

EINA

CASE - AMDC Object Oriented Development tools

SUMMARY

◆ CASE - Tools

- Origin
- Evolution
- Types

◆ AMDC

◆ EINA Architecture

- Modules
- Target Users
- Life Cycle Support

◆ EINA MODULES

- Kernel
 - Variable Templates
 - Change Management
 - Requirements Engineering
- Version Control
- UML Modeling Language
- OCL Constrain Language
- Language Support

◆ EINA MODULES (Cont.)

- Reusability
- Error Tracking
- Knowledge Base
- Database Modeled with MDA
- COCOMO Cost Control
- Team management

◆ Other EINA Modules

- Document Generation
- Graphical Meta-Modeling
- Formal Methods

◆ EINA Step by Step

◆ To be Completed...

EINA CASE - AMDC OO Tools is a technologically advanced, robust and sophisticated product for the automatic development of mission critical computer applications and systems... ***still under construction!***

EINA has been conceived and constructed under the direction of professor **Allen Peralta**, PhD in Computer Science by the Universitat Politècnica de Catalunya, MA in Computer Science by the Universitat Autònoma de Barcelona, Bachelor of Science in Computer Technology by the New York Institute of Technology and with postgraduate studies of Masters of Science in Operational research by the Polytechnic Institute of New York. It is National Prize of Computer science CCS 1975, coauthor of 8 books and author of numerous scientific publication.

A great part of Eina product has been developed by graduated students of computer science that accepted one of the designed modules as an end of career work, under the guide and supervision of EINA's responsible engineering team who later integrated these modules into the final product.

At present time EINA's development (88.6% completed) is halted, for this reason EINA Informàtica S.L. owners and partners are looking for a viable formula to finish it

EINA

CASE - AMDC Object Oriented Development tools

AMDC - CASE Tools

Computer-Aided Software Engineering, (**CASE** Tools) it's a set of tools, programming languages and technical means in general that allow the automatic development of computer applications.

CASE Tools reduce costs, accelerate development and provide comprehensive documentation that can be used for future maintenance or enhancements.

First CASE tools were announced and appeared as early as the 1960s and created a great expectation.

More than forty years ago **procedural** languages were in use. Aids for computer applications and systems development were very little or null. First results frustrated expectations.

Modern, **non-procedural**, languages have made CASE Tool's expectations possible.

- ◆ **Declarative** or Logic Languages, Programs contain assertions of true facts, deduce other information from them. Expressed as declarations that various statements are true (or not). Focus is on facts and rules
- ◆ **Functional** Languages, Programs contain task-oriented instructions similar to procedural paradigm. Expressed as functions returning values and Focus is on action or task, but function-based rather than imperative
- ◆ **Object-Oriented Languages**, (OOPL) Programs are simulations of real-world behavior. Expressed by identifying objects and asking them to perform certain tasks and focus is on object, THEN on task. They allow developers to centered in the problem to be develop

CASE Tools Types

CASE Tools Types are grouped according to the products that are obtained with their use. They are:

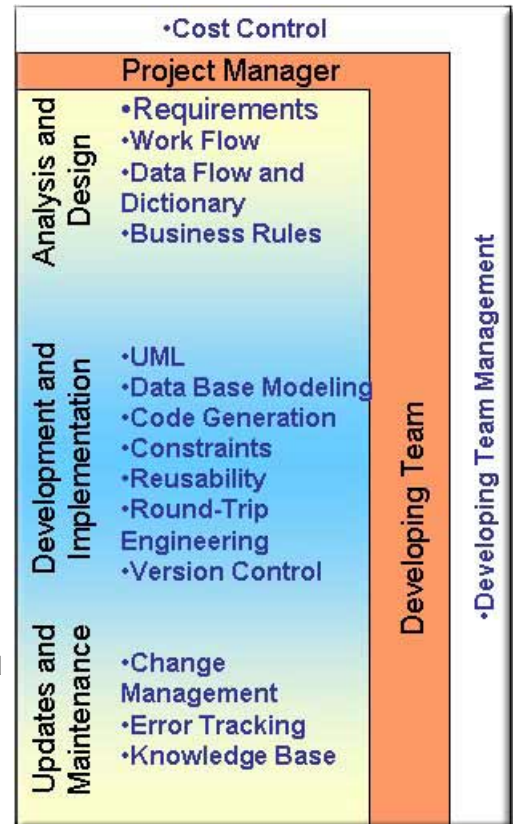
- ◆ **Upper CASE:** Facilitate the elaboration of project's Analysis and Management.
 - ◆ **DFD** (Data Flow Diagrams),
 - ◆ **ERD** (Entity Relationship Diagrams),
 - ◆ Work Flow,
 - ◆ Object Oriented,
 - ◆ Data dictionary,
 - ◆ Business Rules,
 - ◆ Analysis Tools
- ◆ **Lower CASE:** Facilitate computer's product development closer to machine code
 - ◆ Code Generation,
 - ◆ Forms Generation,
 - ◆ Report Generation,
 - ◆ Document generation, and
 - ◆ Import/Export Utilities

Early lower level CASE Tools where well accepted from the beginning.

