

THE MANTRA IDP

Designed by



**4633 Old Ironsides Drive, Suite 415
Santa Clara, California 95054
Phone (408) 492 0771**

This is IT – The Mantra for Code Developers...

Web-applications development involves a complex interaction of industry standard APIs, protocols, programming, markup and query languages, connection to disparate sources and content creation tools. Building even modestly complex web applications requires a team of developers in a variety of roles with a broad range of skill sets. Typically, this development complexity is managed with a class of software called "*web application frameworks*". However, current frameworks tend to focus on limited ranges of development roles/skills, or require specialized knowledge in order to be used productively.

What is really required is a web-applications development platform that facilitates process flow amongst the various stages in the development process, provides a basic framework for the application, allows for integration and collaborative structures to be established with external and legacy systems, reuses services and code, and most of all enables a orchestrated, coordinated and collaborative application build.

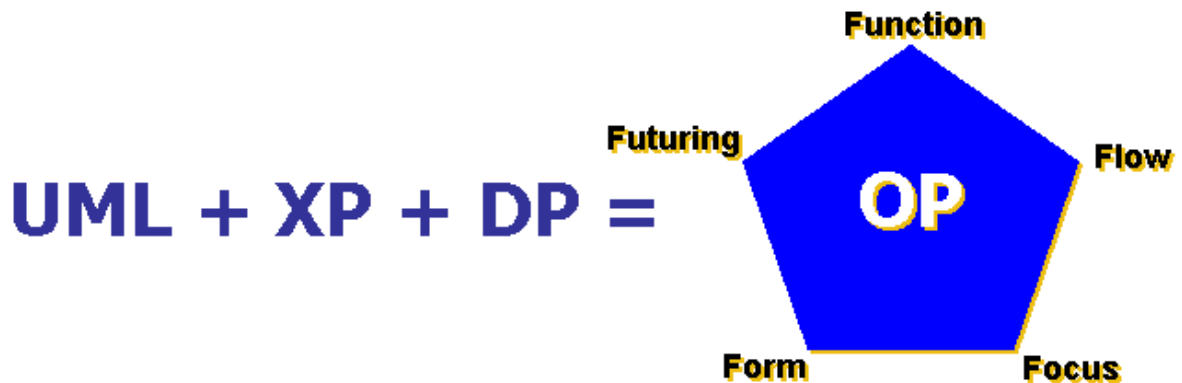
MANTRA does just that....

Software Engineering is defined as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software. It deals with applying methods and scientific knowledge to create practical cost-effective solutions for the design, construction, operation and sustenance of software and associated products. Engineering is the methodology of applying math and science to synthesize a constrained solution to the needs of a customer. Software engineering is applying these principles to software.

MANTRA is just that....

Based on solid foundations of the Software Design Philosophy called **Organic Programming (OP)**, it uses tenets of *UML, XP and Agile Software* development practices to transform rote code to automated procedures so that the users need concentrate only on the business logic when developing applications. OP incorporates the lifecycles of waterfall, staged, evolutionary delivery, incremental and spiral approaches to software development to create robust, scalable and flexible systems.

