**Rocket Reference**

**Website Project**

**Dan Budziak**

**Dan Paplaczyk**

# Theory & Background

The term software engineering comes from the idea of building applications that were as equally predictable in quality, cost, and time as a project that was developed in civil engineering [1]. When conducting any project, there needs to be a great deal of planning prior to the actual development and in the 1960s, developers were looking for a better process to build software applications that like any other major project, was on a plan and stuck to a budget. By coming up with a plan before beginning the development of a project, software engineers can oversee several different processes that go into the lifecycle of software such as the maintenance of an application even after deployment. By having a strict plan in place, developers were able to become more efficient through having more interaction with their client.

In order to ensure that the created software is in good standing, software engineers must follow a certain code of ethics to ensure that the goals of the client are met. The ethics consist of public, client and employer, product, judgment, management, profession, colleagues, and self [2]. These ethics allow and maintain integrity between the software engineer and the client. By having a code of ethics in place, it allows for ensuring that a software engineer doesn’t do anything that would put the public in harm while keeping the best interest of the client in their minds.

The software development process allows for software to be built in a timely and efficient matter that follows a budget through different phases that improve the applications design and product management. One of the most foundational software development processes is the agile life cycle where developers focus on constantly having working application while building improvements to the application throughout iterations of development [3]. By utilizing the agile software development life cycle, developers must be able to respond to change throughout the applications development process [3]. By utilizing this method, developers are able to get direct feedback from the client as they release new modifications little by little and are able to alter and correct aspects based on the customers needs.

When developing any kind of software, the client and the engineer must have a mutual understanding of what needs the goals of the application are, how they need to be implemented and when everything needs to be finished. A software requirements specification (SRS) is where the client will submit cases on how the user should interact with the application and should include a detailed description of all functional and non-functional requirements agreed upon by the developer and the client [4]. An SRS lays a solid foundation for both parties to get a general understanding on how the application should work from a user’s point of view and the expectation that the client has for the developers to meet.

When using software system modeling, the goal is to make sure that the application can be understood in an easy and abstract way. By doing so, you are able to present your expectations for the application to the client in an efficient and understandable way [5]. Modeling the system can also be used for the developers to get a better understanding of different things that need to go into the software. For example, a developer can model all the different servers and databases that the software needs to communicate with before they begin writing the code for the product. This allows for them to get a better understanding of everything before they must keep changing things because another aspect needs to be added to make the application work. A great way to implement this type of modeling is by using Unified Modeling Language (UML). UML is great for creating sets of diagrams to visualize and construct the specifics of a software system. By using mostly graphical designs of the software architecture, UML allows for a streamlined approach in clarifying all aspects to the client [5].

Software testing can be done for various different things but is one of the most important aspects when developing an application. Essentially, there are two different groups of testing used to test both the functional and non-functional components of an application [1]. By testing the system, we are making sure that the application meets the SRS requirements that were agreed upon by both the developer and the client of the application [1]. Another important component of testing is ensuring that the application is secure and cannot be accessed or that anything can be deleted by unauthorized persons. For example, when an application uses a database to display data, nobody should be able to directly access that unique database besides the application developers and the client.

Software needs to constantly be changed or evolved for a number of reasons. When it comes to software evolution, developers need to constantly update and “evolve” the application as times change. This is another perk of using a development life cycle such as agile because as time goes on and you gain user interaction and feedback, developers can see what is being used on the application, how certain aspects are being used and then formulate conclusions on what needs to be changed moving forward. A good software system will be able to stand the test of time, resulting in developers being able to update and change features, add new features, adjust to business requirements, and improve performance [1].

# References:

[1] Fox, A., Patterson, D., & Joseph, S. (2016). Engineering software as a service: An Agile approach using Cloud Computing. San Francisco, CA: Strawberry Canyon.

