Software Engineering – Fall 2015
(CSC 4350/6350)
TR. 5:30 pm – 7:15 pm

Rao Casturi
09/17/2015
http://cs.gsu.edu/~ncasturi1
Requirement Elicitation
Requirement Engineering

• **First step** for understanding the “System”
• **Very Critical** for the System Success
• 2 Components
  – Requirement Specification (Gathering)
  – Analysis Model
Requirement Engineering

Key Ideas:

– This is about the communication among the resources working on the project
– Human interaction
– Story board about the System Proposed
– Understanding the problem and articulate it back
– Ability to establish the boundaries
Nonfunctional requirements:
This activity describe aspects of the system that are not directly related to the functional behavior of the system. Nonfunctional requirements include a broad variety of requirements that apply to many different aspects of the system, from usability to performance.

Functional requirements: This describe the interactions between the system and its environment independent of its implementation. The environment includes the user and any other external system with which the system interacts.
Goal of this activity is to capture all the “Shall” statement sentences from the requirement documents as project requirements as a tabular form.

End of this activity the “Requirements Trace Matrix” is a deliverable to the project team.

Why is this activity important?
- Scope Creep
- Foundation for next phases (Use Cases, Categories, Classes, Methods)
- Cost & Time lines of project
- Future development (Incremental) Issues
Functional Requirements

SatWatch is a wrist watch that displays the time based on its current location. SatWatch uses GPS satellites (Global Positioning System) to determine its location and internal data structures to convert this location into a time zone.

The information stored in SatWatch and its accuracy measuring time is such that the watch owner never needs to reset the time. SatWatch adjusts the time and date displayed as the watch owner crosses time zones and political boundaries. For this reason, SatWatch has no buttons or controls available to the user.

SatWatch determines its location using GPS satellites and, as such, suffers from the same limitations as all other GPS devices (e.g., inability to determine location at certain times of the day in mountainous regions). During blackout periods, SatWatch assumes that it does not cross a time zone or a political boundary. SatWatch corrects its time zone as soon as a blackout period ends.

SatWatch has a two-line display showing, on the top line, the time (hour, minute, second, time zone) and on the bottom line, the date (day, date, month, year). The display technology used is such that the watch owner can see the time and date even under poor light conditions.

When political boundaries change, the watch owner may upgrade the software of the watch using the WebifyWatch device (provided with the watch) and a personal computer connected to the Internet.
Non-Functional Requirements

Quality requirements for SatWatch

• Any user who knows how to read a digital watch and understands international time zone abbreviations should be able to use SatWatch without the user manual. [Usability requirement]
• As the SatWatch has no buttons, no software faults requiring the resetting of the watch should occur. [Reliability requirement]
• SatWatch should display the correct time zone within 5 minutes of the end of a GPS blackout period. [Performance requirement]
• SatWatch should measure time within 1/100th second over 5 years. [Performance requirement]
• SatWatch should display time correctly in all 24 time zones. [Performance requirement]
• SatWatch should accept upgrades to its onboard via the Webify Watch serial interface. [Supportability requirement]

Constraints for SatWatch

• All related software associated with SatWatch, including the onboard software, will be written using Java, to comply with current company policy. [Implementation requirement]
• SatWatch complies with the physical, electrical, and software interfaces defined by WebifyWatch API 2.0. [Interface requirement]
 Requirement Elicitation Concepts:

- Functional Requirements
  - MUST to have
  - Interactions with the system

- Non Functional Requirements
  - Usability
  - Reliability
  - Robustness
  - Supportability
  - Performance

- Greenfield, Re Engineering, Interface Development
  - From Start
  - Develop or work on existing system
    - Build UI/Interfaces

- Completeness, Consistency, Clarity, Accuracy
  - Complete with all variations
  - Unambiguous
  - Correct

- Realism, Verifiability, Traceability
  - Should be realistic
  - Able to verify the outcome
  - Should match with the original requirements
Requirement Elicitation Activities:

• Identify the Actors (Roles)
  – During this phase the system developers will try to layout the various users and their roles
  – This helps understanding of the Application Domain

