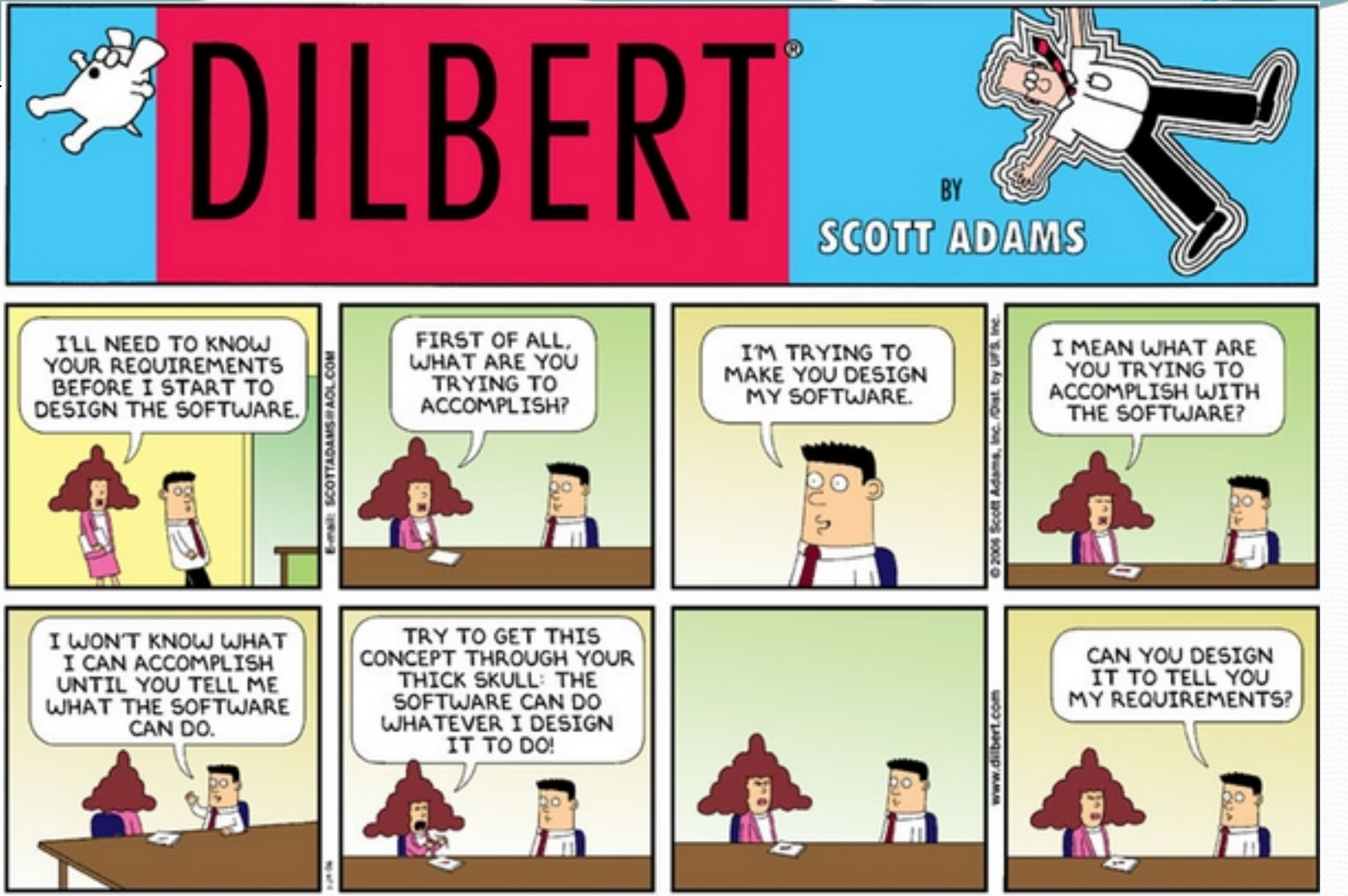

DILBERT®
BY SCOTT ADAMS
I'LL NEED TO KNOW YOUR REQUIREMENTS BEFORE I START TO DESIGN THE SOFTWARE.
E-mail: SCOTTADAMS@AOL.COM
FIRST OF ALL, WHAT ARE YOU TRYING TO ACCOMPLISH?
I'M TRYING TO MAKE YOU DESIGN MY SOFTWARE.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
I MEAN WHAT ARE YOU TRYING TO ACCOMPLISH 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?

Specifying Requirements

- Two types of Requirements
  - **Functional** -- What the customer needs the system to enable then to do.
  - **Non Functional** – What the customer needs the system to be. (non functional requirements can lead to functional system requirements.)

Non-functional Requirements:

- Can relate to all, some or just one specific function delivered by the system
- They may define **constraints** on the system or **characteristics**, **quality** of the system.
- They may relate to emergent system properties
- Can be subdivided into
  - product requirements (performance, reliability, usability),
  - organisational (standards, policies, hardware, other software)
  - external (legal, ethical, security, privacy & interoperability).

Property	Measure
Speed	Processed transactions/seconds User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Non-functional Requirements:

- Can be difficult to verify.
  - easy to use,
  - User-friendly
  - rapid user response,
  - should minimise user errors.
- To be testable they need to be quantifiable.
  - F ple

    An ex perator should be able t system func hours traini

    After training an ave an 2 errors should be made in a day.

Testable – pass / fail

easurable

Non-functional Requirements:

- *What are the non functional requirements for our Dog Door?*

Use a Scenario

Todd and Gina's Dog Door

What the Door does

1. Fido barks to be let out.
2. Todd or Gina hears Fido barking.
3. Todd or Gina presses the button on the remote control.
4. The dog door opens.
5. Fido goes outside.
6. Fido does his business.
   6.1 The door shuts automatically.
   6.2 Fido barks to be let back inside.
   6.3 Todd or Gina hears Fido barking (again)
   6.4 Todd or Gina presses the button on the remote control.
   6.5 The dog door opens (again).

7 Fido goes back inside.
8 The door shuts automatically.

Non-functional Requirements for our Dog Door

- The dog door must work reliable, when the remote is used.
- The door system must be usable by the Dog and Owner.
- The look of the dog door must match will the rest of the door.
- The door must be secure and safe to use.

All these are sensible non functional requirements BUT **vague**.
How would you **TEST** these, to meet the **Customers expectations**.

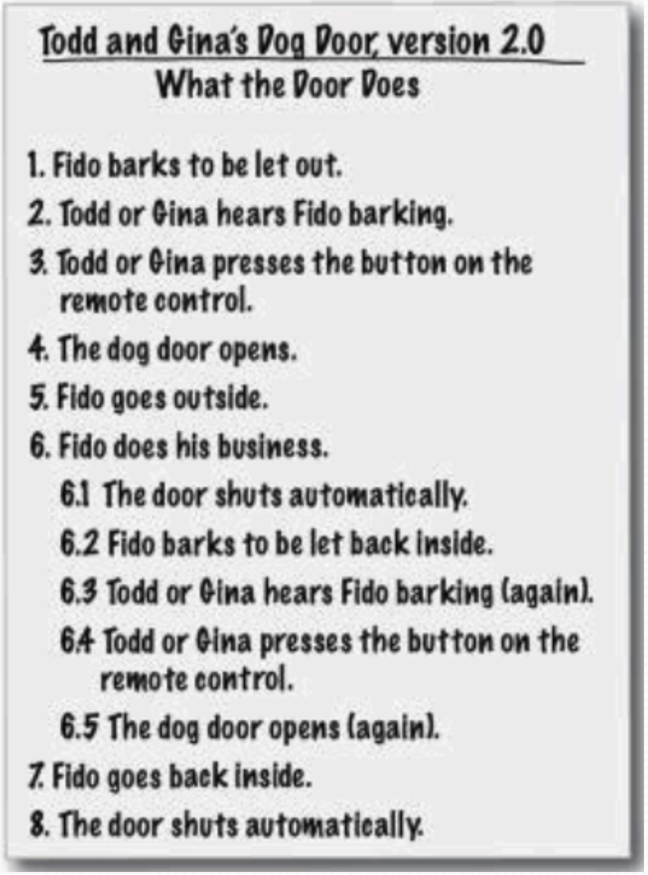
Better written Non-functional requirements

- The dog door must work reliable, when the remote is used
- The dog door must open 99% of the time on request from within the house at a range of 100 metres from the door .
- The dog door must close automatically 100% of the time.
- The remote must have a range of 100 metres from the door inside the house.
- The door much be usable by the Dog and Owner.
- Owner should be able to use the system straight away without help
- The door should be 90 cm in height, and 40cm in width
- The look of the dog door must match will the rest of the door
- The dog door should be available in the Dulux colours Juicy Red, Striking Cyan. Lemon Drizzle or Buckingham.
- The door must be safe and secure to use.
- The door should not close on the dog injuring it.
- When closed the dog door should be locked 100% of the time.

Better written Non-functional requirements

- Reliability
  - The dog door must open 99% of the time when the remote is pressed from within the house at a range of 100 metres from the door .
  - The dog door must close automatically 100% of the time.
  - The remote must have a range of 100 metres from the door inside the house.
- Size
  - The door should be 90 cm in height, and 40cm in width
- Portability
  - The dog door should be available in the Dulux colours Juicy Red, Striking Cyan. Lemon Drizzle or Buckingham.
- Safety and Security
  - The door should not close on the dog injuring it.
  - A person, should not be able to gain access to the house via dog door.

Non functional requirement are often written into the acceptance criteria of functional requirements

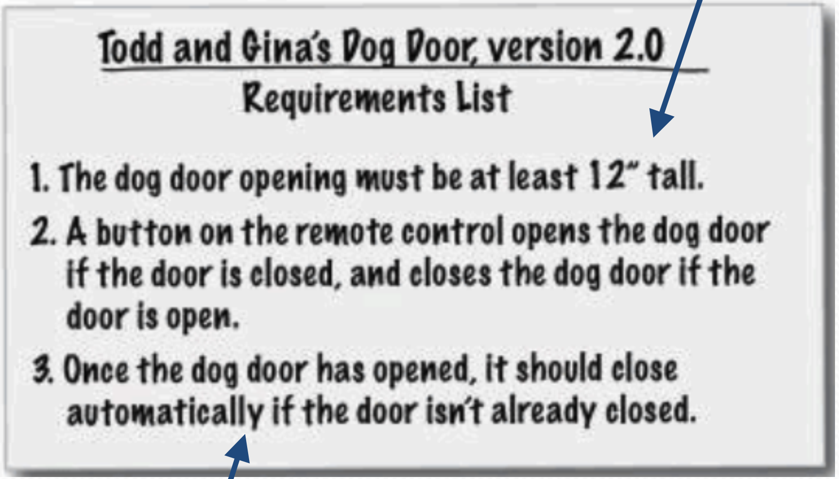


Todd and Gina's Dog Door, version 2.0

What the Door Does

1. Fido barks to be let out.
2. Todd or Gina hears Fido barking.
3. Todd or Gina presses the button on the remote control.
4. The dog door opens.
5. Fido goes outside.
6. Fido does his business.
   - 6.1 The door shuts automatically.
   - 6.2 Fido barks to be let back inside.
   - 6.3 Todd or Gina hears Fido barking (again).
   - 6.4 Todd or Gina presses the button on the remote control.
   - 6.5 The dog door opens (again).
7. Fido goes back inside.
8. The door shuts automatically.

Non functional. Not what your system does



Todd and Gina's Dog Door, version 2.0

Requirements List

1. The dog door opening must be at least 12" tall.
2. A button on the remote control opens the dog door if the door is closed, and closes the dog door if the door is open.
3. Once the dog door has opened, it should close automatically if the door isn't already closed.

Non functional. Security issue

Draft set of requirements

Non functional requirement are often written into the acceptance criteria of functional requirements

- **Dog Door system**
- Requirement 1. (Must)
  - **Requirement : Todd and Gina MUST be able to open the Dog Door flap to let the dog through.**
  - Acceptance Criteria.
    - Demonstration of
      - On using the remote device to select to open the dog door, the system should open the Dog Door flap
      - The Door should remain open for 5 Minutes
      - The Door should them close and lock.

Response Time

Security

Why do we develop a requirements document?

- Ensure the client understands what is going to be developed.
- To agree all of the functions, constraints, characteristics and quality
  - do you have enough information to design the software
- Check and agree any assumptions
- Not in the requirements then not in the code
- Best place to catch problems