How does development lead time affect performance over the ramp-up lifecycle? : evidence from the consumer electronics industry

How does development lead time affect performance over the ramp-up lifecycle?
Evidence from the consumer electronics industry

Andreas Pufall, Jan C. Fransoo, Ad de Jong, Ton de Kok

Beta Working Paper series 390

BETA publicatie	WP 390 (working paper)
ISBN	
ISSN	
NUR	804
Eindhoven	June 2012

How does development lead time affect performance over the ramp-up lifecycle? Evidence from the consumer electronics industry

Andreas Pufall[1,2], Jan C. Fransoo[2], Ad de Jong[2], A. G. (Ton) de Kok[2]
andreas.pufall@nokia.com, j.c.fransoo@tue.nl, a.d.jong@tue.nl, a.g.d.kok@tue.nl

1 Nokia GmbH
Product Creation Center Ulm
P.O. Box 2907
DE-89081 Ulm
Germany

2 Technische Universiteit Eindhoven
School of Industrial Engineering
P.O. Box 513
NL-5600 MB Eindhoven
Netherlands

Date: 26 June, 2012.

How does development lead time affect performance over the ramp-up lifecycle? Evidence from the consumer electronics industry

Abstract

In the fast-paced world of consumer electronics, short development lead times and efficient product ramp-ups are invaluable. The sooner and faster a firm can ramp-up production of a new product, the faster it can start to earn revenues, profit from early market opportunities, establish technology standards and release scarce development resources to support new product development projects. Yet, many companies fail to meet their time-to-market and time-to-volume targets and the complex interrelationships between product characteristics, development lead time and ramp-up performance are largely unexplored. In response to these limitations our study focuses on three research questions: (1) To what extent is ramp-up performance determined by development lead time and product complexity? (2) How do these relationships change in the course of the ramp-up lifecycle? and (3) How can the results be explained? Our results contribute to the field of operations management in three ways. First, we offer a more comprehensive and enriched analysis of the drivers for development lead time and ramp-up performance in the cell phone industry. Second, we demonstrate that late schedule slips – although disastrous for customer relations in which due dates are crucial – provide the opportunity to build up (semi-finished) product buffers which in turn increase the initial ramp-up performance. Third, we show that it is important to take these effects into account in a jointly and lifecycle-dependent manner. Thus, our insights support management efforts to anticipate the consequences of product design decisions, predict development schedule risk levels, and make informed decisions about production volume commitments.

Keywords

New product development, development lead time, product complexity, ramp-up performance, PLS path model

1 Introduction

The speed and efficiency with which new products are developed and introduced into large-volume production has an important influence on competitiveness in manufacturing (Hatch and Mowery 1998). Particularly in the field of consumer electronics where product lifecycles shrink, technology advances and competition intensifies, short development lead times and efficient ramp-ups are invaluable for several reasons. First, the faster a company can ramp-up production of a new product, the more quickly it can begin to earn significant revenues from the new product and recoup its development investments (Pisano and Wheelwright 1995). Secondly, fast ramp-ups enable firms to profit from early market opportunities, set technology standards and accumulate experience with volume production. Finally, scarce product development and manufacturing engineering resources can be released to support subsequent product development projects instead of solving production problems. From an operations management perspective, product ramp-up marks the transition between product development and mass production – or as Wheelwright and Clark (1992, p. 8) detail: "In ramp-up the firm starts commercial production at a relatively low level of volume; as the organization develops confidence in its (and its suppliers') abilities to execute production consistently and marketing's abilities to sell the product, the volume increases. At the conclusion of the ramp-up phase, the production system has achieved its target levels of volume, cost and quality." Yet many companies fail to meet their targets regarding product volume, cost, and quality (Kuhn et al. 2002, Schuh 2005). Prior research has identified time related variables and other characteristics as determinants of ramp-up performance. However, most of these studies have examined development lead time (time-to-market) or ramp-up performance (time-to-volume) separately, while their interrelationship has received only little attention (Terwiesch et al. 1999, Gerwin and Barrowman 2002).

For example, Clark and Fujimoto (1991) investigated the effect of strategy, organization and management on lead time, quality and productivity in a field study within the automotive industry. They found that among other drivers the development process, product complexity and manufacturing capabilities relate closely to superior development lead time and ramp-up performance. In a large German case study, Kuhn et al. (2002) provide an analysis of the problems, influential factors and processes during the ramp-up period of mass volume products. They confirm the influence of product characteristics, manufacturing capability, people and organizational issues as well as the used methods and tools on ramp-up performance in terms of cost, quality and time. Another study in the automotive industry (Almgren 2000) identified the relevant types of disturbances at the manufacturing start and categorized them into four groups: product concept, material flow, production technology, and work organization. Additional influential factors are proposed by Keil et al. (2007) as a response to differences in the manufacturing setup of semiconductors compared to automobiles. They categorize the relevant factors into internal (company, organization, technology, people) and external (market and competitive environment, customers, suppliers) factors.

A longitudinal study by Terwiesch et al. (1999) provides a detailed description of the ramp-up period in the data storage industry and sheds light on the various aspects (e.g., organizational learning, experience transfer, problem solving pattern) that are crucial for time-to-volume. Langowitz (1987) observes that an atmosphere of high communication and coordination, together with clear definitions and the use of milestones from the beginning of a project have a positive effect on the initial commercial manufacture of products. Also, companies should attach high emphasis to the manufacturability of their products by including the manufacturing departments in the development process. According to Vandevelde and Van Dierdonck (2003), design teams with empathy towards manufacturing and who employ a formal approach face smoother production start-ups even under the circumstances of increased product complexity and newness. In addition, Pisano and Wheelwright (1995) reinforce the link among manufacturing process innovation, productive product launches and enhanced product functionalities. Matsuo et al. (1997) illustrate how process improvements can lead to significant financial rewards through the use of time-to-volume and break-even time metrics in the capital intensive semiconductor industry. Van der Merwe (2004) proposes a conceptual model that supports the association between different types of novelty (product, personnel, supplier, and process) and learning types that drive ramp-up performance. Finally, Berg (2007) provides a comprehensive literature review of factors that influence production ramp-up performance. Despite these studies and although researchers and practitioners recognize the potential impact of accelerated product development and efficient ramp-up on organizational success (Clark and Fujimoto 1991) the quantitative relationships between the vast number of influential factors and ramp-up performance have only been investigated partially and insufficiently (Krishnan and Ulrich 2001, Kuhn et al. 2002).

In response to these limitations our empirical study extends the work of Pufall et al. (2012) who consider ramp-up performance as being dependent on product complexity with the inclusion of development lead time in order to address three research questions: (1) To what extent is ramp-up performance determined by development lead time and product complexity? (2) How do these relationships change in the course of the ramp-up lifecycle? and (3) How can the results be explained? We take a confirmatory and exploratory approach in order to answer these research questions. First, we operationalize development lead time, product complexity and ramp-up performance based on our conceptual framework (Figure1), then we integrate them into a partial least squares (PLS) path model. Before we explore how these relationships change over time by modifying the time horizon of our dependent variables we are testing our fundamental hypotheses with operational data from a case firm. Finally, we contextualize both findings in order to provide holistic and quantitative insights into the combined and time dependent relationships.

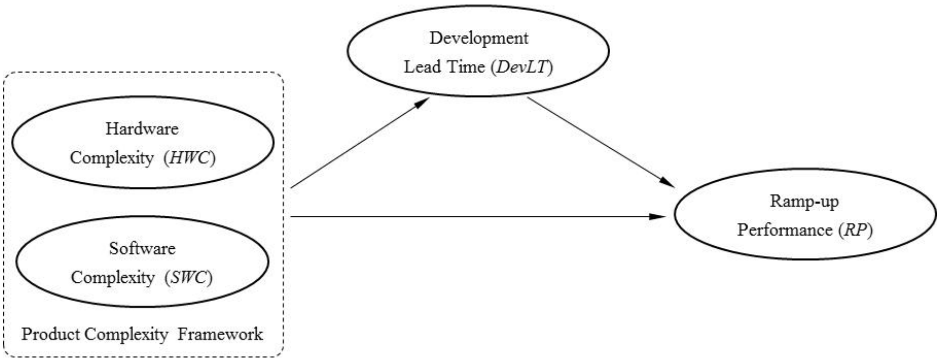


Figure1. Conceptual Framework

2 Measurement Model Operationalization

We begin with the operationalization of development lead time, an endogenous construct predicted by the product complexity framework that also acts as a predictor for ramp-up performance. The logic derives from prior research that regards development lead time as a resource (Mallick and Schroeder 2005) and hence as a critical predictor for ramp-up performance and also as a factor that is dependent on product complexity (Griffin 1993, Murmann 1994, Griffin 1997, Swink 2003).

2.1 Development Lead Time (DevLT)

Framed broadly, product development in the case firm involves five distinct activities: concept development; product planning; several cycles of design, build, integrate and test activities; product acceptance and production ramp-up. Due to general uncertainties at the start of the concept development for a particular product and due to the inherent dynamics in this phase (i.e., cancellations, redefinitions) our development lead time construct represents what Myers and Marquis (1969) call the problem-solving stage of development. This stage is separated into three key development phases. Each phase is framed by milestone reviews that denote business decision points in order to determine whether the previous development phase is completed against a set of clearly defined deliverables. Figure1 outlines the different development phases with their respective milestones and outputs.

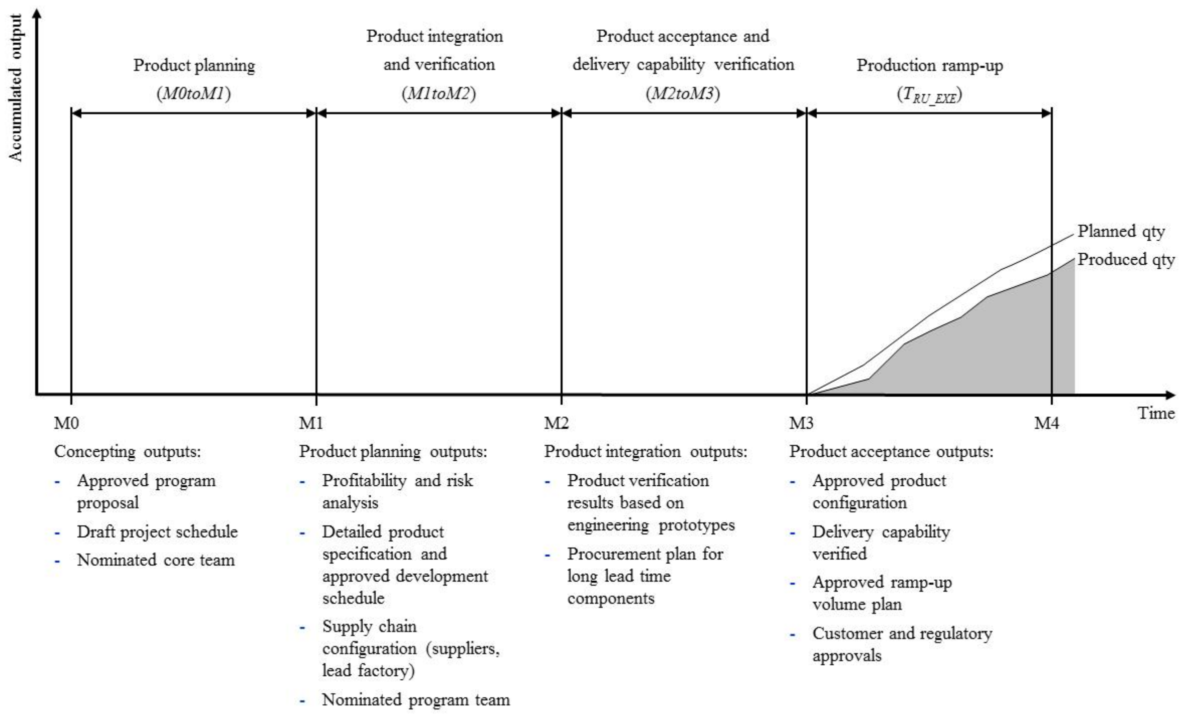


Figure1. Definitions of Major Development Phases

M0toM1. This variable represents the elapsed time – measured in days – between the M0 milestone (i.e., the end of the concept development stage) and the start of the product integration and verification phase. It characterizes the efficiency of the product planning phase.

M1toM2. The product integration stage measured as the elapsed time in days between the M1 and M2 milestone involves several cycles of design, build, integrate and test activities. This is also known as a spiral product development process because the building and testing of prototype models has become such a rapid process that the design-build-test cycle can be repeated many times (Ulrich and Eppinger, 2008).

M2toM3. During the product acceptance phase, engineering prototypes are assembled using the final assembly line and testing processes in the target factory in order to verify the performance of the production system (including supplier operations). These prototypes are also used to verify product reliability and to obtain all necessary regulatory and customer approvals. We measure this variable as the elapsed time in days between the M2 and M3 milestone.

With the approval of the M3 milestone, the production ramp-up finally starts. During this phase, the production output is determined by a push plan since detailed knowledge about sales demand does not exist at this stage. As soon as there is enough confidence to execute production according to sales forecasts, the ramp-up execution concludes. As the transition between ramp-up and mass-production is fluent, we consider different time horizons in our analysis as presented in section 5.4.

2.2 Hardware Complexity (*HWC*)

We consider hardware complexity in structural terms and hence as a property of the product (Rodriguez-Toro et al. 2004). Accordingly we apply Novak and Eppingers' (2001) definition that consists of three elements which serve as the indicators for our hardware complexity construct: (1) the number of product components to specify and produce, (2) the extent of interactions to manage between these components (parts coupling), and (3) the degree of product novelty.

