

# **Software Requirements:** **What should the software do?**

Prof. Jo Atlee

Director of Software Engineering

# Goals for this Lecture

Figuring out what a new software system should do

- Identifying the problem to solve
- Resolving conflicts in requirements
- Effectively communicating requirements

These are not just software engineering skills

These are handy life skills, as well

# What is Software Engineering?

An attempt to address the “Software Crisis”

- Poor software quality
- Cost and deadline overruns
- Systems built are wrong or useless

By applying engineering principles to software products

- Disciplined development process
- Project planning and management
- Analysis of problem to be solved

# Software Requirements

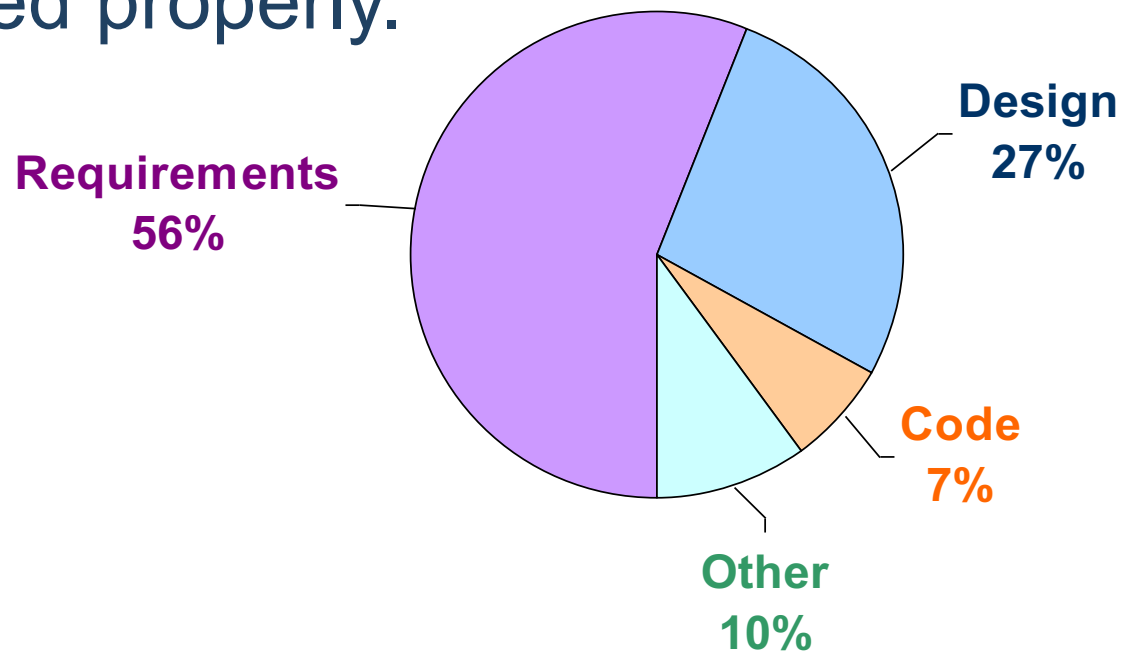
A condition or capability that must be possessed or met by the software for the user or customer to consider the software an acceptable solution to their problem.

- Functional requirements  
*E.g., calculate sums*
- “Quality” requirements  
*E.g., be fast, reliable, easy to learn, ready to use quickly*

Example: On an exam, you have 20 minutes to write a 1-page essay on the history of Canadian citizenship.