EINA's CASE Tools are of the Medium / High level type.

AMDC

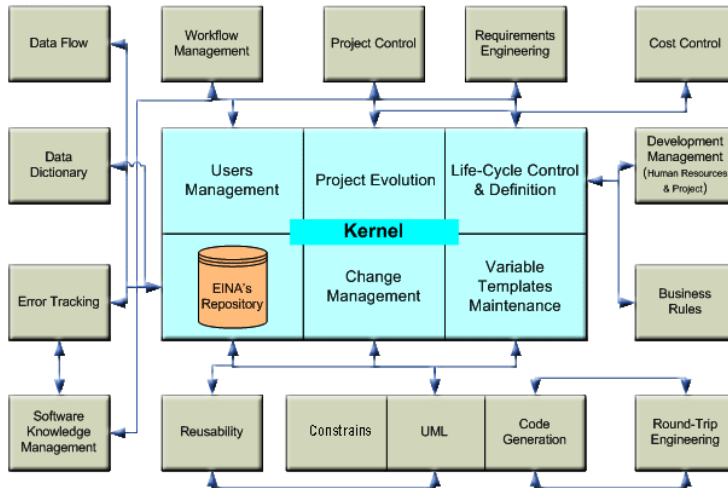
Analysis, Modeling, Design, and Construction tools support formalized methodologies (either object oriented or non-object) that assist in generating application requirements, data definitions, and programming specifications.



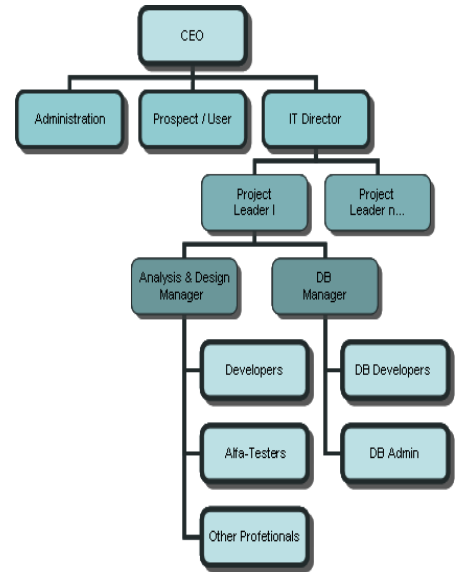
AMDC tools provide automated support of one or more object-oriented methodologies and of some or **all software-development life-cycle phases**, including the ability to construct applications from domains and/or components if that ability is fully integrated and sold with the methodology.

EINA

CASE - AMDC Object Oriented Development tools



Eina Architecture



EINA Target Users

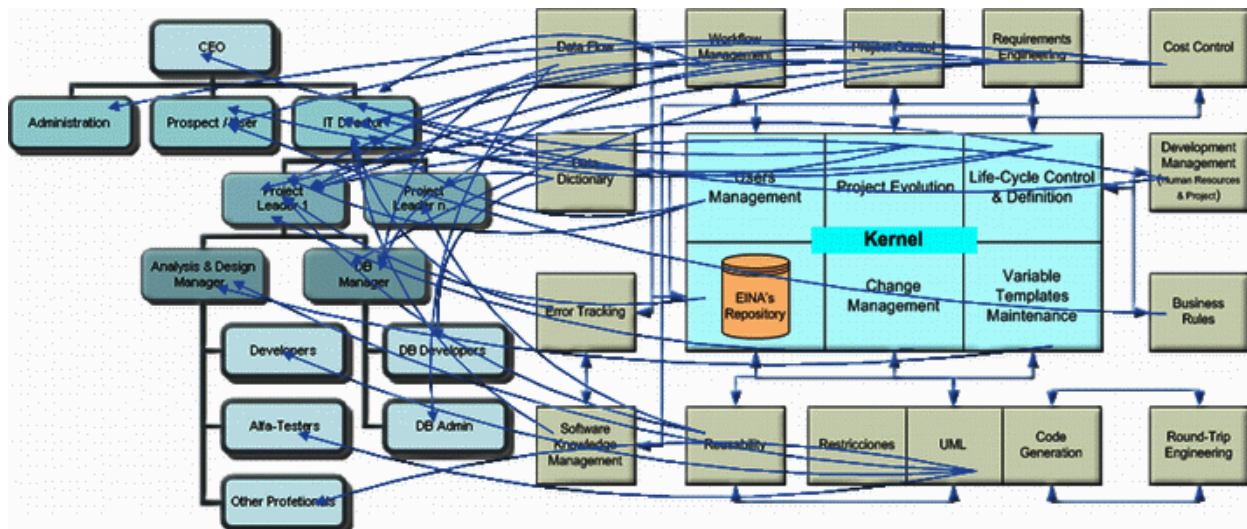
EINA is a tool to be used by all members of a computer project team, of any level, to carry out all necessary tasks to develop programs and/or systems for their end users. Among others, EINA may be use by:

- The **IT Director**, to define methodologies of its technicians.
- The **Project Leader**, to control the project, productivity and costs.
- The **Analyst**, to analyze the functional necessities of users and propose them a designed solution.
- The **Programmers**, to develop programs according to requirements.
- and/or, activating the necessary modules, **all those persons affected by the project.**

EINA Architecture

EINA is an **AMDC** integrated technology product, **Object-Oriented based**, developed in independent modules with specific functionalities intimately integrated

- ◆ EINA is constructed with the following modules:
 - Kernel
 - Requirements Engineering
 - Graphical Version Control
 - UML (Unified Modeling Language)
 - OCL (Object Constraint Language)
 - Language Support
 - Reusability Management
 - Error and Change Tracking
 - Software Knowledge Management
 - Database modeled with MDA
 - COCOMO II (Cost Control)
 - Team Management
- ◆ In addition the following Modules are designed:
 - Document Generation
 - Graphical Meta-Modeling
 - Formal Methods



Life Cycle

EINA

CASE - AMDC Object Oriented Development tools

EINA Modules

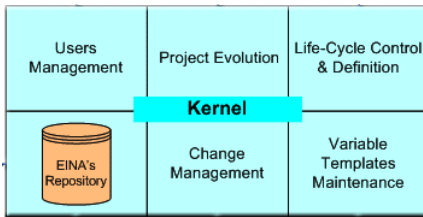
KERNEL

CENTRAL MODULE

System's essential core.

It manages:

- ◆ EINA CASE Tool's database repository.
- ◆ System and Projects Users Maintenance, profiles and privileges.
- ◆ Life cycle definition related to each project.
- ◆ Project's evolution.
- ◆ Variable templates Maintenance,
- ◆ Change Management.
- ◆ Real-time remote collaborative project work



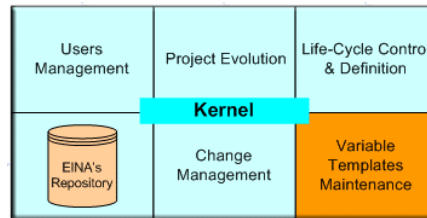
Developed 100%
Integrated 100%

EINA Modules (Cont.)

KERNEL

Variable templates Maintenance

- ◆ Its EINA's documentary extension possibility, following UML extensions profiles and mechanisms
- ◆ These Templates allow users interested in other additional information types to add, among others;
 - A module with "error tracking" characteristics,
 - A cost center,
 - Correction request date,
 - cost, etc..



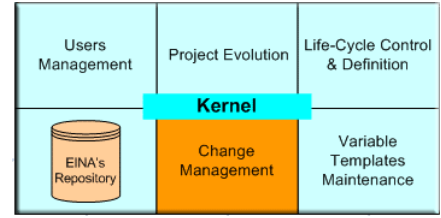
Developed 100%
Integrated 100%

EINA Modules (Cont.)

KERNEL

Change Management

- ◆ This functionality allows a system's development evolution control. For instance: when an Analysis Diagram evolves to a Design Diagram, been two independent elements, "Change Management" records this evolution.
- ◆ In a same way, when a contract requirement is related to its implementation in a Use Case, a monitoring control type, call "traceability", takes place.