Organic programming

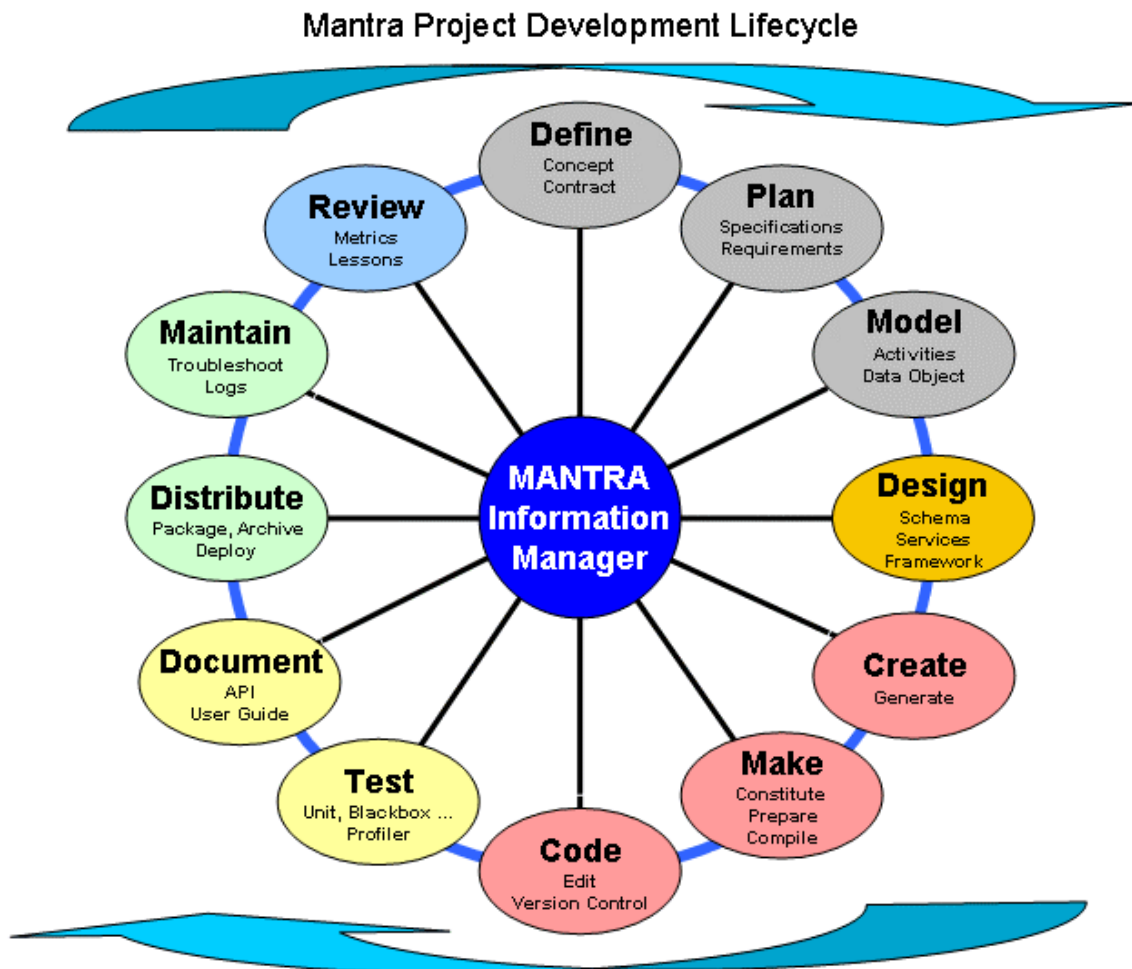


“It is not a process of addition, in which performing parts are combined to create a whole, but a process of unfolding, like the evolution of an embryo”

- Christopher Alexander

Organic Programming is a new software methodology (defined by Nitin Uchil – Founder and CTO of BDrive, Inc.) designed for constructing complex yet flexible structures of program modules. It is about a new philosophy of software engineering - where form and function are one, where documentation and code are written in unison. Much like Organic Architecture, it is an attempt to integrate modules into a coherent whole: a marriage between the programming construct and its functionality and a union between the abstraction and its implementation. It is organized about the concepts of 5Fs (Function, Flow, Focus, Form and Futuring) to build robust, scalable and resilient software architectures. It reinforces the need that software development should be iterative rather than following the waterfall approach of current practices. Organic Programming is actually a deliberate and disciplined approach to software development. It combines the coding structures of UML and the best practices of XP to bring about object-oriented code that is component based and bug free. It follows the philosophy of Gestalt - "The whole is greater than the sum of its parts". It stresses on a "Zero Administration Client" environment once the application is production deployed. It is an assimilation and re-organization of best practices followed by Mindware (part of BDrive, Inc.) in its projects of building knowledge management frameworks at major Automakers to optimize processes in the design and product development phase of Product Lifecycle Management (PLM).

