotechnical system leads to un-learning, then awareness, and then learning of the project approaches more likely to produce positive results. The design of concurrency (e.g., where, why, how much makes sense) is specific to the nature of the system and its architecture.

The workshops are typically led by an experienced facilitator to guide the collaborative teams through the project modeling and forecast iterations. As always, a facilitator's experiences and knowledge of the particular market and technology are helpful, so that probing questions and questions of weakness in the project's design can be translated into the domain of the participants. However, the Project Design approach requires that the project leadership and teams themselves are designing; the facilitation is meant only to provide the platform and method. We have seen over time that the simple, visual, collaborative, and rapid nature of the workshops though allows teams with no background in project management nor in this particular modeling platform to engage more than sufficiently. In fact, most participants report that the Project Design process is more intuitive and engaging than the traditional detailed or ad hoc planning approaches that are displaced.

8.5 Cases

From tens of industrial cases over the last decade, two are shown in this section. The first case is new product development in industrial machinery. The project complexity was due to the inexperience of the program manager and the global distribution of the engineering and manufacturing work. The product system itself, and overall schedule, were of average complexity and scope. In contrast, the second case, from aerospace, was complex in technology risk, product system, scope, and a very unusual teaming across companies. However, the second case was within a single region.

8.5.1 *First Case*

A recent example for a global development program for industrial equipment is shown in below. In Fig. 8.11 a portion of the completed model is shown, with scope for testing of multiple versions of a complex vehicle manufactured and tested in several countries. Figure 8.12 shows the complete model. The complete model (including all dependencies between activities and connections between team, products and activities) can be explored. Various structured views and layers for connections can be turned on and off for clarity. As a collaborative modeling experience, teams explore parts of the project connected from separate workstations then together discuss a shared view at high resolution projected on a large wall.

Based on the visual and parametric model of the project, the cross functional team generated plans for 35 various scenarios over 2 days, balancing scope, phasing, concurrency, team roles, use of critical facilities, and risk mitigation (Fig. 8.13). Across the scenarios which captured the full project scope requested by stakeholders, the teams were able to design a baseline plan with improved likely schedule for market entry by 10 months. Importantly, the design changes and

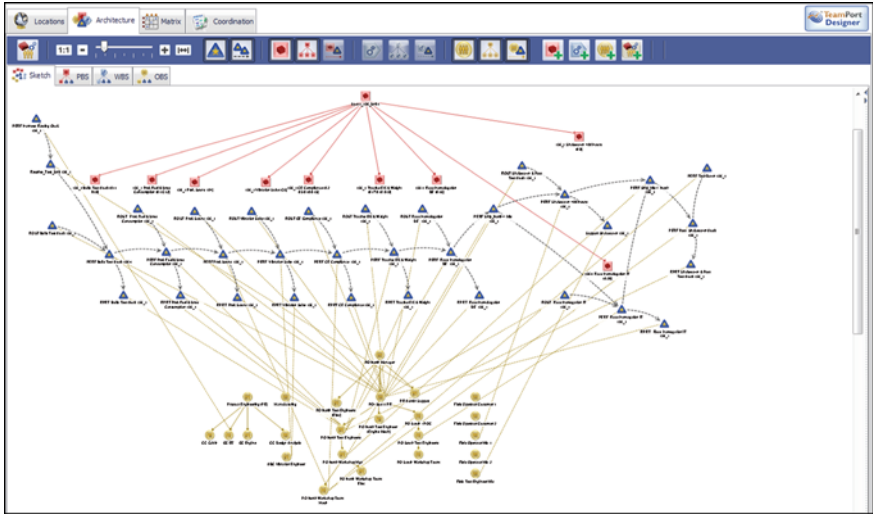


Fig. 8.11 The figure illustrates relationships at a high level between the product breakdown structure depicted by *red squares*, the work breakdown structure depicted by *blue triangles* and the organization breakdown structure depicted by *yellow circles*. Lines (*yellow*) of responsibility between and among teams, mutual and concurrent dependencies between work activities (*black dashed curves*) are shown; these can be hidden or highlighted in layers for clarity when working with the model

