Using Product Development Task Networks to Measure Organizational Stress: A Longitudinal Study

Shawn T. Collins[1*], Joe A. Bradley[2], Ali A. Yassine[2]

[1] Control Systems Engineering Department, Rolls-Royce, Indianapolis, IN 46206

[2] Product Development Research Laboratory, Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801

Index terms: Product Development, Information Flow, Design Structure Matrix, Network Analysis, Centrality, Brokerage, and Key Player Measures, Concurrent Task Execution, Organizational Change.

Abstract

This paper uses Network Analysis (NA) to study task interactions in a product development process (PDP). It explores the impact of significant organizational stress on changing information flow patterns in the PDP at a small engineering company. The analysis identifies structured patterns of centralization, role specialization, and formalized control in the PDP. This validates themes from organizational behavior and quality management literature regarding how organizations change in response to stress. These trends are difficult to uncover with Systems Engineering techniques such as the Design Structure Matrix (DSM) and N^2 method. The results offer several findings. First, reducing variation in task outputs is an understandable approach to controlling a PDP. However, it is important to reduce variation in task inputs as well. Second, tasks, like people, can have varying roles and burdens in terms of how they share information. Companies that want to support multiple concurrent projects must make sure their PDP tasks have an appropriate distribution of labor. Finally, an NA metric called Simmelian ties can identify both valuable and ineffective iteration among groups of tasks. This provides a metric of effective concurrency in a PDP.

---

*Corresponding author. E-mail shawn.collins@alumni.purdue.edu

This paper is based on material presented in two prior conference papers [11, 35] and an unpublished PhD dissertation [36].

1. Introduction

The product development process (PDP) is a complex system composed of an integrated set of tasks that collectively accomplish a defined objective – e.g. developing a new product [1, 2]. Existing Systems Engineering tools can improve our understanding and analysis of these kinds of systems. Graph-based techniques like CPM / PERT [3][1], IDEF [4][2], and flowcharts, in addition matrix-based techniques such as the Design Structure Matrix (DSM) and the N^2 method were successful in capturing the relationships between the tasks in a PDP and in scheduling these tasks accordingly [5-7]. However, almost all these approaches lack the sophistication to uncover the underlying statistical properties of the PDP and its constituent tasks. Furthermore, as the number of tasks or relationships between tasks increases, the corresponding complexity makes these systems increasingly difficult to analyze. As such, standard System Engineering tools provide only a partial view of the development processes they analyze.

We can enhance our understanding of these systems by using established network analysis (NA) techniques. These methods help uncover previously unnoticed trends and properties in the PDP. For example, it is possible to look at patterns of the overall PDP structure and the location of individual tasks within this structure. This is an important analytical shift from viewing a task as having individually-determined characteristics (as is the case in the above mentioned Systems Engineering tools) to viewing it as representing emergent properties - emerging from patterns of information flow between tasks.

Using NA techniques to measure properties of information flow provides a methodology to identify critical product development tasks and interactions that constrain PDP execution. Project managers can use these data to structure their team integration mechanisms based on the information sharing characteristics of their PDP [8, 9]. They can also use these data to identify coordinating mechanisms for groups of concurrently scheduled PDP tasks [10]. Functional managers and process architects can use these data to identify critical or overloaded tasks [11]. They can also evaluate whether tasks defined as critical in the Systems Engineering literature, such as stage gates, exercise significant control over information flow in their specific PDP [12]. This new Systems Engineering approach provides a decision support tool for managers who

---

[1] Critical Path Method / Program Evaluation & Review Technique.

[2] Integrated DEFinition (IDEF) is a family of methods designed to model the decisions, actions, and activities of a system.

must alter ideal PDP sequences due to specific schedule, budget, and expertise constraints encountered on their projects.

This paper conducts a longitudinal study of a small engineering company (Smallcomp) at two points in time. It examines Smallcomp's response to the stress caused by a significant industry downturn, workforce reduction, change in market focus, and increase in the number of products being developed. Specifically, it explores whether NA metrics identify several common findings cited in the organizational change and quality management literature. First, organizations undergoing significant stress tend to respond by centralizing control of their processes and decision making [13, 14]. Second, organizations undergoing significant stress tend to respond by increasing the formal control of their activities [15, 16]. Third a social system that grows in size tends to increase the role of specialists in its distribution of labor [17, 18]. Finally, effective quality management requires controlling variation of both the inputs and outputs for a given process [19].

The rest of the paper proceeds as follows. We begin with a methods overview of NA, followed by background information about Smallcomp. We then use NA metrics to ask two questions. First, what do we learn about Smallcomp's response to stress by using NA to compare its PDP network properties in 2002 and 2006? Second, are the themes from organizational behavior and quality management literature identifiable in the structured patterns of task information flow from Smallcomp's PDP? We close by applying these findings to the general areas of managing PDP execution and team performance.

2. Network Analysis Overview

The basic element of Network Analysis (NA) is a data matrix that defines relationships between each row and column element. For a matrix of product development tasks collected in the left half of Figure 1, the matrix indicates the input / output flow of information between each task. The right half of Figure 1 shows the network drawn from the data matrix. The arrows indicate a dependency relationship, or tie from the row task to the column task. For example, task A receives output from task D, while task B provides output to tasks C, F, and G.

NA methods contain a wide variety of analytic techniques once data are collected in matrix form [20-23] that have been used to study topics as diverse as world systems [24, 25], diffusion of innovation [26, 27], production of scientific knowledge [28], and organizational influence [29, 30]. These methods target three different levels of analysis [31]. First, it is possible to look at patterns of the overall structure and the location of individuals within the structure. Second, it is possible to identify substructures (groups of nodes that are closer to each other than to other groups) based on patterns of relations. Third, it is possible to look at the positions of individuals within the structure. This is an analytical shift from viewing a node as having individually unique characteristics to viewing it as representing some relational attribute. Network position analysis looks at attributes emerging from patterns of ties between nodes. In Systems Engineering terms, NA does not define criticality based on individual task attributes like how long they take or how much they cost to complete. Rather, NA defines criticality based on different aspects of information flow among tasks in a given PDP.

Insert Figure 1: Sample Task Network Matrix and Map about here.

We define the key NA terms [21, 32] used in the rest of this paper below.

