This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Method for Improving Overlapping of Testing and Design

Khadija Tahera, Chris Earl, and Claudia Eckert

***Abstract*—Testing is a critical activity in product development. The academic literature provides limited insight about overlapping between upstream testing and downstream design tasks, especially in considering the qualitative differences between activities that are overlapped. In general, the existing literature treats two overlapped sequential activities as similar, and suggests optimal overlapping policies, techniques, and time–cost assessment. However, this case study-based research identifies that the overlapping of upstream testing with downstream design activities has different characteristics than the overlapping of two design activities. This paper first analyzes the characteristics that affect the overlapping of upstream testing and downstream design activities, and then proposes a method to reduce the time of rework in cases where the upstream testing is overlapped with subsequent redesign phases.**

***Index Terms*—Overlapping, process improvement, product development process (PDP), simulation, testing, virtual testing.**

I. Introduction

PRODUCT development processes (PDPs) are iterative [1], [2], with design and testing cycles repeated several times [3]. An initial design may fail to meet customer requirements, have technical design faults, or raise issues about manufacturability and maintainability. These are revealed by testing upstream designs before commencing downstream redesign activities. As testing can take a long-time, downstream redesign often starts before testing is complete. This overlapping of testing and design activities can incur risk, since redesigning without complete test results might perpetuate faults or miss opportunities to respond to emerging problems. Effective management of this overlap between testing and design activities is a critical issue in engineering design processes within industrial companies.

A substantial literature exists on overlapping [4]–[6]. However, it overlooks the different types of information that are generated by various activities (requirements analysis, design, testing, or manufacturing) and that are exchanged during overlapping. Design and analysis specify design information such as material and geometry. Testing generates performance information, such as fatigue life. Customer needs and requirements analysis produce requirement information which may constrain design or performance information [7]. Similarly, downstream activities require specific types of design, performance, or requirement information, to proceed. While the research literature on overlapping largely addresses generic information exchange, this paper examines specific overlapping between design and testing activities which have different characteristics. Design refines information about a parameter [4], while testing observes, records, and evaluates results about a parameter [8]. Therefore, design is a refinement activity whilst testing is a revealing activity. In particular, testing can reveal unexpected flaws which are termed "deviations" and are discussed in Section IV. The extent of these deviations is a critical input to guide downstream design.

In overlapping, a downstream activity starts in parallel with an upstream activity by relying on the preliminary information that has not yet been finalized and may be communicated to the downstream activity in an informal, ad hoc, manner [9]. The primary risk, namely the risk of rework, associated with overlapping arises from the uncertainties in this preliminary information. Substantial research has been done on understanding, for generic overlapping, the format and timing [9] and frequency [5] of preliminary information exchanged. Other research has focused on effective communication and close coordination among different functional specialists [10]–[12], which allows more concurrency in executing tasks [13].

This paper emphasizes the practical necessity of focusing on specific types of overlapping activities in particular industry contexts and suggests ways to resolve industry issues. The research contribution is in two main areas. First, a model of overlapping incorporates the evolution of testing information to reduce the effect of uncertainties in preliminary testing results. Second the model uses the convergence of results from computer-aided engineering (CAE), considered as virtual testing taking place alongside physical testing, with physical test results to reduce the risks of overlapping of upstream test and downstream design. The model of overlapping is validated in a case study in the automotive sector.

The focus of the study is on relatively long lead-time product development from 6 to 18 months, typical in the automotive sector. In considering wider industry applications of overlapping design and test, the faster paced development processes in consumer products there will be extensive overlap of activities, especially in design and test. Interestingly, this also occurs in the

Manuscript received May 20, 2016; revised October 27, 2016; accepted December 15, 2016. Review of this manuscript was arranged by Department Editor Bin Jiang.

K. Tahera is with Mechanical and Automotive, University of Huddersfield, Huddersfield HD1 3DH, U.K. (e-mail: k.tahera@hud.ac.uk).

C. Earl is with the Faculty of Mathematics, Computing, and Technology, Open University, Milton Keynes MK7 6AA, U.K. (e-mail: C.F.Earl@open.ac.uk).

C. Eckert is with the School of Engineering and Innovation, Open University, Milton Keynes MK7 6AA, U.K. (e-mail: C.M.Eckert@open.ac.uk).

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

Digital Object Identifier 10.1109/TEM.2017.2654223

development of engineer to order products where design, test, manufacture, and assembly take place in parallel processes. So at these ends of a product development spectrum, overlapping of design and test is an integral part of the process, while in the midrange, such as automotive, overlap is forced from product delivery schedules. However, in all industry areas, the tools and methods for planning such overlap are limited. This paper describes a method, which although concentrating on this midrange, may also be applicable in the fast-paced and engineer to order product development. However, the method proposed depends on the quality and extent of test data as well as the scope to build corresponding simulations during product development. These features of the fast paced and engineer to order industries might lag behind the automotive sector.

II. Key Concepts and Related Literature

The overlapping of activities has received significant attention in product development literature. However, the specific information flows involved when testing and design activities are overlapped have not been considered. This section presents the key concepts along with related research. It provides the context for a case study from the automotive engineering sector about overlapping testing and design in engine development.

A. Testing Activities in the Product Development Process

To complete a project, a set of interconnected activities is coordinated in a PDP [14]. PDPs vary across companies but generally prescribe a structure of core activities and outputs at different product development stages. They are used to plan, schedule, and monitor product development. Testing is one of these core activities. In generic PDP models, such as the stage-gate [15], spiral [16], or V-models [17], testing activities are mostly allocated as a part of a validation stage toward the end of the process. Lévárdy *et al.* [18] have stressed that, since testing is often considered as a task toward the end of the PDP, the information flow between the design and testing domains can be insufficient for an effective PDP.

Design flaws, as well as technical and manufacturing issues, are identified through physical testing, which is often required for product certification. For example, the aerospace industries have a rigorous testing regime to pass certification criteria and automobile manufacturers are required to test their prototypes for regulatory and safety standards. But testing is time consuming and costly, typically accounting for up to 50% of total development cost [19]. In the spacecraft and satellite industry, system level integration and testing (I&T) alone costs approximately 35–50% of total development resources [20]. In the software industry, testing can consume 50% or more of the development costs [21]. In response to time-to-market pressures, engineers aim to get more value out of testing without adding time and cost. Planning and coordinating testing and design are, in consequence, a critical issue.

Some literature has addressed how to plan testing as part of product development [18], [19], [22], [23], but testing does not receive the same attention as design and production activities. Accelerating the PDP necessitates close coordination of testing with other activities such as prototype testing and concept verification [9]. Unger and Eppinger [24] and Yassine *et al*. [25] stress the importance of the information exchanges between the domains of design and testing [24], [25], but with the exception of Qian *et al*. [3], limited attention has been given to overlapping testing and design.

B. Activity Overlapping

Overlapping occurs when a downstream activity starts before an upstream activity is completed. In general, overlapping activities can reduce overall product development time [26]–[28]. When the downstream activity starts, it relies on preliminary information available from an overlapping upstream activity. As this information that has not yet been finalized, additional design and rework is often necessary to accommodate the upstream information as it becomes available [4], [11], [27], [29]. This rework can reduce the benefit of overlapping [4], [6]. In the worst case, development costs may increase and product quality may worsen [4]. Several studies have been completed on how to optimize the overlapping process in terms of time and cost trade-offs [2], [26], [29], [30], measuring the effectiveness of overlapping activities [31], a conceptual framework for managing overlapping [3], [4], [32], [33], and assessing risks and uncertainties in overlapping process [28], [34].

Among these studies, Qian *et al*. [3] investigated strategies for overlapping testing and design [3]. They claimed that the testing strategies in an overlapped process differ from those in a sequential process and proposed an analytical model for scheduling tests.

A key work by Krishnan *et al*. [4] provides a generic overlapping model of two interdependent activities which highlights that exchanged information between overlapping activities is critical for their management. This model is based on two concepts: "degree of evolution" and "downstream sensitivity." The "degree of evolution" describes the rate at which information is refined (and the interval/range of uncertainties about the design narrows). "Fast evolution" narrows the interval quickly, while "slow evolution" occurs if information evolves slowly at first and then rapidly toward the end of the process. The "downstream sensitivity" is the relationship between the magnitude of the change in the upstream information and the duration of downstream iteration. In "low downstream sensitivity," substantial changes in the upstream activity can be accommodated readily, in a short period of time, in the downstream activities. "High downstream sensitivity" occurs when small upstream changes require large amounts of rework in the downstream activity. Krishnan *et al.* [4] conclude that, in general, a fast evolution and low sensitivity situation is favorable to overlap as there is less risk of rework than in high sensitivity and slow evolution situations.

