Object-Oriented Customer Education

As customers require more trusted advice to solve their business problems, the choice of education solutions has become a strategic issue that often precedes and directs the choice of technologies.

by Wulf Rehder

Whether you buy a laptop computer or a lawn mower, you expect to learn how best to use it. For some products it is enough to skim the user's manual. For others you need to attend a class. In the past, product training was considered an attribute of product support. It came bundled with the product and was an expected feature like a power cord or the certificate of warranty. This situation has changed. When you buy a toy, batteries are no longer included. Similarly, education is no longer automatically included and free with the large, enterprise-wide solution purchases customers make today.

In such enterprise projects even laptop computers must be designed to work together with many other products that are often distributed in networks over large entities or even different countries. In these environments, training on how to use a single, standalone product is no longer sufficient. Customers now expect more comprehensive services, ranging from training in soft skills such as design methodologies and project management to proficiency in hands-on implementation and online troubleshooting.

The complexity of solutions, the size of customer projects, and the fact that computer systems are increasingly mission-critical for most businesses have led to the unbundling of product training and to the creation of entirely new product lines for professional consulting and education. Training has changed from being a product accessory to being a product itself. Customer education has grown from under the umbrella of product support to becoming a large and profitable industry by itself. In this paper, I will focus on the way HP's customer education, as part of the HP Professional Services Organization, is meeting the new challenges of developing and delivering to customers a cohesive suite of object-oriented education products.

Managing the Transition

It is a truism that every act of learning is a passage from knowing less to knowing more. However, customer education is more ambitious. This ambition shows itself in three ways. First, it is not enough to fashion data and information into a consistent, meaningful body of knowledge. While in a training class, customers must be led from "knowing what" to "knowing how" and being able to apply the new learning to their real-life problems. More knowledge must be transformed into more skills. However, there is a second, complementary aspect to learning: learning means not only acquiring more skills, but also acquiring different skills for new tasks in a changing environment.

The successful management of adapting to this change and the transition to higher levels of knowledge are the objective of customer education for all job roles as shown in Fig. 1.

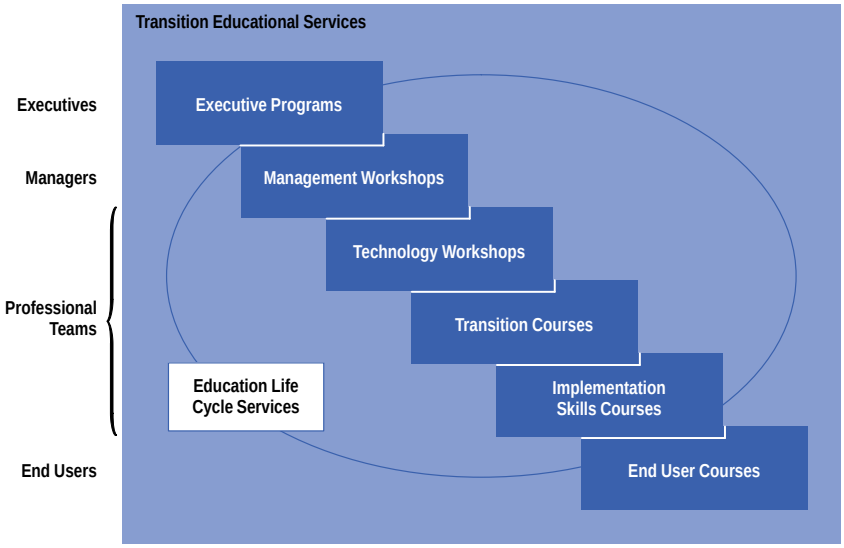

Fig. 1. *Enterprise-wide approach to managing transitions.*

Executives must be made aware of the risks and benefits the change to new technologies and processes entails for the entire company. With this awareness they will acquire the confidence, authority, and credibility to lead their business into previously uncharted terrain. Managers obtain the understanding and expertise to make the right technical decisions for their teams to be successful. Designers and developers master new professional crafts that help them apply the lessons learned for the creation of new products. Finally, end users realize the concrete benefits of the new technologies and processes.

