REQUIREMENTS

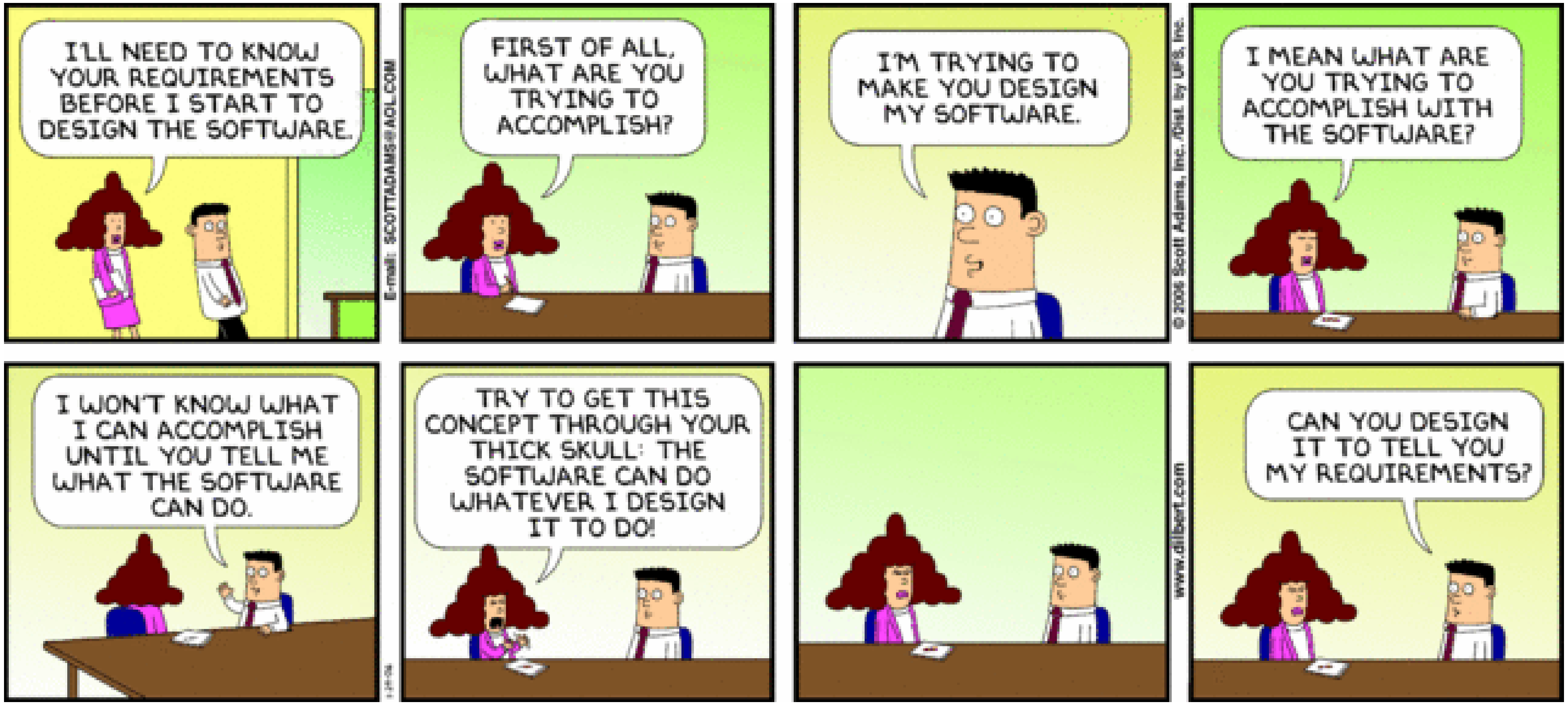
F. Tip and
M. Weintraub

UNIT OBJECTIVE

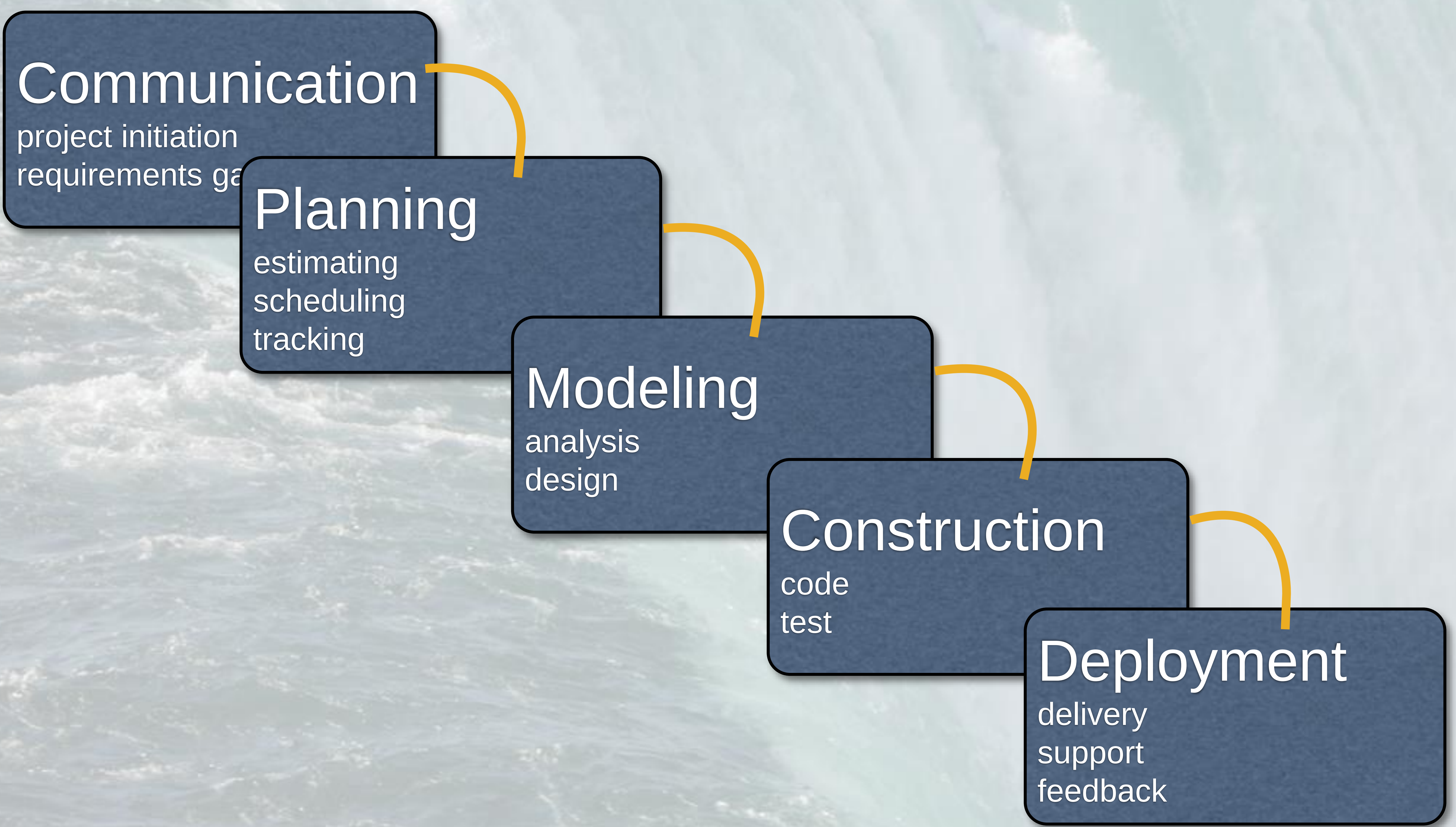
- Understand what requirements are
- Understand how to acquire, express, validate and manage requirements