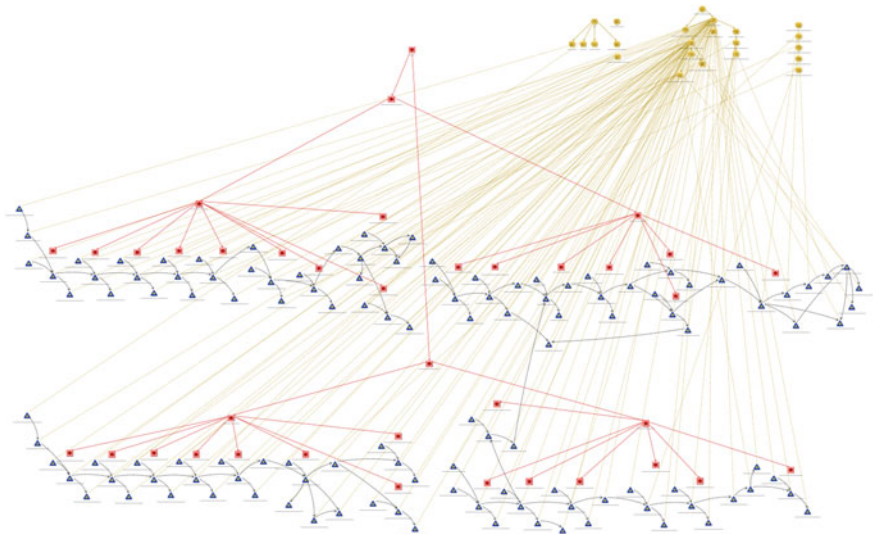


Fig. 8.12 The figure shows the entire project design model for the project described in Fig. 8.11. Projection of such a complex model onto a wall or large screen aids in working with details of the model. Additional views, as matrices, structured layouts, and lists are also generated to promote exploration and a “forest for the trees” view of project architecture

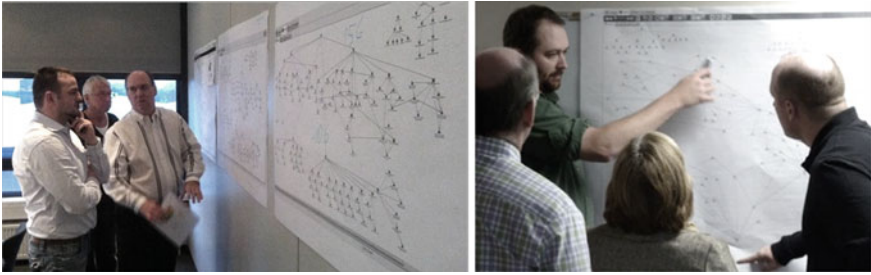


Fig. 8.13 The figure shows workshop design sessions where a printout of a project design model has been posted on the workshop wall. The model in the figure on the *left* is the one shown earlier. Whether on paper or through real-time visual interaction, the teams quickly come to form a shared mental model of their complex work and their own role in it

trade-offs to achieve this 10 month gain were proposed and made together, while balancing systemic impacts on cost and risk.

In a matter of days, the cross-functional team was able to generate a meaningful baseline plan amongst alternatives. The participants were able to easily grasp and interact with the project model with no method training, focus on portions while quickly stepping back to consider total architecture and expected outcomes. In contrast to the detailed Gantt Charts and ad hoc spreadsheets employed previously, the quality of engagement and robustness of the resulting plan were much improved.

8.5.2 *Second Case*

The next case involves two companies, A and B that formed a joint venture to develop a new product. Normally A and B are competitors. Although the venture had been running for several months before the Project Design session, we mutually decided with management to apply the full Project Design methodology ab initio. When we arrived on the scene, the program was midway through the concept development phase and had experienced cost growth from initial estimates.

In the workshop discovery process, ethnographic interviews of key program contributors revealed a stark difference in the new product development cultures of the two partners. For concept development, Company A used a separate team and a flexible, rapid prototyping and learning process compared with Company A's more exacting process for detailed design and development. On the other hand, Company B's culture was to apply its rigorous, detailed design and development process to concept development. In consequence Company A was almost always waiting for Company B to finish its activities so that their work could be integrated and tested. It also transpired that both companies, and, indeed, some groups within a company, used different project estimating methods and earned value management measures

(e.g., level of effort, percent of milestone completion, actual milestone completion). All told there were four project management tools in use across the two companies: Excel, MS Project, Primavera and SAP.

The initial Project Design model of the program, drafted by a few key players with facilitation help, did not close with the venture's current estimates of schedule and cost. The sense was that Project Design had overestimated the "Partnership Drag" or coordination effort and that the model contained incorrect assumptions about Company A, which had not chosen to participate in the initial modeling session. Both companies thought that coordination effort was already built into the reference programs that each used as the basis of estimating their parts of the current program. In view of delays waiting for Company B to complete its activities, Company A conceded that "Partnership Drag" might account for an additional 10 % of coordination time that was not included in the original estimate. As the program execution unfolded, it also came to light that Company A had adopted an inappropriate reference program for its estimating process.

