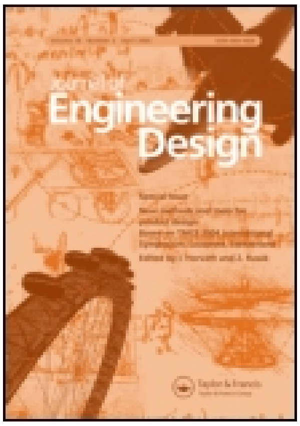


Journal of Engineering Design
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/cjen20

# Improving product development process design: a method for managing information flows, risks, and iterations

Darian Unger [a] & Steven Eppinger [b]

[a] School of Business, Howard University, 2600 6th Street NW, Washington, DC, USA

[b] Sloan School of Management, Massachusetts Institute of Technology, 50 Memorial Drive, Cambridge, MA, 02142, USA

Available online: 27 Oct 2010

*Journal of Engineering Design*
Vol. 22, No. 10, October 2011, 689–699

# Improving product development process design: a method for managing information flows, risks, and iterations

Darian Unger[a]* and Steven Eppinger[b]

[a]*School of Business, Howard University, 2600 6th Street NW, Washington, DC, USA;* [b]*Sloan School of Management, Massachusetts Institute of Technology, 50 Memorial Drive, Cambridge, MA 02142, USA*

(*Received 12 September 2009; final version received 15 September 2010*)

Companies employ a variety of different product development processes (PDPs) to design new products. Well-designed PDPs are necessary to reduce development time, manage risks, and create better products. This study proposes and demonstrates a method with which a company can design a PDP applicable to its unique circumstance. The research uses several case studies and a literature survey to demonstrate the wide variety of PDPs available to companies. The case studies show the importance of matching PDP design to a company's risk profile. The paper employs these lessons to propose a method for improved PDP design. The paper then illustrates the method by discussing how one company is using it to redesign its PDP.

**Keywords:** product development process design; systematic product development < design theory and research methodology; models of the design process

## 1. Introduction

Product development processes (PDPs) are the procedures and methods that companies use to design new products and bring them to market. Competition, technological advancement, market changes, and product life cycles all force companies to develop new products frequently. The goal of this case-based research is to help companies develop products more efficiently by describing and demonstrating a design method to customise PDPs to address the risk profiles of individual companies.

PDPs can vary widely, as shown by product development literature and several company case studies. This paper identifies key components of PDPs and demonstrates how PDPs can be designed and structured differently to manage different risks. The paper also proposes a PDP design method that companies can use to either select or design PDPs that best match their risk profiles. This method was tested when a technology-oriented company adopted it to redesign its PDP, and the results are discussed. Use of the model can help companies avoid making process choices based on ideology and instead design processes for individual needs, leading to faster and better product development.

*Corresponding author. Email: dwunger@howard.edu

ISSN 0954-4828 print/ISSN 1466-1837 online
© 2011 Taylor & Francis
DOI: 10.1080/09544828.2010.524886
http://www.informaworld.com

This paper begins with a review of relevant literature and a summary of design process problems. It then describes business cases derived from companies' process documentation, company interviews, and an approach used in previous research to compare PDPs. That approach included examination of companies' processes through documentation and interviews. Finally, the paper describes how this design method is being applied at a specific company and draws conclusions about its effectiveness.

## 2. Literature review and design process problem

Industrial PDPs vary. Recognition of such PDP variety is relatively new because a traditional, staged process dominated US industry for almost 30 years (Smith and Reinertsen 1992, McConnell 1996, Ulrich and Eppinger 2004). The development of other PDPs, especially over the past decade, expanded the menu of choices for many companies as they reached beyond staged processes to speed the design process and reduce costs (Lim *et al.* 2006, Büyüközkan *et al.* 2007). For example, recent work by Erat and Kavadias (2008) compares older product development prototyping processes and finds them to be insufficient, given current complex architectures. Ho and Lin (2009) present an alternative product development method based on concurrent function deployment and concurrent engineering. Ibusuki and Kaminski (2007) suggest value-engineering PDPs. However, the biggest competitors to traditional staged models are the variants of more flexible spiral processes that have been adapted from software development (Cooper and Edgett 2008). These spiral processes intentionally use design cycles to manage risk (Boehm and Bose 1994, McConnell 1996, Hicks and McGovern 2009). Together, design management research combines with changing industry practices to demonstrate a variety of PDPs. Given an array of choices, designing or selecting the ideal process poses a challenge.