In the case study company, there were many situations where most changes occur toward the end of a long-duration testing activity (i.e., slow evolution) and where substantial redesign results from these changes (i.e., high sensitivity). According to Krishnan *et al*. [4] overlapping these activities in this situation may not be time saving. But many of the overlapping situations arise from overrun and are not planned. For instance, a late arrival of testing prototypes or materials can delay the start, and consequently, delay the finish of a testing activity. The company has no choice but to overlap design tasks with testing, as a design proposal is needed to commence another, often lengthy, procurement activity for the next stage of product development. In particular, physical testing often involves costly and time-consuming procurement, manufacture, and set-up of complex production quality prototypes. Managing the overlap effectively, avoiding unnecessary rework and iteration is more important than delivering time saving. This study concentrates on reducing rework when individual testing and design activities necessarily overlap. While these effects aggregate in complex and evolving ways across the whole process of product development, this paper will focus on the details of improving overlapping between just two activities. The effects on overall project duration will also be addressed but in general terms.

C. Iteration, Rework, and Review

Iteration, summarized as the rework of an activity [2], is an essential characteristic of new PDPs [24]. These iterations can be planned to manage risk through control of redesigns as in a stage-gate process [24] but they may also be unplanned due to unexpected failure in meeting requirements, technical design faults, or changes in requirements [2].

The downstream design iteration or rework can be instigated in two ways. The first is in response to design flaws identified in tests. The second, which is the focus here, arises because of overlapping with upstream testing and is often managed within the same stage in a stage gate process. Design flaws may be fed forward or propagated [35], emerging as late stage problems.

Companies may use gateway reviews, in a "stage gate process," between development stages to monitor progress [25], to prevent cross stage reworks and to reduce the propagation of design flaws. Strict reviews prevent further design until earlier work is finalized, while flexible reviews allow more overlapping between tasks [24]. In many cases, companies stand somewhere between these two extremes. In the early stages of product development, they may use flexible reviews. For instance, concept design can proceed with moderate review where there is still a chance to identify and fix design issues in later stages. However, in later stages, such as in product validation (PV), companies may use strict reviews to prevent design flaws propagating into the marketed products.

D. Information Exchange and Communication

Clark and Fujimoto [36] highlighted that exchanging and communicating preliminary design information rather than later release of complete information can reduce the rework time. They introduced "integrated problem solving" as a method to link the upstream and downstream groups to accelerate the design-build-test cycles. However, examining communication frequency and organizational structure does not address all the issues of using preliminary information effectively in overlapping activities [9]. Two alternative strategies were developed by Terwiesch and Loch [9]: iterative and set-based coordination. These help manage overlapping activities by focusing on the information precision (the accuracy of exchanged information) as well as information stability (the likelihood of changing a piece of information later in the process). This paper extends this research with particular attention to improving information precision.

E. Computer-Aided Engineering (CAE)

The use of CAE can enhance information sharing and communication among different functional specialists [37], [38]. CAE can increase the speed of information exchanges, enable faster execution of individual tasks, incorporate design changes more quickly [13], [39], and allow more concurrency in executing activities [13]. CAE also plays a role in the transfer of problem and solution information from previous projects to the front end of new projects [40]. Companies practicing concurrent engineering are likely to use CAE to support communication within the team and between the team and other product development groups [40]. Thomke and Fujimoto [40] and Loch and Terwiesch [11] identified that using CAE improves information sharing and enhances communication. This allows more concurrency in executing activities [11], [13], [40]. Although these studies demonstrate the general relevance of CAE in implementing concurrent engineering they do not show any specific mechanism or method for applying CAE. In this paper, a mechanism is introduced for using CAE as an intermediary activity between overlapping testing and design activities in order to enhance the effectiveness of information flow.

To summarize, although an extensive literature has addressed the issues and corresponding solutions for managing overlapping activities in PDPs, much of this work consists of general activity models not focused on specific pairs of overlapping activities. Furthermore, to get a realistic view, there is a need for complementing these analytical activity models by investigations of PDPs in companies with pressing problems and constraints in dealing with overlapping activities.

III. Product Development Process in the Case Study

A case study of overlapping testing and design was conducted in a UK-based company that designs and manufactures diesel engines. Diesel engines are complex, incremental, highly regulated products with extensive testing to meet customer requirements, performance standards, and statutory regulations. These engines are used in many applications such as agriculture, construction, material handling, marine, general industrial, and electric power. Testing requirements are different for different applications. A total of 18 interviews were carried out, recorded and transcribed, between 2011 and 2014 with eight engineers including a senior engineer, a development engineer, a CAE engineer, a verification and validation manager, and a validation team leader.

A. Stages and Gateways

The case study company has a structured stage gate process for new product introduction that has seven stages. Each stage leads to a formal gate review, starting from "Launch" to finish at "gateway 7." Based on prescribed criteria, a product must

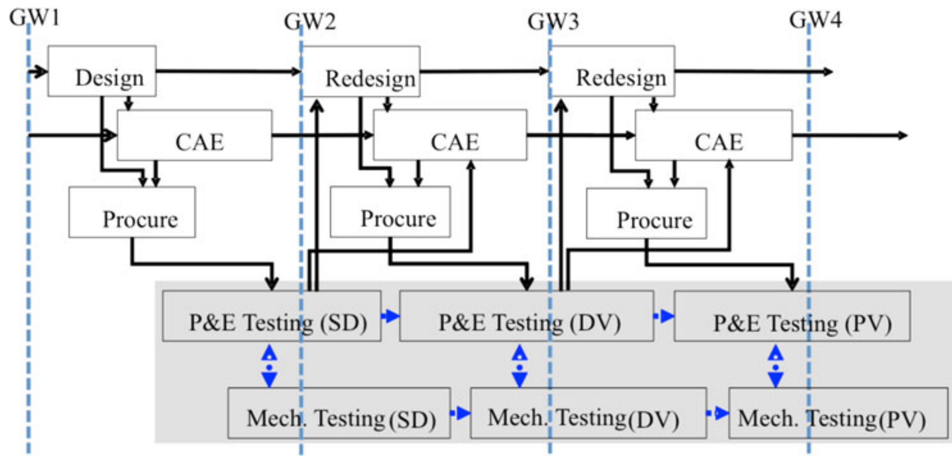

Fig. 1. Schematic of the PD process from gateways 1 to 4.

pass through final gate review before the project proceeds to the next stage. Among the large number of activities in these stages, the core activities: re/design, CAE (e.g., analysis and simulation), procurement (of test prototypes), and physical tests are considered for this study. Fig. 1 presents the structure of these activities from gateway 1 (GW1) to gateway 4 (GW4) that was established through detailed analysis of the PDP structure of the case study company.

Typically, design and development testing starts between GW1 and gateway 2 (GW2) (when R&D works have been completed and the technology has been selected) and continues till GW4, after which the engine is released to production. The three stages between GW1 and GW4 serve different purposes. In each, performance and emission, targets are addressed first and then the mechanical durability and reliability are tested. The three phases are described as follows.

1) Concept/system demonstration (SD) phase lies mostly between GW1 and GW2, and is primarily to demonstrate "performance capability" namely that the technology can deliver the required performance. Combinations of parts from a previous product and newly designed parts are built into an engine called a MULE, which is tested to verify the performance of the new parts. Alternative concepts are analyzed and evaluated in this stage. The product specifications evolve as design decisions are taken. It is assumed that by GW2, the concept will be selected, the components specified and the whole engine built with at least some production parts, ready to be tested for design verification (DV).
2) Design verification (DV) lies mostly between GW2 and gateway 3 (GW3), and is primarily to develop optimal performance and validate hardware at the optimized performance. The aim is to ensure that design outputs meet the given requirements under different use conditions. At this stage, testing focuses on the verification of a chosen design, through detailed analysis and testing of stress, strength, heat transfer and thermodynamics, etc. This stage validates the hardware prior to commitment to expensive production tooling.
3) Product validation (PV) takes place between GW3 and GW4, and checks the effect of production variability on performance and any remaining hardware variation. Hardware testing is limited to late design changes and emissions conformance testing. In this phase, detailed testing for reliability and durability is completed and the product is validated. The mandatory tests required for compliance usually occur during PV phases.

B. *CAE in the Product Development Process*

There are significant uses of the CAE analysis in the case study company. This is shown in Fig. 1 where CAE is picked out as a major activity with multiple uses at each gateway stage. CAE establishes a bridge between design and physical testing activities and is instrumental in developing strategies to minimize the time and costs involved in physical testing. CAE analyses enable the company to carry out optimization earlier in the product development cycle (front loaded), as well as improving product specification to the supplier. The company recognizes the significance of using CAE as a facilitator for product development as Engineer 1 commented,

> "computer simulation is becoming increasingly important to the companies to minimize the effort and expense involved in product development."

