
R E C O M M E N D A T I O N S

Date: May 15, 2004
To: Contact Name
Client Company, Inc.
From: Lisbeth Shaw
Cambridge Publications, Inc.
Subject: Documentation Architecture and Strategy Recommendations

EXECUTIVE SUMMARY

This report includes the findings from the consulting engagement undertaken by Cambridge Publications, Inc. (CPI), to define a documentation architecture, or suite of documentation components, that best supports Client's business.

The following summarizes the anticipated benefits of this document architecture:

- This architecture provides for documentation that should decrease the amount of time Client's staff and VARs spend supporting customers' migrations to [development platform name].
- The documentation specified here should increase customer satisfaction.
- The architecture provides a comprehensive and integrated documentation network, so that every piece fulfills a specific requirement, all requirements are filled, and there is no redundancy.

PROJECT BACKGROUND

Client develops client-server and Internet-based database management software for developing, deploying, and maintaining business applications solutions. [development platform name] an application development and database-independent management system that enhances and extends the UNIX, Windows 2000, and Windows XP Operating Systems, provides a natural interface from Visual Basic and other Windows development systems to postrelational/NF2 databases and their associated structures. Client also facilitates migration from other DBMS's, such as Pick and Reality.

There is currently little documentation supporting [development platform name] and most of that is not current. Without helpful documentation, migrating to [development platform name] requires significant time from client's support team, which is at this point overextended. Client wanted to define an effective documentation-based solution.

FINDINGS

As part of this project, CPI worked with Client's management and support team to define the company's needs and goals for documentation. CPI visited the company headquarters and held phone discussions with these people:

- CEO
- Director of marketing
- Customer support manager (Europe)
- Customer support manager (US)
- VAR

In particular, the customer support managers expressed the following as their most pressing needs:

- Migration support
- Troubleshooting support

THE CURRENT SITUATION

Most of client's direct customers are moving from another, often older, development environment. These customers, who span a wide range of technical expertise, require support as they migrate from the legacy system to client. In many cases, the legacy system resides on a mainframe, meaning that these customers undergo a cultural dislocation; they must move from the green screen to the blue screen, from command line interface to GUI. They must also assume some new responsibilities (setting up of user accounts, for example) at the same time as the new environment simplifies other tasks. These customers rely on client customer support to help them in the migration process, which can be very time-consuming.

Client also works closely with VARs, who must develop their own expertise in client's products. VARs develop turnkey applications for their own customers and must be able to support these applications when the need arises. This means that VARs must be able to find answers to their specific feature- and capability-related questions, answers which reside in the enormous repository of information available online. This information is currently not organized or presented in an intuitive way.

Finally, the current documentation is significantly out of date.

THE PRIMARY NEED

Based on conversations with the customer support managers, the most pressing documentation need is to support migration, which would increase customer satisfaction and reduce staff support call time. There is also an urgent need for troubleshooting support materials.

THE SECONDARY NEED

The next greatest need is for documentation of new products and capabilities, which are often sophisticated and are currently undocumented.

THE TERTIARY NEED

There is also a need to improve the accessibility and usability of existing information. Experienced users can find what they need in the client KnowledgeBase; but it's not easy to do, and once done, the information is not easy to use. However, the information exists, and can be found and used, so this need is less acute than the need for migration, troubleshooting, and new product documentation.

RECOMMENDATION

CPI recommends an underlying structure to the documentation architecture that comprises seven components. These components are identified in the following table, listed in descending order of priority. The table provides an overview of each component. A more detailed description follows the table.

The first three components comprise a sequence: The overview is the most general, followed in the general/specific continuum by the migration methodology, followed by the platform-specific migration and setup procedures guides. Each of these components is able to stand on its own. However, the components are designed to dovetail one with the next. They are much more powerful within this sequence than they are as individual entities.

Component	Description and Notes	Maps to Client-Defined Need
Overview: Client and Migration	A piece that sets forth the [development platform name] architecture, prerequisites for migrating to [development platform name] and the implications of the decisions made as part of the migration process.	It's very important that implementers understand the [development platform name] architecture, as they must make decisions that will affect what they can and can't do, with what degree of difficulty, as they extend their use of Client.
Migration Methodology	This is a step-by-step approach to the migration process. It includes a generic trouble-shooting methodology, which contains techniques for diagnosing and correcting problems.	A migration methodology provides the best tools for managing and supporting the migration process. It also provides a replicable procedure. Client has defined migration as a highly critical area, one which needs a strong support tool.
Platform-specific Migration and Setup Procedures	For each platform, a manual that covers specific migration procedures and builds on the generic methodology. These prescriptive, platform-specific guides lead implementers through the critical setup procedures. Each guide includes specific troubleshooting steps, which spell out platform-specific techniques for diagnosing and correcting problems.	<p>These guides focus on the porting and setup issues facing an implementer Client has identified setup issues as critical.</p> <p>In particular, the transfer of legacy data via tapes is a recurrent and critical issue. There are many potential pitfalls here, and it can take hours to walk an implementer through this process. One objective for this guide is to reduce the tech support time required for the completion of this procedure.</p>
Documentation for New or Auxiliary Products	Example: Transaction Journalling, which many customers are using.	<p>Customers want this.</p> <p>Note: The resources for writing this manual already exist and it's a bounded topic, so we should be able to complete it quickly.</p>