• Identify Scenarios:
  – Have discussions on some example how the system will work
  – Helps in understanding the system better

• Identify Use Case:
  – This is a detailed document of the various scenarios
  – Capture all the possible functions the system should able to perform
Requirement Elicitation Activities:

• Refine the Use Cases:
  – This phase is utilized to refine and eliminate any of the duplicate
  – Re verify that all the requirements re captured and each one has its own Use Case

• Identify the Relations:
  – During this phase the relationship between the various Use Cases
  – Dependencies are identified
  – Identify the common functions in the system

• Identify the Non-Functional Requirements
  – Get a confirmation from the client
  – Document the non functional requirements
Requirement Elicitation Activities: (Actor)

- Actors are **Role Abstraction** - not necessary to map to a person.
- One Actor can take multiple roles.
- Subsystems can be Actors.

**Questions for identifying actors**

- Which user groups are supported by the system to perform their work?
- Which user groups execute the system’s main functions?
- Which user groups perform secondary functions, such as maintenance and administration?
- With what external hardware or software system will the system interact?
### Requirement Elicitation Activities: (Scenarios)

- Scenario is a description of the behavior of an ACTOR
- Single Actor View point
- Can’t replace an Use Case
- Types (As-Is, Futuristic, Testing and Evaluation, Training)

#### Questions for identifying scenarios

- What are the tasks that the actor wants the system to perform?
- What information does the actor access? Who creates that data? Can it be modified or removed? By whom?
- Which external changes does the actor need to inform the system about? How often? When?
- Which events does the system need to inform the actor about? With what latency?
Aggregation of the Scenarios
Use Case is initiated by an ACTOR
Use Case can interact with other Actors
Writing Use Case is an ART

Best Practices of Use Case
✓ Number each Use Case Uniquely
✓ Name the User Case to represent the ACTION (Verb)
✓ Name the ACTORS with the Role (Noun)
✓ Exit and Entry conditions should be clear
✓ Don’t write about the user interface
✓ Exceptions should be outlined clearly
✓ Flow of activity should be clear and numbers for easy identification
Requirement Elicitation Activities: (Refine Use Cases)

- Goal: Completeness and Correctness
- Any missing details to be noted and added to the User Case
- Go over with the client if possible.
- Let an outsider read the Use Case

Best Practices of Use Case Refine

✓ More eyes are better.
✓ Run through with fine tooth comb on scenarios & User Cases
Requirement Elicitation Activities: (Relationships)

- Goal: Find the Relationship between ACTORs and Use Cases
- This will reduce the complexity and makes it easy for developers
- Identify the Common and Exceptional scenarios
- Reduce redundancy

Best Practices of Relationships

☑ Identify common scenarios or user cases
☑ EXTEND for rarely used scenario or use case
☑ INCLUDE when the scenario is used in more than one use case
☑ Don’t over use.
Identifying Terminology / Glossary

- Participating objects should have a name
- Terminology is the first step towards the Analysis
- Need to be updated regularly

Best Practices of Glossary

✓ User Application Domain Names
✓ Clarify any Technical Terms
✓ Data sources to be mentioned
Requirement Elicitation Activities: Non Functional

- **Product**
  - Performance
  - Security
  - Dependability
  - Usability

- **Organizational**
  - Operational
  - Environment
  - Development Standards

- **External**
  - Legal
  - Accounting Standards
  - Regulatory
  - Ethical
• RAD (Requirement Analysis Document)
  1. Introduction
  2. Current System
  3. Future System
  4. System Model
  5. High Level Time Line Diagrams
  6. Appendix (Proto type, Screen Shots etc)
  7. Glossary
Problem Statement
RC University Management Board approved a new Student Registration System to enable the online course requirement for the Computer Science Department. The implementation of the new system is proposed to be in place by 2015 Spring. The new system is called SRS.

