Journal of the Society of Korea Industrial and Systems Engineering
Vol, 29, No. 3, pp.25－32, September 2006.

시스템 다이나믹스를 이용한 자동차 개발 단계에서 Design-Build-Test Cycle의 중복에 대한 연구†

이상돈* · 임익성**

*미국 미시간주 제너럴 모터스 기술센터
**남서울대학교 산업경영공학과

Overlapping Design-Build-Test Cycles in Vehicle Development Process : System Dynamics Approach

Sang-Don Lee* · Ik-Sung Lim**

*General Motors Tech. Center, Warren, MI 48090-9000, USA
**Dept. of Industrial & Management Eng. Namseoul University

자동차 개발 단계는, 본질적으로, 수차례의 개발 단계를 거치면서 설계 그룹들 간의 정보 전달과 교환이 여러 차례 반복적으로 발생하게 된다. 이러한 복잡한 상호 정보 교환 과정을 정확하게 이해하여 업무가 수행되지 못하면 불필요한 재작업을 야기시킬 수 있으며, 이는 비용 낭비 및 계획에 차질이 발생할 수 있다. 이 연구에서는 시스템 다이나믹 모델을 개발하여 빈번한 design-built-test 싸이클이 자동차 개발 단계에 미치는 영향 및 이점들을 고찰하였다.

Keywords : System dynamics, Design iteration, Vehicle development process, Simulation, Design-build-test iteration cycles

1. 서 론

In the automotive industry, vehicle development lead time has a direct impact on the company's profits. For example, the delay in introducing a $10,000 small car to the market is estimated to cost at least one million dollars per day in lost profits [2]. In order to reduce lead time, the Vehicle Development Process(VDP) in the automotive industry has been shifted from sequential-functional to concurrent-team based approach.

In concurrent environment, engineers from various functional groups perform a variety of activities in parallel by iterating and exchanging design information. The concurrent execution of the design activities helps engineers to reduce time delays in exchanging the design information and to consider the vehicle design as a whole.

However the lead time reduction comes at the cost of increased management complexity. Because downstream tasks are often based on incomplete information, the number of design iteration and subsequent rework are increased. Also, since many engineering characteristics are considered at the same time, engineers have to juggle many factors simultaneously, which causes confusion. Dynamic interactions and interdependencies in a concurrent process without the proper management among the various design groups and between development phases usually involves excessive downstream rework, which often leads to significant increases in lead

† The authors are grateful to Dave Helton at vehicle development process center for his valuable inputs and comments.

time and cost [9, 10].

Design iteration is the repetition of design tasks due to fuzzy inputs such as customer needs, technical requirements and design problems. The iteration takes a significant product development cycle time, typically about one third of the project effort. The subsequent redesign is a major cause of long lead time in development [1]. Cooper showed that the delay in discovering rework slows the completion of development tasks approximately 1/4 to 3/4 of the original time [4-8]. Therefore, design iteration and the subsequent rework are supposed to be the primary indicators of lead time performance, as well as measures of design quality.

Recently researchers have studied several aspects of the design iteration. Krishnan et al. [12] analyzed the effect of iterative overlapping, in which downstream development activities begin with preliminary upstream design information and accommodate design changes in the subsequent iteration. Krishnan also discussed the managerial implications of managing risks involved in the simultaneous execution of coupled development phases [13]. Ha et al. developed a model for the optimal design review periods and analyzed the effects of changing the timing of design reviews on the total length of the development process [11]. Steward developed the Design Structure Matrix (DSM) for effective and efficient organization of tasks that interact and iterate to analyze the structure of design problems [20]. Kusiak et al. have combined this methodology with a group technology formulation to identify how tasks should be divided into groups [14]. Smith et al. identified two types of iteration: sequential and parallel [21, 22].

The VDP is an iterative process of design-build-test (DBT) cycles with numerous coupled tasks that are performed by various functional groups. Through these iterative cycles, concept, configuration and other technical details of a vehicle are generated, narrowed, and finalized. The iteration are inevitable because of the complexity in vehicle design. One of the reasons for the occurrence of design iteration in vehicle development is because of the complexity in integrating many components and subsystems developed by various design groups. The internal iteration referes to the design iteration within a design group and the external iteration refers to the design iteration between design groups. They are shorter cycles in longer DBT iteration cycles between development phases. Frequent internal and external design iteration may not improve the vehicle development significantly without proper improvements in the DBT iteration cycles where components and subsystems are integrated. For example, the delay in perfecting the design for each subsystem in a design group by iteration is not in the best interest of the overall program, although locally optimized, due to the difficulty of integrating major subsystems.

