Structuring Manufacturer-Supplier Interaction in New Product Development Teams: An Empirical Analysis*

Stephan Schrader and Jan Göpfert

Ludwig-Maximilians-Universität München

Institut für Innovationsforschung und Technologiemanagement

June 1996

---

* We wish to thank Busty Okundaye for his many helpful comments and for his organizational support during the course of the research. Financial support from the International Motor Vehicle Program, MIT, is gratefully acknowledged.

Abstract

This paper investigates task partitioning among problem solvers. It is proposed that the consideration of differing domains of expertise is an important factor for efficient task partitioning.

An empirical investigation of Product Development Teams in the U.S. truck industry, however, suggests that frequently this principle is violated in manufacturer-supplier relations. In many instances, the vehicle manufacturer (customer organization) performs tasks in the development process for which the more specialized supplier organization would be better suited – the supplier is restricted in his problem-solving ability. In other cases, customer organizations do not perform tasks that depend on their knowledge about the vehicle as a system and the interdependencies among its various components, creating a situation of *unclarity*.

Customer requirements provide the basis for task partitioning between customer and supplier. This study shows empirically that both *restrictive* customer requirements that infringe upon the supplier's domain of expertise and the lack of *clarifying* customer requirements hamper the efficiency of joint product-development processes.

Contents

Keywords

- Automobile Industry
- Communication Patterns
- Crossfunctional Teams
- Manufacturer - Supplier Relation
- New Product Development Teams
- Problem - Solving
- Task partitioning

I. Improving Inter-firm Product Development: A Conceptual Framework

A. Task Partitioning in Joint Product Development

New product development is frequently large-scale problem solving. It may consist of thousands or even tens of thousands of single problem-solving tasks, woven into a complex network of interrelationships [1]. This situation entails two requirements faced by every organization: *differentiation* and *integration* [2]. A major problem-solving process and its constituent tasks can no longer be accomplished by a single individual, but instead must be partitioned among a number of individuals, groups, divisions, or even firms. Since these differentiated tasks are part of a complex, interdependent network, integration through extensive communication and coordination among the problem solvers becomes necessary [1].

Von Hippel addresses the question of task partitioning among several problem solvers from the perspective of task interdependencies. He suggests that "precisely where the boundaries between such tasks are placed can affect project outcome and the efficiency of task performance due to associated changes in the problem-solving inter-dependence among tasks …. Task boundaries between problem solvers often have associated physical and organizational barriers. Such barriers can add to the cost of problem-solvers' efforts to achieve cross-boundary communication and coordination" [1]. Thus, optimizing task partitioning reduces the problem solver's cost in this respect. Von Hippel [1] suggests that tasks be partitioned so as to reduce the need for problem solving across task boundaries by unifying highly interdependent tasks[1].

In using task interdependencies as the only determinant for task partitioning, one implicit assumption is that all problem solvers have similar problem-solving competencies. In many cases, however, the reason to involve several problem solvers in the resolution process is to take advantage of their *different* competencies. Thus, the problem solvers' domains of expertise become another relevant determinant of task partioning.

This argument applies especially to *inter-firm* task partitioning. The knowledge and resources necessary to develop many of today's highly sophisticated products, such as automobiles or electronic goods, often exceed the level of what a single firm can or should accomplish. The expertise needed to perform the various tasks involved in a product-development project is distributed across organizational boundaries. *The organizational challenge of partitioning tasks among firms so that they fit and take advantage of different domains of expertise in joint product development is the central theme of this study.*

B. Domains of Expertise in the Joint Customer-Supplier Problem-Solving Process

One important reason to outsource development tasks to suppliers is the customer organization’s limited domain of expertise. The domain of expertise of an organization shall be defined as the specific area in which the organization has accumulated considerable

---

[1] E.g., the better an auto door is at sealing out noise and dirt (a desirable characteristic), the harder it is to close (an undesirable characteristic). In order to reduce cross-boundary problem solving, the tasks "improve door sealing" and "improve door closing" should be callapsed into an "improve door closing and door sealing" task [1].

problem-solving competency, including expert knowledge and/or corresponding resources that foster problem solving [3], [4].

Sharing development tasks with a supplier means that, by incorporating the supplier's domain, the customer organization's limited domain of expertise is expanded. Two differing domains of expertise are united in order to foster the resolution of development problems. For the following model, let us assume that the supplier's and the customer's domains of expertise are quite distinct.

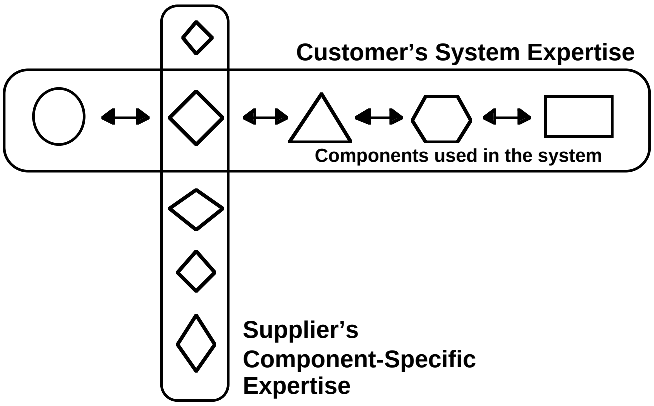


Fig. 1: Assumed Characteristics of the Customer's and Supplier's Domains of Expertise

The customer's domain of expertise is assumed to be wide and system-oriented, being mainly concerned with component interdependencies (see Figure 1). This orientation is a consequence of the final product the customer is producing – a whole system.

Thus, the customer's domain of expertise could consist of knowledge about the product as a system and the relationship among its components, or knowledge about the overall development process. Accumulated resources within the customer's domain of expertise include facilities and equipment needed to develop and manufacture this product (e.g., styling and test facilities, assembly equipment), as well as appropriate personnel. Tasks that arise from the customer's domain of expertise are very much related to the product as a whole and to the coordination of its development activities.

In the automobile industry, the following example tasks depend on the customer's domain of expertise:

- Understanding system constraints such as interface problems of neighboring components (e.g., minimum distance between the exhaust pipe and the fuel tank);
- Project time scheduling;
- Conceptualizing and choosing new technology (e.g., a new restraint system).

In contrast to the customer’s system-oriented domain of expertise*, the supplier's domain of expertise is assumed to be deep and component-specific*, being mainly concerned with the internal design of a particular component (see Figure 1). This again is a consequence of the supplier’s final product – a component within the customer's system. The supplier's domain of expertise lacks the breadth of knowledge accumulated in the customer's organization. But because it is specialized in a particular component and entails detailed knowledge about problems arising from within the component structure, the supplier's domain of expertise is much more focused and in-depth. The supplier has also accumulated considerable resources for developing and manufacturing a specific component (e.g., specific CAD systems, component test gadgets, manufacturing equipment), including specialized employees. Tasks that arise from the supplier's domain of expertise are therefore more component-related and are focused on detailed design[2].

In the automobile industry, the component-specific tasks arising from the supplier's domain of expertise could include:

- Designing the component's internal functionality;
- Selecting the material for the component’s interior parts;
- Refining the manufacturing process in detail.

Assuming that the content of both the customer’s and the supplier’s domains of expertise is relevant to problem-solving in product development, it is beneficial to exploit both domains as much as possible in the development process. This can be accomplished by partitioning the tasks such that they match the problem solvers' domains of expertise.

