



Custom Software Development Approach

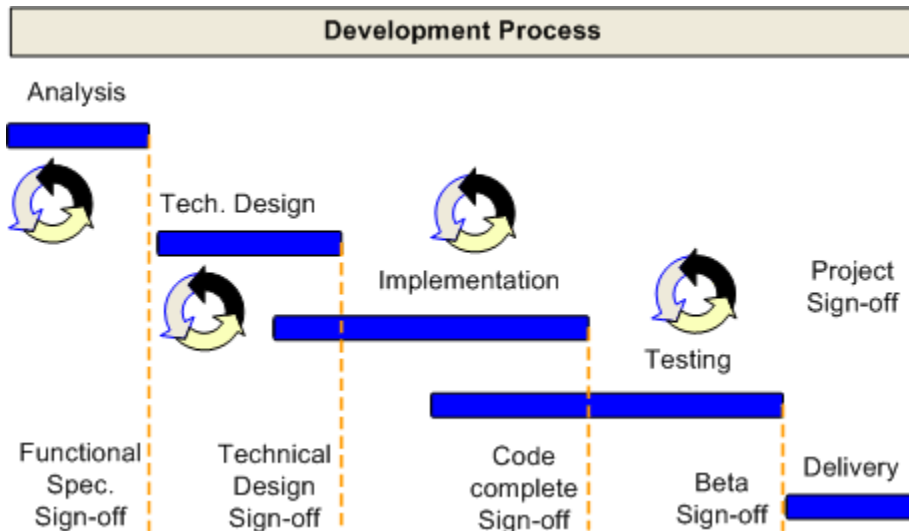
Our approach to custom software development combines benefits from several standard development process models. We tend to have a well-defined, predictable and highly manageable process while allowing for a degree of refinement, flexibility and results review after each major phase.

Phases

Each phase of the development process is structured in sub-phases, which may have iterations in turn. The results of each phase are provided to the client as artifacts in the form of documentation, diagrams and/or source code. We tend to clarify all requirements for the project upfront as much as possible, but we also arrange project plans for possible refinements and changes. After each phase we usually have sign-offs to ensure both the client's satisfaction with the current phase results and delivery of the next phase results on time and on budget.

In a typical project we have the following major phases:

- Discovery or Functional Analysis
- Technical Design
- Implementation
- Testing
- Delivery





Discovery or Functional Analysis

Our Business Analyst has a few sessions with our client's subject experts and users to define the purpose and possible usages of the future system. We define the boundaries of the system and create UI wireframes. We also document assumptions as well as business, security, and scalability requirements. After each such iteration, we refine our documentation based on the client's feedback and additional details. The revised document is provided to the client for review.

If functional specifications are provided to us, our Business Analyst and Application Architect will study them to make sure we have a solid and unambiguous understanding of the requirements and purpose of the project from both a business and technical point of view.

At this point in the project lifecycle, both major and minor functional changes are acceptable, with little or no additional cost to the whole project (if there is no significant change to the original scope of the project).

This phase results in a Functional Specification document, which is signed off by the client, and a project plan draft with the precision of +/- 25-30%.

Note: We encourage our clients to start working with future application users to ensure adequate understanding of the future application functionality, and to prepare for user acceptance testing after the beta version of the application is delivered.

Technical Design

Based on the results of the Discovery and Analysis phase, the Application Architect designs the architecture and conceptual model as well as other technical diagrams, such as the database model, class diagrams, and sequence diagrams. We also define business components, UI components, create a UI prototype, and deployment scenario for the application.

The Application Architect works closely with the Business Analyst to ensure that all requirements are considered and met in the technical design. At the client's request (if feasible), we can provide a choice of several technical design options to the client to choose from (this can be done only during the early stage of the technical design phase). The client's feedback is incorporated into the final version of the technical design.

At this point in the project lifecycle, both major and minor functional changes are acceptable, with little additional cost for the phase. Major changes may or may not result in overall project cost increase (if there is no significant change to the original scope of the project).

This phase results in a Technical Specification document with diagrams, which is signed off by the client, and a refined project plan with the precision of +/- 5-10%.



Implementation

Based on the results of the technical design, we start implementation of the project. We start with a project skeleton that contains all defined components and UI modules. These modules begin with an empty implementation and grow in time as we progress. As a result, the application skeleton becomes more robust each week, demonstrating project progress. All progress is tracked in the project plan and reported to the client on a regular basis.

By client request, this phase can be planned as a series of sub-phases with their own delivery dates and intermediate testing efforts (this option provides more assurance to the client and more control over the results of the project, but it will also add additional cost due to extra testing efforts after each sub-phase).

Regardless of the presence of sub-phases, we conduct daily unit-tests for all business components in the project, in order to ensure that they are not broken with additional changes made during implementation or bug fixes. We also begin the development of a test plan and early pre-test efforts immediately following the beginning of the UI module development to ensure a level of quality on a daily/weekly basis.