Traditional project management methods such as PERT/CPM and precedence diagramming are useful to schedule non-repetitive activities connected by precedence relationship. However, they unintentionally provide a false impression to engineers and managers : no design iteration and a lump sum release of design information [13].

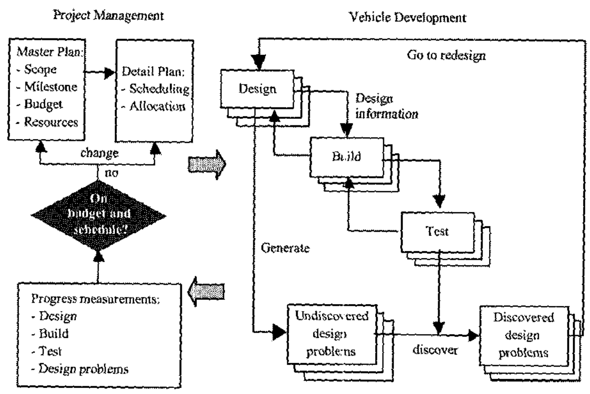
System Dynamics has been successfully applied to analyze the dynamic and iterative nature of new product development. Roberts explored the basic dynamics of an R&D project, and here the concepts of *perceived progress and real progress* were first introduced [16, 17]. Cooper presented the novel concept of the *rework cycle*, which incorporates the notion of *undiscovered rework*, *time to discover rework*, work *quality* and varying staff *productivity* [4]. He demonstrated the impact of rework on development lead time, and showed that rework is the root cause for the "90 percent syndrome". The 90 percent syndrome indicates the phenomenon that for a prolonged time in new product development, the project progress is stagnated near completion. Richardson et al. presented a model for the management of an R&D project that summarized the basic feedback structures of the project management [18]. Ford developed a model explicitly focusing on the relationships among coordination, schedule, and quality, incorporating the internal and external precedence relationship of tasks and development stages, respectively [10]. Rodrigus et al. provided a review on the system dynamics application for project management [19].

The objective of this paper is to analyze the DBT iteration cycles in the VDP, in which numerous interactions and information flows occur among design groups and between development phases. System dynamics was applied in order to analyze the iterative nature of DBT iteration cycles and the impact of performing frequent DBT iteration cycles on lead time.

This paper is organized as follows. A brief discussion of the VDP focusing on DBT iteration cycles is presented in section 2. A system dynamics model for the VDP and simulation results are discussed in sections 3 and 4, respectively. We conclude the paper with a discussion on the organizational aspects of performing frequent DBT iteration cycles.

2. Vehicle Development Process

A simplified overview of the VDP is presented in <Figure 1>. In the VDP, three types of design iteration can be identified: internal design and external design iterations, and DBT iteration cycles. Preliminary design concepts for a component or a subsystem are refined and arrived at after performing hundreds of planned design tasks. The repetition of tasks within a design group is referred to as "internal design iteration."



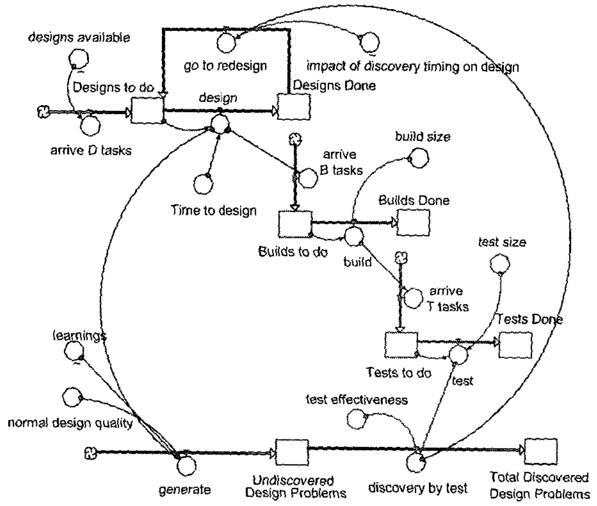
<Figure 1> Overview of the VDP

The external design iteration refer to the repetition of design tasks among design groups in which design engineers from various design groups interact and arrive at the initial designs for all interface zones between components or subsystems.

