

FLiP and You

The Fusebox Lifecycle Process (FLiP) is a process for developing web-based applications. Its primary concern is to create the system that you the customer are expecting.

To that end, FLiP is comprised of a series of steps that represent a movement from phases where changes are quick and inexpensive through phases where changes are increasingly difficult. This means that the emphasis is on spending a lot of time on the early, inexpensive phases (wireframe and prototype) so that later, expensive changes are avoided.

Each step of the FLiP process is designed to provide you with just what you want for your system. The first two phases are formalized conversations between you and your consultant, ensuring that your consultant knows what you want, and that you are comfortable with your consultants understanding of your needs.

The remaining phases are a formalized approach to software development that allows easy tracking of progress and milestones, and comprehensive after-action reporting.

This roadmap is intended to be your guide to FLiP. We encourage you to use it as a means to talk with your consultant throughout the process, so you can achieve a successful deployment.

Who Uses Fusebox?

Fusebox is the most popular framework in existence for CFML-based development. A cursory search using any search engine results in a wide variety of industry leaders who use Fusebox in their web sites.*

Here are a few:

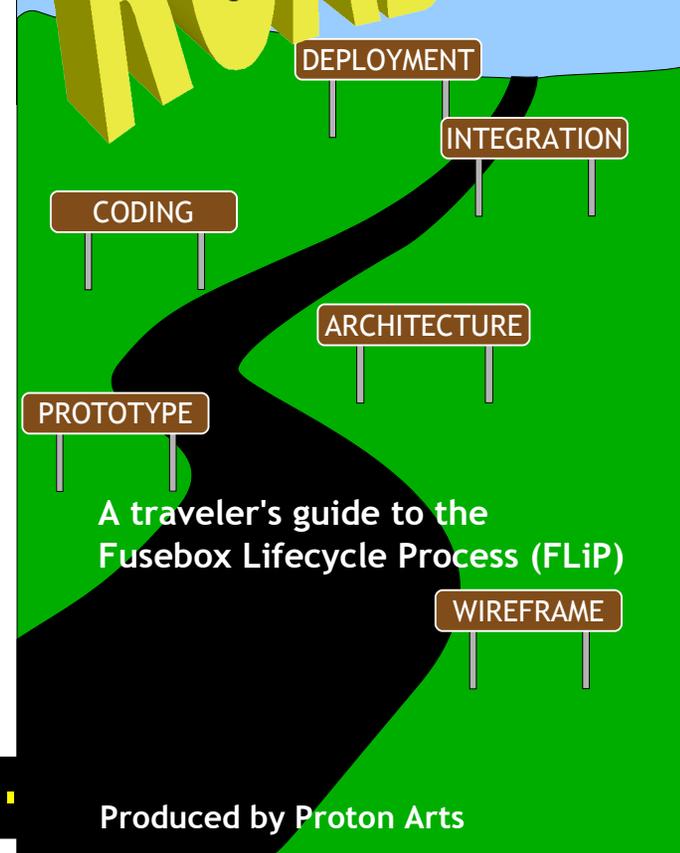
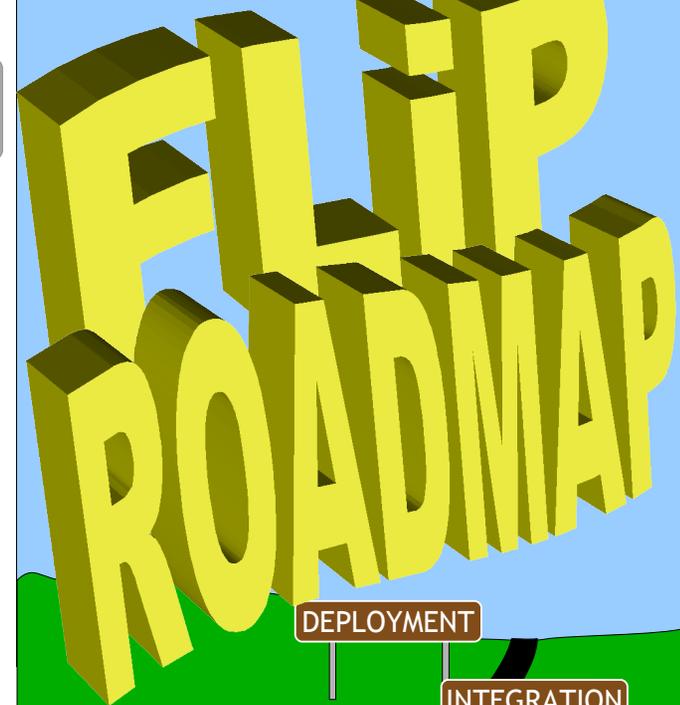
AT&T Business Internet Services
American Airlines
Business Ethics Magazine
mySpace.com
National Down Syndrome Society
The United States Senate
The United States House of Representatives