This paper proposes a PDP design method based on risk management to address the PDP design and selection problem described above. The risk framework is important to PDP design because companies face several risks during new design; numerous authors stress the need to develop effective risk strategies as part of product development (Cooper 2003, Lough *et al.* 2009). Several studies organise the dimensions of risk or categorise development risks as primarily technical, market, budget, or schedule (Keizer and Halman 2009). Technical risk stems from uncertainty about whether a new product will be able meet its own functional and design specifications. Market risk stems from whether those design specifications meet customer needs; if not, a technically successful product could fail in the market (De Meyer *et al.* 2002). PDPs manage risk partially through iterations, which are controlled, feedback-based redesigns. Small iterations may include minor changes to components, while large iterations may include marketing feedback that changes the overall design. PDPs also manage risk through reviews, which are gates or checks between development stages that are meant to confirm adequacy. Strict reviews prevent further design until early work is finalised, while flexible reviews allow more parallel work (Unger and Eppinger 2009).

### 2.1. *Process options and differences*

Staged processes, as illustrated in Figure 1, were popular for decades because of their controlled design structures. These processes methodically follow a series of steps, are characterised by few iterations and rigid reviews, and tend to freeze design specifications early. Frozen specifications help companies by providing stability, creating sharp product definitions, avoiding scope creep, and reducing the need for midstream corrections. Staged processes perform especially well when product cycles have stable product definitions, have high quality standards, and use well-understood technologies, as is often the case for product updates (Cooper 2001, Otto and Wood 2001, Unger and Eppinger 2009).

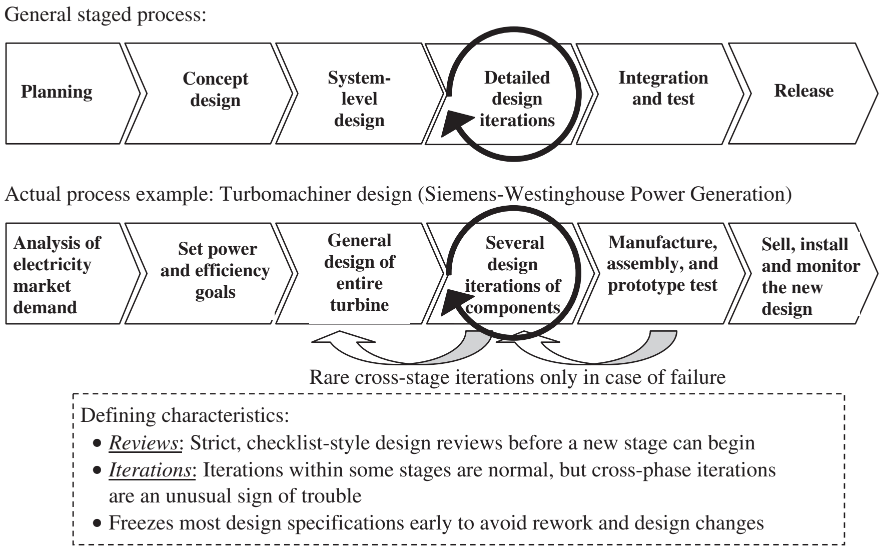
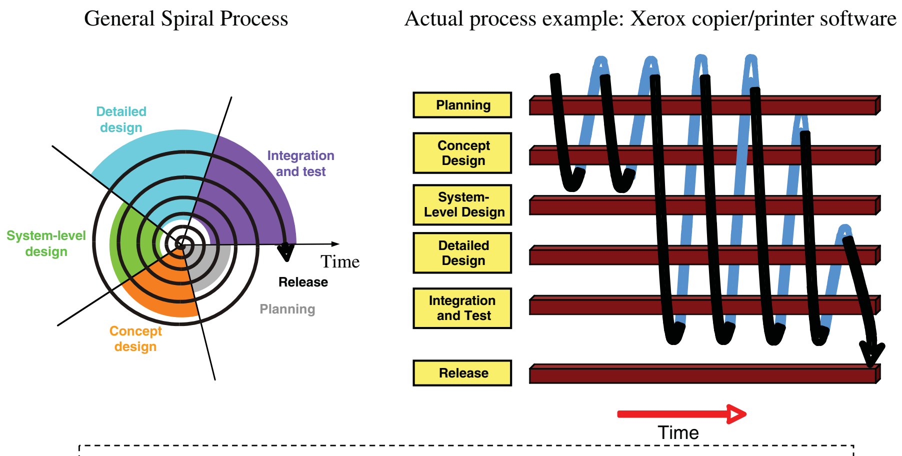