Increased product functionality requires a larger number of components to be integrated into the architecture. Features like global positioning services (GPS), dual display functionality or hands-free stereo audio – to name just a few – cannot be implemented by means of software alone. However, the addition of extra components rises product complexity as a result of the more complex manufacturing process (Boothroyd et al. 1987), supply logistics (Fisher et al. 1999) or verification efforts (Novak and Eppinger 2001). While the definition of Ulrich (1995) states that a component can be any distinct region of the product, we divide our first indicator – the number of product components – into two parts to identify the relevant components in the context of ramp-up performance: *common component count* and *product specific component count*.

Common component count comprises all components from the product's bill of material list like resistors, capacitors, transistors, connectors, shields and integrated circuits, which are assembled onto the printed wiring board. These components are freely available on the market and hence also used in other products from the case company or competitors. *Product specific component count* refers to components in the bill of materials list that are specifically developed for the use in a dedicated product. Hence, the options to use these components in other products or industries are very limited. Examples are plastic covers, antenna elements, stickers, foams, gaskets, displays and cables.

Please note that our definition does not include any sub-components that are inseparably embedded in advanced components (i.e., optical lenses in camera modules or glass layers in display units) as these components are not under managerial control or technically observable.

The second indicator of our hardware complexity construct is *parts coupling*. According to Novak and Eppinger (2001), parts coupling increases complexity because an increased number of interconnected parts within a system results in additional development risks, verification efforts and side effects. In other words: the more complex the interdependencies are, the greater is the added complexity (Williams 1999). In our operationalization of parts coupling, we measure the number of signal networks across all electrical and electromechanical components (i.e., components that carry any electrical functionality and account for around 70% of all components) within a product. R&D teams have confirmed that the reliable integration of components does not only require expert knowledge regarding the components themselves but also firm understanding of their simultaneous interactions. This definition acknowledges the fact that development teams must understand and integrate arrangements of interlinked signal networks – a task that is much more challenging than the simple understanding of component pairs (i.e., individual point-to-point connections).

The third indicator of our hardware complexity construct, *product novelty* describes how much of the product must be redesigned compared to previous products. It can either be conceptualized as the percentage of new designs comprised in the product (Swink 1999) or as the degree of similarity between a certain product and other members of its product family (Coughlan 1992). Our operationalization enriches these existing concepts by measuring the percentage material value within a product that is new to the responsible development center. Products may exhibit unique characteristics depending on whether we observe them at the overall final assembly level or as individual parts and subassemblies (Ulrich 1995). Since most of the components in our study are only observable on the first layer of abstraction (e.g., cameras, displays, processors, etc.) we consider them on a macro level. In this case, any definition based on the simple percentage of new parts would regard the contribution of each component to novelty as equal. However, the integration and testing effort as well as the supply risk during ramp-up for complex cameras, displays or processors is not comparable with the risks and efforts related to simple screws, foams or stickers.

2.3 Software Complexity (SWC)

The increasingly dominant role of software in modern consumer electronics products and its impact on product complexity, development schedules and budgets (Rauscher and Smith 1995, Blackburn et al. 1996) underlines the necessity to treat software complexity as a separate construct within our product complexity framework. Because software complexity is multidimensional in nature (Banker et al. 1998, Zuse 1991), several complexity measures have been proposed. Examples are McCabe's (1976) cyclomatic complexity, Wood's (1986) component/coordination/dynamic dimensions and Halstead's (1977) effort metric. Previous research, however, has found that these measures only vary on a small number of orthogonal dimensions (Banker et al. 1993, Munson and Koshgoftaar 1991) and that they incorporate common properties (Weyuker 1988). Since this would lead to multicollinearity problems we apply a combination of indirect measures – calculated from the design specifications – and direct measures – calculated from the program code (Sunohara et al. 1981). Consistent with our approach to describe hardware complexity, we define software complexity as (1) the number of executable lines of code = SW code size, (2) the degree of software newness = SW newness, and (2) the number of software errors = SW error count. These measures refer only to the operating and application software modules that are under direct control of the firm (i.e., in-house developments). We exclude any software modules that are embedded in advanced components like displays or cameras (e.g., embedded driver software) and optional third-party applications that are not mandatory for the product launch. Thus we focus on the core elements of product software that the company has to provide timely, error free and in compliance with regulatory requirements.

SW code size. According to Huang (1998) the size of a program is one of many factors affecting its complexity. Lines of code measures have been widely discussed in the literature on embedded systems (Broy et al. 2007, Lee 2000) due to the importance of SW code size on system performance and costly memory size decisions. Within this paper we measure SW code size as the number of executable lines of code. All products in our sample are based on identical engineering tools and programming languages (i.e., a proprietary language for lower-level signaling functions and C/C++ for higher-level code). Thus, the number of executable lines of code can be counted in a consistent manner and is hence unbiased within the sample (Krishnan et al. 2000).

SW novelty. The software engineering literature argues that adding more features or increasing the newness of a software product usually increases its complexity (Zuse 1991, Krishnan and Zhu 2006, Laird and Brennan 2006). This is plausible as new software features may have lower design integrity and quality – all else being equal – than existing software code that has already been tested and debugged across existing products. Hence, we operationalize SW novelty as the number of new software features in a product that have not been included in previous products or in the existing software baseline. A new feature is characterized by the necessity to either develop/implement new software components or to modify existing components instead of reusing existing software components.

SW error count. Both, SW newness and SW code size do not sufficiently account for differences in the individual product configuration as these measures regard software components as having in-built complexities that are static and independent of their context. For example, products that contain a decent amount of SW newness and consist of a trivial code size may still require a disproportionate amount of development effort since interactions and side effects in a particular product configuration may result in a large number of errors. Because errors are strongly related to complex programs (Basili and Perricone 1984) and defects are strongly associated with software complexity (Harter and Slaughter 2003, Kafura and Reddy 1987), we include a measure of SW error count (identified during the product integration and acceptance phase while the product passes through a series of standardized acceptance tests) as our third indicator of software complexity.

2.4 Ramp-Up Performance (RP)

Based on the existing ramp-up literature there seems to be a broad consensus to measure ramp-up performance in terms of output against time or plan (Langowitz 1988, Clark and Fujimoto 1991, Terwiesch 1999, Almgren 2000, Kuhn et al. 2002, Merwe 2004, Schuh 2005). For other measures like quality, time, yield or cost there seems to be little – if any – common norm. This is partly due to the fact that each study measured ramp-up performance according to its own specific situation, characterized by different competitive priorities. For example, unit cost might be a negligible performance measure in cell phone projects as the profits gained due to successful early sales (enabled by reliable output against plan) – customers are most willing to pay premium prices during the early product lifecycle – typically outweigh all other cost related drivers. Also, it is economically unwise to focus on absolute ramp-up speed or time because quality and other cost drivers can accumulate to levels that sustainably affect the overall company competitiveness (Voigt and Thiell 2005). Hence, our second endogenous construct represents output against plan, frequently cited as effective utilization (Konopka 1995, Matsuo et al. 1997, Terwiesch et al. 1999). We measure it as the ratio between produced quantity during the ramp-up period and planned quantity at the beginning of the ramp-up period.

2.5 Control Variables

We control for several other variables to strengthen non-spuriousness between the complexity constructs, development lead time and ramp-up performance. First, we control for learning effects with the supposition that a firm gains development experience and hence the performance regarding development lead time and ramp-up will increase over time. We label this variable *linear trend* and measure it as the number of days elapsed between the ramp-up start of each new product versus the ramp-up start of the first product in the study. Second, we control for differences in the planning approach and the number of design-build-test cycles, which may dictate development lead time to a certain extent. We label this variable *planned development lead time*. It represents the number of workdays from concept approval to ramp-up start, estimated at the time of concept approval. Third, we control for whether the project experienced any delay during the product acceptance phase. We measure *slip* as the number of workdays between the estimated (provided at the M2 milestone) and the actual ramp-up start. Compared to projects that are able to ramp-up on time, delayed projects are more likely to face problems during the ramp-up phase that are related to the cause of the original delay. However, project delays can also allow program teams and suppliers to utilize the gained time productively and hence to be more successful regarding ramp-up performance. Forth, we include a sequence of seven dummy variables that are tested one after the other in order to control for plant specific effects. Although the plants in our study represent a standardized and homogeneous capacity pool, empirical studies have reported that factory characteristics may influence production performance (Hayes and Clark 1986, Langowitz 1987, Clark and Fujimoto 1991, Kuhn et al. 2002). Each of these variables is coded as one for the focus factory, zero otherwise and labeled as *factory ID1…7*. Fifth, we control for two additional plant specific effects. In order to measure the level of experience in any given plant we include a variable labeled *years in operation*. This variable represents how long a plant has been in operation prior to the ramp-up start of each product. Furthermore, we capture differences in work force culture and supply network structure that originate from differences in the physical location of each factory with regard to their main supply base that is located in China. The dummy variable *factory location* is coded as one if the plant in question is located in Asia (China and Korea) and zero if it is located in Europe (Germany, Hungary, Finland). Sixth, we add a measure for *sales forecast change* in order to control for changes in the sales volume forecast during the ramp-up execution phase. We use the relative change in the sales forecast between the beginning and end of the ramp-up execution period to remove the effects of sales fluctuation on production execution. For example, there might be low demand for a product compared to the original production plans created at the start of ramp-up. The factories will thus respond with reductions in output, which will in turn lead to low effective utilization levels although production performance itself is good. Finally, we control for *production technology novelty*, a dummy variable that is coded as one if considerable resource investments (e.g., equipment, engineering labor) are needed *and* no contingency plans exist, zero otherwise. We believe that the combination of resource investments and the absence of a back-up plan is of particular interest. We expect that production technology novelty has a negative impact on effective utilization as new technologies entail significant fine-tuning and testing and may – relative to "proven" technologies – expose a firm to risks of failure in terms of durability and reliability (Clark and Fujimoto 1991). Examples that belong to this group are decoration technologies (e.g., high-gloss paint effects), manufacturing process technologies (e.g., RoHS implementation) or production testing technologies (e.g., WCDMA testing). These technologies require considerable resource investments while prevailing technologies cannot be used as backup solutions. Table 1 summarizes the various indicators of the constructs described in this section, as well as their definitions and formulas.

Table 1. Summary of Constructs, Indicators, and Definitions

Development lead time (*DevLT*)
M0toM1 = elapsed time in days between the M0 and M1 milestone (i.e., the product planning phase)
M1toM2 = elapsed time in days between the M1 and M2 milestone (i.e., the product integration and verification phase)
M2toM3 = elapsed time in days between the M2 and M3 milestone (i.e., the product acceptance and delivery capability verification phase)

Hardware Complexity (*HWC*)

common component count = all components in the product's bill of material list that are assembled onto the printed wiring board

product specific component count = total component count – *common component count*

parts coupling = number of signal networks across all electrical and electromechanical components in the product

$$\textit{product novelty} = \frac{\text{material value of new physical components in the product}_{\text{average over period } T_{\text{RU\_EXE}}}}{\text{total material value of the product}_{\text{average over period } T_{\text{RU\_EXE}}}}$$

Software Complexity (*SWC*)

SW code size = source code size in terms of executable lines of code

SW novelty = number of features in the product that have not yet been included in previous products or in the existing software baseline

SW error count = number of reported errors during the product integration/acceptance phase

Ramp-Up Performance (*RP*)

$$\textit{effective utilization} = \frac{\textit{actual production output}_{\text{over period } T_{\text{RU\_EXE}}}}{\textit{planned production output}_{\text{at M3 over period } T_{\text{RU\_EXE}}}}$$

Control Variables

linear trend = time in days between the ramp-up start of each new product versus the ramp-up start of the first product in the study

planned development lead time = time in days between concept approval and ramp-up start, estimated at concept approval

slip = elapsed time in days between the estimated (provided at the M2 milestone) and the actual ramp-up start. Please note that this indicator becomes positive if the actual ramp-up start is ahead of the planned date.

$$\textit{factory ID1…7} = \begin{cases} 1 \text{ for the respective factory} \\ 0 \text{ otherwise} \end{cases}$$

years in operation = indicating how many years a plant was in operation before the ramp-up start of each product

$$\textit{factory location} = \begin{cases} 1 \text{ in case of Asian factories (China, Korea)} \\ 0 \text{ in case of European factories (Germany, Hungary, Finland)} \end{cases}$$

$$\textit{sales forecast change} = \frac{\textit{sales forecast quantity}_{\text{at M3 over period } T_{\text{RU\_EXE}}}}{\textit{sales forecast quantity}_{\text{12 weeks before M3 over period } T_{\text{RU\_EXE}}}}$$

$$\textit{production technology novelty} = \begin{cases} 1 \text{ considerable resource investments needed and no contingency plans exist} \\ 0 \text{ otherwise} \end{cases}$$

3 Structural Model and Hypotheses

Figure 2 illustrates our PLS path model including the previously operationalized constructs and the hypothesized structural relationships between the constructs.