[2] Software engineering code of ethics and professional practice. (2001). Science and Engineering Ethics,7(2), 231-238. doi:10.1007/s11948-001-0044-4

[3] Understanding the Agile Software Development Lifecycle and Process Workflow. (2018, October 12). Retrieved from https://www.smartsheet.com/understanding-agile-software-development-lifecycle-and-process-workflow

[4] Stc. (2015, April 26). Software Requirement Specification (SRS). Retrieved from https://www.softwaretestingclass.com/software-requirement-specification-srs/

[5] (n.d.). Retrieved from https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

# Rocket Reference Design Document

Our group, of Dan Budziak and Dan Paplaczyk, is working on implementing an online resource for space history and rocket enthusiasts. It is a website called [www.RocketReference.com](http://www.RocketReference.com). This will be executed as the Capstone project for our class and will highlight our skills and knowledge of the design cycle.

# Objectives

The main objective of the site is to provide well-sourced information about the technical aspects of the machines used for space exploration. Using primary sources from major space agencies, contractors and historians, the site aims to be the most accurate source of information available.

# Audience

The main audience will be space history enthusiasts. This group knows uses information like this to help them in building model rockets, improving their favorite games, and just increasing their overall knowledge. I have used sites like this to gather information for game development while working on various projects.

The other audience is amateur rocket builders. Many times, these groups want to have accurate data about the rockets they want to recreate and this will provide them a good resource for it.

# Current Market

Other sites exist that provide much of this information, but most of them are weakly sourced, or not sourced at all. The largest site and the first result on most search engines is Encyclopedia Astronautica at <http://www.astronautix.com>. This site has a huge amount of information, but there is not a single source listed on the entire site. In my research, much of the information on Astronautica has proven to be inaccurate. Without knowing the sources of the data, it is hard to determine the validity of the information.

The next largest repository of information comes from Wikipedia. Wikipedia does a much better job of listing sources, but unfortunately, for many of their listings, Encyclopedia Astronautica is the main source. There are additional websites that exist with good information, but none of them are comprehensive enough, or sourced well enough, to be the ultimate resource of information. Some of these sites include:

* Space Launch Report by Ed Kyle - <http://www.spacelaunchreport.com/>
  + Very good information with good sources, but mainly includes detailed information only about major launch vehicles.
* Russian Space Web - <http://www.russianspaceweb.com>
  + Good information on hard to find Soviet era hardware. Some information is incomplete and the site is difficult to navigate.

# Site Structure

Attached to this document is a Sitemap of the layout and top-level organization of the site. There is also a mockup design of the homepage.

Overall, the site will be structured into different components of space exploration. The different sections will be inter-connected with links to the various components. This is an example:

* Launch Vehicle: Delta IV
  + Country: United States
  + Manufacturer: United Launch Alliance
  + Stage: Common Booster Core
    - Engine: RS-68A
      * Propellant: Hydrolox
    - Engine: GEM 60
      * Propellant: HTPB
  + Stage: Delta Cryogenic Second Stage
    - Engines: RL10-B-2
      * Propellant: Hydrolox

Each of these components will have its own entry inside of the category where it belongs. These entries will have links internally to the other components that you see in the above section.

In addition, the first letter of the name will group each entry alphabetically. The user will be able to select all entries in each letter, similar to how an Encyclopedia volume works.

# Scope of Work

The site will be built on the WordPress CMS using the Newspaper theme from tagDiv as the basis for what we will be creating. This theme allows for a lot of visual customization and will allow us to focus on the data and functionality of the site. We will need to create the following plugins:

* Custom Post Types – We are creating custom post types and layouts for the following sections: Engines, Launch Vehicles, Propellants, Satellites & Probes, Spacecraft and Stages. Each of these will require PHP coding, HTML design and CSS design to function properly.
* Custom Functions – We will want categories to display their information in alphabetical order instead of by post date. Another function we will want to have is to only have Menu Links active if there is data inside of the Category.
* Functional User Layout and Design – We want the information to be easily accessible and readable for the user. This is going to require multiple iterations of designs for the various sections. We will want to institute A/B Testing in order to determine the best design.
* Graphic Design – There are a lot of graphics that need to be created for the full project. Some have already been completed and there is a lot more that will be required.
* Research and Data Entry – This will be a multiple years long part of the project that will continue well past the scope of the class.