In many cases, to lower the overall project cost for the client, we use our offshore partners during the implementation and test phases, while keeping all of the knowledge regarding the project on-site in order to better serve our clients and ensure a level of quality in the project.

Both major and minor functional changes are acceptable, with additional cost for the whole project. Major changes are discouraged at this point in the project lifecycle.

The result of this phase is the functionally completed source code, which is ready to go through the Quality Assurance process.

Testing

We start testing efforts as soon as possible during the implementation phase. QA specialists start development of the test plan document based on the functional specification and technical design specifics. This test plan goes through a few iterations or evolves during the implementation phase.

The formal QA testing phase has at least 3 sub-phases (integration, functional, and system tests) which ensure the quality of each individual function as well as that of the entire system overall. All changes in the source code at the last sub-phase are controlled by the Application Architect and technical team leaders, and require their approval.

We provide the client with an opportunity for a final functional review before we start the formal QA phase. Minor changes only are acceptable at this point in the project life cycle to ensure delivery on time and on budget. We usually postpone all major changes until the next version because any major change can result in significant extra costs. This is due to the need to go through portions of all of the previous phases (design, implementation and tests) to implement such changes. However, major changes are still acceptable by client request.

The result of this phase is a beta version of the application, which is delivered to the client's staging environment and ready for user acceptance tests.

Delivery

We deliver the application in two steps:

- Beta version delivery to the staging environment for user acceptance tests
- Final delivery to the production environment.

We usually plan for a few weeks of user acceptance testing. At this time cosmetic changes only are acceptable. We are looking for final user feedback on the defects that were not reproducible or captured during the QA testing phase. Usually these are minor and cosmetic fixes.

We allocate 30-40% of the resources used in the project for this phase, as we don't expect a significant number of changes to be requested or defects to be reported.

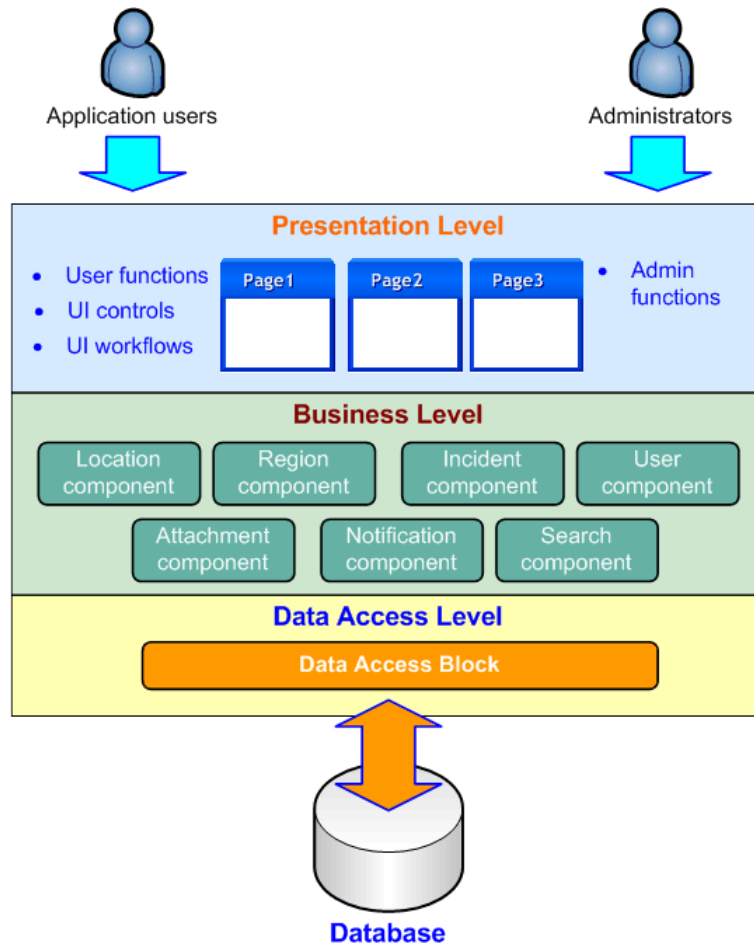
Depending on the technical details of the project, we design a deployment procedure at the time of the Technical Design phase, and then we test it regularly during the Implementation and Testing phases. In the beginning of the Delivery phase, we run these procedures to migrate data from old versions of the application (or other applications) to the new application database (if applicable) in the staging environment. This provides extra deployment testing with the most recent data to ensure a minimum of issues during final data migration in the production environment. This ensures that the client will have minimal downtime during the final application deployment.

A very limited number of minor changes are acceptable at this point in the project life cycle to ensure quality, as well as delivery on time and on budget. Remaining changes are postponed until the next version.

The result of this phase is the final version of the project, delivered to the client's production environment.

