

Where do frameworks fit in application development?

Frameworks are everywhere. Frameworks are available for almost every part of software development and they are not exclusive to the IT industry. Medicine, banking and insurance all have risk assessment frameworks. Education uses learning frameworks, companies are measured by quality frameworks and the USA Department of Defense has a framework for enterprise architecture.

Why are frameworks popular? Where do they apply in software development? How can frameworks help to reduce cost and boost productivity? Before addressing these questions, we need to understand what we mean by framework.

What is a Framework?

Imagine building a house. You start with a vacant block of land, prepare the land and then you lay the foundations. The next step is building the floors, walls, ceilings and roof, followed by wiring the electrical components and connecting the water. At this point the house is operational but not livable. The next stage in the project includes painting, laying carpet, installing appliances and landscaping. Now the house is complete and livable but not finished. The last stage is moving in, setting up furniture, deciding who will occupy the biggest bedroom and so on. A project home is the ultimate framework, everything is complete and moving in is the only work to do. In contrast, the most work occurs when the starting point is a vacant block of land.

In the software context, frameworks give you something on which to build. They provide a quick start to developing an application. The closer the framework is to the finished application the better, reducing the effort needed to build the application. From the house illustration, a framework that provides the foundations, walls and roof is a quick start to finishing the house.

Software frameworks are available for use in each application layer: **user interface, business process & logic, and data access**. They fit into three categories: models, libraries or blue prints:

Model (or template) frameworks are software structures that provide a near complete foundation from which to build applications. Developers install the framework



Richard Lancaster
LANSA Product Center

and add the application specific parts. Model frameworks are typically used in the user interface layer, where they provide assembled components upon which to build the user interface and interactions with the business process and logic layer.

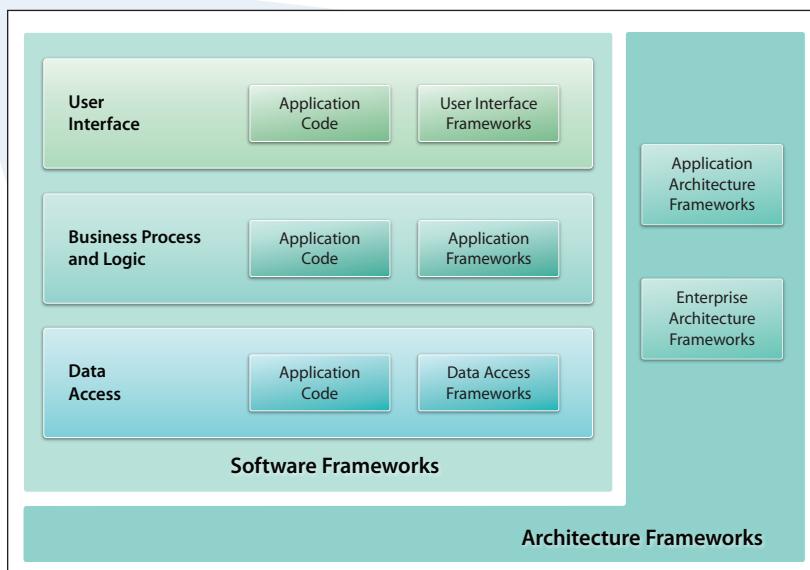
Library frameworks are software collections that provide working software to accomplish specific functions, saving developers the effort to build their own. Library frameworks are available for all application layers, providing specific functions applicable to the layer in which they participate. Examples are JavaScript libraries (jQuery and Yahoo User Interface) in the user interface layer. Application frameworks (Apache Struts, Zend Framework and LANSA Workflow Framework) in the business layer. Database access frameworks (Hibernate, Spring, ADO.NET Entity Framework) in the data access layer.

Blue print frameworks are plans and guidelines for building software. These frameworks consist of concepts, practices, rules and assumptions that describe the architecture and define the way developers will build software. They provide design rules and coding standards that determine the technologies and tools that developers use to build applications.

Why use a Framework?

Software development can be an expensive and imprecise undertaking. It is expensive because of the cost of the IT staff, hardware and software involved. It is imprecise because of the range of variables that influence development activities, such as the requirements definition and the experience and skills of the developers. Companies use frameworks as a way to reduce costs and introduce a higher level of predictability into software development.

Frameworks reduce the software development effort: Software applications consist of code related to the business logic and code related to the infrastructure that holds the application components together. When development projects include a framework, developers will spend less effort on the infrastructure and have more time to work on the business logic. ➔



Software frameworks are available for use in each application layer: user interface, business process & logic, and data access.

Frameworks provide generic services:

Frameworks provide the generic services needed by most applications. Examples are logging, configuration, database access and caching. They include tools and methods that simplify development for event handling, user interface management, data exchange and job processing. This infrastructure code is usually the most detailed and tedious code to write, requiring deep technical skills. Developers are more prone to making errors when working in this context. Using frameworks avoids or minimizes this situation.

