Why Working with Reference Models Increases Process Innovation

Albrecht Ricken and Ansgar Steinhorst

How to make process innovation more efficient? New research in methods for product design can provide valuable lessons to business process innovation. The research by MIT Professor Eric von Hippel suggests that product design done by product users is far more efficient than innovation by product makers. Users need an easy toolkit to describe their needs in their domain and language. Reference models such as SCOR are to Business Process Management what the toolkit is to the product design process.

We start by describing today's inefficiencies in Business Process Management. Next we summarize von Hippel's concepts of "Manufacturer-Based Design" and "Iterative User and Manufacturer-Based Design" and their inherent inefficiencies. Popular approaches in Business Process Management, such as BPR and ERP, rely on these concepts, and their inefficiencies share the very same root causes. Von Hippel recommends, rather, employing User-Based Design, which generically avoids these inefficiencies. User-Based Design requires an appropriate toolkit. This paper argues that in Business Process Management, Reference Models for business processes can take the role of a toolkit to drive User-Based Process Innovation. This helps companies to dramatically improve their efficiency.

Why Process Innovation Is So Inefficient Today

To better align their operations with market needs, companies in the early 90s started reengineering their business processes. Business Process Reengineering (BPR) helped companies to increase overall performance using operational improvements.

A little later, Enterprise Resource Planning (ERP) became widespread, and BPR often helped to justify ERP projects. ERP vendors argued, "If you want BPR, implement our ERP system. This makes your BPR projects so much faster. There is no need to reengineer from scratch – just use the built-in and ready-to-run processes that come with the ERP system."

The reality behind such projects looked quite the opposite: Armies of expensive consultants, piles of project documentation that nobody later cared about, disgruntled users, and a high risk of project failure. "Among very big IT projects – those costing over $10m apiece – 98% fall short."[1] "Nearly every sizeable company has its own horror stories about IT projects"[2]. Often enough, an inflexible legacy system was replaced by a new, inflexible ERP system. An already complex system landscape grew even more complex, which meant even more rigid processes and structures. This has been more a roadblock to innovative new processes rather than a facilitator.

Root Cause: Information Does Not Pass

When analyzing the causes for failure, the usual suspects come up: Executive sponsorship, project management, user training, consultants, integration problems, etc. No question, these factors are important, but they hardly explain the high rate of failure.

According to Prof. von Hippel, the root cause lies in "sticky" information." There are always two groups of people that hold different kinds of information:

[1] Kluth, Andreas: Make it simple. www.economist.com, Oct 28th 2004, p2.

[2] Carr, Nicholas G.: Burned by IT. Industrial Engineer, August 2004, p28.

Business practitioners ("users") who have the need for information. They understand the business problems, but have little understanding of the solution.

Technical experts such as consultants ("manufacturer" in the case of product design) with the solution information. They have little understanding of the business problems, but understand the solution.

Requirement information does not pass easily from business practitioners to technical solution experts.

Manufacturer-Based Design

Traditionally, technical experts elicit the need information from the business practitioners by means of interviews and workshops. In a project this phase is called requirements analysis. Then the solution experts develop and implement a solution. A popular representative of this approach is the waterfall method, which was adapted to ERP implementations and reengineering projects.