Once a sufficient number of components and subsystems are designed, prototypes are built and tested to validate the design concept and discover unknown design problems. The configuration of prototype builds and tests depends on the objectives and the level of vehicle integration at each development stage - such as concept generation, product planning and engineering, and process engineering. The test results are then fed into all related design groups to correct and refine the design of components and subsystems. This process is then repeated until satisfactory results are achieved and is referred to as "DBT iteration cycles." The purpose of these DBT iteration cycles is to discover unknown design problems, which are common in complex vehicle development and escape the quality check earlier in the process - even though the design went through internal and external design iteration. These unexpected problems are usually caused by conflicts in integrating components and subsystems.

3. A System Dynamics Model for the VDP

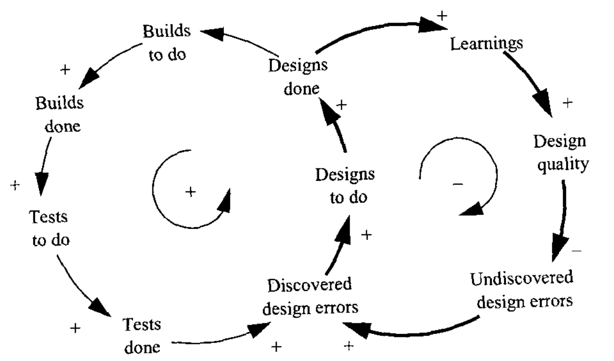
A system dynamics model was developed based on the previous system dynamics models as shown in <Figure 2> [4-8, 10]. Once the voice of customer and the market needs are identified, the technical requirements are developed, which will determine the design scope (designs to do). When a sufficient number of designs is accomplished, physical prototypes are built and are tested to identify whether the designs meet the requirements successfully or not. If design problems are discovered then the released designs are sent back to the related design groups for redesign and the whole process is repeated until tests show no more design problems.



<Figure 2> System Dynamics model for the DBT iteration cycles in the VDP

Since all the "designs to do" are not available at the same time when a project is started, it is assumed that the "designs to do" are available linearly until one hundred weeks resulting in 1000 "design tasks to do" (designs available). The amount of design problems to be discovered in the test stage is primarily determined by the design quality (normal design quality) and affected by the learning as design progress is made. The error discovery rate is affected by the effectiveness of the tests. The effects of iteration and thus learning from design, build, and tests are also incorporated and are a function of the design, build and test progress. Rework has a greater impact as the vehicle design progresses and is included in the system dynamics model - the impact of rework when most of the designs are accomplished and released is much higher than when the designs have just started (impact of discovery timing on design).

The system dynamics model consists of one positive and one negative major feedback loop, as shown in <Figure 3>. The discovered design problems increase the "designs to do," but this positive feedback loop is balanced by the learning. Through working on designing, building, and testing, more knowledge is gained and the design quality improves, thus reducing the number of design problems.



<Figure 3> Major feedback loop for the DBT iteration cycles

4. Analysis of the DBT iteration cycles in the VDP

Input variables are selected to demonstrate the effects of different number of DBT iteration cycles and do not represent a vehicle manufacturer. There are 1000 tasks "to do" for each design, build and test. The initial design quality was set to seventy percent and indicates that, for example, whenever ten designs are accomplished, three designs will be discovered infeasible and have to be reworked after testing. Time to design, build, and test are ten weeks respectively.

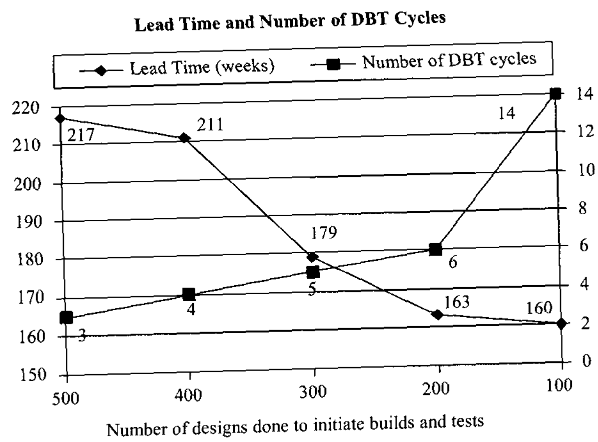
A series of simulations were performed to analyze the transient behavior of DBT iteration cycles by changing the number of "designs done" from 100 to 500 in steps of 100 design tasks, which initiates building and testing. All the input and initial values remain unchanged, except the number of "design tasks done" in a DBT iteration cycle to complete cycle. The cases analyzed are as follows.