Most importantly, during initial modeling, we came to realize that the original program plan was success oriented, that is, there were no provisions for risk mitigation included. About 15 % of the cost growth was attributed to "discoveries and new knowledge," a byproduct of unforeseen uncertainties and emergent behavior.

Company A's culture for conceptual design could be characterized as "hands-off elegance," which translates as hubris. Its venture management finally agreed to meet with the Project Design facilitators to straighten out the faulty assumptions in the program design model that evolved as the product of a three-day workshop (without the participation of Company A). After this was accomplished the venture management team decreed that there would be an additional, fourth day to the workshop with mandatory attendance of key contributors from both companies. The fourth day of the workshop produced tens of feasible plan forecasts, a subset of which came close to the desired parameters. Several options were identified, which needed further study, for shortening the schedule and reducing the cost of the program.

In another example of reversing a poor cultural habit, the Project Design methodology brought new understanding about the program to the several members of the program team who were not part of the original planning and were, in fact, looked upon as cogs in the program's "machinery." This improvement occurred as part of the social process associated with the design workshop.

8.6 Comparison with Contemporary Approaches

Along with a unique analytic treatment of coordination activity, the Project Design concept is built on our field experiences with ideas pulled from different lines of contemporary thinking, practices, and tools. We focused on improvements in the areas of representation, analytics of prediction, and social process. Why are these improvements necessary? Because the established body of knowledge and

standards of program management, although satisfactory for stable environments, aren't reliable in today's environments that are expanding in both technical and social complexity in ways that exhibit emergent behavior. Today's program teams need to be able to anticipate and react quickly to surprises that grow from seemingly insignificant anomalies.

Better representation of interdependencies among elements in the program architecture together with analytics for scenario evaluation are two key improvement areas. In programs encumbered with "natural" variation and other uncertainties that are foreseeable but likely ignored, as well as with the unforeseeable consequences of the choices that teams have to make, the probability of surprising outcomes is high. These factors form the technical dimension of program management. How teams make choices, or which demands teams choose to satisfy or ignore, is conditioned by deeply-seated habits and culture. To tackle this dimension, we seek to improve the social process of program management; we call this social process "Human-in-the-Loop Optimization." Project Design is above all a learning process that continues to occupy our research.

The architectural representation of a program as the three elements of product, process and organization is imbedded in the Quality Function Deployment methodology that started to be used in the 1980s in the United States; the three-part representation was also a feature of the master database, called the PPO, that was one foundation of DARPA's initiative in concurrent engineering (DICE) conducted in the late 1980s to early 1990s by a team of industries and universities. The Project Design improvement, in this case, has been to employ a graphical engine to simplify building and manipulating the architectural representation and to make visible the interdependencies and dependencies among its parts. Through our notion of scope, Project Design offers a convenient way of capturing and displaying interdependencies and graphically tracking progress in resolving them. These and other reports create situational awareness among the performing teams.

The analytics in Project Design are associated with an agent behavior based engine that runs a parallel large number of "demand-activity-supply" loops, perturbed to emulate variation for each program design configuration. In essence, each loop represents a collision between demand and supply that is mitigated with repeated iterations. Within the simulation engine, each such loop accounts for demand work, rework, and their complexity, coordination demand and constraint, roles and responsibility, and availability and skill level of resources. Because of its approaches to representation and simulation, Project Design sidesteps the traditional project management network methodologies such as program evaluation and review technique (PERT), critical path method (CPM) and conditional diagramming method (CDM). It is not necessary to identify either the critical path or the critical chain, which Goldratt persuasively argues is a more important measure of program schedule than the critical path [12]. Instead, Project Design automatically identifies the program activities and interdependencies that are currently the largest levers on overall program performance and, therefore, should attract the greatest attention of the program teams. As an input option to the Project Design system,

one can specify to ignore (or assume) coordination activity; the resulting simulation then approximates the CPM solution.

The social process of Project Design is presently still evolving along with our ethnographic and analytic research of each workshop event and surrounding cultural experiences. In Chap. 3 we discussed the socio-cultural model of an organization and we recounted Gharajedaghi's [13] notion of culture as operating system. A motivating question, recently put to us by a physicist, is, "Can culture be modeled?" We humbly believe that the answer is affirmative, although the confirmation will take more work and will be a topic for another day. Moreover we assert that traditional models of work indeed represent cultural characteristics, though hidden as embedded assumptions and simplifications.