The third defining component for contemporary customer education is its comprehensiveness. Fig. 2 specifies the three branches of a company's assets that need to transition together in a balanced way: its people, processes, and technology. All three are centered on serving common business objectives.

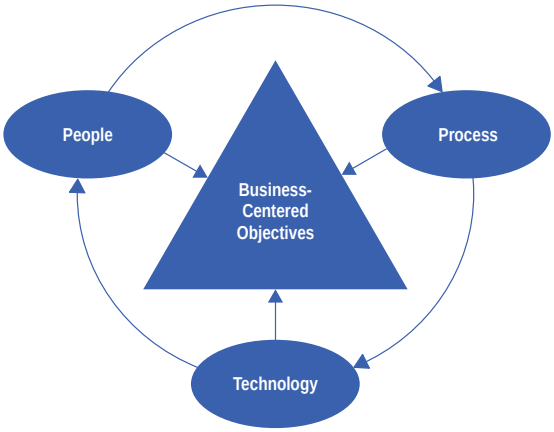

Fig. 2. *HP's people, process, and technology approach.*

This brief sketch has far-reaching practical consequences for the positioning, development, and delivery of customer education solutions. They do not merely add value to a product, but create their own suite of added values. Fig. 3 shows this value chain from the point where the actual interaction with the customer occurs (for instance, course content research and development phases are left out). As appropriate, some phases will be traversed repeatedly, depending on the results obtained so far and on the quality measures (e.g., completeness, level of detail) applied in the particular phase. Therefore, the links of the chain need to be interpreted as cycles. Under the name of education life cycle services, this simplified framework articulates the fact that customer education teaches how to manage change and how to evolve new skills.

