

Build Phase

Introduction

Give yourself a pat on the back. You have made it to class this far! Your client project has evolved from a short description of the project to the **Build Phase**. This means you have:

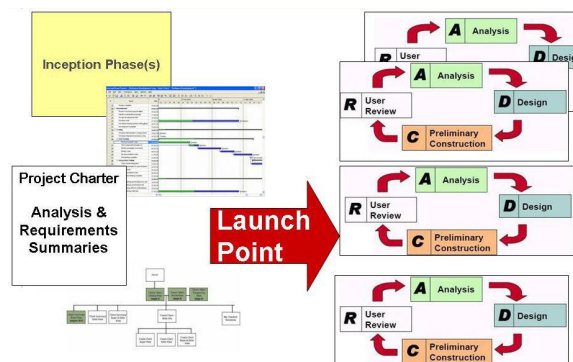
- Identified and collaborated with your stakeholders.
- Distinguished the symptoms from the actual problems you must solve by doing a root cause analysis.
- Created DFDs, Milestone Summaries, Gantt Charts, Network Diagrams and Use Cases.
- Narrowed your project's scope to fit the semester's limited schedule, and your team and client's resources. (Remember the triple constraint triangle).
- Established the functional and non-functional requirements.
- Identified risks and mitigation strategies.
- Calculated ROI.
- Analyzed the organizational impact.
- Evaluated system alternatives.
- Created various prototypes.
- Identified hardware and software environments and created your sandbox.

You have a project charter: a plan for execution that serves as the contract between you and your client. The Inception Phase is over; now it is time to begin executing the plan to build the promised functionality for a release this semester. Figure 1 illustrates the process for a team that plans four simultaneous, functional build iterations.

Description

The **Build Phase** is a series of iterations focused on specific functionality. Each iteration is comprised of Drill Down activities for Analysis, Design, Construction and User Review for the specified area. The number of functional phases is determined by the scope you set in the Inception Phase, the number of team members, and their skills.

Figure 1. Launch Point for a Team Planning Four Functional Build Iterations



Drill Down Analysis tasks focus on the requirement details for each functional area, such as what web pages and reports are needed and what information is entered and displayed. During these iterative loops, application developers work with key users to learn what support materials are needed; for example, is training relevant for in-house support or will on-line help be adequate?

During these function-specific phases a business analyst may work directly with the users, observing their work tasks and how their current process is either facilitated or hindered by their current system. The business analyst will work with a programmer on drill down design details to be sure the design meets the requirements; the programmer will then construct the actual system. Or a programmer-analyst will work directly with the users and do his or her own design, construction, and reviews with the user.

As the build phases begin the focus is on software functionality and planning for project hand-off. A project manager or business analyst will analyze what documents are needed for stakeholders and developers who support the project in production. Or programmer-analysts will need to take notes on these requirements during the build process. No matter how your team is structured, analysis for user materials and developer materials are important tasks for

these build iterations. These activities are the focus of future class readings.

Drill Down Design tasks focus on “how” to deliver the required functionality for each area. It is here that you will also figure out how you will retrieve the data for the website or the report. Your database design and information architecture may have been completed at the detail level during your Inception Phase or that work may have stayed high level until the build iterations. Final details must be worked out and verified during the build iterations.

For a custom developed system the analysis and design proceed together. The design issues for the database, reports, and user support materials will be complex, especially for dashboards that allow stakeholders to provide a range of reports from several transaction processing systems. For systems relying on purchased packages, such as Oracle Financials and SAP modules or open-source applications, much of your work will focus on what customizing must be done so that stakeholders gain the most from the system. Modification (and design) should be minimal to keep time spent on initial development low.

Consider the total cost of ownership. For example, regulated industries such as energy and health care must adhere to government regulations. When regulations change and the packaged system must be modified additional cost will be incurred by the organization. This is especially important for financial systems, since changes to government regulations can be extensive.

Drill Down Construction includes coding, unit testing and writing preliminary documentation. Remember, for you this is a 12-week project; for your client it is their livelihood and will be an ongoing project so be professional. Include detailed comments in your code so that future programmers can easily update and make enhancements to your 12-week project.

User Review includes user testing and review. Most importantly it includes user feedback to “get it right.” Your team’s discussion of the coded functionality should include plans for documentation and training needs as well. Use your client meetings to show results of your work to your client early. It does not have to be

finished or perfect. User feedback is vital to the success of your project.