Case 1 : Builds and tests start once 500 designs are done
Case 2 : Builds and tests start once 400 designs are done
Case 3 : Builds and tests start once 300 designs are done
Case 4 : Builds and tests start once 200 designs are done
Case 5 : Builds and tests start once 100 designs are done

4.1 Development Lead Time and the Number of DBT Iteration Cycles

The development lead time and the number of DBT iteration cycles changing the number of "design tasks done" in a DBT cycle is shown in <Figure 4>. Since the project scope requires 1000 tasks to accomplish, it is expected to have two DBT cycles. However, because of the undiscovered rework actually there are three DBT cycles.

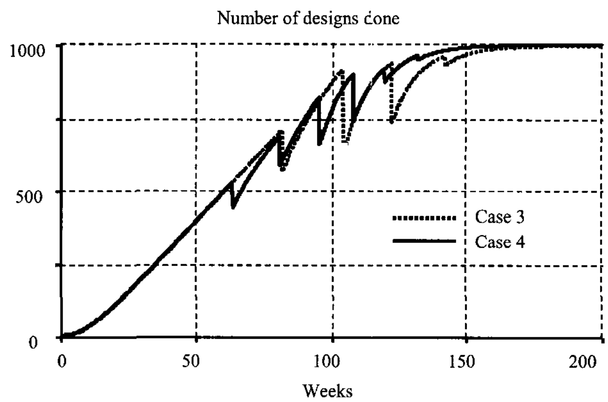
The figure shows that there are non-linear relationships in lead time and the number of DBT iteration cycles. The development lead time decreases as the number of "designs done" to initiate builds and tests decreases, until the lead time flattens at 200 "designs done". The number of DBT iteration cycles increases as the number of "designs done" to initiate a DBT cycle decreases. The number of DBT iteration cycles increases greatly when fewer than 200 designs are done to initiate builds and tests (case 3). From the simulation results, the optimal number of DBT cycles can be determined by considering the lead time reduction and the setup costs for build and tests.



<Figure 4> Lead time and number of the DBT cycles

4.2 Design Progress

In order to understand the dynamic behavior of DBT cycles, cases 3 and 4 are discussed in detail. The number of "designs done" for case 3 and case 4 is shown in <Figure 5>. For case 3, the total number of "designs to do" is 1000, therefore three DBT iteration cycles are expected. However, because of the discovered design problems, five DBT iteration cycles performed actually. Even though 300 designs are done at 50 weeks, the first DBT iteration cycles is not completed until 80 weeks because of the build and test delays. This is almost half of the development lead time. Designs are continuously performed during the build and test stages, and the development lead time is 179 weeks. The discovered design problems force engineers to withdraw the "designs done" for reworks. The drops in "designs done" represent the amount of discovered design problems which have to be corrected in the next phase. The fifth DBT cycle reveals no design problems to be reworked, and the design of a vehicle is complete. The discovery of design problems in the later stages of the development creates a lot of confusion and fire fighting, as the project completion date approaches.

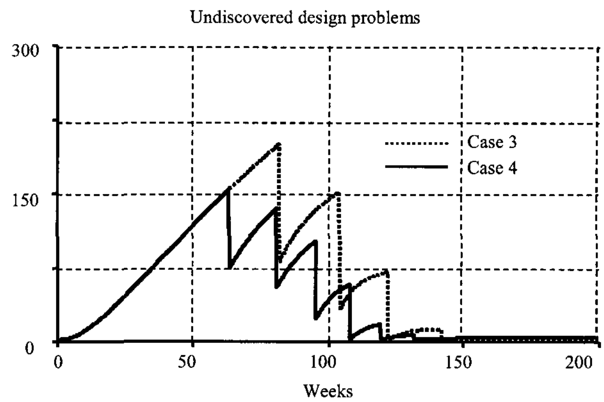


<Figure 5> Number of "designs done" for case 3 and case 4

For case 4, the first DBT cycle is completed at 60weeks. It shows smaller drops in the "designs done" than that in case 3. The development lead time is 164 weeks, and there are nine DBT iteration cycles in total.