Thanks go to Martin Schedlbauer and to Andreas Zeller for allowing incorporation of their materials

I'LL GO UP AND FIND OUT WHAT THEY NEED AND THE REST OF YOU START CODING !
SHELTON


I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH WITH THE SOFTWARE?
I WON'T KNOW WHAT I CAN ACCOMPLISH UNTIL YOU TELL ME WHAT THE SOFTWARE CAN DO.
TRY TO GET THIS CONCEPT THROUGH YOUR THICK SKULL: THE SOFTWARE CAN DO WHATEVER I DESIGN IT TO DO!
www.dilbert.com
CAN YOU DESIGN IT TO TELL YOU MY REQUIREMENTS?

WATERFALL MODEL (1968)



Communication
project initiation
requirements ga

Planning
estimating
scheduling
tracking

Modeling
analysis
design

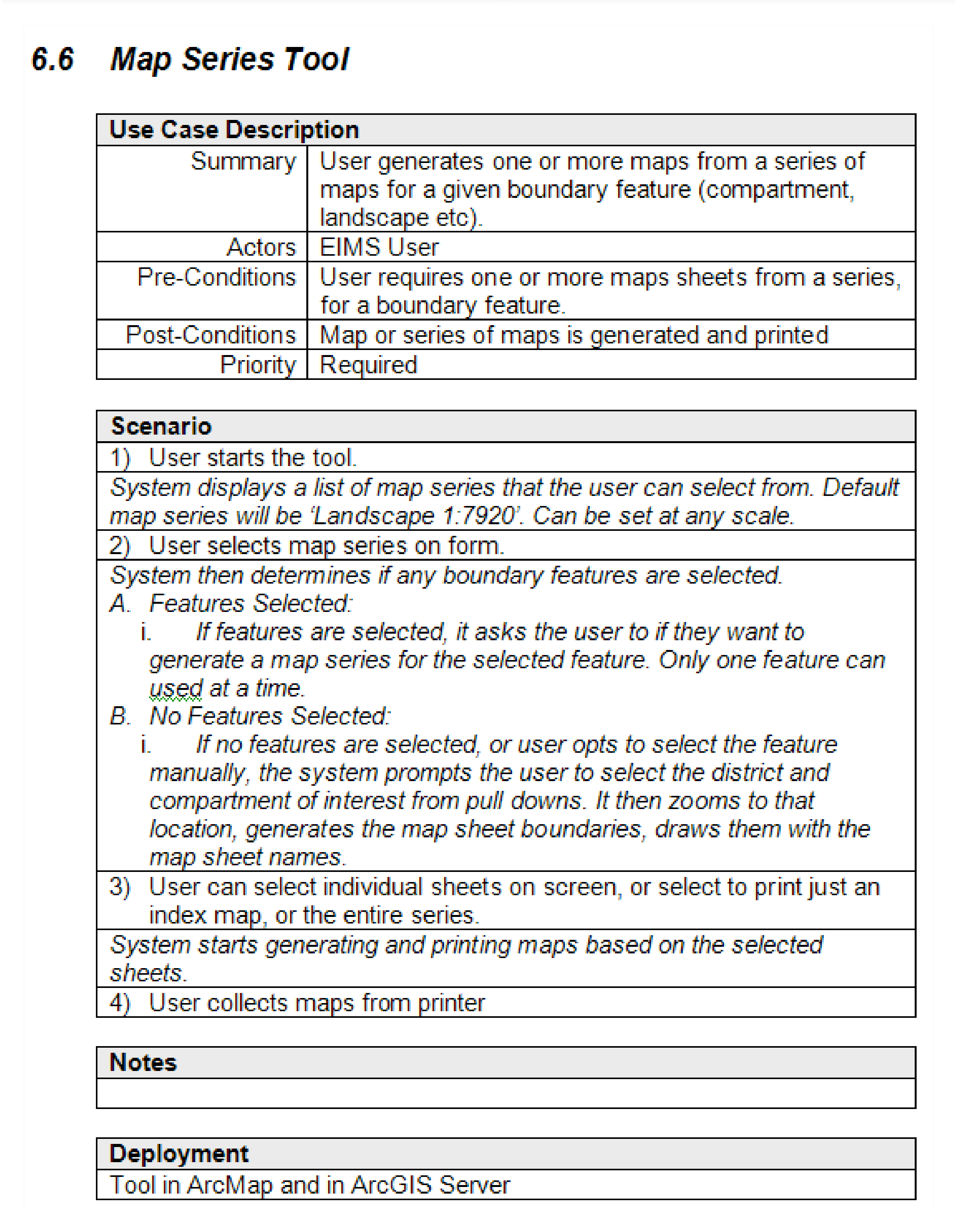
Construction
code
test

Deployment
delivery
support
feedback

COMMUNICATION

Communication

project initiation
requirements gathering



6.6 Map Series Tool

Use Case Description	
Summary	User generates one or more maps from a series of maps for a given boundary feature (compartment, landscape etc).
Actors	EIMS User
Pre-Conditions	User requires one or more maps sheets from a series, for a boundary feature.
Post-Conditions	Map or series of maps is generated and printed
Priority	Required