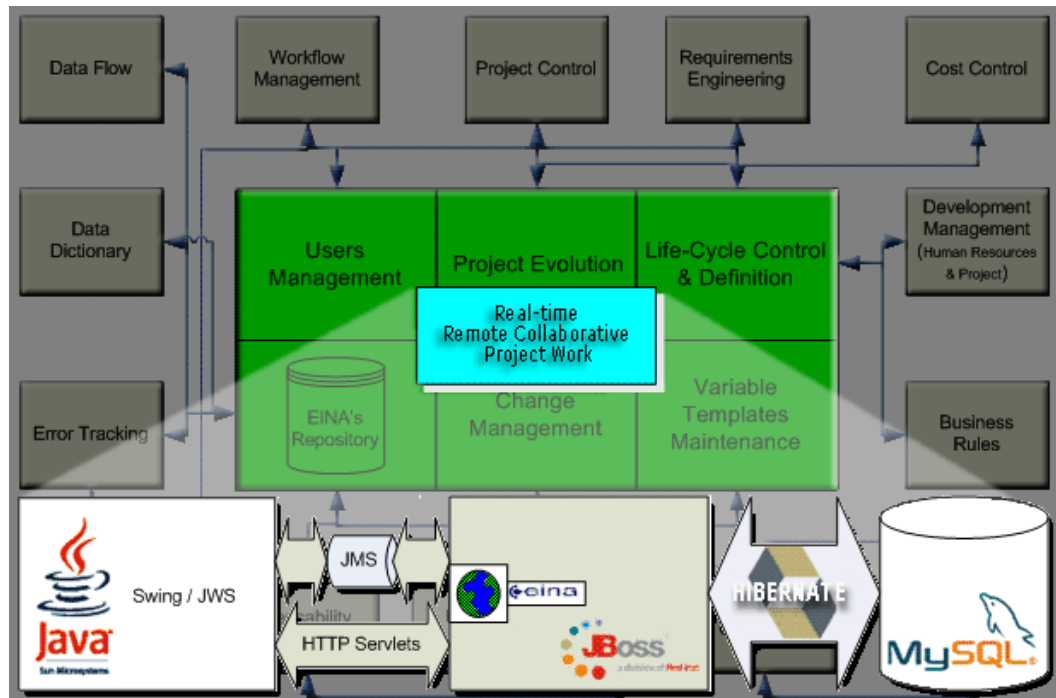


Developed 100%
Integrated 100%

Remote Collaborative Project Work

- Supports the complete life cycle in a complex environment..
- The tool has been designed to be used by all kind of computer professionals, whatever its level or specialty, in a cooperative way, supporting the work of all development teams locally and globally

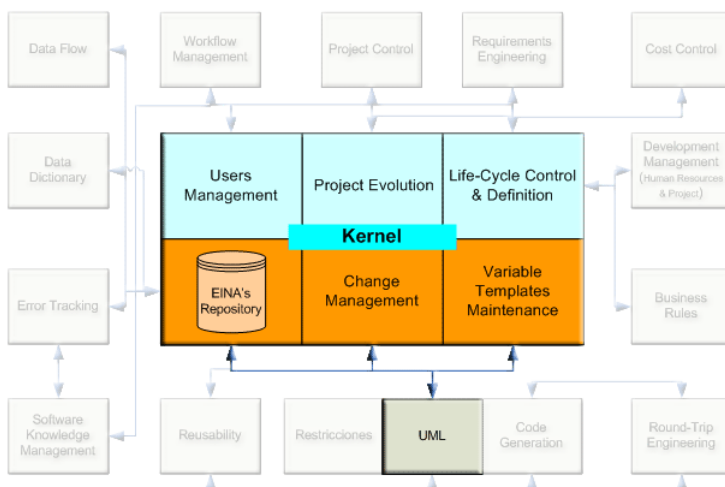
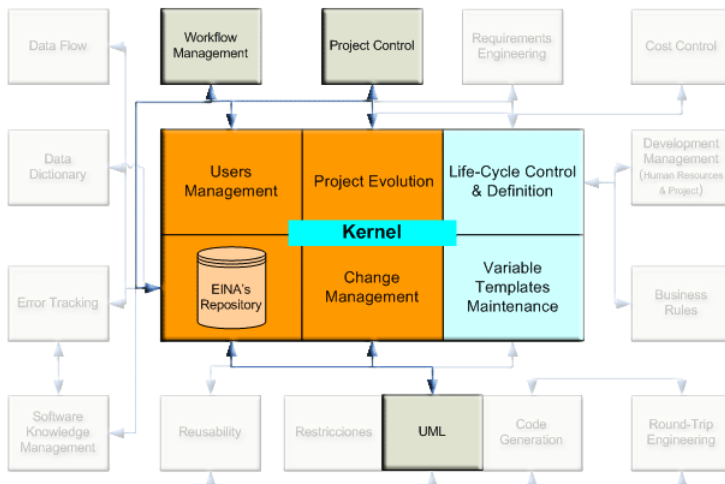
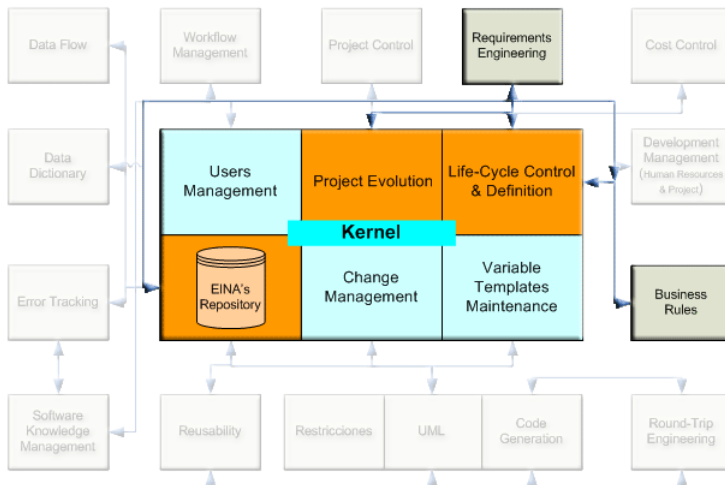
Developed 100%
Integrated 100%