# Building the Right System

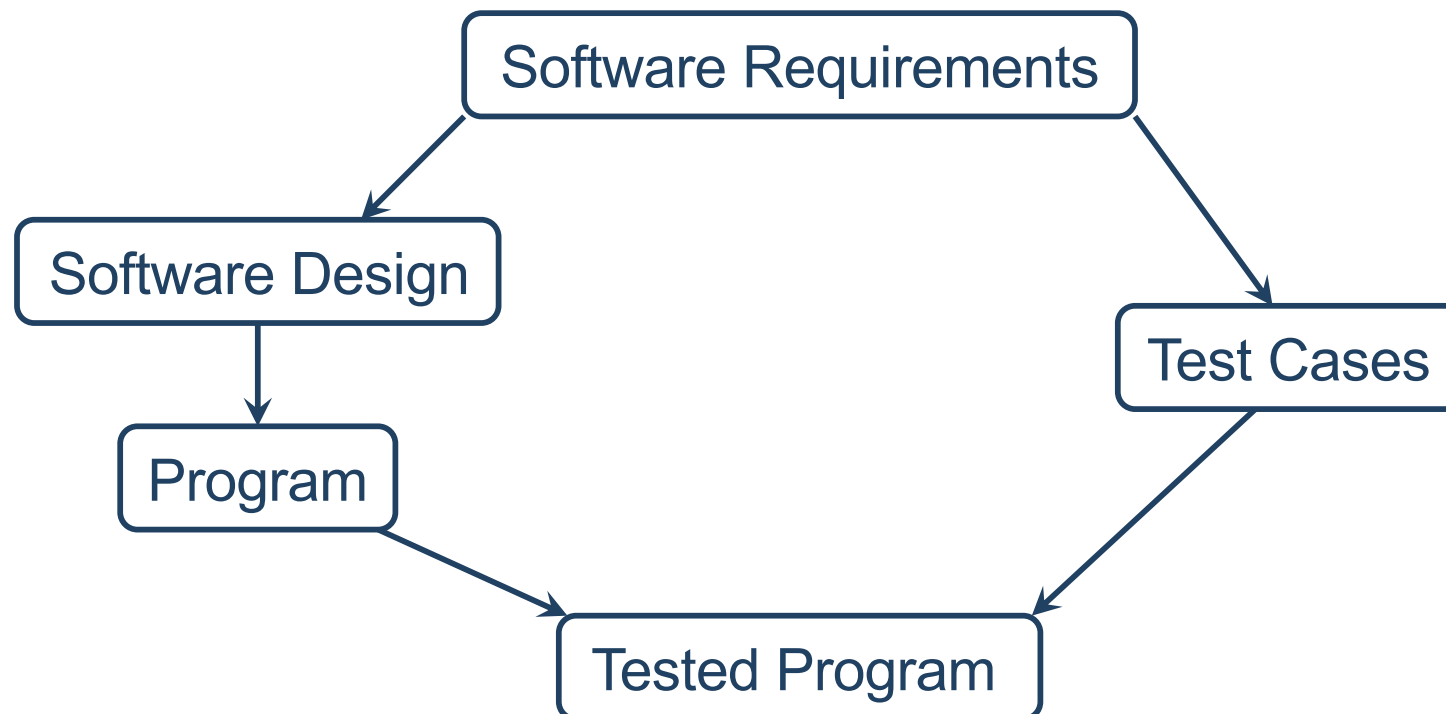
More software systems fail because they don't meet the needs of their users than because they aren't implemented properly.



This means that software developers often don't know *what the software is supposed to do*.