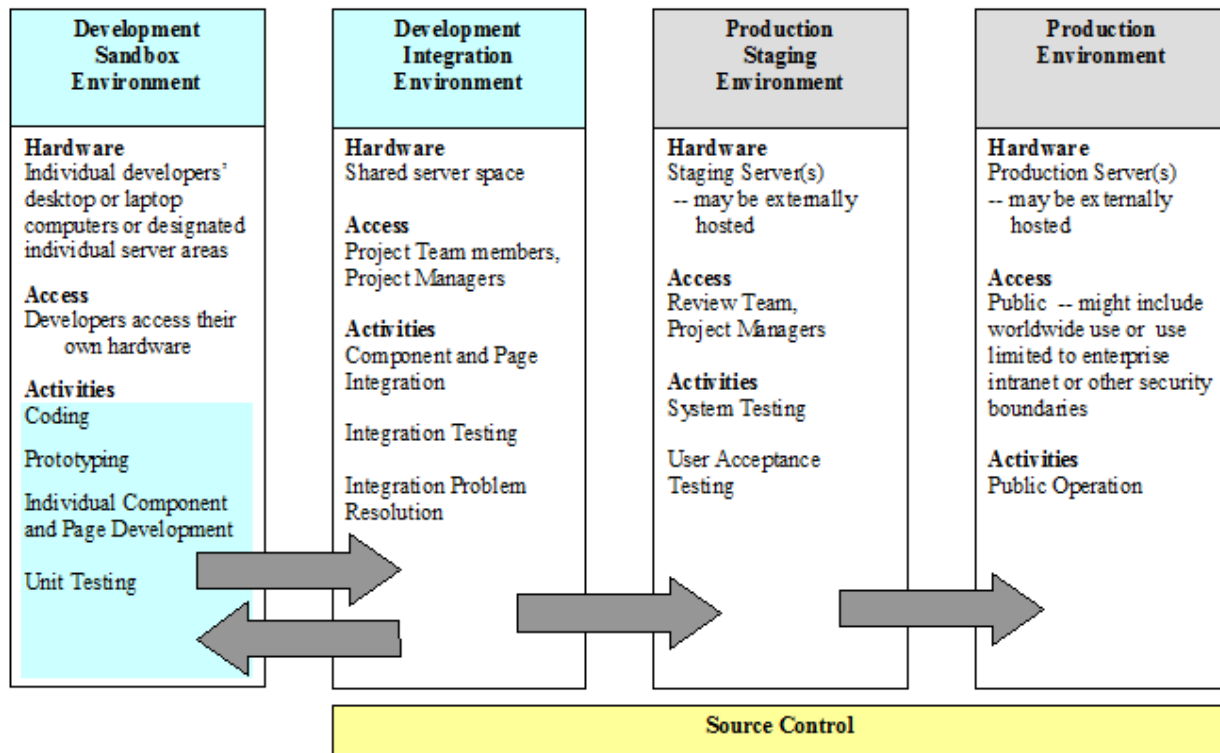
Purpose

The goal of this iterative process is to create an acceptable solution to your client’s problem. Gaining expertise in their work responsibilities and organizational goals will allow you to ask for constant feedback, so your application development team can tweak the design and provide the software and documentation necessary to meet their goals.

The iterative process works best when you and your team delegate responsibilities. In most projects there will be multiple software modules to create, so each one of you will take a module and do the coding. Of course your client will expect the “look and feel” of modules to be the same so be sure that your team’s coding and I/O standards are well defined and understood. Then, make sure that everyone adheres to the standards.

Creating Staging Environments

As soon as the platform for the production system is chosen, the developers should immediately create a developer’s environment, often called the **developer’s sandbox**. The sandbox includes the necessary software development tools and preliminary test data that are required to construct the new system. The environment duplicates project files locally, enabling team members to exercise their creativity and rapid development skills while not interfering with the clients’ stable production environment. Ideas, proof of concepts and prototypes can be built in the sandbox with little constraint on the resources normally required to install, configure, and execute software in production. Once a software module or series of modules has been completed and approved, the module(s) can be taken out of the sandbox and promoted into an integrated test environment for further testing, eventually leading to a next level of testing with all system components before installing the software in the production environment. See Figure 2.

Figure 2. Coding and unit testing occur in the first four software staging environments


Developers using sandboxes should communicate often to ensure that each team member is kept up to date with the overall project. Version control software, such as Microsoft's Visual Source Safe, may be purchased to help manage this process. The lead programmer creates a quick sample module and tests it in the production environment to be certain that the sandbox has the same or a compatible version of the development software. The advantages of creating a development environment separate from the production environment are:

- Reduces the risk of introducing errors into production environment.
- Reduces the risk of failing to complete all functions by including test data for all possible errors and time periods, such as quarterly and year-end reports.
- Saves time and money because you are working with short test data files and less expensive hardware with easier access than the production environment. For example, a desk top development environment that

simulates a Web-based production environment that includes dozens of servers will be much less expensive processing than the actual production server farm maintained by an operations staff.

- Eliminates any potential interference with production activity for completed systems.

Once the team has completed all of the functions for the current release, a user-review will determine whether the developers should proceed to additional functional build iterations or to Final Construction.

Tips for Successful Transition from the Build Phases to Final Construction

- Be sure to test migration of code and compatibility of development and production environments early!



- Compile notes on reference material needed for users and developers.
- Test at least one quick reference guide with your users early in the process to learn their level of comfort with terms. This is most likely to be a guide for administrative activities to keep the site up-to-date, since your site should be self-explanatory or need only minimal on-line help for customers.
- Hold focus groups or create usability surveys for any site customers who are key to the success of your project. Work with your users to learn who might provide helpful feedback.