What is expected by the SRS:

The Student Registration System (SRS) shall include the ability for any accepted student to view his or her classes for a given semester. The SRS shall permit any active student to add new classes or modify existing classes. The SRS shall give the users to view the student records. The SRS will be able to print the records if needed. The SRS shall give the ability for the administrator to modify the student records for the given semester. The SRS shall log the history. The Course coordinator or department assigned person can view the student educational progress reports. SRS shall provide the functionality to pay the student dues. SRS will be used to capture student grades on the specific classes taken by the student.

The SRS will run on any standard browser with standard user authentication. SRS will not accept any payments directly but directed to a third party vendor to accept the payments by credit cards only. The data will be updated on SRS from the payment system by end of every day 8:00 pm.
**Requirement Traceability Matrix**

- Simple Tabular view of the requirements collected
- Living document over the project life

<table>
<thead>
<tr>
<th>Entry #</th>
<th>Para #*</th>
<th>HCC Requirements Traceability Matrix</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.0</td>
<td>The SEM shall monitor all occupied living quarters.</td>
<td>SW</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>There shall be a total of 48 living quarters.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>Air Pressure, Temperature, and Oxygen % shall be obtained once per minute.</td>
<td>SW, P</td>
</tr>
<tr>
<td>4</td>
<td>2.1</td>
<td>There shall be three sensors in each living quarter.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>5</td>
<td>2.1</td>
<td>There shall be one sensor for each environmental condition.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>6</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 1% but &lt; 2% from the nominal value shall cause the window in the panel to be lit.</td>
<td>SW</td>
</tr>
<tr>
<td>7</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 2% but &lt; 3% from the nominal value shall cause the window in the panel to be lit and flash two times per second.</td>
<td>SW</td>
</tr>
<tr>
<td>8</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 3% from the nominal value shall cause the window in the panel to be lit and flash four times per second and sound an audible alarm.</td>
<td>SW</td>
</tr>
<tr>
<td>9</td>
<td>2.2</td>
<td>There shall be one audible alarm.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>The panel shall accommodate 48 annunciators arranged in six rows.</td>
<td>SW⁴</td>
</tr>
<tr>
<td>11</td>
<td>3.0</td>
<td>Each annunciator shall be mapped to a unique living quarter.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>12</td>
<td>3.0</td>
<td>Each annunciator shall accommodate three windows.</td>
<td>SW⁵</td>
</tr>
</tbody>
</table>

Source: Use Cases combined with BOOCH OMT UML – Putnam Tecxel & Charles Williams
## RTM Format

<table>
<thead>
<tr>
<th>Entry Number</th>
<th>Paragraph #</th>
<th>Requirement</th>
<th>Type</th>
</tr>
</thead>
</table>

GSU: Software Engineering - CSC4350/6350
- Rao Casturi
Case Study Problem Statement: Habitat Control Center (HCC)

HCC-1.0 INTRODUCTION

A Habitat Control Center (HCC) is to contain 48 living quarters and a software system, named Sealed Environment Monitor (SEM), which is to act as a monitor of all the living quarters for the habitat personnel in the HCC.

HCC-2.0 SEALED ENVIRONMENT MONITOR (SEM)

The SEM shall monitor all occupied living quarters. There shall be a total of 48 living quarters, each of which may, or may not, be occupied at any one time.

HCC-2.1 MONITORED DATA

For each occupied living quarter, the SEM shall obtain, once per minute, the following data:

- Current air pressure (pounds per square inch)
- Current temperature (degrees Fahrenheit)
- Current oxygen level (as a percentage).

This information shall be obtained from three sensors that are located inside each living quarter. There shall be one sensor per environmental condition.

---

It is important to note that the authors do not have access to actual alarms and sensors, consequently the interfaces are emulated in the software.
HCC-2.2 ALARM CONDITIONS

For each of the items in Paragraph HCC-2.1, the SEM shall immediately react to the following situations as indicated:

- For a changed value that represents a deviation of $\geq 1\%$ but $<2\%$ from the nominal values found in the database, the appropriate window in the panel shall be lit.
- For a changed value that represents a deviation of $\geq 2\%$ but less than $3\%$ from the nominal values found in the database, the appropriate window in the panel shall be lit and flash at a rate of two times per second.
- For a changed value that represents a deviation of $\geq 3\%$ from the nominal values found in the database, the appropriate window in the panel shall be lit and flash at a rate of four times per second. Additionally, an audible alarm shall be sounded.