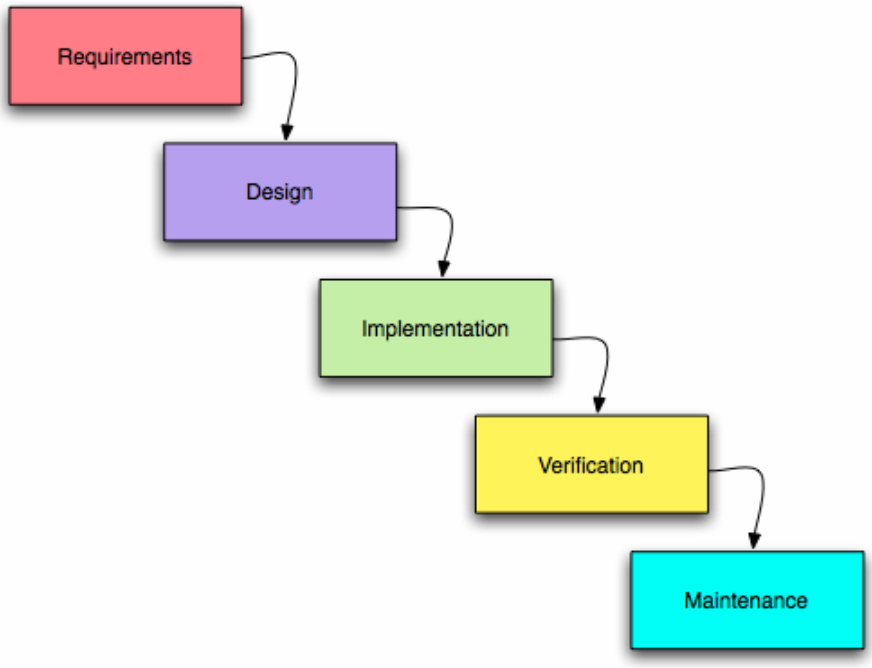

Figure 1. The unmodified waterfall model. Source: http://en.wikipedia.org/wiki/Waterfall_model

In von Hippel's terms, this is Manufacturer-Based Design.[3] "Systems analysts begin the development of a new software product by meeting with users at the start of a project to determine user needs and agree on a written product requirements specification. Manufacturer-based developers then work isolated from further user contact until the completed product is delivered months or even years later."[4]

Iterative User and Manufacturer-Based Design

"Iterative user and manufacturer-based design" allows for trial and error. The approach inserts iteration cycles between business practitioners and solution experts. "Iteration is visible in method steps that

[3] vgl. von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, p 5

[4] von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, p 6

require developers to return to users for more data during the course of a development project."[5] It can be seen in many evolutions of the waterfall method such as the Spiral Model by Barry Boehm[6]. Similar evolutions have been developed for ERP implementations, such as SAP's ASAP method and Peoplesoft's COMPASS. Each claims to allow iterations between users and solution experts.

Six-Sigma is an example of a popular reengineering approach that can include iteration. It has internal consultants ("black belts") who deliver solutions to the process owners. Six-Sigma's DMAIC method stresses the importance of understanding the process owner's needs by means of interviews and data collection. Iteration comes in through the "Voice of the Customer." The first steps of DMAIC require that information pass between business people and solution experts.

Sticky Information

Both approaches, "manufacturer-based design" and "Iterative user and manufacturer-based design," need to pass need information from business people to solution experts. This creates problems:

Requirements may change during or after solution design. Users often realize this, but may not have "an incentive to engage in 'system level' problem-solving that can affect the need."[7]

Information is "sticky." Passing it from the business practitioner to the solution expert is expensive and comes at a high risk of errors and incompleteness. In part this is due to the nature of need information which can be tacit (= not explicit) and, thus, hard to explain. Other reasons are the explanation skills of business practitioners and the interview skills of solution experts, just to name a few.

Von Hippel concludes: "We observe that the cost of transferring information called for by a problem-solver from one location to another can be very high even when the need information has a low stickiness per 'unit' – simply because a great deal of such information may be needed by product and service designers."[8]

Thus, any approach requiring the business to explain its needs to solution experts faces some inherent inefficiencies. The failure rate of ERP and reengineering projects (see above) over the past 15 years gives empirical evidence for this point.

Von Hippel recommends to "avoid user-manufacturer iteration for all projects – because there are inevitably costs and time lags associated with starting up and shutting down problem-solving activities." On the upside, Iterative User and Manufacturer-Based Design is less likely to completely fail than pure Manufacturer-Based Design. This is due to its built-in feedback-loops. On the downside it is likely to incur the costs of passing information several times during any given project and, thus, the most expensive approach.

As an alternative to both, von Hippel proposes reframing the design problems, making the interactions unnecessary. This can be achieved by dividing the problems into sub problems that each group – business people and solution experts – can solve in isolation.[9] He calls his approach User-Based Design.