# Individual Contributions

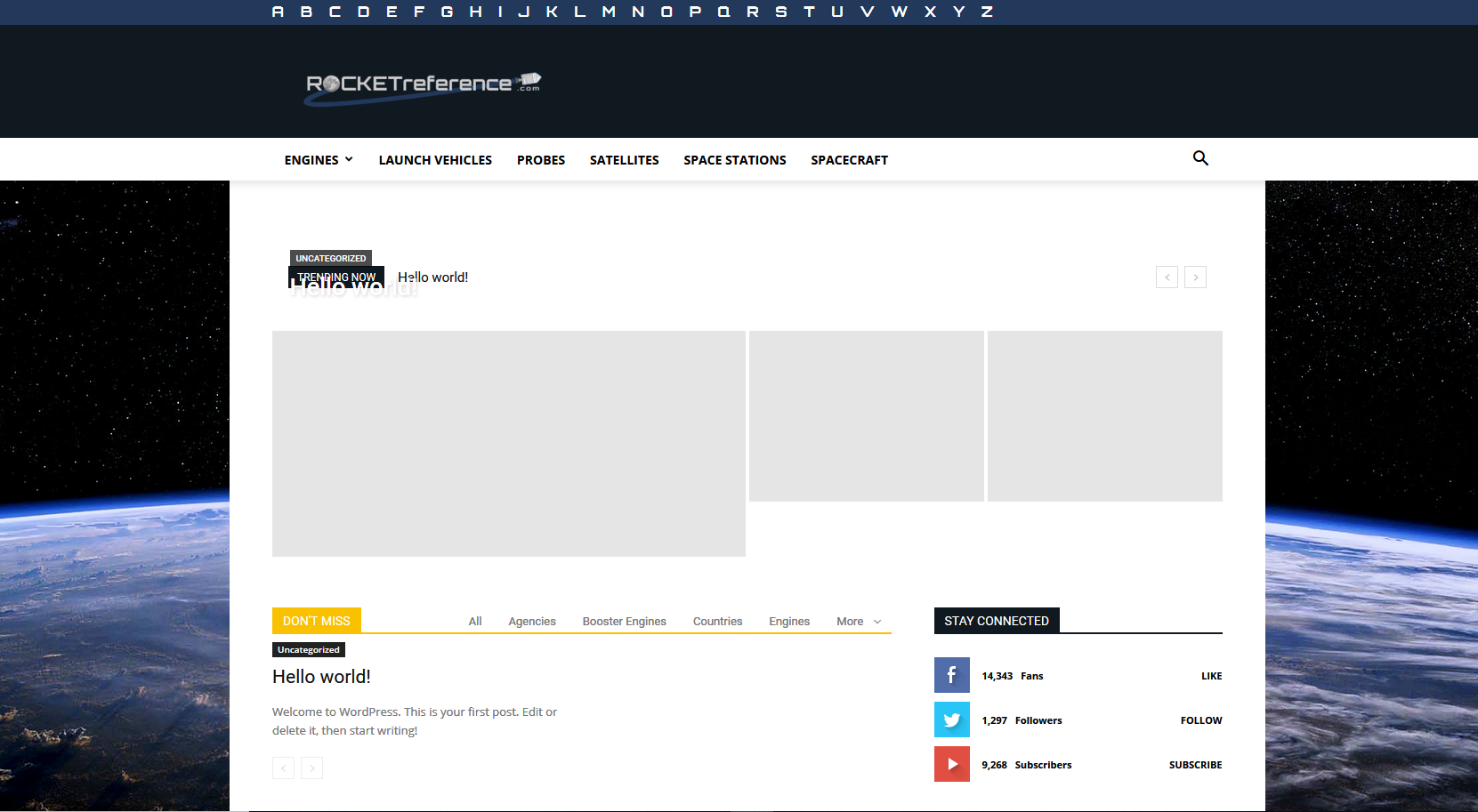
We are obviously in a different situation than many with the pivot of our project from the Fairmont school website to our new project. During the first four weeks of class, both Dan’s met with the client from Fairmont multiple times to learn about the scope of the work that was required. Both of us also looked into options on where we would be hosting the site and what CMS we should use to help them. Ultimately it came down to them choosing a different route and combining their projects into one. It is by far the best outcome for the client, so even though we did not continue on the project, I believe that we had a positive impact on helping them get to the decision they made for their needs.