This analysis of company processes showed distinct phases in the application of CAE. These are identified as modeling, analysis, and virtual testing. At the early stages of a product development, the CAE analyses are used to investigate tradeoffs, usually in a mathematical representation of a system and its dynamic behavior. These models allow the engineers to simulate the interaction between components, for example, how an engine performs in a context, when given a load requirement for speed and acceleration. From these component-level computer aided design (CAD)/CAE analyses "design briefs" are created for individual components. These component level CAE analyses are performed after design work starts and often in parallel to design. A further level of CAE analysis and simulation is performed to identify the behavior and performance of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

the systems/components in response to specific environmental conditions. These types of CAE are usually advanced analyses tailored to specific issues. They are used to narrow down the boundary conditions and provide detailed information to the physical test engineers. These types of CAEs are referred as "virtual testing," because they serve the same purpose as the physical testing in that they examine whether a design meets specifications and requirements. Virtual testing is distinguished from earlier CAE because, just like physical testing, it is performed once the initial design is completed and design data and information are released to suppliers for procurement of test materials, e.g., physical prototype, testing components. Virtual testing complements and assists physical testing. For example, in a performance test, virtual testing can predict when to measure a value or in what conditions, and predicts the value that will be measured in a physical test. If the expected values do not correspond to test, measurements engineers can assume that either the analytical method applied for CAE analysis is not accurate or there are mismatches between the test settings and the CAE. The case study company's physical testing depends on CAE analysis before components, modules, or systems go to actual physical testing. Detailed discussion on virtual testing can be found in Tahera [41, pp. 94–99].

C. Physical Testing in the PD Process

Engines are tested in sequence for system demonstration (SD), then DV and PV, as illustrated in Fig. 1. In practice, several versions (at least three) of the same engine are tested simultaneously in parallel test-beds, where each bed replicates a particular set of specifications and operating conditions. Testing in one phase can identify design issues and lead to (re)design in the next phase. For instance, if testing in the SD phase identifies a failure or mismatches with the specification of a component, then in the next DV phase, engineers both redesign the component, including analysis of how changes affect other components or the whole engine performance, as well as conducting further detailed design specifically for the DV phase. The validation manager will require tests to be planned both for that particular component and for affected components. The testing activities may not be the same as in the previous stage but incorporate new testing parameters. Further retesting might occur in a different mode. For instance, CAE analysis or virtual testing might be sufficient to verify a design change resulting from a physical test with further physical testing not necessary. However, major changes in design will require new system level physical testing and this can delay product development significantly.

The duration of a test is often defined, i.e., if an engine test cycle is designed to run for 1000 h (i.e., the engine is in test-beds for 8 weeks), it must run for that specific time, unless a failure occurs earlier. Even if a failure occurs, engineers are likely to replace the failed component and continue the test to learn about the behavior of other components' and their durability in the complete test cycle. Therefore, if a physical test starts later than planned there is little chance that the duration of the test can be shortened. As the company shares testing facilities across several projects, the validation manager plans the tests and allocates the test-beds very early in the process, usually during stages 1 and 2. If a test-bed is occupied longer than planned then the next batch of tests is disturbed and test-bed schedules are mismatched. Delay in testing activities in one phase can delay the related activities in subsequent phases. As a result, delays aggregate and cause overall design process delay and late time-to-market.

The case of diesel engine development identifies that the long lead-time for procurement of test prototypes or components, and the long duration of physical tests when set alongside industry constraints on lead times and delivery dates causes significant overlaps among the activities. These substantial overlaps between testing in one phase and (re)design in the next take place in each stage.

D. Gateway Reviews and Decisions to Overlap

The company has a strong emphasis on maintaining each gateway using gateway-reviews for assessment and monitoring. The gateway review takes place in each stage of the PD process at a prescribed time and critical managerial decisions are taken after these reviews. At each stage, activities are scheduled in such a way that the gateway timeline can be maintained. However, often the gateway review takes place before testing is completed, as frequently testing activities take longer than initially planned. Engineers decide to overlap gateway stages, as another lengthy procurement process needs to start immediately to meet the schedules of the next phase. For example, the DV phase testing may still be on-going while the engineers are forced to start (re)design for the PV phase as well as procurement for the subsequent PV testing (see Fig. 1). Without final testing results, the company engineers encounter considerable uncertainties in redesigning and procuring for the next phase. These uncertainties cause more rework in design and errors in the procurement process, which can then lead to an iteration of a single phase. This situation causes the DV or PV phases to extend over two gateway stages. A brief examination of another case in the automotive sector where a company designs and manufactures fork-lift trucks revealed a similar situation where testing stretches across gateway stages.

IV. Overlapping Testing and Design Activities

Several issues arise when downstream design is overlapped with upstream testing tasks in addition to the factors of upstream evolution and downstream sensitivity introduced by Krishnan *et al.* [4]. This section maps out these additional factors and examines associated issues of information transfer. The term "evolution" (as introduced by Krishnan *et al.* [4]) refers to the refinement of upstream information as used in downstream processes. Such evolution that runs from a preliminary to a final value within an "initial interval," as seen in the left half of Fig. 2, is applicable for design activities. This concept of evolution may not adequately describe how information from testing activities is generated. This is because testing activities do not refine but reveal the value of a parameter. For example, design engineers in the case study company assumed that a design of an engine would produce power between 190 and 195 kW at 2200 r/min. A design analysis (e.g., CAE analysis) enables engineers to pre-

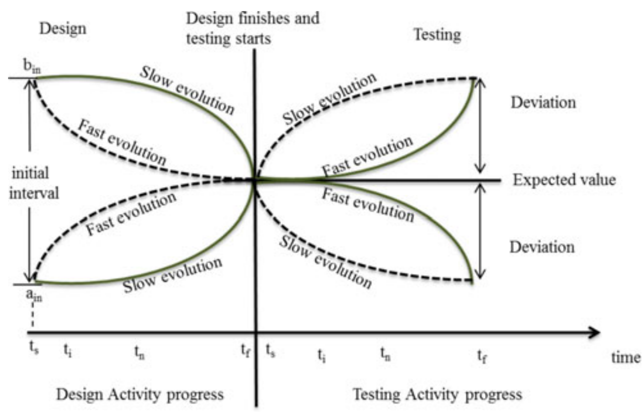

Fig. 2. Information evolution in design vs testing activity.

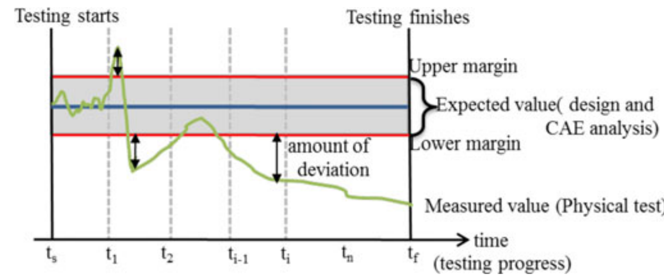

Fig. 3. Schematic of expected and measured values and associated deviations at different times during a test.

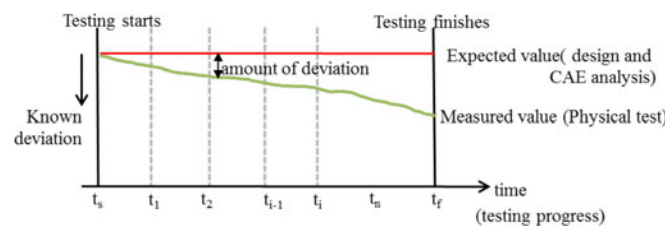

Fig. 4. Simplified model of deviations between expected and measured values during a test.

dict the expected value (i.e., according to specifications) of a parameter before commencing a test. Usually, engineers will allow some margin, for instance, a variation of ± 2 kW in engine power, in these expected values. The testing process then reveals the actual or measured value of the parameter. The design is successful if the measured value is within the expected values. Otherwise the design has flaws, which are indicated by the deviation between expected and measured values.

If a design is not accepted, engineers use the deviation between measured and expected values to guide improvement of the design during downstream iterations. In overlapping an upstream testing task with downstream design task this deviation plays an important role.

A. Deviation in Test Results

"Deviation" is the difference between the expected value of a parameter and an actual measurement of that parameter, at the time of an assessment (e.g., test). In an iterative design and testing process, testing results usually drive the subsequent re(design) activities. A control system analogy can be used to describe an iterative design and testing process. A control system monitors, compares, and adjusts at a sequence of time points. A monitoring device makes a measurement, and reports it to the comparator, which compares it with the predetermined desired value. A decision rule uses the result from the comparator to adjust an effector. Similarly, in a performance test, actual measurements of a parameter or the behavior of a product are taken and compared with expected values identified in design analysis to identify the deviation.

Also, during a lengthy durability test, for example, in a "deterioration factor" test, conducted over a lengthy period of time, intermediary test measurements are taken at a sequence of time points between start t_s and finish t_f (t_s, t_1, t_2 ... t_n ... t_f), as in Fig. 3. Engineers know that the performance of an engine will change over the time and they allow an acceptable margin for each time point. This is illustrated in Fig. 3 with a range of expected values specified by design and CAE prior to the test. Engineers will know how much they expect the product to deteriorate after say 200 or 500 h of running the test. If the product deteriorates below an allowable limit, or margin, at that time, then it is deemed under-designed. If an engine performs above the margin then it is assumed to be over-designed. Therefore, if the engine produces any value under or above the expected values (including margins) then these deviations are not acceptable (see Fig. 3) and indicate that redesign is required.

