

Accelerate time-to-market through Framework-driven development

by Jothi Rengarajan

If converting ideas into software solutions was easy, there would be far more successful software products than there are in the world today. The bottom-line is that building software is a complex effort—and when you need to be early-to-market with a quality product, it is even more challenging. Framework-driven development is an approach that attempts...

Table of Contents

Abstract.....	2
1. Introduction.....	3
2. Definition of a Framework.....	3
3. Characteristics of a framework.....	4
4. Benefits of a Framework	4
5. Types of Framework.....	5
6. Components of an enterprise product framework.....	7
7. Role of the Enterprise Product Framework in Product Development.....	9
8. Conclusion.....	10
9.About Aspire Systems	10



Accelerate time-to-market through Framework-driven development

Abstract

If converting ideas into software solutions was easy, there would be far more successful software products than there are in the world today. The bottom-line is that building software is a complex effort – and when you need to be early-to-market with a quality product, it is even more challenging. Framework-driven development is an approach that attempts to increase developer productivity by making use of pre-built components and a solid underlying structure, which has already been tested.

Frameworks help maintain developer discipline, limit unnecessary re-invention of the wheel and provide features that are almost universal requirements. Enterprise product frameworks go one step further by providing a pre-built product architecture layer with many cross-concerns and design decisions addressed. This helps development teams focus on solving business issues and reduces development time by more than 30%.

This paper features a few such enterprise frameworks that will help software vendors (ISVs) enhance their product quality and crash their development time.



Accelerate time-to-market through Framework-driven development

1. Introduction

“How do I build a high quality product within a short span of time?”

This is a question which many ISVs (Independent Software Vendors) are trying to answer everyday. Building a software product has its own set of challenges. Unlike software applications that cater to a known set of users and is custom-built for this group of users, a software product typically caters to a much larger audience with a wide variety of users.

Many software products need to scale quickly and yet be stable and perform well. Products undergo substantial change over their lifetime and when new versions are released, the older versions still need to be available and maintained. Hence, products need to be highly extensible, maintainable, configurable, with a robust architecture and superior performance. All of this requires a team of developers doing things in a consistent way.

So what does it take to build a product with all these qualities in a compressed time frame?

A highly experienced product development team is vital to building a great product. And, almost every start-up or established software organization tries to either assemble one or out source to product development experts. However, getting a team that is mature and has worked together over a period of time is not always feasible due to various reasons. This is where frameworks can really help by providing a template for the entire team with pre-built components, and a solid foundation with cross-concerns and design decisions already addressed and tested.

As good frameworks are typically built by experienced architects and by themselves involve a huge amount of effort and investment, developers using the framework only need to follow the pattern and build specific functionalities on top of it. In any case, even if an experienced team is involved, there is no need to re-invent the wheel when there are available solutions/frameworks for common product requirements. Therefore, leveraging the appropriate enterprise framework for your development efforts can help crash your product roadmap substantially.

2. Definition of a Framework

What is a Framework?

There are several definitions available for this term. Some of the definitions that are closest to the current context are:

“A Framework is a semi-finished software (sub-) system intended to be instantiated. A framework defines the architecture for a family of (sub-) systems and provides the basic building blocks to create them. It also defines the parts of itself that must be adapted to achieve a specific functionality.”

“A Framework is an abstract design that embodies how the application works, and it has "hooks" where you can "inject" your module, or component.”

“A Framework is a set of pre-fabricated software building blocks that developers can use, extend, or customize to suit their application”



Accelerate time-to-market through Framework-driven development

To explain in much simpler terms, a framework is a package of components that facilitates software development by allowing developers to focus on product/business requirements rather than dealing with the standard low-level details such as logging, caching, authentication, etc. It provides a structure where an application must follow a certain flow and contracts and hence allows developers to reuse more than just code - the design and architecture reuse is important.

3. Characteristics of a framework

- **High flexibility and extensibility:** Framework should leverage on design patterns to achieve flexibility and provide a general-purpose application design.
- **High maintainability:** As a framework keeps evolving, the design and the code should be highly maintainable.
- **Easy and clear usage:** Framework level APIs should not have any ambiguity and should be very easy to use.
- **High performance and scalability:** One can never achieve high performance if the base is weak, and hence frameworks should offer high performance and scalability.
- **Testable:** The framework should be completely testable, both individually, as well as, as a whole.
- **Support integration:** The framework should have enough hooks to be integrated with external systems.

4. Benefits of a Framework

Quality benefits

- **Improved maintenance and orderly feature evolution:** When we add new features or capabilities to a framework, they work in all applications on the same system that are derived from that framework.
- **More consistent and modular code:** Code reuse provides consistency and common capabilities between programs, no matter who writes them. Frameworks also make it easier to break programs into small pieces.
- **More reliable and robust code:** Code inherited from a framework has already been tested and integrated with the rest of the framework.
- **Appropriate technology choices:** A framework has already made the necessary investments for the technology choice, limits choices for the developers and avoids incorrect decisions.

Productivity Benefits

- **Less code to write:** Much of a program's design and structure, as well as its code, already exist in the framework.
- **More focus on areas of expertise, less focus on areas of system compatibility:** Developers can concentrate on the specific handling and presentation requirements of the data in a particular problem domain. They don't have to be experts at advanced design concepts to solve system complexity which is well handled by the framework.



Accelerate time-to-market through Framework-driven development

Indirect Benefits

- **Generic solutions that can be reused for other, related problems:** Developers who are used to dealing with frameworks tend to think in terms of generic solutions rather than special solutions. They can even design their own programs as frameworks to be reused by themselves or others.
- **Improved integration of related programs:** As all programs based on a particular framework inherit not only common code but also common design, developers who are working on similar programs have a better chance of understanding and working with each other's code. Also, the programs they develop are more likely to work together consistently and reliably.

5. Types of Framework

Different types of frameworks are emerging to solve different problems such as Middleware framework (frameworks that are commonly used to integrate distributed applications and components, for example ESB and BPEL frameworks), web application framework ([software framework](#) that is designed to support [dynamic websites](#), for example Struts) and so on. From a product perspective, enterprise application frameworks and enterprise product frameworks are catching up.

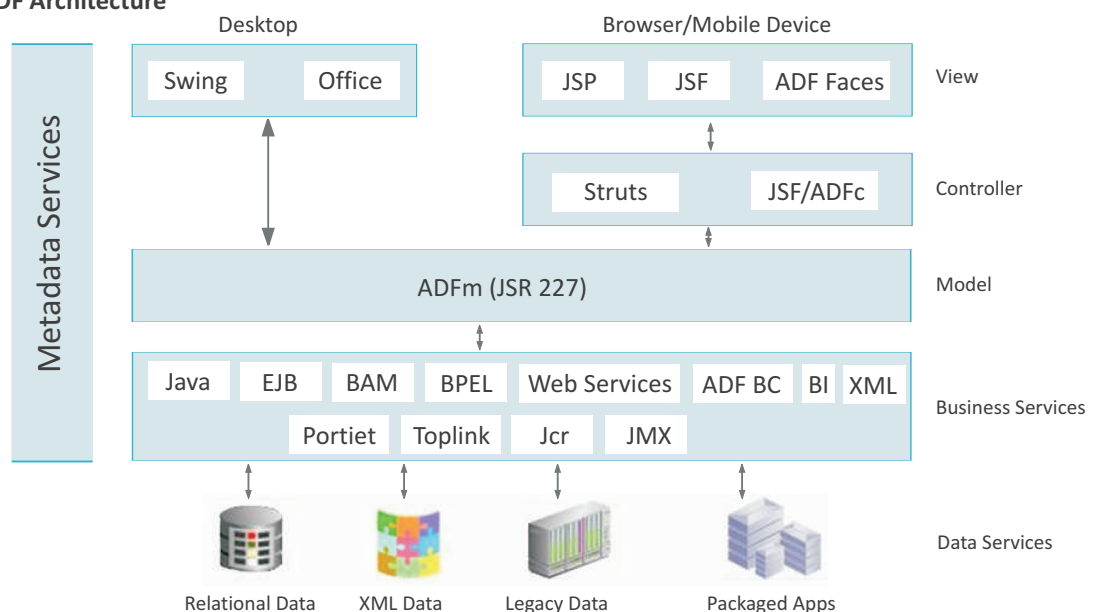
Enterprise Application Frameworks

These frameworks address broad application domains and they are the cornerstone of enterprise business activities. Every enterprise application requires plumbing services such as security, access control mechanism, logging, automated testing and so on. Enterprise Application frameworks package these common services into a core framework that is tested once and reused across many projects.

Oracle ADF: Oracle Application Development Framework (Oracle ADF) is an innovative, yet mature Java Enterprise application development framework available from Oracle. This simplifies Java Enterprise application development by minimizing the need to write code that implements the application's infrastructure allowing the users to focus on the features of the actual application.

Oracle ADF provides these infrastructure implementations as part of the framework. To recognize a set of runtime services is not enough, Oracle ADF is also focused on the development experience to provide a visual and declarative approach to Java EE development through the OracleJ Developer 11g development tool.

Oracle ADF Architecture

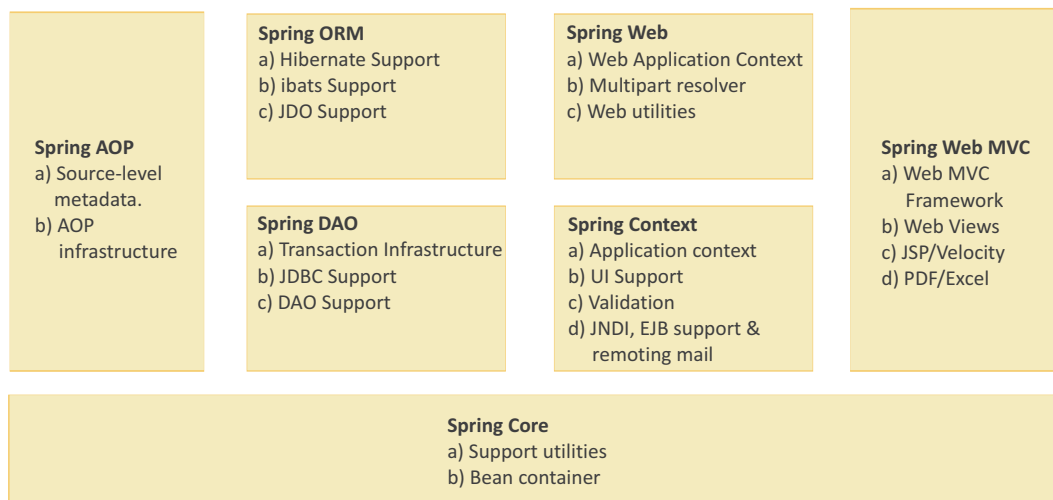




Accelerate time-to-market through Framework-driven development

Spring

Spring provides a light-weight solution for building enterprise-ready applications, while still supporting the possibility of using declarative transaction management, remote access to your logic using RMI or webservices, mailing facilities and various options in persisting your data to a database. Spring provides an MVC framework, transparent ways of integrating AOP into your software and a well-structured exception hierarchy including automatic mapping from proprietary exception hierarchies.



Enterprise Product Framework

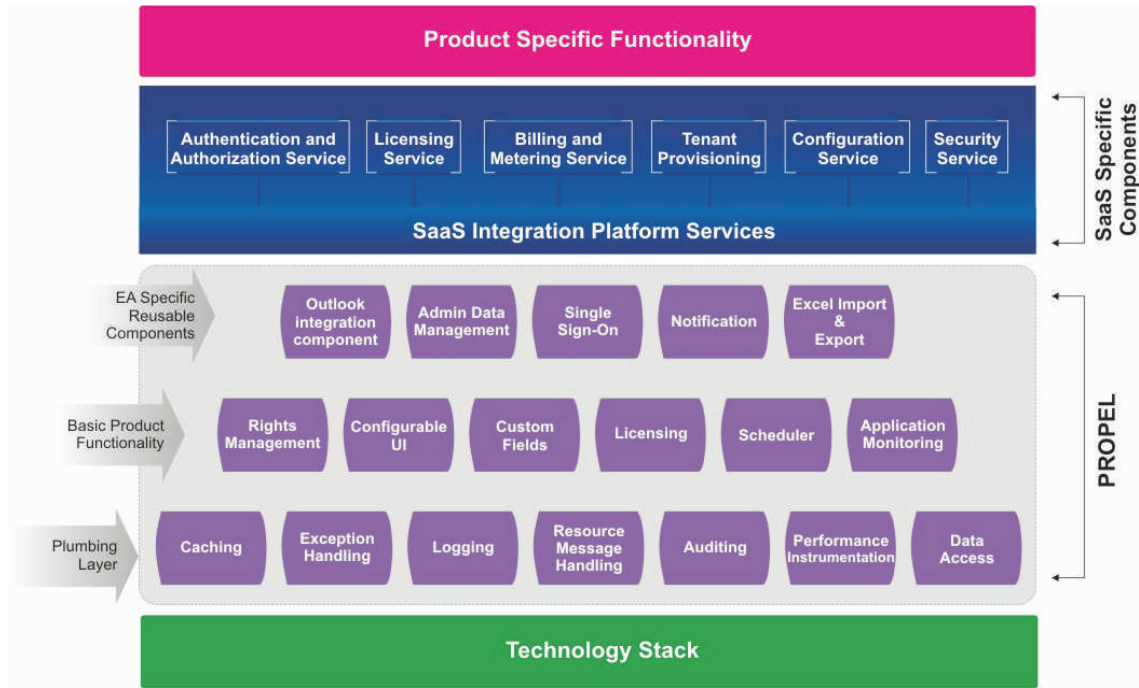
While there are many application frameworks out there, not many fulfill the requirements of an enterprise product such as configurability, extensibility etc. They act more as a general purpose, Rapid application development (RAD) tool and typically address problems that are one level lower than those required by products. For example, every product would need UI configurability, permission management, etc. While these are generic product requirements, they are not addressed in enterprise application frameworks. These are some of the features required by any product and developers tend to spend time in building these essential features. Enterprise Product Frameworks address such generic product requirements.

Propel is one such enterprise product framework, built by Aspire Systems, a pioneer in Outsourced Product Development, with over 12 years of product-building expertise. Propel is part of the technology Platform that forms a part of Producteering™ (Aspire's unique approach to building products better and faster). It is built on the base of proven application frameworks and adds more sophisticated services on top of it which is a common need for all enterprise class products. It is available in 2 flavors - one for On-Demand products and another for in-premise products. Each of these products have both the Java and .NET versions.



Accelerate time-to-market through Framework-driven development

6. Components of an enterprise product framework



Plumbing Stack

The components in these stacks form the base layer for building the entire product.

<p>Exception Management</p> <p>Robust and consistent mechanism to handle exceptions in a product. This has: Ability to configure and extend the handlers Ability to rethrow an exception Ability to prioritize exceptions</p>	<p>Logging</p> <p>Provides the base for logging throughout the product. This feature is very helpful after you go-live for production monitoring. Ability to categorize Extensible and configurable Distributed</p>	<p>caching</p> <p>Robust mechanism to cache data and improve user experience. Ability to add, delete and refresh cache Configure the cache storage Refresh cache based on dependencies</p>
<p>Data Access Layer</p> <p>Strong data access layer helps the developers work with multiple databases. Can work with multiple databases – MS SQLServer, Oracle, DB2, MySQL, etc. Easily configurable</p>	<p>Audit Framework</p> <p>Provides a robust framework to audit the important actions across all the layers. The level of the audit can be configured.</p>	<p>Performance Instrumentation</p> <p>This component enables a developer to instrument performance data in the product Enables accurate capturing of performance data Logs the execution time for each method if configured</p>
<p>Performance Instrumentation(cont..)</p> <p>Installs configured performance instrumentation counters Enables manipulating performance counters</p>	<p>Resource Message Handling</p> <p>Provides a robust framework to configure the resource text in the product. Has the facility to configure the messages in different data stores such as database, XML file or resource file.</p>	



Accelerate time-to-market through Framework-driven development

Product Functionality Stack

<p>Rights/Security Management</p> <p>Transmits user and object identity across all layers in an easy-to-access manner. Extensible and configurable role provider Extensible and configurable rights provider Ability to handle multiple roles for a user Ability to configure role level access and permission level access.</p>	<p>Configurable UI</p> <p>Provides robust framework for configuring the behavior of user interface on the fly based on the rights of an individual. Has the ability to turn on/off the fields or make the fields editable/ non editable based on the rights</p>	<p>Custom Fields</p> <p>Provides a robust framework to add, update and delete custom informational column to the data entities on the fly without code level changes.</p>
<p>Licensing</p> <p>This component takes care of the licensing aspects of the product. It provides license based on named user concept. Upgrading or downgrading the number of user licenses is very easy using this framework.</p>	<p>Scheduler</p> <p>Enables developers to run jobs in the background thread. The schedule timing and the jobs can be easily set up and the status of the jobs can be tracked.</p>	<p>Application Monitoring</p> <p>Enables the end user to analyze some statistics of the product such as slow response pages, widely accessed pages etc.</p>

Enterprise Application Specific Reusable Components

<p>Single Sign-On</p> <p>Provides the ability to integrate with other products with a single sign-on. Has services for integrating with active directory.</p>	<p>UI Controls</p> <p>Prebuilt UI controls such as Captcha Control, Ajax controls and so on</p>	<p>Notification System</p> <p>Provides a robust framework to configure and handle notification messages in a product. The notification data provider and handlers for the notification message can be configured and extended.</p>
<p>Admin Data Management</p> <p>Provides the developer the ability to create admin data screens just by using XML configurations. Has the ability to work with relationship fields</p>	<p>CSV Export and Import Component</p> <p>Provides the ability to read CSV files and write to it very easily</p>	<p>Outlook integration component</p> <p>Provides the ability to synch the data from outlook to web bi directionally.</p>



Accelerate time-to-market through Framework-driven development

SAAS Stack Components

This stack is available only for the SaaS flavor of Propel

<p>Authentication and Authorization Service</p> <p>Provides the ability to authenticate using a federated authentication. Provides each tenant the ability to configure their ACL policy for menus, pages, business actions, fields and data.</p>	<p>Licensing Service</p> <p>Provides the product administrator the ability to configure different kinds of flexible licensing models for tenants</p>	<p>Billing and Metering Service</p> <p>Provides the ability to configure different feeds for metering, and generate a billing report out of it for each tenant based on the licensing model</p>
<p>Tenant Provisioning</p> <p>Provides the product administrator the ability to add different tenants to the system with their basic configurations.</p>	<p>Configuration Service</p> <p>Provides each tenant the ability to configure their UI Provides each tenant the ability to configure Business rules Provides each tenant the ability to configure the workflow.</p>	<p>Security Service</p> <p>Provides each tenant the ability to configure their own mechanism for data encryption.</p>
<p>Integration Platform</p> <p>Propel SaaS is built using a complete SOA architecture which makes it easier for the product to be integrated</p>		

7. Role of the Enterprise Product Framework in Product Development

Developers can use these frameworks as product / project templates. When you use these frameworks to start your product development, the plumbing layers and other enterprise specific features are immediately available. That way, developers can focus building the business features required by the product rather than focusing on the engineering layer.

Typically, as these enterprise product frameworks are tested in different user conditions and for different kind of products, quality will be high. Also, during the product development, developers have to choose different tools, technologies and components. While having a lot of choice is good, it also brings in complexity. Frameworks have already invested significant time and effort in choosing the right technologies and tools required for building a robust product.



Accelerate time-to-market through Framework-driven development

Conclusion

Enterprise Frameworks have many advantages and if rightly adopted, can add a lot of value to your development efforts. It can crash product development time by nearly 30% and help you get-to-market faster. There are several ISVs who are benefiting from using and extending these frameworks. While increasing the productivity of the team, Enterprise Product Frameworks also bring in a very high quality of the product being developed. And that's a win-win situation!

ABOUT ASPIRE SYSTEMS

Aspire Systems is an Outsourced Product Development firm committed to helping our customers build software products better and faster. We work with some of the world's most innovative Independent Software Vendors and software-enabled businesses, ranging from start-ups to established industry leaders, transforming the way software is built.

Aspire provides complete product lifecycle services, ranging from new product development and product advancement to product migration, re-engineering, sustenance and support. Our product development teams are spread between our Global Innovation Center in Chennai, India and offices in the United States.



Transforming Product Development

Aspire Systems India Private Limited

Plot No 1/D-1, SIPCOT IT PARK, Siruseri,

Tamil Nadu - 603 103

Tel : +91-44-67404000. Fax: +91-44-67404234

E-mail : info@aspresys.com

Web: www.aspiresys.com

Aspire Systems, Inc.

4010 Moorpark Avenue, Suite 110,

San Jose, CA - 95117, USA

Tel: +1-408-260-2076

Fax: +1-408-904-4591

E-mail : info@aspresys.com