C. Requirements as a Reflection of Actual Task Partitioning

> "Requirements are the language of engineers."
>
> Busty Okundaye, former GM Design Engineer

Partitioning the tasks of a development project among firms can help to handle the project's complexity. It is a reaction to the multitude of tasks that can no longer be accomplished by a

---

[2] This characterization is, of course, ideal-typical. Suppliers more and more provide component "systems", e.g., complete instrument panels instead of single components. The described distinction, however, can still be kept since even a system supplier's domain of expertise is focused on this system, which remains a subsystem of the system to which it contributes (e.g., a vehicle).

single firm. Due to the interdependencies between tasks (especially when partitioned among different organizations, which provides additional boundaries [5]), information must be exchanged in order to coordinate and link these tasks. *Requirements*[3] *reflect this interplay between task partitioning and reintegration.*

On one hand, because they coordinate tasks across task boundaries, requirements can be seen as an integration instrument. On the other hand, they describe the tasks to be accomplished and therefore represent the actual task boundaries. Requirements are the means of communicating customer needs and expectations to suppliers.

Requirements specify and define the problems that must be resolved by the supplier in the development process and determine the product to be delivered by the supplier. *Since requirements define the development problem, they already contain parts of the problem solution.* In other words, a requirement results from parts of the problem-solving process that are already solved. Thus, the quality and types of requirements provided by the customer determine the possible and necessary degree of supplier involvement in the problem-solving process.

D. Fit and Misfit between Requirements and Domains of Expertise

In a joint development project, requirements regarding the functionality of the components need to be established. This task falls under the customer's responsibility. The more requirements a customer states, the more of the development problem he solves. As we have

---

[3] The term "requirement" will be used for any kind of expectation that must be met (vague or specific, oral or written), whereas "specification" refers only to a more specific requirement that is fixed either literally or in a drawing. For details on requirements and specifications, see [6].

already discussed, however, one reason for joint product development is that the customer may not be competent to solve every detail of the overall problem. Therefore, he needs to involve suppliers that supplement his domain of expertise.

This, however, implies that the customer should limit his problem-solving activities to his own domain of expertise. In cases in which the customer's problem-solving activities go beyond his domain of expertise and infringe on the supplier's domain, the supplier's problem-solving ability is restricted unduly.

In other cases, the customer may knowingly or unknowingly leave some tasks to the supplier, even though he rather than the supplier possesses the problem-solving expertise. In this case, the supplier's task is not defined clearly enough.

Thus, regarding the "fit" between the customer's statement of requirements and his domain of expertise, four cases can be distinguished (see Figure 2). A fit between the customer's statement of a requirement and his domain of expertise occurs when a stated requirement *is* part of the customer's domain of expertise. A fit also occurs when a requirement is *not* stated that is *not* part of the customer's domain of expertise.

	The customer states a requirement...	The customer does not state a requirement...
...that *is* part of his domain of expertise	**Fit**	**Unclarity** (Misfit B)
...that is *not* part of his domain of expertise	**Restrictiveness** (Misfit A)	**Fit**

Fig. 2: Fit and Misfit between Requirements and the Customer's Domain of Expertise

Misfits A and B can be interpreted as customer *restrictiveness* and customer *unclarity*:

Restrictiveness describes a situation in which the customer states requirements in the problem-solving process that are *not* part of his domain of expertise, for example, requirements concerning detailed component-design issues that do not relate directly to the component interfaces. In such cases, the customer infringes on the supplier's domain of expertise, which limits the supplier's problem-solving abilities without improving overall system integration. Thus, restrictive requirements presumably result in a suboptimal problem-solving process.

Unclarity describes a situation in which the customer does not state requirements that are part of his system-related domain of expertise, e.g., requirements that concern component interdependencies. This situation creates unclarity in the problem-solving process due to the lack of solution elements and thus presumably results in a suboptimal problem-solving process.

Both restrictiveness and unclarity are proposed to result in negative performance effects in the joint development process. In the following sections, this proposition will be further explained and substantiated.

1. Effects of Restrictiveness

Several different effects of restrictiveness can lead to the proposed negative performance implications:

a) Limitation of the Problem Solution Space

> "I have often found that some of the most restrictive specifications are quite arbitrary and not based on performance requirements. In such cases, the constraining specifications can be relaxed without causing a performance decrement. Furthermore, I have found that as the constraints are relaxed, production costs are usually reduced"
>
> George Smith, Product Design Consultant [7]

A requirement reduces the problem solution space (that is, the number of possible solutions) by eliminating degrees of freedom. Each specification of a problem dimension eliminates solutions that do not comply with this specification. If the customer states a requirement that is not part of his domain of expertise, a number of feasible and perhaps better approaches to the problem solution may be excluded unnecessarily.[4]

For example, a control cable to open the truck hood is to be designed, running from the interior of the vehicle into the engine compartment. If the customer does not require a certain core material, the supplier is free in assigning this variable. He might find out that a recently developed synthetic core material would actually fit the customer's needs and beat the traditional steel wire in price and flexibility. Thus, the supplier is able to generate the decision about the problem variable "material" himself and to provide a feasible solution to the problem. The customer, however, may require a certain core material for the cable, e.g., twisted cores of stainless steel. It might be a functional necessity of the cable to be able not

---

[4] "Better" can mean two things: (1) a better process outcome (*solution*) and/or (2) a less resource-constraining problem-solving *process* (see Figure 7).

only to pull but also to push the core. If the customer not only specifies this function (which is part of his system expertise) but also the solution in terms of the material (which is not part of his domain of expertise), the requirement becomes restrictive and the solution space has been limited unnecessarily from the start.

b) Lack of Incentive to Seek Better Solutions

When the customer specifies requirements, the supplier has no particular need to seek other, perhaps better, solutions. A supplier's main concern is to fulfill the requirement, assuming that the customer "knew what he did". Such a situation means that the supplier need not enter a resource-consuming problem-solving process to generate the necessary piece of information himself and is not motivated to understand the custmomer’s functional problem. Thus, the problem-solving process might be performed suboptimally, since other, maybe better, solutions are not considered.

c) Worst-Case Assumption

Restrictive requirements are usually set by the customer before the actual problem-solving process evolves. Therefore, these requirements are set in accordance with the customer's limited domain of expertise and based only on an preliminary understanding of the problem. This can lead to worst-case assumptions that result in over-specification of the product, overlooking the possibility that during the course of the problem-solving process more "generous" settings could have been found. Involving the supplier in the decision about such a requirement would probably have resulted in a better problem-solving process.

In the example under discussion, let us assume that it becomes clear during the course of the problem-solving process that the cable gains unexpected stability due to the combined effects of cable core, guide, and cover. If a certain guide stability was prespecified by the customer, possible savings through the application of a thinner guide are lost.

d) Inefficient Requirement Specification Process

It costs the customer resources to provide detailed requirements. As shown before, a requirement contains parts of the problem solution. Therefore, before laying out requirements, a solution must be partly worked out, involving time-consuming design efforts, referring to data books, calculations, and documentation of the requirement. A supplier, with his component-specific domain of expertise, would probably have achieved these tasks more efficiently.

In our example it is assumed that the costumer has specified the detailed design of the control cable. This involves requirements regarding core material, cover, guide, longitudinal load, side load, fastening mechanisms, etc. If the cable design does not belong to the customer's domain of expertise, his cost of working out this design will probably be higher than that of the supplier, for whom the same development task would have been routine.