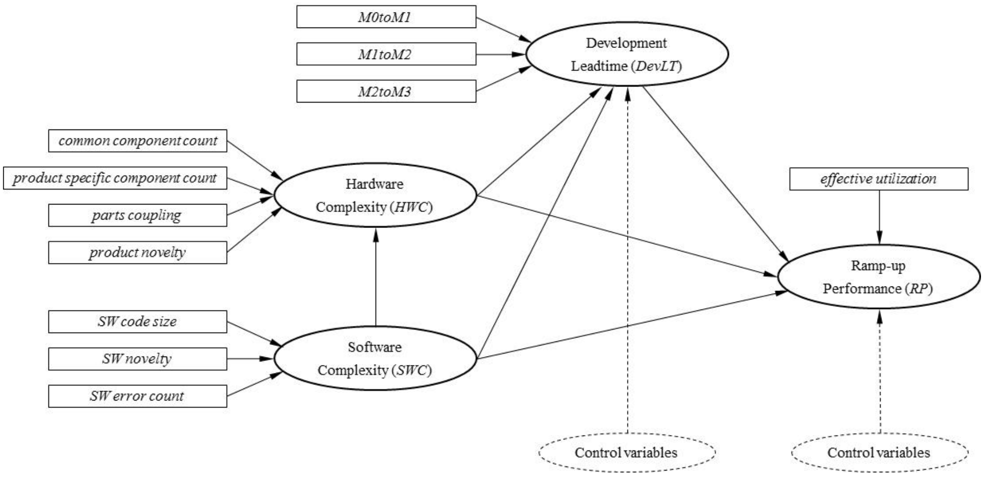


Figure 2. The PLS Path Model

All embedded systems and cell phones in particular are unique because they demonstrate a hardware-software co-design problem – hardware and software have to be designed together in order to make sure that the implementation does not only function properly but also meets all goals regarding performance, cost, and reliability (Wolf 1994). A new design is typically started with the creation of a requirements specification that includes all functional requirements – in other words the specific behavior of the system and nonfunctional requirements, including operability, certification, and cost. In a next step, an initial architecture is proposed including functions that are either assigned to be directly implemented into the hardware or into the software running on the hardware. Traditionally, hardware development dominated system development because of longer lead times and logistical dependencies on external suppliers. Consequently, software development started when hardware development was already at a stage where changes would be extensive (Graaf et al. 2003). However, hardware release cycles are expensive and inflexible, so software based implementations have become a more common approach (Lee 2000, Graaf 2003, Sangiovanni-Vincentelli and Martin 2001). Additionally, most of the new innovations in cell phone products originate from software features like social networking clients, augmented reality, picture processing or games. Hence, the hardware configuration (e.g., processor speed, memory size, interfaces, camera pixel size) is determined by a rising amount of functional requirements that are implemented into the software. We therefore hypothesize the following:

HYPOTHESIS 1. *Hardware complexity (HWC) will increase along with higher levels of software complexity (SWC).*

Several studies in the literature on operations management, product development and software engineering have found a positive association between product complexity and development lead time (Clark and Fujimoto 1989, Griffin 1993, Callahan and Moretton 2001, Swink 2003). According to Swink (2003) this is likely to be a result of the growing size and uncertainty of the design task if increasing numbers of interacting components are involved. Completing such a design task requires several design-build-test iterations or problem solving cycles until the requirements are met, regardless whether they are related to software or hardware. As a result, lead time in a program will be affected by the length of each problem solving cycle, the number of iterations and patterns of informational linkage among the cycles (Clark and Fujimoto 1989). Although these arguments are valid for both complexity dimensions, hardware complexity may dominate as a result of the longer development lead times for components. On the other hand, software can also play a dominate role, due to the steady shift of functionality from hardware to software (Lee 2000, Graaf 2003, Sangiovanni-Vincentelli and Martin 2001) and due to the tendency to fix hardware errors via software solutions (Rauscher and Smith 1995, Graaf et al. 2003). These arguments lead to the following two hypotheses:

HYPOTHESIS 2. *Higher levels of software complexity (SWC) are associated with longer development lead times (DevLT).*
HYPOTHESIS 3. *Higher levels of hardware complexity (HWC) are associated with longer development lead times (DevLT).*

Because the replication process of software in embedded systems is simple (i.e., it typically consists of a simple programming step during production) we could conclude that manufacturing performance can no longer be adversely affected by the level of software complexity after the software has been released for production. However, the manufacturing environment for cell phones is characterized by complex multistage assembly lines and includes several test phases. Potential drivers for ramp-up performance include the tight interdependence between these automatic test systems and the product software as well as the on-time availability of approved software variants. Addressing these issues one at a time, we note that complex product software necessarily increases the test software algorithms (Schaub and Kelly 2004). Firstly, due to the increased number of test steps in order to calibrate wireless protocols or other components and secondly, due to the interactions between the product software and the test system that also have to be managed. Both conditions may lead to decreased yield and hence to decreased production output. In addition, production output is dependent on the scheduled readiness of customer specific software variants. Complex software, however, makes it more likely that customers revise requirements (Thomke and Bell 2001), find new errors after pretesting or request changes in the customization options. Manufacturers are often forced to adapt or lower production plans due to these circumstances. Therefore we state the following:

HYPOTHESIS 4. *Higher levels of software complexity (SWC) are associated with lower ramp-up performance (RP).*

A rich stream of studies has demonstrated that product complexity in terms of physical product characteristics is negatively associated with ramp-up performance (Langowitz 1987, Swink 1999, Terwiesch et al. 1999, Vandevelde and Van Dierdonck 2003, Keil et al. 2007). Higher levels of complexity and uncertainty regarding product and technology will cause more difficulties in the attempt to realize a smooth production start-up (Vandevelde and van Dierdonck 2003). Complex products may push the limits of manufacturing process capabilities and require more process steps, thus opening up more opportunities for process failure (Swink 1999). At the same time extra learning efforts regarding manufacturing engineering and improved operator training are required to achieve the desired performance level. Engineering changes may appear more frequently and cause the most disruptive effects at the very beginning of the manufacturing start (Coughlan 1992). Similar challenges are also likely to occur at upstream partners (i.e., suppliers), thereby disturbing ramp-up performance due to problems with material supply and quality (Almgren 2000). For these reasons we state the following hypothesis:

HYPOTHESIS 5. *Higher levels of hardware complexity (HWC) are associated with lower ramp-up performance (RP).*

Cell phone projects involve different organizational functions that are concurrently conducting development tasks in order to minimize development lead time. However, prior research has been inconclusive regarding the actual effectiveness of process concurrency or its accelerating effect on operational outcomes (Tatikonda and Montoya-Weiss 2001). For example, Wheelwright and Clark (1992) found that concurrency is beneficial to operational outcomes other than lead time because more information can be shared between organizational functions. Improved understanding of requirements and limitations enables cooperative problem solving activities as well as earlier anticipation of project challenges. On the contrary, concurrency may lead to increased risk if the downstream function is forced to make decisions (such as orders of product specific manufacturing equipment) before final fixed data is available from the upstream function. This can delay production line approvals and cause rework — ultimately leading to missed output targets. Furthermore, the longer the time available to study user needs and develop and test alternative concepts for technical feasibility, the greater is the likelihood that a better solution will be found (Mallick and Schroeder 2005). Also, extended development lead times may be the result of additional problem solving cycles or pilot production rounds that improve product and process maturity levels and hence facilitate a smooth production ramp-up. We posit the latter effect and conjecture our last hypothesis:

HYPOTHESIS 6. *Longer development lead times (DevLT) are associated with higher ramp-up performance (RP).*

Please note that we explore the effects that are described in hypotheses 4, 5 and 6 under different time horizons of the ramp-up execution period in section 5.4. For brevity we do not discuss the related hypotheses.

4 Methodology and Research Setting

In order to test our hypotheses, we employ a variance based structural equation modeling approach known as PLS path modeling. Compared to covariance based structural equation modeling approaches (e.g., LISREL), PLS is particularly well suited for studies using operational data and if the primary research objective is the maximization of explained variance in the endogenous constructs (i.e., prediction) instead of achieving model "fit". Operational data frequently violates the requirement of multivariate normality and sample sizes are limited by the number of real life cases. PLS, however, does not make any assumptions of the underlying distribution and provides stable estimates even if the ratio of observations to parameters is small (Wold 1982, Fornell and Bookstein 1982, Chin 1998, Hair et al. 2011). In addition, PLS allows for the simple configuration of formative measurement models. Consistent with the decision rules by Jarvis et al. (2003) and Petter et al. (2006) our indicator operationalization advocates a formative coding scheme. Increasing the value of any indicator in a construct will directly translate into a higher or lower score for the overall construct scale, regardless of the value of other indicators. However, this has methodological implications. The concepts of internal consistency, reliability and convergent validity are not meaningful if formative indicators are involved (Hair et al. 2011). Hence, we base our formative indicators on theoretical rationale as outlined in section 2 and focus on the pure operational aspects of product complexity, development lead time and ramp-up performance. This allows for conceptually rigorous and complete definitions supported by statistical criteria. Before we applied the algorithm we standardized our data (mean = 0, variance = 1) in order to support the interpretation of the path coefficients. Although not a precondition in PLS, data standardization is recommended if the variables are measured with different scales as in our case.

4.1 Research Setting

The unit of analysis for our study is the individual cell phone, designed and manufactured by Nokia, a Finnish corporation headquartered in Espoo, Finland. With its strong R&D presence in 16 countries and around 130.000 employees worldwide, Nokia's strategic direction aims to integrate content, applications and services into its mobile computers, smartphones and mobile phones. In 2010, Nokia sold 453 million mobile devices and is therefore particularly interested in predictable and rapid ramp-ups in order to avoid lost sales and to quickly release R&D teams into new product development activities.

4.2 Data Collection

We collected project data from 46 distinct cell phone projects between 2005 and 2008. In order to avoid biased results we randomly draw samples from R&D sites in Denmark, Finland, China and Germany to cover a wide variety of price points, form factors and total sales volume ranges. As a result of our close cooperation with the case company (one of the authors is employed as a ramp-up manager by Nokia) we were able to employ multiple methods of data collection and multiple data sources (i.e., interviews, project documentation, data archives, company reports, and questionnaires) for each product. Only two projects were dropped because of incomplete data panels.

The data for calculating the product development lead time indicators are taken from a project management reporting tool that records all planned and actual milestone dates throughout the project lifetime. Programs can only move to the next development phase and update the milestone dates if certain, well defined criteria are met and the respective milestone review is approved by the steering group. This property allows us to precisely calculate the duration of different development phases and other timing related variables (linear trend, planned development lead time, slip).

The data for our hardware complexity indicators – specifically common component count and product specific component count – were taken from the product data management system that is used for managing the bill of materials list. Our product novelty indicator was calculated with the help of the product specification documentation and the sourcing parts lists. The first tool provides an overview of the novelty/re-use status of each component while the second provides the material prices. The parts coupling indicator originates from the electronic design automation (EDA) tool.

The software related indicators were extracted from the compiler log files (SW code size), the requirements management system (SW novelty) and the error management databases (SW error count). Nokia products generally follow a highly disciplined and standardized software development and verification process that creates reliable and consistent data regarding the aforementioned software characteristics.

The variables regarding effective utilization and sales forecast change were calculated using data from a management information system report that contains sales forecasts, production plans and actuals on a monthly time scale. As a response to the short-term planning cycle of Nokia that spans a timeframe of 12 weeks we decided to peg our ramp-up execution period T_{RU_EXE} to this time horizon in the initial model which we relax in section 5.4. During this period most resources are considered to be fixed due to equipment and material procurement lead time limitations.

Finally, we enhanced our data with interviews, a written questionnaire that was completed by each ramp-up manager and milestone review documentation. Using these sources we were able to gain additional insights into the qualitative issues that appear during each ramp-up and to gather data in order to operationalize our production technology novelty indicator. Descriptive statistics for our data and their correlations are presented in the appendix. All variables were screened for abnormal observations to avoid outlier bias in the PLS calculations. Due to confidentiality reasons the variable SW error count is normalized to have a mean equal to 1000.

5 Results

Following the structural equation modeling logic, the assessment of a PLS model follows a two-step approach that involves separate assessments of the measurement model and the structural model (Hair et al. 2006, Hair et al. 2011). All parameters within the model were estimated using smartPLS (Ringle et al. 2005) and XLSTAT version 2011.2.01. Additional statistics were calculated with Minitab version 16.1.1.

5.1 Validation of the Measurement Models

Formative measurement models are examined based on their indicator weights, their significance, their loadings and the degree of multicollinearity (Chin 1998, Tenenhaus et al. 2005, Hair et al. 2011). Significance levels were estimated by means of a *t*-statistic that is generated by a bootstrapping technique (based on 500 resamples). Additionally, we evaluated significance levels by reviewing bias and percentile ranges of the bootstrap output. We also used the conservative “construct level sign change” option (initial weight setting = 1.0, abort criterion = 1.0E-5) after checking the individual bootstrap results and applied two tail tests although the direction of our hypotheses would allow for single tail tests. We calculated the variance inflation factor (VIF) for the assessment of multicollinearity. As the highest VIF in our measurement model turned out to be around two, multicollinearity is not likely to distort the estimates as a result of excess redundancy. In the context of PLS, the critical cut-off value for VIF is 5 (Hair et al. 2011).

Table 2 presents the results of our most conclusive measurement model (M10). Other measurement models that are part of the structural model variants that will be discussed in the next section are only provided in the appendix since their difference is negligible. All formative indicators have strong and significant weights – with the exception of common component count – and there is no co-occurrence of negative and positive indicator weights in the same construct. Note that single-indicator constructs always appear significant (as used for the control variables and ramp-up performance). This suggests that common component count does not provide additional explanatory power beyond the other indicators within the hardware complexity construct although it still represents an important aspect (the loading or bivariate correlation is substantial and significant). In other words, common component count is *absolutely* important but not *relatively* (Cenfetelli and Bassellier 2009). Its theoretical relevance, however, is justified as there are only few overlaps with the other indicators leading us to keep the indicator in the model. The novelty indicators in particular (product novelty and SW novelty) deserve closer attention. Their relative contribution to the complexity constructs is disproportionally high, emphasizing the importance of novelty as a key characteristic of complexity that can be used to predict development lead time and ramp-up performance. In summary, our measurement model is characterized by robust and significant indicators that capture the domain of our constructs. This provides a strong foundation for construct validity.