The Fusebox Lifecycle Process (FLiP) isn't something you can use a search engine to detect. However, anecdotal evidence suggests that the vast majority of Fusebox developers also use FLiP to help ensure the success of their projects.

In addition, a large number of Fusebox developers create applications for intranet environments, which are not open to public view. Many internal development efforts have taken advantage of the benefits of Fusebox and FLiP.

*Inclusion in this list does not indicate endorsement of Fusebox, FLiP, Fusebox Inc., Proton Arts or this brochure.

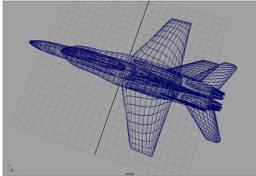
FUSEBOX



1 Wireframe

In the Wireframe phase, your application is modeled from a business process perspective, without regard to how it's going to look. The main concern during this phase is how the application will work; what it will look like is a concern to save until the next phase.

During wireframing, your Fusebox consultant will be asking you about ideas like how you currently do business, what services you'd like the new system to provide, and who will be using the system.



You'll also be asked to identify a person as the decision-maker for the project. This person will be vital throughout the FLiP process, helping your consultant to know when your needs have been met.

3 Architecture

Once you and your consultant have finished the Prototype phase, you can relax a bit while the "techies" dive into the heavy work.

The Architecture phase is where your consultant takes the front-end developed in the previous phase and plans out the application's structure from a technical perspective.



All the program code that will be written is specified during the Architecture phase, so the programmers can produce well-written modules that meet the demands of your system.

At the end of this phase, your consultant will have a complete plan for the coding, testing and deployment of the project.

5 Integration Testing

The coders write modules and unit test them, then send them back to architect for integration testing.

Integration testing is just the process of making sure the overall application can do the things it is expected to do. The architect runs through a variety of "use cases", which are just sets of activities that users need to be able to perform in the system. If the system can "carry its weight", it passes integration testing and is ready to be deployed.



Depending on how your project is structured, you or your people might participate in integration testing, or it might be handled entirely by your consultant. It's best to talk it over with your consultant so you know how it will work for your project.

2 Prototype/Front-End

The Prototype (or Front-End) phase is where the project really begins to look like something you'd like to use. Your consultant will lead the construction of a mock-up of the application, so you can get the feel for how it will actually work.

Feedback is critical in this phase. Just like when an optometrist offers various lens choices, asking, "Is this one better, or that one?", so your consultant will give you the ability to choose exactly how you'd like your application to look and feel.



This all occurs prior to any architectural or coding work, so that the application will work the way you want it to work. It is critical that you ask questions during this phase, since the application will end up looking just like the front-end that is developed now.

4 Coding & Unit Testing

The Coding and Unit Testing phase may be a bit unsettling from your perspective. There is a lot of work going on during this phase, but you won't see much happening from the user perspective. In reality, the programmers are hard at work, putting in place the software to make your system run as well as it looks.

As each module of code is written, its programmer subjects it to unit testing. Unit testing is based on directions provided by the architect, and ensures that each module does its job properly. It's kind of like taking your temperature—the programmer is checking to make sure everything looks normal. If it does, then the code goes back to the architect for integration testing.



6 Deployment

Deployment is an anti-climactic even in the FLiP process. All of the careful planning and up-front design work done in the previous phases means the resulting system will be just as you expect it to be—no surprises, and you can unleash the greatest thing since the telegraph key on an eager world.

That said, there are always a few curveballs that seem to find their way in during deployment. Your consultant will be concerned with issues like required availability schedules for the system, backup requirements, etc.



You may also agree to repeat some of the use cases done during integration, as assurance that it is working as you expect it to work.