For the Rocket Reference website, Dan Paplaczyk has done all of the design documents, planning and actual coding on the site. In this situation, Dan P. is one of the project members, but also in a strange way, the client as well. The idea for the site was his to begin with, so most of the design elements have been provided by him. Now that we have a working design document, a mock-up of the site and a plan of action, both members will be entrenched in the hands on work of the site.

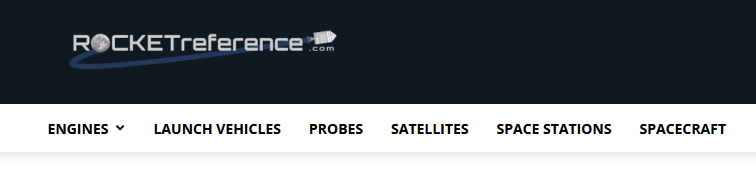
Dan Budziak transitioned late from the Fairmont project to the Rocket Reference project. This put him in an unfair position of not being able to work on the design aspects of the site. Instead, Dan focused heavily on the Theory and Background information for the project and researched and wrote the paper for this Milestone.

# Currently Implemented

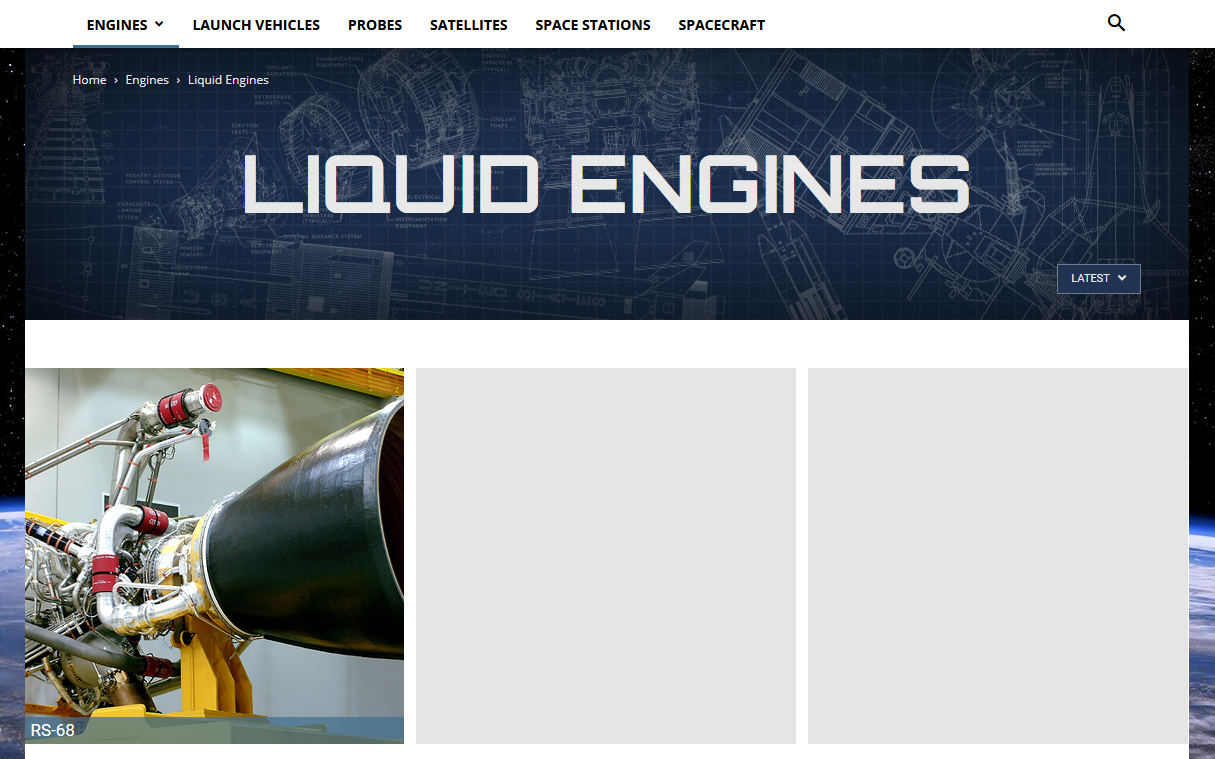
General Layout of the Site Completed with a Logo. There is also links to the individual letters that will store the posts.