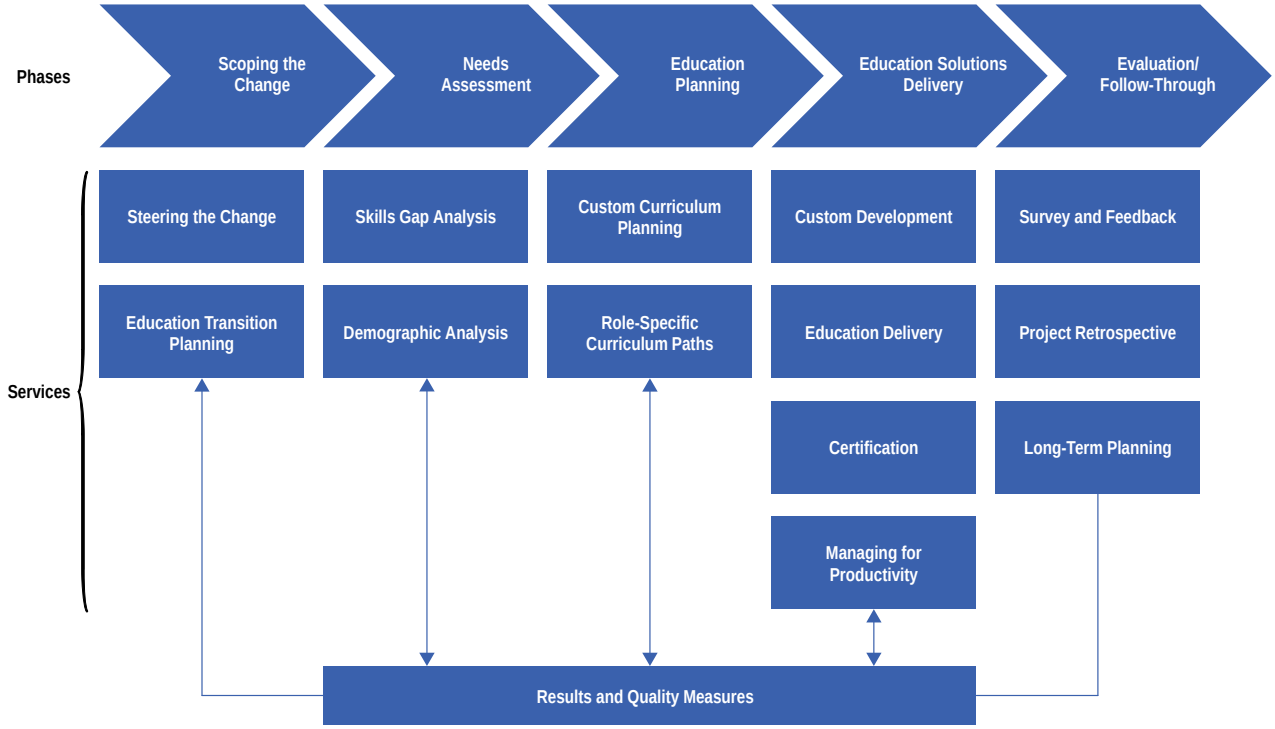

Fig. 3. *Education life cycle services.*

Customer education has become the industry of facilitating the transition from Tennyson's "blind and naked ignorance" to St. Thomas Aquinas' skill of man "to know what he ought to do."

Know Thyself

Before answers about the right path to object technology can be given, the right questions about the starting point, the path itself, and the goals have to be asked. To evaluate the starting point, HP's customer education services have developed a workshop called skills gap analysis.[1] Fig. 4 shows a step-by-step outline of this course. During the analysis, which is done jointly with the customer, the following documents are created to serve as the basis for the next transition steps:

- A written statement about business needs
- An inventory of current skills
- A list of additional skills to close the gap between current skills and identified needs
- Validation of findings and determination of action items.

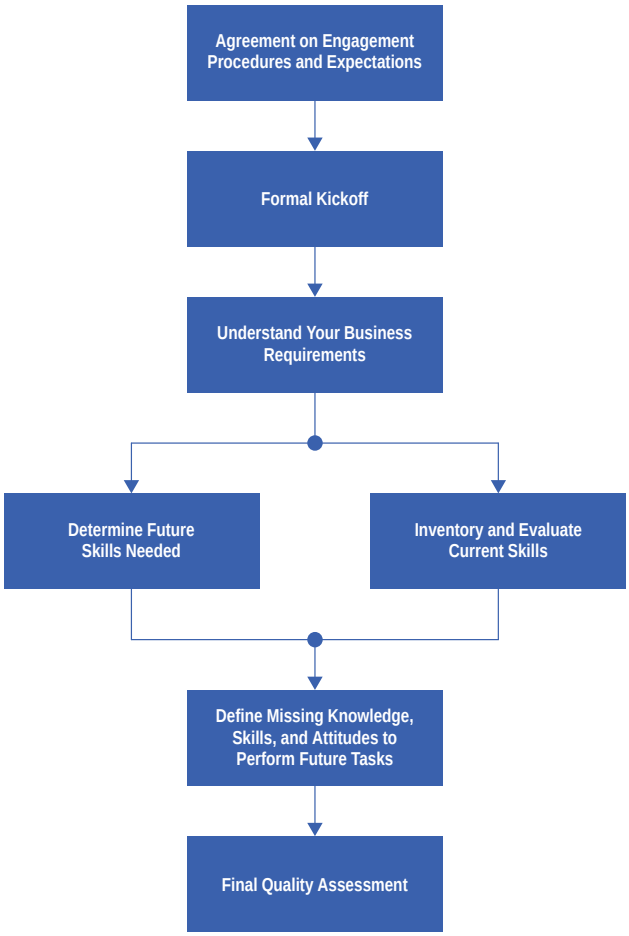

Fig. 4. *Skills gap analysis methodology.*

A skills gap analysis addresses a company's overall training needs and by itself does not result in a detailed training plan. To be more relevant to the discussion, we will focus on objects. The customized, object-specific version of a skills gap analysis is the object-oriented transition assessment workshop.[2] Similar to the skills gap analysis, the customer and at least two of HP's educational and technical consultants work through three sets of questions, assessing:

- The goals of a transition to objects
- The present skill level and object exposure
- The customer's current software development process.