About Mantra



Mantra, seamlessly steps thru the different entities of application define, plan, model, design, develop and deploy of database, services, code, business logic and front end interface, using a visual web-enabled depiction scheme that enables collaboration and re-use of existing design patterns. It simplifies and integrates the design, testing, debugging, deployment and administration of complex applications. It allows business analysts, application architects, software developers and *clients* to work in a single environment, minimizing the need to acquire, learn, and use multiple products, thus reducing software development costs, improving code quality and decreasing build time.

Key Features

Web Based Integrated Development Platform

Mantra provides a web-based solution that handles all aspects of software development from System Design, Planning, Development, Documentation, Testing, Distribution, Maintenance and Reviews. Being web based all project members can access the development platform from anywhere. It also provides for iterative development involving the client in every stage, ensuring that at all times development is resulting in deliverables, per client requirements. Changes can also be incorporated with ease, and implemented seamlessly throughout the development process.

Central Repository of Project Information:

Mantra provides a single place to store and view project information – articles related to project, project meta data, team members and their roles, milestones in project development, emails related to project etc. Storing information in a central repository enables all project stakeholders to be abreast with latest developments in the project at any given point of time.

Web-based Collaborative Project Management:

Being a web based development platform Mantra provides for seamless collaboration between geographically dispersed team members. Members can view requirements submitted to them, share files, import and export code, compile and execute code, edit code etc. from virtually anywhere. Mantra has in-built mechanisms for coordinating and integrating code from different team members, versioning etc.

Visual Modeling Interface

Mantra provides a visual modeling interface enabling developers to visualize the architecture and relationships of an application.

Automated Code Generation

Mantra auto generates a significant portion of the code resulting in accelerates development. Auto generation is enabled using Mantra's proprietary design patterns technology. Mantra also auto generates connectivity to external sources such as databases, file services etc. Design patterns also infuse the reuse of components optimizing the code development process. Developers can build libraries of components and select and customize them according to application requirements.

Monitoring Mechanism

Mantra provides a web based monitoring mechanism to administer sources and web-servers. Users can view active sources and test them, start and stop web-servers with a web-enabled interface. The monitor also provides adhoc query generating capability across multiple databases. Mantra also automatically summarizes code by category, so that project stakeholders can gain an understanding of the application development effort. Mantra provides for backup and storage of application data. Mantra also has an in-built project manger, which manages the timeline and progress of the development effort.

Built-in Test Capabilities:

Mantra has built-in test utilities that can perform different types of testing – source, unit, module, blackbox, whitebox, stress, regression and optimize. Testing is automated eliminating manual effort to do testing. Testing can also be implemented side-by-side with development, ensuring quality of the code.

Documentation Management:

Mantra handles all document needs of application development – User Guides, Javadoc API's, Reference Manuals, User Guides etc. using LIPI – BDrive's proprietary knowledge management technology. LIPI provides for document collaboration, web site content and versioning using a web based interface. LIPI not only to classify content related to application builds but also represents procedures, methods and encyclopedias. LIPI is essential to manage the content bits that influence the whys of an application build and to harmonize the best practices for future reuse.

Distribution, Maintenance and Reviews:

Applications are deployed and maintained using a web-interface. Maintenance is facilitated through a request management system that is coupled to an internal Bug-tracking Utility. Clients can submit issues and monitor the progress of the issue using a web-enabled interface. Mantra also facilitates cataloging of best practices and methodologies, lessons learnt etc., so that knowledge gained through each development effort is not lost.

