Dilbert on Requirements Again

Why is your project four months behind?
I still don't have the user's requirements because she's a complete nut job.

It's your job to manage that process!
I complained to her boss, who promptly misinterpreted the problem and ordered her to work on the wrong stuff.

Then every member of her family got a serious illness. Then she got called to jury duty.
She promised to give me the requirements this afternoon.

It was too hard to come up with my own requirements, so I just copied the requirements from another product.

Is the other product similar to what you want?
Where are you going with this?
Requirements & Their Models

- What are the two domains? Why is it useful to distinguish between them? What is software engineering relative to these two domains?

- What are the main facets of this enterprise?
  - What are the critical elements in each of these facets?
  - What are their limitations?
  - How do they relate to each other?

- Requirements $\rightarrow$ specifications $\rightarrow$ programs
  - What differentiates each of these from the others
  - What is critical for each of these transformations?

- What is critical in the relationship between programs and the world?
Requirements & Their Models

- Why is the problem so critical in SE?
  - Is design of the problem a critical issue? Why? What essential problems do we find here? What techniques do we have at our disposal to manage these essential problems?
  - What is Jackson’s advice in the relationship between the problem and the solution?

- What kinds of justification do we use for focusing more on the machine than the problem? What are the advantages and problems of each?

- What are the four principles of descriptions?
  - How do they help us in understanding the problem and providing a solution?
  - What are their limitations, or their implications?
Evaluating Requirements

- What is the critical question we need to answer about requirements? What are the problems in addressing this question with documents?
- What are the benefits of prototyping?
- What characteristics to prototypes need to have to be effective?
- What are the two major approaches? Which does Brooks support? What do we gain from each?
- What are different kinds of techniques used?
  - What are their individual advantages and disadvantages?
  - Under what conditions do they best apply?
- Can we use prototyping beyond evaluations? If so how and what benefits do we gain thereby?