Table 2. Measurement Model Results

	M10 (final model)			
	Path weight	*t*-value	Loading	VIF
Development lead time (*DevLT*)				
M0toM1	0.62 ***	5.73	0.76	1.05
M1toM2	0.75 ***	3.60	0.59	1.42
M2toM3	0.53 ***	2.79	0.16	1.37
Hardware complexity (*HWC*)				
common component count	0.07	0.56	0.45	1.65
product specific component count	0.28 **	2.18	0.63	1.53
parts coupling	0.38 **	2.29	0.68	2.14
product novelty	0.71 ***	4.76	0.84	1.09
Software complexity (*SWC*)				
SW novelty	0.54 ***	3.85	0.77	1.16
SW code size	0.40 ***	3.22	0.52	1.02
SW error count	0.51 ***	5.18	0.74	1.15
Ramp-up performance (*RP*)				
effective utilization	1.00	--	1.00	1.00

Notes: N = 46.

*** p ≤ .01, ** p ≤ .05, * p ≤ .10, two-tailed tests based on 500 bootstrap resamples.

5.2 Validation of the Structural Models

The primary evaluation criteria for the structural model are the R^2 measures (=predictiveness) and the level and significance of the path coefficients (Chin 1998, Tenenhaus et al. 2005, Hair et al. 2011). Goodness of fit (GoF) indices or blindfolding procedures (Q^2) are only applied in the presence of reflective measurement models as they are based on the portion of explained variance in the indicators (see Stone 1974, Geisser 1977, Tenenhaus et al. 2005, Hair et al. 2011). All structural model parameters are estimated with the path-weighting scheme (for a discussion of weighting schemes, see Chin 1998, Tenenhaus et al. 2005). Despite the slightly increased VIF value of SWC → RP we discovered no inconsistency in the results during our multicollinearity assessment. With this assessment we tested additional models by removing one complexity construct at a time.

Table 3 reports the results of our hypothesis tests with regard to our most conclusive model (M10). Further results related to the models M1 through to M10e are provided in the appendix. In model M1 we summarize the effects of non-significant control variables (sales forecast change, years in operation, production technology novelty and linear trend – which is initially significant but disappears in the presence of planned development lead time). In M2 we control for planned development lead time and slip. Both variables turned out to be strong and significant. M3 through M9 each include one factory ID variable to strengthen our claim of non-spuriousness and to justify the inclusion of our last control variable — factory location. In the models M10a through to M10d we tested alternative forms of M10. Compared to the original model in M10 we considered time horizons of 4, 8, 16 and 20 weeks in M10a, M10b, M10d and M10e for the calculation of effective utilization.

Table 3. Structural Model Results

	M10 (final model)		
	Path weight	*t*-value	VIF
Path coefficients			
Direct effects			
SWC --> HWC	0.89 ***	30.11	1.00
HWC --> DevLT	-0.05	0.36	5.08
SWC --> DevLT	0.36 *	1.93	5.91
HWC --> RP	-0.39 *	1.69	5.40
SWC --> RP	-0.44	1.47	7.87
DevLT --> RP	0.36 *	1.90	2.02
Control variables			
planned development lead time --> DevLT	0.74 ***	7.59	1.40
slip --> RP	0.27 **	2.30	1.24
factory location --> RP	0.22 **	2.53	1.20
Total effects			
HWC --> RP	-0.41 *	1.74	--
SWC --> DevLT	0.31 ***	2.86	--
SWC --> RP	-0.68 ***	5.06	--
Coefficient of determination (R^2)			
HWC		0.79	
DevLT		0.87	
RP		0.48	
RP (R^2 adj)		0.41	

Notes: N = 46.
*** p ≤ .01, ** p ≤ .05, * p ≤ .10, two-tailed tests based on 500 bootstrap resamples.

5.3 Key findings

There is support for our hypothesis that hardware complexity is dependent on the level of software complexity (H1). This result corroborates the view that feature implementations are gradually shifting towards software implementations that can run on more generic hardware. Development still follows a hardware-software co-design approach but most of the new innovations in cell-phones originate from new software features and hence determine the hardware requirements. We observe a negative and significant relationship between software complexity and development lead time (H2) while controlling for planned development lead time. This is an interesting observation and it suggests that software characteristics determine the time expenditures to develop a cell phone. These two findings are important since they demonstrate the important role of software in the development process of new cell phones. By contrast, the effect of hardware complexity on development lead time is not statistically different from zero (H3). One possible explanation is that software complexity is the main and dominant predictor of development lead time. Hence, hardware complexity is downgraded to a mere enabler without additional explanatory power beyond the software complexity influence. Furthermore, our results indicate, on average, a negative effect of software complexity on ramp-up performance (H4) although the replication process was initially considered to be negligible. While the direct effect is non-significant there is a significant and negative total effect, suggesting that any effect of software complexity on ramp-up performance is mediated by hardware complexity (as illustrated in Table 4). We provide data for two commonly used methods, the product-of-coefficients approach (Sobel 1982) and the bootstrapping approach (Preacher and Hayes 2008). While the product-of-coefficients approach requires multivariate normality (Preacher and Hayes 2008) we are in favor of the bootstrapping approach that provides a nonparametric approximation of the sampling distribution regarding the indirect effect. It is thus more consistent with the nonparametric world of PLS.

Table 4. Results of the Mediator Analysis

		Product of coefficients results		Bootstrapping results (90% CI)	
Mediator	Point estimate (indirect effect)	*SE*	*Z*	Percentile lower	Percentile upper
HWC	-0.35 *	0.20	-1.72	-0.71	-0.06
DevLT	0.13	0.11	1.18	0.00	0.35
HWC & DevLT	-0.02	0.06	-0.30	-0.12	0.07

Note: * $p \leq .10$, two-tailed test based on 500 bootstrap resamples.

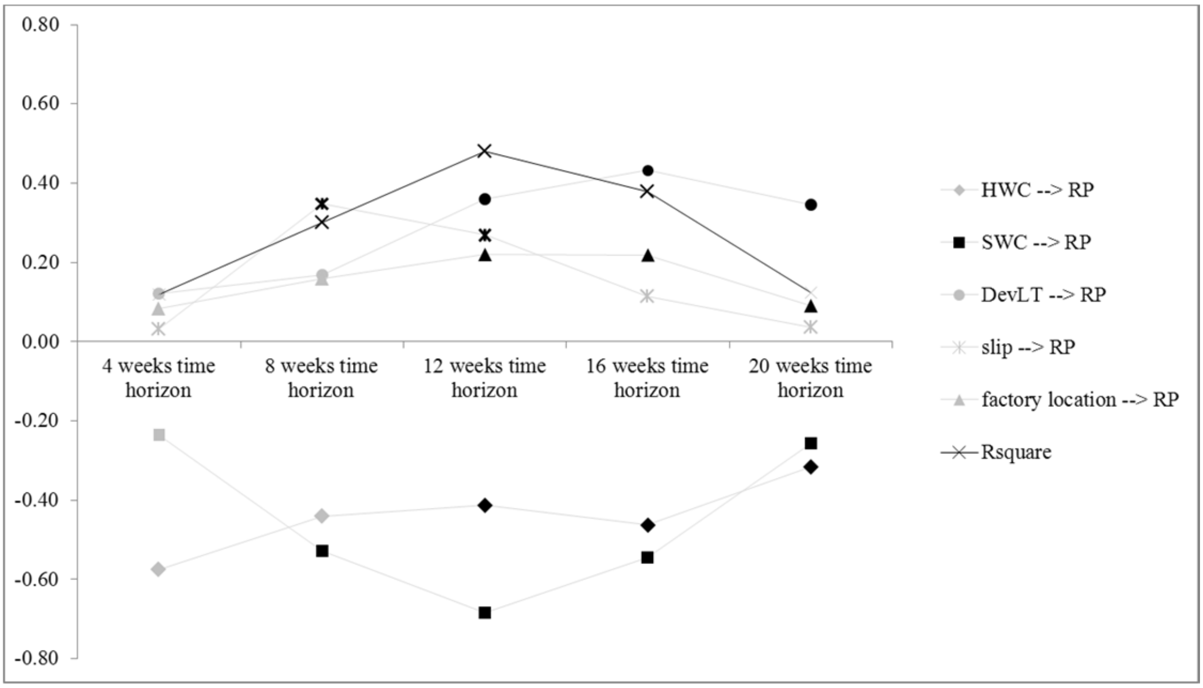
As a result, ramp-up performance is dependent on software complexity — e.g. by manufacturing test system yield or software variant readiness — but the effect is mediated through hardware complexity. This is supported by our finding that hardware complexity has a significant direct effect on ramp-up performance (H5). We will discuss this effect in more detail in the next section. In any case, our results suggest that the physical characteristics of a cell phone have a negative impact on performance levels in the form of material supply/quality problems or more sophisticated and hence error prone manufacturing set-ups. Likewise, there is empirical support for hypothesis 6, stating that ramp-up performance decreases as development lead time increases. This finding is in line with our argument that extended development lead times may allow for additional problem solving or pilot production cycles that in turn correspond with improved product and process maturity levels and ultimately result in improved ramp-up performance.

Of the significant control variables, planned development lead time has a positive significant relationship with development lead time. Because software complexity is still significant while controlling for planned development lead time, this suggests that early plans can only provide an imperfect prediction of the actual development lead time. Our results also indicate that products experience a higher ramp-up performance on average if they are exposed to late schedule slips. This is likely due to the instance that this sudden and precious gain of time is used for improvements in the material and production status. The results from the factory ID analyses suggest the absence of spurious effects related to plant specific effects that may explain the presented results. All path coefficients remain stable under the individual insertion of each factory ID variable although we see a pattern in the relative influence of factory ID on ramp-up performance dependent on the location of the respective factory. On average, Asian facilities tend to perform better than their European counterparts – a finding which is supported by the significant effect of factory location on ramp-up performance.

5.4 Lifecycle analysis

Due to the dynamic nature of the ramp-up phase an exploration of the lifecycle behavior is justified. We tested several variants of the above model (M10) for which we used different time horizons in order to calculate the ramp-up performance indicator. The various model parameters are recorded in Figure 3. Starting with the model parameters that are calculated over a ramp-up execution period of 4 weeks (T_{RU_EXE} = 4 weeks) we find non-significant path coefficients and only a marginal R^2 value. In other words, our model fails to predict the very early ramp-up phase. This is most likely the result of the chaotic and dynamic environment at this stage that is only partially ascertainable by formal models. Next, we analyze the parameters over an 8-week period and hence achieve two additional insights. First, the increased R^2 indicates predictive relevance and second, slip becomes strong and significant. Yet, the positive effect of slip on ramp-up performance fades out in subsequent models. We will outline the potential mechanism of this effect in the next section but this suggests that factories build up product buffer while they are waiting for the final sales start which in turn helps them to fulfill higher performance levels during subsequent ramp-up stages. Moving further to the right on the figure (T_{RU_EXE} = 12 weeks) we find the model parameters illustrated that were already discussed previously. Additionally, this model displays the highest R^2 value. In order to explain the results beyond the 12 weeks' time horizon we experimented with additional variables and discovered that longer time horizons require additional and slightly different predictors. As an example, production plan adaptations and hence production output changes are more likely to occur over longer time horizons in comparison to short term plans in which most resource availabilities are rather fixed. As one would expect, ramp-up specific difficulties diminish and the focus shifts to factors that are related to mass volume planning.

Figure 3. Lifecycle Analysis of Different Model Parameters (Total Effects)



Note: bold symbols indicate significance, numerical details are provided in the appendix.

6 Discussion and Managerial insights

Although extant operations management and product development literature implies that ramp-up performance plays a role in the overall performance of a company (Clark and Fujimoto 1991, Pisano and Wheelwright 1995, Hatch and Mowery 1998, Kuhn et al. 2002) we have little systematic and quantitative understanding of the factors that are affecting ramp-up performance. Our findings contribute to this relatively understudied research area in analyzing the combined and lifecycle dependent quantitative relationships between development lead time, product complexity, slip and ramp-up performance.

First, our findings suggest that the mechanisms by which complexity characteristics affect ramp-up performance differ significantly. Hardware complexity seems to be a continuous and steady source of difficulty for manufacturing during the whole ramp-up phase as identified in our lifecycle analysis. This suggests that problem solving cycles related to physical characteristics (e.g., design faults that create assembly failures or low yield levels at supplier operations) are slow and cumbersome. While the root causes might be identified quickly; the actual engineering change is mostly tardy as solutions often require new or improved materials but long lead times or excessive pipeline inventory slows down the implementation process. Based on our experience this may take up to several weeks or even months during which production teams have to cope with the current situation and hence often fail to deliver output on plan. Software complexity on the other hand demonstrates a completely different lifecycle pattern. After the production release of the initial software and the start of the replication process we do not experience any negative effect of software complexity on ramp-up performance. However, as soon as the initial phase is over — a phase in which only a limited amount of customers have been served — we find a considerable negative influence on ramp-up performance. At least two assumptions are consistent with this result. New and complex cell phone software must be compatible with other products in the network (Chiesa et al. 2002) and frequently requires changes on the network side. Often, these modifications are not simultaneously available in all networks and hence pose a potential delay for second wave approvals. Also, the creation and verification of customer specific variants is a more complicated process in the presence of high software complexity. More options are available and side effects are more likely to occur thus leading to delays in production execution and to decreased ramp-up performance. However, the overall problem solving cycles regarding software issues are shorter than those caused by hardware problems since the negative effect of software complexity on ramp-up performance fades out rather quickly. This is consistent with the software variant creation release cycle in Nokia that may take several days or weeks at the most.