Fig. 4 shows a schematic, which presents a simplified case (of Fig. 3) in which the expected value is a single value rather than a range. In practice, this might be the mean of the distribution of expected values and is represented as the upper straight line (in red). The lower line (in green) represents the measured values. A physical test starts at t_s and finishes at t_f. Since the design meets specification based on the best knowledge available at t_s (or rather there is no information to indicate that it does not), the red and green lines meet at t_s. During the testing process, test measurements are taken and the actual value of a parameter at any point is identified.

Deviation, at a time point, is identified as the difference between test measurements and expected value. The magnitude of the deviation is shown with a double-headed arrow in Fig. 4. Fig. 4 depicts a case of under-design, with measured product performance gradually degrading and the deviation increasing monotonically. This considerable simplification is an assumption of the model developed here. The sloping line represents the evolution of test results over time, which tends to show increases in deviation of the design from expected performance. The deviation does not, in practice, decline linearly. The "amount of deviation" identifies how much change or improvement will be required in the downstream redesign tasks.

The difference between test measurements at different times can reveal the "degree of evolution" [4], i.e., how fast the deviation is changing in approach to the final value of the deviation at t_f. The "amount of deviation" plays a significant role along with "degree of evolution" and "sensitivity" (Krishnan *et al.*

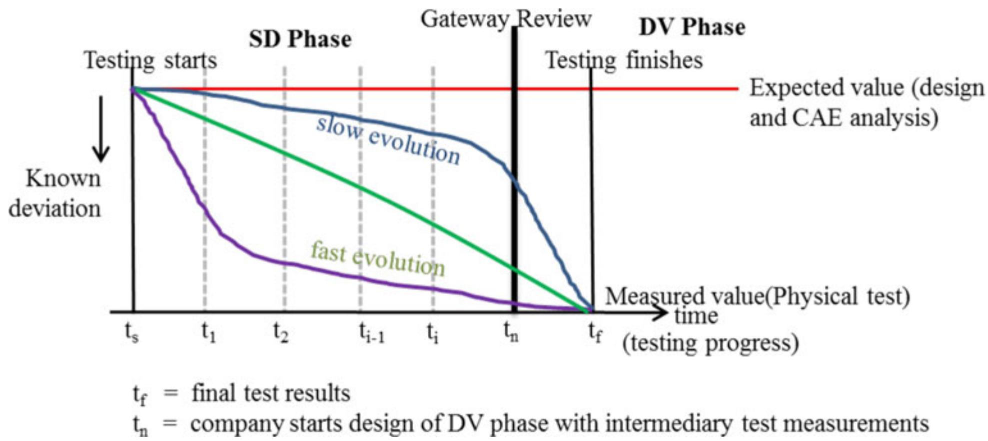

Fig. 5. Information evolution in a physical test with intermediary gateway review.

[4]) in managing the overlapping between upstream testing and downstream design tasks.

B. Enforced Overlap After Gateway Review

Fig. 5 presents the testing process in Fig. 4 overlaid on the gateway stages with an intermediary gateway review at t_n. In this case, a test starts at the SD phase but is completed at DV phase. The gateway review takes place at t_n, before the testing tasks are completed. Because of the gateway review, it is necessary for engineers to start the design and procurement process of the DV phase at this gateway time t_n to meet the schedules of the next phase. The dotted paths after gateway review, in Fig. 5, represent the improvement of the design with downstream redesign activities to correct the measured deviation.

Fig. 5 shows two extreme cases of information evolution in testing. For fast evolution, starting re/design at the gateway review may not be a significant problem (the lower curve in Fig. 5), because the information from testing at t_n is nearly complete. However, in slow evolution testing (the upper curve in Fig. 5), large changes to the test measurements occur after t_n, hence redesign starting at the gateway has significant uncertainty. To start the subsequent design activities at the gateway t_n, engineers need to minimize uncertainty of the predicted final value of a test at this point so that the downstream design will not suffer significant rework. These predictions, although relying to some extent on engineering judgment, can also take into account the profile of the intermediary test results, namely the degree of evolution. The analysis mentioned below formalizes the effects, and advantages, of overlapping.

C. Overlap and Rework

In this section basic notations for overlapping and rework are illustrated with an example of upstream testing and downstream design, with durations d_t and d_d, respectively.

The total duration of these tasks is $D_n = d_t + d_d$, when overlapping is not applied [see Fig. 6(a)]. When design and test overlap [see Fig. 6(b)], let d_e be the elapsed time between the starting time of upstream testing and the starting time of downstream redesign. Also, since downstream design starts with preliminary assumptions from the upstream testing, some of the downstream design might eventually require rework of duration d_x. Overlapping will provide time saving if $(d_e + d_x) < d_t$. In general, delaying the start of the downstream design, i.e., increasing the d_e, will allow more upstream testing results to be accumulated and $d_x = 0$ at $d_e = d_t$, when there is no overlapping, i.e., downstream design starts after finishing the upstream testing. In the company of this study, d_e depends on the time point for a gateway review. The key issue the company faces is how to effectively transfer the information about preliminary testing to the downstream design activities with reduced uncertainty, so that rework d_x in downstream design is significantly less than the overlap $d_t - d_e$.

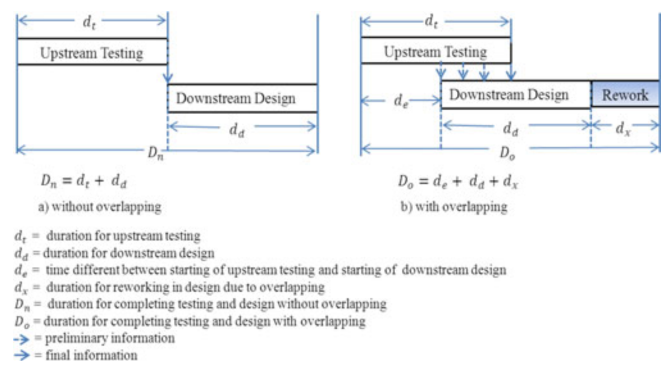

Fig. 6. Overlapping: durations and rework.

V. Method for Reducing Uncertainties in Overlapping

To reduce the likelihood of downstream rework in overlapping physical testing and design activities, there is a need for a mechanism that can accurately estimate the final value of a parameter faster than the physical testing itself and transfer that information to downstream design. This research identifies that "virtual testing" can act as such a mechanism. Virtual testing takes intermediary/preliminary test results and uses them to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

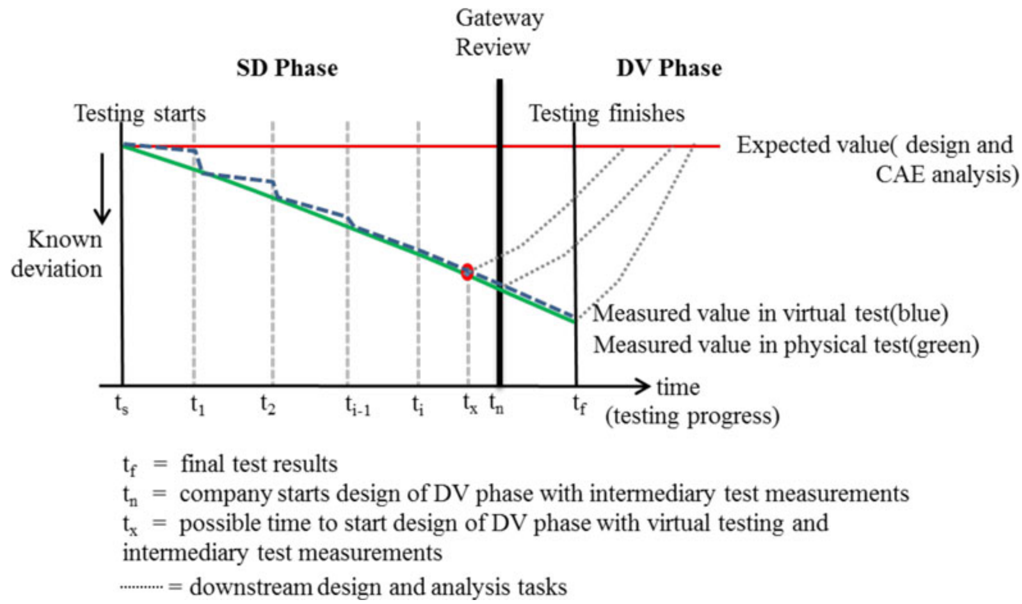

Fig. 7. Mirroring physical test in virtual testing by parallel executions of virtual and physical testing.

generate improved values quickly for downstream design tasks. There are two steps.