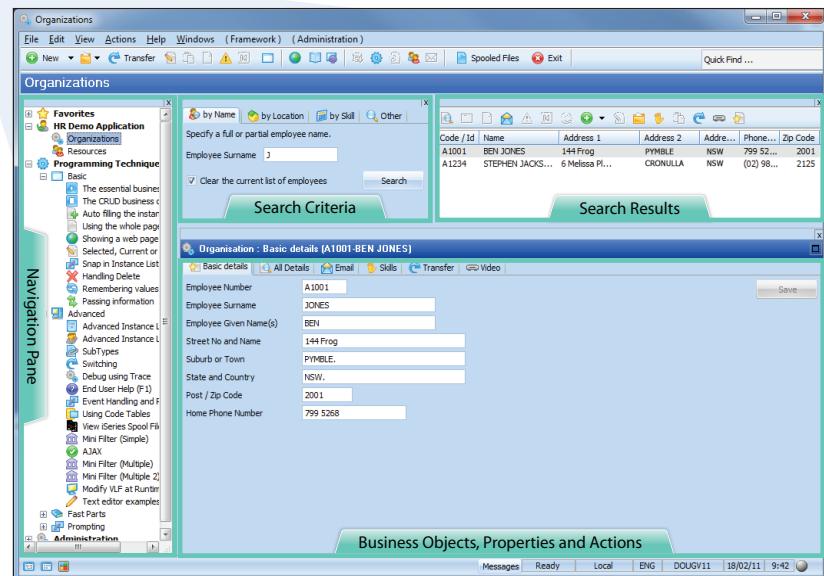
Frameworks reduce complexity: Many of the design decisions necessary when starting development from scratch are already built into the framework. The framework includes both the design decisions and the code to implement the decisions. Therefore, developers do not waste time reinventing the wheel. Less experienced developers will build better quality software by adding to a framework, rather than writing all the code. Better quality software reduces potential for errors and on-going maintenance effort.

Frameworks, like Visual LANSA Framework, encourage developers to work on component development. Components are small pieces of code that perform well defined and specific functions. They are less prone to error and easier to test. Components promote reuse, as a component can be part of several applications. Frameworks enhance componentization by encapsulating implementation details behind known and established interfaces. Component development improves software quality by localizing the impact of design and implementation changes.

Frameworks improve productivity, quality and consistency: Frameworks make the software development cycle a more predictable process by providing a standardized architecture and a standardized development approach. For example, developers work with pre-defined rules for coding and proven methodologies for building parts of an application. They also use a given set of development tools.

An architecture framework ensures that all developers work the same way to produce code that is understandable and maintainable. Developers produce software with a more consistent quality, while productivity improves because individual developers no longer need to make decisions about fundamental architecture and design issues.

Working with standardized and reusable designs facilitates automating development tasks. Frameworks often provide wizards and builders that generate code and reinforce the framework methodology. Wizards and builders improve productivity by reducing the effort to build an application.



The Visual LANSA Framework provides the infrastructure for an application including the security, user interface, search services, business objects management and navigation services.

Where do Frameworks fit in Application Development?

Frameworks provide a wide range of functionality and services. Frameworks, like Microsoft .NET, provide the basic computing services that developers rely on to build applications. At the other extreme, frameworks provide an entire application.

The following three level hierarchy illustrates the range of frameworks:

Software frameworks are the models, templates and libraries that we discussed earlier. They manage discrete components of applications. Developers use these frameworks to build applications.

Application frameworks are partially built applications or application shells. These frameworks are working applications without business code. Developers add the business code to the framework to complete the application.

The Visual LANSA Framework is an example of an application framework. Application frameworks may include the use of software frameworks.

Solution frameworks are completed and working applications. They include both business code and infrastructure code. These frameworks require configuration and integration, and may require a small development effort to add or customize specific business logic.

LANSA Commerce Edition is a solution framework for online ordering and inquiry. IBM offers industry solution frameworks, such as for the banking, government and insurance industries.

LANSA and the use of Frameworks

The Visual LANSA Framework (VLF) is an example of an application framework. It provides the infrastructure for an application including the security, user interface, search services, business objects management and navigation services.

The user interface includes a set of components common to most commercial style applications. The components are the navigation pane, search pane, search results pane and one or more tabs for viewing and/or maintaining business objects.

The VLF includes software frameworks, such as the code assistants. The code assistants are wizards that lead developers through the code generation process. By answering a set of questions and defining rules, developers generate most of the code. They can add to the generated code to complete the application.

The VLF also provides services for user management (who is allowed to use the application), authority management (who is allowed to view or edit business objects) and server management.

The VLF incorporates many design decisions, letting developers build applications with a consistent look and feel and a standardized architecture.

Applications developed with the VLF can be deployed as a client application or as a rich Internet application. The deployment options of the VLF again demonstrate how frameworks reduce cost and remove complexity from software development. ■