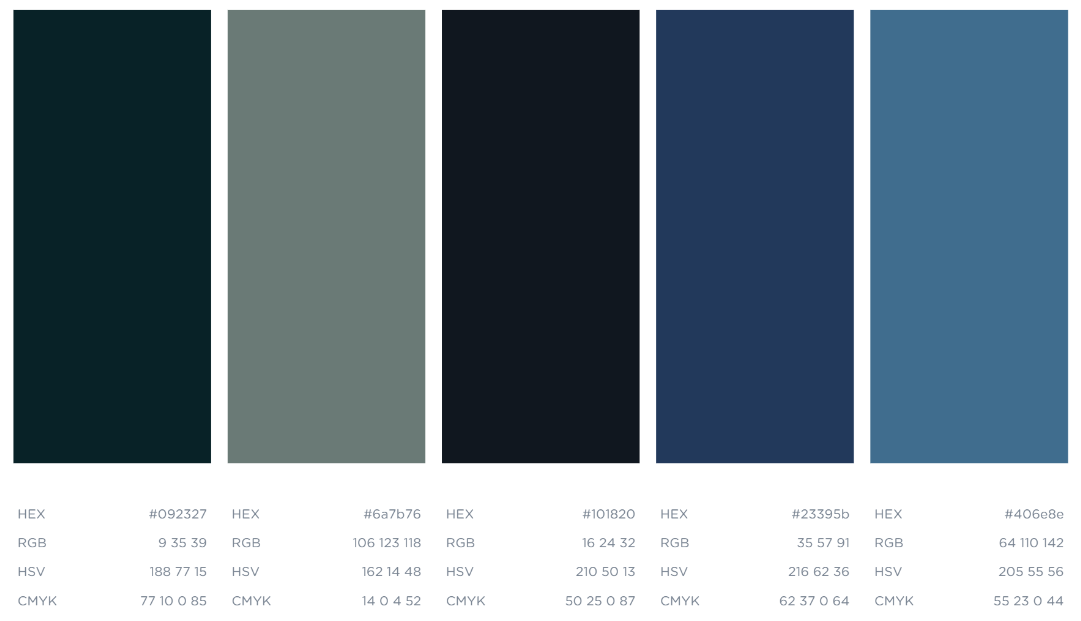
Categories have been fully implemented and they are available as part of the main menu:



Custom Category Design along with newly created graphics for the Category headers implemented:



Color Pallet Selected for the site:



# Next Steps

1. We are looking to focus on an interim web host as we work on the development of the project. Currently the site is hosted on Dan P.’s personal computer as it requires a server. Currently it is using a WAMP server locally. There are some options where we might be able to host student sites for free, and then we will transition to a paid host once enough of the development is completed and we can soft launch.
2. Create the custom WordPress functions for using the custom post types we will need for the different components of space history we are going to provide.
3. Begin the primary source research to determine what we want to start with and how much we will be able to get implemented before the completion of the project.

# SiteMap