A selection of some of these questions is enumerated on this page. In the transition assessment workshop, the skills gap analysis culminates in the preparation of a list of the ten biggest obstacles for a successful move to objects, jointly agreed upon by the customer and HP's consultants. These obstacles are different from company to company, but they typically fall into the categories of management commitment, organized barriers, fear of change, scarcity of resources, and loyalty to

legacy systems. Rarely are the inhibitors purely technical; the switch to new object-oriented tools and products is less problematic than overcoming the "soft" issues just mentioned.

This list of obstacles is the document upon which HP's team bases its recommendations for a concrete object adoption agenda, including job-specific curriculum paths. Such a detailed plan is the final outcome of the object-oriented transition assessment workshop.

After the workshop, with the enthusiasm usually quite high, many software development teams want to start their first object-oriented development project without delay. At this juncture, the HP consultant assumes the role of a mentor and monitors the speed, direction, and results of the transition that is now under way. See Subarticle 12b "***Starting an Object-Oriented Project,***" which summarizes a few caveats collected from many mentoring sessions.

Four Pillars of Soft Skills

A glance at the life cycle in Fig. 3 shows that the next phase is education planning. Based on the skills needs, curriculum paths are created that match specific jobs and roles designed to fill the needs. If, for example, system modeling skills are missing, the joint HP-customer team may define the new role of a system architect and recommend a series of courses to retrain designers to become architects. Once the roles are identified, the solutions will be designed and implemented. Experience has shown that this is not yet the place to select technologies (such as tools and implementation languages) or products (middleware, databases). Instead, the success of a transition to object technology appears, as our case studies with customers have shown, to be determined by the mastery of four soft skills: software architecture,[3] analysis and design methodology,[4] project management,[5] and systematic reuse.[6]

Software Architecture. Of the four skills mentioned above, architecting a software system is perhaps the most difficult, yet the most important and least well-understood skill. For the sake of brevity, three of the most important aspects of this difficulty are discussed here. First, there are at least four different views of a system architecture that emphasize different but overlapping concerns of high-level system design (see Fig. 5).

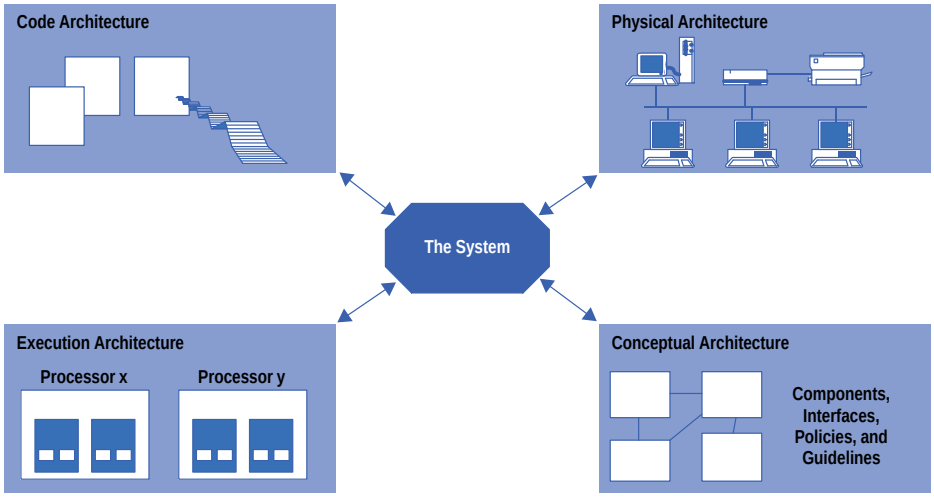

Fig. 5. *Four architectural views.*

Second, there is the choice of a viable reference architecture for an enterprise, which is a blueprint realization of an architecture that best fits a given business purpose.

One of the most successful frameworks for such a reference architecture is the so-called three-tier architecture (see Fig. 6). Once the tiers with their subsystems and interface specifications have been defined, it is possible to map products into the framework. For example:

- VisualBasic, Powerbuilder, or VisualWorks for the presentation layer
- C++ to build application programs whose components may be running on distributed servers
- A database or data warehouse like HP's Depot/J for the data management system
- Softbench for the development environment
- The Object Request Broker (ORB) software for the infrastructure logic that manages the communication among the distributed software components.

