

This Free E-Book is brought to you by Natural-Aging.com.

100% Effective Natural Hormone Treatment
Menopause, Andropause And Other Hormone Imbalances
Impair Healthy Healing In People Over The Age Of 30!

Effective Software Development Process

By Blane Warrene

Effective Software Development Process by Blane Warrene

There is never a guarantee of project success when endeavoring to build a sophisticated application. However, there are established steps to follow that will ensure a clear, concise scope, support for the team involved, and a solid opportunity for successful deployment.

Previous experience has shown many project and technical teams that the keys to failure can be as follows:

- * Lack of senior management support and business planning.
- * Lack of clear and detailed design specifications.
- * Lack of functional specifications which read as a handbook reference for the technical team involved to start writing code.

Writing Your Declaration of Dependence

A project always starts as an idea which generally provides a solution. Often these are drummed up in brainstorming sessions and scratched out onto legal pads or napkins. It is this step which leads to your project charter, or your declaration of dependence.

The key in this phase is to include, from the start, all parties involved, including the most senior managers impacted by this effort. This means the technical team, development manager, project manager, business representatives from the impacted departments, and preferably, the CIO and a non-technical senior management representative.

By utilizing this group in developing the charter, there is unity in the business plan mission for the project from the beginning. This is also the opportunity to better understand the lifecycle on the business side for the developers and for the developers to articulate the process they go through to build the solution. This provides a forum for business users to discuss the processes they use or would use in this application, giving some insight, perhaps, into interface design and application flow. Additionally, the technical team can present the process for moving from design specification to functional specifications to a test-ready application.

Effective Software Development Process

A process should also be agreed upon for ongoing status reporting and future resources which will be needed, such as quality assurance staff and alpha/beta testers. Depending upon the timeline, these resources may need to be identified and notified at this stage for budgeting and scheduling purposes.

This completed document defines the entire scope of your project, its mission statement, supporters, reporting processes and the ultimate end result with broad timelines. It also clearly reflects the interdependencies required to successfully complete the initiative.

Business Requirements

This standard accepted process needs little explanation. However, it is the step which will enable the technical team to develop the design and functional specifications we will discuss later.

Most important is to ensure this document is written in user-friendly language and format. This is also

the document to build your projects glossary and definitions document. It enables a final review of the scope by the team that built the Declaration of Dependence to ensure all aspects are covered before the process moves forward to design specifications.

It is also important to avoid including design elements in this document. For example, if a data mart will be built and accessed via this application, it is important to define the definitions of a record, but not to define field structures, data types and naming conventions.

This document will produce the necessary definitions of business processes and needs to identify hardware and software specs, as well as components and elements needed inside of the design specification.

Design Specifications

Design specifications are the "meat and potatoes" of the project for the technical team. This is where the major system goals will be established and will very likely sound similar to the mission statement implemented in the Declaration of Dependence or the charter. Several reviews of the charter and business requirements by the technical team will lead to an introductory high-level technical document which ties systems, components and modules, and database needs to business processes and tasks in the proposed application.

Top Level Design

This should be capable of technically describing and defining the application without necessarily specifying underlying language to be used. Additionally, this is where all challenges should be identified:

- * How is the overall application to be organized?
- * Are all systems and sub-systems clearly noted and defined?
- * Have all functions been defined for component development?
- * Have all data definitions been converted into data structures and types?
- * Are there existing systems which can be leveraged for some of this application, or is it 100 percent original development?
- * What will be built and what will be bought? Cost analysis of components required versus available commercial components.

Effective Software Development Process

- * Limitations of internal resources.
- * Interfaces and systems outside of internal control, which impacts future change management and application updates.
- * Does the architecture meet the standards of the organization and industry standards?
- * Scale of the system to handle the level of use and growth of systems such as data storage, new interfaces, future improvements and modules.

User Interface Design

This can be a web browser based GUI or a software GUI. Either way, consideration should be given to both design of the interface and discussion of how to handle changes to the interface so as not to disrupt underlying application development.

Database Design

This is the place to architect the top-level view of the database for this application. Depending upon the scale, perhaps some ETL (extraction, transformation and load) tools may be used for design purposes. This is an area where challenges and risks can be identified further, as mentioned earlier. This would include reliance upon data sources or interfaces outside of the control of the technical team.

Roles and Responsibilities

A final step is to assign resources to the varying sections of the whole application. This is where expertise and experience are married to the components and systems required to complete the project.

The completed design specifications will naturally lead to a functional specification. This makes it important to take an additional step prior to moving forward. This is a good time to bring the original charter group back together with a presentation that shows how the design of the system answers the business requirements and meets the overall scope of the charter. This discussion will most likely bring about some changes to the design, and the previously agreed upon change management process should be used to update the design specification and move on to the final stage prior to getting the programming started.

