



CONTACT
DESIGNS

DB
Dun & Bradstreet

arizona
AZIPA
Arizona Internet Professionals Association

Scottsdale
Chamber of Commerce

BBB Better Business Bureau
Fairness in the Marketplace

Greater Phoenix Chamber of Commerce

Introduction

Every software development effort goes through a lifecycle, a process that includes all activities in the development cycle that take place up to initial release. The main function of a lifecycle model is to establish the order in which a project specifies, implements, tests, and performs its activities. The appropriate process can streamline your project and help ensure that each step moves you closer toward your goal.

Some of the more popular software lifecycles are the traditional waterfall, the spiral or rapid application development (RAD) model, the [Rational Unified Process](#)[™] method, and [extreme programming](#) (XP). Our experience in the industry has led us to adopt a synthesis of these methodologies by borrowing the best practices from each as they apply to Internet software development.

Each project presents unique challenges specific to its environment. Therefore, the process we follow to achieve the goals set forth by each project can differ widely. The purpose of this paper is not to specify a process that can be used to accommodate all software projects, but rather to illustrate the process that we use on most projects where the requirements are not dynamic and the initial release is fully quantifiable. This process has seven phases, each of which are covered in more detail below: (1) Discovery, (2) Estimate, (3) Design, (4) Confirm Design and Estimate, (5) Construction, (6) Testing, and (7) Implementation.

Process

1. Discovery

In this stage, our software engineers and consultants will work with the customer to investigate the requirements, define the scope of the project, and formulate an overall schedule. Primary actions carried out in this stage include:

- Investigate the business requirements
- Forecast the infrastructure needs of the solutions given the organization's present and future needs
- Preliminary analysis of the performance, security, integration, and maintainability issues
- Identify the schedule for development and implementation
- Finally, a requirements document is authored.

2. Estimate

Once the requirements document is complete, we are able to estimate the time it will take to design the application and create the design document.

3. Design

Using the information gathered during discovery, we formulate an initial design that generally consists of a database schema (if a relational database is involved), [UML](#) and site information flow diagrams, a formal schedule with concrete milestones, and a list of functional and documentation requirements. During the design process, the project is examined at an atomic level and a blueprint of each functional part is created. The document produced will include every screen, every report, every data import process, etc. with complete details on how each part will be created, and the interrelationships between the parts. The design is created to stand by itself, in that any developer with the appropriate skills should be able to construct the application to spec from the design document. The design process addresses the requirements of the project. It is during

the design phase that decisions are made on how best to apply technology in order to satisfy the goals of the project. Where the discovery phase provides a description of what is needed, the design phase provides a description of how to implement the requirements. The design process includes making decisions on what is the most appropriate programming platform and server platform for the application. Often a mix of technologies is appropriate to use in a given application. Generally, every attempt is made to leverage appropriate technologies to provide for rapid development time and keep costs to a minimum.

For projects that are expected to have several versions following the initial release, or are medium to large in scope, a version release schedule is included that specifies which known functional requirements will be included in which releases and the estimated timelines involved. Versioned releases enable the project team to respond to continuous changes in scope, schedule, and project risk. By frequently updating the product, not only does the development team communicate with the customer, but suggestions for future releases of the product come directly from customer's use of the product. The team delivers a core set of features in its first release and adds features incrementally in later releases until it achieves the full vision for the product. Later versions allow the team to revalidate or update product vision as business requirements change.

4. Confirm Design and Estimate

Our consultants will work with the customer to verify that the functional requirements meet their needs, describe the design and its impact on scalability and integration issues, and summarize the implementation process. Once the application is designed, it is possible to estimate with some accuracy the hours it will take to construct the project. Only when all of the details of the project are known is it possible to come up with a realistic estimate for any sophisticated programming project.