Architectural principles

We employ “best practices” in our application architectures and technical designs. In a typical application, we use a combination of both object-oriented and transaction-oriented approaches, while promoting N-tier architecture. Therefore, we create applications that are easily maintainable, extensible, and scalable, while providing excellent performance for individual user operations.



A typical application consists of three main tiers: *Presentation Tier*, *Business Logic Tier* and *Data Access Tier*.

The Presentation tier includes the components responsible for rendering the User Interface of the application, and supporting interaction between user and application.

The Business Logic tier functionality relates to the execution of business rules and business tasks of the application.

The Data Access tier has components designed for data retrieval/storage in the RDBMS, such as MS SQL Server.



Quality control

Quality control occurs in multiple stages. We conduct daily unit-tests to ensure a level of quality while developing individual components in the project. We conduct integration tests to ensure smooth and seamless integration of different parts of the application. We conduct preliminary QA tests during the implementation phase to ensure the level of quality on a daily and/or weekly basis. Therefore, our clients can access (by request) and see the actual progress of the project (beginning in the second half of the implementation phase. Finally, during the formal QA testing phase, we ensure that the project has no known defects and that any fix made in the code does not indirectly break the rest of the application.

If we are working with our offshore partners, we ensure that all development and QA testing is done based on our strict development rules, according to our coding style, and that all of the code is delivered to us on a daily basis. In addition to the usage of offshore testers, we use on-site QA testers in order to have a second line of quality control, and to make sure that the application meets business, functional and technical requirements.

Sign-offs

After each phase in the project lifecycle, we deliver all artifacts of the phase in the form of documents and/or source code. In order to ensure that the client is satisfied with the results of each phase, we ask them to review the results, provide feedback, and/or sign off after all feedback is incorporated. These regular results reviews and sign offs allow our clients to have more control over the overall quality, as well as more control over the functional changes made during the project lifecycle.

If functional changes are requested, sign offs help to ensure the delivery of the next phase results on time and on budget. Based on the requested changes, the plan for the next phase and/or overall project plan may be refined to accommodate the changes, and may result in additional costs. Such project plan changes are presented to the client for approval and sign off as part of the phase results sign off.

We usually expect the client to provide feedback within 3-5 days. The actual time frame can be agreed upon according to the client's schedule.

Importance of timely cooperation

We understand the importance of delivering your project with high quality, on time and on budget. We do our best to meet all date agreements and provide our clients with all results on time. We also depend on our client's schedule and ability to provide feedback in timely manner. We appreciate the provision of any feedback within 3-5 days.



This helps us to keep overall project costs low, and also to keep moving forward with the implementation of subsequent phases.

We ask our clients to let us know about any possible delays regarding feedback in advance so that we may arrange for any possible imposed changes in dates and/or costs.

Change requests

After the Discovery and Functional Analysis phase is complete, any functional changes requested can be incorporated into the functional specification document as an amendment or appendix. We provide the client with a change request form upon request.

Because new functionality has not been estimated and incorporated into the initial project plan, the estimate of such functional changes will be provided to the client for approval and/or prioritization as part of the current phase sign off process.

Staging environment and UAT

At the final phase of the project lifecycle (and before final delivery of the project), we provide the client with a beta version of the application. We recommend having a formal user acceptance testing phase, where users can try this beta version to make sure that it meets their expectations, based upon the functional specification and all signed off change requests.

Equipment and licenses

The UAT phase requires both human and hardware/software resources to be provided by the client. We assume that our clients will arrange for the necessary software licenses and hardware, as well as provide the human resources needed to conduct user acceptance tests for the duration of the UAT phase.

UAT process

We provide clients with an account in our defect tracking system, and fix any defects logged in the system during this phase. We also evaluate all requested changes, to make sure that the change can be incorporated into the project without additional costs or changes in the project schedule. All other changes are kept in our defect tracking system and can be incorporated into the next version of the application if desired.

Deployment to production environment

Production deployment is done after all reported defects are fixed and user feedback is addressed.



Code deployment

Depending on the technical details of each project, we may arrange for the deployment procedure during earlier phases of the project. We also perform regular deployment tests as part of our quality assurance efforts throughout the project lifecycle.

Data migration

We request a snapshot of the old versions of the database(s) with the actual data at the time of Technical Design to ensure we have addressed all possible data migration issues in deployment procedures. We test all such procedures during the Implementation and Testing phases.

In the beginning of the Delivery phase, we request another snapshot of the old database, and run procedures to migrate data from the old database(s) to the new application database in the staging environment. This provides extra deployment testing on the most recent data to ensure a minimum of issues during final data migration in the production environment. This also ensures that the client will have minimal downtime during final application deployment.

Final data migration is scheduled during off hours and is done with all possible precaution, including old data backups and deployment rollback if there are any issues.

Post deployment support

We will fix all reported defects within 1 month of application deployment to the production environment. We expect any such defects to be logged and properly documented into our defect tracking system.