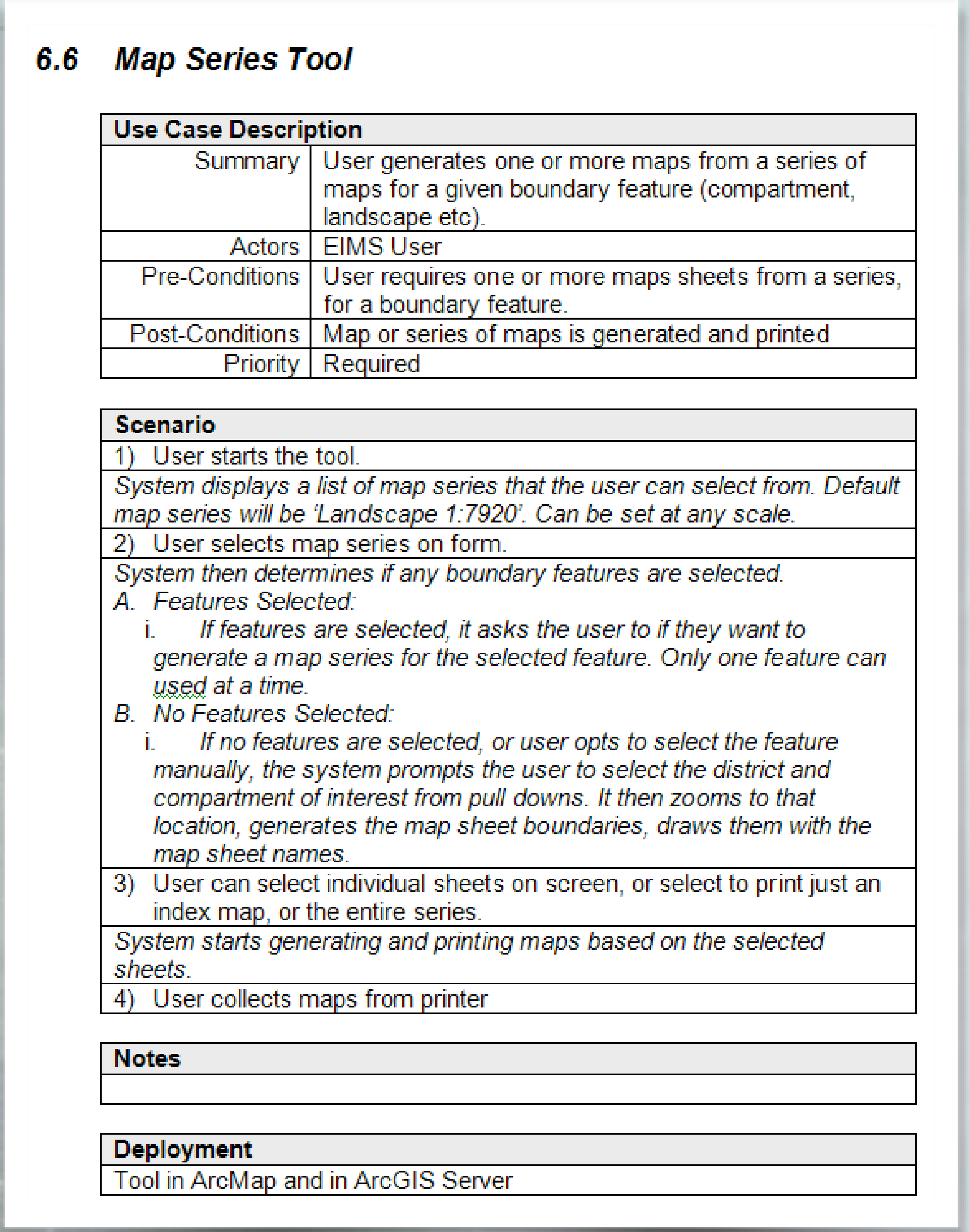
Scenario
1) User starts the tool.
System displays a list of map series that the user can select from. Default map series will be 'Landscape 1:7920'. Can be set at any scale.
2) User selects map series on form.
System then determines if any boundary features are selected. *A. Features Selected:* *i. If features are selected, it asks the user to if they want to generate a map series for the selected feature. Only one feature can used at a time.* *B. No Features Selected:* *i. If no features are selected, or user opts to select the feature manually, the system prompts the user to select the district and compartment of interest from pull downs. It then zooms to that location, generates the map sheet boundaries, draws them with the map sheet names.*
3) User can select individual sheets on screen, or select to print just an index map, or the entire series.
System starts generating and printing maps based on the selected sheets.
4) User collects maps from printer

Notes

Deployment
Tool in ArcMap and in ArcGIS Server

COMMUNICATION

How do we get there?



6.6 Map Series Tool

Use Case Description	
Summary	User generates one or more maps from a series of maps for a given boundary feature (compartment, landscape etc).
Actors	EIMS User
Pre-Conditions	User requires one or more maps sheets from a series, for a boundary feature.
Post-Conditions	Map or series of maps is generated and printed
Priority	Required

Scenario
1) User starts the tool.
System displays a list of map series that the user can select from. Default map series will be 'Landscape 1:7920'. Can be set at any scale.
2) User selects map series on form.
System then determines if any boundary features are selected. *A. Features Selected:* *i. If features are selected, it asks the user to if they want to generate a map series for the selected feature. Only one feature can used at a time.* *B. No Features Selected:* *i. If no features are selected, or user opts to select the feature manually, the system prompts the user to select the district and compartment of interest from pull downs. It then zooms to that location, generates the map sheet boundaries, draws them with the map sheet names.*
3) User can select individual sheets on screen, or select to print just an index map, or the entire series.
System starts generating and printing maps based on the selected sheets.
4) User collects maps from printer

Notes

Deployment
Tool in ArcMap and in ArcGIS Server

REQUIREMENT (ANSI/IEEE STANDARD 610.12-1990)

1. A condition or capability needed by a user to solve a problem or achieve an objective.

2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

3. A documented representation of a condition or capability as in (1) or (2).

A requirement is a description of a system feature, capability, or constraint and should **focus on what** a system should do **rather than how** it should or could be done

EXPRESSING REQUIREMENTS (ENGLISH) – RFC 2119

In many standards track documents several words are used to signify the requirements in the specification. These words are often capitalized. This document defines these words as they should be interpreted in IETF documents. Authors who follow these guidelines should incorporate this phrase near the beginning of their document:

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Note that the force of these words is modified by the requirement level of the document in which they are used.

1. MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

2. MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

3. SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

4. SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

5. MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

THE CHALLENGE WITH NATURAL LANGUAGES

- It’s expressive, intuitive, and universal

- It may be vague and ambiguous, and statements are open to reader interpretation

The assignment SHALL be due on the date assigned.

So, when is it due?

At day/mon/year hh:mm is reasonably clear

At day/mon/year is not

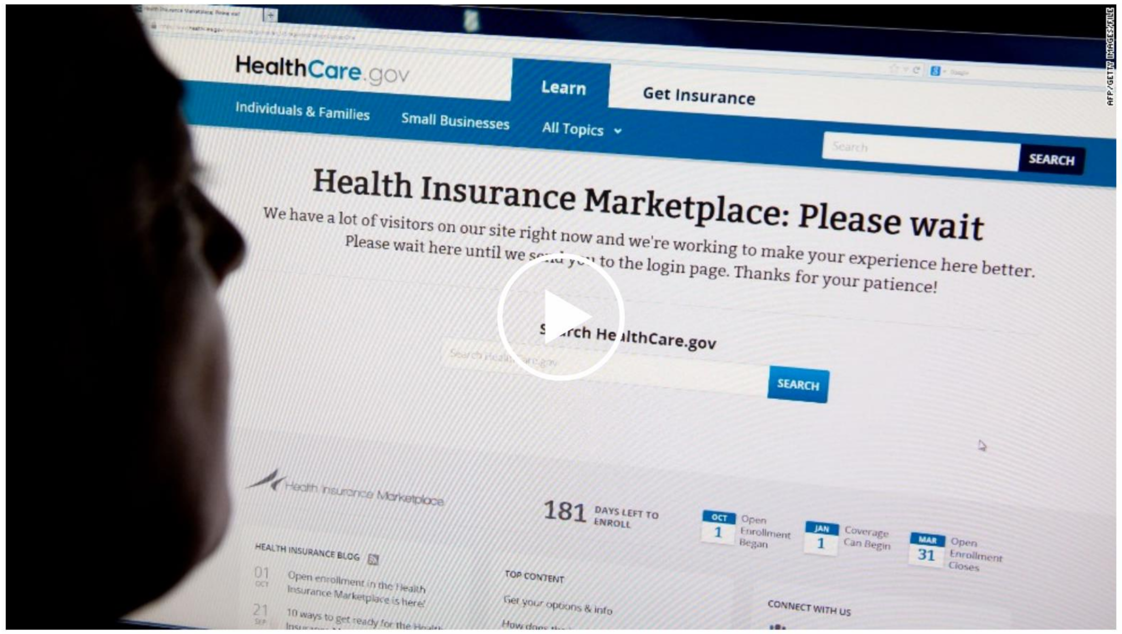
ANOTHER AMBIGUOUS EXAMPLE

I saw a man on the hill with a telescope.

1. *There's a man on a hill, and I'm watching him with my telescope.*
2. *There's a man on a hill, who I'm seeing, and he has a telescope.*
3. *There's a man, and he's on a hill that also has a telescope on it.*
4. *I'm on a hill, and I saw a man using a telescope.*
5. *There's a man on a hill, and I'm sawing him with a telescope*

SOFTWARE DISASTERS

DENVER AIRPORT



Health Insurance Marketplace: Please wait

We have a lot of visitors on our site right now and we're working to make your experience here better. Please wait here until we send you to the login page. Thanks for your patience!

Search HealthCare.gov

SEARCH
Health Insurance Marketplace
181 DAYS LEFT TO ENROLL
OCT 1 Open Enrollment Began
JAN 1 Coverage Can Begin
MAR 31 Open Enrollment Closes
HEALTH INSURANCE BLOG
01 OCT Open enrollment in the Health Insurance Marketplace is here!
21 SEP 10 ways to get ready for the Health Insurance
TOP CONTENT
Get your options & info
CONNECT WITH US

OBAMACARE

- Mariner 1 (1962)
  Rocket crash due to missing dash
- Eole 1 (1971)
  72 weather balloons get wrong cmd
- Nimbus 7 (1978)
  Satellite misses ozone hole for 6 yrs
- HMS Sheffield (1982)
  Exocet rocket id'ed as "friend"
- Stanislaw Petrow (1983)
  Russia detects global nuclear attack
- Therac 25 (1985)
  Radiation overdose kills six

- Stock crash (1987)
  Dow Jones loses 22% in one day
- Vincennes (1988)
  Passenger jet mistaken to be F-14
- Patriot (1991)
  Misses to shoot down Iraqi Scud
- Climate Orbiter (1999)
  Confuses metrics and imperial
- US Blackout (2003)
  50 mln affected for 5 days
- Apple SSL bug (2012)
  18 months w/o SSL authentication
- 3200 prisoners released early (2015)

- Nest Thermostat users left in the cold (2016)
- HSBC major outage (2016)
- Delta Airlines: power outage causes system-wide failure worldwide (2016)
- …

GLASS’ LAW

Requirement deficiencies
are the prime source
of project failures.

- ~45% of project failures involve requirements phase issues (Chaos Study)
  - Incomplete requirements (13%)
  - Lack of user involvement (12%)
  - Changing specifications (9%)
  - Unrealistic expectations (10%)

This and other laws are found in Endres/Rombach: Handbook of Software and Systems Engineering.