Key Benefits

- One integrated platform for the entire development process.
- Web-enabled collaborative development process, which can unify geographically, distributed development teams.
- Automates most of the development process resulting in significant savings in time and cost.
- Providing expert-level insights with guidance on their proper application and adaptation using pattern technology
- Iterative and highly flexible client-driven development
- Significant reuse of components throughout the development process
- Streamlines development by integrating with leading IDE's
- Development becomes highly collaborative and all stakeholders can gain deep visibility into the progress of the project at any given point of time.
- Complete open standards approach integrating with all operating systems, existing legacy systems and relational databases

Management of the Application

Mantra dynamically creates a progress metrics dashboard based on data collected from your development platform. All team members benefit from having an easy-to-use and always-updated place to go for all project information. It catalogs all discussions pertaining to the project in one place for future reference. Mantra presents the results graphically so that you can easily assess project progress and quality. This allows you to better predict which areas will require special attention and where to focus scarce resources to stay on schedule, and enables you to make decisions based on quantitative analysis. Accurate and accessible project status information allows you to better anticipate problems or delays and make informed decisions in order to avoid them. Better project predictability helps you to keep projects on schedule and on budget. The Manage entity also enables you monitor access to a project with its built in login feature. This built-in security prevents unauthorized access/ use of project information. It also enables project managers to determine who is working in what phase of the project and re-allocate work if appropriate. It also offers a web-based bug tracking system to report, assign, solve, test and close bugs.

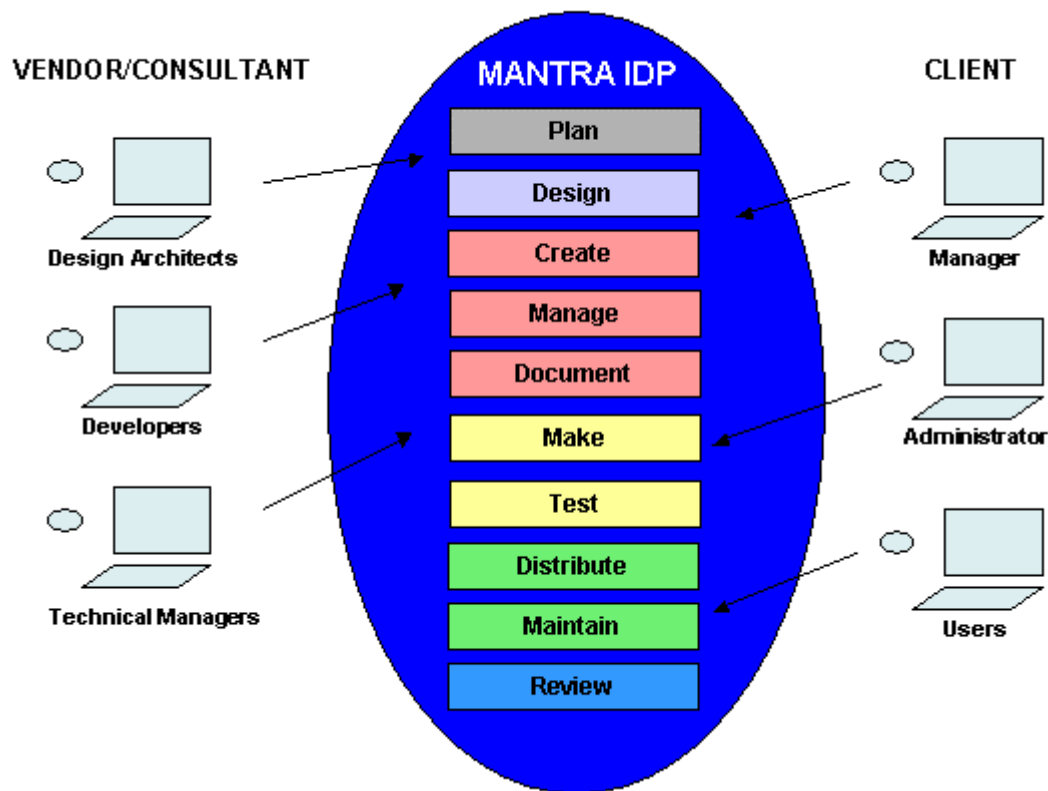
The Creation

Mantra develops applications in a 12-Phase, 6-Stage, web-based visual environment that consists of:

1. **Conceptualize**
2. **Architect**
3. **Develop**
4. **Validate**
5. **Deploy**
6. **Review**

Everything related to the above stages is stored in an XML file and a back-end relational database, so that change-control, version-control and synchronicity can be facilitated. MANTRA auto-generates the SQL scripts, the connection properties (services) and the application framework. These can then be imported to your local IDE (JBuilder or Visual Cafe) to incorporate the business logic. The following table shows the map between the stage and phase and describes what each facet does:

Phase	Stage	Description
Define	Conceptualize	Concept definition and Proposal
Plan	Conceptualize	Requirements Analysis, Specifications, Use Cases, Workplan
Model	Architect	Create visual representation of the flow using IDEF structures
Design	Architect	Skeletal Representations – Schema, Services and Framework.
Create	Develop	Auto-generate code, Migrate to local IDE
Edit	Develop	Edit code segment with WebDAV/Delta V version control
Make	Validate	Compile, Code for local viewing
Test	Validate	Multiple Testing Philosophies to validate the application
Document	Develop	Auto-create – API, Manuals and Guides
Distribute	Deploy	Create Archives (JAR, WAR, EAR) for client, manage Builds
Maintain	Deploy	Handle Maintenance requests, Surveys and Troubleshooting
Review	Review	Catalog – sunset Reviews, Lessons Learned and Best Practices



The ultimate goal of MANTRA is to be the single source to create, develop code, manage, distribute and maintain applications for the Enterprise

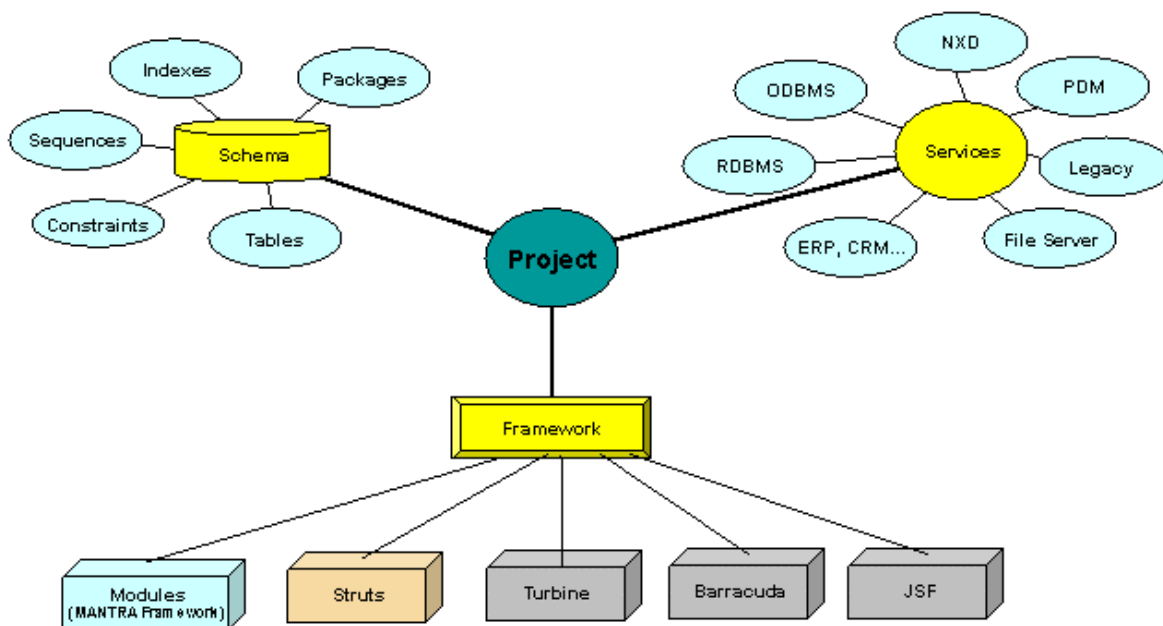
The Conceptualize Stage

Mantra enables architects and managers lay the foundation of any IT project in a collaborative manner. It helps in forming the team for the project, assigning roles to the team members (based on their Skill sets) and depicting the basic XML file for the project. It also enables them to catalog articles if any, related to the project and define requirements for the project. It allows for an iterative Requirements definition process, its analysis and the creation of Specifications and Use Cases, ensuring that development at all times is proceeding per the need of the client.