- **Node:** Any single element of the data matrix. This would be tasks A through G in Figure 1.
- **Link:** Any tie between two nodes in a network. This is a non-zero value in the Figure 1, which indicates information flow between two tasks.
- **Indegree Centrality:** The number of links entering a node. This is the number of tasks providing input to a DSM task.
- **Outdegree Centrality:** The number of links exiting a node. This is the number of tasks receiving output from a DSM task.
- **Cluster:** Divide a network into *n* specified separate groups that maximizes the density of ties within a group, and minimizes the density of ties to other groups. This identifies which concurrent tasks in a PDP can be allocated to separate phases (e.g. Concept, Preliminary Design, etc).
- **Density:** For a given network, the ratio of the actual number of links divided by the maximum number of possible links. This describes the concurrency within a cluster, as well as the strength of coupling between phases.
- **Brokerage:** For three nodes B, E, and G in Figure 1, B has three potential brokerage roles to pass information between nodes E and G. B is a *coordinator* if all three nodes are in the

same PDP phase. B is a *gatekeeper* if E is in a different phase from B and G. B is a *representative* if G is in a different phase from E and B.

- **Key Fragment Node:** Identify a set of *n* nodes which, if removed, would maximally disrupt communication among the remaining nodes. These are PDP tasks which, if omitted or done incorrectly, have greatest potential to cause project rework from incorrect information.
- **Key Reach Node:** Identify the set of *n* nodes that is maximally connected to all other nodes. These PDP tasks are the most efficient points to introduce new information to the product development activities.
- **Transitivity:** Nodes B, E, and G in Figure 1 are transitive because E provides output to B, B provides output to G, and E also provides output to G.
- **Simmelian Ties:** A simmelian tie is reciprocal and part of a transitive triple. The reciprocal tie between nodes F and G is simmelian because both F and G have ties to node D.
- **Path:** Any route between two nodes. In Figure 1, the paths from E to F are E-B-G-F, E-B-F, and E-G-F.
- **Geodesic Distance:** Shortest path between two nodes. The geodesic distance between nodes E and F is two links, where either B or G is an intermediate node. In a PDP, this is the fewest intermediate tasks that transmit information between two given tasks.
- **Individual Node Analysis:** Identify key tasks based on their relational roles as information transmitters in the PDP.
- **Node Group Analysis:** Identify and measure groups of tasks that interact more with each other than with other tasks in the PDP.
- **Overall Network Analysis:** Identify trends from treating the entire PDP as a single entity of tasks that share information.

3. Company Overview of Smallcomp

The data for this paper were collected from the product development process at Smallcomp. Smallcomp formalized a stage-gate system [33, 34] for its product development activity during an initiative to expand its product line from demonstration units to increased volume. The stage-gate system contains a series of business and technical reviews held after each phase of product development to ensure that the maturing technical design still matches the expected market characteristics. Concurrent to this formalization, Smallcomp began investigating matrix based project management tools during a quality improvement initiative to streamline its product development task execution. This resulted in their PDP being captured and evaluated using a traditional time-based Design Structure Matrix in 2002 [12, 35]. Shortly after this formalization, Smallcomp cancelled two high profile programs, lost contract bids with major customers, and significantly reduced its workforce. After the resulting downturn, Smallcomp changed its market focus, expanded the number of programs from 1-2 at a time to 4-5 at a time. Smallcomp's process architects regularly updated its PDP to add tasks or change the information input / output flow defined between each task. Their goal was to avoid the patterns exhibited before the 2002 downturn. The PDP was re-captured in Design Structure Matrix format for a graduate school project in 2006 supervised by the first author [11]. This paper conducts a longitudinal analysis of Smallcomp's PDP at these two points in time using the NA metrics from the previous section. It examines whether the themes from organizational behavior and quality management literature are identifiable in structured patterns of information flow from Smallcomp's PDP in 2002 and 2006.

Insert Table 1: Smallcomp Overall PDP Network Comparison about here.

4. Longitudinal Network Analysis of Smallcomp's PDP

4.1. Overall PDP Structure

Table 1 compares overall network measures of Smallcomp's PDP between 2002 and 2006. The primary change is toward increased centralization and reduced use of intermediate tasks to transmit information. This is shown by the drop in longest path distance from 8 to 6, by the increased number of paths of length 2 or less, and by the decreased number of paths of length 5 or 6. The 21% increase in the number of links between tasks also indicates increased centralization.

Insert Figure 2: Brokerage Analysis Comparison across PDP Phases about here.

Other network parameters explore the nature of this centralization on the structure of overall communication flow and information transfer within the PDP. Brokerage analysis considers information sharing load within and between PDP phases. Figure 2 shows the number of tasks in each PDP phase that exceeded a relative brokerage value of 3. In both cases the high coordination is expected because this is when the concept is taking shape. It decreases as the PDP progresses into later stages. For a project manager, the value of this information is in highlighting critical brokerage tasks during a given phase, as well as tasks that are important for managing interfaces between product development (PD) phases. In 2006, there is increased coordination within the Concept, Mechanical Integration and Detailed Design phases. This shows more iteration within the particular design before passing information to the next phase. The increased representation from the Concept phase also shows an emphasis on controlling the information leaving the Concept phase. The increased gatekeeping in Functional Integration and Detailed Design show a focus on controlling information entering these two phases.

Insert Figure 3: 2002 Outdegree Centrality Control Chart about here.

Insert Figure 4: 2006 Outdegree Centrality Control Chart about here.

Figure 3 and Figure 4 use outdegree centrality to measure task information sharing load based on the number of tasks it provides output to. The axis progression is from beginning to end of the project execution. The upper and lower control limits (UCL, LCL) are drawn based on the two-sigma point around the average value. There is a slight increase in outdegree centrality between 2002 and 2006. However, there are two significant changes. First, the early concept tasks have moved significantly above the UCL. This is consistent with Figure 2, which showed the increased coordination brokerage in the System Definition phase. The change reflected here is that tasks during the System Definition phase are being required to share their output with more tasks than in 2002. This would result in an increased team coordination burden. Second, the pattern in 2002 is largely cyclical from left to right. By contrast, the pattern in 2006 shows a reduction from left to right. This shows tighter control and reduced variation on the task outputs.

Insert Figure 5: 2002 Indegree Centrality Control Chart about here.

Insert Figure 6: 2006 Indegree Centrality Control Chart about here.

Figure 5 and Figure 6 use indegree centrality to measure task information sharing load based on the number of tasks it receives input from. The axis progression is from beginning to end of the project execution. The control limits are drawn based on the two-sigma point around the average value. There is a slight increase in average indegree centrality between 2002 and 2006. However, the changes in outdegree centrality do not reappear. The peak value for indegree centrality is still around the end of the Concept phase. The pattern remains cyclical through the rest of the PDP. Unlike outdegree centrality, where the task output load qualitatively changed in 2006, the task input load maintains the same cyclical pattern in both years.