Step 1: calibrate a virtual model and validate it through physical test measurements.

Step 2: predict final test results through simulation using the virtual model.

A. Step 1: Validation of a Virtual Model

Initially, the measurements created by virtual testing can vary from the corresponding measurements through physical testing for several reasons. These include the virtual CAE model is not accurate, theories or assumptions in the virtual test are not correct, and the model on which the virtual test is based is not calibrated and validated due to lack of practical data.

A physical test can only be assisted with virtual testing if a virtual test is accurate and validated. More precisely the following conditions are necessary.

1) The supporting virtual CAE model is accurate.
2) The model is calibrated and validated accurately with practical test measurements.
3) Sufficient test measurements are gathered to have a confidence in test measurements.

The process of virtual model calibration and validation are discussed below through analyzing the way that a virtual test works alongside a corresponding physical test.

The simulation of the virtual model starts in parallel with physical testing at t_s (see Fig. 7). The company takes measurements from physical tests at several set points, for example, at $t_1, t_2 \ldots t_n \ldots t_f$. The simulated results of virtual testing should also be collected at the same time points. That is, if the test measurements are taken after 150 cycles, for example, which took twenty-four hours of running the physical test, the simulation results also have to be collected after equal number of cycles (i.e., 150 cycles), which might take considerably less time, say only two hours. At t_1, the physical test provides the first measurements of the parameter, on the current product under test. These measurements then will be available to compare with the simulated results, considering that both were running for same number of cycles. These initial test measurements will indicate the product's behaviors and consequently ensure that the type of analysis (for example linear or nonlinear analysis) is appropriate to meet requirements. The virtual model can be adjusted and improved according to the physical measurements in test.

Further simulation of the virtual model produces the values according to these measurements, which are compared again with the next test measurements at t_2. Any variations between physical and simulated results will require the model or its parameters to be adjusted. In a number of iterations, the virtual model will be adjusted and improved until the simulated results are representative of the physical test results. This will be expressed as a convergence between the test measurements and the virtual test predictions. If at time point t_i, simulation predicts the testing measurements accurately then at this point, the virtual model is effectively calibrated and validated with the current test measurements. Engineers also need to take a decision about whether the virtual model is validated and calibrated against sufficient test measurements. They continue simulation and testing until they have sufficient physical test data to calibrate and validate the virtual model before moving to Step 2.

B. Step 2: The Prediction of Final Test Results

Step 1 of calibrating and validating with actual testing measurements ensures that the virtual model accurately predicts a product's behavior revealed if the test were to run to completion at the final time planned point $t_{f.}$, beyond the gateway. To start a downstream design task before the end of testing, accurate predictions of the final values (i.e., the value at $t_{f.}$) of the measured parameter are required to minimize the significant rework in downstream tasks. For example, if at the point t_x, where the virtual model is validated there are still 1000 cycles of physical testing to run, the same number of cycles can be simulated in virtual testing faster than the physical tests. In this way, the uncertainty about the prediction of a final value of parameter, at an earlier point, can be reduced.

It is observed that an engineer might decide to start downstream design earlier than the gateway at t_n. With recent improvements in CAD tools, downstream design changes/rework and associated CAE analysis can significantly reduce rework extent and duration. The downstream design sensitivity can also be minimized through the effective communication between test engineers and design engineers. Other factors such as a product's modularity, robust design, and anticipation by downstream designers of changes in upstream information can all help reduce the sensitivity of downstream design [4].

A question naturally arises. In the case of slow evolution with significant deviations after the virtual model is validated at t_x, will this virtual model be able to simulate that results? For the case study company the answer is "yes." This company has a long history of developing engines and testing them. They understand the product and their testing procedure, because, most of the test procedures have been running for many years (as confirmed by Engineer 1). The engineers involved in the case study were consistent in their advocation of expanding the role of simulation and virtual testing in product development. The contribution here is in the way that this information is used to improve the process and its timely application rather than in a specific area of virtual testing. The virtual testing provides the point at which redesign can start effectively while overlapping test is conducted "in process." Convergence between virtual and physical test can aid decisions on overlap based on current information on the likelihood of final outcomes of test. However, systematic optimization of design/test overlap is more problematic as estimating the likelihood of rework in a design and test cycle depends on emerging test information. The method presented here assists engineers to identify a decision point when overlap becomes possible. As the following example will illustrate, they usually recognize the slow evolution tests and the point when the most of the changes occur in a test. With the help of virtual testing, the engineers will be able to decide if they need to wait until that point is reached.

VI. Implications for Product Development Duration: Examples

The examples focus on two illustrations of how the relation between d_e (starting time of downstream design after start of test) and d_x (rework time) might change, in the proposed method. In the first, an upstream test and a downstream design activity overlap across gateway stages. In the second, a set of overlapping test activities is considered.

A. *Overlap of an Upstream Test and Downstream Design Activity*

Consider the test in the case study company, which assesses engine performance under gross thermal cycles. Physical tests for gross thermal cycling provide an example of a lengthy endurance test, which checks the fatigue resistance of the cylinder head. This example was chosen because, frequently this test runs over the gateway stages and engineers need to start downstream redesigning while this test is still running. This is a critical test because it is performed on a core engine component, namely the cylinder head. Any changes of this component will impact significantly on the total engine system. Also, this test is very costly to run. This test is usually planned for the DV phase, at least three times for three variations of engines, and in recent company projects the norm is to repeat it at the PV phase to validate any remaining hardware variations.

This gross thermal test is a procedure for determining the thermal fatigue resistance of core engine components, by subjecting the engine to controlled, rapid coolant temperature change cycle. The cycle is normally applied to evaluate the cylinder head and cylinder head gasket. However, other engine components are also subjected to this gross thermal test. Each test cycle is 7 min (420 s) and at least 8500 cycles must be achieved. This equates to approximately 1000 h of test and means that the engine is in test bed for at least eight weeks. The objective of this test is that when an engine is run for extended periods (1000 h) in the test cycle given in this specific procedure, it will mirror the conditions that the engine will meet in service over the full lifecycle. The testing team records the data stream from several set points on the engine as the physical testing progresses. Cycle adherence is checked and sensor readings are taken every 24 h. Test measurements are recorded every day for this test. Finally, the whole engine is checked once the test is finished. The actual examination of the engine will range from simple visual inspections to accurate measurements of degradation of a given characteristic, i.e., wear of a component surface or rate of change of performance, leaks, and cracks.

This example was created by Engineer 1 and Engineer 3 who have many years of experience in the case study company. They know that there are significant amounts of overlapping between activities in their PDP and realize that overlapping the downstream design with upstream testing is critical to timely product delivery. But the company and its engineers lack a method of managing this overlapping. Redesign for the next phase of product development usually takes around 8 weeks for the cylinder head and associated gasket. It must take place immediately after a gateway review, even if the test is still running. Engineers will acquire as much information as they can from the upstream testing by observing the pattern of test measurements as well as through engineering judgments. They will have meetings with test engineers, product engineers, and senior validation managers to decide about emerging test data and when to release this information to the downstream design team.

For the purpose of comparison, the engineers were asked how the behavior of d_e and d_x would be observed in a regular case without virtual testing (upper curve "a" in Fig. 8). The horizontal axis represents the elapsed time, d_e in days since the start of test. The vertical axis represents the estimated time required for rework in downstream design. Engineers identified that the test does not produce any significant results during the first 28 days and most of the fatigue of the components starts to appear in the second half of the test (from day 28 to the end of the test at day 57). Thus, they do not recommend starting the downstream design before day 28 and set d_e at a minimum of 28 days. If the company starts redesigning after

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

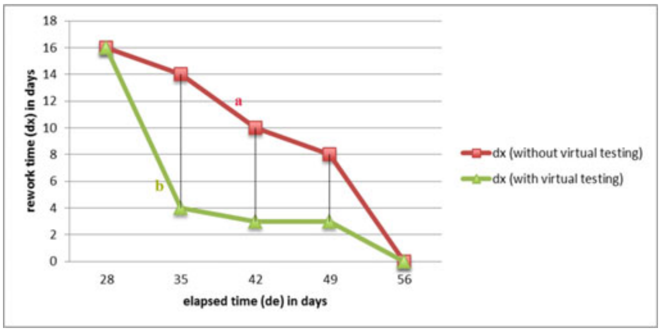

Fig. 8. Change in behavior of d_e and d_x with virtual testing.

28 days, they might need as long as a further 14 days to make design changes identified in the test. After 7000–7500 cycles of the test (i.e., the last week), they can decide more accurately about the product's behavior. However, they can only do final investigations after completing the test and many unexpected phenomena might appear which were not possible to predict earlier. This might cause a significant rework taking as long as 14 days, i.e., doubling the total duration for redesign, with D_0 becoming larger than D_n (see Fig. 6).