---

[5] von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, p 6

[6] see Boeh, Barry W.: A Spiral Model of Software Development and Enhancement. Computer, May 1988.

[7] von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, p 9

[8] von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, p 16

[9] see: von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, p 25

User-Based Design

In User-Based Design, it is the user himself who designs his product. Applied to BPM, it is the process owner who innovates business processes for his needs.

This can be very efficient, if users are given a "solution kit" that allows them to design a solution for themselves. The solution kit splits the entire process of solution development into two parts:

The user designs his product. The kit allows him to enter efficiently the need information and visualize the solution. To be certain, the kit needs to appeal to the occasional user. It needs to be easy to use, explain constraints, and allow for trial and error when designing a solution, e.g., via a simulation component.

The need information in the kit passes to the solution experts who in isolation turn it into a product.

Example of User-Based Design – Custom design of integrated circuits[10]:

> *In the past, these circuits were developed in an interactive process between the customer and the manufacturer. The customer would specify required functions, and the manufacturer would design the chip. Then he would send a prototype to the customer for testing. This iteration would go on until the customer was satisfied and approved the design. As every prototype is costly, you would have expected this process to be very expensive.*
>
> *The advent of the Application Specific Integrated Circuit (ASIC) method revolutionized the design process. The chip manufacturer provides customers with software where they can design their custom logic and test it out until satisfied. The software is a CAE[11] package with a user-friendly interface. If relevant to the customer's design task it gives information about solution options and constraints such as the maximum number of physical layers on the chip or the chip size. This helps customer always to design solutions that satisfy his/her needs and stay within the chipmaker's limitations. Design and simulation features allow the customer to experiment in a trial and error mode until satisfied he/she then passes his/her solution to the chip manufacturer.*

The toolkit in the example above helps the customer to design a solution from the customer's perspective. This is not yet the technical realization the manufacturer will produce. The manufacturer certainly may still have to make a substantial additional effort. The toolkit empowers users creating a product specification with the following characteristics:

The user can be reasonably sure that his/her specification matches his/her needs and how it does so.

The manufacturer knows that the user specification is complete, consistent, and within his/her capabilities, and can create the final product without user iteration.

As von Hippel convincingly showed, User-Based Design multiplies design efficiency. The ASIC method reduced design time by more then 2/3.[12]

User-Based Process Innovation

If User-based Design has such a dramatic impact on product innovation, what about the business process domain?

The idea is that practitioners design innovative processes by themselves the way they understand it. The toolkit helps them to articulate and structure need information.

Increasing the efficiency through User-Based Process innovation is the very vision of Business Process Management. The business process layer provides a logical level for practitioners and provides an

---

[10] see: von Hippel, Eric: User Learning, "Sticky Information," and User-Based Design. WP# 3815-95, May, 1995. https://hpds1.mit.edu/bitstream/1721.1/2574/1/SWP-3815-32867610.pdf, pp 26

[11] CAE = Computer-aided engineering

[12] see: von Hippel, Eric: Democratizing Innovation. The MIT Press, 2005, p 148.

abstraction from technical considerations. In fact, the practitioner may ignore what is below the process level, whether this involves outsourcing or changes to an ERP system. Business practitioners design business processes to their needs on-the-fly whenever necessary. Anything below the process layer adjusts, but is not their concern.

The perfect toolkit would let the user state business objectives and the priorities between them and helps the user to stay focused on the business value. It supports an analysis of the As-Is and helps users designing their innovation. The toolkit automatically propagates a new process design to the lower levels: It designs a matching workflow, initiates or changes relations with collaboration partners, and adjusts IT systems. The perfect toolkit would provide a common set of process definitions and allow the user easily to design new changes. It makes complex integrations between business processes transparent, helps to align company with process goals, and allows simulating changes to processes.