2. Possible Reasons for Restrictiveness

Considering the negative impacts on the problem-solving process that have been outlined, the question arises: Why do customers issue restrictive requirements? There are several possible explanations:

a) Relation-based Explanation: The customer might have no trust in the supplier's problem-solving capabilities*:* To a large extent, distrust is a question of organizational culture, particularly the basic assumptions predominant in the customer's organization [8]. It is also a question of the customer engineers' opinions about a supplier, based on subjective criteria and past experience.

b) Context-determined Explanation: The power of routine and tradition is strong. Only In recent years, suppliers have become increasingly involved in the development process [9]. The transition to giving the supplier more responsibility is hard to accomplish, due to the long tradition that the customer specifies every detail.

c) Conflict-determined Explanation: It is the engineer's task to foresee technical problems. If problems occur in the engineer's sphere of responsibility, this might have negative personal consequences, whether actual or perceived. Therefore, an attitude of "do everything yourself" has developed in order to minimize this risk – ignoring the strategic possibility of further minimizing that risk by involving experts with a more specific domain of expertise.

d) Cognitive Explanation: If "specifications are the language of engineers" (see Section C.), one can assume that their minds work alike. Engineers have developed the mental capability of breaking down objects into details with an admirable swiftness. It is hard to stop this process at a certain point and to accept the fact that someone else might be better suited to continue the task.

3. Effects of Unclarity

A situation of unclarity is created when the customer does not state requirements that are part of his domain of expertise. Several different effects can lead to the proposed negative performance implications of customer unclarity:

a) Lack of Relevant Problem Dimensions

Selecting relevant problem dimensions for a component-development problem can be done meaningfully only with knowledge of the functional needs arising from the interdependencies within the system in which the developed component will be used. Thus, selecting relevant problem dimensions is part of the customer's domain of expertise.

When the customer fails to provide a requirement expressing a technical interdependence, a situation of objective unclarity, as defined in Section I. D., results. In this situation, the problem-solving process lacks an essential element – a necessary customer input into the process is missing. This unclarity, however, must not necessarily be *perceived* by the supplier. He may perceive his task as being defined clearly even though an essential requirement is missing. Since the missing specification of a relevant problem dimension is not realized by the supplier, the negative consequences will show up only later in the product-development process. This case is particularly problematic becuase of the the tremendous cost of late engineering changes.

Example: Consider the same control cable mentioned in the previous example. In the assembly operation, a considerable side load will perhaps be applied to the cable in the worker's effort to lead it around another component. If this side load is not specified in a customer requirement, the part is likely to break in the assembly operation since it was not designed to withstand this unusual force[5]. A supplier, with his domain of expertise focused on the cable design itself, probably could not have foreseen this side load, whereas the interdependence between "cable development" and "assembly operation" is part of the customer's domain of

---

[5] We are indebted to a cable supplier engineer interviewed for this illustrative example, who experienced a similar situation.

expertise. Since the latter did not provide a relevant problem dimension, the problem-solving process is likely to result in a poor problem solution.

b) „Stuck in Unclarity"

There are cases in which the knowledge of a requirement is indispensable to the continuation of the problem-solving process, since it provides a basic design feature. The design of a component is therefore "stuck in unclarity" in cases where such a requirement is missing.

For example, a basic design feature of a control cable is the maximal longitudinal load that it is supposed to hold. This requirement ultimately determines the choice of the cable's core material, its thickness, the stability of fastening mechanisms, etc. Therefore, if this basic requirement is missing, it first must be generated in order to continue the cable-design process. If the responsible customer engineer is not available to provide this requirement, the development process can be disturbed significantly.

c) Poor Quality of Problem Solution

One way that a supplier can substitute for the lack of a requirement is to estimate the customer's needs in this respect. If not part of the supplier's domain of expertise, however, the estimate faces considerable problems. In terms of the result, it is probable that the estimate is either too high or too low. A too-high estimate can result from a worst-case assumption (see Section I. D.) that tries to exclude all eventualities. In this case, the problem solution will be qualitatively high, but unnecessarily expensive in production and development. On the other hand, too-low estimate due to a lack of problem understanding, results in a poor problem solution that does not meet the customer's functional needs.

Assume, for example, that the customer did not specify the dimension of maximum side load for the cable, although this is part of his domain of expertise. It is probable that the supplier tries to reduce the evolving unclarity by estimating this requirement himself instead of waiting for the customer to do it. To be sure to deliver a functioning component, he can assume a high side load impact and designs the part accordingly. This design will be unnecessarily expensive in production and development in cases where a side load will not be applied in the later assembly operation. Conversely, the supplier could assume, based on his incomplete knowledge of the customer's assembly processes, that no side load will be applied, which can result in later breakage of the part.

d) Poor Problem-Solving Efficiency

If problem solving related to a missing requirement is not part of the supplier's domain of expertise, not only can the problem solution be suboptimal, but so can the resolution process itself. This is because the supplier, due to his inappropriate domain of expertise, is not suited to perform this process efficiently.

Again assume, for example, that the problem-solving process lacks the customer's specification of a maximum side load. What would probably have been routine for the customer, since it falls into his domain of expertise, might be a resource consuming problem-solving process for the supplier. He must study the documentation of previous projects, write inquiry letters to customer's assembly operators, or even simulate assembly operations.

4. Possible Reasons for Unclarity

Although customer unclarity obviously inhibits the problem-solving process, it frequently prevails. Several explanations can be brought forward for this observation:

a) Cognitive explanation [10]: Like every human being, the customer engineer who issues requirements has only a limited cognitive capacity. He might not be fully aware of the problem structure, the interdependencies, and relevant problem dimensions, and therefore not be able to provide the necessary clarifying requirements. In other words: the engineer's "personal" domain of expertise does not fit the task he has to perform.

In the example used above, the customer engineer himself may not know about the side load applied to the cable during assembly operation. This situation and the arising unclarity could probably have been avoided by involving an assembly operator in the development process.

b) Context-determined explanation [10]: Product development processes are dynamic. Requirements are defined in the context of a specific situation. Situations, however, are subject to change in the course of the development process. Stated requirements bind the customer as well as the supplier and deprive both of the possibility of adapting to a changed situation. Since they are definite and static, requirements limit flexibility and freedom in problem-solving processes.

Example: Requiring a certain side-load stability of the cable will probably raise the price of the part. Due to the dynamic of the development process, the customer might not know in an early stage whether a side load will be applied in the assembly operation. A requirement will therefore bind him to the higher priced part, which may later prove to be unnecessary.

c) Conflict-determined explanation [10]: In some instances, unclarity prevents conflicts. Such a situation occurs when different individuals have different goals and fear the consequences of a clash triggered by the clear statement of positions more than they do the consequences of unclarity [11].

For example, a customer design engineer and his supervisor differ fundamentally as to of how the assembly operation should be ordered. In one case the side load will be applied, but not in the other. Requiring the side-load stability will anticipate this decision and could therefore provoke a clash, with personal consequences for the design engineer. Thus, he might choose to refrain from stating this requirement.

d) Cost-oriented Explanation: Achieving clarity consumes resources. As shown in Section C., the development of a requirement is a problem-solving activity. This partial solution must be worked out, interdependencies need to be recognized, and, finally, the requirement must be documented in some form. Not performing these tasks saves the customer some resources in the short term, but the ultimate cost of unclarity is probably considerably higher than these savings.

Example: Some resources are required for the customer engineer to find out about the interdependence between the cable development and the assembly operation. For example, an assembly operator must be consulted to learn about the assembly operation, which might mean visiting the assembly plant. In this very limited view, it is more "effective" to refrain from these activities and not provide the side-load requirement.

II. Joint Product Development in the North American Truck Industry: An Empirical Analysis

The previously developed fit-model has been tested empirically in the North American light-truck industry. Light trucks include pick-up trucks and vans with no more than 10,000 pounds gross vehicle weight, and usually have two axles. Light trucks, and the respective product-development processes, are very similar to what is found in the well-studied automobile industry. Thus, the insights obtained are widely transferable.[6] The light-truck industry, however, offers the advantage of not having been studied extensively in the past, thus increasing the cooperation of the research partners.

