**October 4, 2017 1pm Meeting**

**Attendance:** Brian Pigford, Joseph Hays, Emmanuel Maku, Nick Myers, Dylan Weigartz

**Facilitator:** Brian Pigford

**Recorder:** Dylan Weigartz

**Use Cases:**

Find ratings on club(Nick)

Find information on certain clubs (meeting times, location, what they do, general member population) - (Brian)

Club accountability(Advisor) - Joseph

Look at the comments (Dylan)

Club reform (looking at data over time in order to make changes to club) - Emmanuel

**Actors:** Students, incoming students, faculty, club presidents, family, administrators

**Nonfunctional requirements**:

Def: **Non**-**functional requirements** - can be divided into two main categories: Execution qualities, such as security and usability, which are observable at run time. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system. (Stackoverflow)

These are called "**non**-**functional requirements**" or sometimes "Quality Attributes." For example, attributes such as **performance**, security, usability, compatibility. aren't a "feature" of the system, but are a required characteristic. (CalPoly CS):

**Compatibility:** Compatible to all browsers and androids.

**Accountability:** Make sure someone doesn’t violate college guidelines with review (inappropriate, etc)

**Maintainability:** Maintaining databases by updating every couple seconds.

**Security:** each account will have a password feature protected by a login.

GUI representation of the ratings,

**Interface-** People shouldn’t have to click through 6 different menu’s to get where they have to go.

**Conventions and templates-** Don’t write sloppy code, and make sure to make variable names relevant.

**Layering-** Reduces complexity of the software design.

**Algorithmic Complexity-** Write simple and easy to read code

**Hashing-** Important for efficient storage and retrieval of data.

**Caching-** Storing Data in memory to be used at a future date.

**Cloud Computing-** Able to access information and data across computing platforms of all types.

**Concurrency-** Two events occurring or happening at the same time.

**Security-** Protecting the confidentiality and integrity of system assets should be paramount to every computer professional.

**Relational Databases-** A database whose structure recognizes relations among stored items of information.

**Discussion:**

Do you agree or disagree with the following statement: “Since we deliver multiple increments to the customer, why should we be concerned about quality in the early increments—we can fix problems in later iterations.” Explain your answer.

We disagree because it is a unprofessional and unethical way to treat clients. The problems that are early onset would not be discovered, until later on when they have been built upon and become harder to correct. Like you would build a house you need to build your software with a good foundation. The client’s requirements can be fully realized without attempting to fully meet them from the first increment. Quality is important at all stages to insure the client continues to use your services.