The Architect Stage

Mantra provides comprehensive system for modeling a complete e-business application by designing the schema, services and framework in this phase. Using Model Driven Architecture (MDA) concepts, UML 2.0 and the Integrated Definition (IDEF) structure, the Data Objects and Activities related to the application are dynamically created. Design Patterns are used for developing and deploying component based frameworks that are robust, scalable, resilient, reusable and extensible This enables a broad class of application developers to become proficient in developing applications for the complex n-tier Internet technologies common in today's e-business environment. It increases the productivity of

developers by providing a complete modeling environment and extending those models into actual generated code and executable applications. This allows for a more iterative design process, which results in higher quality applications brought to market more quickly. Mantra's architectural tools define the business model, transaction model, user interfaces, integrations, and the application's deployment configuration in a language-neutral, technology-independent approach. This produces applications that are agile, independent of both platform and channel and constantly evolving.



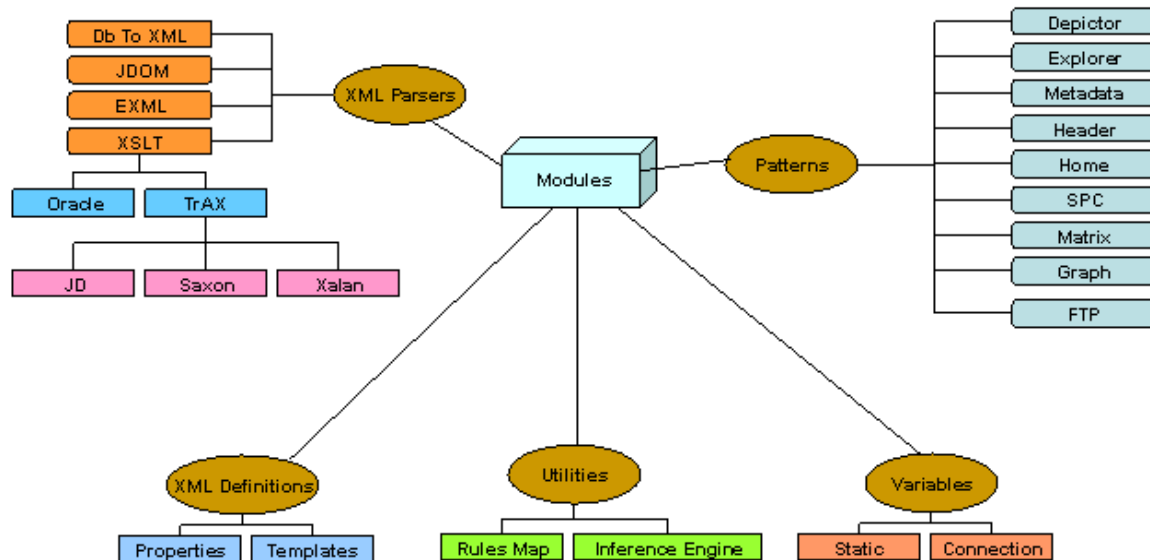
As indicated above, the Design Phase currently has 3 Components:

Schema: This helps design relational databases (the definition of tables, constraints, indexes, sequences and packages).

Services: This helps define the connection properties (the definition of sources and the mechanisms to use). It also helps us create Web Services to collaborate between disparate systems.

Framework: This enables the building of components using the MVC (Model-View-Controller) frameworks in Mantra, Struts, Turbine and Barracuda. Using Java Server Faces (JSF) is anticipated for future releases of Mantra.