There shall be only one audible alarm.

HCC-3.0 ANNUCNIATOR PANEL

The annunciator panel is located in the control center of the sealed habitat. The panel of annunciators shall consist of 48 annunciators, arranged in six rows (A–F) of eight annunciators in each row (numbered 1–8, respectively). Additionally, each annunciator shall be mapped to a unique living quarter by the location in the annunciator panel. For example, annunciator C-5 corresponds to living quarter C-5. Figure HCC-1 depicts the envisioned panel display.

Each annunciator in the panel shall be composed of three parts: an air pressure warning window, an oxygen warning window, and a temperature warning window. Each window shall be identified by an appropriate legend. Figure HCC-2 depicts a typical annunciator display that is composed of three windows, one for each environmental condition.

HCC-4.0 NOMINAL VALUES

All nominal values shall be found in the database. There shall be an option for an Operator to redefine the values of the environmental conditions maintained in the database. The Operator shall be able to redefine all three environmental nominal values to be used for all living quarters and apply this change to SEM processing.

HCC-5.0 ALARMS

The audible alarm-sounding and window-flashing features shall be only turned on by the SEM. The SEM shall continue to turn on these warning indicators as long as the alarm condition continues to exist. The warning indicators shall only be turned off by the Operator.
How do we complete this activity?

- Simple steps to follow to complete the activity
  - Agree upon a set of initial documents to extract the requirements
  - Look for any sentence with “SHALL” word
  - Any sentence with “Will” be extracted and discussed further
  - Prepare the Requirements Trace Matrix (RTM)

Example of this activity

HCC-1.0 INTRODUCTION

A Habitat Control Center (HCC) is to contain 48 living quarters and a software system, named Sealed Environment Monitor (SEM), which is to act as a monitor of all the living quarters for the habitat personnel in the HCC.

HCC-2.0 SEALED ENVIRONMENT MONITOR (SEM)

The SEM shall monitor all occupied living quarters. There shall be a total of 48 living quarters, each of which may, or may not, be occupied at any one time.

Facts but not in a form of requirements

First set of "Shall" identified with an unique line entry
- Req. Entry Number
- Paragraph Number
- Requirement sentence

- Number the paragraphs of the agreed upon document
- Any sentence can now be identified by the paragraph #
- Scan the sentences for SHALL and note them and number them as individual items on an EXCEL Spreadsheet or any tool which can capture as individual items with Sorting facility
First set of “Shall” identified with an unique line entry
- Req. Entry Number
- Paragraph Number
- Requirement sentence

Some Definitions
- **Shall** statement is a conceptual requirement for the complete system development
- **Requirements Trace Matrix (RTM)** an organized list of system requirements gathered during the initial phase from the system specification or requirement documents.
- **Derived Requirements** where the “Shall” can’t be seen but can still be a requirement for the overall project or system

<table>
<thead>
<tr>
<th>Entry #</th>
<th>Paragraph #</th>
<th>Requirements Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.0</td>
<td>The SEM shall monitor all occupied living quarters</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>There shall be a total of 48 living quarters</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>There shall be 3 sensors in each living quarter</td>
</tr>
<tr>
<td>4</td>
<td>2.1</td>
<td>There shall be 1 sensor for each environmental condition</td>
</tr>
</tbody>
</table>
All Windows for the same environmental condition will be same.

Each environmental condition will be represented by a different color.

Note:
- The “Will” sentences usually grouped together either at the end or with the appropriate section with the order.
- Capture Derived Requirements without any paragraph number.
### What is the final deliverable?