REQUIREMENTS SET THE STAGE FOR SUCCESS

- A requirement defines a commitment between the clients and the tech team for what the system needs to accomplish

- Risks
  1. Each individual understands the same statement differently

  2. Understanding what is actually needed is not clear
     1. *Real versus perceived needs*
     2. *Technology not appreciating difficulty, explicit or implied*

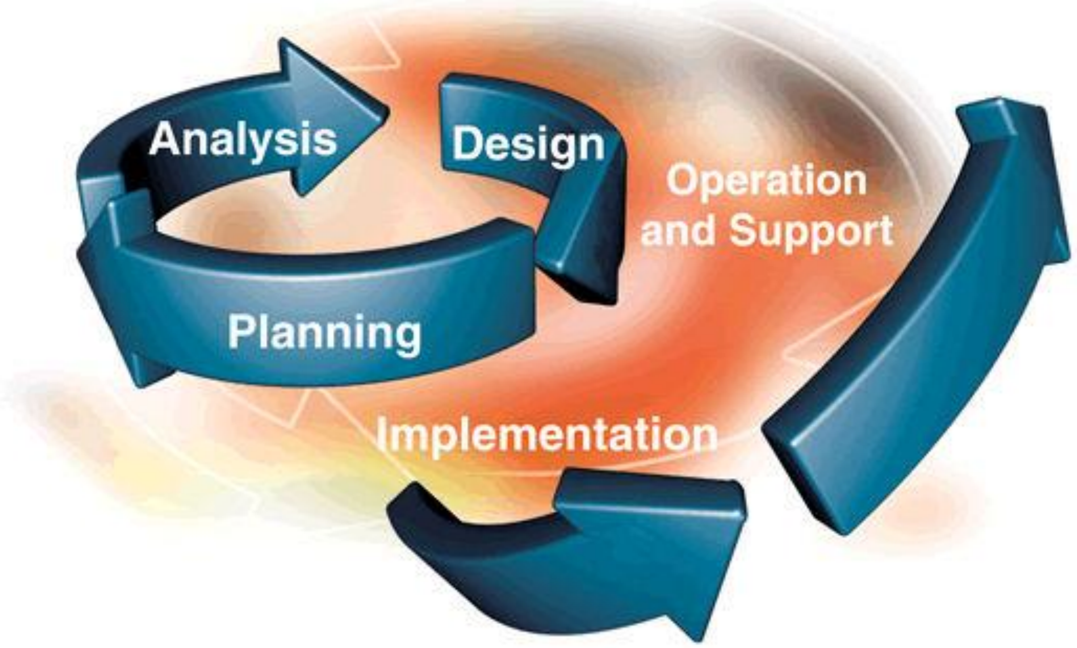
At the end of the day, it’s about what gets delivered. But if you don’t know where you are going, it’s hard to aim right. And then, the project is called research.

REQUIREMENTS ANALYSIS
ANSI/IEEE STANDARD 610.12-1990

- The process of studying user needs to arrive at a definition of system, hardware, or software requirements.
- The process of studying and refining system, hardware, or software requirements.

ANALYSIS VS DESIGN

- **Analysis** = **what** the software should do
  - Software functionality
  - Software properties