The MANTRA framework constitutes the application using Design Patterns, XML Parsing Techniques, Inference Engines and Rules Map as show below:



The Development Stage

Mantra enables build applications in a collaborative, efficient manner using its proprietary patterns based technology. It combines the features of automatic code generation, seamless deployment, project management, bug tracking and Information Sharing all in one module. It comprises the following modules:

Create: Uses Pattern based technology to accelerate code generation. Provides UML visualization, code templates, and automatic or manual synchronization of code and models. Specifically it provides for:

- Single design-to-code experience
- Manual or automatic code synchronization
- User definable patterns and code templates to automate repetitive coding tasks
- Free-form modeling for customized, domain-specific shapes
- Cross-model references and versioning up to the class and diagram level allow structuring to fit any project.
- Import/Export of Files and Information to and from the local IDE.

Code Edit: Dynamic via the web editing code segments is facilitated with version control using CVS. We are investigating WebDAV's Delta V protocol to better manage configurations and code versions.

Document: Documentation is an inherent part of Mantra's Software Development process – Mantra creates software documentation automatically and flexibly while maintaining consistency with your software development tools and processes. It provides a single document-automation solution across the software lifecycle and creates APIs (JavaDocs), Reference Manuals, User Guide and Developer's Guides via a XML based template management system called "Publish".

Make: Compiles and Deploys the application. It uses Apache's ANT to create the requisite links and command scripts to be able to serve the application on the local web-server.

Test: Manages the Entire Testing Process. It includes the following types of tests.

1. **Source Test.** Check if backend sources are alive.
2. **Unit Test:** Independent testing of the individual elements without any interactions.
3. **Module Test:** This will test the Servlet Output of entering a module.
4. **Blackbox:** The tester knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. Here the tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications.
5. **Whitebox:** White-box testing strategies include designing tests such that every line of source code is executed at least once, or requiring every function to be individually tested.
6. **Stress:** Stress testing is subjecting the system to an unreasonable load while denying it its resources (RAM, disc, mips, interrupts etc.) needed to process that load. The idea is to stress a system to the breaking point in order to find out bugs that will make the break potentially harmful.
7. **Regression:** Regression Testing ensures that reported errors are indeed fixed and that fixes made to the application do not cause new errors to occur.

The Deployment Stage

Compiles, Deploys and Maintains the Application build for a client. Using ANT, it versions the build and creates an exact compiled environment that can be distributed to the client's web/application server. It is made up of two phases: distribute and maintain.

Distribute: Mantra allows you to select from a broad range of deployment technologies and then automatically constructs complete application code optimized for those technologies, from your application model. Mantra generates executable applications across all layers of an n-tier system. Optimization techniques ensure high performance scalability and availability for your application. Data flow and transitions are minimized for efficient transaction processing. Mantra supports the leading deployment technologies common in business applications, including:

- **Application Servers** - Automated deployment for J2EE application servers, including IBM Web Sphere, BEA Web Logic, Oracle 9iAS, HP Application Server
- **Relational Databases** - including IBM DB2, Microsoft SQL Server, Oracle and Sybase
- **Messaging** - including XML (and EDI and other formats through translators), along with leading transport platforms such as JMS, IBM MQ Series and TIBCO
- **Wireless devices** - using the WML standard
- **Browsers** - including Netscape, IE and others

Maintain: In the maintain phase, MANTRA manages the Maintenance issues – the creating of the contact, handling of requests (accepting them and converting them to issues, and validating them) and the lifecycle of updating the version/revision to include changes. It also includes cataloging of Feedbacks and the maintaining of the FAQ list. Surveys can also be created and disseminated to the end-users to check on the health of the application.