Curve "b" in Fig. 8 shows the potential for using virtual testing. After the first 28 days of the test, the engineers will be able to use test measurements, combined with historical data to virtually model the behavior of the component under test. The virtual model will be calibrated and validated using daily test results over the next 7 days. To run a simulation for the remaining 3000–3500 cycles in the test program will take about a day. Therefore, the subsequent design could be started any time after $28 + 7 + 1 = 36$ days. As curve b in Fig. 8 shows, the maximum benefit of using the parallel virtual testing is gained around day 36. After that a few more days in rework can be saved but with added costs of communication and running the simulation. At this point, the engineer might take critical decisions about time and cost. They might decide to wait for gateway review or possibly start the downstream design tasks even earlier.

Virtual testing of one phase also assists the CAE analysis of next phase. As design is assisted by the CAE analysis, any changes in design can be done in considerably shorter time. Therefore, the duration in downstream design rework d_x can be reduced substantially with the proposed addition of virtual testing. Learning from the parallel virtual testing may also reduce the uncertainties in procurement.

B. *Overlapping a Set of Tests*

The aim of the second example is to explore the effect of using the proposed method of parallel virtual and physical testing on the overall duration of a group of testing activities rather than a single pair of testing and redesign activities. The overlapping of a set of test activities were modeled as a flow diagram and analyzed through simulation using the Cambridge advanced modeler (CAM) [42] to evaluate the effect on total duration. The CAM modeler sets out a network of tasks representing precedence. Probabilities of successful completion of each task and associated durations are modeled, together with the iterative loops required if a task has not been completed successfully.

In an ideal case, these tests should be planned to be performed sequentially with two iteration loops shown in Fig. 9. However, the company cannot perform all these tests sequentially because mechanical tests such as fatigue resistance and wear tests take a significant amount of time. These lengthy tests might be overlapped with assumptions about previous tests. For example, the fatigue resistance test requires input from a vibration test and starting it earlier implies making assumptions about the results from the upstream vibration test. The same considerations apply to overlapping the fatigue and wear tests. Making such assumptions runs the risk of incurring extra rework at downstream design. The effects of employing parallel virtual testing in reducing rework can be quantified using process simulation modeling.

1) Modeling the Revised Testing Activities: Currently, engineers overlap testing tasks to reduce the overall completion time but the degree of overlap varied significantly between projects. Therefore, this study simulated two scenarios: first the current sequential ("as-is") testing process (in Fig. 9) and second changes incorporated into the future ("to-be") processes with a revised flow of overlapping activities supported by virtual testing (see Fig. 10). For simplicity in modeling both sequential and overlapping scenarios, the iterations in the performance test shown Fig. 9 are ignored. This helps to focus on the iterative effects that are due to overlapping. The number of physical tests and their durations are the same in both scenarios.

Fig. 10 shows the modeling of the "to-be" flow diagram where the vibration, fatigue, and wear tests are reconfigured as iterative activities represented by the diamond boxes in the lower part of the figure. This means that when these tests are finished, they "may" or "may not" feed the information to the successor tests. The simulation logic interprets these situations by not forcing their successor tests to wait for these tests to complete before starting. For instance, "fatigue resistance" will not wait for "vibration test" to complete before it starts. However, if "vibration test" feeds information into "fatigue resistance" later on during a simulation, then "fatigue resistance" test will be reworked along with all its successors that had already been executed. From Fig. 10, it can be seen that the flows from "vibration test" to "fatigue resistance" is labeled as "iterate again." This means that the information feed will only occur in a case of error, i.e., the assumptions made by fatigue resistance to start early have turned out to be inaccurate; therefore, rework is necessary. The likelihood of rework can be set in the iterative constructs for these two tests. Although likelihood of rework in design, as a proposal before testing, cannot be set in advance, test results provide the relevant information on the likelihood of rework. Furthermore, this example of vibration and fatigue testing presents an interaction between two tests which means that rework of one test, the fatigue test, may be necessary because of the results of another test, the vibration test. The likelihood of rework of the fatigue test emerges during the iterative process.

Within every activity, a representative minimum, expected, and maximum duration was estimated for each physical test in

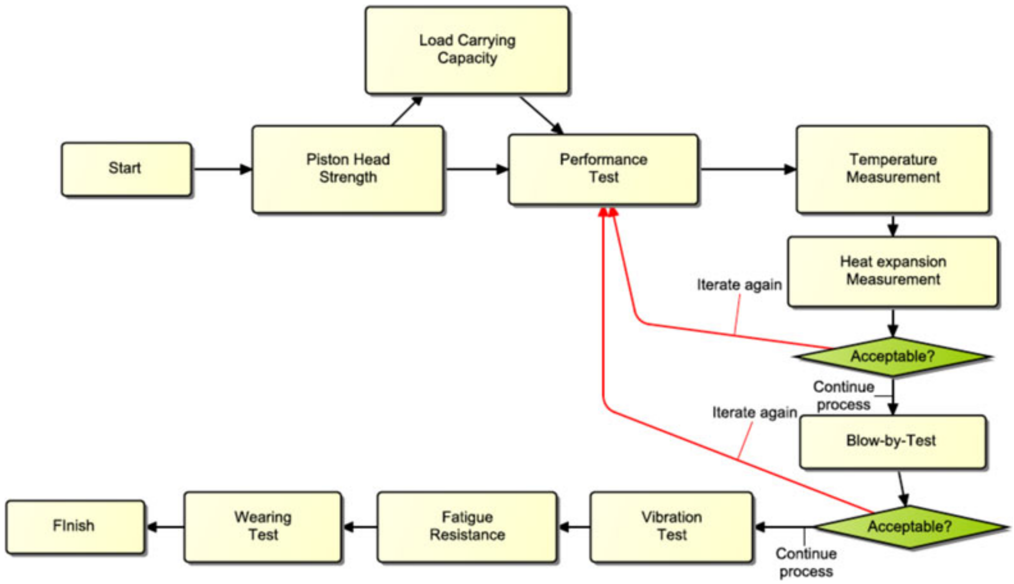

Fig. 9. Flow diagram for sequential testing activities.

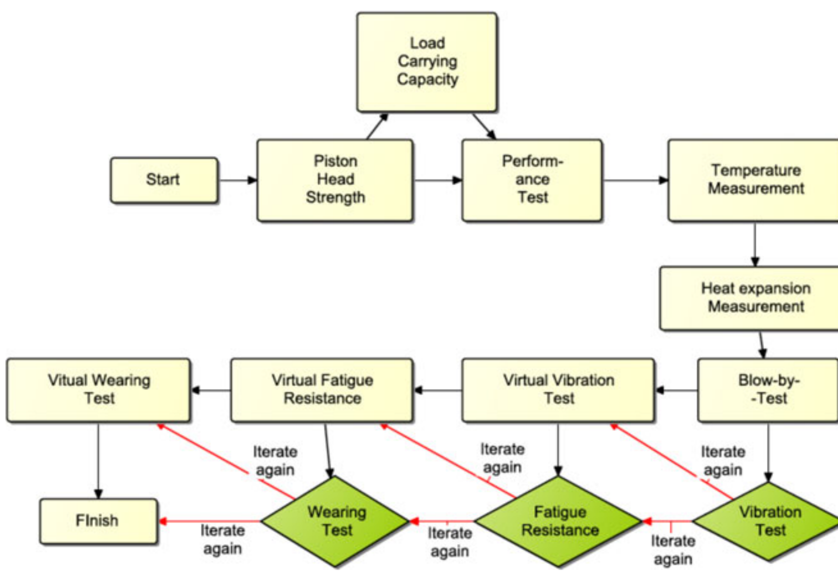

Fig. 10. Flow diagram of testing plan supported by virtual testing.