Second, this study sheds light on the relationship between product complexity and development lead time in the cell phone industry. The context is representative for consumer electronics where the influence and importance of software has increased substantially in the past decade and time to market is of critical importance. As already lined out in the introduction of this paper, the majority of new features are developed in software and hence determine the development schedules to a large extent. Examples are browsers, mobile TV, gaming, augmented reality or dual SIM functionality as well as the option to provide and access local relevant content (e.g., social networks). Hardware on the other hand becomes a commodity available from various sources and in different configurations. This circumstance requires the involvement of purchasing managers in the development process. However, since purchasing managers have a strong focus on material cost, their involvement can sometimes correlate negatively with the adherence to development schedules. For example, the implementation of second source components or the adjustments of existing code to cost efficient but slow processors require considerable time and effort for coding, testing and the implementation. Another reason for the strong effect of software complexity on development lead time – while controlling for planned development lead time – might be related to the presence of planning fallacy. Software teams seem to make decisions based on delusional optimism rather than on a rational weighting of gains, losses, and probabilities (Lovallo and Kahnemann 2003). This often results in planned development lead times that accentuate positive assumptions as they were created in an intuitive and unobjectionable process and are therefore often unlikely to hold.

Third, the results highlight that the detrimental effects of product complexity are exacerbated by shorter development lead times. Since shorter development lead times mean that project activities have to be executed faster than normal, project managers are less able to predict activity outcomes, thus on-time performance is more difficult to achieve (Swink 2003). In particular and as mentioned above, development lead times are driven by software complexity. As a consequence, this drives hardware development activities away from the critical path enabling projects to fine-tune hardware activities while waiting for the software release. For example, suppliers that are responsible for physical parts (e.g., plastic covers) can run multiple test-batches until they are sure that the process works faultlessly and that they can match the desired quality levels. This is in contrast to projects with highly compressed schedules in which fine-tuning activities often continue during the ramp-up phase, hence leading to discontinuous output and quality instabilities. Also, we believe that more design-build-test cycles or additional time for evaluation activities between the cycles (e.g., for second source materials) may lead to higher product maturity levels and ultimately to stronger and more sustainable ramp-up performance as indicated by the lifecycle effect of development lead time on ramp-up performance in Figure 3. According to Wheelwright and Clark (1992) prototyping and its role in design-build-test cycles is a core element of development and an area that offers major opportunities for management to improve the effectiveness and efficiency of their development process.

Finally, we observe a positive relationship between slip and ramp-up performance which supports our argumentation that late schedule slips are advantageous for ramp-up performance. There are several possible theories for why this relationship exists. Most late schedule slips are the result of delayed software approvals or material deficiencies (Almgren 2000, Kuhn 2002). For example, the flexibility of software makes it vulnerable to late additions or changes in order to correct for hardware problems (Rauscher and Smith 1995) or in order to quick-fix errors that are detected during the product acceptance phase. Material deficiencies are more likely to occur at the start of the ramp-up because suppliers switch from pre-production tools to mass-production tools for which only limited experience exists. In any case, these issues occur late, often unexpectedly and may lead to ramp-up delays. On the other hand, there is a strong managerial tendency, motivated by higher gross-margins of new products, to start production gradually. Despite lost capacity for existing products and rising inventory levels for semi-finished products the potential gains in production experience, material quality and new product availability (semi-finished product buffers can be converted into final products via simple assembly and "re-programming" steps as soon as the necessary approvals are in place) compensate for negative aspects. According to Terwiesch and Bohn (1998), running engineering trials and efforts regarding the improvement of yield and production speed at the beginning of the ramp-up phase might limit the uptime in the short run, but will often lead to an increased performance during the rest of the ramp-up period. In other words, late schedule slips do not only enable factories to ramp-up faster after the actual ramp-up start but it also enables them to boost their throughput as demonstrated by the strong impact of slack at the beginning of the ramp-up period. However, as soon as experience levels saturate and buffers are used up as shown in the lifecycle analysis this positive effect on ramp-up performance fades out.

While the primary focus of this research is to predict ramp-up performance levels and to explain the phenomena around this subject, several managerial issues and practical implications arise from the work. Our findings highlight the drivers of development lead time in cell phone projects. While planning accuracy is affected by the presence of planning fallacy in the software domain and while it is best tackled by taking an outside view (Lovallo and Kahnemann 2003), the actual development lead time is determined by software complexity. Thus, the managerial implication of this finding is that shorter development lead times are more likely to be achieved if firms focus on strategies to either cope with software complexity or to decrease it instead of decreasing hardware complexity by using fewer and less novel components. A common approach in time-paced markets where new products or upgrades are released on a regular basis is to slip everything that cannot be completed in time to the next product in the sequence (Eisenhardt and Brown 1998). Yet we do not argument that all sources of software complexity are counterproductive and must therefore be eliminated or reduced to lower levels. We rather claim that firms – especially if operating in highly competitive and dynamic markets – need to understand the potential impact of their choices and use available best practices (e.g., Swink 1998) in order to accommodate to the higher levels of complexity that the market imperatives entail. In this sense, software postponement may be a successful approach, although it received only little attention. In other words, manufacturing and production testing activities should be independent of the product software. They should rather use a basic core software for manufacturing before the final customer specific software package is programmed into the phone at the last possible step in production. This allows firms to ramp-up production of semi-finished products as soon as the product hardware is ready even if the final software is not yet released for production. This process continues until a maximum buffer level is reached or until the final software package is released for production. In a next step the semi-finished products are quickly converted into end products via simple re-programming steps just before packing. While this approach does not necessarily decrease development lead time it increases early product availability and hence ramp-up performance. This is partly due to the potential consumption of product buffers but also due to advances in pre-ramp-up learning. However, this effect is dynamic and diminishes after a 12 weeks' time horizon.

Also, our conclusions underscore the importance of the right product-factory fit and the relevance of software variant management. We feel that the division of complexity into hardware and software elements provides a suitable approach to guide management decisions in order to apply the most efficient strategy for any particular product. For example, products with high hardware complexity may gain considerably from DFM activities, early manufacturing involvement and a careful selection of the lead factory with particular focus on the right product-factory fit — the fit between the demands a product is likely to make upon a factory and the existing competence of the factory to which it is to be introduced (Langowitz 1988). Such an approach, however, is unlikely to be successful in coping with software complexity as this would rather require efficient and proactive variant management activities.

Finally, our findings underscore the importance of development lead time management with regard to ramp-up performance. Managers in the consumer electronics industry are tempted to accelerate product development in order to launch products just before special events such as industry trade shows or high peak sales periods (e.g., Christmas). While the timing of revenues critically depends on development lead time (time-to-market) firms must be careful not to over-accelerate product development. Over-acceleration can have a significant negative impact on customer relations and competitor market share if firms fail to achieve required ramp-up performance levels (time-to-volume). Although it is known that the fast introduction of high-tech products to the market helps to achieve overall commercial success (Mallick and Schroeder 2005), creating too much overlap between phases makes it difficult for teams to anticipate changes possibly resulting in products that are not optimized for volume manufacturing by the time of their launch (Krishnan et al. 1997). For example, products that are targeted for Christmas sales (i.e., ramp-up start in September) may not gain much from overly compressed schedules and an earlier product launch in August when the demand is still only moderate. In such a case, firms may gain more if they allow their teams to execute as planned and to focus on ramp-up performance since the volumes that are missed during the Christmas period are ultimately lost. However, the relationship between gains in ramp-up performance and speed of product development is complex and deserves further research attention in order to weigh up launch timing against development completeness (Kalish and Lilien 1986).

7 Conclusions

This paper examines the quantitative interrelationships between development lead time, product complexity, slip and ramp-up performance over different time horizons. We complement previous studies regarding ramp-up performance in three ways. First, we demonstrate that software complexity is the dominant driver for development lead time in cell phone projects. Second, we identify development lead time as an important predictor for ramp-up performance. While longer development lead times facilitate higher product maturity and thus sustained ramp-up performance, later market introductions of new products imply a negative impact on revenue inflows. Third, our model also suggests that late schedule slips, although disastrous for customer relations in which due dates are crucial, provide the opportunity to build up (semi-finished) product buffers which in turn increase the initial ramp-up performance. In conclusion, we contribute to the field of operations management by offering a more comprehensive and enriched understanding of the drivers for development lead time and ramp-up performance in the cell phone industry. We also contribute to the existing research by providing an alternative view into the specific and lifecycle dependent effects of development lead time, product complexity and slip on ramp-up performance. Thus, our insights support management efforts to anticipate the consequences of product design decisions, to predict development schedule risk levels and to make informed decisions about production volume commitments.

Although our findings are firm specific we believe that our results can also be generalized to fit the wider consumer electronics industry because: (1) modern consumer electronics products like hi-fi systems, game consoles, cameras and flat screens share similar product and development characteristics with cell phones, (2) our operational data were taken from different geographical development centers with different cultural and managerial properties and (3) our results extend existing empirical work from other industries.

While this study makes a significant contribution to the academic literature and provides guidance for managerial practice, there are also limitations that provide opportunities for further research. First, our conclusions are based on a limited number of real life cases that were carried out in a single firm. A wider analysis including different firms from the consumer electronics industry utilizing a larger sample size might reveal additional effects and thus potentially enhance the capability to make generalizations that exceed the scope of the consumer electronics industry. Second, with respect to the predictors that are involved in our study we believe that there are further factors such as organizational forces or supply chain elements (as outlined in the literature review) that may contribute to development lead time or ramp-up performance. A fruitful extension could identify and specify these factors and assess their impact on the given model parameters to advance our theoretical understanding of the contingent relationships. Third, as we mentioned earlier, firms need to make a conscious choice between project acceleration and ramp-up performance. This circumstance prompts for formal models to find the optimal level of project acceleration that maximizes the total revenue inflow for given levels of product complexity. Finally, future research should revisit the effectiveness of knowledge transfer across projects although we did not find an impact of linear trend on ramp-up performance in our study. We might expect to find a weak integration of knowledge from past related projects, since products develop through a sequence of changes that tend to build on past experience (Clark 1985). Therefore, the operational and strategic importance of knowledge management (Sherman et al. 2005) deserves special attention, most notably with respect to knowledge transfer across projects and across sites.

Appendix:

Descriptive Statistics and Correlations (Pearson) Between Variables

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1. *M0toM1*	1.00																	
2. *M1toM2*	0.14	1.00																
3. *M2toM3*	0.24	0.23	1.00															
4. *common component count*	-0.10	0.19	0.33	1.00														
5. *product specific component co*	0.31	0.07	0.08	0.36	1.00													
6. *parts coupling*	0.24	0.07	0.22	0.62	0.58	1.00												
7. *product novelty*	0.18	0.55	0.42	0.25	0.22	0.24	1.00											
8. *SW novelty*	0.12	0.65	0.50	0.31	0.28	0.22	0.83	1.00										
9. *SW error count*	0.33	0.26	0.41	0.34	0.28	0.40	0.54	0.36	1.00									
10. *SW code size*	0.44	0.11	0.06	0.18	0.66	0.60	0.12	0.14	0.08	1.00								
11. *effective utilization*	-0.08	0.00	-0.12	-0.32	-0.42	-0.52	-0.34	-0.26	-0.53	-0.23	1.00							
12. *linear trend*	0.33	-0.42	-0.30	-0.13	0.31	0.34	-0.29	-0.48	0.03	0.47	-0.03	1.00						
13. *planned development lead time*	0.73	0.58	0.47	0.07	0.12	0.19	0.37	0.38	0.43	0.19	-0.07	-0.10	1.00					
14. *slip*	0.05	0.21	0.61	0.16	0.15	0.05	0.49	0.48	0.28	0.08	0.00	-0.08	0.09	1.00				
15. *sales forecast change*	-0.15	0.15	0.01	0.08	-0.24	-0.09	-0.09	-0.01	-0.09	-0.13	0.21	-0.44	0.02	-0.13	1.00			
16. *years in operation*	0.04	-0.10	-0.13	-0.36	-0.39	-0.35	-0.24	-0.18	-0.09	-0.28	0.19	0.09	0.03	-0.07	0.09	1.00		
17. *production technology novelty*	0.06	0.19	0.02	0.21	0.04	0.01	0.20	0.05	0.23	-0.14	-0.14	-0.12	0.20	0.16	0.17	0.10	1.00	
18. *factory location*	0.20	-0.29	-0.31	-0.36	0.08	-0.06	-0.29	-0.46	-0.31	0.25	0.34	0.53	-0.10	-0.17	-0.09	-0.12	-0.15	1.00
Mean	84.80	106.83	52.67	350.35	149.70	555.43	0.16	51.48	1000.00	6.2M	0.77	624.02	218.02	7.07	1.07	6.62	0.35	0.72
S.D.	56.34	42.41	23.58	58.21	59.45	134.34	0.19	61.50	2185.40	4.0M	0.20	295.92	64.08	16.36	0.37	3.08	0.48	0.46
Minimum	1.00	20.00	1.00	216.00	52.00	287.00	0.00	0.00	-1214.35	0.7M	0.38	0.00	80.00	-18.00	0.36	2.80	0.00	0.00
Maximum	250.00	227.00	130.00	457.00	294.00	842.00	0.73	242.00	7215.65	20.2M	1.18	1147.00	401.00	91.00	1.91	12.50	1.00	1.00
Count	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46	46