For now we are concerned about the social and cultural interactions, within and between teams that attend the collaborations necessary to resolve interdependencies. For example, in researching collaboration Schrage has recognized that managing relationships is more important than managing individuals [14]. Although Schrage's thesis of "No More Teams" finds that teams are neither required nor necessarily desirable for collaboration in today's environment, research on high-performance teaming, some of it dating to World War II, has long supported that the best teams strike a balance between task orientation and attention to relationships. Whether in a team setting or not, collaboration is unlikely to occur between two experts who will not share knowledge with one another because of a soured relationship. It is up to the Project Design workshop facilitator to establish norms of behavior and ground rules that will build relationships. We also note that Project Design workshop participants are often leaders or representatives from performing teams and may not be members of the same team. It is, therefore, all the more important that the Project Design workshop sets and demonstrates the tone for the culture of the entire program.

Schrage further argues, with several examples, that the social process of collaboration is considerably improved by the use of collaboration tools. We identify Project Design workshops as our fundamental collaboration tool, because the workshop gathers together contributors from many corners of an organization (or organizations) and encourages them to create productive relationships (collaborations) to design a feasible program. The software support system of Project Design is another collaboration tool; since it rapidly executes simulations of different program design scenarios, it provides the workshop participants with near-instantaneous feedback on the feasibility of their designs. This feature brings three benefits: (1) collaboration can proceed at the speed of human conversation; (2) collaborators can learn rapidly about the behavior of their system (program) design choices and master their optimization; and (3) collaborators can feel a sense of significant accomplishment for a relatively small investment of their time. In principle, given the architectural representation and simulation analytics of Project Design, the technology exists to automate the optimization of a program's design. However, automating the optimization would vitiate the relationship building, deep learning and feelings of accomplishment and ownership that derive from the current, human-in-the-loop social process of Project Design.

In 1997 the National Center for Manufacturing Sciences (NCMS) initiated a benchmarking investigation of global companies to discover exemplary product development processes. After the benchmarking team was exposed to Toyota, it stopped looking at other companies and redirected the investigation to study Toyota's product development process at a deeper level. The results of this study were documented by Michael Kennedy in the book *Product Development in the Lean Enterprise* [15]. Toyota had created a knowledge-based development environment (culture) that rested on the knowledge of individual workers: their understanding of needs, information availability, responsibility and teaming interaction. In this knowledge based environment, the system architecture emerges from the interaction of all functional perspectives. Interaction occurs through natural communication and through integrating events where team decisions about next actions are taken. The parallels with the learning, interaction (collaboration) and integrating events (workshops) of Project Design are evident.

8.7 Benefits

Engineering project plans created through a Project Design process are generated more quickly with increased accuracy by incorporating realistic drivers of feasibility. Although Project Design can handle a large amount of detail, teams soon discover that an elevated level of abstraction provides better and faster learning about essential program and system knowledge than a very detailed plan. Besides work demand and ability variation, Project Design allows teams to account for foreseen uncertainties by building risk mitigation activities into the program's process structure. When program teams are surprised by an unforeseen or emergent uncertainty, they can rapidly incorporate learning loops and recovery actions into the program design and assess the extent of any setback on overall schedule and cost by running simulations.

We adopt the idea of culture as operating system and help the teams to build a common culture in the process of designing and executing the program. This step is essential, because most global teams start with their individual cultures and, therefore, are a recipe for social complexity [16]. Project Design begins with ethnographic interviews, following a tested format, of a number of key team members. From the interview, facilitators identify cultural issues or behaviors that may need to be dissolved or replaced, either during the design of the program or during its execution. The data from each new workshop feeds ongoing research to improve the social process of Project Design.

Another benefit is found in the adaptability to either external or internal changes. If a customer orders a change, if something in the external or internal environment changes in a way that invalidates an assumption, or if a risk mitigation activity fails, to name three circumstances, the design of the project can be modified rapidly, enabling the teams to re-design the program with accuracy and awareness [17].

8.8 Conclusion

Project Design is a platform for effective planning and ongoing dialogue of teams on modern, complex projects. These projects often have no feasible central planner with complete awareness of work practices, background knowledge, and priorities of dispersed teams. Even if a central planner or automated scheduler possessed the information required for a good plan, there remains need for a social process during which teams take time to propose, negotiate, prototype, iterate, and learn. If deeply embedded assumptions—hidden and effective in past projects—are to be exposed, the project design process must predict unexpected outcomes at a pace of learning by teams. In order for teams to deviate from their existing work culture, unexpected impacts from misallocated, poorly timed, or missed coordination need to be confronted as surprises.

A very good workshop session often begins with forecasts that cause great concern. In this way the *Project Design* process enables teams who might otherwise not share work culture to adjust, develop shared awareness, and converge towards a common, feasible, and optimal plan. In the cases where disparate behaviors and abilities lead to challenges that cannot be overcome within the horizon of the project, the project architecture can be otherwise designed to mitigate potential negative impacts.