What is the final deliverable? Requirements Traceability Matrix

#### Derived Requirements

**HCC Requirements Traceability Matrix**

<table>
<thead>
<tr>
<th>Entry #</th>
<th>Para #</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.0</td>
<td>The SEM shall monitor all occupied living quarters.</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>There shall be a total of 48 living quarters.</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>Air Pressure, Temperature, and Oxygen % shall be obtained once per minute.</td>
</tr>
<tr>
<td>4</td>
<td>2.1</td>
<td>There shall be three sensors in each living quarter.</td>
</tr>
<tr>
<td>5</td>
<td>2.1</td>
<td>There shall be one sensor for each environmental condition.</td>
</tr>
<tr>
<td>6</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 1% but &lt;2% from the nominal value shall cause the window in the panel to be lit.</td>
</tr>
<tr>
<td>7</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 2% but &lt; 3% from the nominal value shall cause the window in the panel to be lit and flash two times per second.</td>
</tr>
<tr>
<td>8</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 3% from the nominal value shall cause the window in the panel to be lit and flash four times per second and sound an audible alarm.</td>
</tr>
<tr>
<td>9</td>
<td>2.2</td>
<td>There shall be one audible alarm.</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>The panel shall accommodate 18 annunciators arranged in six rows.</td>
</tr>
<tr>
<td>11</td>
<td>3.0</td>
<td>Each annunciator shall be mapped to a unique living quarter.</td>
</tr>
<tr>
<td>12</td>
<td>3.0</td>
<td>Each annunciator shall accommodate three windows.</td>
</tr>
<tr>
<td>13</td>
<td>3.0</td>
<td>Each window shall be identified with an appropriate legend.</td>
</tr>
<tr>
<td>14</td>
<td>4.0</td>
<td>All nominal values shall be found in the database.</td>
</tr>
<tr>
<td>15</td>
<td>4.0</td>
<td>The Operator shall be capable of reding the nominal values in the database.</td>
</tr>
<tr>
<td>16</td>
<td>4.0</td>
<td>The Operator shall be capable of reding all three nominal values in the database to be used for all living quarters.</td>
</tr>
<tr>
<td>17</td>
<td>4.0</td>
<td>The Operator shall be able to redefine the nominal value for a specific living quarter.</td>
</tr>
<tr>
<td>18</td>
<td>5.0</td>
<td>Only the software shall turn on the audible alarm.</td>
</tr>
<tr>
<td>19</td>
<td>5.0</td>
<td>Only the software shall flash a window.</td>
</tr>
<tr>
<td>20</td>
<td>5.0</td>
<td>The software shall continue to turn on the warning indicators as long as the alarm condition exists.</td>
</tr>
<tr>
<td>21</td>
<td>5.0</td>
<td>The Operator shall turn the audible alarm off.</td>
</tr>
<tr>
<td>22</td>
<td>5.0</td>
<td>The Operator shall turn the window off.</td>
</tr>
<tr>
<td>23</td>
<td>3.0</td>
<td>Each environmental condition will be represented by a different color.</td>
</tr>
<tr>
<td>24</td>
<td>3.0</td>
<td>All windows for the same environmental condition will be the same color.</td>
</tr>
<tr>
<td>25</td>
<td>—</td>
<td>Because the Operator can reset the nominal values in the DU, he/she needs to be able to reset the SEM to operate on those new values.</td>
</tr>
<tr>
<td>26</td>
<td>—</td>
<td>The SEM does not need to monitor unoccupied living quarters, so the Operator can set living quarters to occupied/unoccupied to enable the SEM to operate more efficiently.</td>
</tr>
</tbody>
</table>

Only One per project Requirements Traceability Matrix

GSU: Software Engineering - CSC4350/6350 - Rao Casturi

29
### What is Type Column?

#### Pre-agreed categories

- **Software Requirement (SW)**
- **Hardware Requirement (HW)**
- **Derived Requirement (DR)**
- **Nice to have Requirement (NTH)**
- **Software Constraint (SWC)**