- **Design** = **how** it should do it

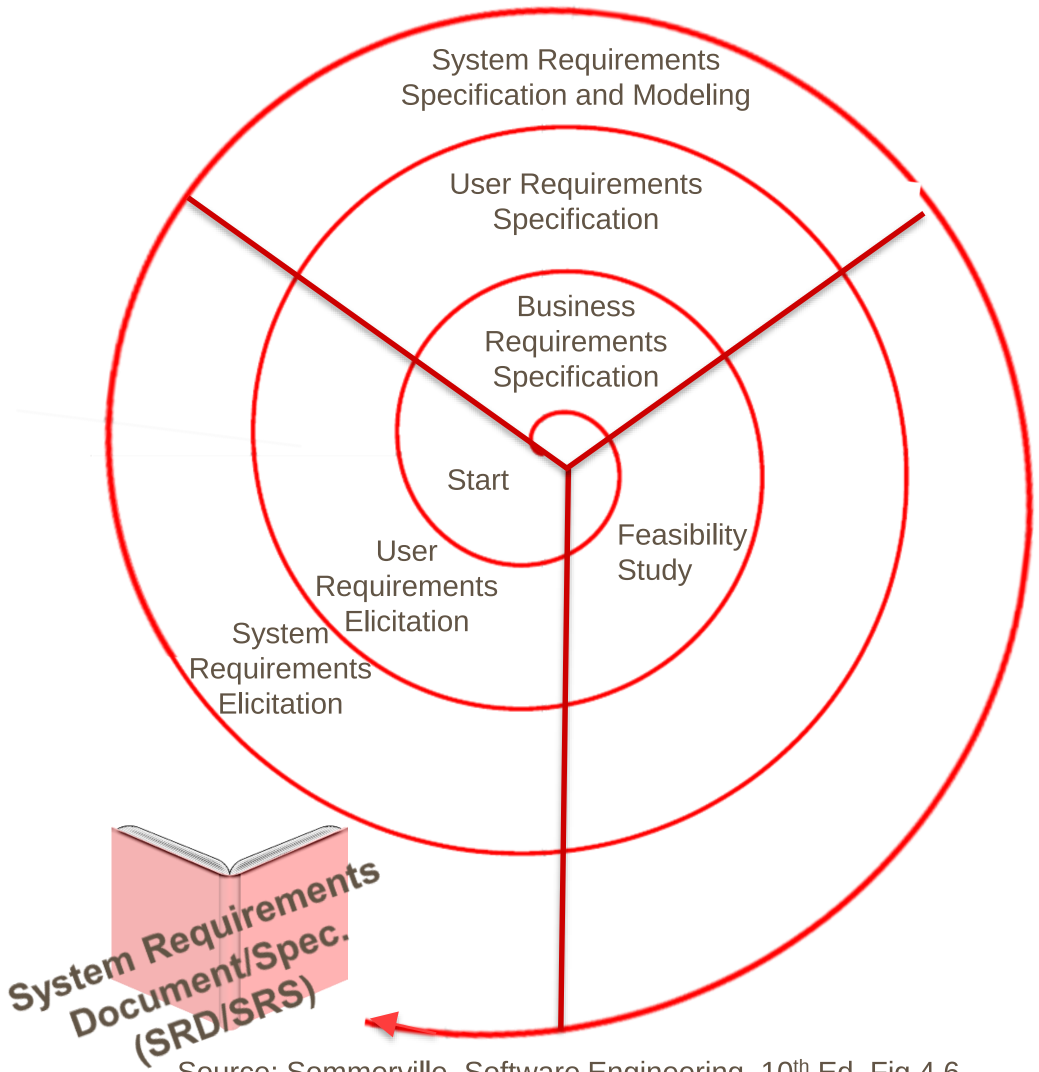
CLASSICAL ENGINEERING VIEWPOINT

- “We must know [exactly] what to build before we can build it”
- *Leads logically to waterfall process*
- … but is this realistic for today’s systems?

YOUR REQUIREMENTS DOCUMENT IS THE BIGGEST I'VE EVER SEEN.
scottadams@aol.com
www.dilbert.com
IT'S TOO BIG TO READ, BUT I CAN GUESS FROM ITS WEIGHT WHAT MUST BE IN THERE.
© 1999 United Feature Syndicate, Inc.
9/9/99
YOU KNOW IT'S A MULTI-USER, GLOBAL SYSTEM, RIGHT?
NO, I'M NOT GETTING THAT.

REQUIREMENTS ANALYSIS

- Identify Stakeholders
- Elicit Requirements
- Identify Requirements
- Prototypes


System Requirements Specification and Modeling
User Requirements Specification
Business Requirements Specification
Start
Feasibility Study
User Requirements Elicitation
System Requirements Elicitation
System Requirements Document/Spec. (SRD/SRS)

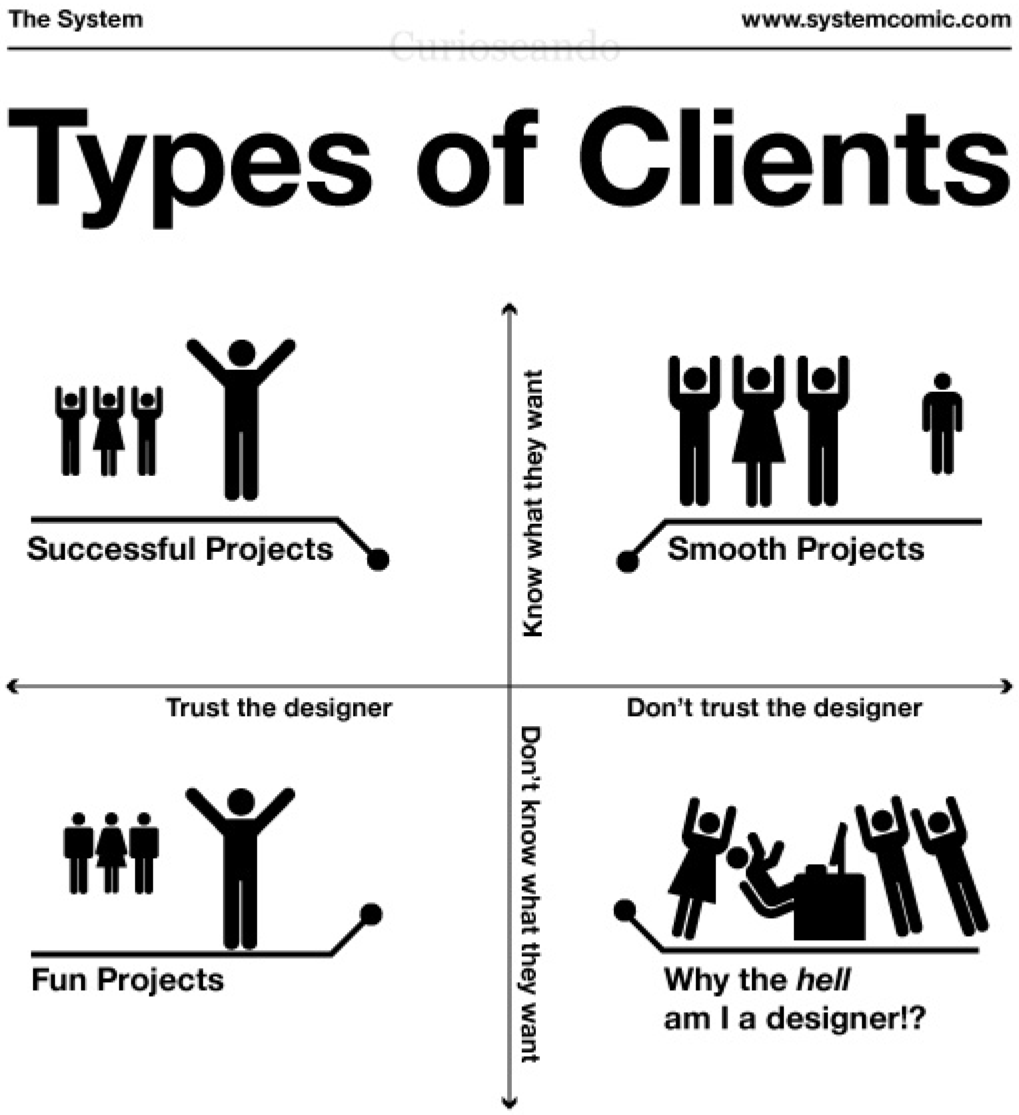
Source: Sommerville, Software Engineering, 10th Ed, Fig 4.6

STAKEHOLDERS

- Persons or organizations who…

1. Have a valid interest in the system

2. Are affected by the system



THERE ARE OFTEN MANY STAKEHOLDERS

1. Anyone who operates the system
   - Normal and maintenance operators
2. Anyone who benefits from the system
   - Functional, political, financial and social beneficiaries
3. Anyone involved in purchasing or procuring the system
4. Organizations that regulate some or all of the system
   - Financial, safety, or other regulators
5. Organizations responsible for systems that interface with the system under design
6. People or organizations opposed to the system

ELICIT REQUIREMENTS

- Interviews are the best way to elicit requirements
- Explore requirements systematically
- Sounds simple – but is the hardest part!