Functional Specification

The goal of a functional specification is to achieve a handbook reference for the technical team to develop, integrate, test and finally release a completed application. This document is much easier to complete with the previous road map documents in hand. The technical team will never lose sight of the business goals through the charter and requirements, and the design document has set the parameters through which the project will be accomplished. Also, the repeated steps of communicating back to the original group keeps the flow of information on status and issues between business and technical teams and support does not waver for the project.

Now is the time to address language specifics and how issues such as hardware and software dependencies and limitations will be addressed, performance requirements, security requirements, error handling and overall logging requirements.

The applications varying sections are addressed with the resource(s) in mind. This will enable a very

clear timeline to be drawn and matched against the original timeline from the charter and updates may need to be provided to the original group (keep in mind those QA and testers we mentioned) for scheduling. Again, be sure to use the change management process.

The completed functional specification will read like a technical reference guide for the resources to construct the application. It will provide a timeline to be followed and reported upon. And ultimately, it should lead to successful testing and completion of the project.

Conclusion

Critical to the success of any endeavor is communicating issues and solutions, and the steps to achieve results. In application development, communication is very critical, not just between technical team members, but also between senior management and business users involved in the project, as well as your QA and testers who will sign-off prior to deployment.

The steps of any application project use the business charter, business requirements, design and functional specification to ensure scope management and that goals are met with the end result.

However, there can be a disconnect between these parties as the project is underway and the coding begins. This can lead to loss of momentum and support from non-technical staff and managers, scope creep and/or misunderstandings, which may lead to failure.

To avoid this, consider setting broad parameters for reporting status and general communications

when building the charter at the inception of the project. These can be scheduled meetings, e-mail updates and even presentations as needed. Also ensure your change management process is in the charter, and that it entails communicating change requests to this group with a clearly defined process for approvals and comments.

While some of the project will seem technical and difficult to follow by non-technical team members, the group remains informed, can feel they are invested in the success of the project, and can be your biggest cheerleaders when obstacles arise.

Prior to participating in the founding of BMRW & Associates, Blane served most recently as the Director of Technology for VESTAX Securities Corporation. In this capacity he was responsible for the management and evolution of the IT infrastructure and services for internal operations and field technology services for VESTAX financial advisors. Blane brings substantial knowledge in application & data integration, mining and management .

How To Develop Software For Your Business

By David Berube

Software development is a risky business.

Effective Software Development Process

Many software developers are barely skilled at their trade, much less at business. They can't understand how your business operates – and their code isn't unreliable.

It crashes constantly.

It isn't professional.

It hurts your business.

If you want software developed for your business, what can you do?

You need a software development professional – a consummate businessman, and you need to deal with him on your terms. How can you do that?

First, you need to make sure that you find a professional who understands business – your business. He has to listen, and be cautious before giving advice. If he tries to drop a pre-packaged solution on you, drop him. Make sure he charges by the project, not by the hour – hourly rates are great for developers, but not for clients. Hourly rates mean they get paid *more* for every bug they create – so they create as many as they can.

Once you have chosen a developer, the next step is to come to an agreement on what needs to be done. Don't worry about the particular technology used – the developer should worry about that. Worry about what you need on your end – and stick to your guns! You know your business best, and your developer should respect that.

After you have an agreement, you get a contract signed, and then the real fun begins. Insist that your developer give you regular progress updates – including early versions of the software (called 'builds'). That way, bugs can be detected early, and there will not be any surprises when the project is over. Depending on the project, you may be able to get a build every few days, or every week. The more frequent, the better.

When the project is finally over, you get to take a deep breath. Relax. Sip champagne – or Heineken, if that's your thing. Don't get too relaxed, though, because you will have to start the process over again soon. Business is constantly changing, and your software has to evolve with your business. That's good, though, because every project makes your business more profitable, and more profit in less time is always a good deal.

David Berube is a software developer and IT consultant solving business problems. He's also a prolific writer and speaker. If you'd like a powerful, innovative developer, check out

How To Develop Software For Your Business
A Time-Saving Programming Tactic That Doesn't Work

Project/Program Management Best Practices for Success in ANY Industry!
Make your own Computer Software – without Programming!
Comparing Data Recovery Software

Self Improvement PLR Kit
File Resource Meter Software
How to Use Your Mind for Study
Free Government Grants
Squeeze Machine



This Free E–Book has been brought to you by Natural-Aging.com.

100% Effective Natural Hormone Treatment
Menopause, Andropause And Other Hormone Imbalances
Impair Healthy Healing In People Over The Age Of 30!