5. Construction

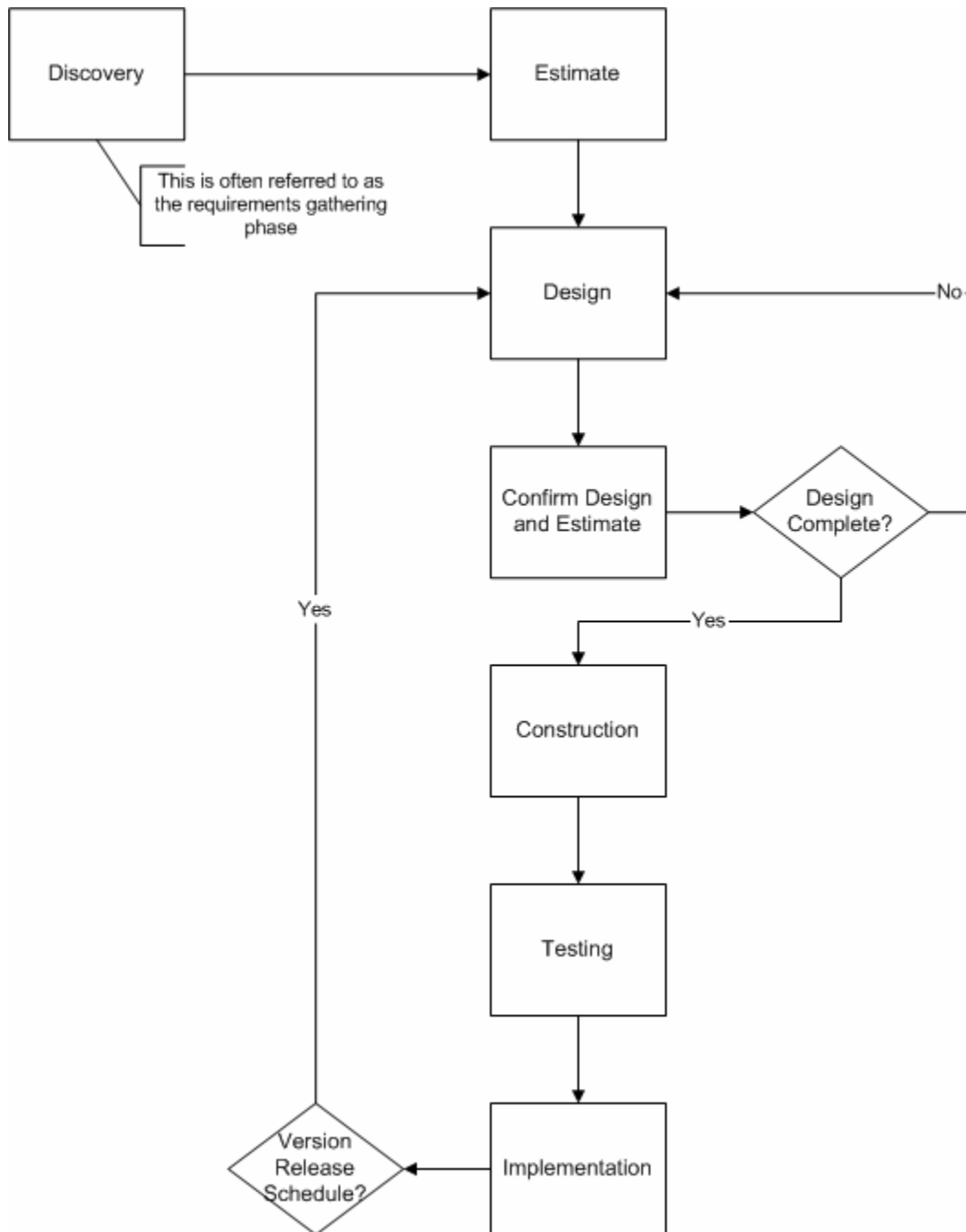
The construction phase is where the functional parts outlined in the design step are broken down into programming tasks. Our software engineers construct the design on our development servers where project managers can continually gauge progress and make adjustments as needed to meet milestones.

6. Testing

Once the project is near completion, overall system testing is done to make every assurance that the application is free of bugs and is running optimally. During the testing phase, code is profiled and rigorously load tested to verify it conforms to requirements.

7. Implementation

During the implementation phase, the application is deployed to the production environment and tested. Depending on the project, implementation may happen in phases so that a subset of features may be available for review throughout the construction phase – this is generally the case for large projects that have a version release schedule. Frequently, static interface design elements are available prior to the completion of back-end programming, which allows for a review and refinement of the visual design concurrent with the back-end construction.



Factors That Affect Projected Delivery

Our experience has shown us that certain environmental factors can prolong milestones. In order to help prevent this from happening, it is important to note factors that contribute to this environment:

- Project scope is broad where the business objectives are obscure or broad.
- Project data is complex and voluminous data must be analyzed, designed and created within the scope of the project.
- Many people must be involved in the decisions on the project, the decision makers are not available on a timely basis or they are geographically dispersed.
- The technical architecture is unclear and much of the technology will be used for the first time within the project.
- **Some functional requirements have not been fully communicated to us or have been ignored completely.**