WHY IS ELICITATION HARD?

1. **Problems of scope**

   What is the boundary of the system? • What details are actually required?

2. **Problems of understanding**

   Users do not know what they want • don’t know what is needed • have a poor understanding of their computing environment • don’t have a full understanding of their domain • omit “obvious” stuff • are ambiguous

3. **Problems of volatility**

   Requirements change over time

4. **Problems of availability**

   People who know what is needed are usually in demand doing their job.

IDENTIFY REQUIREMENTS

1. Functional requirements
2. Nonfunctional requirements
3. Constraints
4. Contract-style requirements
5. Use cases (user stories)

TYPES OF REQUIREMENTS

FUNCTIONAL REQUIREMENTS

An action the product must take [to be useful].

It describes what the system should do.

1. The product SHALL track individual payments of coffee servings.
2. The product MUST heat water to 150F.

NON-FUNCTIONAL REQUIREMENTS

A *property* or *quality* the product must have.

1. The product shall be accessible in English and Spanish.
2. The product must be capable of serving 45 cups of coffee per hour.

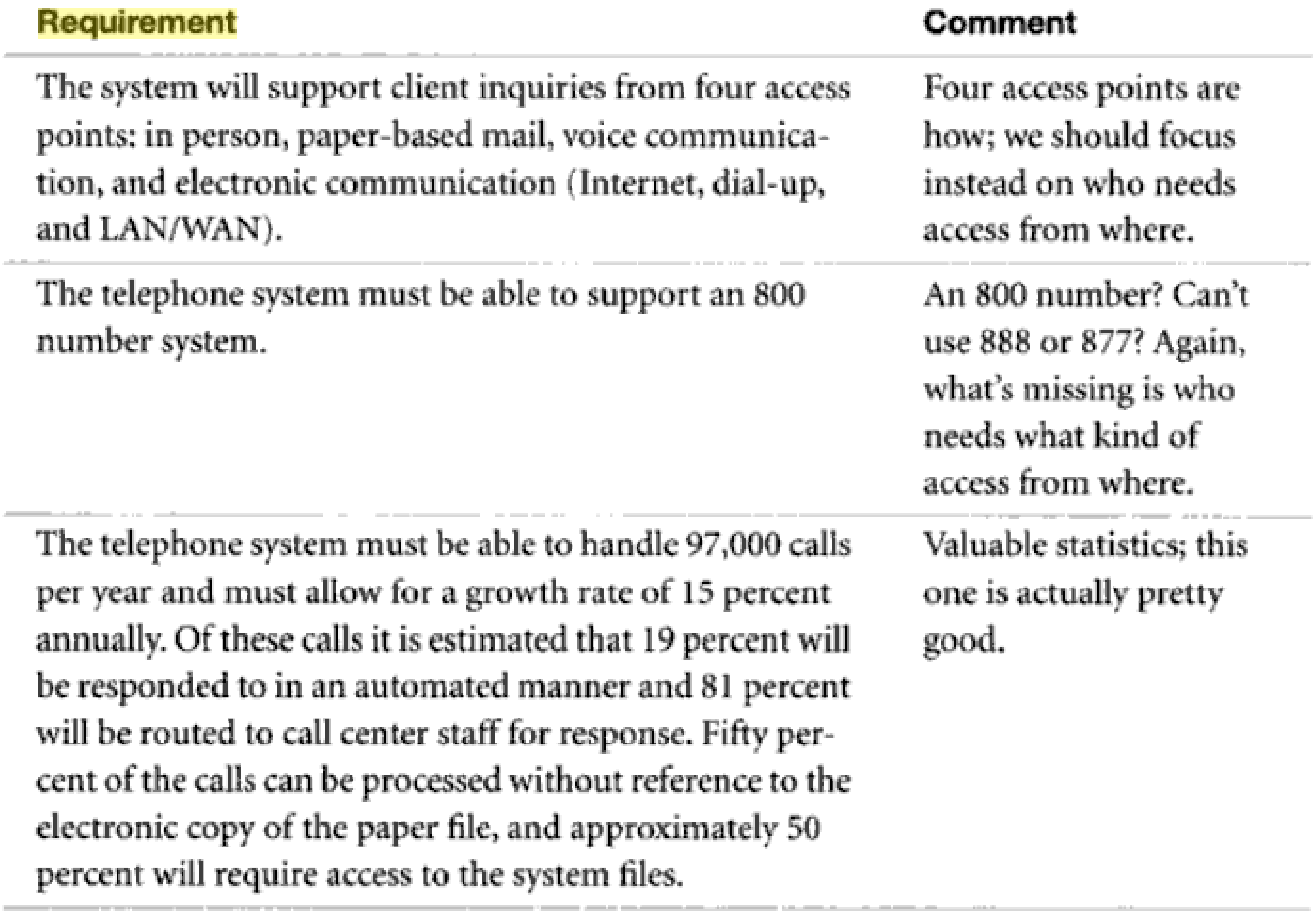
Requirements about performance, reliability, scaling, environmental, regulatory, safety, and security usually fall into this category.

CONSTRAINTS

Global requirements – on the project or the product

1. The product must be available by March 1.

CONTRACT STYLE



Requirement	Comment
The system will support client inquiries from four access points: in person, paper-based mail, voice communication, and electronic communication (Internet, dial-up, and LAN/WAN).	Four access points are how; we should focus instead on who needs access from where.
The telephone system must be able to support an 800 number system.	An 800 number? Can't use 888 or 877? Again, what's missing is who needs what kind of access from where.
The telephone system must be able to handle 97,000 calls per year and must allow for a growth rate of 15 percent annually. Of these calls it is estimated that 19 percent will be responded to in an automated manner and 81 percent will be routed to call center staff for response. Fifty percent of the calls can be processed without reference to the electronic copy of the paper file, and approximately 50 percent will require access to the system files.	Valuable statistics; this one is actually pretty good.