#### Table: HCC Requirements Traceability Matrix

<table>
<thead>
<tr>
<th>Entry #</th>
<th>Para #</th>
<th>Requirement</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.0</td>
<td>The SEM shall monitor all occupied living quarters.</td>
<td>SW</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>There shall be a total of 48 living quarters.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>3</td>
<td>2.1</td>
<td>Air Pressure, Temperature, and Oxygen % shall be obtained once per minute.</td>
<td>SW, P</td>
</tr>
<tr>
<td>4</td>
<td>2.1</td>
<td>There shall be three sensors in each living quarter.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>5</td>
<td>2.1</td>
<td>There shall be one sensor for each environmental condition.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>6</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 1% but &lt;2% from the nominal value shall cause the window in the panel to be lit and flash two times per second.</td>
<td>SW</td>
</tr>
<tr>
<td>7</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 2% but &lt; 3% from the nominal value shall cause the window in the panel to be lit and flash two times per second.</td>
<td>SW</td>
</tr>
<tr>
<td>8</td>
<td>2.2</td>
<td>A current value representing a deviation &gt;= 3% from the nominal value shall cause the window in the panel to be lit and flash four times per second and sound an audible alarm.</td>
<td>SW</td>
</tr>
<tr>
<td>9</td>
<td>2.2</td>
<td>There shall be one audible alarm.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>The panel shall accommodate 48 annunciators arranged in six rows.</td>
<td>SW⁴</td>
</tr>
<tr>
<td>11</td>
<td>3.0</td>
<td>Each annunciator shall be mapped to a unique living quarter.</td>
<td>HW, SWC</td>
</tr>
<tr>
<td>12</td>
<td>3.0</td>
<td>Each annunciator shall accommodate three windows.</td>
<td>SW⁵</td>
</tr>
<tr>
<td>13</td>
<td>3.0</td>
<td>Each window shall be identified with an appropriate legend.</td>
<td>SW</td>
</tr>
<tr>
<td>14</td>
<td>4.0</td>
<td>All nominal values shall be found in the database.</td>
<td>SW</td>
</tr>
<tr>
<td>15</td>
<td>4.0</td>
<td>The Operator shall be capable of redefining the nominal values in the database.</td>
<td>SW</td>
</tr>
<tr>
<td>16</td>
<td>4.0</td>
<td>The Operator shall be capable of redefining all three nominal values in the database to be used for all living quarters.</td>
<td>SW</td>
</tr>
<tr>
<td>17</td>
<td>4.0</td>
<td>The Operator shall be able to redefine the nominal value for a specific living quarter.</td>
<td>SW</td>
</tr>
<tr>
<td>18</td>
<td>5.0</td>
<td>Only the software shall turn on the audible alarm.</td>
<td>SW</td>
</tr>
<tr>
<td>19</td>
<td>5.0</td>
<td>Only the software shall flash a window.</td>
<td>SW</td>
</tr>
<tr>
<td>20</td>
<td>5.0</td>
<td>The software shall continue to turn on the warning indicators as long as the alarm condition exists.</td>
<td>SW</td>
</tr>
<tr>
<td>21</td>
<td>5.0</td>
<td>The Operator shall turn the audible alarm off.</td>
<td>SW</td>
</tr>
<tr>
<td>22</td>
<td>5.0</td>
<td>The Operator shall turn the window off.</td>
<td>SW</td>
</tr>
<tr>
<td>23</td>
<td>3.0</td>
<td>Each environmental condition will be represented by a different color.</td>
<td>NTH</td>
</tr>
<tr>
<td>24</td>
<td>3.0</td>
<td>All windows for the same environmental condition will be the same color.</td>
<td>NTH</td>
</tr>
<tr>
<td>25</td>
<td>—</td>
<td>Because the Operator can reset the nominal values in the DB, he/she needs to be able to reset the SEM to operate on those new values.</td>
<td>SW, DR</td>
</tr>
<tr>
<td>26</td>
<td>—</td>
<td>The SEM does not need to monitor unoccupied living quarters, so the Operator can set living quarters to occupied/unoccupied to enable the SEM to operate more efficiently.</td>
<td>SW, DR</td>
</tr>
</tbody>
</table>
What is next Deliverable?
Due: 09/22/2015

1. Problem Statement with “Shall” statements
2. RTM (4 columns ID#, Paragraph#, Description of the requirement, Type)
3. Project Timeline (Gantt Chart with Tasks and allocation up to now)
4. Terminology
5. Rational for the project and the requirements

Presentations: Each Group Present their project to the Class
Questions?