However, it is advisable to postpone these technology choices until after a thorough analysis and design methodology has been applied based on the particular customer requirements and the anticipated use cases of the planned system. See ***Article 10*** for a definition of use cases.

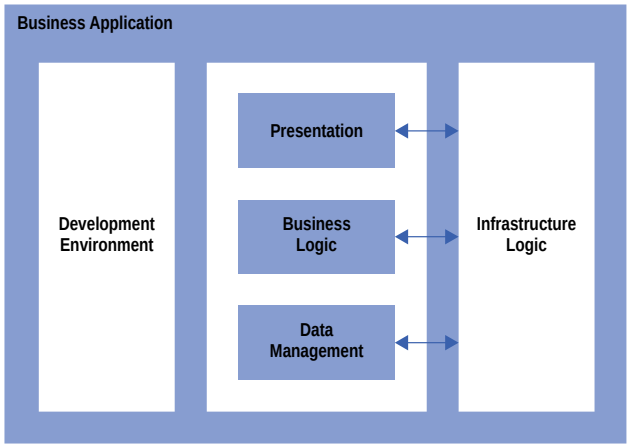

Fig. 6. *The three-tier architecture framework.*

The third difficulty is the lack of a generally accepted notation that is simple to apply and learn and yet rich enough to express the complex semantics of objects and their interactions in the different layers of a software system. HP and its partners are working together in committees chartered by the Object Management group (OMG) to formulate such a unified architectural language.

Analysis and Design. Better known and more mature than architectural models are the software analysis and design methods. They are often called methodologies, to distinguish them from the methods (i.e., the procedures or functions) owned by objects. A methodology defines a process that allows the division of work into distinct phases, each of which has well-defined exit criteria (e.g., finishing a graphic object model, drawing all dependency diagrams, and agreeing on design documents). The goal is to translate informal customer requirements into a more formal structure that then guides the implementation. Besides structured analysis and structured design other methodologies include the waterfall life cycle model of software development and HP Fusion.[7]

Project Management. Once the software architecture has been chosen (e.g., a three-tier reference model) and a methodology team has gone through the phases of system requirements, analysis, and design, a project team needs to be chosen to implement the design that realizes the architecture and solves the business problem. At this point of the transition, thinking about the peculiarities of object-oriented project management becomes important. Because of the inherent modularity of object-oriented design and the ensuing independence and autonomy of subteams, team building and communication may become an issue. New roles and responsibilities, such as framework architect, pattern designer, and class librarian need to be integrated. Since object-oriented design favors the implementer who postpones coding and (re)uses components as much as possible, performance evaluation and reward systems need to be reconsidered. This is opposed to the model of rewarding the implementer who "hacks" out the most code.

Reuse. The fourth of the recommended soft skills essential for a successful move to object-oriented software development is the incorporation and long-term management of systematic reuse. This course combines a discussion of

reuse technology (frameworks, patterns, software kits, components, and standards), and tools and processes with organizational and management issues. These latter nontechnical concerns often have the biggest impact on change management and the success of the transition to objects.[8] In the spirit of hands-on skill development, the second part of this course simulates the steps of systematically building reuse into a software organization. Fig. 7 shows the incremental steps from no reuse to systematic reuse through stages that mirror the phases of the Capability Maturity Model (CMM), which is widely used in the assessment of software skills.[9,10,11]

Projects versus High Volume

From the discussion above it should be obvious that the approach to customer education requirements for the transition to objects is not simply a matter of technology and product training. Just as an information technology department is much more than a random collection of computers and wires, so is today's customer education more than a collection of training courses. It has become an industry with finely tuned product lines that match the requirements of job groups by providing comprehensive training paths, from introductory courses to in-depth specialized skills.

However, in addition to these task-oriented, individualized curriculum paths, increasing emphasis is being put on integrated curricula for project teams, departments, and entire organizations. This latter trend has led to two distinct, but collaborating branches within the customer education business. One branch addresses the difficult, unique custom software project or the transition of, say, a COBOL programming team to Smalltalk proficiency. Efforts like these are resource-intensive, of high complexity, and more often than not also low-volume affairs. (They are the human learning system equivalents of highly sophisticated hardware and software systems, which usually need to be custom-made.)