EINA

CASE - AMDC Object Oriented Development tools

EINA Modules



Requirements Engineering

- ◆ The "Requirements Engineering" module records and control user's requirements, normally contained in a contract.
- ◆ "Requirements Engineering" module allows to register all user requirements contained in a contract, establish Traces with its UML implementation and plans necessary tasks for its implementation.
- ◆ "Requirements Engineering" control module guarantees that each requirement is implemented as specify in the contract. (Cost of an error in the management process is from 10 to 20 times bigger than an error in analysis, design and/or programming).

Developed 90%
Integrated 100%

Graphical Version Control

EINA's Kernel has a "Graphical Version Control" or "Change Management" module for programs, diagrams and any other artifact. EINA compares versions to find differences.

Comparing textual data is an easy task to accomplish. Showing diagrams differences requires showing graphical, textual and hidden documentary templates differences.

Developer 100%
Integrated 100%

UML (Unified Modeling Language)

This module supports diagramming UML 2.0 models.

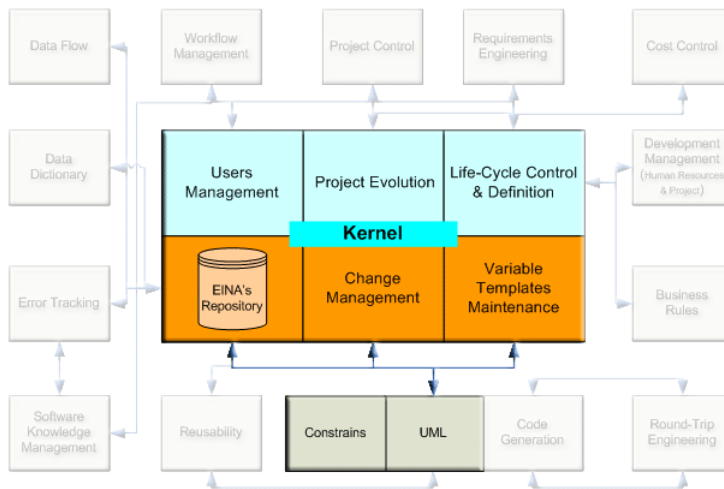
EINA supports UML version 2.0 with all its graphic requirements and/or de meta-model. It also supports Diagram Interchange Available Specification.

Developed 100%
Integrated 100%

EINA

CASE - AMDC Object Oriented Development tools

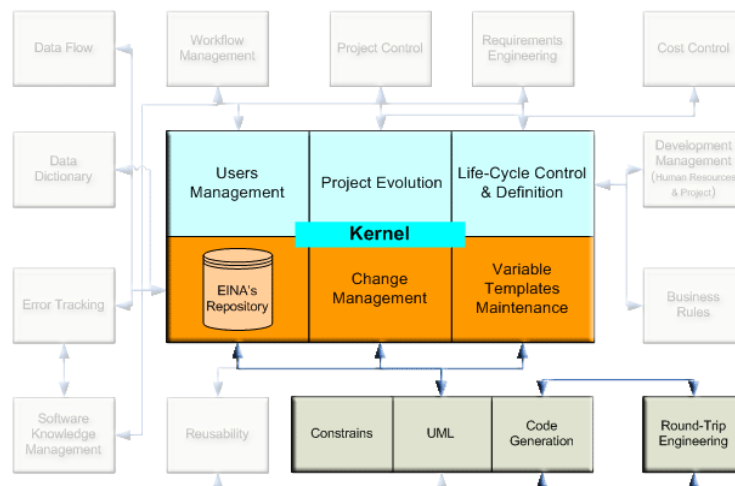
EINA Modules



Language Support

- ◆ EINA is designed to support all commercial object-oriented "round-trip engineering" languages. In addition to generate code, EINA can import code modifying the repository design and/or generating new versions. To ensure round-trip's functionality, EINA does not use mark-up languages.
- ◆ Code generation can be adapted to users preferences customizing it through variable templates.
- ◆ EINA supports **Java, Smalltalk, C++, C# and Cobol 200x**.
- ◆ To adapt EINA's development tool to the specific supported language and to avoid syntactic errors as much as possible, a "language environment" can be specified during design model's development.

