

AN INTERNATIONAL FOCUS ON CASE METHODOLOGY

by Malcolm Lane, Ph.D.

Introduction

Barents Group, LLC is a KPMG company that emerged from the Policy Economics Group of KPMG Peat Marwick in Washington, DC, and provides professional services in emerging markets around the world. Areas of specialization including:

1. fiscal reform (tax policy, tax administration, budget, and accounting)
2. financial institutions, and
3. privatization,

require computerization for a variety of applications. Barents' International Computer Practice focuses on the development and support of applications for use by foreign governments at the federal, state and local level. Many of our project countries are in their infancy of computerization. In such environments, there is a unique opportunity to introduce well-developed systems analysis, software design methodologies and tools, and teach government counterparts how to properly use both.

Traditional methodologies simply do not work in countries facing so much change and having little experience in the use of information technology. It is difficult enough to develop a feeling of ownership on the part of the client in any environment, but even more difficult in the environments in which we work. Users must be involved in the computerization process from day one.

Typical Client Environment

It is our experience that the same scenario repeats itself over and over in almost every country. In general, the following characteristics are typical to our project environment:

- There is little or no computerization in place
- applications that are automated are implemented on either obsolete mainframe or Minicomputer equipment running vendor proprietary software

- Microcomputers are just beginning to find their way into government departments, but are not typically stand alone systems
- Technical staff are paid far below the wages of the private sector, and retaining competent staff is very difficult
- English language skills are often weak and documentation is entirely in English, making it difficult for technical staff to easily learn about new products;
- The government officials consider the purchase of hardware to be the only step necessary to computerized applications; there is very little thought given to the applications themselves
- Schedules set for computerization are unrealistic and cannot be met, thus causing all computerization efforts to lose credibility
- There are no standards or methodologies in place in computerization projects, most attempts are, more or less, brute force efforts, and
- For the most part, there are very few software development tools available to government staff, primarily because of a lack of funds

The environment is one that is ready for the introduction of some methodology for requirements analysis, and software design and implementation. Simply teaching new techniques and methodologies will not make them become a reality. The introduction of a CASE tool helps to enforce the discipline of a methodology, while making it easier for staff to follow the methodology.

Requirements for CASE Tools

Knowing the difficulties that would be encountered in the environments in which we are working, there were certain characteristics that were required for any CASE tool to be considered.

- **Process and Data Modeling Capabilities** – The CASE tool had to provide the capabilities for defining both the process model and entity model, and have full repository support. Either Yourdon or Gane & Sarson rules had to be supported in the process model. The entity model had to provide for normalization rules.
- **PC-Based** – Virtually all client countries had some experience with the DOS environment on a “PC”, and such computers were already available in country. The selection of a PC-based tool had the advantages of being able to run on the existing client system and allowed us to purchase inexpensive Intel-based systems to support the tool when such systems were not already available. The most important reason for the PC-based requirement was the capability of being able to run the tool on a laptop computer that could be used in a

teaching and demonstration environment using LCD display panels and an overhead projector. As will be discussed later, our methodology requires constant client interaction, and the tool could then be taken to the client on demand for workshops and periodic reviews.

- **Compatibility with the Development Environment** – Consideration was given for the use of an integrated CASE tool, i.e., one that virtually is part of the development environment which, for the most part, was a RDBMS fourth generation language. The difficulty with this is that we cannot dictate the development environment to our clients. If by some chance, an RDBMS had already been adopted as a standard, we could not justify introducing yet another for the technical staff to support. Even when we could recommend a particular RDBMS, we had to take into account the skills within the country as well as local support for the RDBMS itself.
- **SQL Schema Support** – Even though an integrated CASE tool was not to be considered, there was still a need to support the generation of SQL schema for “mainstream” database systems.
- **Reverse Engineering** – In many environments, the capability of being able to reverse engineer at least the data model from the database schema would be extremely valuable. The reason for this was that most systems already implemented by the client had little or no documentation. The capability to bring the data model into the repository would greatly assist in documenting and understanding existing systems and would introduce the clients’ technical staff to the CASE tool using an application with which they were familiar.
- **Foreign Language Support** – Because of the many different countries in which we work, the CASE tool should provide support for foreign languages (e.g., Russian, Polish, etc.) Such support could be provided by the CASE tool supplier itself, or by third party products.
- **Flexibility** – The tool selected should allow us to make enhancements to meet any specialized need that might evolve in a given client environment. A minimum capability was to have a clear understanding of the repository format. Another consideration was flexibility in the repository itself, i.e., could information be added to the repository outside of the rules or fields established by the tool itself?
- **Dynamic Product** – Any tool selected had to be one that would improve over time. This meant that regular releases had to be part of the selected vendor’s plan. This could be best verified by the past history of product releases.
- **Product Security** – Because of the working environment in many countries, we felt strongly that the product should be protected against

piracy. A product with hardware protection seemed to be the most appropriate security for such environments.

- **Multi-user Support** – As we looked to the future in each of our project countries, we recognized that while stand-alone DOS PC platform was required for the CASE tool, it also should be able to provide multi-user support on a network (Novell) platform. This would allow for CASE project sharing and reduce the per user cost of the product and its maintenance.
- **Cost** – The cost of the CASE tool had to be reasonable if governments were to be able to purchase and maintain the product. This was another reason that a stand-alone tool rather than an integrated tool was considered.
- **Ease of Use** – No matter how many of the above criteria a CASE tool satisfied, it would be of little value in our international projects if it was not easy to use. Hence, ease of use was a primary consideration in the selection of a CASE tool.
- **Complement to Rapid Prototyping Methodology** – Experience has proven to us that the use of rapid structured prototyping that allow users to “test drive” the proposed computerized system prior to full implementation is the best way to guarantee user involvement and to product user-defined systems that truly meet the users’ requirements. Our methodology is centered around rapid structured prototyping and constant user involvement in the requirements definition stage of the project. What we required was a CASE tool to complement the prototyping activity requirements definition.

Visible Analyst Initial Experience – Visible Analyst (VA) was initially chosen because it met the majority of the criteria discussed previously.

VA became an integral part of our methodology in a new project in Central America. The application development environment in this country was FoxPro in a Novell network environment. VA was used during all stages of the software development lifecycle, from prototyping / requirements definition to documentation.

Even in the first year of use, VA proved to be the required complement of the rapid prototyping-based methodology that we use in our projects. The combination of the prototype, the VA representation of requirements and design, along with the accompanying paper documents in an excellent suite of products that define the application being developed accurately and clearly to the client.

Furthermore, Visible Analyst has proven to be a reliable and robust product. It has evolved to meet our needs over time.

The use of Visible Analyst has been very successful in the international environments in which we work. It is now in use in nine project countries and operates under Windows in Russian, Arabic, Spanish, and English. It has allowed us to introduce a structured methodology in many environments where there simply was no methodology and has proven to be the tool that makes it possible for the local staff to truly follow the methodology.

While Visible Analyst for Windows meets most of the criteria initially set forth, the combination of its cost, power, and ease of use are the primary reasons for its success in our international projects.