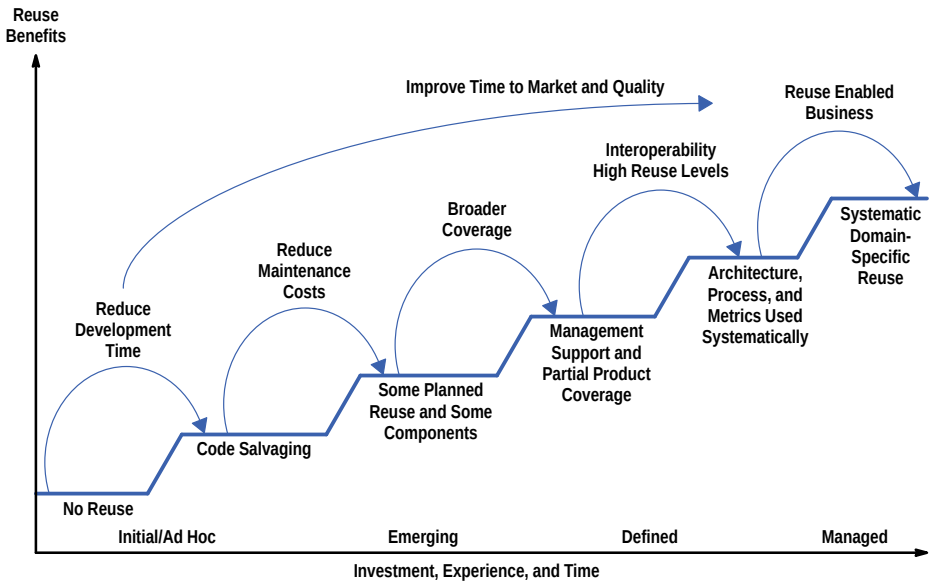

Fig. 7. *Incremental approach to reuse and the resulting benefits.*

For custom-made education solutions to be affordable, such highly complex offerings need to be created in a repeatable and modular manner. Examples of custom courses are the total immersion programs. In these programs, which are variously known in the industry as residency programs or boot camps, entire teams are led through a four-to six-week customized curriculum to object-oriented literacy.

The other, complementary branch of customer education addresses the high-volume, lower-complexity demands. These are requests for standard programming language courses, fundamentals of operating systems, system administration, networking, and relational databases—all of which figure prominently in most two-or-three-tier business application developments.

These conditions of serving widely diverging interests are posing challenges for the development, sales, and delivery of education solutions in general and object-oriented education in particular. The challenges are similar to the ones known in traditional product development:

- Primary and secondary research explore the market conditions
- Investigations define product possibilities
- Curriculum creation involves outsourcing, partnerships, and collaborations with product divisions and the field
- After going through the typical lab cycles, prereleased material is validated in alpha and beta tests.

In parallel, marketing collateral is being prepared, data sheets, sales briefs, and advertising copy are written, catalogs appear worldwide, and indirect and direct sales are made. To be successful, a well-managed and diverse team of course designers, business developers, solution architects, education advisors, technology specialists, consultants, and instructors needs to be trained and deployed worldwide. Issues of localization, government regulations, copyright protection, postrelease support, updates, and pricing (for instance, discounts, volume buying, specials) are again not different from the rollout of major hardware and software products.

In light of these considerable complexities, training vendors may be tempted to define their solutions by offering a variety of topics for which they have in-house technology expertise and then to reshape the customer needs along the lines of these topics. The true challenge consists, however, in basing education solutions on the transition assessment workshops and education plans that have been crafted and agreed upon jointly by the customer and education consultants. Only such solutions have the strategic impact of preceding and guiding the choice of implementation technologies.