Table I (although actual values are not presented to preserve confidentiality) represented as a triangular probability density function (TriPDF). Similarly, a TriPDF model was used to assign durations to the corresponding virtual tests. For instance, the duration for virtual vibration test is set as TriPDF (1.5, 2, 2.5) for the first iteration. Here, it is assumed that in the best case, the virtual vibration test can be calibrated and validated with necessary and sufficient test measurements within halfway through of the vibration test (i.e., 1.5 days). It is most likely that it will take 2 days and in a worse case, it can take as long as the total duration of vibration tests (i.e., 2.5 days). A virtual vibration test will take significantly shorter time for the case of iteration. As a working assumption, the duration for consecutive iterations of the virtual vibration test has been set at 1 day. Furthermore, the virtual vibration test will not be performed once

TABLE I
TEST NAMES AND DURATION

Test Name	Minimum, Expected, Maximum Duration (in days)
Piston Head Strength	0.8, 1.0, 1.2
Load Carrying Capacity	1.8, 2.0, 2.5
Performance Test	0.8, 1.0, 1.2
Temperature Measurement	1.0, 1.5, 2.0
Heat Expansion Measurement	2.5, 3.0, 3.5
Blow-By-Test	0.9, 1.0, 1.1
Vibration Test	2.5, 3.0, 3.5
Virtual Vibration Test	1.5, 2.0, 2.5
Fatigue Resistance	5.5, 6.0, 6.7
Vibration Fatigue Resistance	3.0, 3.5, 5.5
Wearing Test	8.0, 10.0, 12.0
Virtual Wearing Test	5.0, 6.0, 8.0

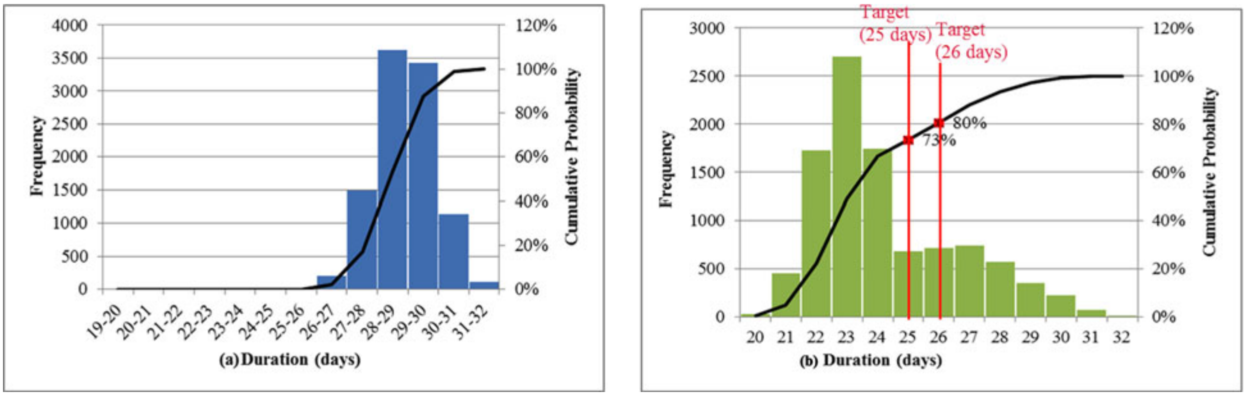

Fig. 11. (a) Histogram for sequential process duration and (b) overlapping process duration.

the physical vibration test is finished (finish-to-finish relationship). Other virtual tests follow a similar logic.

Currently, engineers decide the starting time of a downstream activity by looking at the progression of upstream tests. They also use experience and tacit knowledge. But in this proposed method, elapsed time (see Section IV-C) is determined by estimating the time that is required to calibrate and validate the respective virtual tests. This has been modeled by the inputs from a virtual test being mandatory to start a corresponding physical test. For instance, a fatigue resistance test needs inputs from the virtual vibration test to start. This means that the fatigue resistance test can only start when the virtual vibration test is calibrated and validated (see Fig. 10).

2) Simulation and Analysis of the Model: These two scenarios, sequential and revised flow, were then executed using 10 000 Monte Carlo simulation runs. The first used the ideal sequential ("as-is") testing process with fictitious duration values, shown as a histogram distribution in Fig. 11(a). The mean duration is 28.91 days with a standard deviation of 0.94 days for the given activities.

In the best case, this process will complete in 25.82 days and the worst case it may take up to 32 days. The chance of completing these tests on ~30 days is 80%. Second, the overlapping ("to-be") process is created by varying the probability of rework [in Fig. 11(b)]. If the engineers want to reduce the completion time to 25 days, for instance, and still want to achieve the 80% confidence that the project will finish on time, then they will have to reduce the rework time that is due to overlapping.

Typically, rework in one activity can propagate rework in other activities and higher order activities require careful consideration when the probability of rework is set. To keep this exercise simple, the propagation effects of rework on higher order activities have been ignored. Also the likelihood of rework for each iterative construct (i.e., the vibration, fatigue, and wearing tests) has been set equal and then systematically varied. Fig. 11(b) shows a histogram of durations in the Monte Carlo simulation with the cumulative distribution curve of simulated durations when likelihood of rework is set at 30%. In this case, the likelihood of finishing within a target of 25 days is 73% and can be increased to 80%, if 26 days are allowed. Similarly, from Table II, it can be seen that if the likelihood of rework can be

TABLE II
LIKELIHOOD OF REWORK

Likelihood of rework(%)	Mean (days)	Standard deviation, σ (days)	Likelihood of finishing within 25 days (%)	Likelihood of finishing within 26 days (%)
50	24.05	3.07	56	65
40	23.35	2.58	65	73
30	23.04	2.29	73	80
25	22.85	2.12	77	84
20	22.80	1.96	81	86

decreased to 20%, the likelihood of achieving the target of 25 days goes up to 81%. Table II shows a range of values for the likelihood of rework to execute 10 000 Monte Carlo simulation runs.

Not surprisingly, this analysis reveals that if the likelihood of rework can be reduced, there is a greater benefit of overlapping. In the proposed method, the likelihood of rework can be reduced by improving the capability of virtual testing. This means that if elapsed time can be increased, i.e., the time to start the downstream test can be delayed, and then additional time is available to calibrate and validate the virtual models with real test data, which can benefit in reducing the likelihood of rework. This might increase the total duration slightly, but can provide higher confidence of completing the given activities within the target time. Hence, engineers need to make a decision on how much confidence they want to achieve to finish a network of activities within target time, and on how much delay they can allow to build up before the start of the downstream activity in a case of overlapping. This kind of simulation analysis is useful when engineers are negotiating the time and cost targets, as well as choosing an acceptable risk when planning testing activities.

VII. DISCUSSION

As this is an analytical model and any timings for virtual model implementations are only estimates, the time estimations in Table II may be unrealistic. The time required to create a virtual model depends on the CAE department's skills and experience, and the availability of similar models. The number of iterations between virtual and physical testing will vary depending on the level of uncertainty and the accuracy and completeness of communication between testing engineers, design engineer, and CAE engineers.

Different tests benefit from integrating virtual testing with physical testing in different ways. Some benefit by focusing the tests, or identifying future values to minimize the number of iterations, while others require shorter running times. For example, in a constant speed and load situation, an engine has its quantities of fuel and air intake regulated, with the goal of achieving desired power ratings. An engine might require several iterations in design and test to achieve these desired power ratings. A virtual testing using a validated model can predict the likely consequences of certain values of fuel and air intake of the engine, thus suggesting appropriate values for next iteration in redesign and physical testing.

Not all physical tests will benefit from this approach. For example, in a case of fast evolution test where information evolves quickly and engineers can start downstream design tasks quite accurately with acceptable sensitivity, the test does not require support from parallel virtual testing. Also, there will be cases where virtual tests cannot assess the phenomenon which physical testing addresses. The design of seals is an example, where although virtual models are in principle buildable, they may be too complex, take excessive time, or be insufficiently accurate.

VIII. Conclusion and Future Work

Overlapping between upstream testing and downstream design occurs in each stage of product development due to long-lead time for procurement and lengthy physical tests. Late design changes affect the lead-time for procurement of prototypes. This unwanted and unavoidable overlapping increases the risk of extended rework time and iterations in the PDP.

This paper proposes a conceptual model of integrating virtual and physical testing to support overlapping between upstream testing and downstream redesign. Virtual testing is carried out in parallel to the physical testing in such a way that virtual testing can be calibrated through intermediary physical testing results. It can, therefore, simulate remaining physical test runs and provide more accurate information into subsequent redesign tasks and reduce rework.

The proposed method of parallel virtual and physical testing was validated with the senior engineer in the company. It was highlighted that this combined approach of physical and virtual testing methods had the potential to reduce iterations and thereby the number of physical prototypes saving time and cost. It would be useful to model and simulate the overall PDP including estimations of testing time. This would require considerable input from experienced engineers with adequate knowledge of planning the validation and testing activities.

Creating and using virtual models may increase the costs and resources consumed in each stage of the process. In any product development, balancing cost increases against possible time savings will be of critical importance. This model will need to be further developed by assessing the additional time, effort, and resources required. For instance, intermediary-testing measurements taken from physical tests may not be in a form that can be readily used for virtual testing and the time required for repeatedly comparing physical test measurements with virtual simulated results for convergence may require further examination. There is significant scope for future research on the role of virtual testing in product development, particularly in integrating design and test.

References

[1] A. Yassine and D. Braha, "Complex concurrent engineering and the design structure matrix method," *Concurr. Eng.*, vol. 11, pp. 165–176, 2003.

[2] C. Meier, T. R. Browning, A. Yassine, and U. Walter, "The cost of speed: Work policies for crashing and overlapping in product development projects," *IEEE Trans. Eng. Manage.*, vol. 62, no. 2, pp. 237–255, May 2015.

[3] Y. Qian, M. Xie, T. N. Goh, and J. Lin, "Optimal testing strategies in overlapped design process," *Eur. J. Oper. Res.*, vol. 206, pp. 131–143, 2010.