General staged process:
Planning
Concept design
System-level design
Detailed design iterations
Integration and test
Release
Actual process example: Turbomachiner design (Siemens-Westinghouse Power Generation)
Analysis of electricity market demand
Set power and efficiency goals
General design of entire turbine
Several design iterations of components
Manufacture, assembly, and prototype test
Sell, install and monitor the new design
Rare cross-stage iterations only in case of failure
Defining characteristics:
• Reviews: Strict, checklist-style design reviews before a new stage can begin
• Iterations: Iterations within some stages are normal, but cross-phase iterations are an unusual sign of trouble
• Freezes most design specifications early to avoid rework and design changes

Figure 1. A staged PDPs.

Figure 1 first displays a general process that shows the regimented, staged nature of the PDP. Stages are generally separate and sequential, except for the detailed design stage, which includes several internal iterations before the prototyping can begin. The staged process is called the ‘waterfall’ process by some practitioners (Erdogmus and Williams 2003) because of its one-way nature. Once a stage is complete, it is generally difficult or expensive to go back. However, companies sometimes must revisit design issues from previous stages. The second part of Figure 1 displays a case study result supporting this finding. The process and practice from Siemens–Westinghouse Power Generation makes accommodation for cross-phase (as opposed to internal) iterations in case of serious problems or failure. Such backward steps are rare, but possible.

The most important limitations of staged processes are their rigidity and long lead times (Anand and Kodali 2008, Biazzo 2009). Staged processes are sometimes too inflexible for companies in dynamic markets that require short development times or changes during development. These deficiencies manifest themselves in two ways. First, staged processes can expose companies to market risk, especially if early specifications or assumptions are poor. An example of market risk is when a company develops a product that meets design specifications perfectly, only to find from early prototypes or market research that the design specifications missed evolving market demands. Learning about this design problem near the end of the process makes corrections difficult or impossible; so staged processes are sometimes difficult for projects with vague or changing requirements. Second, staged processes sometimes have difficulty handling parallel tasks within stages. As a result, the length of each stage may be constrained by the slowest activity within the stage, thus lengthening the development process and delaying new designs (Smith and Reinertsen 1992).

These limitations led companies to explore other development processes. One of them, the spiral PDP, differs from the staged process because of its flexibility and anticipation of feedback (Zamenopoulos and Alexiou 2007). Figure 2 shows two versions of the spiral process: the first is a generalised version developed by Unger and Eppinger (2009). The second is an actual, specific version from a company case study of Xerox, Inc. Unlike the staged processes, the spiral process shown in Figure 2 includes a series of planned iterations that incorporate feedback and span several phases of development. It is a relatively recent PDPs that has been adopted by many software



Defining characteristics:

- *Reviews* : Less formal reviews resemble status-checks during different loops
- *Iterations* : Cross-stage iterations for feedback are normal
- Allows flexible design specifications early to avoid the need for major rework

Figure 2. A spiral PDPs.

companies. Spiral process proponents assert that it reduces burdensome and expensive rework in software, thus lowering development time and cost (Boehm 1988, Gilb 1988, McConnell 1996, Su *et al*. 2007).

The spiral PDP repeats regular steps, including concept development, system level design, detailed design, and integration and testing. The process is flexible; the actual number and span of loops can vary. The spiral process requires managers to evaluate risk early in the project, which is useful because major costs have yet to be incurred (Boehm 1988, Yoram and Paz 2008). The spiral process also has several disadvantages. First, its complexity requires significant management attention. Second, the lack of rigid specifications can potentially lead to delays in developing complex subsystems (Unger 2003). Finally, a simple spiral process with minimal uncertainty and only one loop would closely resemble a staged process; thus, the spiral process may be overkill for ordinary projects that could use a simpler staged process (Boehm and Bose 1994).

The processes described above are only two of many potential processes, but they form two ends of a spectrum for PDPs. At one end of the spectrum, staged processes maximise managerial control and focus on managing technical risk (whether a design will actually work well). At the other end of the spectrum, spiral PDPs maximise flexibility and focus on managing market risk (whether a design will match consumer needs). In between the two ends are an array of PDP variants, including design-to-budget, evolutionary prototyping, and several others. Design-to-budget processes begin as staged processes, but then combine the detailed design and testing stages to induce developers to iterate through several designs until a budget limit is reached. This has the advantage of controlling costs, but puts market applicability and quality at risk. Evolutionary prototyping, on the other hand, focuses on gaining fast feedback from early prototypes. This can be advantageous when initial specifications are vague and would benefit from experimentation, but the process has no clearly-defined end. Each of these processes has distinct advantages and disadvantages, suggesting that companies should design or select the most appropriate PDPs for their own specific cases. The next section includes a visual explanation of how the processes may be compared, thus providing a research methodology that can be used for process design as well as comparison.