Point Solutions and Product Training

Supporting the larger picture of education solutions outlined above are several training offerings that are more specialized, narrower in scope, or tool and technology related. Here, training usually tracks the release, purchase, and installation of products. As a consequence, training courses have to be updated in a rhythm following the product updates. This especially includes languages converging towards standards, such as ANSI C++, different implementations of new languages, such as Java, and products that bridge evolving de facto standards, such as those for distributed computing. Examples of the latter are the Object Request Broker (ORB) implementations which adhere to the Common Object Request Broker Architecture (CORBA) standards and serve as interoperability middleware between CORBA objects and the emerging Microsoft® OLE automation product suite. Such software has to be supported by several operating systems and communication protocols. In the case of HP's ORB Plus 2.0 these are the HP-UX*, SunSoft Solaris, and Microsoft NT platforms and the IIOP (Internet

Inter-ORB Protocol, platform independent) and DCE CIOP (Common Inter-ORB Protocol, HP-UX only) standards. Using the IIOP protocol, ORB Plus 2.0 will interoperate, for instance, with Distributed Smalltalk software from ParcPlace-Digitalk.

From these typical examples it becomes obvious that narrow, specialized point solutions and product training can be as labor-intensive as the solutions centered around the care for people and processes. Since the competitive pressure for training on shrink-wrapped products is fierce (you can learn C++ in community colleges almost free), larger education providers have surrounded themselves with satellites of smaller, agile partners, who can, in the analogy used before, be compared to suppliers of hardware and software parts.

Challenges and New Directions

One of the most exciting events in the emergence of object technology is the recent promise of its convergence with internet technology. To begin with, Java is a C++-like object-oriented language that allows the objects (for instance, the ones created in its applets) to be shared over the net in a platform independent way. Java has spawned several new customer education offerings, including ones on web security and on how to use the web for commercial transactions.

Furthermore, with the web becoming more familiar as a medium for information exchange, it is also fast becoming a candidate for alternative training delivery, complementing computer-based training (CBT), CD ROMs, and the traditional lecture and lab format. Such a departure from copyrighted class material to an essentially open, public forum creates new challenges, but these challenges are no more severe than the ones faced by software distribution and publishing on the net. This is especially true in the high-volume, point-solution, and product-training market where the material is rapidly becoming part of a commodity business with small differentiating value and practically no proprietary lock on content. Instead, as Tim O'Reilly[12] suggests (and practices for his own on-line business of computer books), more important than copyright is the development of a brand identity that represents a consistent, trusted selection of high quality. This is where high-volume customer education may be headed in the future.

Acknowledgments

My view of customer education as an autonomous business has evolved in discussions with Patricia Gill-Thielen, Brian McDowell, Tom Ormseth, Morris Wallack, and Ann Wittbrodt. While they are not to blame for my opinions, I hope that they'll accept my thanks for mentoring me when I joined their team.

References

1. *Skills Gap Analysis Workshop*, Product Number H6230A.
2. *Object-Oriented Transition Assessment Workshop*, Product Number H6290X+002.
3. *Software Architecture Workshop*, Product Number H6290X+009.
4. *Analysis and Design Methodology Workshop*, Product Number H5851S.
5. *Project Management Workshop*, Product Number H6516S.
6. *Systematic Reuse Workshop*, Product Number H6514S.
7. T. Cotton, "Evolutionary Fusion: A Customer-Oriented Incremental Life Cycle for Fusion," *Hewlett-Packard Journal*, Vol. 47, no. 4, August 1996, pp. 25-38.
8. M. Griss, I. Jacobson, and P. Jonsson, *Software Reuse: Architecture, Process, and Organization for Business Success*, Addison-Wesley, January 1997.
9. W.S. Humphrey, *Managing the Software Process*, Addison-Wesley, 1989.
10. M.C. Paulk, B. Curtis, M.B. Chrissis, and C.V. Weber, *Capability Maturity Model for Software, Version 1.1*, Software Engineering Institute, CMU/SEI-93-TR-24, February 1993.
11. D. Lowe and G. Cox, "Implementing the Capability Maturity Model for Software Development," *Hewlett-Packard Journal*, Vol. 47, no. 4, August 1996, pp. 6-14.
12. T. O'Reilley, "Publishing Models for Internet Commerce," *Communications of the ACM*, Vol. 39, no. 6, June 1996, pp. 79-86.

▶ Go to Subarticle 12a

▶ Go to Subarticle 12b

▶ Go to Journal Home Page