Detailed Model Results

	M1 (non-sign controls included)				M2 (factory ID control excluded)				M3 (factory ID1 included)				M4 (factory ID2 included)				M5 (factory ID3 included)			
	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF
Measurement model																				
Development lead time (*DevLT*)																				
M0toM1	-0.11	0.33	0.11	1.05	0.62 ***	5.27	0.76	1.05	0.62 ***	5.38	0.76	1.05								
M1toM2	1.15 ***	3.27	0.72	1.42	0.75 ***	3.43	0.59	1.42	0.75 ***	3.59	0.59	1.42								
M2toM3	0.81 ***	3.07	0.22	1.37	0.53 ***	2.68	0.16	1.37	0.53 ***	2.79	0.16	1.37								
Hardware complexity (*HWC*)																				
common component count	0.18	0.76	0.50	1.65	0.07	0.58	0.45	1.65	0.07	0.66	0.45	1.65								
product specific component count	0.12	0.70	0.43	1.53	0.28 **	2.20	0.63	1.53	0.28 **	2.25	0.63	1.53								
parts coupling	0.10	0.34	0.49	2.14	0.38 **	2.33	0.68	2.14	0.38 **	2.45	0.68	2.14								
product novelty	0.85 ***	3.33	0.95	1.09	0.71 ***	4.58	0.84	1.09	0.71 ***	4.69	0.84	1.09								
Software complexity (*SWC*)																				
SW novelty	0.76 ***	3.04	0.92	1.16	0.54 ***	3.81	0.77	1.16	0.54 ***	3.76	0.77	1.16								
SW code size	0.12	0.44	0.26	1.02	0.40 ***	3.21	0.52	1.02	0.40 ***	3.41	0.52	1.02								
SW error count	0.40 ***	2.97	0.68	1.15	0.51 ***	5.74	0.74	1.15	0.51 ***	5.61	0.74	1.15								
Ramp-up performance (*RP*)																				
effective utilization	1.00	--	1.00	1.00	1.00	--	1.00	1.00	1.00	--	1.00	1.00								
Structural model (path coefficients)																				
Direct effects																				
SWC --> HWC	0.90 ***	5.07		1.00	0.89 ***	29.20		1.00	0.89 ***	28.20		1.00	0.89 ***	28.20		1.00	0.89 ***	28.20		1.00
HWC --> DevLT	-0.09	0.25		5.17	-0.05	0.36		5.08	-0.05	0.36		5.08	-0.05	0.36		5.08	-0.05	0.36		5.08
SWC --> DevLT	0.70 *	1.74		5.43	0.36 **	1.97		5.91	0.36 *	1.93		5.91	0.36 *	1.93		5.91	0.36 *	1.93		5.91
HWC --> RP	0.03	0.07		6.96	-0.28	1.09		5.10	-0.39	1.52		5.41	-0.30	1.52		5.16	-0.28	1.52		5.10
SWC --> RP	-0.79 **	2.10		7.81	-0.64 **	2.04		6.89	-0.49	1.57		7.44	-0.61	1.57		7.09	-0.65	1.57		6.94
DevLT --> RP	0.55 *	1.79		2.66	0.41 **	2.04		1.95	0.39 **	2.13		1.96	0.40 **	2.13		1.98	0.42 **	2.13		1.96
Control variables																				
linear trend --> DevLT	-0.33 *	1.74		1.12	--	--		--	--	--		--	--	--		--	--	--		--
linear trend --> RP	0.05	0.25		1.71	--	--		--	--	--		--	--	--		--	--	--		--
sales forecast change --> RP	0.12	0.75		1.35	--	--		--	--	--		--	--	--		--	--	--		--
years in operation --> RP	0.11	0.74		1.33	--	--		--	--	--		--	--	--		--	--	--		--
production technology novelty --> RP	-0.17	1.26		1.21	--	--		--	--	--		--	--	--		--	--	--		--
planned development lead time --> DevLT	--	--		--	0.74 ***	8.03		1.95	0.74 ***	7.81		1.96	0.74 ***	7.81		1.96	0.74 ***	7.81		1.96
slip --> RP	--	--		--	0.25 **	1.98		1.23	0.29 **	2.32		1.26	0.24 **	2.32		1.28	0.25 **	2.32		1.25
factory ID1...7 --> RP	--	--		--	--	--		--	-0.22 **	2.00		1.15	-0.09 **	2.00		1.07	0.02 **	2.00		1.08
factory location --> RP	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
Total effects																				
HWC --> RP	-0.02	0.06		--	-0.30	1.13		--	-0.41	1.53		--	-0.32	1.53		--	-0.30	1.53		--
SWC --> DevLT	0.62 ***	3.76		--	0.31 ***	3.04		--	0.31 ***	2.86		--	0.31 ***	2.86		--	0.31 ***	2.86		--
SWC --> RP	-0.43 **	1.98		--	-0.76 ***	5.51		--	-0.72 ***	5.56		--	-0.75 ***	5.56		--	-0.77 ***	5.56		--
Coefficient of determination (R^2)																				
HWC	0.80				0.79				0.79				0.79				0.79			
DevLT	0.62				0.87				0.87				0.87				0.87			
RP	0.37				0.44				0.48				0.44				0.44			
RP (R^2 adj)	0.26				0.38				0.41				0.37				0.36			

Notes: N = 46, measurement models for M4 to M9 are left out for clarity as they are in line with M3.

*** p ≤ .01, ** p ≤ .05, * p ≤ .10, two-tailed tests based on 500 bootstrap resamples.

	M6 (factory ID4 included)				M7 (factory ID5 included)				M8 (factory ID6 included)				M9 (factory ID7 included)			
	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF
Measurement model																
Development lead time (*DevLT*)																
M0toM1																
M1toM2																
M2toM3																
Hardware complexity (*HWC*)																
common component count																
product specific component count																
parts coupling																
product novelty																
Software complexity (*SWC*)																
SW novelty																
SW code size																
SW error count																
Ramp-up performance (*RP*)																
effective utilization																
Structural model (path coefficients)																
Direct effects																
SWC --> HWC	0.89 ***	28.20		1.00	0.89 ***	28.20		1.00	0.89 ***	28.20		1.00	0.89 ***	28.20		1.00
HWC --> DevLT	-0.05	0.36		5.08	-0.05	0.36		5.08	-0.05	0.36		5.08	-0.05	0.36		5.08
SWC --> DevLT	0.36 *	1.93		5.91	0.36 *	1.93		5.91	0.36 *	1.93		5.91	0.36 *	1.93		5.91
HWC --> RP	-0.26	1.52		5.93	-0.25	1.52		5.14	-0.23	1.52		5.38	-0.29	1.52		5.16
SWC --> RP	-0.65	1.57		7.15	-0.59	1.57		7.02	-0.63	1.57		6.90	-0.63	1.57		6.94
DevLT --> RP	0.42 **	2.13		1.97	0.43 **	2.13		1.95	0.36 **	2.13		2.37	0.42 **	2.13		1.97
Control variables																
linear trend --> DevLT	--	--		--	--	--		--	--	--		--	--	--		--
linear trend --> RP	--	--		--	--	--		--	--	--		--	--	--		--
sales forecast change --> RP	--	--		--	--	--		--	--	--		--	--	--		--
years in operation --> RP	--	--		--	--	--		--	--	--		--	--	--		--
production technology novelty --> RP	--	--		--	--	--		--	--	--		--	--	--		--
planned development lead time --> DevLT	0.74 ***	7.81		1.96	0.74 ***	7.81		1.96	0.74 ***	7.81		1.96	0.74 ***	7.81		1.96
slip --> RP	0.25 **	2.32		1.25	0.26 **	2.32		1.23	0.24 **	2.32		1.27	0.26 **	2.32		1.23
factory ID1...7 --> RP	-0.02 **	2.00		1.30	0.19 **	2.00		1.37	0.10 **	2.00		1.38	0.05 **	2.00		1.04
factory location --> RP	--	--		--	--	--		--	--	--		--	--	--		--
Total effects																
HWC --> RP	-0.28	1.53		--	-0.27	1.53		--	-0.25	1.53		--	-0.31	1.53		--
SWC --> DevLT	0.31 ***	2.86		--	0.31 ***	2.86		--	0.31 ***	2.86		--	0.31 ***	2.86		--
SWC --> RP	-0.76 ***	5.56		--	-0.67 ***	5.56		--	-0.73 ***	5.56		--	-0.76 ***	5.56		--
Coefficient of determination (R^2)																
HWC		0.79				0.79				0.79				0.79		
DevLT		0.87				0.87				0.87				0.87		
RP		0.44				0.46				0.44				0.44		
RP (R^2 adj)		0.36				0.39				0.37				0.37		

Notes: N = 46, measurement models for M4 to M9 are left out for clarity as they are in line with M3.

*** p ≤ .01, ** p ≤ .05, * p ≤ .10, two-tailed tests based on 500 bootstrap resamples.

	M10 (final model)				M10a (TRU_EXE = 4 weeks)				M10b (TRU_EXE = 8 weeks)				M10d (TRU_EXE = 16 weeks)				M10e (TRU_EXE = 20 weeks)			
	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF	Path weight	*t*-value	Loading	VIF
Measurement model																				
Development lead time (*DevLT*)																				
M0toM1	0.62 ***	5.73	0.76	1.05	0.58 ***	4.76	0.73	1.05	0.61 ***	4.74	0.76	1.05	0.62 ***	5.47	0.77	1.05	0.58 ***	4.74	0.73	1.05
M1toM2	0.75 ***	3.60	0.59	1.42	0.79 ***	3.75	0.61	1.42	0.76 ***	3.53	0.60	1.42	0.74 ***	3.77	0.58	1.42	0.79 ***	3.53	0.60	1.42
M2toM3	0.53 ***	2.79	0.16	1.37	0.56 ***	2.79	0.17	1.37	0.53 **	2.57	0.16	1.37	0.54 ***	2.63	0.17	1.37	0.58 **	2.57	0.19	1.37
Hardware complexity (*HWC*)																				
common component count	0.07	0.56	0.45	1.65	0.04	0.26	0.39	1.65	0.09	0.58	0.41	1.65	0.10	0.81	0.44	1.65	0.08	0.41	0.41	1.65
product specific component count	0.28 **	2.18	0.63	1.53	0.21	1.38	0.51	1.53	0.20	1.31	0.55	1.53	0.36 ***	2.84	0.69	1.53	0.32	0.62	0.62	1.53
parts coupling	0.38 **	2.29	0.68	2.14	0.21	1.01	0.52	2.14	0.36 *	1.80	0.62	2.14	0.39 **	2.18	0.70	2.14	0.29 *	1.80	0.62	2.14
product novelty	0.71 ***	4.76	0.84	1.09	0.85 ***	7.26	0.94	1.09	0.78 ***	5.21	0.89	1.09	0.66 ***	3.57	0.80	1.09	0.75 ***	5.21	0.87	1.09
Software complexity (*SWC*)																				
SW novelty	0.54 ***	3.85	0.77	1.16	0.69 ***	4.73	0.88	1.16	0.59 ***	3.53	0.82	1.16	0.53 ***	3.18	0.76	1.16	0.62 ***	3.53	0.83	1.16
SW code size	0.40 ***	3.22	0.52	1.02	0.28 **	1.98	0.41	1.02	0.33 **	2.30	0.45	1.02	0.48 ***	3.33	0.59	1.02	0.41 **	2.30	0.53	1.02
SW error count	0.51 ***	5.18	0.74	1.15	0.41 ***	3.39	0.68	1.15	0.50 ***	4.09	0.74	1.15	0.45 ***	4.88	0.68	1.15	0.40 ***	4.09	0.66	1.15
Ramp-up performance (*RP*)																				
effective utilization	1.00	--	1.00	1.00	1.00	--	1.00	1.00	1.00	--	1.00	1.00	1.00	--	1.00	1.00	1.00	--	1.00	1.00
Structural model (path coefficients)																				
Direct effects																				
SWC --> HWC	0.89 ***	30.11		1.00	0.90 ***	27.34		1.00	0.89 ***	25.83		1.00	0.89 ***	27.78		1.00	0.90 ***	28.21		1.00
HWC --> DevLT	-0.05	0.36		5.08	-0.05	0.28		5.41	-0.04	0.27		5.07	-0.09	0.63		5.14	-0.10	0.63		5.41
SWC --> DevLT	0.36 *	1.93		5.91	0.37 *	1.81		6.16	0.35 *	1.86		5.84	0.40 **	2.10		5.93	0.43 **	2.05		6.18
HWC --> RP	-0.39 *	1.69		5.40	-0.57	1.50		6.04	-0.43	1.35		5.53	-0.42 *	1.77		5.49	-0.28	1.60		5.91
SWC --> RP	-0.44	1.47		7.87	0.24	0.49		8.97	-0.19	0.47		8.16	-0.30	1.19		7.84	-0.12	1.15		8.55
DevLT --> RP	0.36 *	1.90		2.02	0.12	0.47		2.10	0.17	0.79		2.03	0.43 **	2.00		2.01	0.35 **	2.06		2.09
Control variables																				
linear trend --> DevLT	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
linear trend --> RP	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
sales forecast change --> RP	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
years in operation --> RP	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
production technology novelty --> RP	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
planned development lead time --> DevLT	0.74 ***	7.59		1.40	0.72 ***	7.47		1.37	0.73 ***	7.80		1.39	0.73 ***	7.75		1.38	0.72 ***	7.59		1.37
slip --> RP	0.27 **	2.30		1.24	0.03	0.21		1.29	0.35 **	2.50		1.26	0.12	1.04		1.23	0.04	1.06		1.26
factory ID1...7 --> RP	--	--		--	--	--		--	--	--		--	--	--		--	--	--		--
factory location --> RP	0.22 **	2.53		1.20	0.08	0.52		1.33	0.16	1.38		1.21	0.22 **	2.28		1.15	0.09 **	2.49		1.20
Total effects																				
HWC --> RP	-0.41 *	1.74		--	-0.58	1.54		--	-0.44	1.39		--	-0.46 *	1.84		--	-0.32 *	1.69		--
SWC --> DevLT	0.31 ***	2.86		--	0.33 ***	3.00		--	0.31 ***	2.89		--	0.32 ***	2.96		--	0.34 ***	2.84		--
SWC --> RP	-0.68 ***	5.06		--	-0.24	1.09		--	-0.53 ***	2.75		--	-0.54 ***	4.00		--	-0.26 ***	3.95		--
Coefficient of determination (R^2)																				
HWC		0.79				0.81				0.80				0.80				0.81		
DevLT		0.87				0.87				0.87				0.88				0.88		
RP		0.48				0.12				0.30				0.38				0.12		
RP (R^2 adj)		0.41				0.01				0.21				0.30				0.01		

Notes: N = 46.

