**CruzSafe Software Versioning Proposal**

Original Creation Date: 5/30/2019

Proposal Author: Richard Torres

**Proposed Format:**

*[Major]. [Minor]. [Revision Number]. [Release Status]*

**Introduction:**

In order to release our product, we must define a way in order to identify the product builds we have released in order to more accurately determine where any reported problems may lie. In this format, we have four major components: Major Update, Minor Update, Release Status, and Revision Number. With this Software Versioning format, I believe that we will be able to precisely determine which alteration the bug or feedback report may refer to.

**Field 1: Major Update:**

This field will dictate the major changes made or features added to the application. This includes critical bug fixes, new features such as camera or gallery support, or major component optimizations. This field should perhaps be incremented sparingly, as not every change may be large enough to warrant an increment. In addition, a large bundle of minor changes may be considered a major update.

**Field 2: Minor Update:**

This field will dictate any minor changes made to the application, such as minor optimizations, bug fixes, rewording, replacing assets, etc. Such changes should not involve any major change in the functionality of the application and should mainly focus on fine tuning of any major updates. This field can be incremented more frequently.

**Field 3: Revision Number:**

This field should only be incremented if the Major and Minor update pair has been rewritten or revised. This should allow us to record any changes made without having to increment the Minor Update field should it simply be an alternate means of writing the same code. This mainly serves to explore alternative versions of the same code without making any alterations to the Major or Minor Update fields. Should the team decide on a finalized version of the revision, it should increment either the Major or Minor Update field depending on the size of the change. By default, this field should be “0”.

**Field 4: Release Status:**

This field should dictate the state in which the Major and Minor Updates have been released. Simply put, this field should determine if the build number is in Alpha, Beta, Release, or Final. This field should only contain values ranging from 0-3; Alpha should be denoted as “0”, Beta as “1”, Release as “2”, and Final, should it be necessary, as “3”. As far as testing goes, Alpha will likely remain within the development team and would likely never be seen by the general populace. Beta, on the other hand, will likely be release to a few volunteers for public testing. Regarding Release, it should be the defining version that all members of the general populace should receive. Finally, the Final status should only be used when development on the application has ceased, and the team or organization has moved on from the project.

**Field Hierarchy:**

In order to keep the Software Version Number clean and easy to read, certain fields should be reset when another has been incremented. The main parent in this case should be the Major Update field. Whenever the Major Update field would increment, the Minor Update and Revision Number fields should revert to “0” in order to prevent large numerical values from accumulating (EX: 5.152.30.1). Similarly, when the Minor Update field has incremented, the Revision Number field should reset to “0”. The Release Status field should be the only field not affected by any of the other fields and should only be changed to dictate how the build has been released.