Developer 65%
Integrated 65%



OCL (Object Constraint Language)

- ◆ "Object Constraint Language", is a modeling language that is part of the UML standard version 2.0. It is used to specify all kinds of constraints, pre- and post-conditions, guards, etc. over the objects in the different model.
- ◆ Constraints can be set over the meta-model and/or over the model.
- ◆ With OCL business standards for the language, project, etc. can be define. This constraints will be checked by an interpreter. In addition to Boolean data, the interpreter can also return numerical data, sets, etc. so it can be use as a query over meta-models and/or models.
- ◆ Another advantage of OCL query is the possibility of obtaining, metrics for the applications. When evaluating a system complexity, Metrics gives productivity and quality figures related with the amount of error occurred during the program life cycle.

Developed 100%
Integrated 100%

EINA

CASE - AMDC Object Oriented Development tools

EINA Modules

Reusability Management

◆ Reusability Management is essential in an Object-Oriented environment to improve development productivity and quality. An error free certified component or "asset" to be reused saves programming time and improves the application quality.

◆ Reusability Management begins with a certification process to define the policy of the product to be developed, then follows a cataloguing process of the component using key words. The component can be placed physically in the library or it can be store in an external disk.

◆ EINA's reusability has many possibilities, in addition to the component cataloguing feature it also can catalogue documentary templates, processing manuals, etc. This is accomplished defining components classes with different information levels.

Developed 100%
Integrated 75%

Error and Change Tracking

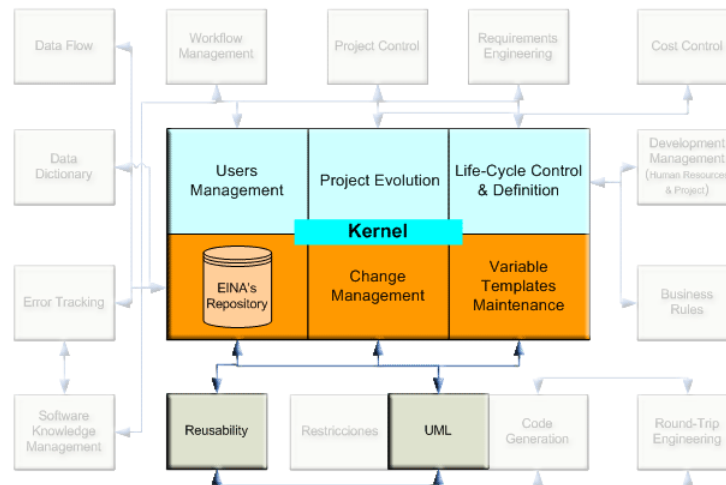
◆ When an error is identified it is register into the database and a correction process begins.

◆ The following data is registered for each error:

- user who identified the error,
- a problem description,
- error's detection date,
- error's severity,
- cost center to charge the correction of the error and
- technician assigned to fix it.

◆ Error correction protocol:

- The technician assigned to the error performs a cost and time estimation.
- The technician creates a "Trace" between the error in the program and all modules to be modified.
- The technician assigned to the error takes the correction task.
- The completion date is registered once corrected the error (to

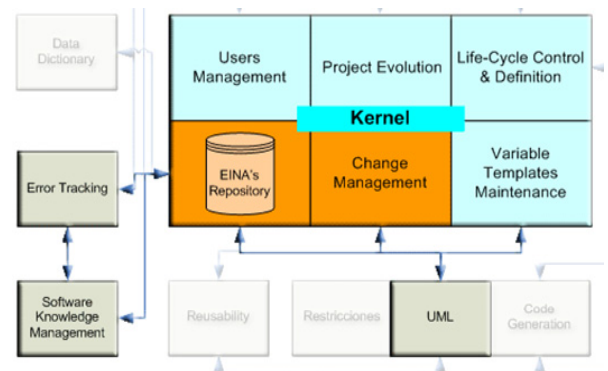


keep a record of response time to users),

- a new release is issued to the user.