4.2. PDP Substructures

At the substructure level, NA looks for patterns of information flow among groups of tasks within the PDP. Table 2 defines five task clusters, the density of each cluster in 2006, and the change in density from 2002 in italics. For example, the Concept Development phase density in 2006 was 0.421, with an increase of 0.087 from 2002. The density changes in Table 2 show that the 2006 PDP has increased modularity compared to 2002. This is shown by the high diagonal (within-phase information flow) densities compared to off-diagonal (between-phase information flow) densities. Most of the feedback (densities below the diagonal) values decreased, while several feed-forward (densities above the diagonal) values increased. This shows an increased focus on coupling information to downstream PD phases, and an attempt to reduce the intentional feedback to previous phases. These changes show Smallcomp’s PDP becoming less iterative and more sequential. Within this general shift, the increased densities show more iteration during the Concept, Mechanical Integration, and Validation phases. The decreased densities show less iteration during the Functional Integration and Detailed Design phases.

Insert Table 2: PD Clusters and Density Changes about here.

The density changes are consistent with the brokerage analysis in Figure 2. The increased within-phase density is reflected by the increased coordination roles. The increased feed-forward density is reflected by the increased gatekeeping and representative roles.

For Simmelian ties, tasks A and B must share information with each other, and then provide output to a common task C. This metric identifies small groups of tasks that are highly iterative and must be coordinated. In 2002, the Simmelian ties in Table 3 show a very strong focus on iterating tasks that involve requirements. The group containing P110 (requirements), P120B (design points), and P130 (concept review) deals with coordinating the overall project approach. The group containing P120C (electrical architecture), P120E (define assemblies), and P120F (mechanical architecture) deals with coordinating design of the electrical and mechanical architectures. There are two important points of divergence in 2006. The first is the new tasks in the Simmelian groups. The first three groups of tasks are the same for both years. The fourth group replaces the process schematic (P120) with the field support concept (145A). The 2006 data contain another group of concept tasks that deals with coordinating the design points (P120B) and system model (P120D) before the Concept Review (P130). Each of these additions shows a different distribution of labor as the functional groups performing those tasks receive more visibility in Smallcomp's PDP.

Insert Table 3: Critical Iteration Identified by Simmelian Ties about here.

The 2002 data contained several groups dealing with the general theme of coordinating requirements with mechanical design information and verification testing after the concept phase. The 2006 data contain only two groups, dealing with the manufacturability and certification activities before product release. The groups involving P145A, P270, and P435 show an emphasis on iteratively involving manufacturing, field support, and certification. This identifies a change in focus from "Get the requirements right, and the design will work out" (2002) to "Involve downstream activities outside Engineering to influence the design." It is consistent with Smallcomp's change in emphasis from fielding demonstration units to increasing sales volume.

The second point of divergence is the iterative relationships that the process architects believed should be taking place, but which didn't necessarily occur in practice. The two sets involving P315 (reliability analysis) in 2002 were recognized but generally poorly executed. This is also the case for the two sets involving P120C (electrical architecture) in both years, which deal with coordinating design of the electrical and mechanical architectures. These are coordination themes that Smallcomp engineers agree need to exist, but some of which haven't been successfully implemented.

4.3. Individual PDP Tasks

This section uses Key Player analysis [32] to identify tasks that are efficient points for introducing new design information, or whose omission adds potential churn to PDP execution. In product development terms, the reach tasks in Table 4 are the most efficient places to introduce new design information. The bias is very strongly toward design changes in the Concept phase. The results are consistent with daily activity at Smallcomp, as well as with standard PD theory. In both cases, 70% of the PD network has already been reached by the transition from Concept to Preliminary Design at Business Gate 1. In 2002, 5 out of 10 reach tasks were formal reviews. The other 5 dealt with informal interaction (e.g. P257 – Software Requirements or P340 – Assembly Drawing). In 2006, 6 out of the 8 reach tasks involve formal reviews. Smallcomp's response to stress, as measured by the change in Reach tasks, is a reduced number of insertion points for new information, and an increased use of formal reviews for those insertion points. It also shows a stronger focus on reaching the PDP through the Concept phase. 73.5 % of the network is reached by Business Gate 1 in 2006, compared with 68.8% in 2002.

Insert Table 4: PDP Key Player "Reach" Tasks about here.

The "Fragment" nodes in Table 5 lists steps that, if removed, would split the PDP network into multiple, less well-connected task groups. Omitting or completing these tasks incorrectly contributes to fragmented information in the remaining elements of the PDP. These steps are opportunities for management "containment intervention" to ensure they are not overlooked or inadequately completed. For example, several engineers and managers identified accurate organizational design as an issue during brainstorming sessions at Smallcomp. This feedback was a key element of an internal procedure revision to provide guidelines that Integrated Product Development teams not be formed until after the System Preliminary Design Review. The 2006 data shows that task P200 no longer has a key fragment player role in the PDP network. Replacing P215D with P130 shows that the fragment power of trade studies has shifted upstream from functional integration to system definition. The fragment potential of the business gates has shifted downstream, with P298 replacing P195.

Insert Table 5: PDP Key Player "Fragment" Tasks about here.

5. Discussion: Using Network Measures to Manage PDP Execution

Based on the analysis in this paper, we discuss three findings relative to managing PDP execution. First, there is a general trend toward reduced iteration and centralized control. This is shown by the reduced geodesic distances in Table 1, the changing density values in Table 2, and the increased visibility of design reviews and business gates in Table 4 and Table 5. This control is also shown by the reduced variation in the 2006 Outdegree Centrality control chart in Figure 4. However, the Indegree Centrality charts in Figure 5 and Figure 6 show roughly the same variation in 2002 and 2006. This is consistent with the risk management literature on decision-making fallacies following disasters [15]. Smallcomp invested effort to control where its PDP tasks sent output, but did not change how they collected input. **Broader lesson:** Although focusing on task outputs is an understandable approach to controlling a PDP, it is important to reduce variation in task inputs as well.

Second, the brokerage analysis shows a different distribution of labor for three phases in 2006. The Concept phase must both coordinate and represent. The Mechanical Integration has fewer gatekeeper tasks. The Functional Integration has more gatekeeper tasks and fewer representative tasks. In contrast, Detailed Design and Validation showed some change in the relative magnitude of each brokerage role, but not in the position of the roles relative to each other. The changing brokerage roles in the first three phases show a qualitative shift in how information is transmitted from these phases to others in the PDP between 2002 and 2006. Data on PDP definition at multiple points in time is scarce. However, this finding is consistent with broader theory that sees social systems exhibiting increased specialization and division of labor as span of control increases [17, 18]. Here, span of control is the ability to support multiple projects concurrently. **Broader lesson:** Tasks, like people, can have varying roles and burdens in terms of how they share information. Companies that want to support multiple concurrent projects should make sure their PDP tasks have an appropriate distribution of labor.