*** p ≤ .01, ** p ≤ .05, * p ≤ .10, two-tailed tests based on 500 bootstrap resamples.

References

Abernathy, W. J., Clark, K. B. 1985. Innovation: Mapping the winds of creative destruction. *Research Policy*, 14, 3-22.

Almgren, H. 2000. Pilot production and manufacturing start-up: The case of Volvo S80. *International Journal of Production Research*, 38 (17), 4577-4588.

Banker, R. D., Datar, S. M., Kemerer, C. F., Zweig, D. 1993. Software complexity and maintenance costs. *Communications of the ACM*, 36 (11), 81-94.

Banker, R. D., Davis, G. B., Slaughter, S. A. 1998. Software development practices, software complexity, and software maintenance performance: A field study. *Management Science*, 44 (4), 433-450.

Basili, V. R., Perricone, B. T. 1984. Software errors and complexity: An empirical investigation. *Communications of the ACM*, 27 (1), 42-52.

Berg, M. 2007. *Factors affecting production ramp-up performance*, Licentiate Thesis, Jönköping University of Technology, Jönköping.

Blackburn, J. D., Scudder, G. D. Van Wassenhove, L. N. 1996. Improving Speed and Productivity of Software Development: A Global Survey of Software Developers. *IEEE Transactions on Software Engineering*, 22 (12), 875-885.

Boh, E. F., Slaughter, S. A., Espinosa, J. A. 2007. Learning from experience in software development: A multilevel analysis. *Management Science*, 53 (8), 1315-1331.

Boothroyd, G., Dewhurst, P., Knight, W. 1987. *Product design for manufacture and assembly*, 2nd ed., Marcel Dekker Inc., New York.

Broy, M., Krüger, I. H., Pretschner, A., Salzmann, C. 2007. Engineering Automotive Software. *Proceedings of the IEEE*, 95 (2), 356-373.

Callahan, J., Moretton, B. 2001. Reducing software product development time. *International Journal of Project Management*, 19, 59-70.

Cenfetelli, R. T., Bassellier, G. 2009. Interpretation of Formative Measurement in Information Systems Research. *MIS Quarterly*, 33 (4), 689-707.

Chiesa, V., Manzini, R., Toletti, G. 2002. Standard-setting process: Evidence from two case studies. *R&D Management*, 32 (5), 431-450.

Chin, W. W. 1998. The partial least squares approach to structural equation modeling, in Marcoulides, G. A. (ed.), *Modern Methods for Business Research*, Lawrence Erlbaum Associates, New Jersey.

Clark, K. B. 1985. The interaction of design hierarchies and market concepts in technological evolution. *Research Policy*, 14, 235-251.

Clark, K. B., Fujimoto, T. 1989. Lead Time in Automobile Product Development Explaining the Japanese Advantage. *Journal of Engineering and Technology Management*, 6, 25-58.

Clark, K. B., Fujimoto, T. 1991. *Product development performance: Strategy, organization and management in the world auto industry*, Harvard Business School Press, Boston.

Coughlan, P. D. 1992. Engineering change and manufacturing engineering deployment in new product development, in Susmann, G. I. (ed.) *Integrating Design and Manufacturing for Competitive Advantage*, Oxford University Press, New York, 157-177.

Eisenhardt, K. M., Brown, S. L. 1998. Time pacing: Competing in markets that won't stand still. *Harvard Business Review*, (March-April), 59–69.

Fisher, M., Ramdas, K., Ulrich, K. 1999. Component sharing in the management of product variety: A study of automotive braking systems. *Management Science*, 45 (3), 297-315.

Fornell, C., Bookstein, F. L. 1982. Two structural equation models: LISREL and PLS applied to consumer exit-voice theory. *Journal of Marketing*, 19, 440-452.

Geisser, S. 1977. The predictive sample reuse method with application. *Journal of the American Statistical Association*, 70, 320-328.

Gerwin, D., Barrowman, N. J. 2002. An Evaluation of Research on Integrated Product Development. *Management Science*, 48 (7), 938-953.

Graaf, B., Lormans, M., Toetenel, H. 2003. Embedded Software Engineering: The State of the Practice. *IEEE Software*, 20 (6), 61-69.

Griffin, A. 1993. Metrics for Measuring Product Development Cycle Time. *Journal of Product Innovation Management*, 10, 112-125.

Griffin, A. 1997. The effect of project and process characteristics on product development cycle time. *Journal of Marketing Research*, 34, 24-35.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., Tatham, R. L. 2006. *Multivariate data analysis*, 6th ed., Pearson Education Inc., New Jersey.

Hair, J. F., Ringle, C. M., Sarstedt, M. 2011. PLS-SEM: Indeed a Silver Bullet. *Journal of Marketing Theory and Practice*, 19 (2), 139-151.

Halstead, M. H. 1977. *Elements of software science*, Elsevier Science Publishing, New York.

Harter, D. E., Slaughter, S. A. 2003. Quality improvement and infrastructure activity costs in software development: A longitudinal analysis. *Management Science*, 49 (6), 784-800.

Hatch, N. W., Mowery, D. C. 1998. Process innovation and learning by doing in semiconductor manufacturing. *Management Science*, 44 (11), 1461-1477.

Hayes R. H., Clark, K. B. 1986. Exploring the sources of productivity differences at the factory level, in Clark, K. B., Hayes, R. H., Lorenz, C. (eds.), *The Uneasy Alliance, Managing the Productivity-Technology Dilemma*, Harvard Business School, Boston.

Huang, S. J., Lai, R. 1998. On Measuring the Complexity of an Estelle Specification. *Journal of Systems Software*, 40, 165-181.

Jarvis, C. B., MacKenzie, S. B., Podsakoff, P. M. 2003. A critical review of construct indicators and measurement model misspecification in marketing and consumer research. *Journal of Consumer Research*, 30 (2), 199-218.

Kafura, D., Reddy, G. R. 1987. The use of software complexity metrics in software maintenance. *IEEE Transactions on Software Engineering*, 13 (3), 335-343.

Kalish, S., Lilien, G. L. 1986. A market entry timing model for new technologies. *Management Science*, 32 (2), 194–205.

Keil, S., Lasch, R., Deutschländer, A., Schneider, G. 2007. Einfluss- und Erfolgsfaktoren auf den Ramp Up in der Halbleiterindustrie, in Otto, A. and Obermaier, R. (eds.), *Logistikmanagement,* Deutscher Universitäts-Verlag, Wiesbaden (in German).

Konopka, J. M. 1995. Capacity Utilization Bottleneck Efficiency System – CUBES. *IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part A*, 18 (3), 484-491.

Krishnan, M. S., Kriebel, C. H., Kekre, S., Mukhopadhyay, T. 2000. An empirical analysis of productivity and quality in software products. *Management Science*, 46 (6), 745-759.

Krishnan, V., Eppinger, S. D., Whitney, D. E. 1997. A Model-Based Framework to Overlap Product Development Activities. *Management Science*, 43 (4), 437-451.

Krishnan, V., Ulrich, K. T. 2001. Product development decisions: A review of the literature. *Management Science*, 47 (1), 1-21.

Krishnan, V., Zhu, W. 2006. Designing a family of development-intensive products. *Management Science*, 52 (6), 813-825.

Kuhn, A., Wiendahl, H. P., Eversheim W., Schuh, G. 2002. *Fast Ramp Up—Schneller Produktionsanlauf von Serienprodukten*, Verlag Praxiswissen, Dortmund (in German).

Laird, L. M., Brennan, M. C. 2006. *Software Measurement and Estimation. A Practical Approach*, John Wiley and Sons, Hoboken.

Langowitz, N. S. 1988. An exploration of production problems in the initial commercial manufacture of products. *Research Policy*, 17, 43-54.

Lee, E. A. 2000. What's Ahead for Embedded Software. *Computer*, 33 (9), 18-26.

Lovallo, D., Kahnemann, D. 2003. Delusions of Success. How Optimism Undermines Executives' Decisions. *Harvard Business Review*, (July), 56-63

Mallick, D. N., Schroeder, R. G. 2005. An integrated framework for measuring product development performance in high technology industries. *Production and Operations Management*, 14 (2), 142-158.

Matsuo, H., McIvor, R., Martin, J. D., Ng, J. 1997. Profiting from Process Improvements in the New Semiconductor Manufacturing Environment. *Technology and Operations Review*, 1 (2), 1-16.

McCabe, T. J. 1976. A complexity measure. *IEEE Transactions on Software Engineering*, SE-2 (4), 308-320.

Myers, S., Marquis, D. G. 1969. *Successful Industrial Innovation*, National Science Foundation, Washington.

Munson, J. C., Khoshgoftaar, T. M. 1991. The use of software complexity metrics in software reliability modeling. *Proceedings of the International Symposium on Software Reliability Engineering*, Austin, TX.

Murmann, P. A. 1994. Expected development time reductions in the German mechanical engineering industry. *Journal of Product Innovation Management*, 11, 236-252.

Novak, S., Eppinger, S. D. 2001. Sourcing by design: Product complexity and the supply chain. *Management Science*, 47 (1), 189-204.

Petter, S., Straub, D., Rai, A. 2007. Specifying formative constructs in information systems research. *MIS Quarterly*, 31 (4), 623-656.

Pisano, G. P., Wheelwright, S. C. 1995. The New Logic of High-Tech R&D. *Harvard Business Review*, (September-October), 94-105.

Preacher, K. J., Hayes, A. F. 2008. Asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods*, 40 (3), 870-891.

Pufall, A., Fransoo, J. C., de Kok, A. G. 2012. The Impact of Product Complexity on Ramp-Up Performance. *BETA Working Paper 389*, Eindhoven University of Technology, Eindhoven.

Rauscher, T. G., Smith, P. G. 1995. From experience; time-driven development of software in manufactured goods. *Journal of Product Innovation Management*, 12, 186-199.

Ringle, C. M., Wende, S., Will. A. 2005. *SmartPLS 2.0 (M3) Beta*, University of Hamburg, available at http://www.smartpls.de.

Rodriguez-Toro, C., Jared, G., Swift, K. 2004. Product-development complexity metrics: A framework for proactive-DFA implementation. *International Design Conference–Design 2004*, Cranfield University, United Kingdom.

Sangiovanni-Vincentelli, A., Martin, G. 2001. Platform-Based Design and Software Design Methodology for Embedded Systems. *Design & Test of Computers, IEEE*, 18 (6), 23-33.

Schaub, K. B., Kelly, J. 2004. *Production testing of RF and system-on-a-chip devices for wireless communications*. Artech House inc., Norwood MA.

Schuh, G., Kampker, B, Franzkoch, B. 2005. Anlaufmanagement, Kosten senken – Anlaufzeit verkürzen – Qualität sichern. *Werkstattstechnik online*, Jahrgang 95, H. 5, pp. 405-409 (in German).

Sherman, J. D., Berkowitz, D., Souder, W. E. 2005. New Product Development Performance and the Integration of Cross-Functional Integration and Knowledge Management. *Journal of Product Innovation Management*, 22, 399-411.

Sobel, M. E. 1982. Asymptotic confidence intervals for indirect effects in structural equations models, in Leinhart, S. (ed.) *Sociological Methodology*, Jossey-Bass, San Francisco, 290-312.

Stone, M. 1974. Cross-validatory choice and assessment of statistical prediction. *Journal of the Royal Statistical Society, Series B*, 36 (3), 111-133.

Sunohara, T., Takano, A., Uehara, K., Ohkawa, T. 1981. Program complexity measure for software development management. *IEEE Transactions on Software Engineering*, SE-4, 100-106.

Swink, M. 1998. A tutorial on implementing concurrent engineering in new product development programs. *Journal of Operations Management*, 16, 103-116.

Swink, M. 1999. Threats to new product manufacturability and the effects of development team integration processes. *Journal of Operations Management*, 17, 691-709.

Swink, M. 2003. Completing projects on-time: how project acceleration affects new product development. *Journal of Engineering and Technology Management*, 20, 319-344.

Tatikonda, M. V., Montoya-Weiss, M. M. 2001. Integrating operations and marketing perspectives of product innovation: The influence of organizational process factors and capabilities on development performance. *Management Science*, 47 (1), 151-172.

Tenenhaus, M., Esposito, V., Chatelin, Y. M., Lauro, C. 2005. PLS path modeling. *Computational Statistics & Data Analysis*, 48, 159-205.

Terwiesch, C., Bohn, R. E. 1998. Learning and process improvement during production ramp-up. *Information Storage Industry Centre Report*, 98-01.

Terwiesch, C., Chea, K. S., Bohn, R. E. 1999. An exploratory study of international product transfer and production ramp-up in the data storage industry. *Information Storage Industry Centre Report*, 99-02.

Thomke, S., Bell, D. E. 2001. Sequential testing in product development. *Management Science*, 47 (2), 308-323.

Ulrich, D. T. 1995. The role of product architecture in the manufacturing firm. *Research Policy*, 24, 419-440.

Ulrich, K. T., Eppinger, S. D. 2008. *Product design and development*, 4th ed., McGraw-Hill, New York.

van der Merwe, E. 2004. *A conceptual framework for ramp-up manufacturing*, PhD Thesis, Cambridge University, Cambridge.