[4] V. Krishnan, S. D. Eppinger, and D. E. Whitney, "A model-based framework to overlap product development activities," *Manage. Sci.*, vol. 43, pp. 437–451, 1997.

[5] A. Yassine, B. Maddah, and N. Nehme, "Optimal information exchange policies in integrated product development," *IIE Trans.*, vol. 45, pp. 1249–1262, 2013.

[6] J. Lin, K. H. Chai, Y. S. Wong, and A. C. Brombacher, "A dynamic model for managing overlapped iterative product development," *Eur. J. Oper. Res.*, vol. 185, pp. 378–392, 2008.

[7] S. Suss, K. Grebici, and V. Thomson, "The effect of uncertainty on span time and effort within a complex design process," in *Modelling and Management of Engineering Processes*. New York, NY, USA: Springer, 2010, pp. 77–88.

[8] IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990, 1990.

[9] C. Terwiesch, C. H. Loch, and A. De Meyer, "Exchanging preliminary information in concurrent engineering: Alternative coordination strategies," *Org. Sci.*, vol. 13, pp. 402–419, 2002.

[10] A. Yassine, K. R. Chelst, and D. R. Falkenburg, "A decision analytic framework for evaluating concurrent engineering," *IEEE Trans. Eng. Manage.*, vol. 46, no. 2, pp. 144–157, May 1999.

[11] C. H. Loch and C. Terwiesch, "Communication and uncertainty in concurrent engineering," *Manage. Sci.*, vol. 44, pp. 1032–1048, 1998.

[12] K. B. Clark and T. Fujimoto, *Overlapping Problem Solving in Product Development*. Boston, MA, USA: Div. Res., Harvard Bus. School, 1987.

[13] A. Yassine, K.-C. Kim, T. Roemer, and M. Holweg, "Investigating the role of IT in customized product design," *Prod. Plan. Control*, vol. 15, pp. 422–434, 2004.

[14] T. R. Browning and R. V. Ramasesh, "A survey of activity network-based process models for managing product development projects," *Prod. Oper. Manage.*, vol. 16, pp. 217–240, 2007.

[15] R. G. Cooper, "Stage-gate systems: A new tool for managing new products," *Bus. Horiz.*, vol. 33, pp. 44–54, 1990.

[16] B. Boehm and W. J. Hansen, "Spiral development: Experience, principles, and refinements," Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Rep. no. CMU/SEI-2000-SR-008, 1998.

[17] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," in *INCOSE Int. Symposium*, vol. 1, no. 1, pp. 57–65, 1991.

[18] V. Lévárdy, M. Hoppe, and T. R. Browning, "Adaptive test process: an integrated modeling approach for test and design activities in the product development process," in *ASME 2004 Int. Design Eng. Tech. Conf. Comput. Inf. Eng. Conf.*, 2004, pp. 241–250.

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

[20] A. L. Weigel, "Understanding the enterprise value of test: characterizing system test discrepancies in the spacecraft industry," Master's thesis, Mass. Inst. Technol., Cambridge, MA, USA, 2001.

[21] A. Bertolino, "Software testing research: achievements, challenges, dreams," in *Proc. Future Softw. Eng. IEEE Comput. Society*, 2007, pp. 85–103.

[22] C. H. Loch, C. Terwiesch, and S. Thomke, "Parallel and sequential testing of design alternatives," *Manage. Sci.*, vol. 47, pp. 663–678, 2001.

[23] S. Thomke and D. Bell, "Optimal testing in product development," Harvard Bus. School, Boston, MA, USA, Working Paper 99-053, 1999.

[24] D. Unger and S. Eppinger, "Improving product development process design: A method for managing information flows, risks, and iterations," *J. Eng. Design*, vol. 22, pp. 689–699, 2011.
[25] A. A. Yassine, R. S. Sreenivas, and J. Zhu, "Managing the exchange of information in product development," *Eur. J. Oper. Res.*, vol. 184, pp. 311–326, 2008.
[26] T. A. Roemer and R. Ahmadi, "Concurrent crashing and overlapping in product development," *Oper. Res.*, vol. 52, pp. 606–622, 2004.
[27] J. E. V. Gerk and R. Y. Qassim, "Project acceleration via activity crashing, overlapping, and substitution," *IEEE Trans. Eng. Manage.*, vol. 55, no. 4, pp. 590–601, Nov. 2008.
[28] J. Wang and Y. I. Lin, "An overlapping process model to assess schedule risk for new product development," *Comput. Ind. Eng.*, vol. 57, pp. 460–474, 2009.
[29] T. A. Roemer, R. Ahmadi, and R. H. Wang, "Time-cost trade-offs in overlapped product development," *Oper. Res.*, vol. 48, pp. 858–865, 2000.
[30] A. K. Chakravarty, "Overlapping design and build cycles in product development," *Eur. J. Oper. Res.*, vol. 134, pp. 392–424, 2001.
[31] C. Terwiesch and C. H. Loch, "Measuring the effectiveness of overlapping development activities," *Manage. Sci.*, vol. 45, pp. 455–465, 1999.
[32] S. M. Bogus, K. R. Molenaar, and J. E. Diekmann, "Strategies for overlapping dependent design activities," *Constr. Manage. Econ.*, vol. 24, pp. 829–837, 2006.
[33] K. Tahera, C. Earl, and C. Eckert, "Integrating virtual and physical testing to accelerate the engineering product development process," *Int. J. Inf. Technol. Manage.*, vol. 13, nos. 2–3, pp. 154–175, 2014.
[34] Q. Yang, T. Lu, T. Yao, and B. Zhang, "The impact of uncertainty and ambiguity related to iteration and overlapping on schedule of product development projects," *Int. J. Project Manage.*, vol. 32, pp. 827–837, 2013.
[35] P. G. Smith and D. G. Reinertsen, "Shortening the product development cycle," *Res. Technol. Manage.*, vol. 35, pp. 44–49, 1992.
[36] K. B. Clark and T. Fujimoto, *Product Development Performance: Strategy, Organization, and Management in the World Auto Industry*. Cambridge, MA, USA: Harvard Bus. Press, 1991.
[37] S. Ku, "Introduction of 3D-CAD and its effects in automobile industry: 3DCAD, communication among firms, efficiency of development, cause and effect model," *Org. Sci.*, vol. 37, pp. 68–81, 2003.
[38] Y. Park, T. Fujimoto, and P. Hong, "Product architecture, organizational capabilities and IT integration for competitive advantage," *Int. J. Inf. Manage.*, vol. 32, pp. 479–488, 2012.
[39] C. L. Tan and M. A. Vonderembse, "Mediating effects of computer-aided design usage: From concurrent engineering to product development performance," *J. Oper. Manage.*, vol. 24, pp. 494–510, 2006.
[40] S. Thomke and T. Fujimoto, "The effect of "front-loading" problem-solving on product development performance," *J. Product Innov. Manage.*, vol. 17, pp. 128–142, 2000.
[41] K. Tahera, "The role of testing in engineering product development processes," Ph.D. dissertation, Eng. Innov., Open Univ., Milton Keynes, U.K., 2014.
[42] D. C. Wynn, D. F. Wyatt, S. M. T. Nair, and P. J. Clarkson, "An introduction to the Cambridge advanced modeller," presented at MMEP '10 Cambridge UK, Jul. 2010.

Khadija Tahera received the bachelor's degree in computer engineering from Stamford University, Bangladesh; the M.Sc. degree in astronautics and space engineering from Cranfield University, Cranfield, U.K.; and the Ph.D. degree in engineering design processes from the Department of Engineering and Innovation, Open University, U.K., in 2002, 2008 and 2014, respectively.

She is currently working as a Lecturer with the University of Huddersfield, Huddersfield, U.K. Before starting her Ph.D., she worked as a Researcher at Open University and Cranfield University.

Chris Earl received the Ph.D. degree in design from Open University, Milton Keynes, U.K., in 1981

He was in faculty positions in Bristol and Newcastle, U.K. At the Newcastle Engineering Design Centre, Newcastle, his industry-based research centered on planning design and manufacturing processes in the development of engineer to order one-of-a-kind products. He also developed his work in generative and functional descriptions of design, as well as developing design for cost methods. Since 2000, he has been a Professor of design with Open University. He served as the Head of the Department of Design and Innovation followed by the Dean of the Faculty in Mathematics, Technology, and Computing with Open University, and is currently a Professor Emeritus. His research interests include modeling design processes and their constituent activities such as test in a range of domains including engineering, industrial design, and architecture.

Claudia Eckert received the Ph.D. degree in design from Open University, Milton Keynes, U.K., in 1997, and the M.Sc. degree in applied artificial intelligence from the University of Aberdeen, Aberdeen, U.K., in 1990.

She is a Professor of design at Open University, Milton Keynes, U.K. Her research interests include understanding and supporting design processes, and in particular, engineering change and processes planning. She is also working on comparisons between design domains.