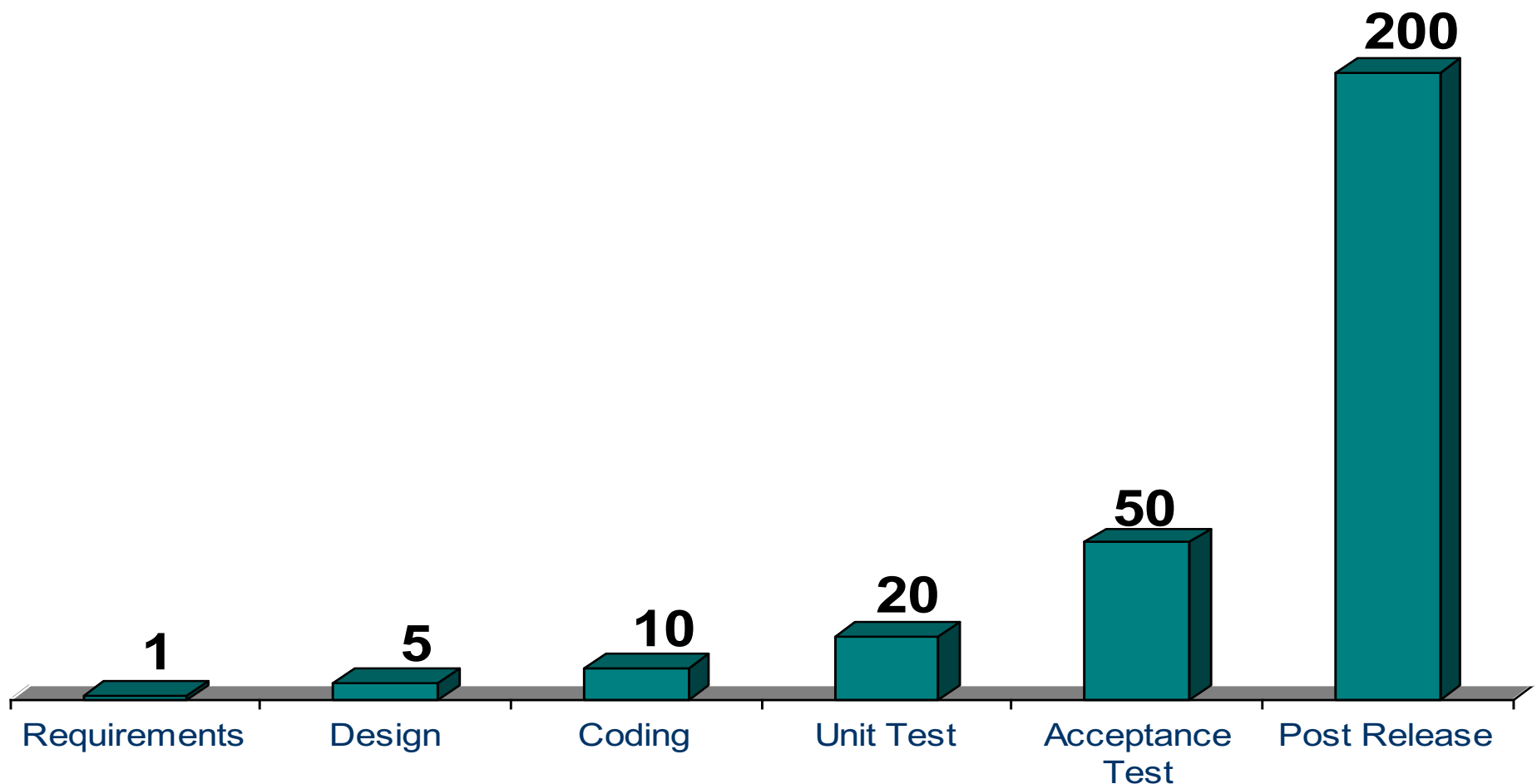
# Requirements Errors are Elusive

Test cases that are derived from the same faulty requirements as the software cannot reveal requirements errors.



# Requirements Errors can be Expensive

Relative cost of requirements errors, depending on when they are detected and fixed.



# Problem for Audience to Solve

Write down the next number in the sequence

14

23

28

33

42

51

59

68

77

86



# Recognizing Number Sequences

In the early 1980's, an IBM researcher was developing a software system that would recognize patterns in a given number sequence and would produce the next number in the sequence.

He (and his program) were stumped by this number sequence:

14, 23, 28, 33, 42, 51, 59, 68, 77, ...

# Nature of Requirements Errors

User requirements are often

- Vague - customer doesn't know what (s)he wants
- Ambiguous - open to multiple interpretations
- Contradictory
- Incomplete
- Impractical or impossible to implement
- Just plain wrong

**Exercise:** Typical requirements errors that a clothes shopper would make when talking to a sales person.

# Points to Remember

- It's a lot easier to solve the problem than it is to identify the problem.

# Requirements Exercise

I am the mayor of the Swiss village of Wengen, and I would like a transportation device for tourists.

You are the designer I have hired.

What are your initial thoughts on possible solutions to my problem?

# Teasing out the requirements

If we know that we don't know – we **ask**

- Who are the stakeholders?
- What is (are) their problem(s)?
- When do they need a solution/product?
- Where will the product be deployed?
- Are there any constraints?
- How will the product be used?
- Why is each requirement necessary?

# Points to Remember

- It's a lot easier to solve the problem than it is to identify the problem.
- The stated problem is often not the real problem - ask lots of questions and confirm assumptions.