---

[6] In many cases, the same components and technology are used for automobiles and light trucks, personnel are frequently shifted between both fields, and the same manufacturing facility often produces light trucks and passenger cars simultaneously.

A. Product Development Teams

Due to the complexity of new product-development projects, the development tasks cannot be accomplished by the vehicle manufacturer alone. Parts of the development tasks are delegated to external suppliers. The consequent problem is that those delegated tasks and their solutions must be coordinated and reintegrated into the overall project.

How do manufacturers delegate development tasks to suppliers while still generating an integrated solution? The investigated vehicle manufacturers in the North American truck industry[7] cope with the integration needs regarding complex subsystems by establishing an interwoven structure of *Product Development Teams (PDT)*[8]. This concept, first applied to this industry in the late 1980s, has since been refined continually. The basic idea is that each PDT takes responsibility for the development of complex or historically troublesome subsystems of the future vehicle[9], bringing together the competencies of various functions. Consequently, the respective component suppliers are frequently included in the PDTs.

Thus, a predominant function of the PDT is to combine and unify the customer's and supplier's differing domains of expertise. The customer's task in the joint PDT is to specify and define the subsystem and its components in relation to the interdependencies to other

---

[7] The following description is based on 23 interviews with Product Development Team members, which we conducted with 2 manufacturers of light trucks and 11 of their component suppliers in November and December of 1993 in the Detroit metropolitan area. Interesting is the fact that both investigated manufacturers, although competitors, apply almost the same structures to accomplish internal and external integration. Thus, no further distinction need be made in this respect, except for terminology.

[8] The terminology varies between manufacturers. One uses the term to describe the Product Development Team, while another refers to it as Program Module Team (PMT), and did not introduce a specific term. The structure, however, was found to be almost the same. For simplicity, a development team will generally be referred to as "PDT" for the rest of this paper.

[9] One interviewee described the components that PDTs deal with as *Gray Box Parts*, putting them elegantly in between Clark and Fujimoto's [12] Black Box Parts and Detail-Controlled Parts in regard to supplier involvement.

subsystems in the vehicle, and thus finally assure the proper interaction of the whole system. The supplier's task is to produce a specific component that fits the specified customer needs.

A PDT consists of members representing a broad range of functions, such as marketing, purchasing, engineering, and manufacturing, as well as different supplier representatives (see Figure 3). The maximum number of members is 20 to 30, but only rarely will all assigned members be present at the meetings. Thus, the actual working size is usually less than 10 people. In the investigated cases, PDT meetings were held between twice a week and once a month, depending on the development stage and the need for information exchange between the involved parties. A PDT's main function is *not* the complete development of the concerned component among all members during the meetings, but rather the joint coordination of the development process and the discussion of major problems.

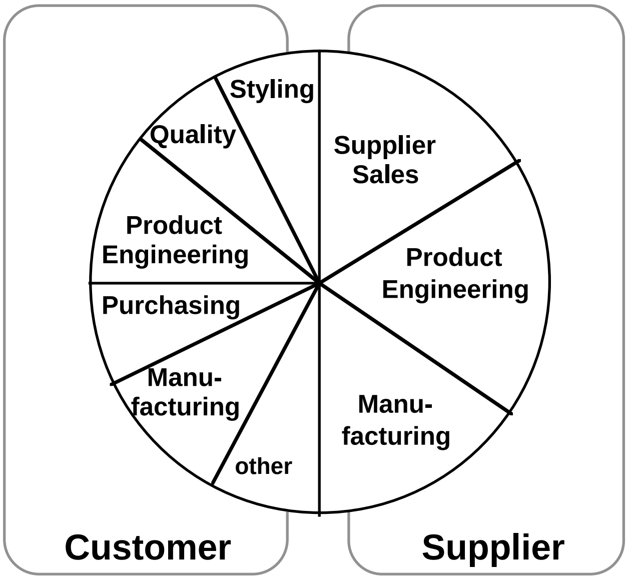


Fig. 3: Setup of a Typical Product-Development Team

A PDT does not have a fixed shape in the course of the development project, but rather adapts dynamically to current needs tasks to be performed, and frequency of meeting. The coordination and intercomponent interaction among different PDTs is accomplished by

System Development Teams (SDTs), [10] which consist of representatives from the various PDTs as well as from the Program Team that controls the overall project (see Figure 4).

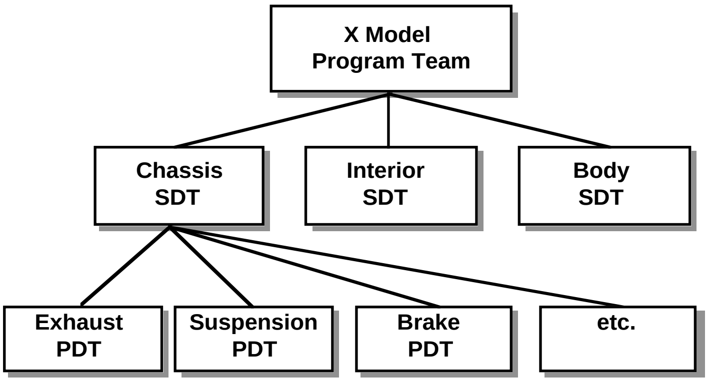


Fig. 4: A Typical PDT in its Metastructure

Product Development Teams were chosen for this analysis because they reflect the distribution of work between suppliers and customers, as assumed in the conceptual framework developed in the first part of this paper.

B. Product Development Team Performance

Evaluating product-development performance is a multilevel problem. The level of analysis determines the criteria upon which performance is evaluated. The performance of each level can be partly explained by the performance of the next level (see Figure 5).

---

[10] Among the SDT's responsibilities are, for example, overall project timing issues, piece cost and investment tracking for all components covered by it, tracking of PDTs in regard to their targets, addressing customer complaints, and managing design changes that have an impact on more than one PDT's subsystem.

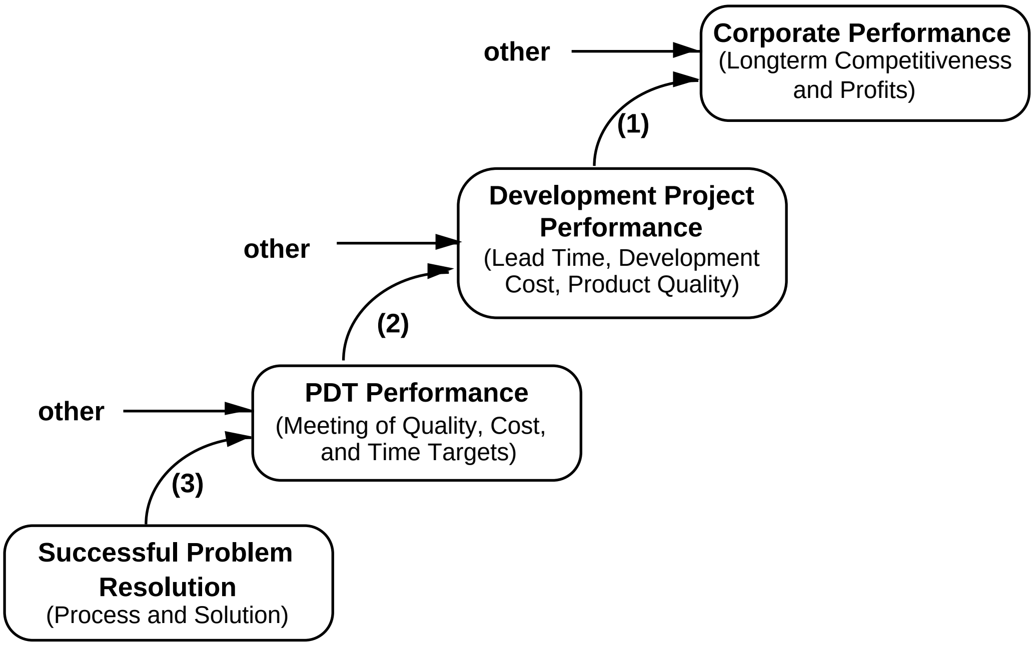


Fig. 5: Levels of Performance

Each development task consists of many problem-solving processes. "Performance" in problem solving can generally be measured in two dimensions: (1) the process result and (2) the efficiency in reaching this result. The former is represented by the achieved *quality* of the outcome, while the latter can be evaluated in terms of consumed resources — *time* and *cost*.

This study focuses on the performance of individual PDTs. Absolute performance figures such as the lead time for component development, development cost, or component quality evaluations are not helpful in this respect since they reflect overall performance and target set on a project level [12]. Crucial to project performance, however, is the extent to which single PDTs meet their assigned targets within the project metastructure. *Thus, PDT performance can be defined as the extent to which a PDT is able to meet its established time, cost, and quality targets.*[11]

---

[11] This approach is used, for example, in a model for performance evaluation in the consumer electronics industry [13].

a) PDT time performance: If the launch of a new model must be delayed because of the failure of a PDT to comply with timing targets, the impact on project performance will be significantly negative.[12] To ensure that the PDT will meet its final delivery date, various timing targets (i.e., "milestones") are set and monitored throughout the phases of the development process. Meeting timing targets can be considered one of the most important performance measures for development teams (as well as for suppliers[13]).

b) PDT cost performance: Since the decisions about component-development budgets are out of the sphere of influence of the PDT, the main performance criterion for the PDT in regard to cost is to what extent it was able to meet the budget specified for the developed component in the product-planning phase. The main cost-drivers in an established PDT are engineering changes to parts or drawings that have already been released[14]. These changes can become extremely costly when made late in the development process, especially after process design or tooling has commenced[15] [14].

c) PDT quality performance: According to Juran and Gryna's terminology [15], the PDT is concerned with *conformance* quality, which refers to how well products conform to the product design or specifications, including reliability, defects in the field, fit, finish, and durability. One of the main tasks of a PDT is to deliver a subsystem that complies with quality targets as stated in various quality attainment procedures.The General Motors GP-3

---

[12] For example, in the case of a car that sells for $10,000, previous research indicates that each day of delay in market introduction costs an automobile firm over $1 million in lost profits [9].

[13] Evaluations of the timeliness of suppliers, both in development and in production, are accurately traced by customers and serve as a decision base for the future choice of suppliers.

[14] According to Clark and Fujimoto [12], engineering changes are the rule rather than the exception in product development.

[15] According to interviews with customer engineers, some suppliers cause and take advantage of design changes: "Some of them make most of their money with costly design changes" (anonymous citation).

procedure[16], for example, includes "events" like fulfilling test standards for prototypes and samples or attainment of a certain process quality.

C. Hypotheses

In Section I. D. the concepts of customer restrictiveness and unclarity in joint product-development processes were derived. Both are expected to result in negative performance effects. The following hypotheses operationalize customer restrictiveness and unclarity on a PDT level and state relationships between these situations and performance measures.

1. Hypotheses Regarding Customer Restrictiveness

In the context of a PDT, the customer can restrict the supplier in several ways. The customer can (1) overspecify the problem *solution*, (2) overdefine the problem *structure*, (3) overspecify the problem-solving *process,* or (4) overspecify the *implementation* of the solution. Those four types of customer Restrictiveness are reflected in the following hypotheses:

Hypothesis 1*: The extent to which the customer pre-specifies a component's final production print is associated negatively with performance in the PDT*. Final production prints are the blueprints that (supposedly) contain all the detailed information necessary to finally produce a component.[17] This information goes far beyond defining the functionality of

---

[16] The General Procedure 3 (GP-3) is a sample approval procedure used by General Motors. Each one of the Big Three automakers (GM, Ford, Chrysler) has specific quality-attainment procedures in place. This, of course, is highly confusing for suppliers who deal with more than one customer. Following a 10-year effort, the Big Three recently agreed on a joint quality-attainment procedure for future product development named AIAG (Automotive Industry Action Group).

[17] Including, for example, information about dimensions, tolerances, materials, and surface finish.

a component. Since the detailed design of a component in a joint development process, and especially its final design, have been assumed to be a part of the supplier's domain of expertise, customer *pre-specification* of production prints indicates that the customer overspecified the problem solution.

Hypothesis 2: *An unreasonable number of KPCs and KCCs specified by the customer restrict the supplier and lower performance in the PDT.* Key Product Characteristics (KPCs) and Key Control Characteristics (KCCs) point out the most critical areas of a component[18]. Every KPC/KCC is a basic component requirement that has a significant influence on the component design. An inappropriately high number of KPC/KCCs specified by the customer means, therefore, that the problem structure is to a large extent predetermined.

Hypothesis 3: *The more timing requirements in the joint development process are coerced by the customer, the lower performance in the PDT will be.* Timing requirements structure the development process by setting deadlines. If timing requirements are set without supplier participation, problem-solving processes are predetermined by the customer. Since the supplier loses the flexibility of adapting the development process to his needs, presumably performance suffers.

Hypothesis 4: *The prohibition of minor improvements to the production process by the supplier without explicit permission of the customer is negatively associated with performance in the PDT.* Even more than detailed component design (product engineering), the development of the product's production process (process engineering) is considered part of the supplier's domain of expertise. A customer who insists on being asked permission for

---

[18] Whereas a KPC *defines* a particular feature (e.g., the minimum radius of an exhaust pipe), a KCC is concerned with *measuring* this feature (e.g., the control requirements to ensure a certain pipe radius). E.g., a fender might have 500 requirements but only 15 KPCs/KCCs that must be precisely controlled in production.

even minor improvements to the production process that do not affect the final product hinders the launch of the developed component into production.

2. Hypotheses Regarding Customer Unclarity

Unclarity can be generated by the customer (1) by specifying the *tasks* to accomplish insufficiently and (2) by not clarifying how to measure successful task *fulfillment*. Both sorts of unclarity are reflected in the following hypotheses.

Hypothesis 5: Unclarity of blueprints due to a lack of necessary information from the customer's domain of expertise is associated negatively with performance in the PDT. Requirements represent information about the tasks to be accomplished (see Section C.). Thus, blueprints are a means of communicating task definitions to the supplier. If the customer fails to contribute this information from his domain of expertise, PDT performance will be accordingly low.

Hypothesis 6: Unclarity in testing requirements is associated negatively with performance in the PDT.

The customer usually demands that the first supplied components be tested in independent laboratories. Specifying the test requirements is part of the customer's domain of expertise. These requirements determine when a problem-solving process can be terminated; if they remain unclear, the test results probably will not be accepted later by the customer and may lead to resource-consuming retesting.

Hypothesis 7: Unclarity in the requirements related to quality-attainment procedures for the submission of production parts is associated negatively with PDT performance. Quality-attainment procedures contain a variety of procedural and technical requirements that are set up and supposedly provided by the customer organization. Only when quality-attainment procedures are successfully passed will component production be approved. Thus, quality-attainment procedures indicate clearly whether development tasks are fulfilled appropriately. Unclarity in this respect might hamper PDT performance due to resource-consuming repetitions of procedures.

Hypotheses 1 through 7 are summarized in Figure 6.

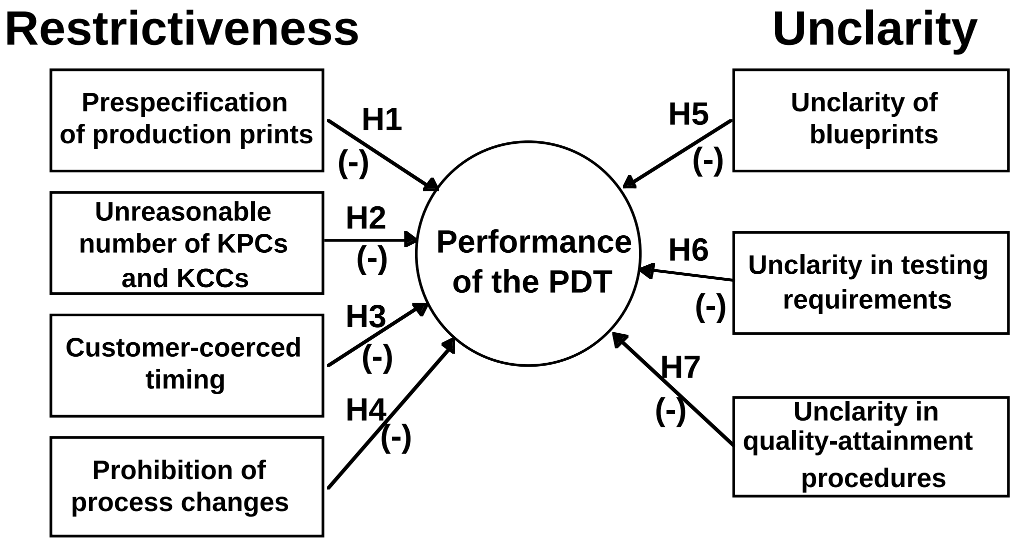


Fig. 6: System of Hypotheses

D. Research Method

In an extensive pilot study, 26 qualitative interviews, including two major vehicle manufacturers and nine of their suppliers, were conducted, mainly in the metropolitan areas of Detroit and Boston. The interviews, ranging from 2 to 4 hours helped us to understand the topic investigated and demonstrated its practical relevance. The theoretical model as well as the successive questionnaire design were enriched strongly by the information obtained in those interviews.

The hypotheses derived in the previous section were tested using two questionnaire surveys. These surveys investigated recent joint product developments between customer and suppliers in the context of Product Development Teams.

The empirical investigation was carried out in two successive steps: First, a survey was conducted in one vehicle manufacturer’s organization. Performance evaluations and names of suppliers and their representatives who had participated in these PDTs were thereby provided. The information requested was somewhat sensitive, especially in connection with naming the evaluated suppliers. Thus, the first survey was limited to one large vehicle manufacturer, using existing contacts. This limitation helped to increase trust, an important consideration due to the sensitive data. Investigating only one manufacturer also controlled for firm effects.

Sample selection was especially important since the customer survey also provided the sample selection for the supplier survey. The final selection of specific PDTs fulfilling our methodological requirements was accomplished with the help of two senior managers in the investigated vehicle manufacturer's organization. We asked them to name PDTs that developed subsystems with a relatively high level of complexity. In this way, the number of suppliers involved in an individual PDT was reduced, yet the intensity of cooperation between customer and supplier presumably increased with the higher complexity. Limiting the PDT in this way also controls for complexity effects.

The two managers also built up contacts with the team leaders of the selected PDTs, who finally assessed the questionnaire, since it was assumed that team leaders could provide the most accurate supplier evaluations, especially in terms of performance. The questionnaire was filled out by PDT team leaders under the supervision of a research team member. There were several advantages to this approach: (1) eventual questions could be answered immediately,

(2) personal presence built up trust, (3) the questionnaire's return rate was 100%, and (4) comments regarding the questionnaire could be obtained directly from the interviewees. Thus, the second survey could be improved accordingly. The selected customer sample can be described as follows:

- PDTs of one major vehicle manufacturer in the North American truck industry were investigated in detail.
- A total of 15 (completed) PDTs ancompassing 23 suppliers and 54 supplier representatives were selected.
- The 15 PDTs developed components for 1994 and 1995 truck models.
- The surveys were filled out by PDT team leaders.

The second survey was conducted mainly among those suppliers that had been named in the first survey, in turn evaluating characteristics of the customer and PDT performance. In this way, a mutual evaluation was obtained, presuming that evaluation of others is less distorted than self-evaluation.[11] Suppliers were selected for the sample mainly from the information provided in the customer survey, but the survey also covered suppliers that worked with another large automobile manufacturer. The supplier sample can be described as follows:

- 11 PDTs were selected.
- The PDTs were held by two major vehicle manufacturers in the North American truck industry.

- Five different supplier organizations were included.
- The survey was filled out by supplier representatives who participated regularly in the PDT.

E. Test of Hypotheses

The following analysis is limited by the relatively small sample size. Therefore, significance in a statistical sense is hard to obtain, although the small sample size causes high correlations. The intent of this small-scale empirical investigation should not be understood as a conclusive statistical test of the stated hypotheses, but rather as a substantiation of the qualitative argument and as a stimulus for further research.

1. Restrictiveness: Empirical Findings

Hypotheses 1 to 4 postulated negative associations between performance in the PDT and restrictive customer requirements. Table 1 summarizes the empirical findings.

Table 1: Restrictive Requirements Hamper PDT Performance (Hypotheses 1 through 4)

Correlation between Performance and Measures of Restrictiveness	Time Performance	Cost Performance	Quality Performance
Degree of final production-print prepecification (Hypothesis 1)	-0.57*	-0.51*	n.s.
Amount of KPCs and KCCs given by the customer (Hypothesis 2)	-0.78***	-0.78***	n.s.
Customer-coerced timing (Hypothesis 3)	-0.52$^+$	-0.48$^+$	n.s.
Prohibition of minor process changes (Hypothesis 4)	n.s.	n.s.	+0.79***

n=10[19]; Significance levels (one-tailed): $^+$ 0.15, * 0.1,*** 0.01

---

[19] One case had to be excluded due to missing values in the performance evaluation.

Hypothesis 1 and 2 are supported by the data only with regard to the performance dimensions of time and cost, but not with regard to quality performance. Thus, customer restrictiveness (in terms of production-print prespecification and number of KPCs and KCCs) is correlated negatively with efficiency (cost and time) of the joint development process, but not with the quality of its result.

Customer-coerced timing (Hypothesis 3) is associated weakly with efficiency (cost and time) in the PDT, whereas the association with quality performance is again not significant.

Hypothesis 4 has been rejected fully. The prohibition of process changes without explicit permission from the customer is, in contrast to what was hypothesized, associated positively with quality performance, but not with process efficiency (in terms of time and cost performance). This means that suppliers who had to obtain permission from their customers to make minor changes in the production process performed significantly better with regard to quality than did "unrestricted" suppliers. Two arguments can be made to explain this unexpected result.

First, it is possible that the need to obtain permission represents a stimulus for discussion and exchange between customer and supplier. The fact that permission must be obtained does not mean that no changes will take place or that the supplier's suggestions are not taken seriously. A mutual discussion about the matter might result in an even better solution than the one implemented. Thus, prohibiting process changes without permission is not restrictive in the sense that the supplier's domain of expertise is intruded upon, if the customer’s involvement can be interpreted as being supportive. This is also a possible explanation of why efficiency measures (cost and time) do not correlate with the prohibition of process changes.

The mutual exchange and improvement process is resource-consuming and, thus, has positive effects on the quality of the solution but not on efficiency.

Second, a certain degree of customer control over the production process may be beneficial since it prevents the supplier from applying changes that in fact *do* affect the product in ways unforeseen by the supplier.

2. Unclarity: Empirical Findings

Hypotheses 5, 6, and 7 postulate negative associations between performance in the PDT and customer unclarity due to missing requirements. Table 2 summarizes the empirical findings.

Table 2: Customer Unclarity Hampers PDT Performance (Hypotheses 5 through 7)

Correlation between Performance and Measures of Unclarity	Time Performance	Cost Performance	Quality Performance
Unclarity of blueprints (Hypothesis 5)	-0.47*	-0.40+	n.s.
Unclarity of test requirements (Hypothesis 6)	-0.39^{+}	-0.42+	n.s.
Unclarity of quality attainment procedures (Hypothesis 7)	-0.59**	-0.51*	+0.45^{+}

n=10; Significance levels (one-tailed): $^{+}$ 0.2, * 0.1,** 0.05

The empirical data moderately supports Hypotheses 5 through 7 with regard to development process efficiency (time, cost) but not with regard to the process result (quality). This outcome is similar to what was found in the test of hypotheses regarding customer restrictiveness. Efficiency of product development processes in PDTs (in terms of cost and time performance) is associated negatively with unclarity in customer requirements (regarding production prints, test requirements, and quality-attainment procedures). Quality performance (process result), however, is not associated with unclarity as postulated.

At first glance, the positive correlation between unclarity of quality attainment procedures and quality performance is confusing. This correlation can be explained, however, within the fit-model developed in section D: A supplier to whom quality-attainment procedures and the accompanying requirements are unclear might try to estimate those matters. Since passing the procedures is an absolutely necessary condition to launch component production, this estimate is likely to be too high. The result is a partly over-designed component, the development process of which is unnecessarily resource-consuming in terms of time and cost. In addition, the quality-attainment procedures might be performed inefficiently due to the supplier's inappropriate domain of expertise.

3. Correlation Between Clarity and Restrictiveness

An unexpected empirical finding were relatively high correlations between variables measuring customer unclarity and variables describing restrictiveness[20]. Table 3 demonstrates these correlations using the example of production-print clarity.

Table 3: Clarity of Production Prints Correlates with Measures of Restrictiveness

Correlation Between Clarity of Blueprints and Measures of Restrictiveness	Clarity of Blueprints [a]
Degree of final production-print prespecification by the customer	0.88***
Inappropriately high number of KPCs and KCCs given by the customer [b]	0.58*
Supplier's perception of limited problem-solving abilities due to restrictive requirements	0.57*

n=10; Significance levels (two-tailed): * 0.1, *** 0.01; [a] "Clarity of blueprints" is the inverse of variable "unclarity of blueprints"

These correlations suggest two possible situations: (a) High customer clarity coincides with high customer restrictiveness, and (b) customers that are less restrictive simultaneously seem to be less clear in their requirements. Why are these situations likely? A plausible explanation is that clarification and Restrictiveness are "transmitted" by the same means – requirements – and therefore might be difficult to distinguish. Thus, it is likely that either clarifying *and* restrictive requirements are stated or that neither of them is.

---

[20] Of the 12 possible correlations (with 4 variables measuring restrictiveness and 3 measuring clarity), 6 are significantly (0.1 level) associated with coefficients higher than 0.50. There are no significant negative correlations.

a) High customer clarity and restrictiveness: It is likely that a thoughtful engineer who clarifies a problem well by considering important problem dimensions and stating appropriate requirements at the same time identifies an original problem solution. It is probable that he will then impose this solution through a consequent set of restrictive requirements (see also Section I.D.1) infringing intentionally or unintentionally on the supplier's domain of expertise and restricting the number of possible solutions. In addition, distrust of the supplier's problem-solving capabilities as well as the power of tradition and routine (see Section I.D.2.) probably foster statement of both clarifying *and* restrictive requirements.

b) Low levels of clarity and restrictiveness: On the other hand, a customer who decides to delegate most of the design tasks to his suppliers will accordingly be less restrictive in his requirements. At the same time, it is highly possible that he will neglect his task of clearly defining the problem, assuming the supplier knows what to do – which might not be the case. Thus, the supplier is unrestricted but lacks the necessary problem clarification that arises from the customer's domain of expertise.

III. Conclusions

A. Conclusions from the Empirical Analysis

Three main conclusions can be drawn from the empirical study:

(1) Customer restrictiveness in regard to overspecifying the problem solution, the problem structure, or the problem- solving process leads to inefficiency in joint product development:

A major reason for joint product development is to increase the customer’s limited domain of expertise. Whereas the customer’s domain of expertise is mainly concerned with the functionality and interaction of the various components within the final product, the supplier's domain of expertise concerns mainly the detailed component design.

The empirical data indicates that a customer who infringes on the supplier's domain of expertise by setting restrictive requirements hinders efficient product development. Restrictiveness can occur in several ways. The customer can (1) overspecify the problem solution, (2) overdefine the problem structure, or (3) overspecify the problem-solving process. All three possibilities lead to inefficiency in the joint development process (measured as the extent to which time and cost targets were met).

However, no negative effect of restrictiveness on the development *result* (quality) was found. This unexpected finding may indicate that the customer's quality expectations can be met although they might be reached inefficiently due to restrictiveness.

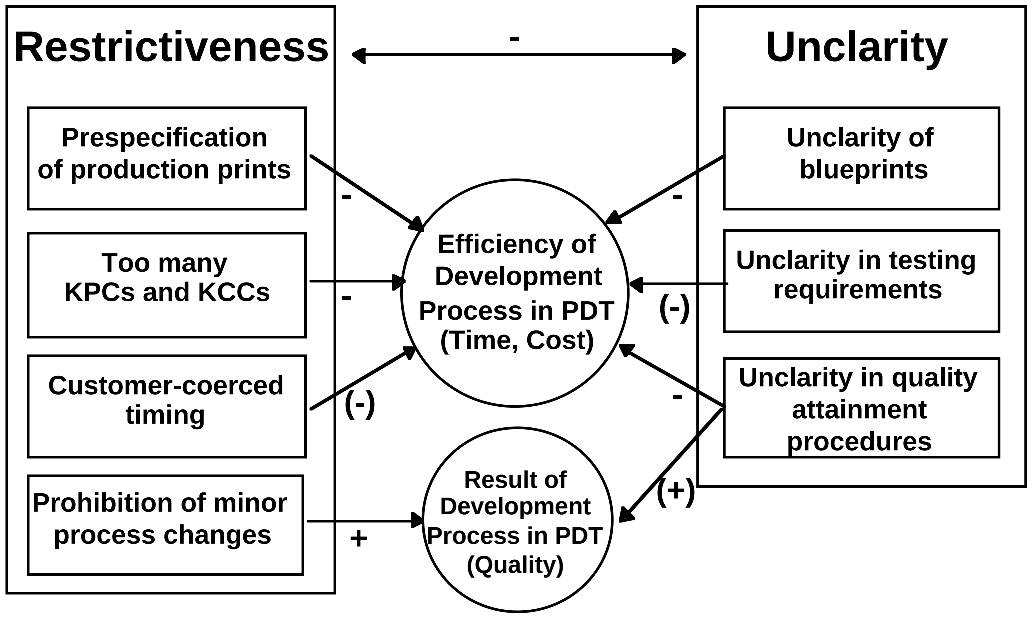
Also, prohibiting process changes without customer permission, in contradiction to what has been postulated, fosters the quality of a solution and has no effect on the efficiency of the development process. This indicates that a customer's controlling of the supplier's production process appears to be to some extent beneficial, stimulating discussion and exchange of information more than it restricts the supplier.

