Functional Requirements Comments. Some slides and examples Adapted from “Requirements Spec Revisited” (Dan Fleck of GMU)
Functional Requirements:

- A functional requirement is something the system must do.
- A functional requirement is testable
  - fold this activity right into the writing
- A general rule is: a functional requirement is a “shall statement”
In engineering, a requirement is a singular documented need of what a particular product or service should be or do. It is most commonly used in a formal sense in systems engineering or software engineering. It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user.

In the classical engineering approach, sets of requirements are used as inputs into the design stages of product development. Requirements show what elements and functions are necessary for the particular project.
• *Cut to Borland’s “Requirements Types” chart for a moment here.
  – just notice what is included
  – completeness of the picture
Functional Requirements

- These can be high level or low level (generally we’re at high level in this class)
  - High level: The system shall charge users credit cards for purchases
  - Low level: The system shall validate all passwords contain upper and lowercase characters and one number
Functional Requirements

- Are **testable**
  - your “test lead” should be collecting sets of basic tests generated for each requirement linked to the requirement
    - this is part of writing the requirement!
- Are things the system you are developing **must do**
- Should be **one thing** (not multiple). (Because a requirement is a single entity... it passes or fails as one piece)
- Should have a **source** (who/what decided this was required)
- Should have a **priority**
Functional Requirements

• Should **not be a design choice** (this is hard to get right).
  - The system shall store user information including name, DOB, address and SSN. **Good!**
  
  - The system shall store user information in an Oracle database including name, DOB, address, SSN. **bad**
    - Is Oracle really REQUIRED? Hard to say... maybe, but probably not. This is a decision you would make at implementation design time.
    - Question: Does the customer care that you use Oracle? MySQL? Etc.. Maybe someone found some other MUCH BETTER approach storing the data on moon rocks.
    - Again: This is hard to avoid... and I’m not to concerned with it on the SRS, but I want you to be very aware of when you are making design choices instead of required features.
Functional Requirements

- Must have a **unique** ID.
  - When testing you need to reference REQ-1 or REQ-287. Multiple things cannot be labeled REQ-1.
  - Later our test cases will say: This test case validates requirements REQ-1, REQ-27, and REQ-56.
Functional Requirements

• **Bad requirements examples (discuss):**
  – The system shall validate and accept credit cards and cashier’s checks. High priority.

  – The system shall process all mouse clicks very fast to ensure user’s do not have to wait.

  – The user must have Adobe Acrobat installed.

explanation on the next slide...
Functional Requirements

- **Bad requirements examples:**
  - The system shall validate and accept credit cards and cashier’s checks. High priority.
    - Problem: two requirements instead of one.
    - If the credit card processing works, but the cashier’s check validation does not... is this requirement pass or fail? Has to be fail, but that is misleading.
    - Maybe only credit cards are high priority and cashier’s checks are low priority.
  - The system shall process all mouse clicks very fast to ensure user’s do not have to wait.
    - Problem: This is not testable. Quantify how fast is acceptable?
  - The user must have Adobe Acrobat installed.
    - Problem: This is not something our system must do. It could be in the constraints/assumptions or maybe operating environment sections, but is not a functional requirement of our system.
What I want in the SRS

- A reference to the Use Case this requirement is from (for a very small number they may not have a use case.. But most will).
- The source of the use case - end users, marketing, sys admins, internal... think about who may care most about this use-case
- The event that starts the use case (if it’s in the use-case that is fine)
- Priority of the requirement
- A unique number for the more detailed requirement
- A short description of the more detailed requirement
Template

• 3.1 Withdraw money from ATM
  – 3.1.1. Description and priority
    – Customer withdraws money from the ATM. The system issues the money and debits their account. This feature is high priority.
  – 3.1.2. Stimulus/Response Sequence
    *Usually similar or the same as basic flow from the Use Case*
    1. Customer chooses the checking option on the ATM
    2. Customer chooses the amount of money needed
    3. Customer confirms the choice
    4. System validates the amount
    5. System debits the customer’s account
    6. System issues money to the user
  – 3.1.3. Functional Requirements
    – REQ 3.1 ...
    – REQ 3.2 ...
    – REQ 3.2 ...
Use Case

- **Use case name and identifier:** UC-3 - Withdraw money from ATM
- **Objective:** The customer withdraws money from the ATM and the system will debit the customer’s account.
- **Priority:** High
- **Source:** Customer
- **Actors:** Customer
- **Preconditions:** User is logged in. User account balance is nonzero.
- **Post conditions:** Money dispensed to the user and their account balance has been updated.
- **Flow of Events**
  1. Basic Flow
     1. Customer chooses the checking option on the ATM
     2. Customer chooses the amount of money needed
     3. Customer confirms the choice
     4. System validates the amount
     5. System debits the customer’s account
     6. System dispenses cash
  2. Alternative Flow 1: At step 4 amount is not a multiple of $20
     1. An error message is displayed telling customer they must use multiples of $20
     2. Return to step 1.2.
  3. Alternative Flow 2: At any step user presses “cancel”.
     1. System returns to the main menu
  4. Exception Flows:
     1. Storage is locked due to backup in progress. See U5 for details.
- **Includes** (other use case IDs): UC-5 - Exception occurs
- **Special Requirements:** None
- **Notes/Issues** - Need to figure out what happens if customer wants more than account balance, should machine dispense part of it and then display message? TBA-UC-3 [Owner: N. Patel, Due Date: 2/10/2009]
Requirements derived from this Use-Case

- Derived Functional Requirements:
  - Req U.3.1 The system shall provide an option to withdraw money
  - Req U.3.2 The system shall query the user for the amount of money
  - Req U.3.3 The system shall query the user for the account type
  - Req U.3.4 The system shall validate the amount is available in the user’s account before releasing funds to the user
  - Req U.3.5 The system shall validate the amount is a multiple of $20.
  - Req U.3.6 The system shall debit the user’s account upon withdrawal of funds
  - Req U.3.7 The system shall be able to issue a specific amount of money to the user.