The benefit of performing frequent DBT iteration cycles is shown after 120 weeks. The number of "designs done" for case 3 is higher than that in case 4, until around 100 weeks. However, this is reversed after 120 weeks, since the discovered design problems are corrected earlier and there are not many design errors to be worked on. This demonstrates that earlier discovery of rework provides an improved opportunity to resolve a problem, especially as the project completion date nears.

The undiscovered rework for the cases 3 and 4 is shown in <Figure 6>. It reveals that as frequent DBT cycles are performed, the average undiscovered rework lessens.



<Figure 6> "Undiscovered design problems" for case 3 and case 4

The benefits of performing frequent DBT iteration cycles, thus plan-do-check-action cycles concept, can also be applied to internal and especially external design iteration. The frequent and early discovery of design problems among design groups will reduce the time delay in discovering design problems and promote bilateral communication. One noticeable behavior is the lack of progress after 100 weeks. This is the result of correcting the discovered rework and is known as the "90 percent syndrome" in the new product development processes [5-7].

4.3 Perceived and Real Design Progress

Perceived design progress means that undiscovered rework is not included when computing design progress, while the real design progress contains the undiscovered rework. The formulae used to compute the perceived and real design progress are :

Perceived design progress

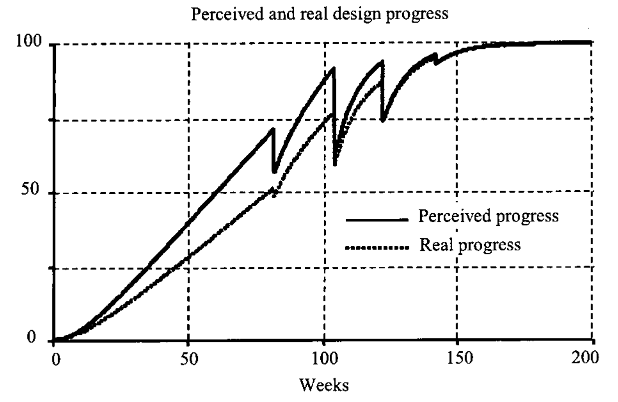
= (designs done)/design tasks to do

Real design progress

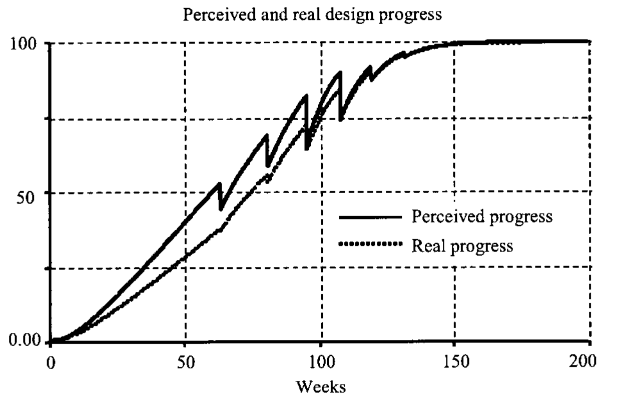
= (designs done-undiscovered design problems) / design tasks to do

The perceived and real design progress for cases 3 and 4 are shown in <Figure 7> and <Figure 8>, respectively. Case 3 shows a larger gap between the perceived and real design progress than case 4. The gap for case 3 lessens after

0125 weeks, whereas the gap for case 4 lessens after 110.



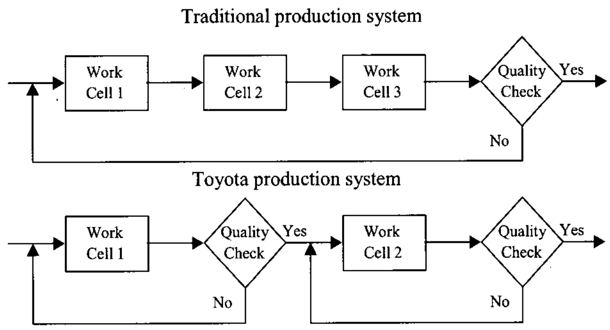
<Figure 7> Perceived and real design progress for case 3



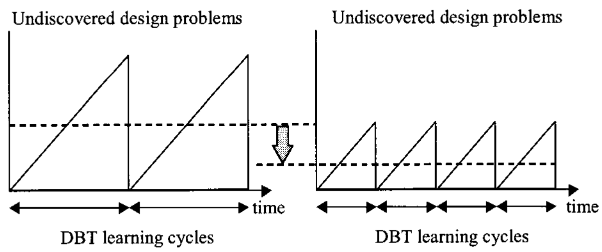
<Figure 8> Perceived and real design progress for case 4