Finally, 40% transitivity (relatively constant between 2002 and 2006) in a network of people would show vulnerability to error because individuals are getting the same information from multiple sources. For a PDP, transitivity means that a task receives input from both another task, and that task's input. This could be a sign of inefficiency. However, it could also be used as a measure of concurrency in the PDP. The task groups in Table 3 show both valuable (involve field support, manufacturing, or certification) and ineffective (coordinate with the reliability model and the electrical architecture) iteration. **Broader lesson:** Transitivity in a PDP network is not inherently bad. If used as a design criterion, we hypothesize that Simmelian ties, a special case of transitivity, provide a metric of effective concurrency in the PDP.

6. Conclusions and Future Work

One fundamental tenet of Quality Management [19, 37-40] is the importance of data-driven decisions for process control. With the shift in emphasis upstream of manufacturing to product design [41], a growing body of literature emphasizes the importance of improving processes that involves creation and transfer of design knowledge [33, 34, 42-44]. In this context, transmitting information between groups is a significant element of engineering program execution. Managers seeking to make meaningful improvements in the product development cycle may often incorrectly identify actual sources of interdependencies because of the complex and interacting nature of organizational processes, tasks, and relationships [45-48].

The NA metrics described in this paper provide a way to analytically determine the key relationships in complex product development environments, rather than relying on an individual or set of individuals to accurately understand the entire process. This allows product development processes to be emergent, rather than pre-defined phenomena. Real-time adjustment to ideal PDP sequences is often a fact of life due to schedule, budget, and expertise constraints. We believe that using NA metrics to evaluate information flow provides tools that help managers responsible for making decisions in these contexts.

This paper quantitatively validates themes from organizational behavior and quality management literature by identifying structured patterns of information flow in the changes to Smallcomp's PDP. NA metrics provide this validation by identifying information flow patterns at multiple levels. The overall level shows increased centralization and a process being controlled by focusing on the variation in how tasks provide outputs of their information. The phase level shows increased centralization, a qualitative shift in how three phases handle their information flow, and increased specialization as the iterative task groups show a new focus on manufacturing and field support. The individual task level shows increased formalized control as the design reviews and business gates take on more important key player roles.

These patterns raise several questions for companies that create products in volatile markets like Smallcomp's. What documentation and other team integration mechanisms should be put in place to manage the increased information collection and dissemination load during the System Definition phase shown in Figure 4 and Figure 6? What team integration mechanisms should be put in place to effectively insert manufacturing and field support into the design iterations shown in Table 3? Given the increased centralization and emphasis on formal control, what scheduling mechanisms are in place to ensure the appropriate stakeholders are actively involved in the key tasks identified in Table 4 and Table 5? The answers to these questions will shape interventions that project managers must make on a day-to-day basis to execute their specific designs.

One area of future work is to examine which changes to a PDP affect overall team effectiveness. The increased centralization in Smallcomp's 2006 PDP suggests a structure where communication between groups is "purposeful." This structure is less likely to stimulate random or informal communication between functional groups, especially if the groups are working on multiple projects simultaneously. However, the expectation of Smallcomp's PDP architects was that engineers working on the same project would collaborate informally as part of their daily activities. A potential future intervention could formalize interaction across functional groups for problem solving as opposed to expecting that team members would initiate informal discussions. The potential value of interventions to improve team effectiveness rather than redesign the PDP was a recurring finding of work at Smallcomp [36].

References

[1] T. R. Browning, F. Fricke, and H. Negele, "Key Concepts in Modeling Product Development Processes," *Systems Engineering*, vol. 9, pp. 104-128, 2006.

[2] INCOSE, *Systems Engineering Handbook: A "What to" Guide for all SE Practitioners*: International Council on Systems Engineering, 2004.

[3] M. Spinner, *Improving Project Management Skills and Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[4] A. Kusiak, *Engineering Design: Products, Processes, and Systems*. San Diego, CA: Academic Press, 1999.

[5] T. R. Browning, "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," *IEEE Transactions on Engineering Management*, vol. 48, pp. 292-306, 2001.

[6] T. Bustnay and J. Ben-Asher, "How Many Systems Are There? – Using the N^2 Method for Systems Partitioning," *Systems Engineering*, vol. 8, pp. 109-118, 2005.

[7] A. A. Yassine and D. Braha, "Four Complex Problems in Concurrent Engineering and the Design Structure Matrix Method," *Concurrent Engineering Research & Applications* vol. 11, 2003.

[8] T. R. Browning, "Integrative Mechanisms for Multiteam Integration: Findings from Five Case Studies," *Systems Engineering*, vol. 1, pp. 95-112, 1998.

[9] T. R. Browning, "Designing System Development Projects for Organizational Integration," *Systems Engineering*, vol. 2, pp. 217-225, 1999.

[10] D. Batallas and A. Yassine, "Information Leaders in Product Development Organizational Networks: Social Network Analysis of the Design Structure Matrix," *IEEE Transactions on Engineering Management*, vol. 43, pp. 570-582, 2006.

[11] S. T. Collins, J. A. Bradley, and A. A. Yassine, "Measuring Organizational Responses to Stress: A Longitudinal Study of Product Development Task Networks," presented at Eighteenth Annual International Symposium of the International Council on Systems Engineering, Utrecht, The Netherlands, 2008.

[12] S. T. Collins, A. A. Yassine, and S. P. Borgatti, "Evaluating Product Development Systems Using Network Analysis," *Systems Engineering*, vol. 12, pp. forthcoming, 2009.

[13] J. G. March, *Decisions and Organizations*. New York, NY: Basil Blackwell, Inc, 1988.

[14] J. G. March, *A Primer on Decision Making: How Decisions Happen*. New York: The Free Press, 1994.

[15] B. M. Staw, L. E. Sandelands, and J. E. Dutton, "Threat Rigidity Effects in Organizational Behavior: A Multilevel Analysis," *Administrative Science Quarterly*, vol. 26, pp. 501-524, 1981.

[16] K. E. Weick and K. M. Sutcliffe, *Managing the Unexpected: Assuring High Performance in an Age of Complexity*. San Francisco: Jossey-Bass Inc., 2001.