◆ "Change tracking" operates in a very similar way. In addition it has other features like associated cost centers, lower priorities, etc.

Developed 90%
Integrated 100%

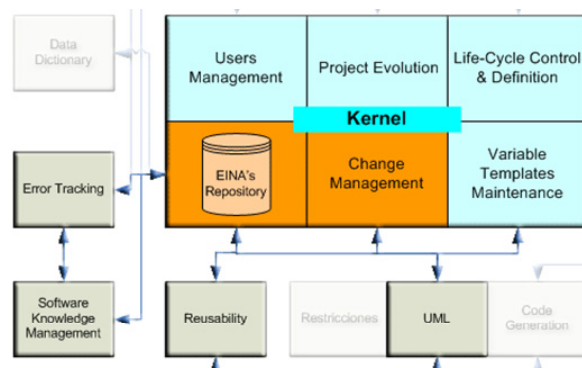


Software Knowledge Management

◆ The same module that allows cataloguing computer components can also list any type of document related or not to a component. These documents can be associated to others to create a document network, or correlations, that describe part of the organization's computer knowledge.

Documents can be internal or external and if in XML format, cataloguing may be much more effective.

Developed 100%
Integrated 100%



EINA

CASE - AMDC Object Oriented Development tools

EINA Modules

Database modeled with MDA

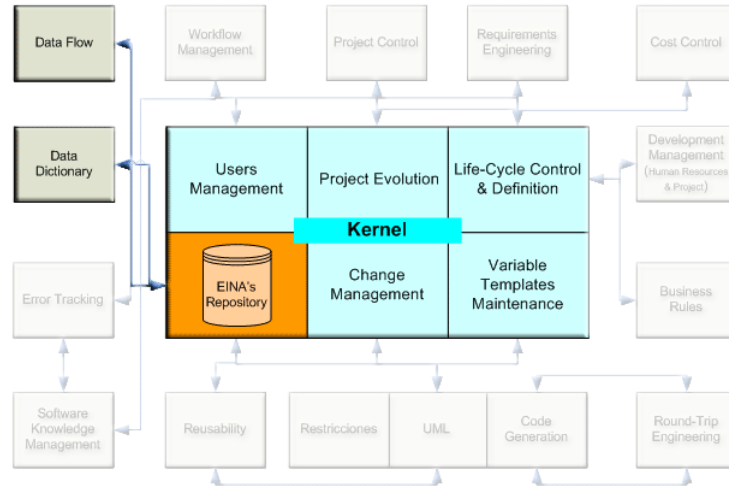
- ◆ There are many analyses and design AMDC Tools that support database's design process, and construct objects and/or components that allow database's applications access..
- ◆ Some situations require a round-trip between SQL and data base design, not frequently supported by AMDC tools. For instance when, in order to obtain an occasional needed information, a change to a table design arises (add, delete or modify one or more fields). The change to the database and the necessary programming are done, but the change is rarely registered. If this happens often, actual database design can not be recreated from existing textual SQL.
- ◆ EINA's database module based on the Relational Model -not MLU-, facilitates both; developers and DBA's functions.

Desarrollado al 50%
Integrado al 50%

EINA supports COCOMO II

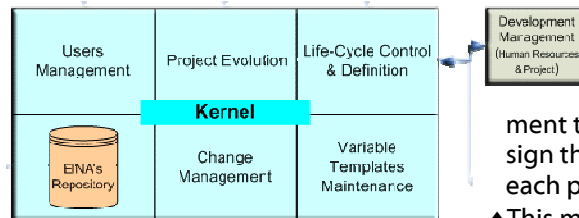
- ◆ In a comter Project costs control many parameters are involved:
 - Development environment,
 - Hardware and software infrastructures, and
 - Base Software and hardware stability etc.
- ◆ Other aspects like application's size, life cycle, quantity and skills of persons assigned to the project, etc. affect cost estimation and cost it self.
- ◆ COCOMO II (Constructive COST Model) its the best application's cost estimation model developed by Barry Boehm.