The gaps between the perceived and real design progress are critical in project management because of scheduling and resource allocation. Underestimating the undiscovered design problems will create confusion and fire fighting. As the project completion data is approaching, the usual way to recover the stagnated design progress is to either allocate new engineers or to reduce the project scope. However, allocation of new engineers who are unfamiliar to the project will hinder the team's overall productivity and quality [4-8]. Also the reduction of the project scope usually resulted in a mediocre product.

The effects of performing frequent DBT iteration cycles correspond with the recommendations from the Toyota production system, as shown in <Figure 9>: reduction of a lot size and quality control after a part is processed (a DBT iteration cycle in the VDP). One of the recommendations from the Toyota production system is the reduction of a lot size, which can be considered as the number of "designs done" and undiscovered reworks to initiate a DBT iteration cycle in the VDP. The undiscovered rework can act as a lot sizing problem in economic order quantity (EOQ) in inventory control, as shown in <Figure 10>. Also the Toyota system encourages workers to perform quality checks immediately when a part is processed, which prohibits defective parts from flowing to the next work cells and consuming valuable time and efforts. By performing more frequent DBT iteration cycles, infeasible design problems are detected earlier and more frequently, and prevent further design refinements from being performed on infeasible designs.



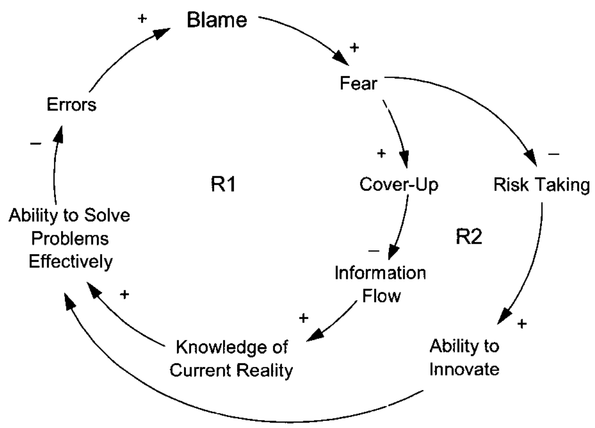
<Figure 9> Quality control in Toyota production system



<Figure 10> The impact of undiscovered design problems and frequent DBT cycles

5. Organizational Considerations for Performing Frequent DBT Iteration Cycles

The frequent design iteration and DBT iteration cycles require careful modifications in the organization, specifically regarding the generation and discovery of rework. When there has been criticism for causing rework, engineers hesitate to release design information early and frequently. Krishnan contends that preliminary design information should be exchanged and utilized [12]. Paul shows the undesirable effects of blame as shown in <Figure 11> [15].



<Figure 11> The reinforcing cycles of blame

Blame often results in the last moment, lump sum release of design information in the later stages, which usually creates over-design problems in the VDP. The benefits of "Do it right the first time" cannot be over-emphasized, but "doing it right" requires careful attention. It could have undesirable effects -engineers may try to make their designs as perfect as possible before they release design information, thus delaying learning opportunities. This could lead to asking for more time to perfect their designs by different subsystem groups, thus resulting in possible over-design problems. However, during vehicle integration, unexpected problems still occur due to complexity of a vehicle design. A minor design change will have devastating and cascading effects to other designs once most designs are accomplished.

Clark D.W. suggests that each new design bug should be treated as a positive indicator of progress, instead of regarding design problems negatively [3]. One performance measure for design progress would be the monitoring of the design error discovery rate.

The frequent DBT iteration cycles do not indicate that the hardware build and test costs will increase as more DBT iteration cycles are performed. Advances in computer technology such as Computer-Aided Design and Engineering, Manufacturing, and testing enable many aspects of a vehicle to be designed, built, and tested with computer models. They can significantly reduce the long lead time in prototype tooling, building and testing. They can also avoid the unfortunate reality that the lessons learned from the hardware builds and tests often do not reach the related design groups before the next refinement of the design is achieved. The advanced computer technologies should be employed as a means of locating design problems as early and frequently as possible rather than as a primary tool to reduce the number of hardware builds and tests. Also they should also be used as a way to increase the number of DBT iteration cycles and thus to improve the learnings.