# Conflicting Requirements

Sometimes, users (think they) know exactly what features or properties they want in a new software system.

But different users may have conflicting ideas about what the software should do.

**Exercise:** Students Records Information System

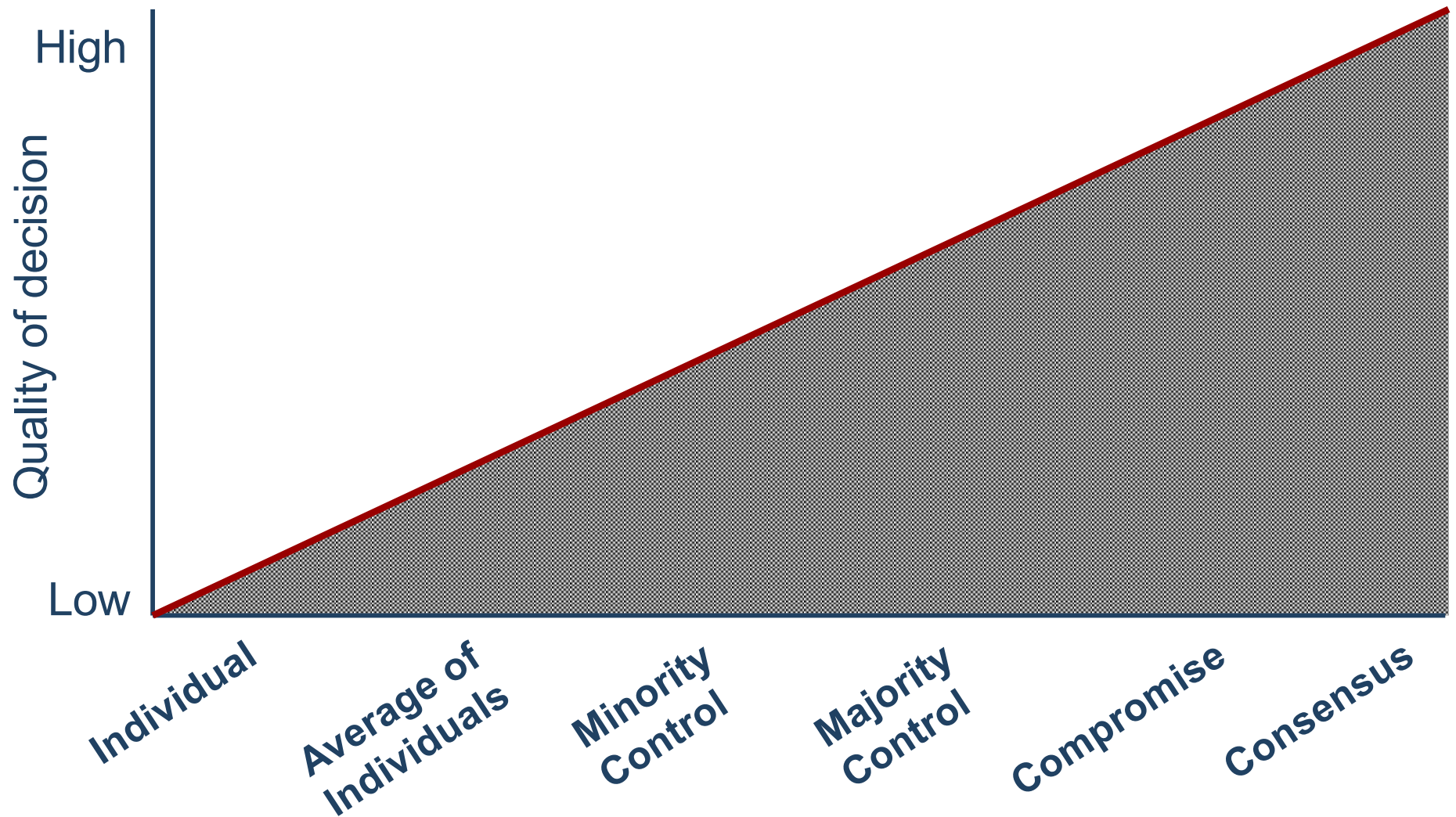
- Who are the stakeholders?
- What information might each stakeholder want?

