# Development Process

The first iteration should focus on configuration of a client installation.

1. Create the use cases for each sensor type. Identify and describe each scenario as a use case, describing how the hub sends sensor events.
2. Create class diagrams using the use cases to identify key objects and attributes. Examples: Client, Hub, Sensor. Create sub-classes specific to each sensor type.
3. Create Python classes for each of the domain objects in the class diagrams.
4. Create MySQL tables matching domain objects.
5. Create ORM persistence from domain objects to SQL tables.
6. Create unit tests to test Python classes and ORM persistence.
7. Review.

The second iteration should focus on creating events when significant sensor state transitions are detected.

1. Identify sensor state transistions and messages sent.
2. Create test harness to generate Sensor messages to be read by message reader, which will generate event objects and persist these using ORM.
3. Create Python code to create state transitions and generate alerts/events when a significant sensor state transition takes place.
4. Create tests to simulate each state transition. Validate.
5. Review.

**Development Environment**

The development environment should include the following, with secure access provided to team members:

1. Specify development tools for Python / Django development. PyCharm is recommended.
2. Setup phpBB forums for supporting discussions and feedback.
3. Setup repository for use cases and class diagrams.
4. Setup Linux development server, Django server and MySQL database for deployment of integrated code and tests.
5. Create web page or wiki page that points to development artifacts and describes deployment details. Should also contain links to Climax documentation.
6. Setup GIT repository for source code. Specify layout of directories and document.
7. Setup Phabricator for code reviews.
8. Create a nightly backup of all artifacts and verify this is working.