Developed 100%
Integrated 0%

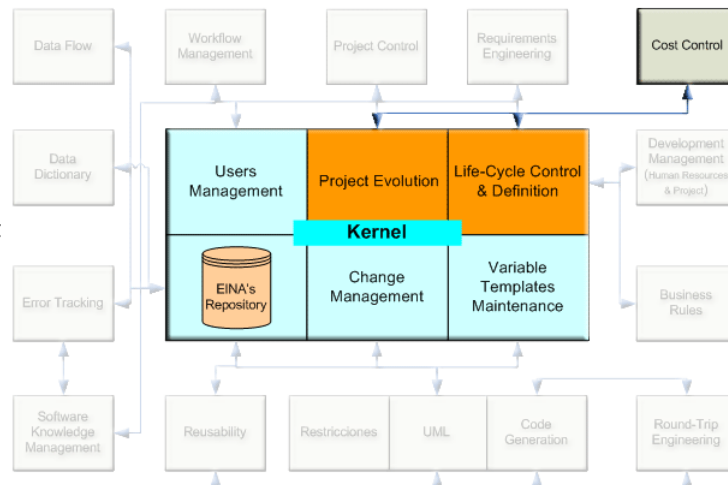


Team Management

- ◆ Team Management module allows:
 - tasks planning,
 - estimate assigned work loads to each person,
 - detect time, budget, etc. deviations.
- ◆ This module is very close related to all system's development steps (analysis, design, programming), captures individual and/or collective development's metrics that can be use to improve future development teams estimations and design them to be more adjusted to each project.
- ◆ This module is linked to costs centers.



Developed 90%
Integrated 90%



EINA

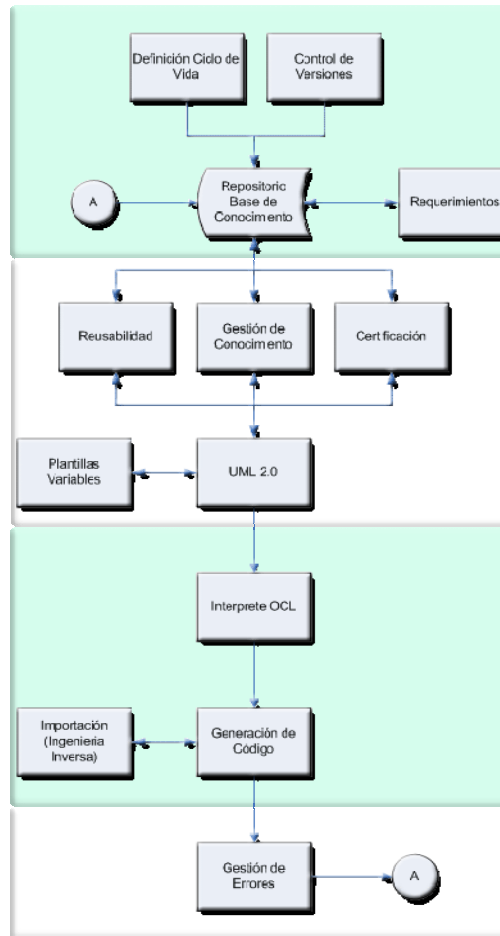
CASE - AMDC Object Oriented Development tools

Other EINA Modules

The following modules are designed and/or developed to some extent:

- ◆ **Document Generation** Documentation will allow to define and integrate the type of document to be generated; Help, Users Guide, Technical Manual etc.
- ◆ **Graphical Meta-Modeling** In addition to standards, there are other useful diagrams that can complement UML 2.0. For instance hierarchical diagrams to show the organizations' organigram, document's flow, etc. EINA's Graphical Meta-Modeling will permit to define diagrams at a meta-model level, establishing application's figures, rules and associations to create a diagram.
- ◆ **Formal methods** are mathematically based techniques for the specification, development and verification of error free (or almost) software and systems applications. Formal Methods are not used very much because they are complex and require a great preparation. EINA pretends to include UML compatible formal methods for the development of mission critical computer applications and systems.

EINA Paso a Paso



- ◆ **Project Management:** Allows project's Life Cycle definition, creation, updating and version control.
- ◆ **Requirements Management:** Allows project requirement's captures, management and updating.
- ◆ **Data Modeling:** Supports Database design.
- ◆ **Application Modeling:** Allows UML Application's analysis, design and development.
- ◆ **Variable Templates:** This Module supports all other Modules.
- ◆ **Reusability:** All EINA's generated objects and artifacts.
- ◆ **Code Generation:** Allows machine execution of modeled systems, including their related constrains.
- ◆ **Constrains Control:** Project's Constrains Generation and management.
- ◆ **Error Management:** Allows error's identification, record and traceability.
- ◆ **Document Generation:** Advanced support to reports generation.

To be completed...

The product, according to its present development stage (about 86.6%), works correctly as projected, fulfills all functional characteristics of its creators and has successfully pass all required tests. At present time (third quarter 2006), development has been halted for 18 months. To start marketing it, it's necessary:

- ◆ Finish coding and upgrade technology
- ◆ Formalize and translate into English manuals; User, Programming, Guides and Tutorials
- ◆ Design selling policy and marketing strategic
- ◆ Launch the product

(Talk to us about details – contact us-)

Contact Us: