



Best Practices in Modern Application Development for Custom Business Software

Introduction

Palm Beach Software Design utilizes a modern hybrid “.Net Best Practice” strategy that places the individual productivity, delivery, and financial requirements of each client company as its highest priorities.

Waterfall Design

Traditional software design follows a waterfall strategy of completing a rigorous and time-consuming requirements analysis, followed by design, development, and testing phases. It is straight forward but does not always lend itself to today’s cultural need for instant gratification and seeing fully completed modules along the way. It relies on extensive planning and proactive preparation for a whole project and allows for parallel development. Parallel development means that multiple teams can work on individual modules at the same time, integrating them as the project progresses.

Agile Design

A newer agile model involves breaking the project into production-worthy modules, allowing for the entire programming team to complete one module at a time, and integrating them at the completion of each one, in a circular fashion. This allows client companies to see and show completed modules and “appreciate” individual modules as the project progresses. At the completion of the circle, the wheel starts around again, and the process begins again with the next module.

Strengths and Weaknesses

The biggest strength of the exclusive use of the proactive waterfall methodology is also its greatest weakness. The extensive planning for the scope and requirements of the project and the way work is completed requires patience and commitment on the part of the client. Work on several, or many modules, at one time may be partially completed, dependent upon factors from other modules. It may be difficult for the client to wait until well along the process for multiple modules to be in a state where full functionality can be shown. However, realistic delivery dates within budget are more common.

The biggest strength of the exclusive use of the agile methodology is also its greatest weakness. It is a reactive, rather than a proactive strategy, which allows both client companies and software designers alike to “design on the fly”. This means conceptual functions previously never conceived of can be added as the project progresses. At regularly scheduled progress meetings or scrums, as they are called, there is always a new game plan as both client and designers get creatively excited about adding new features and functionality. However, these new, conceptual features and functionality often adversely affect previously completed and tested modules, thus substantially overextending budgets and delivery dates.

The exclusive use of agile methodology may be best utilized as a strategy when the project is extremely large and complex, and/or the client company is in an uncertain business climate. Another scenario for the exclusive use of the agile method would include a company where the management, company direction, or needs, are constantly changing. This method allows for individual completed modules to join an asset library if the project is substantially altered or scrapped during the development year. While the agile method is acceptable for extremely large business entities that have the resources to absorb cost overruns and delays, it can cripple a smaller enterprise or mid-market company with tight budget constraints and delivery timeframes.

Modern Hybrid Approach

Palm Beach Software Design's hybrid ".Net Best Practice" strategy combines the best of traditional waterfall methods with the most important iterative aspects of agile design, providing for high-quality custom application development, delivered on-time, within a contracted budget.

Best Practices in Modern Hybrid Software Design and Development

1. Discovery Phase

In the beginning of the project, software design companies utilizing modern hybrid best practices take the time to learn about the current workflow and data flow of the client company's business and machine interfaces, which can be referred to as the Discovery Phase. These best practices include:

- **Discussion:** The full scope and requirements of a project are discussed to establish expectations of function. This is a complete description of what the software will do, not how it is accomplished.
- **Analysis:** A full analysis of existing business processes to define and document the scope and requirements for each major sector or module of a project is undertaken. This involves personal interaction with key employees directly responsible for working with particular modules every day. This ground level analysis often uncovers issues of which upper management is unaware or has not taken into consideration.
- **Investigation:** Determination and recommendation of the optimal technologies and languages required by the application to best meet the long-term goals of the client company is undertaken. Determination of existing component architecture to speed development and reduce costs is included.
- **Scope and Requirements:** The creation and client approval of a robust scope and requirements document to memorialize requirements, expectations, and technology-use plan is carried out.
 - **Note:** No scope and requirements document can ever completely encompass every single requirement for an unlimited time period because the software development process is dynamic and the client's requirements likely evolve over time. However, the more complete the scope and requirements document, the greater the probability of appropriate functionality and on-time delivery within a set contract price. A robust scope and requirements document vastly reduces the possibility of large cost overruns due to impactful change orders during the implementation phase.
- **Project Plan/Timeline/Contract:** Includes the creation and client approval of a detailed project plan and timeline with milestones, along with the full development or implementation contract.

Palm Beach Software's success rate, over thirty years of software development, is based on a process that minimizes mistakes and expensive rewrites because we model the data and logic with a complete understanding of the client's requirements.

2. Design Phase

When the scope and requirements process and project specifications are complete, the design phase begins. Best practices include:

- **Visual Modeling:** The mockup creation of screens, forms, and reports, including look and feel. This process involves multiple meetings to discuss placement of items on the screens, the business intelligence required to increase productivity and reduce quality decision time-frames, as well as the way the user experience is developed.
- **UX Design:** User experience is designed. Best practices require that a smooth-flowing user experience and responsive design with as much reduction in complexity as possible. It includes the capture of the structure and behavior of user experience and the hiding or exposure of details appropriate to the task.
- **Data Modeling:** Data is structured for current requirements, with the ability to expand and encompass future additions and changes to products and services as growth occurs.

Upon completion and approval of visual models and UX design by the client, the implementation phase is entered.

3. Implementation Phase

Implementation is the writing of code. Best practices include:

- **Iterative Milestone Approval:** All required changes or fixes to meet client objectives are discussed and approved at each defined milestone; this provides for short-term focus. The impact of iterative change requests is assessed and discussed. Approved change requests are prioritized by client management for the current iteration or planned for a second phase of the software after initial release. Additional cost may or may not be incurred. Without explicit client management control of change requests, development can degrade to chaos and project delivery is substantially delayed. Client understanding of change request delivery timeframes is crucial for on-time delivery within contract price.
- **Software Personnel**
 - **Project Manager and Software Architect:** Their job is to manage the project, and meet with the client regularly to refine the understanding of how the business runs, so the software always meets and/or exceeds expectations. The project manager demonstrates the current work that has been accomplished and discusses any new issues, resolved items, functionality changes, and workflow changes (changes in the way the client does business, if any) in the implementation phase.
 - **Technical Team Lead:** A seasoned software developer leads the team for each sector or module of the project. The Team Lead makes decisions on technology and coaches other developers while contributing as the lead programmer.

- **Quality Assurance:** Includes a continuous repetitive testing and debugging loop for responding to bugs and necessary logic changes within the implementation. Either module or integrated full system testing is performed dependent upon the percentage of completed work within the implementation phase.

Although testing has been ongoing, upon completion of the implementation phase, full system quality verification begins.

4. Software Quality Verification

Best practices defines quality software as an application or product which meets or exceeds client-approved specifications in the scope and requirements document by objective measures, including:

- “3 Pass” Testing
 - **Functionality:** Follows scope and requirements specifications
 - **Errors and Exceptions:** No errors or cryptic messages.
 - **User Experience (UX):** Ease of use, look and feel

Note: All work should be posted on secured staging servers for easy testing and monitoring.

5. Delivery and Training

Best practices include:

- On-time or agreed to delivery based on approved change requests
- Initial delivery presentation
- Client-performed software quality verification before the software goes live.
- Agreed to initial client personnel training just prior to “go live” date.
- Installation training support during 1st 30 days after “go live” date.

About Palm Beach Software Design

Palm Beach Software Design is comprised of a small, tight team of software and business professionals dedicated to helping businesses up to \$75M in sales improve their growth 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 many of our clients are Florida based, but we serve clients as far away as New York and California.

Please contact us at 561-572-0233 and visit us on the web at www.palmbeachsoftware.com.