## 3. Cases and comparative approach

Krubasik (1998) argues that 'product development is not monochromatic ... not all product development is alike. Each situation has a different context ... [implying] different managerial actions'. Yet choosing or designing a PDP is difficult, and is complicated by the fact that each process has its champions and star examples. For example, Cooper (2001), Pahl and Beitz (1996), Smith and Reinertsen (1992), and Song *et al.* (2009) argue persuasively for the effectiveness of the staged PDP. Others advocate the use of the spiral process in software development, denounce the deficiencies of rigid staged processes, and call for flexible prototyping (Boehm 1988, Gilb 1988, Hekmatpour and Ince 1988, Boehm and Bose 1994, McConnell 1996, Su *et al.* 2007). Case studies and models documenting individual companies' processes include a range of companies from software designers to automobile manufacturers (Cusumano and Selby 1995, Ward *et al.* 1995, Cusumano and Nobeoka 1998, MacCormack 2000, Levardy and Browning 2009). This range of PDPs makes it difficult to compare them, much less select or design PDPs.

To address this problem, the research in this paper applies a paradigm introduced by two relevant studies (Unger 2003, Unger and Eppinger 2009) which compared a variety of PDPs from the software, power generation, automotive, defence, communication, and aviation industries. All companies' processes were determined through examination of their process documentation and through interviews with designers, engineers, and managers. The company case studies include the following:

- Siemens–Westinghouse Power Generation, which used a strict stage gate process to design turbomachinery.
- Aviation Technology Systems, which used a spiral process to design air traffic control software.
- United Technologies Corporation, which used variants of stage gate processes to design helicopters and jet engines.
- Xerox, Inc., which used a stage gate process with a nested spiral process to develop copier and printing machines.
- Integrated Development Enterprises (IDe), which used an evolutionary delivery process to design development change management software.
- ITT Industries, which used a staged process with progressive freezes of functional specifications to design military radios and satellites.
- Microsoft, which used a spiral process in designing operating and application software.

In examining these and other case studies, earlier work developed a system of comparison – which is adapted in Figure 3 as an original matrix – that uses iterations and reviews to create unique identifiers for each type of process, allowing companies to be categorised by the type of PDP they used and the type of risk (market or technical) they faced (Unger and Eppinger 2009). Figure 3 uses the metrics from this previous research to create a new matrix that maps different PDPs by iteration and by design review with a 0–5 scale on each axis. Case studies indicated that processes with rigid reviews and narrow iterations tended to be staged processes that manage technical risk well, while market risk was better managed by spiral processes with flexible reviews and comprehensive iterations.

Once designers analyse PDPs through the lens of iteration and review, they can sort them not by shape, but by these key characteristics and their ability to manage certain types of risk. For example, prior research was able to identify and place existing companies' PDPs in a matrix similar to Figure 3. Siemens–Westinghouse Power Generation, for example, had an iteration score of 1 and a review score of 1.5, placing it firmly in the quadrant representing staged processes. Microsoft, and IDe, on the other hand, employed the comprehensive iterations and more flexible reviews typical of spiral processes (Unger and Eppinger 2009). The present research suggests that

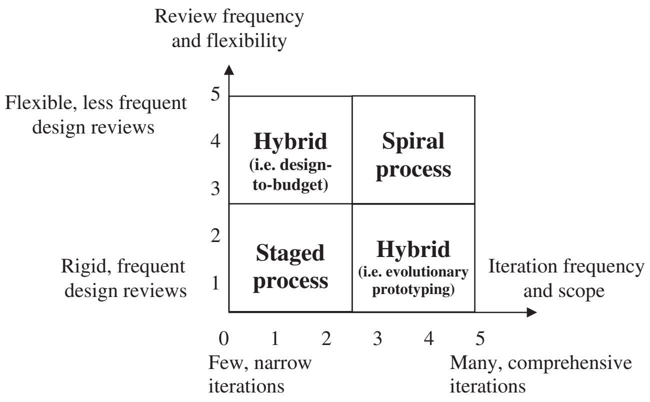

Review frequency and flexibility
Flexible, less frequent design reviews
Rigid, frequent design reviews
Hybrid (i.e. design-to-budget)
Spiral process
Staged process
Hybrid (i.e. evolutionary prototyping)
Iteration frequency and scope
Few, narrow iterations
Many, comprehensive iterations
0 1 2 3 4 5

Figure 3. PDP comparison based on common review and iteration characteristics.

the method displayed in Figure 3 can be applied so that it serves not only as a basis for comparison, but also as a tool for PDP design. If companies can identify their primary development risks, they can then tailor specific design reviews or iterations in their new PDPs.