(2) Customer unclarity regarding task definition leads to inefficiency in joint product development: Customer requirements are necessary to define the development problem. They provide information that arises from the customer's domain of expertise. The empirical data shows that customer unclarity due to a lack of requirements that define the tasks to be accomplished, as well as how to evaluate the final results, hinders efficiency in the joint development process.

Similar to the findings regarding restrictiveness, the process outcome (quality) is not correlated with customer unclarity. This again may indicate that the given quality must be met under all circumstances, even when reached inefficiently due to unclarity.

(3) The higher customer clarity, the more the customer usually restricts the supplier at the same time: Customer clarity seems to coincide with customer restrictiveness, and vice versa. This means that customers either tend to be clear and restrictive in their requirements or unclear and not restrictive. A possible explanation concerns difficulties in the distinction between clarifying requirements (arising from the customer's domain of expertise) and restrictive requirements that intrude upon the supplier's domain of expertise.

Figure 7 summarizes these empirical findings. The initial propositions regarding the negative impacts of restrictiveness and unclarity were supported empirically only in part. Whereas the *efficiency* (time and cost) of joint development processes between customers and suppliers in the context of PDTs was hampered as expected, process *result* (quality) seemed to be not — or partly even positively — related to customer restrictiveness and unclarity.



+ = positive influence; (+) weak positive influence; - = negative influence; (-) = weak influence

Fig. 7: Summarized Findings

B. Task Partitioning: Conflicting Approaches?

We began with the question of how to partition tasks among different problem solvers. Von Hippel [1] addresses this question under the perspective of task interdependencies. He suggests that in task specification highly interdependent tasks be unified so as to reduce the need for problem solving across task boundaries.

This study has shown the importance of another factor for task partitioning — the problem solvers' domains of expertise. It has been suggested that when problem solvers have different domains of expertise, these should be exploited by partitioning tasks accordingly.

The two approaches may lead to different ways of task partitioning when highly interdependent tasks do *not* fall into one problem solver's domain of expertise (see Figures 8 and 9). Both ways of task partitioning, however, are suboptimal. In Figure 8 one problem solver must perform a task that is beyond his domain of expertise; in Figure 9 many task interdependencies occur across task boundaries.

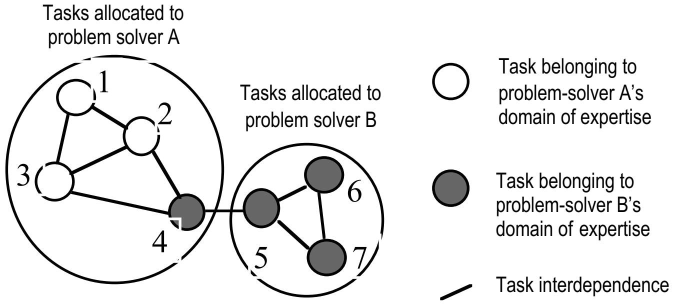


Fig. 8: Task Partitioning According to Task Interdependencies

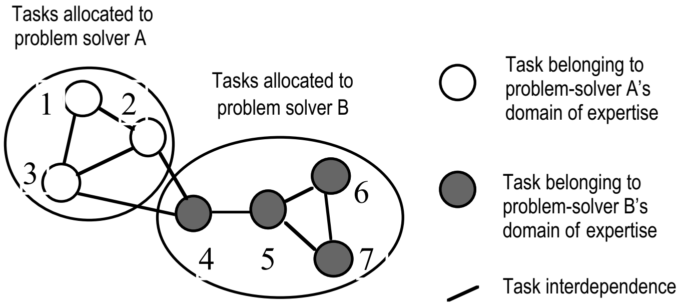


Fig. 9: Task Partitioning According to the Problem-solvers' Domains of Expertise

What can be done about this situation? Two possible solutions are suggested: (1) Problem solver A in Figure 8 expands his domain of expertise and learns to perform the critical task 4, or (2) measures to reduce task interdependencies among Tasks 2, 3, and 4 are implemented, for example through changing the design or standardizing the interface.

Since both solutions will cost resources, the one that is estimated to be less expensive should be chosen. This cost estimation can, of course, deliver only rough numbers because too many variables are still unclear. But even rough calculations in regard to time or direct cost might help to decide which alternative is preferable in a specific case.

C. Managerial Implications

What are the basic implications of this study in regard to the practical management of joint product-development processes? First of all, it seems necessary that firms in general, and especially those in the customer's position (which implies a large influence on the supplier interface structure), become aware of the strengths and limitations of their own domain of expertise, as well as of their partners' domains. This recognition can lead to a more efficient structure of task partitioning. Possibly the most important finding is that improving task partitioning does *not* necessarily require major organizational "revolutions" on a strategic level but can, at least to some extent, be accomplished by the individual engineer who deals daily with suppliers. This is possible because the statement of requirements represents a basic means of task partitioning. Thus, an individual customer engineer who is responsible for the development of requirements, by deciding which requirements to state and which not to state, has considerable influence on the *factual* task partitioning between his own organization and the supplier with which he deals. This finding is even more interesting, since it demonstrates that integration instruments that are introduced at a corporate level (e.g., Product Development Teams) must be accompanied by activities on the individual level in order to foster development efficiency.

IV. References

[1] E. von Hippel, "Task partitioning: An innovation process variable", *Research Policy*, vol. 19, pp. 407-418, 1990.

[2] P. R. Lawrence and J. W. Lorsch*, Organization and Environment: Managing Differentiation and Iintegration*. Cambridge: Harvard University Press, 1967.

[3] S. Schrader, *Zur Organisation der Schnittstelle zwischen Zulieferern und ihren Kunden: Stand der Forschung und neue empirische Befunde*. University of Kiel: Working paper 318, Summer 1993.

[4] S. Schrader, *Organization of the Supplier-Manufacturer Interface: The Importance of Domain Specific Communication*. Unpublished MIT working paper, February 1994.

[5] T. Allen, *Managing the Flow of Technology*, Cambridge: MIT Press, 1986.

[6] M. Fischer, *Engineering Specifications Writing Guide*. Englewood Cliffs, N.J: Prentice-Hall, 1983.

[7] G. L. Smith, "A model for Designing", *Industrial Engineering*, pp. 35-38, November 1989.

[8] E. H. Schein, *Organizational Culture and Leadership*. San Francisco: Jossey-Bass, 1985.

[9] K. Clark, "Project scope and project performance: the effect of parts strategy and supplier involvement on product development", *Management Science*, vol. 35, pp. 1247-1263, 1989.

[10] J. Hauschildt, *Innovationsmanagement*. München: Vahlen, 1993.

[11] D. Gebert and L. von Rosenstiel, *Organisationspsychologie*. Stuttgart: Kohlhammer Verlag, 1989.

[12] K. Clark and T. Fujimoto, *Product Development Performanc.* Boston: Harvard Business School Press, 1991.

[13] Z. Y. Chang and K.C. Yong, "Dimensions and indices for performance evaluation of a product development project", *Journal of Technology Management*, vol. 6, pp. 155-167, 1991.

[14] S. Wheelwright and K. Clark, Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality. New York: Free Press, 1992.

[15] J. Juran and F. Gryna, *Quality Planning and Analysis*. New York: McGraw-Hill, 1980.