[17] R. L. Bee, *Patterns and Processes: An Introduction to Anthropological Strategies for the Study of Sociocultural Change*. New York, NY: The Free Press, 1974.

[18] U. Hannerz, *Cultural Complexity: Studies in the Social Organization of Meaning*. New York, NY: Columbia University Press, 1992.

[19] W. E. Deming, *Out of the Crisis*. Cambridge, MA: MIT Center for Advanced Engineering Studies, 1986.

[20] R. A. Hanneman and M. Riddle, *Introduction to Social Network Methods*. Riverside, CA: University of California, Riverside (published in digital form at http://faculty.ucr.edu/~hanneman/), 2005.

[21] S. Wasserman and K. Faust, *Social Network Analysis Methods and Applications*. Cambridge: Cambridge University Press, 1999.

[22] J. Scott, *Social Network Analysis: A Handbook*. London: Sage Publications, Ltd., 2000.

[23] G. Chartrand, *Introductory Graph Theory*. Mineola, NY: Dover Publications, Inc, 1985.

[24] R. L. Brieger and P. E. Pattison, "Structures of Economic Interdependence Among Nations," in *Continuities in Structural Inquiry*, P. M. Blau and R. Merton, Eds. New York, NY: The Free Press, 1981, pp. 353-380.

[25] D. Snyder and E. L. Kick, "Structural Position in the World System and Economic Growth, 1955-1970: A Multiple-Network Analysis of Transnational Interactions," *American Journal of Sociology*, vol. 84, pp. 1096-1126, 1979.

[26] R. S. Burt, "Cohesion Versus Structural Equivalence as a Basis for Network Subgroups," *Sociological Methods and Research*, vol. 7, pp. 189-212, 1978.

[27] R. S. Burt, "Social Contagion and Innovation: Cohesion Versus Structural Equivalence," *American Journal of Sociology*, vol. 92, pp. 1287-1335, 1987.

[28] R. J. Brymn, "Structural Location and Ideological Divergence: Jewish Marxist Intellectuals in Turn-of-the-Century Russia," in *Social Structures*, B. Wellman and S. D. Berkowitz, Eds. Cambridge: Cambridge University Press, 1988, pp. 359-379.

[29] J. Galaskiewicz and K. R. Krohn, "Positions, Roles, and Dependencies in a Community Integration System," *Sociological Quarterly*, vol. 25, pp. 527-550, 1984.

[30] F. J. Roethlisberger and W. J. Dickson, *Management and the Worker*. Cambridge, MA: Harvard University Press, 1939.

[31] M. Kilduff and W. Tsai, *Social Networks and Organizations*. Thousand Oaks, CA: Sage Publications, 2003.

[32] S. P. Borgatti, "Identifying Sets of Key Players in a Social Network," *Computational Mathematical Organizational Theory*, vol. 12, pp. 21-34, 2006.

[33] R. G. Cooper, S. J. Edgett, and E. J. Kleinschmidt, "Optimizing the Stage-Gate Process: What Best Practice Companies Do - II," *Research Technology Management*, vol. 45, pp. 21-27, 2002.

[34] R. G. Cooper and E. J. Kleinschmidt, "Stage Gate Systems for New Product Success," *Marketing Management*, vol. 1, pp. 10-29, 1993.

[35] S. T. Collins and A. A. Yassine, "Evaluating Product Development Task Interactions Using Network Analysis," presented at Seventeenth Annual International Symposium of the International Council on Systems Engineering, San Diego, CA, 2007.

[36] S. T. Collins, "Expertise, Disagreement, and Bad Things: Turbulent Knowledge amidst Organizational Stress in Engineering R&D," in *Anthropology*, vol. Ph.D. Storrs, CT: University of Connecticut, 2007.

[37] R. E. Devor, T. Chang, and J. W. Sutherland, *Statistical Quality Design and Control*. Upper Saddle River, NJ: Prentice Hall, 1992.

[38] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. Princeton, NJ: Van Nostrand, 1931.

[39] G. Taguchi and Y. Wu, *Introduction to Off-Line Quality Control*. Nogoya, Japan: Central Japan Quality Control Association, 1980.

[40] F. Taylor, *The Principles of Scientific Management*. New York, NY: Harper, 1911.

[41] K. Nichols, "Better, Cheaper, Faster Products - by Design," *Journal of Engineering Design*, vol. 3, pp. 217-228, 1992.

[42] R. G. Cooper, "Perspective: Third-Generation New Product Processes," *Journal of Product Innovation Management*, vol. 11, pp. 3-14, 1994.

[43] D. Tapping and T. Shuker, *Value Stream Management for the Lean Office*. New York, NY: Productivity Press, 2003.

[44] S. Thomke and D. Reinersten, "Agile Product Development: Managing Development Flexibility in Uncertain Environments," *California Management Review*, vol. 41, pp. 8-30, 1998.

[45] N. Siggelkow, "Misperceiving Interactions among Complements and Substitutes: Organizational Consequences," *Management Science*, vol. 48, pp. 900-917, 2002.

[46] J. D. Sterman, "Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment," *Management Science*, vol. 35, pp. 321-339, 1989.

[47] B. Brehmer, "Dynamic Decision Making: Human Control of Complex Systems," *Acta Psychologica*, vol. 81, pp. 211-241, 1992.

[48] N. P. Repenning, P. Goncalves, and L. J. Black, "Past the Tipping Point: The Persistence of Fire-Fighting in Product Development," *California Management Review*, vol. 43, pp. 44-63, 2001.