## 4. Applying case study lessons: proposing a PDP design method

Case studies and prior research demonstrated that companies implement vastly different PDPs. They also demonstrated that managers lack guidance in choosing or designing such processes. Simply adopting a PDP that works for another company is unlikely to lead to success; yet, some companies employ ill-fitting PDPs based on history, mimicry, ideology, or a vague comparison of process shapes and diagrams. Instead, PDPs should be methodically customised to different companies, and thus the need for an improved PDP design method is real and immediate. This section applies lessons from the case studies to propose and apply a helpful tool: a PDP design method that can assist companies in planning or selecting their PDPs. The PDP design method matches risks to specific process iterations and reviews, thus helping companies design processes that suit their own risk profiles and abilities to integrate or prototype. To demonstrate its utility, this research applies the method to redesign the PDP at a manufacturing company.

### 4.1. *PDP design method proposal*

In formulating a new PDP design method, the relevant lessons from case study results include the following:

- Companies face unique sets of individual development risks and integration abilities that should be the basis for PDP design.
- PDPs are composed of design iterations and reviews. We should consider how best to use iteration and review cycles in customised PDP designs.
- Design iterations can be employed to address various risks. The specific risks addressed depend upon what activities are involved in the iterations and upon the timing of the iterations. For example, market risks can be managed by iterations that allow market feedback during the design process.
- Design reviews may also help in managing risks, both in conjunction with and independently of iterations. The types of risks managed depend on the characteristics of the design reviews.

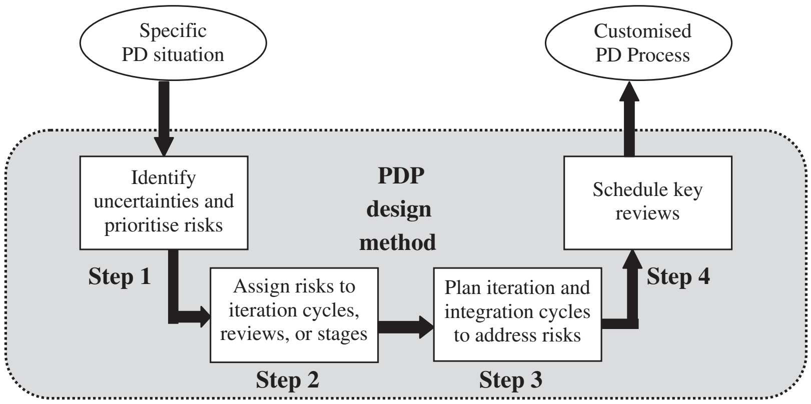

Specific PD situation
Customised PD Process
Identify uncertainties and prioritise risks
PDP design method
Schedule key reviews
Step 1
Assign risks to iteration cycles, reviews, or stages
Plan iteration and integration cycles to address risks
Step 4
Step 2
Step 3

Figure 4. PDP design method.

For example, budget risks can be managed with the help of frequent reviews that provide necessary control.

To translate these descriptive lessons into prescriptive actions that can help companies better design PDPs, this research proposes a PDP design method with four key steps. Not every PDP needs to be designed from scratch; so the same steps can be used either to select which of many existing PDPs may best fit a company, or to modify a process already in use. Figure 4 illustrates the steps and shows how the proposed PDP design method helps in matching companies with appropriate PDPs.

The method begins by identifying and prioritising the risks faced in a specific development programme. The risk identification is based on both past experience and recognised uncertainties. Past experiences, such as a history of lateness or product quality issues, are a relatively easy way for PD planners to estimate which risks are greatest in a current development programme. Uncertainties that are recognised, such as ambiguous customer focus group results or the knowledge that a company is introducing a new product to an untested customer base, can also help identify risks once potential impact costs are assigned to those uncertainties. The company should be able to categorise most risks as technical, market, schedule, or budget, although some risks will defy classification. For example, specification definition often falls in the category of market risk, but a hardware/software interface issue that could arise during integration might result in both technical and schedule risk. Risks are then prioritised. There are often one or two high-probability risks that are likely to ensure failure if they are not addressed. The resulting risk profile becomes the focus of the PDP design method.