While not a perfect toolkit with, some reference models for business models come close. A reference allows practitioners to articulate their process needs easily in a structured way and understand constraints such as interrelations between processes. The table below explains analogies between a solution kit as described by von Hippel and a reference model for business processes, namely SCOR:

	User-Based Design (von Hippel)	Process Innovation in Supply Chain Management
Holder of Need Information	Product buyer	Owner of a supply chain or a process within
Holder of Solution Information	Manufacturer	Outside experts such as internal or external consultants. For system tasks, this could be a computer programmer.
Tool	Product-specific solution kit	Supply Chain Operational Reference
Iterative User and Manufacturer-Based Design Process	Manufacturer receives need information from buyer mostly by means of interviews. He develops a prototype, asks user for test, develops a new prototype, etc., until buyer approves.	Consultant analyzes As-Is, gets To-Be through interviews with process owners, develops prototypes until process owners approve. Prototypes could also include an ERP solution.
User-Based Design Process	Product buyer develops a solution using the solution kit. Manufacturer builds the product from this.	Owner of the supply chain or process owner analyze As-Is and develop To-Be processes using SCOR. Based on this To-Be, consultants can develop a workflow or IT solution without too much process owner interaction.

There are some differences between von Hippel's concept of a solution kit and SCOR: SCOR is not a toolkit but a reference. In a more narrow sense, its practitioners cannot develop their solution in a trial and error way as there is no simulation component in SCOR. SCOR offers other capabilities such as a system of metrics which makes it easy for practitioners first to analyze their As-Is.

Yet they have a lot of things in common: Both a solution kit and SCOR are for the use of practitioners from the business. They define a standard language that matches the domain and explains constraints. In SCOR, such constraints are, e.g., dependencies between processes. Owners of supply chains and processes find it easy to analyze their As-Is and develop innovative, well-aligned To-Be processes by themselves. SCOR makes interdependencies between processes transparent, which makes it easy for

practitioners to understand process integration and to develop innovative new business practices.

SCOR also provides a common language between teams of practitioners. In many instances organizations do not have a common language to describe their work. Same terms can mean different things to different parts of a large company. This causes confusion and inefficiencies.

As in the ASIC example, the output of SCOR is a user solution, not the final product. With SCOR practitioners create a business solution specific enough for workflow experts or ERP specialists to translate it into a workflow or ERP solution. Compare the results of SCOR to the results of ASIC (see above):

The practitioner can be reasonably sure that his specification of the new business solution matches his needs and how it does so.

The IT department knows that the specification is complete and consistent. They can create the final implementation without user iteration.

The list below describes a typical sequence in User-Based Process Innovation using SCOR:

Get the boss interested: Select overall supply chain targets from the performance attributes in SCOR. They reflect the kind of results senior management cares about.

Don't take a hammer to fix a screw: The practitioner identifies performance gaps. The process reference helps locate the root problem, tying it to a reference process, and understanding its nature: The problem may be in the material flow, in the planning process, an interface between two systems, etc. This avoids assigning blame exclusively to IT. For a process problem, Information Technology is like the hammer to the screw. Tying a problem to a reference process and understanding is critical for designing a good solution.

Create your innovation: The practitioner designs new practices for affected processes or redesigns the supply chain. The Supply Chain Reference helps him/her to safely navigate through his/her supply chain processes: It highlights options for alternative processes and best practices and ensures that no process is overlooked and that no process integration point is forgotten. The result is a business solution. It describes in simple graphics and natural language a new supply chain design, new practices, new material and information flows, etc. It refers to SCOR's best practices and explains how to apply them to a specific process. This is the last step for the practitioner.

Technical solution: The solution teams receive the business solution and translate it into a technical solution. This requires little interaction with the practitioner. BPM tools such as UML attempt to automate this step.

In the past few years, SCOR delivered results in supply chain improvement as dramatic as those that von Hippel quotes for User-Based Design. Similar process references for customer relationship management (MCOR) and product design (DCOR) are under way, and we expect their returns to be no less impressive.

Dr. Albrecht Ricken and **Ansgar Steinhorst** founded Navedon to help customers with their innovation processes in areas of business process design and SAP applications. They worked for 15 years as consultants for global companies in ERP and Supply Chain Projects. They can be reached via email at Contact@navedon.com

