

PDSA.com

Solutions for the Real World

PDSA Special Report

The Importance of Prototyping

The Importance of Prototyping

Users are NOT programmers! I see many analysts and programmers forget this during the up-front design phase of a custom application development project. They throw all sorts of database diagrams, UML documents, specification documents, and other workflow artifacts at the user until the user's eyes glaze over. This is where a simple prototype comes in handy.

Users Need Visuals

Over the past 25+ years of designing and architecting applications for users, I have found one constant: users want to see the user interface before the application is complete, period. The sooner a user can see and interact with their screens, fields, reports etc., the better the overall application will be in meeting the real business need. I have found very few users who understand database diagrams, UML diagrams or workflow diagrams. However, if you put a screen in front of them and show them how they can enter basic data, move from one screen to another, or see a mocked-up report, they understand what they are paying for.

While we need to build database diagrams, create data dictionaries, normalize our database tables, create UML diagrams, workflow diagrams, and programmer specification documents, most users never need to see these project artifacts. In fact, if you show these to them, they won't understand them anyway, so just keep them to yourself.

A Prototype-Centric Development Methodology

I know a lot of analysts and programmers who start with UML or database design as the first step in developing a new application. I respectfully disagree with this approach for several reasons:

1. Users cannot inform you of all the inputs they need in just a few meetings. This will leave you with incomplete UML and missing database fields.
2. Users don't understand databases. They only know about the data they need, not how it is stored.
3. Database diagrams are too complicated for users to look at.

4. Users can not look at 500+ columns in tables and know whether or not you have missed something. Users think in terms of entries they have to fill out on forms or the values they see on reports.
5. UML and workflow diagrams are confusing for most users.

Instead of using the above approaches, why not try a fresh approach? Create some input screens that will eventually become a part of the final application. Now the user can see if something is missing. After all, you will spend as much time in Visual Studio mocking up a screen as you would in your favorite UML tool or database diagramming tool. The end result is much better because the user sees progress faster and will be able to tell you if something is missing much quicker.

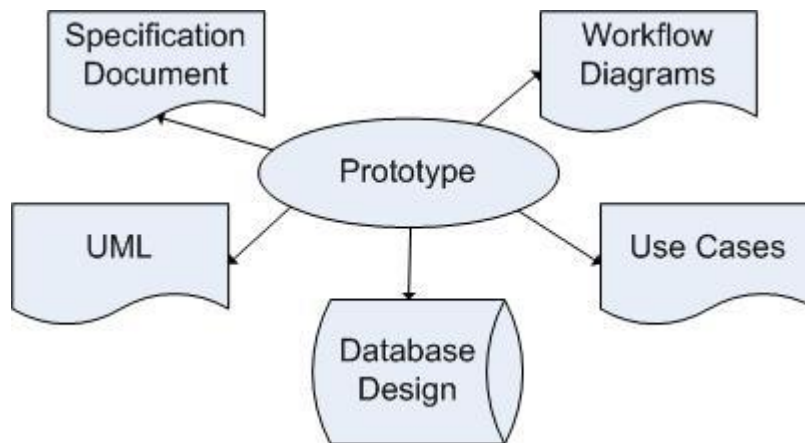


Figure 1: A good prototype will help generate all your other project design artifacts

Step 1: Discovery Phase

The first step in any development project is to gather the reasons why an application is being created in the first place. You will also need to establish a case for developing this application in order to get the budget required for the development process.

1. Gather and write down the business requirements.
2. Establish a financial case for creating the application (ROI).
3. Work with the user to see where you can help improve their process.

Step 2: Design Phase

After Step 1 is complete, start gathering the specifications that will eventually become the final application. Again, instead of starting with database diagrams or UML documents, start by cracking open Visual Studio and prototyping screens based on what the user has communicated to you. As you develop the screens you will also create a specification document that captures the business rules for each screen and each element on the screen. Some of these rules will need to be filled in after you show the prototype to your user.

In addition to the screens, there are two other items to be prototyped: work flow processes and reports. In most applications there will generally be some processes that need to be performed. These processes should be defined in a class with stubs of the method calls that need to be performed to create the correct outputs. Reports are just as critical as screens to prototype. If the user identifies a report that has some fields on it that you have not seen anywhere on your screens, then you have identified something that either needs to be placed on a screen somewhere, or some process must be created to calculate the fields for the report.

What you end up with is a more iterative dialog with your user. The user can see progress on the application, they can make better suggestions on how the screens should look and what things might be missing or can be eliminated. You get the advantage of more focused meetings with your user, plus you are building trust that you truly understand their problems. By prototyping you have also gathered all inputs and all outputs. Once you have all inputs and outputs identified you can more easily list the columns that belong in your tables, and thus your database design will go much easier. Also, if you have reviewed (and watched) your user interacting with your prototype, your UML and workflow diagrams are much easier to create. In fact, you can now relate these diagrams back to the prototype and the diagrams will make much more sense to your users.

Below is the iterative process you should follow during this phase to create the final specification document, the diagrams, the database design and the prototyping that is ready to be coded in the next phase.

1. Prototype screens.
2. Create mock-ups of reports.

3. Write specification document.
 - a. Write business rules
 - b. Create a database design.
 - c. Create UML diagrams.
 - d. Create workflow diagrams.
 - e. Create use cases.
 - f. Create test cases.

4. Show the user and repeat steps 1-3

The above steps are a very iterative process where you continue to refine the application until everyone is satisfied with all screens, processes, workflows, database fields and reports. Only then can you begin to code this application.

Summary

Creating a prototype can speed up your development process. A prototype helps you identify all input and output fields. It also helps you define the workflow for your application. Once you have these items in your prototype it will be much easier to come up with workflow diagrams, UML and use cases, and your database design will go much quicker since you know all the fields required by your application.

Contact Information

If you would like to know more about the information in this special report, please contact either Paul D. Sheriff or Michael Krasowski at PDSA.

Paul Sheriff

(615) 675-4632

PSheriff@pdsa.com

Michael Krasowski

(714) 734-9792 x223

Michaelk@pdsa.com

Company Information

PDSA, Inc.
17852 17th Street
Suite 205
Tustin, CA 92780

Tel (714) 734-9792
Fax (714) 734-9793
www.pdsa.com

PDSA.com
Solutions for the Real World