6. Summary and Conclusion

The VDP is an iterative process of "design-build-test," with numerous coupled tasks that are performed by different functional areas. The design of a new vehicle evolves through several development stages before it is finalized. By going through the internal and external design iteration and DBT iteration cycles, components, subsystems, and the vehicle are synthesized, analyzed and validated to develop and meet requirements. The purpose of the DBT iteration cycles is to identify unknown design problems, which are common in complex vehicle development. These unexpected problems are usually caused by conflicts in integrating components and subsystems.

A simple system dynamics model was developed in order to understand the benefits of performing frequent DBT iteration cycles. This model provided a stepping-stone for further in-depth understanding of the effects of DBT iteration cycles in new product development processes. Finding design problems early and frequently in the development stages can reduce chances that not realizable designs are further developed using erroneous design information. It reduces the number of new problems for the next DBT iteration cycles, resulting in a relatively lower cost of change. The effects of performing frequent DBT iteration cycles correspond with the recommendations from the Toyota production system. The concept of EOQ in inventory control could lead to a new research area in order to understand the effects of undiscovered design problems on development lead time.

References

[1] Blackburn, J. D.; *New product development : the new time wars*, In : Time-based competition, Business One Irwin, Homewood, IL., 1991.

[2] Clark,, K. B. and Fujimoto, T.; *Product development performance: strategy, organization, and management in world auto industry*, Harvard Business School Press, Cambridge, MA., 1991.

[3] Clark, D. W.; "Bugs are good : a problem-oriented ap-

proach to the management of design engineering," *Research Technology Management*, pp. 23-27, May-Jun. 1990.

[4] Cooper, K. G.; "Naval ship production: a claim settled and a framework built," *Interfaces*, 10(6) : 20-36, 1980.

[5] Cooper, K. G.; "Benchmarks for the project manager," *Project Management Journal*, 1993.

[6] Cooper, K. G.; "The rework cycle : why projects are mismanaged," *PMNETwork*, 1993.

[7] Cooper, K. G.; "How it really works and reworks," *PMNETwork*, 1993.

[8] Cooper, K. G.; "The $2,000 hour: how managers influence project performance through the rework cycle," *Project Management Journal*, 15(1) : 11-24, 1994.

[9] Eppinger, S. D.; "Model-based approaches to managing concurrent engineering," *Journal of Engineering Design*, 2(4), 1991.

[10] Ford, D. N.; *The dynamics of project management*, Ph.D. thesis, MIT, 1995.

[11] Ha, A. Y. and Porteus, E. L.; "Optimal timing of reviews in concurrent design for manufacturability," *Management Science*, 41(9) : 1431-1447, 1995.

[12] Krishnan, V., Eppinger, S. D., and Whitney, D. E.; "Accelerating product development by the exchange of preliminary product design information," *J Mechanical Design*, 117 : 491-497, 1995.

[13] Krishnan, V.; "Managing the simultaneous execution of coupled phases in concurrent product development," *IEEE Transactions on Engineering Management*, 43(2), 1996.

[14] Kusiak, A. and Wang, J.; "Efficient organizing of design activities," *International Journal of Production Research*, 31(4) : 753-769, 1993.

[15] Paul, M.; "Moving from blame to accountability," *The System Thinker*, 8(1), 1997.

[16] Robers, E. B.; *The dynamics of research and development*, New York : Harper & Row, 1964.

[17] Robers, E. B.; "A simple model of R&D project dynamics", *R&D Management*, 5(1), 1974.

[18] Richardson, G. P. and Pugh, A. L.; *Introduction to system dynamics modeling with Dynamo*, MIT Press, Cambridge, MA, 1981.

[19] Rodrigues A. and Bowers, J.; "System dynamics in project management: a comparative analysis with traditional methods," *System Dynamics Review*, 12(2) : 121-139, 1996.

[20] Steward, D. V.; "The design structure system: a method for managing the design of complex systems," *IEEE Transactions on Engineering Management*, 28(3) : 71-74, 1981.

[21] Smith, P. R. and Eppinger, S. D.; "Identifying controlling features of engineering design iteration," *Management Science*, 43(3) : 276-293, 1997.

[22] Smith, P. R. and Eppinger, S. D., "A predictive model of sequential iteration in engineering design," *Management Science*, 43(8) : 1104-1120, 1997.