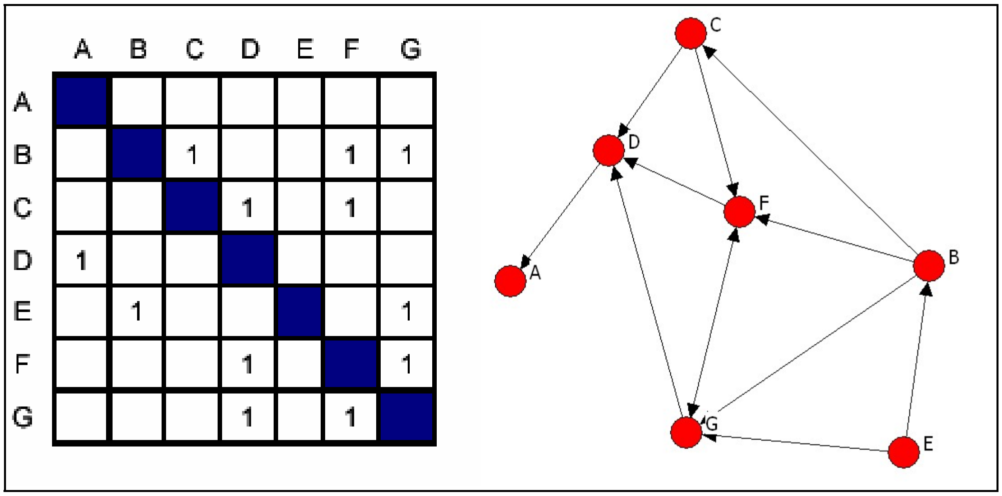

Figure 1: Sample Task Network Matrix and Map

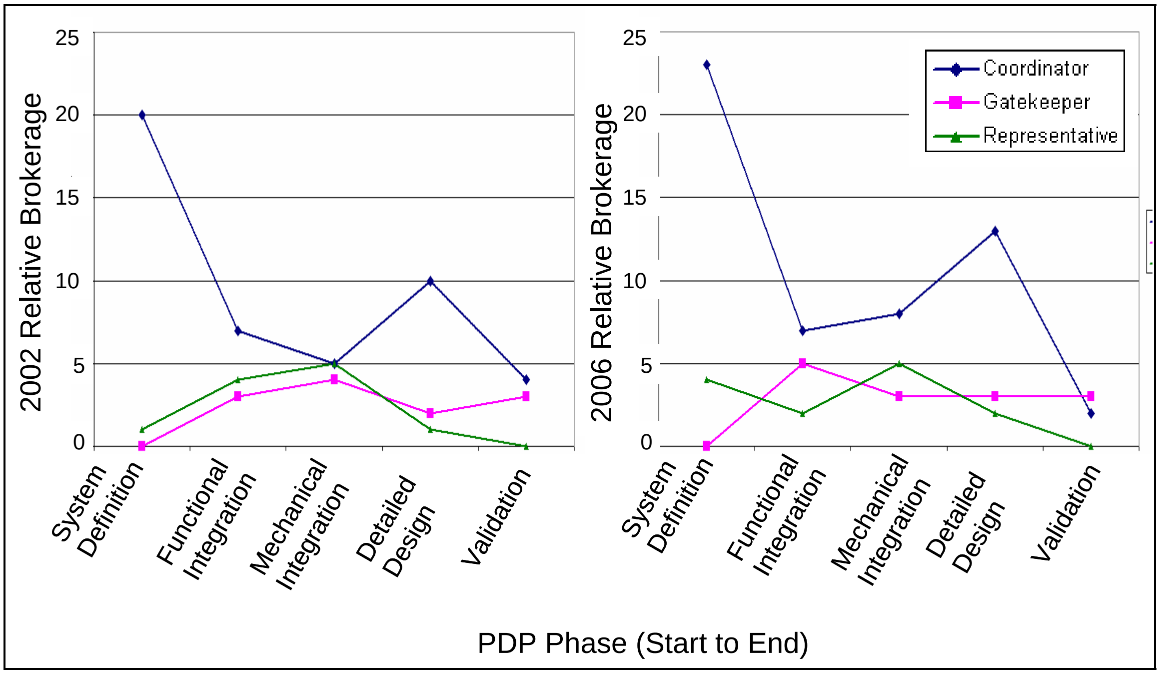

Figure 2: Brokerage Analysis Comparison across PDP Phases

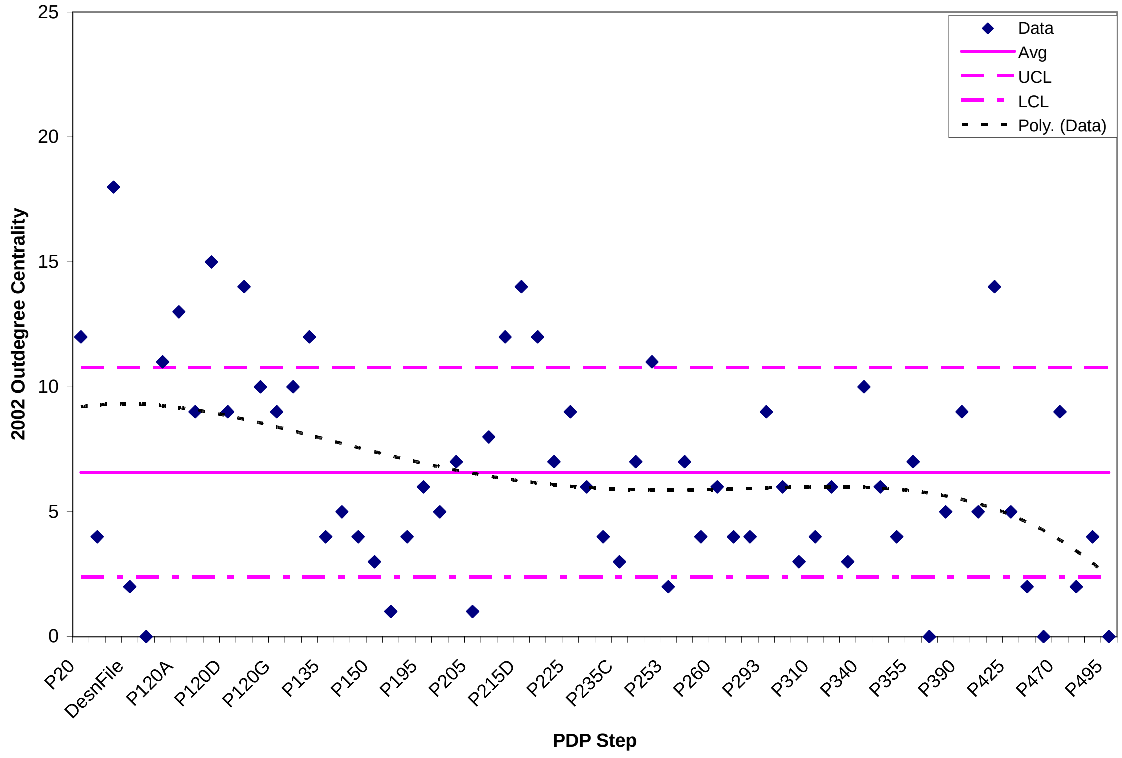

Figure 3: 2002 Outdegree Centrality Control Chart

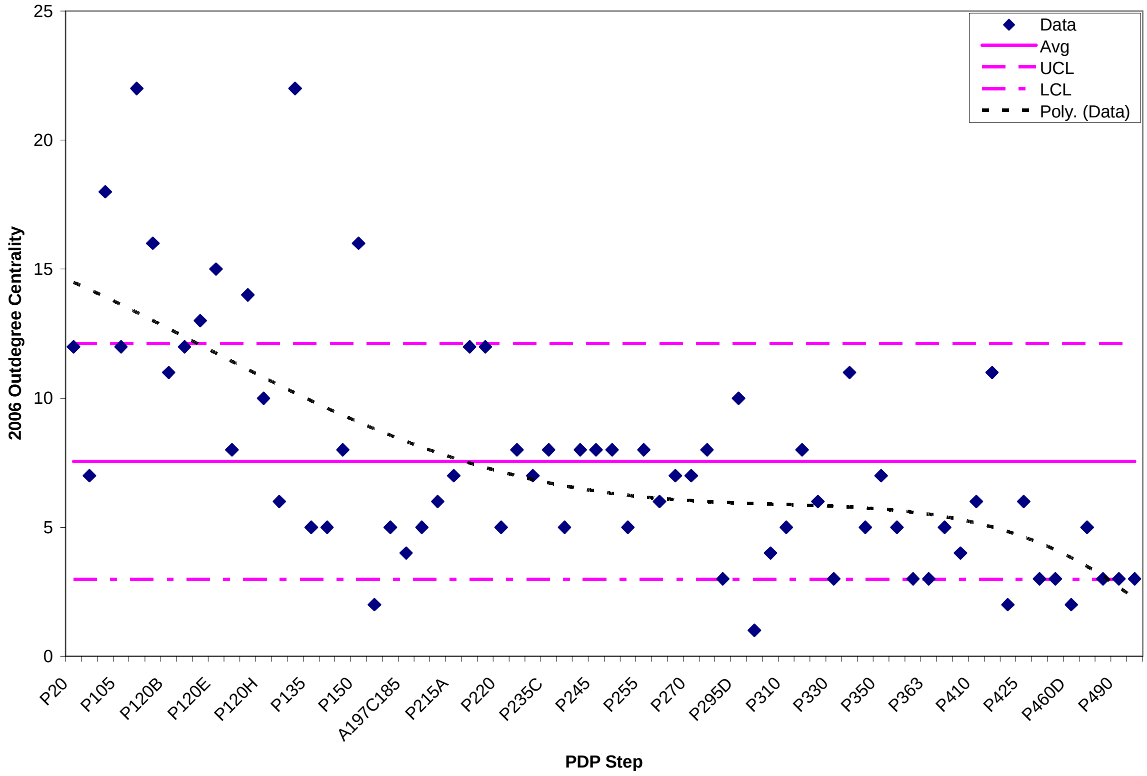

Figure 4: 2006 Outdegree Centrality Control Chart

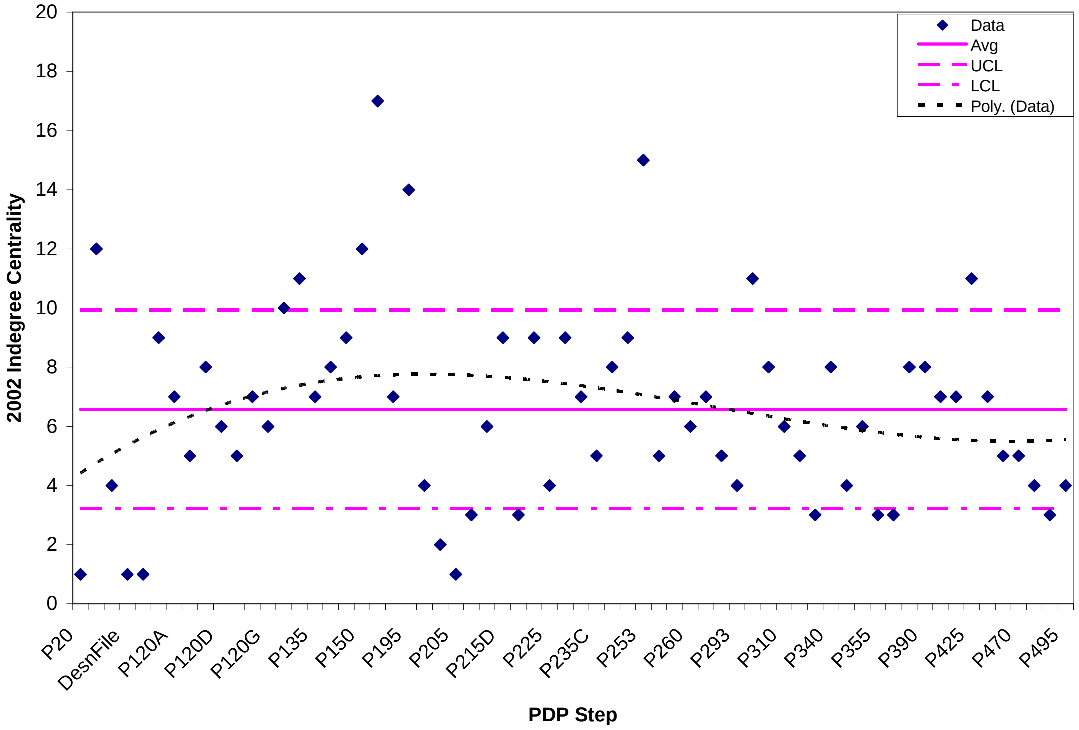

Figure 5: 2002 Indegree Centrality Control Chart

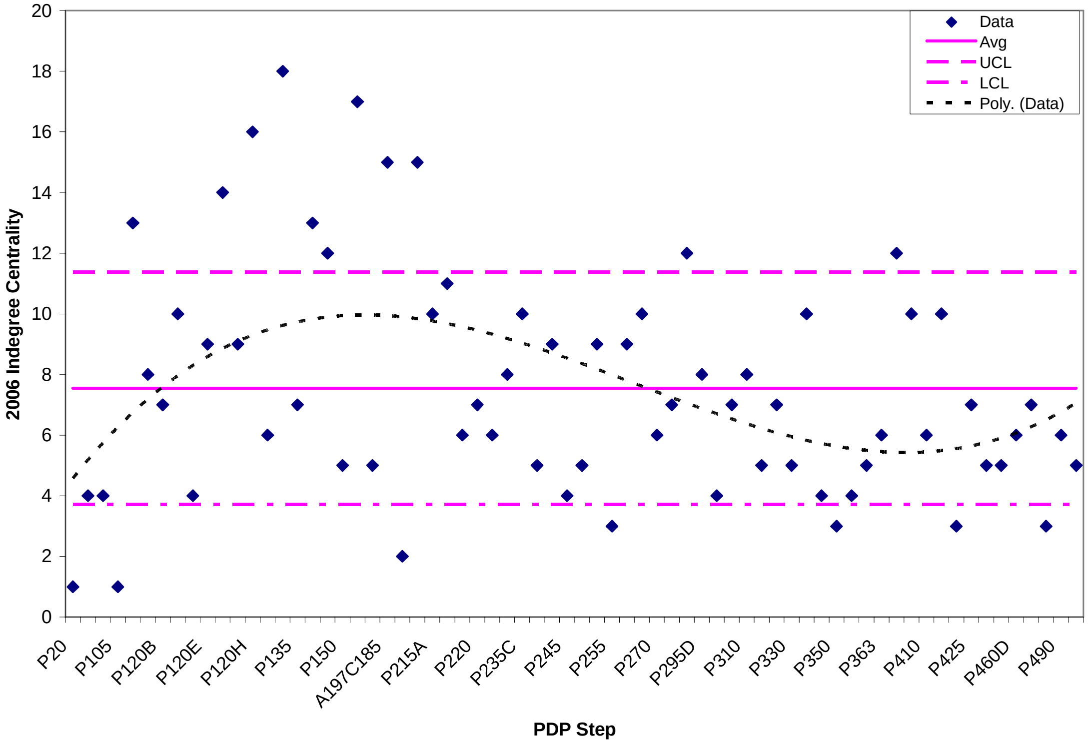

Figure 6: 2006 Indegree Centrality Control Chart

Table 1: Smallcomp Overall PDP Network Comparison

NA Metric	**2002 PDP**			**2006 PDP**			**% Change**
# of Projects	1 – 2			4 – 5			
Project scope	Demonstration prototypes			Low volume production			
# of Nodes	64			68			6
# of Links	385			467			21
% Density	10.4			11.3			0.9
% Transitive Triples	41.3			39.2			-0.9
Geodesic Distances	Distance	# Paths	% of Paths	Distance	# Paths	% of Paths	
	1	385	0.13	1	467	0.16	0.03
	2	781	0.26	2	1022	0.36	0.10
	3	780	0.26	3	814	0.28	0.02
	4	438	0.15	4	420	0.15	0.00
	5	298	0.10	5	140	0.05	0.05
	6	170	0.06	6	13	0.01	-0.05
	7	96	0.03				-0.03
	8	24	0.01				-0.01

Table 2: PD Clusters and Density Changes

	Concept Development	Functional Integration	Mechanical Integration	Detailed Design	Validation
Concept Development	**0.421** ***0.087***	0.040 *0.024*	0.091 *0.065*	0.00 *0.00*	0.00 *0.00*
Functional Integration	0.00 *0.00*	**0.311** ***-0.062***	0.069 *-0.037*	0.107 *0.054*	0.00 *0.00*
Mechanical Integration	0.00 *0.00*	0.153 *-0.029*	**0.424** ***0.038***	0.066 *0.017*	0.00 *-0.042*