Stability of experience in a learning environment can turn complex activities into simple ones as behaviors of the system are understood—people develop shared work practices and build up tacit knowledge. Learning, as the generation and transfer of information across dependencies, drives transformation from complexity to simplicity.

In contrast, a traditional master planning or automatic scheduling method is not likely to lead teams to explore embedded local behaviors which drive misallocated, poorly timed, or missed coordination. In this way the *Project Design* process enables teams who might otherwise not share background to develop awareness, adjust, and converge towards a common, feasible, and optimal plan. In the cases where disparate behaviors and abilities lead to challenges that cannot be overcome within the horizon of the project, the architecture can be designed to mitigate potential negative impacts.

Case studies over the last decade show that small architectural changes may lead to surprising outcomes as projects become more complex. From each team's perspective, the way that the integrated architecture generates demand for work and coordination may appear in combinations inconsistent with the team's existing work culture. The design of a project may unknowingly disrupt the potential of embedded practices, abilities, and knowledge. If a team's work culture acts as an organizing driver to decrease uncertainty in the integrated sociotechnical system over time, a surprising sudden shift in various demands and costs of coordination will increase uncertainty. In complex projects a small change to alignment of the team's abilities (supply) to the need for work and coordination (demands) can lead these very same embedded practices to be wasted or, moreover, trigger unexpected delay, poor

quality, and propagating rework. A team unaware of these impacts—following their own best judgment—may in fact be a cause of systemic poor performance. In these cases, given the counterintuitive root cause of these difficulties, teams in frustration may harden their belief, instead assuming that the cause of difficulty must be the behaviors of other teams.

Project Design is part of a new generation of methods that seek to model real-world dynamics of project work, provide early, architectural views of the project as a sociotechnical system, and allow forecasting of the range of likely outcomes in cost, schedule, and quality. These methods—rather than displacing people during the planning process through automation—leveraging interactive visualization and collaboration technologies to involve people in exploring the range of structures and behaviors. The design and optimization of the integrated project as sociotechnical system includes the awareness and commitments of the people who will perform together.

References

1. Moser B, Mori K, Suzuki H, Kimura F (1997) Global product development based on activity models with coordination distance features. In: Proceedings of the 29th international seminar on manufacturing systems, Osaka, pp 161–166
2. Bucciarelli L (1994) Designing engineers. The MIT Press, Cambridge
3. Holt A (1989) Organizing computer use in the context of networks. In: COMPCON Spring'89. Thirty-fourth IEEE computer society international conference: intellectual leverage, digest of papers, pp 201–207
4. Oravec J (1996) Virtual individuals, virtual groups, vol 11. Cambridge University Press, Cambridge
5. Schein E (2006) Organizational culture and leadership, vol 356. Wiley, Hoboken
6. Kerzner H (2013) Project management: a systems approach to planning, scheduling, and controlling. Wiley, Hoboken
7. Mantel S, Meredith J, Shafer S, Sutton M (2011) Project management in practice, 4th edn. Wiley, Hoboken
8. Christian A, Grasso K, Seering W (1996) Validation studies of an information-flow model of design. In: Proceedings of the 1996 ASME design engineering technical conferences
9. Baligh H, Burton R, Obel B (1996) Organizational consultant: creating a useable theory for organizational design. *Manage Sci* 42(12):1648–1662
10. Jin Y, Levitt R (1996) The virtual design team: a computational model of project organizations. *Comput Math Organ Theor* 2(3):171–195
11. Ni M, Luh P, Moser B (2008) An optimization-based approach for design project scheduling. *IEEE Trans Autom Sci Eng* 5(3):394–406
12. Goldratt EM (1997) Critical chain. North River Press, Great Barrington
13. Gharajedaghi J (2011) Systems thinking: managing chaos and complexity, 3rd edn. Elsevier, Burlington
14. Schrage M (1995) No more teams: mastering the dynamics of creative collaboration. Currency Doubleday, New York
15. Kennedy M, Ward A (2003) Product development for the lean enterprise. Oaklea Press, Richmond

16. Kahane A (2004) Solving tough problems: an open way of talking, listening, and creating new realities. Berrett-Koehler, San Francisco
17. Moser BR, Wood RT, Hiekata K (2014) Risk management in the design of engineering as sociotechnical systems. In: Cha J et al (eds) Moving integrated product development to service clouds in global economy. Proceedings of the 21st ISPE Inc international conference on concurrent engineering, IOS Press, Amsterdam, pp 635–646