Team 3 **Exercise 1: Choose a Process Model**
CIS 422
Dr. Stuart Faulk Due Date: 01/16/2017

1 Iterative Development

Team 3 has decided to choose an iterative style of software development for Project 1, with the inclusion of these four phases: Inception, Elaboration, Construction, and Transition (I-I, I-E, I-C, and I-T).

1.1 Iterative Development & Project Goals and Constraints

- **Goal: Complete Documentation**
  The life cycle of our development phase splits into four main phases with a large emphasis of our initial time spent working on a Requirements Analysis (RA). The RA will create a proper Software Requirements Specification (SRS), a Software Design Specification (SDS), and low-fidelity prototypes. To determine requirements, we will perform a Hierarchical Task Analysis (HTA) and create User/Activity Scenarios to guide these documents within the project specifications. An iterative life-cycle will allow us test these methods of analyses and determine that we have a solid starting point.

- **Goal: Functional and High-Quality Code**
  A iterative cycle (with emphasis on the inception phase) will minimize the risk of miscommunication and lack of organization. A properly laid out SDS will provide substantial guidance for coding the project in a small group.

- **Goal: Clear Roles and an Equal Spread of Work**
  A iterative development will lay out everyone's role in the inception phase, and we will be able to plan out the exact documents and code we need to create to finish the project. Since an iterative development allows for this level of documentation and feedback on the documentation/prototyping, it will be easier to plan for subsets of projects to be completed by different group members.

- **Goal: High-Bandwidth Communication**
  While scheduled meetings and a general dialog between members will help with obtaining good communication, it will be helpful to go back to the design documentation in the inception phase of the iterative cycle to affirm each ones' roles.

- **Constraint: Scope**
  The majority of the project's scope will be determined in the inception phase of our iterative development cycle, and will be iterated on throughout feedback cycles.

- **Constraint: Schedule**
  An iterative cycle will allow us to plan our phases of development within the final deadline in mind so that we can best determine if we are on schedule or not.

- **Constraint: Resources**
  Since we will have a continuous cycle of feedback of our prototypes, analysis, and code, we can make the best use of our resources (hours spent on the project) by minimizing the amount of time fixing errors. Time fixing errors will be minimized because of the shortened amount of time required to fix previous documents since the last round of feedback.

1.2 Iterative Development & Project Risks

- **Risk: Meeting Deadlines**
  An iterative process allows for planning out each phase of the iteration so that we can determine if we are off-schedule quickly.

- **Risk: Miscommunication**
  An iterative process will create many design documents in the inception of the project, which should minimize this risk of miscommunication as group members will have many resources to go back to determine what they should be doing (Group messaging, SRS, SDS, and User/Coding handbooks).

- **Risk: Improper Requirements Analysis**
  This risk should be minimized by the iterative development because we will collect certain prototypes such as HTAs and User/Activity scenarios that we will get feedback on early in the development cycle.

2 Other Process Models

- **Waterfall Method:**
  Software development happens too late with the waterfall method, which creates an instance where errors would be too costly to fix. It is more difficult to determine the correct amount of time spent on each stage as this is our first project working together as a group, and because each stage needs to be completed before moving onto the next.

- **Prototyping Method:**
  This process is not as complete as an iterative development cycle, and many of the advantages created by this process are found in the iterative cycle itself.

- **Spiral Method:**
  While there are risks to every software project, we do not believe that this project requires this level of emphasis on risk.

- **Agile Method:**
  The Agile method would not suit the needs of project 1 as it does not emphasize documentation and the amount of planning that is required of project 1's grade. Since this is also our first group's project together, we do not know the required communication and scheduling that would be necessary for this developmental process to work well.