Detailed Design	0.00 *0.00*	0.006 *0.006*	0.012 *-0.009*	**0.363** ***-0.092***	0.153 *-0.014*
Validation	0.00 *0.00*	0.00 *0.00*	0.00 *-0.028*	0.00 *-0.056*	**0.524** ***0.157***

Table 3: Critical Iteration Identified by Simmelian Ties

2002		2006	
Task	Task Name	Task	Task Name
110	Product Requirements	110	Requirements
130	Concept review	130	Concept review
120B	Design points	120B	Design points
120C	Electrical architecture	120C	Electrical architecture
120E	Define assemblies	120E	Define assemblies
120F	Mechanical architecture	120F	Mechanical architecture
130	Concept review	130	Concept review
120C	Electrical architecture	120C	Electrical architecture
120F	Mechanical architecture	120F	Mechanical architecture
120A	Define process schematic	120E	Define assemblies
120E	Define assemblies	120F	Mechanical architecture
120F	Mechanical architecture	145A	Field support concept
230	Prelim Assembly Dwgs, ICD	130	Concept review
245	Approve allocation requirements	120B	Design points
295	Preliminary design review	120D	Preliminary system model
220	Draft allocation requirements	252	Updated Design Info: P&I, System Model, Rqts
245	Approve allocation requirements	270	Detailed Mfg Plan
293	Allocation audit	285	Assembly Prelim Design
230	Prelim Assembly Dwgs, ICD	435	Certification testing
245	Approve allocation requirements	490	General Application Readiness Review
293	Allocation audit	495	Business Gate 4
315	Reliability analysis		
390	Critical Design Review		
420	Verification testing		
315	Reliability analysis		
340	Detailed Assembly Dwg		
390	Critical Design Review		

Table 4: PDP Key Player "Reach" Tasks

