Reading Feedback

Name: Dazhi Li

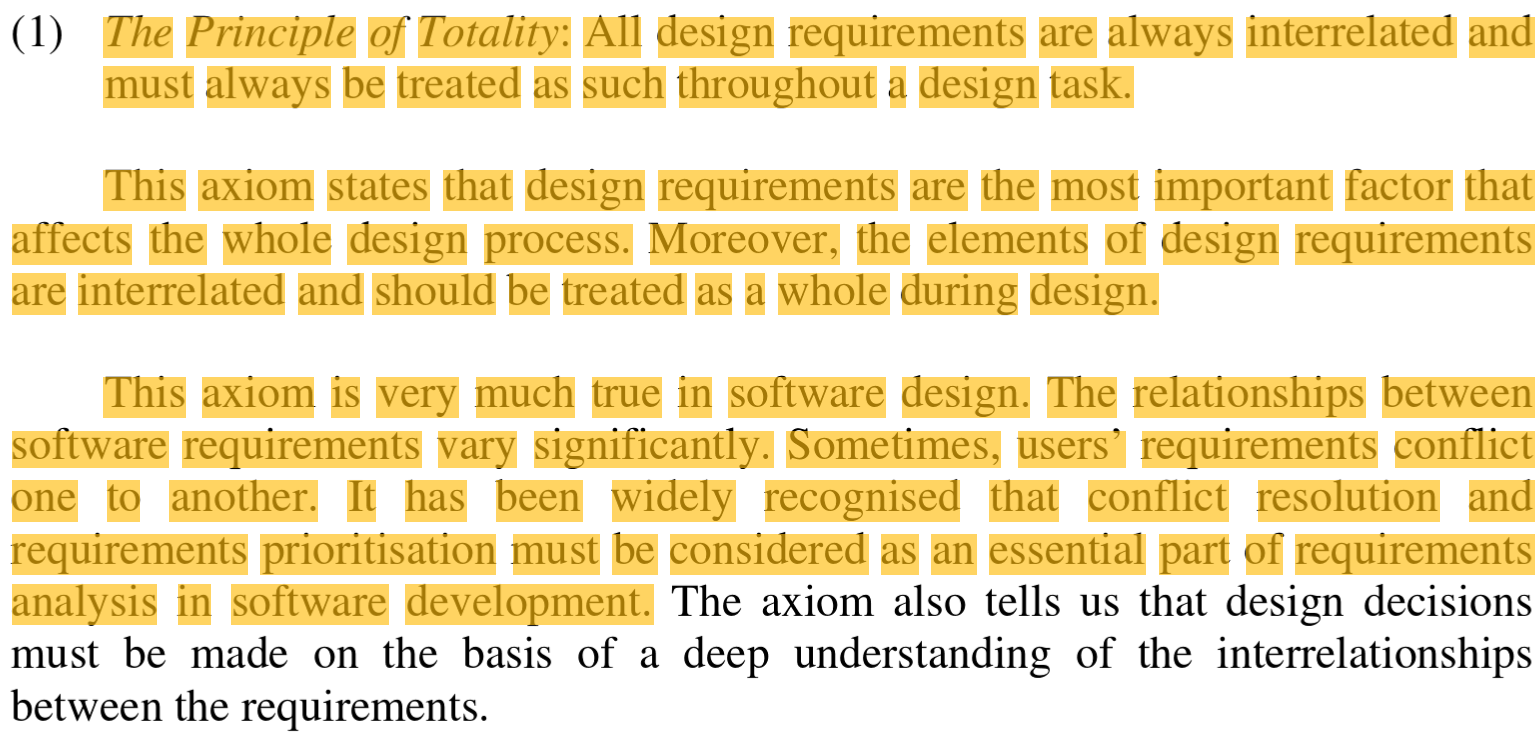
**One sentence conclusion:**

These readings make me clear about what is a design, what qualities does a good design obtains and how architect made decisions in designs. These qualities are classified by different people on different perspectives. What I found insightful is both books mentioned that a good design consider not only the architecture design but also the process of a plan. The process of a plan includes feasibility which is not an attribute of a design. What makes me muddy is the architectural patterns and tactics, which are utilized by architect to make decisions to achieve quality attributes. I think any example of architectural pattern and tactic will make me understand this part better.

# Software Design Methodology

## Chapter 1 page 16

Muddy



My question:

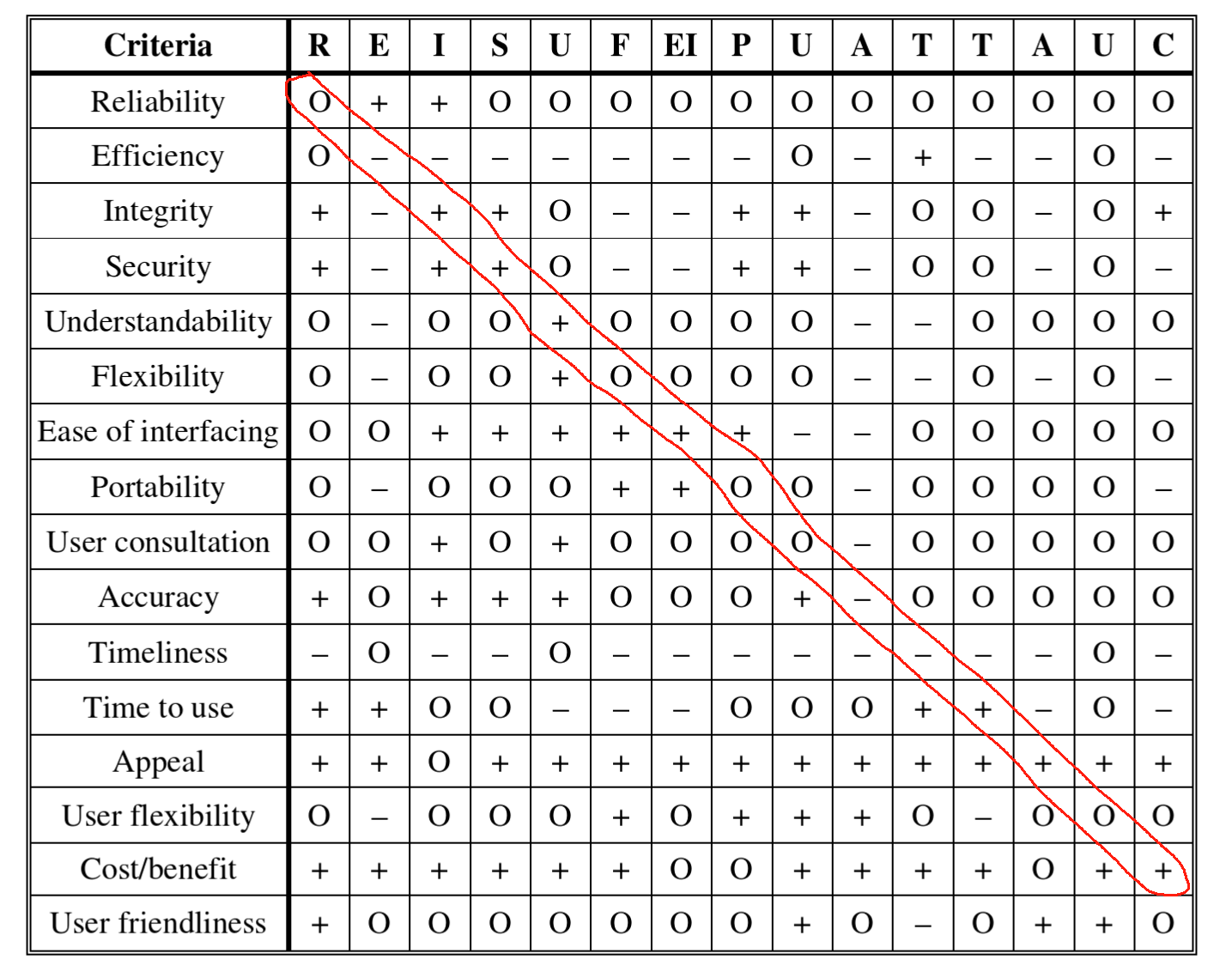
Does all the designs should be scrapped and started all over again when new requirements conflicts with the previous one?

Especially, is there any unresolvable conflict between two requirements?

------------------------------------------------------------------------------------------------------------------

## Chapter 2 page 31

Muddy

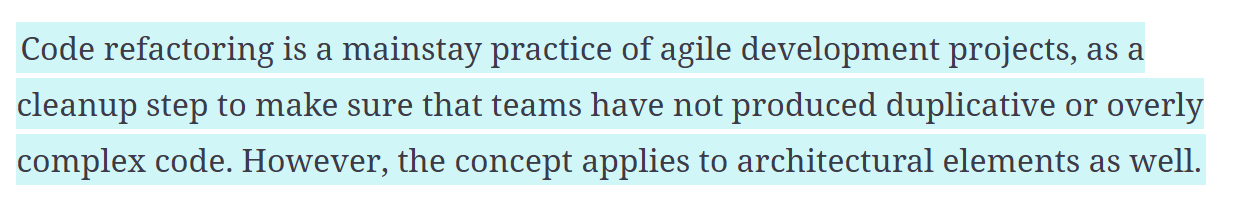


My question: Take the row 3 efficiency as an example. Why increasing efficiency is bad for efficiency? How could we explain the diagonal relationships between quality attributes?

# Software Architecture in Practice

### Chapter3

Insightful



Usually, developers consider refactoring code is a waste of time because they are writing the same code twice to achieve the same feature. However, refactoring reduces code complexity and increases an important attribute, the flexibility. In a agile development, flexibility is the essence for faster design and implementation.