Component	Description and Notes	Maps to Client-Defined Need
User's Guide	A platform-independent, workflow-based guide that covers all of the core macro-level tasks.	This is a user- and procedure-oriented piece. Given that most issues relate to migration, a user's guide is not really necessary now. However, if the migration process were to draw on a less technical user population, such a piece would become more important.
Reference Guide	Documents every feature; covers everything.	This is more immediately relevant than a user's guide, as it would provide a usable encyclopedic resource. Users have said they want to be able to find information easily, and a reference guide would help them do it.
Extending Client	Modular pieces on using [development platform name] with new technology, or to solve problems not previously within the Client purview. Scenarios and plenty of examples.	Both customer support managers, and VAR all cited a need for more readily accessible, easier-to-use documentation of the core system. In particular, VAR identified the need for easy-to-understand instructions that would make it more straightforward to use [development platform name] in different ways.

OVERVIEW: CLIENT AND MIGRATION

This document addresses both immediate and longer term needs. In the short term, an overview of [development platform name] provides an important orientation for the implementer, who is moving from one environment to a very different one. Longer term, this piece will provide the tools for the new customer to understand the [development platform name] architecture, which in turn will enable him or her to understand the implications of choices he or she must make as part of the migration configuration.

MIGRATION METHODOLOGY

This is one of the most important components in the proposed documentation set. Migration from one environment to another is a major event. A well thought out and systematically followed methodology, which breaks the process into distinct phases and steps where the following are identified:

- The types of issues that need to be considered
- The types of decisions that need to be made and activities that need to be performed

- The module or area in [development platform name] to which this phase of the methodology applies
- For each phase, typical problems and their solutions

The methodology can be broken into finer steps, mapping out a workflow. These defined sequences of small, dependency-driven steps help make the migration manageable.

PLATFORM-SPECIFIC MIGRATION AND SET-UP PROCEDURES

These guides build upon the generic methodology and offer specific instructions related to the different departure platforms. The directions here are prescriptive, rather than descriptive, and are tied to particular legacy platforms. These guides lead the user (the system administrator) through the setup procedures. Each guide includes a platform-specific troubleshooting section.

EXTENDING [DEVELOPMENT PLATFORM NAME]

These guides are modular, rather than comprehensive, and focus on how to diagnose problems, providing problem/solution scenarios with plenty of examples. In different language and with differing emphases, each of the people we spoke with noted the importance of easy to find, easy to read, easy to use information on the core [development platform name] functionality. It's also important to document new and creative uses of [development platform name] and new connections between [development platform name] and other products, especially new technology.

NEW PRODUCT DOCUMENTATION

Everyone we spoke with noted the importance of documenting new products, or [development platform name]'s ability to interface with new software. Specifically, transaction journaling, which is properly speaking not part of the core functionality, is widely used and represents "the new [development platform name]". For reasons of both utility and marketing, it would be beneficial to provide documentation for this and other modules like it.

RELATIVE LEVEL OF EFFORT FOR DEVELOPING THE COMPONENTS

These estimates are ballpark figures, designed to give Client a rough preliminary idea of how difficult one piece is in relation to another.

Component	Relative Magnitude/Difficulty
Platform-specific Migration and Setup Procedures	1
Documentation for New or Auxiliary Products	2.5
Extending Client	4
Reference Guide	4
User's Guide	4.5

Overview: Client and Migration	5
Migration Methodology	10

ADDITIONAL RECOMMENDATIONS

Here are some questions that arose in the course of our conversations, and recommendations in response to these questions:

Question	Recommendation
In what order should these documentation components be done?	We recommend that the first three, which really form a continuum, be written in the order in which they are listed. Other components (doc for new or auxiliary products, user's guides, reference guides, and guides on extending [development platform name]) should be written on an as-needed basis. The piece on transaction journalling is an example of this.
How should the documentation be updated?	<p>Although more than one person said that everyone should be able to update the documentation whenever a change to the software occurs, we recommend that there be standards and procedures for doing this. If there is no control over what, when, and how information gets added to the documentation, problems with documentation maintenance will multiply. Client will not be able to rely on the documentation with a high level of confidence.</p> <p>The Knowledge Base uses this model, and is an example of the benefits and the drawbacks of unmediated user input to documentation.</p>
Should the documentation be prepared for electronic or paper delivery?	Electronic availability would be helpful, but some users (for example, on the factory floor) do not have online access. Some users also want or need to write on their documentation. We recommend a range of styles and delivery media.
What are the first steps for implementing this architecture?	We recommend that Client begin with a detailed documentation plan for the first three books, and that the initial documentation effort include a book design. This will ensure that there is a consistent look and feel as the documentation set develops over time.