The case studies have shown how iterations and reviews address risks. Once risks have been identified and prioritised, each risk is then assigned to a planned iteration cycle (either within or across PD stages) and to a design review. For example, market risk and uncertainty about whether a design will meet customer needs is often addressed by broad feedback and design iterations. Low-priority risks can be simply assigned to stages rather than to an iteration and review combination. For example, technical risk regarding the design of an isolated product component can be assigned to a detailed design stage with only minor (intra-stage) iterations among design engineers. More complicated risks are assigned to specific planned iteration cycles and reviews. For example, high customer uncertainty and the resulting market risk may be assigned to planned, cross-stage iterations that incorporate one prototype or customer test per cycle. Each iteration provides feedback that reduces risk in the next round. A critical product launch date giving rise to high-schedule risk (linked also to budget risk) may prompt a company to include reviews at regular time intervals rather than only at the end of each stage.

Once the risks are assigned to iteration cycles and reviews, the PDP is designed and defined as that unique combination of iterations and reviews. The result is a PDP that addresses a project's major development risks.

### 4.2. *Method demonstration: Printco application*

The PDP design method has already been applied to a sample company. The company, with the pseudonym Printco, is a 1500-employee engineering and manufacturing company that develops industrial printing technologies. Its printers and coders are capable of marking unusual boxes, bags, and irregularly-shaped products for inventory and shipping. Printco's products are mainly hardware, although some machines also include important software components. The company expressed interest in using the lessons of this research immediately because it was in the early stages of redesigning its own PDP. This section discusses how the method was used to prescribe process improvements for Printco, which has implemented PDP changes based, in part, on these suggestions.

The PDP design method followed the mapping described in Figure 4. In the first step, risks in the company's various PD efforts were identified and prioritised through interviews with designers, engineers, and managers. One example was a 'clean sheet' product: a brand new electronics marker for printing on computer components. This product introduced major technical risks because the marker was so novel that chemical engineers were uncertain about whether the new ink and nozzle combinations would clog or print smoothly on various substrates. Another example was a series product: a new model of a coding device used to print codes on irregular plastic food bags. Market risk outweighed technical risk because its core components were well understood from earlier models. Other products faced both market and technical risk, and were delayed by technical challenges that extended late in their development programmes.

Next, risks were assigned to specific iteration cycles or reviews. It became clear that Printco's risks did not match its existing iteration and review scheme. As shown in the top row of Table 1, the former Printco PDP incorporated design iterations of medium breadth. Most iterations were intra-stage with usually only one or two cross-stage loops which were manifested as rework. The degree of planning of these iterations was low because they were neither encouraged nor overtly expected. Printco reviews were more frequent than its iterations, but lacked rigidity. The result was a mismatch between risks and PDP attributes, and therefore a series of late projects and quality control problems.

To complete the next step, new iterations and reviews were planned. In the case of new ink concerns, the risks were best addressed by the inclusion of three iterations to experiment with

Table 1. The former and proposed Printco PDPs for new products.

| | Iteration | | Review | |
|---|---|---|---|---|
| | Scope | Frequency (number of cross-stage loops) | Flexibility | Frequency |
| Former PDP | 1–2 | 1–2 | 4 | 1–2 |
| Proposed PDP | 2–3 | 2–3 | 2 | 1 |

Proposed PDP incorporates more comprehensive iterations for greater market feedback and reduced market risk

Proposed PDP incorporates more rigid reviews for improved scheduling and reduced technology risk

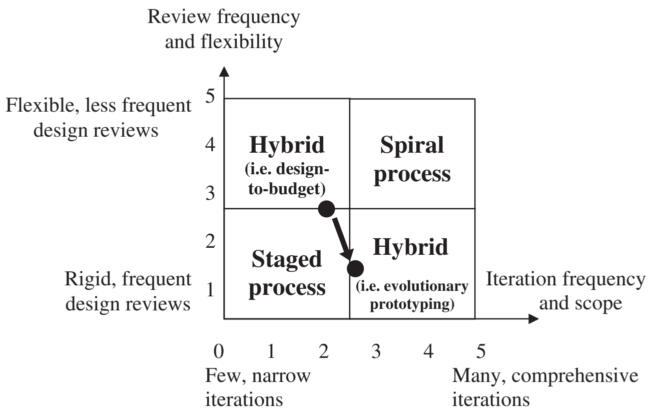

Review frequency and flexibility
Flexible, less frequent design reviews
Rigid, frequent design reviews
Hybrid (i.e. design-to-budget)
Spiral process
Staged process
Hybrid (i.e. evolutionary prototyping)
Iteration frequency and scope
0 1 2 3 4 5
Few, narrow iterations
Many, comprehensive iterations

Figure 5. Mapping of the former and proposed Printco PDPs.

several chemical and nozzle combinations. These iterations required a broader scope than Printco had used previously because they were planned to include prototypes. In the case of market uncertainty with the next generation of a product series, the best way to manage the market risk was to assign the improved printing features to rigid, market-based reviews in order to determine whether the new features addressed customer needs adequately.

In Table 1, the former and recommended PDPs can be compared using the iteration and review metrics described earlier. As shown in Figure 3, spiral-like processes managing market risks tend to have broader design iterations, and therefore have values closer to 5. In contrast, staged processes managing technical risk tend towards low values, which indicate more limited iterations and rigid reviews.

The PDP design method called for reassigning Printco's risks to iterations and reviews that would correct the problems of the old process. It recommended a hybrid approach of rigid reviews and modestly planned prototyping iterations. This would move Printco from the left side of Figure 3 to the lower right quadrant, reflecting increased flexibility in iterations and more restrictive reviews. This can be seen in Figure 5, which superimposes the company's shift in the framework described earlier in Figure 3.

More specifically, the PDP design method recommended that Printco make some of its reviews more rigid to establish greater control over market and schedule risk. It was also suggested that, rather than eliminating the cross-stage iterations (rework) that formerly delayed the product launch, the company should acknowledge the need for such iterations and plan market prototypes within them. Other companies have seen success in incorporating both the rigid reviews of staged processes and the planned, early prototype iterations of spiral processes. Printco, on the other hand, had faced difficulty in building prototypes early enough to garner the information necessary to substantively change design. Information from 'late stages' frequently arrived so late that change was costly and difficult. The company was not taking advantage of its ability to perform more frequent integrations.

These recommendations based on the PDP design method were presented first to Printco's senior PD manager and then to an assembly of engineers, marketers, and managers. The suggestions generated much discussion and were well received. Printco has made PDP changes based, in part, on these recommendations; however, results of the process changes will not be known until at least the end of several product development cycles in which they are implemented. The Printco demonstration does not yet validate the PDP design method, but the positive reaction to the recommendations and projected cost reductions suggest that the method is a reasoned application of the iteration and review lessons from earlier cases. Future research over several years will continue to redesign PDPs at several companies and gather data to see how well the method improves and speeds product design and development.

## 5. Conclusions

Companies are constantly designing and developing new products, and seek to reliably improve their PDPs. Companies have difficulty designing or selecting the process that best suits them, in part because of a history of ideological decisions and the difficulty of distinguishing different PDPs based on shapes or diagrams alone. This research used a series of case studies to define the components that distinguish PDPs from each other, providing companies with a more analytical method of comparison. It then proposed and applied a PDP design method based on risk, iteration, and review.

This PDP design method is a directly applicable research contribution, and provides companies with a framework for efficiently designing PDPs that suit specific project needs. The method allows project planners to evaluate risks and to manage those risks with specific elements of the PDP. Just as segmentation is a valuable tool in marketing products, dividing PDPs into reviews and iterations can be helpful in product design and development. Having good ideas for new products is not enough; to successfully bring those products to market, companies also need to design PDPs that address their specific needs.

## References

Anand, G. and Kodali, R., 2008. Development of a conceptual framework for lean new product development process. *International Journal of Product Development*, 6 (2), 190–224.

Biazzo, S., 2009. Flexibility, structuration, and simultaneity in new product development. *The Journal of Product Innovation Management*, 26 (3), 336–353.

Boehm, B., 1988. A spiral model of software development and enhancement. *IEEE Computer*, 1 (5), 61–72.

Boehm, B. and Bose, P., 1994. A collaborative spiral software process model based on theory W. *Proceedings of the 3rd International Conference on the Software Process, Applying the Software Process*. Reston, VA: IEEE, 59–68.

Büyüközkan, G., Baykasoglu, A., and Dereli, T., 2007. Integration of internet and web-based tools in new product development process. *Production Planning & Control*, 18 (1), 44–53.

Cooper, R.G., 2001. *Winning at new products*. 3rd ed. Cambridge: Perseus Publishing.

