



Why choose .NET? Business and IT Application Advantages

Palm Beach Software Design utilizes the .NET platform from Microsoft® for many reasons, which will be detailed throughout the course of this document. However, the most important reason is that .Net was built specifically as a robust cost efficient development platform for business applications. It is ideal for:

- Accounting (A/R, A/P, G/L) applications
- Customer Relationship Management
- Inventory Management
- Internet Partner Integration
- Mobile Applications (hand-held devices)
- Production Applications
- Resource Management (Supply)
- Web Services and Web Applications

Exactly what is the .NET Framework?

In 2002, Microsoft released a platform for development that was written from scratch to solve some common issues in application development. Prior to 2002, application development timeframes were often long and therefore, expensive. Further, applications did not deploy easily across different hardware and it was difficult to change things within them without significant re-write. In the past, these factors negatively affected the cost of custom software ownership. The creation of .NET changed the game, making it possible for reasonable development times and ROI for all enterprise and many mid-market and small business organizations. For a technical overview of exactly what .NET entails, we reference the official Microsoft® website:

“The .NET Framework is a managed execution environment that provides a variety of services to its running applications. It consists of two major components: the common language runtime (CLR), which is the execution engine that handles running applications; and the .NET Framework Class Library, which provides a library of tested, reusable code that developers can call from their own applications. The services that the .NET Framework provides to running applications include the following:

- Memory management. In many programming languages, programmers are responsible for allocating and releasing memory and for handling object lifetimes. In .NET Framework applications, the CLR provides these services on behalf of the application.
- A common type system. In traditional programming languages, basic types are defined by the compiler, which complicates cross-language interoperability. In the .NET Framework, basic types are defined by the .NET Framework type system and are common to all languages that target the .NET Framework.
- An extensive class library. Instead of having to write vast amounts of code to handle common low-level programming operations, programmers can use a readily accessible library of types and their members from the .NET Framework Class Library.
- Development frameworks and technologies. The .NET Framework includes libraries for specific areas of application development, such as ASP.NET for web applications, ADO.NET for data access, and Windows Communication Foundation for service-oriented applications.
- Language interoperability. Language compilers that target the .NET Framework emit an intermediate code named Common Intermediate Language (CIL), which, in turn, is compiled at run time by the common language runtime. With this feature, routines written in one language are accessible to other languages, and programmers can focus on creating applications in their preferred language or languages.
- Version compatibility. With rare exceptions, applications that are developed by using a particular version of the .NET Framework can run without modification on a later version.
- Side-by-side execution. The .NET Framework helps resolve version conflicts by allowing multiple versions of the common language runtime to exist on the same computer. This means that multiple versions of applications can also coexist, and that an application can run on the version of the .NET Framework with which it was built.
- Multi-targeting. By targeting the .NET Framework Portable Class Library, developers can create assemblies that work on multiple .NET Framework platforms, such as Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Phone, and Xbox 360.”

(Quotation at <https://msdn.microsoft.com/library/hh425099%28v=vs.110%29.aspx>)

IT-Based Development Advantages

For business, understanding beyond the above-listed definition isn't really necessary because .Net framework applications are transparent to users. The important IT-based benefits to business organizations are myriad:

- ❖ The easiest advantage to discuss is security. You may know, from your IT people, as well as from recent news stories, how vulnerable every company is with regard to hackers. Attacks routinely come from all over the world even into smaller organizations. One of the advantages of the .NET framework is the increased level of application security offered to the business organization. Security elements were coded into the framework deliberately from the start; they were not added as an afterthought as is the case with

other development platforms. No computer system or application is “hack proof” but .NET based applications offer reasonably staunch protection.

