Chapter 6 summary

**Categorizing or grouping the requirements**

* Business vs functional
* By viewing stakeholder
* Reusability

Most common requirement categories are:

* Business (what the business needs to do)
* Functional (how the business requirement is accomplished)
* Nonfunctional (system requirements)
* Technical (specificiations0

**Core Requirement Components (exist in all four categories of requirements)**

* Data (entities, attribute and relationships)
* Process (Use Case)
* External Agents (actors)
* Business Rules (constraints in which the business operates)

**Data**

 Relationships represent business rules

 Entities are things (nouns) customer, product, order

 Data is processed into information

 Attributes are the detail fields

 Are they unique, mandatory, repetitive

**Process Use Case**

 Start with a verb and include a noun

**External Agents (Actors)**

 Person organization of system that the business interacts with

 Actors are external to the software solution

 Important to determine internal vs external actors, external actors are not affected by the project where internal actors may have their work flows changed

**Business rules**

 Decision points

 Many rules are data related

**Analysis Techniques and presentation formats**

 **Glossary:** of terms, insure they are consistently used during conversations

 **Workflow Diagrams,** show how work is accomplished (UML Activity Diagram, flowcharts or process maps

 Standard shapes

Swim lanes

Can be used for documenting “As is” or “to be”

Can be detail or high level

Can be used for business or functional

 Business (what does the business do) Take order

Functional (how does the business do it) Answer phone call, write down items, create invoice

 **Entity Relationship Diagram:** represents data requirements

 This comprises the data model

 Logical data model

 Looking at the business processes from a different view.

 Logical data model facilitate reuse of data and remove of redundancy

 **Business Process modeling with Decomposition Diagram**

 Presentation of processes without sequence or relationship between them

 Looks like and organization chart with hierarchy built in

 Rules for building the diagram

 Parent child relationships only

 Only one component type (rules, entity, actor or rule), do not mix

 Every parent must have more than one child

 No sequence in represented, no arrows for direction

 **Use Case Diagram**

 Use case is a goal of the software

 How does the software interact with the users (actors)

 Actors are people, organizations or systems

 Use case is an oval

 Box shows the boundary of the software

 Lines are association between the actors and the software

Use case diagrams will have accompanying case descriptions which include a happy path interaction aka primary path

There may be redundancy when representing data and process

 **Prototype / Simulations (Justinmind)**

 Screen layouts

 Report layouts

 Data Entry layouts

 A story board is a presentation of a series of screens

 **Other Tools**

Event Modeling

 Entity State

 Object Modeling/Class Modeling

 User Stories

 Traceability Matrices

 CRUD Matrix

Gap Analysis

 Data Flow Diagram

Options for documenting requirements

 Text

 Graphics

 Text and Graphics

 Choose the options that best communicates to the specific user

**“AS is” vs “To Be”**

Deliverable to Project Matrix

 Deliverable to Audience Matrix

 Listen for facts vs Opinion

 Important to document “as is”, “what the problem is” and “recommend solution”