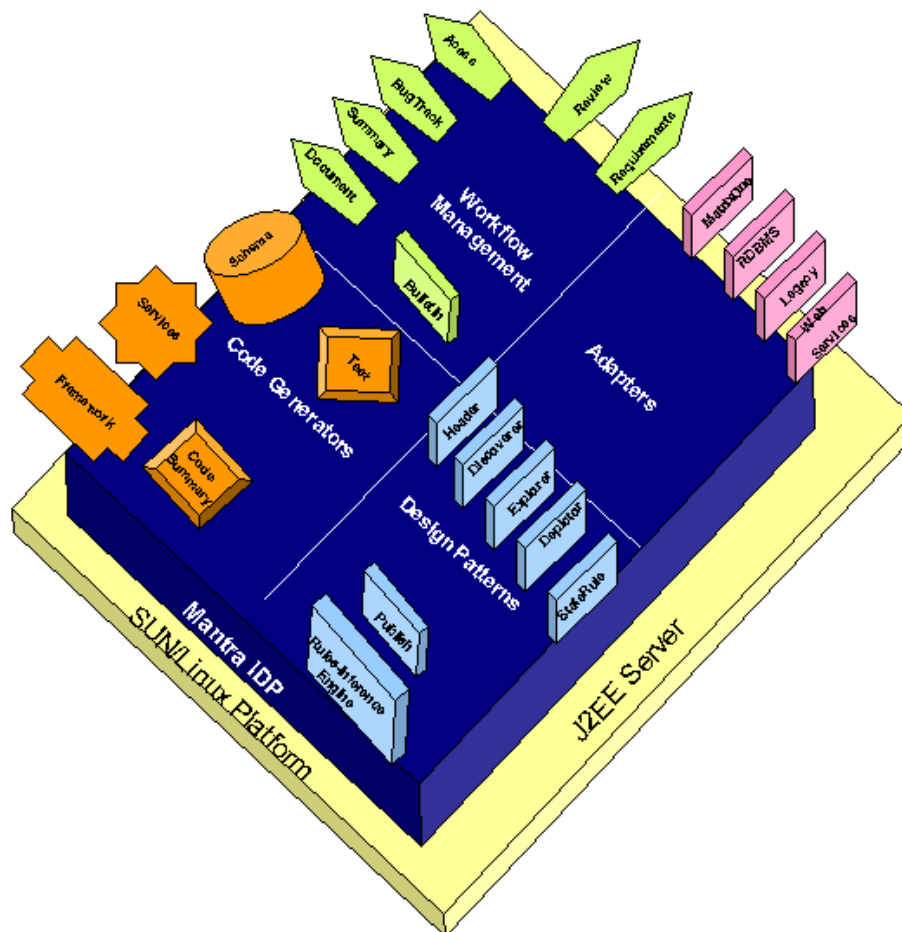
The Review Stage

In the review stage, Lessons Learned and Best Practices are cataloged for later review so that Knowledge artifacts can be showcased for re-use in the next version of the project or in another project. Summary Metrics of the project can also be gathered and sunset reviews conducted so that the overall state of the application is ascertained.

Our Differentiator

	Typical	BDrive
Idea	Concept	Concept
Product Spec	Microsoft Word	MANTRA™
Design Spec	Microsoft Word	MANTRA™
RAD	Rational Rose	MANTRA™
Project Plan	Microsoft Excel	MANTRA™
Task Management	Microsoft Project	MANTRA™
Resource Scheduling	None	MANTRA™
Document Management	Microsoft Word	MANTRA™
Object Modeling	Together J	MANTRA™
Database Development	ERWin/JDesigner	MANTRA™
Application Development	JBuilder	MANTRA™
UI Development	MM Ultradev	MANTRA™
Development Environment	Win32	Browser Based
Deployment	Migration involved	No migration required
Version Control	Clearcase/Purify	MANTRA™
Security	Some Tool	MANTRA™
Workflow	Some Tool	MANTRA™

The MANTRA Architecture Stack



In Summary

Mantra provides an ideal environment for developers to create applications in a rapid development framework. Consultants can use it to architect their solutions as they are creating strategies and process maps for enterprises. Since the platform is web enabled, collaborative development and management of the application are facilitated so that large-scale infrastructures can be built. None of the existing tools come even close to matching the agility of MANTRA's application build.

Move over Rational, Together Soft ...- there is a new kid in town.