Cooper, L., 2003. A research agenda to reduce risk in new product development through knowledge management: a practitioner perspective. *Journal of Engineering and Technology Management*, 20 (1–2), 117–140.

Cooper, R.G. and Edgett, S., 2008. Maximizing productivity in product innovation. *Research-Technology Management*, 51 (2), 47–58.

Cusumano, M. and Selby, R., 1995. *Microsoft secrets*. New York: The Free Press.

Cusumano, M. and Nobeoka, K., 1998. *Thinking beyond lean*. New York: The Free Press, 157–182.

De Meyer, A., Loch, C., and Pich, M., 2002. Managing project uncertainty: from variation to chaos. *MIT Sloan Management Review*, 43 (2), 60–67.

Erat, S. and Kavadias, S., 2008. Sequential testing of product designs: implications for learning. *Management Science*, 54 (5), 956–968.

Erdogmus, H. and Williams, L., 2003. The economics of software development by pair programmers. *The Engineering Economist*, 48 (4), 283–319.

Gilb, T., 1988. *Principles of software engineering management*. Reading, MA: Addison-Wesley Publishing Company.

Hekmatpour, S. and Ince, D., 1988. *Software prototyping, formal methods and VDM*. Reading, MA: Addison-Wesley Publishing Company.

Hicks, C. and McGovern, T., 2009. Product life cycle management in engineer-to-order industries. *International Journal of Technology Management*, 48 (2), 153–167.

Ho, Y. and Lin, C., 2009. A concurrent function deployment-based and concurrent engineering-based product development method for original design manufacturing companies. *Journal of Engineering Design*, 20 (1), 21–55.

Ibusuki, U. and Kaminski, P., 2007. Product development process with focus on value engineering and target-costing: a case study in an automotive company. *International Journal of Production Economics*, 5 (2), 459–474.

Keizer, J. and Halman, J., 2009. Risks in major innovation projects, a multiple case study within a world's leading company in the fast moving consumer goods. *International Journal of Technology Management*, 48 (4), 499–517.

Krubasik, E.G., 1998. Customize your product development. *Harvard Business Review*, November–December, pp. 4–9.

Levardy, V. and Browning, T., 2009. An adaptive process model to support product development project management. *IEEE Transactions on Engineering Management*, 56 (4), 600–620.

Lim, L., Garnsey, E., and Gregory, M., 2006. Product and process innovation in biopharmaceuticals: a new perspective on development. *R & D Management*, 36 (1), 27–36.

Lough, K.G., Stone, R., and Turner, I., 2009. The risk in early design method. *Journal of Engineering Design*, 20 (2), 155–173.

MacCormack, A., 2000. *Towards a contingent model of the new product development process: a comparative empirical study*. Harvard Business School Working Paper Series, 00-077.

McConnell, S., 1996. *Rapid development: taming wild software schedules*. Redmond: Microsoft Press, 133–162.

Otto, K. and Wood, K., 2001. *Product design*. Englewood Cliffs, NJ: Prentice Hall.

Pahl, G. and Beitz, W., 1996. *Engineering design, a systematic approach*. 2nd ed. London: Springer.

Smith, P.G. and Reinertsen, D.G., 1992. Shortening the product development cycle. *Research-Technology Management*, 35 (3), 44–49.

Song, L., Song, M., and Di Benedetto, C.A., 2009. A staged service innovation model. *Decision Sciences*, 40 (3), 571–599.

Su, C., Chen, Y., and Yung-Jye Sha, D., 2007. Managing product and customer knowledge in innovative new product development. *International Journal of Technology Management*, 39 (1/2), 105–130.

Ulrich, K.T. and Eppinger, S.D., 2004. *Product design and development*. 3rd ed. New York: McGraw Hill.

Unger, D., 2003. *Product development process design: improving company response to market pressure, regulation, and changing customer needs*. Thesis (PhD). Cambridge, Massachusetts Institute of Technology.

Unger, D. and Eppinger, S.D., 2009. Comparing product development processes and managing risk. *International Journal of Product Development*, 8 (4), 382–402.

Ward, A., *et al.*, 1995. The second Toyota Paradox: how delaying decisions can make better cars faster. *Sloan Management Review*, 36 (3), 43–61.

Yoram, R. and Paz, A., 2008. Managing product quality, risk, and resources through resource quality function deployment. *Journal of Engineering Design*, 19 (3), 249–267.

Zamenopoulos, T. and Alexiou, K., 2007. Towards an anticipatory view of design. *Design Studies*, 28 (4), 411–436.