

The CM Database: To Buy or to Build?

Reed Sorensen

Computerized Thermal Imaging Inc.

For some organizations, it makes sense to develop a database system that supports reporting, auditing, problem tracking, and certain project management (PM) functions rather than buy a commercial configuration management (CM) product that additionally provides capabilities such as version control, build, merge, and conflict identification. This article provides examples of organizations that have developed a CM database and discusses the relationship between the PM and CM disciplines.

Project managers and program managers need ready information about the status of the system they are maintaining/developing to answer the questions:

- What are the pieces of the system?
- How do the pieces relate?
- Which pieces are being purchased?
- Which pieces are under maintenance contract?
- What am I spending to maintain a component?
- How many change requests are outstanding and against which components?
- What is the status of a change?
- What components present the greatest business, schedule, and technical risks?

These are a sample of many questions that can be asked.

Project Management and Configuration Management

A search for answers may bring a person to more than one management discipline, namely the project management discipline and the configuration management discipline. EIA Standard IS-649 [1]¹ notes the relationship between these disciplines, and Ovum observes that the relationship will be reflected in the tools that support PM and CM [2].² While tools can be bought to help answer the questions, the challenge is in finding one that most closely meets your functional, budgetary, and cultural requirements.

This article assumes you have determined that most of your questions are of a CM status reporting nature. Tables 1 and 2 list the kind of information typically attributed to PM and CM. The tables also enhance understanding of the relationship between the PM and CM disciplines if you modify the PM information table by substituting the words "problem/change" for "project." Often a problem/change becomes a project if it is big enough.

actual progress with respect to planned progress
individual resource profiles (skills, labor rate)
work breakdown structure codes and associated fields
dependencies for project tasks
project calendar
schedule impacts due to increased workload
flexible cost reporting (e.g. by account, department)
critical path
Gantt chart view of project schedule

Table 1. Representative PM Information

list of configured items
map changes/problems to engineer assigned
status of problem /changes (open, deferred, closed)
report progress on problem /changes
version information
differences between versions
file revision history

Table 2. Representative CM Information

Buy vs. Build

A manager with access to in-house software development expertise may, at some point, consider the expense and effort of evaluating, procuring, customizing, and maintaining a commercial-off-the-shelf product (COTS) vs. the expense and effort of building and maintaining a tool. While COTS tools usually include some kind of database, they are not specifically a database. For the manager who is focused specifically on the kind of functionality that a database system, including a graphical user interface and database engine, can provide, the COTS CM tools are not necessarily the most attractive option. Often an in-house developed database is a better match functionally. Some may be more easily procured, if the in-house development resources are available.

Aside from functionality and economics, cultural issues may sway a manager to build rather than buy the tool. The argument, "We are a development organization; why pay outsiders to do what we can do?" carries some weight. Your people may simply want to build their own tool vs. learn a COTS product, so there is a morale issue or bias.

The vendor of a COTS CM product might point out that the vendor's product can be customized to any process, that it will be more mature than a home-grown tool (better documentation, ready training, fewer defects) and that maintenance costs will be lower. There is the potential additional advantage of having all the data and CM functionality in a single system as noted by Ovum's Clive Burrows:

Gathering management information is greatly simplified if change features are part of the CM system—without them, complex cross-references between different databases are required, and full navigation and searching may not be possible.

Unfortunately, many CM vendors have developed their own add-on capability in this area using new development tools, different databases, and even a different style of user interface. In some cases, the only area of commonality is the product "badge" created by the marketing department [3].

In practice, the buy vs. build decision is not an either/or solution. This author sees a lot of organizations that use COTS products for some CM requirements and "roll their own" CM database to handle status accounting and project management requirements. Here are three such organizations.

Three Air Force Examples

1. In the 1980s, the Automatic Test Equipment projects in TIS at Hill AFB had developed an in-house database in dBase IV, a nonrelational database tool available from Inprise Corporation (formerly Borland). In 1997, TIS used its own software developers to migrate the database to a Microsoft Access implementation to control the status accounting requirements for configuration management.

When TIS was contracted to

configuration manage the GTACS³ software, it followed the Program Office mandate to use and implement CCC Harvest. The GTACS software development community required a robust code management tool. After a year of working with Harvest, TIS chose to keep its Microsoft Access database for another year and then migrate it to Oracle and also retain CCC Harvest. The bottom line was that it needed Harvest for the code management and it liked its in-house developed database for information management.

2. CIDSS at Peterson AFB maintains the software for the Space Environmental Support System. It chose Microsoft Access to implement Project Logging and Tracking Tool (PLATT), a database that tracks requirements submitted by the customer as well as internally generated software maintenance tasks. While not experienced with Microsoft Access as a development tool, one CIDSS person with strong software development background and the ability to learn implemented the database. Version control of software code at CIDSS is implemented using Configuration Management System by Digital Electronics Corp.
3. TISHB at Hill AFB develops operational flight software for the F-16 multirole fighter. For the past seven years it has used a database that TISHB developed to track changes to the software and associated test stands. The approach has been successful for at least two reasons:
 - 1) the Sybase Server used to implement the database is used for several other applications and was already available when the buy or build decision was made.
 - 2) the responsiveness of in-house Sybase expertise means changing requirements are implemented readily. As a capability becomes obsolete, it is deleted from the system. Version control, file merging, and build management are implemented at TISHB using Concurrent Versions System freeware with a user interface that

TISHB developed.

There is a common thread in these three scenarios. All of the above organizations are using some COTS product for CM, but also chose to implement a database on their own.

Recommendations

Define your process and requirements.

Whether you buy or build, half the battle is defining your requirements [4]. Requirements drive your database design and implementation, or your evaluation and customization of a COTS product. Documenting your current approach to CM is part of the definition task. You may also have in mind a desired future process that differs from the current process; if so, also document the future process.

Walk, do not run.

As a manager, can I live with my current mode of operation, and if so, for how long? Boeing's approach to developing a database over several years was successful for them [5]. You will need to trade off the time required to implement a database vs. the cost and quality issues to implement that database. Faster implementation means more initial expense (purchase or development) or less quality (less functionality and/or more bugs).

Use as little automation as practical.

Some level of automation will be appropriate in managing the data that answers the questions posed in this article's introductory paragraph, but you will want to use as little as practical. This will maximize your answers and minimize the cost of maintaining the automation. Do not maintain a high-end tool when a spreadsheet will provide the answers.

Conclusion

As a manager, do I have internal expertise that is available for implementing a solu-

tion based on my well-defined requirements? Do I have the needed database development tools available? If so, the build option is attractive. Table 3 is a representative list of database development tools. ♦

About the Author

Reed Sorensen serves in configuration management, quality assurance, and technical publication roles at CTI (Computerized Thermal Imaging Inc.), which deploys thermal imaging and associated technologies for use in medical screening, diagnosis, and patient management. Sorensen has more than 20 years experience:



- 1) developing and maintaining software and documentation, and
- 2) improving configuration management and documentation processes.

He has published articles in *CROSS-TALK* on various software related subjects.

Computerized Thermal Imaging Inc.
 476 Heritage Park Blvd., Suite 210
 Layton, Utah 84041
 Voice: 801-776-4700
 Fax: 801-776-6440
 E-mail: reeds@siinet
 Internet: http://www.cti-net.com

References

1. *EIA/IS 649 National Consensus Standard for Configuration Management*, August 1998, page 1
2. *Ovum Evaluates: Configuration Management*, 1999, Page 12.
3. Burrows, Clive. "Configuration Management: Coming of Age in the Year 2000." *CrossTalk*, March 1999.
4. See Susan Grosjean's *Building a Configuration Management Database: Nine Years at Boeing*, in this issue.
5. IBID.

Product	Vendor	http://
DB2	IBM Corporation	www.software.ibm.com/data/db2
Inform ix	Inform ix Software Inc.	www.inform-ix.com
dBase	Inprise Corporation	www.inprise.com
Access	Microsoft Corporation	www.microsoft.com/office/access
SQL Server	Microsoft Corporation	www.microsoft.com/sql/?RLD=183
VisualFoxPro	Microsoft Corporation	msdn.microsoft.com/vfoxpro/
Oracle	Oracle Corporation	www.oracle.com
Sybase	Sybase, Inc.	www.sybase.com

Table 3. Some database development tools and vendors that may or may not be useful for your organization⁴

Notes

1. Configuration management principles underlie sound business practices used throughout industry and government to provide: . . . Access to accurate information essential to the product's development, fabrication, production, use, maintenance, procurement, and eventual disposal.
2. ...in future the scope of what is considered to be CM will undoubtedly include strong links with project management systems.
3. *Ground-based C2 elements of the Theater Air Control System* supporting air operations performed by the Combat Air Forces.
4. This is a representative list, not an exhaustive one. The list is for information only; no endorsement of these products or vendors is implied.

Call for Articles

If your experience or research has produced information that could be useful to others, *CROSS TALK* will get the word out. We welcome articles on all software-related topics, but are especially interested in several high-interest areas. Drawing from reader survey data, we will highlight your most requested article topics as themes for future issues. In future issues of *CROSS TALK*, we will place a special, yet nonexclusive, focus on:

CMMI

June 2000

Submission deadline: Feb. 1, 2000

Personal Software Process and Team Software Process

July 2000

Submission deadline: March 1, 2000

Object-Oriented Technology

August 2000

Submission deadline: April 3, 2000

We will accept article submissions on all software-related topics at any time; issues will not focus exclusively on the featured theme.

Please follow the *Guidelines for CROSS TALK Authors*, available on the Internet at <http://www.stsc.hill.af.mil>.

Ogden ALC/TISE

ATTN: Heather Winward

7278 Fourth Street

Hill AFB, Utah 84056-5205

You may e-mail articles to features@stsc1.hill.af.mil.
or call 801-775-5555 DSN 775-5555.

Have you changed your suit?

New job? Different e-mail address? Updated contact information?

In an effort to keep *CROSS TALK*'s mailing database up-to-date we would like to know what is new with you.



Fill out our easy-to-use form at <http://www.stsc.hill.af.mil/Crosstalk/crosstalk.html> (click 'subscription form')

or respond in writing at:

fax: 801-777-8069

e-mail: stsc.custserv@hill.af.mil

mail: CrossTalk

O O-ALC/TISE

7278 Fourth Street

Hill AFB, Utah 84056-5205

Your information will be promptly updated and you will continue to receive all the latest software engineering information from *CROSS TALK* The Journal of Defense Software Engineering.

First-time readers, please fill out the free subscription form located between pages 16 and 17

or visit our web site at <http://www.stsc.hill.af.mil/Crosstalk/crosstalk.html>