Vandevelde, A., Van Dierdonck, R. 2003. Managing the design-manufacturing interface. *International Journal of Operations & Production Management*, 23 (11), 1326-1348.

Voigt, K. I., Thiell, M. 2005. Fast ramp-up—Handlungs- und Forschungsfeld für Innovations- und Produktionsmanagement", in Wildemann, H. (ed.) *Synchronisation von Produktentwicklung und Produktionsprozess: Produktreife – Produktanläufe – Produktionsauslauf*, TCW Verlag, München (in German).

Weyuker, E. J. 1998. Evaluating Software Complexity Measures. *IEEE Transactions on Software Engineering*, 14 (9), 1357-1365.

Wheelwright, S. C., Clark, K. B. 1992. *Revolutionizing product development*, The Free Press, New York.

Wildemann, H. 2007. *Anlaufmanagement, Leitfaden zur Optimierung der Anlaufphase von Produkten, Anlagen und Dienstleistungen*, TCW-Verlag, München (in German).

Williams, T. M. 1999. The need for new paradigms for complex projects. *International Journal of Project Management*, 17 (5), 267-273.

Wold, H. 1982. Systems under indirect observation using PLS, in Fornell, C. (ed.) *A Second Generation of Multivariate Analysis*, Vol. 1, Praeger, New York, 325-347.

Wolf, W. H. 1994. Hardware-Software Co-Design of Embedded Systems. *Proceedings of the IEEE*, 82 (7), 967-989.

Wood, R. E. 1986. Task complexity: Definition of a construct. *Organizational Behavior and Human Decision Processes*, 37, 30-82.

Zuse, H. 1991. *Software complexity, measures and methods*, De Gruyter Verlag, Berlin.

Working Papers Beta 2009 - 2012

nr.	Year	Title	Author(s)
390	2012	How does development lead time affect performance over the ramp-up lifecycle? Evidence from the consumer electronics industry	Andres Pufall, Jan C. Fransoo, Ad de Jong
389	2012	The Impact of Product Complexity on Ramp-Up Performance	Andreas Pufall, Jan C. Fransoo, Ad de Jong, Ton de Kok
388	2012	Co-location synergies: specialized versus diverse logistics concentration areas	Frank P.v.d. Heuvel, Peter W.de Langen, Karel H. v. Donselaar, Jan C. Fransoo
387	2012	Proximity matters: Synergies through co-location of logistics establishments	Frank P.v.d. Heuvel, Peter W.de Langen, Karel H. v.Donselaar, Jan C. Fransoo
386	2012	Spatial concentration and location dynamics in logistics:the case of a Dutch province	Frank P. v.d.Heuvel, Peter W.de Langen, Karel H.v. Donselaar, Jan C. Fransoo
385	2012	FNet: An Index for Advanced Business Process Querying	Zhiqiang Yan, Remco Dijkman, Paul Grefen
384	2012	Defining Various Pathway Terms	W.R. Dalinghaus, P.M.E. Van Gorp
383	2012	The Service Dominant Strategy Canvas: Defining and Visualizing a Service Dominant Strategy through the Traditional Strategic Lens	Egon Lüftenegger, Paul Grefen, Caren Weisleder
382	2012	A Stochastic Variable Size Bin Packing Problem With Time Constraints	Stefano Fazi, Tom van Woensel, Jan C. Fransoo
381	2012	Coordination and Analysis of Barge Container Hinterland Networks	K. Sharypova, T. van Woensel, J.C. Fransoo
380	2012	Proximity matters: Synergies through co-location of logistics establishments	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
379	2012	A literature review in process harmonization: a conceptual framework	Heidi Romero, Remco Dijkman, Paul Grefen, Arjan van Weele
378	2012	A Generic Material Flow Control Model for Two Different Industries	S.W.A. Haneya, J.M.J. Schutten, P.C. Schuur, W.H.M. Zijm

377	2012	Dynamic demand fulfillment in spare parts networks with multiple customer classes	H.G.H. Tiemessen, M. Fleischmann, G.J. van Houtum, J.A.E.E. van Nunen, E. Pratsini
376	2012	Improving the performance of sorter systems by scheduling inbound containers	K. Fikse, S.W.A. Haneyah, J.M.J. Schutten
375	2012	Strategies for dynamic appointment making by container terminals	Albert Douma, Martijn Mes
374	2012	MyPHRMachines: Lifelong Personal Health Records in the Cloud	Pieter van Gorp, Marco Comuzzi
373	2012	Service differentiation in spare parts supply through dedicated stocks	E.M. Alvarez, M.C. van der Heijden, W.H.M. Zijm
372	2012	Spare parts inventory pooling: how to share the benefits	Frank Karsten, Rob Basten
371	2012	Condition based spare parts supply	X.Lin, R.J.I. Basten, A.A. Kranenburg, G.J. van Houtum
370	2012	Using Simulation to Assess the Opportunities of Dynamic Waste Collection	Martijn Mes
369	2012	Aggregate overhaul and supply chain planning for rotables	J. Arts, S.D. Flapper, K. Vernooij
368	2012	Operating Room Rescheduling	J.T. van Essen, J.L. Hurink, W. Hartholt, B.J. van den Akker
367	2011	Switching Transport Modes to Meet Voluntary Carbon Emission Targets	Kristel M.R. Hoen, Tarkan Tan, Jan C. Fransoo, Geert-Jan van Houtum
366	2011	On two-echelon inventory systems with Poisson demand and lost sales	Elisa Alvarez, Matthieu van der Heijden
365	2011	Minimizing the Waiting Time for Emergency Surgery	J.T. van Essen, E.W. Hans, J.L. Hurink, A. Oversberg
364	2011	Vehicle Routing Problem with Stochastic Travel Times Including Soft Time Windows and Service Costs	Duygu Tas, Nico Dellaert, Tom van Woensel, Ton de Kok
363	2011	A New Approximate Evaluation Method for Two-Echelon Inventory Systems with Emergency Shipments	Erhun Özkan, Geert-Jan van Houtum, Yasemin Serin

362	2011	Approximating Multi-Objective Time-Dependent Optimization Problems	Said Dabia, El-Ghazali Talbi, Tom Van Woensel, Ton de Kok
361	2011	Branch and Cut and Price for the Time Dependent Vehicle Routing Problem with Time Window	Said Dabia, Stefan Röpke, Tom Van Woensel, Ton de Kok
360	2011	Analysis of an Assemble-to-Order System with Different Review Periods	A.G. Karaarslan, G.P. Kiesmüller, A.G. de Kok
359	2011	Interval Availability Analysis of a Two-Echelon, Multi-Item System	Ahmad Al Hanbali, Matthieu van der Heijden
358	2011	Carbon-Optimal and Carbon-Neutral Supply Chains	Felipe Caro, Charles J. Corbett, Tarkan Tan, Rob Zuidwijk
357	2011	Generic Planning and Control of Automated Material Handling Systems: Practical Requirements Versus Existing Theory	Sameh Haneyah, Henk Zijm, Marco Schutten, Peter Schuur
356	2011	Last time buy decisions for products sold under warranty	M. van der Heijden, B. Iskandar
355	2011	Spatial concentration and location dynamics in logistics: the case of a Dutch provence	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
354	2011	Identification of Employment Concentration Areas	Frank P. van den Heuvel, Peter W. de Langen, Karel H. van Donselaar, Jan C. Fransoo
353	2011	BOMN 2.0 Execution Semantics Formalized as Graph Rewrite Rules: extended version	Pieter van Gorp, Remco Dijkman
352	2011	Resource pooling and cost allocation among independent service providers	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
351	2011	A Framework for Business Innovation Directions	E. Lüftenegger, S. Angelov, P. Grefen
350	2011	The Road to a Business Process Architecture: An Overview of Approaches and their Use	Remco Dijkman, Irene Vanderfeesten, Hajo A. Reijers
349	2011	Effect of carbon emission regulations on transport mode selection under stochastic demand	K.M.R. Hoen, T. Tan, J.C. Fransoo G.J. van Houtum
348	2011	An improved MIP-based combinatorial approach for a multi-skill workforce scheduling problem	Murat Firat, Cor Hurkens

347	2011	An approximate approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
346	2011	Joint optimization of level of repair analysis and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
345	2011	Inventory control with manufacturing lead time flexibility	Ton G. de Kok
344	2011	Analysis of resource pooling games via a new extenstion of the Erlang loss function	Frank Karsten, Marco Slikker, Geert-Jan van Houtum
343	2011	Vehicle refueling with limited resources	Murat Firat, C.A.J. Hurkens, Gerhard J. Woeginger
342	2011	Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information	Bilge Atasoy, Refik Güllü, TarkanTan
341	2011	Redundancy Optimization for Critical Components in High-Availability Capital Goods	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
339	2010	Analysis of a two-echelon inventory system with two supply modes	Joachim Arts, Gudrun Kiesmüller
338	2010	Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh	Murat Firat, Gerhard J. Woeginger
335	2010	Attaining stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens
334	2010	Flexible Heuristics Miner (FHM)	A.J.M.M. Weijters, J.T.S. Ribeiro
333	2010	An exact approach for relating recovering surgical patient workload to the master surgical schedule	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	Efficiency evaluation for pooling resources in health care	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	The Effect of Workload Constraints in Mathematical Programming Models for Production Planning	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan

330	2010	Using pipeline information in a multi-echelon spare parts inventory system	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	Reducing costs of repairable spare parts supply systems via dynamic scheduling	H.G.H. Tiemessen, G.J. van Houtum
328	2010	Identification of Employment Concentration and Specialization Areas: Theory and Application	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	A combinatorial approach to multi-skill workforce scheduling	Murat Firat, Cor Hurkens
326	2010	Stability in multi-skill workforce scheduling	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	Maintenance spare parts planning and control: A framework for control and agenda for future research	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	Near-optimal heuristics to set base stock levels in a two-echelon distribution network	R.J.I. Basten, G.J. van Houtum
323	2010	Inventory reduction in spare part networks by selective throughput time reduction	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	The selective use of emergency shipments for service-contract differentiation	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	Preventing or escaping the suppression mechanism: intervention conditions	Nico Dellaert, Jully Jeunet.
319	2010	Hospital admission planning to optimize major resources utilization under uncertainty	R. Seguel, R. Eshuis, P. Grefen.
318	2010	Minimal Protocol Adaptors for Interacting Services	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
317	2010	Teaching Retail Operations in Business and Engineering Schools	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.

316	2010	Design for Availability: Creating Value for Manufacturers and Customers	Pieter van Gorp, Rik Eshuis.
315	2010	Transforming Process Models: executable rewrite rules versus a formalized Java program	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
314	2010	Getting trapped in the suppression of exploration: A simulation model	S. Dabia, T. van Woensel, A.G. de Kok
313	2010	A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows	
	2010		
312	2010	Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures	Osman Alp, Tarkan Tan
311	2010	In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints	R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	The state of the art of innovation-driven business models in the financial services industry	E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case	R. Seguel, P. Grefen, R. Eshuis
308	2010	Effect of carbon emission regulations on transport mode selection in supply chains	K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	Interaction between intelligent agent strategies for real-time transportation planning	Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	Internal Slackening Scoring Methods	Marco Slikker, Peter Borm, René van den Brink
305	2010	Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules	A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	Practical extensions to the level of repair analysis	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance	Jan C. Fransoo, Chung-Yee Lee
302	2010	Capacity reservation and utilization for a manufacturer with uncertain capacity and demand	Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	Spare parts inventory pooling games	F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	Capacity flexibility allocation in an outsourced supply chain with reservation	Y. Boulaksil, M. Grunow, J.C. Fransoo
298	2010	An optimal approach for the joint problem of level of repair analysis and spare parts stocking	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	Responding to the Lehman Wave: Sales Forecasting and Supply Management during the	Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx

		Credit Crisis	
296	2009	An exact approach for relating recovering surgical patient workload to the master surgical schedule	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	An iterative method for the simultaneous optimization of repair decisions and spare parts stocks	R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	Fujaba hits the Wall(-e)	Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller
293	2009	Implementation of a Healthcare Process in Four Different Workflow Systems	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	Business Process Model Repositories - Framework and Survey	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	Efficient Optimization of the Dual-Index Policy Using Markov Chains	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	Hierarchical Knowledge-Gradient for Sequential Sampling	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	Anticipation of lead time performance in Supply Chain Operations Planning	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	Inventory Models with Lateral Transshipments: A Review	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	Efficiency evaluation for pooling resources in health care	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	A Survey of Health Care Models that Encompass Multiple Departments	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	Supporting Process Control in Business Collaborations	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	Inventory Control with Partial Batch Ordering	O. Alp; W.T. Huh; T. Tan
282	2009	Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way	R. Eshuis
281	2009	The link between product data model and process model	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	Inventory planning for spare parts networks with delivery time requirements	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	Co-Evolution of Demand and Supply under Competition	B. Vermeulen; A.G. de Kok
			B. Vermeulen, A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	
			R. Seguel, R. Eshuis, P. Grefen

277	2009	An Efficient Method to Construct Minimal Protocol Adaptors	
276	2009	Coordinating Supply Chains: a Bilevel Programming Approach	Ton G. de Kok, Gabriella Muratore
275	2009	Inventory redistribution for fashion products under demand parameter update	G.P. Kiesmuller, S. Minner
274	2009	Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states	A. Busic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	Separate tools or tool kits: an exploratory study of engineers' preferences	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum
272	2009	An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	Distributed Decision Making in Combined Vehicle Routing and Break Scheduling	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	Similarity of Business Process Models: Metics and Evaluation	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	Vehicle routing under time-dependent travel times: the impact of congestion avoidance	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	Restricted dynamic programming: a flexible framework for solving realistic VRPs	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: http://beta.ieis.tue.nl