- ❖ Modern web applications almost always include elements of responsive design, which means that they can be used on multiple platforms including desktops and tablets. Specifically designed Windows-based mobile applications are also increasingly used to handle day-to-day business in the field and they present their own complexities from a software development perspective. Another advantage of the .NET platform is the use of components which are reusable on different platforms without the requirement of individual code for that device platform. Error opportunities are reduced, the code is easier to manage, and the development cost savings are substantial.
- ❖ **Another cost saving efficiency embedded within the .NET platform is the ability to call upon and utilize existing templates and libraries within the framework as well as to build additional templates and libraries as required by the application. This substantially reduces the complexity of the application modules and speeds development. There are reusable class libraries which help developers be more productive and provide the ability to concentrate on the unique logic required within the application rather than “re-inventing the wheel” each time for standard tasks. The .NET framework also allows developers to create smaller, reusable services for communication, which provides outside affiliated vendors and suppliers efficiency in the way they contact you through the web and by phone or fax. Note: Palm Beach Software Design has developed in .NET in a wide variety of business applications over time, garnering large libraries of error-free re-usable code, which speeds development and reduces costs, thereby adding to client-realized ROI.**

ROI-Based Business Advantages

It stands to reason that all of the efficiencies in development phase listed above translate to substantial savings, and therefore, increased ROI resulting from initial investment as well as time to initial deployment. However, calculations of ROI must include both tangible benefits and intangible benefits. Estimating them in advance, however, isn't always as clear cut as executive management would prefer and the final ROI calculation cannot be made until the application has been working on the organization's behalf for a year.

There will be obvious tangible benefits including increased sales, reductions in inventory costs and customer service costs, etc., but there will be intangible benefits as well. Reducing time to market for a new product or higher customer retention due to increased service satisfaction may not even be conceived of until the new application suite is deployed, but they must be included when the final or ultimate ROI is calculated. In general, they will be harder to quantify before the application is deployed, however, should be calculated against the cost of the software investment by the end of the 12 month period following installation.

Another factor that deserves consideration within the ROI calculation, which almost always left out of initial ROI calculations, is the scalability factor. Fast growing mid-market companies are often in danger of quickly

outgrowing their software investment, but that is not a problem for .NET based applications because .NET was designed for scalability and larger loads are expected over time. Integration with Microsoft's Azure cloud services also contributes to scalability without the requirement for large hardware systems-based investment. Reduced maintenance costs should also be included as well-designed .NET applications by experienced .NET developers include easily modified configuration settings without code re-write. Deployment across different hardware without support issues and the reliability of the .NET framework contribute to ROI as well. A dollar figure must be assigned to each of these factors before the calculation of total ROI.

An accurate way to calculate ultimate ROI is the total of benefits derived from the software use in the first year divided by the cost of development. The payback period, derived from the total of benefits during the first year divided by 12 months applied to the cost of development, gives the number of months until the investment has been earned back by the organization.

For example, if the total calculated ROI is \$500,000, and the total development investment in the project is \$240,000, then the payback period is calculated as follows:

Total ROI first year: $500K/12=41,666$ ROI per month

Cost of development: $240K/41666 =5.76$ or project payback completed within 6 months

Conclusion

In conclusion, .NET is a reliable, secure, cost-efficient framework on which to build business applications, but you don't have to take our word for it:

".NET is a brilliant strategy that enables Microsoft to define the next shift in the software business. Gartner believes Microsoft is now providing more vision and influence regarding this shift than any other vendor." Gartner Group

Palm Beach Software Design programmers are expert in .NET development. We specialize in it because it is good business and it is good for business. .NET lets us reuse, extend, and scale, which we recognize as goals for good software, both for efficient development times, and general business benefits. Palm Beach Software Design has successfully implemented hundreds of business applications of all shapes and sizes since 2002.

About Palm Beach Software Design

Palm Beach Software Design is comprised of a small, tight team of software and business professionals dedicated to growing businesses up to \$75M by helping them to improve their potential by making operations more efficient, increasing sales and public impact, and modernizing for today's business climate using technology and software as a basis. We are process-driven, with high standards of excellence, and a dedicated staff. We have been in business for 30 years, and although we are a Florida-based company, we serve clients

throughout North America. Please contact us at 561-572-0233 and visit us on the web at www.palmbeachsoftware.com to learn more about how Palm Beach Software Design, Inc. can help your business get that competitive business advantage.