CONTRACT STYLE

Classify product features as

1. **Must-have** features
   “The product must conform to ADA accessibility guidelines”
2. **May-have** features
   “The product may be voice-controlled”
3. **Must-not-have** features
   “The product supports only one language”

Be explicit about must-not-have features!

WE INTERVIEWED HUNDREDS OF USERS AND TURNED ALL OF THEIR SUGGESTIONS INTO FEATURES.

USE CASE

- A set of actors and actions they take to achieve a goal (or fail in some way)
- Two elements

1. An **actor** is something that can act – a person, a system, or an organization
2. A **scenario** is a specific sequence of actions and interactions between actors (where at least one actor is a system)

Useful for clients as well as for developers

ACTORS AND GOALS

- What are the boundaries of the system? Is it the software, hardware and software, also the user, or a whole organization?
- Who are the primary actors – i.e., the stakeholders?
- What are the goals of these actors?
- Describe how the system fulfills these goals (including all exceptions)

EXAMPLE: SAFEHOME

CAM1
CCM2
CAM3
CAM4
The view from surveillance cameras in a control room at Evins
photo by Amber Novak

INITIAL SCENARIO

Use case: display camera views
Actor: homeowner

If I'm at a remote location, I can use any PC with appropriate browser software to log on to the SafeHome Web site. I enter my user ID and two levels of passwords and, once I'm validated, I have access to all the functionality. To access a specific camera view, I select "surveillance" and then "select a camera". Alternatively, I can look at thumbnail snapshots from all cameras by selecting "all cameras". Once I choose a camera, I select "view"…

REFINED SCENARIO

Use case: display camera views
Actor: homeowner

1. The homeowner logs on to the Web Site
2. The homeowner enters his/her user ID
3. The homeowner enters two passwords
4. The system displays all major function buttons
5. The homeowner selects “surveillance” button
6. The homeowner selects “Pick a camera”…

ALTERNATIVE INTERACTIONS

- Can the actor take some other action at this point?
- Is it possible that the actor encounters some error condition? If so, which one?
- Is it possible that some other behavior is encountered? If so, which one?

Exploring alternatives is key to successful requirements analysis!

SAFEHOME

Use-Case Template for Surveillance

Use-case: Access camera surveillance—display camera views (ACS-DCV).

Primary actor: Homeowner.

Goal in context: To view output of camera placed throughout the house from any remote location via the Internet.

Preconditions: System must be fully configured; appropriate user ID and passwords must be obtained.

Trigger: The homeowner decides to take a look inside the house while away.

Scenario:

1. The homeowner logs onto the *SafeHome Products* Web site.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Exceptions:

1. ID or passwords are incorrect or not recognized—see use-case: "validate ID and passwords."
2. Surveillance function not configured for this system—system displays appropriate error message; see use-case: "configure surveillance function."
3. Homeowner selects "view thumbnail snapshots for all cameras"—see use-case: "view thumbnail snapshots for all cameras."
4. A floor plan is not available or has not been configured—display appropriate error message and see use-case: "configure floor plan."
5. An alarm condition is encountered—see use-case: "alarm condition encountered."

Priority: Moderate priority, to be implemented after basic functions.

When available: Third increment.

Frequency of use: Infrequent.

SafeHome

Use-Case Template for Surveil

Use-case:	Access camera surveillance—display camera views (ACS-DCV).
Primary actor:	Homeowner.
Goal in context:	To view output of camera placed throughout the house from any remote location via the Internet.
Preconditions:	System must be fully configured; appropriate user ID and passwords must be obtained.
Trigger:	The homeowner decides to take a look inside the house while away.

Scenario:

1. The homeowner logs onto the *SafeHome Products* Web site.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Exceptions:

1. ID or passwords are incorrect or not recognized—see use-case: "validate ID and passwords."
2. Surveillance function not configured for this system—system displays appropriate error message; see use-case: "configure surveillance function."
3. Homeowner selects "view thumbnail snapshots for all cameras"—see use-case: "view thumbnail snapshots for all cameras."
4. A floor plan is not available or has not been configured—display appropriate error message and see use-case: "configure floor plan."
5. An alarm condition is encountered—see use-case: "alarm condition encountered."

SAFEHOME

Use-Case Template for Surveillance

Use-case: Access camera surveillance—display camera views (ACS-DCV).

Primary actor: Homeowner.

Goal in context: To view output of camera placed throughout the house from any remote location via the Internet.

Preconditions: System must be fully configured; appropriate user ID and passwords must be obtained.

Trigger: The homeowner decides to take a look inside the house while away.

Scenario:

1. The homeowner logs onto the *SafeHome Products* Web site.
2. The homeowner enters his or her user ID.
3. The homeowner enters two passwords (each at least eight characters in length).
4. The system displays all major function buttons.
5. The homeowner selects "surveillance" from the major function buttons.
6. The homeowner selects "pick a camera."
7. The system displays the floor plan of the house.
8. The homeowner selects a camera icon from the floor plan.
9. The homeowner selects the "view" button.
10. The system displays a viewing window that is identified by the camera ID.
11. The system displays video output within the viewing window at one frame per second.

Exceptions:

1. ID or passwords are incorrect or not recognized—see use-case: "validate ID and passwords."
2. Surveillance function not configured for this system—system displays appropriate error message; see use-case: "configure surveillance function."
3. Homeowner selects "view thumbnail snapshots for all cameras"—see use-case: "view thumbnail snapshots for all cameras."
4. A floor plan is not available or has not been configured—display appropriate error message and see use-case: "configure floor plan."
5. An alarm condition is encountered—see use-case: "alarm condition encountered."

Priority: Moderate priority, to be implemented after basic functions.

When available: Third increment.

Frequency of use: Infrequent.

BE SKEPTICAL

- Do the requirements meet the *real* needs?
- Do any requirements conflict?
- Is the requirements set complete?
- Are the requirements realistic / feasible?
  - Technically realistic
  - Budget realistic
- Are the requirements verifiable?
  - Can tests be defined so one can demonstrate the system satisfies the requirement?

FINAL NOTE ABOUT REQUIREMENTS

The job may never be done.

If the problem is complex enough, it will likely never be described completely.

Change happens.

- The environment changes.
- Once used, new priorities or requirements come to light.
- Compromises get exposed; priorities change.