

Improve the Usability Architecture of PHPList

Author: Tarang Sagar

Tarang is a senior open source developer at Comnez. His hands on experience in modifying the various open source platforms has made it possible for our clients to get simplified solutions to complex business problems.

The Challenges

PHPList is a web based open source email marketing solution. The application has several useful features and functionality wise it is the one of the best email marketing software solutions available in the industry. The application however poses some serious challenges with its usability architecture. We realized this only after deploying the system for various clients and hearing repeated and similar issues related to user experience. Most of these clients were non technical and found it really hard to understand the various functions provided on its user interface. That is when we found out that the system is friendlier to a technical audience compared to a non technical user base.

This was a very challenging issue because even though the system was highly functional and useful, the users were not able to take advantage of it due to a poor usability scheme. When this challenge was presented to us, we were left to figure out a better usability scheme without too many directions from our clients. All they said was that they want a more user friendly solution.

Our Approach

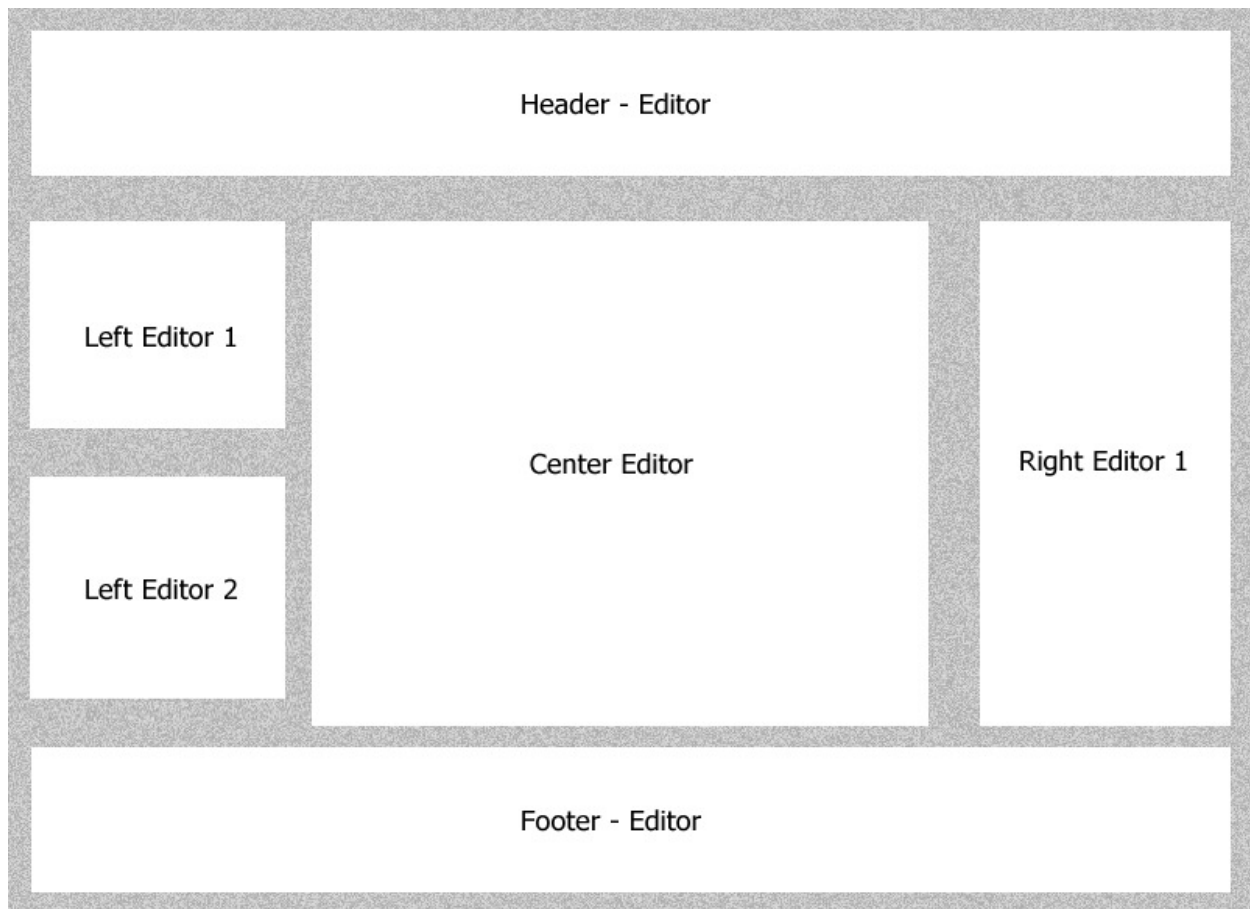
The system was given to our usability architect for a thorough examination. After a careful analysis of the system and keeping in mind the various feedbacks we had received, we finally concluded the following:

- The navigation structure of the system was not very intuitive because of the way the various elements were named and arranged.
- There were a lot of functionalities which were never used by an average user and were useful for advanced technical users only.
- The most important area that needed to be improved was editing of the emails and email templates. The default editor expected the end users to have a basic understanding of HTML. Our goal was to have anyone use it, irrespective of their knowledge level in HTML.
- There was no way for the users to insert images in the emails using a point and click approach. The users were expected to upload the images to a server location of their choice and the URL for the images had to be entered manually.

Our Architecture

Keeping the user base in mind we developed a scheme that would preserve the HTML editing capabilities of the emails and at the same time enable the users to create and modify the emails without having to worry about the layout issues.

We divided the email into header, footer and body columns where users had the ability to create any kind of layout using these combinations. The pre-existing email templates were programmed in a way that they would automatically split into different layout tiles in the edit mode. The following diagram shows the wireframe of a three column layout for an email newsletter.



As you can see, the email is divided into different tiles forming a unique area for the content and each area has its own editor. In our case we were using the TinyMCE editor and we modified the code of TinyMCE to add functionalities required by this new editing scheme. The biggest advantage of this architecture was that it always kept the layout of the email intact.

We added an image library and also a content library for the users to store their images and content and insert it into various emails as required. This way they would no longer be expected to manage the images and content items manually. Also we added an additional feature that would enable the users to save any email as a template for reuse.

Technical Implementation

The implementation of the architecture described in the previous section required code changes to the various areas of PHPList. These changes were mainly applied to the core PHP files and also new files containing new PHP code were added to implement the enhancements such as adding the image and content library to the system. There were database changes as well and a few existing tables were altered and a few new tables were added. The technical skills required to perform these tasks were PHP & MySQL on a Linux platform.

The most challenging technical implementation for this project involved the customization of the TinyMCE HTML editor. This is a java script based editor that involves complex java script coding. My past experience working with several other open source programs helped a lot and shortened the learning curve that was required for these customizations.

We applied agile methodology for this project. This was done because we were looking to seek constant involvement from our client in terms of providing the right direction to the project and add useful functionalities for an improved system.

Overall the process was very exciting and after several iterations and several rounds of end to end testing we were proud to see what we had accomplished.

Results

As a result of all the effort we put into this project we finally evolved PHPList into a very user friendly application. We upgraded our existing clients that very using the basic version of the software and they were really pleased with what we had done.

Not only that, we got some very good feedback from our sales team who started presenting the new and improved version of this software as a subscription service to prospective clients. There was an immediate visible difference in the way the new system was accepted by our business associates.

Since the completion of this project the support issues have largely reduced and we also feel that the training department is delivering better results in terms of making the users more comfortable with the improved system.

We have met the goal but our sustaining engineering efforts have not stopped. We firmly believe that there is still a lot of room for improvement and we will continue to go down the road that leads to a better system.