PDP Step	Tasks Reached (2002)	% of 2002 Network	Tasks Reached (2006)	% of 2006 Network
P20: Business Gate 0: Approval to develop concept design	21	32.8		
P120E: Concept Review			23	33.8
P150: Program Operating Plan (Schedule / Budget)			41	60.3
P165: Conduct Concept Design Review - Approval to proceed with system design.	36	56.3		
P195: Business Gate 1 - Released Requirements and approval to proceed with preliminary design.	44	68.8	50	73.5
P215D: Audit Preliminary Design - Select "Best" Alternative	50	78.1		
P235E: System PDR			58	85.3
P257: Controller Software Requirements	55	85.9		
P295: PDR for Power Plant, Assembly, & Components	59	92.2	63	92.6

P340: Detailed Power Plant Design (Detailed Assembly Drawing)	61	95.3		
P390: Critical Design Review			66	97.1
P420: Define and Conduct Verification tests	62	96.9	67	98.5
P425: Business Gate 3 - Approval to proceed with Verification & Validation Phase.	63	98.4		
P470: Validation Testing			68	100
P475: Prepare installation and operation manuals	64	100.0		

Table 5: PDP Key Player "Fragment" Tasks

PDP Step	Network Fragmented
P130: Conduct Concept Trade Studies	2006
P195: Business Gate 1 – Released Requirements and approval to proceed with preliminary design.	2002
P200: Establish Organization For Design & Development Phase	2002
P215A: Identify System-level Embedded Requirements	2006
P215B: P&I Diagram & Electrical 1-line	2002 / 2006
P215D: Audit Preliminary Design – Select "Best" Alternative	2002
P215E: Detailed Heat & Mass Balance	2002 / 2006
P230: Prelim PP Package - Assembly drawings, BOM, ICD	2002
P298: Business Gate 2	2006

INDEX OF FIGURES

INDEX OF TABLES