# Decision Making

In your experience as a member of a group, how did the group make a group decision?

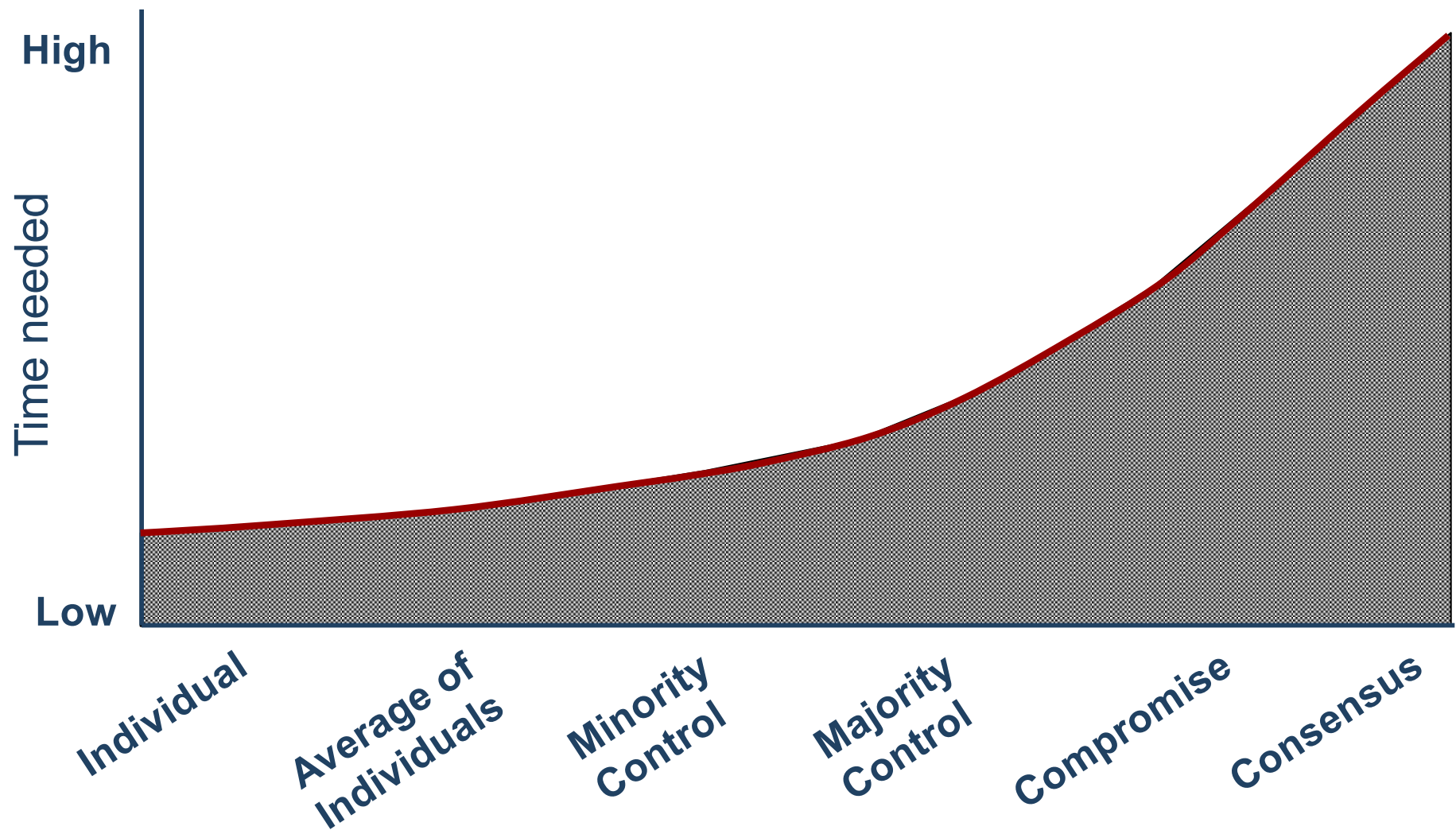


# Decision type vs. Quality of decision



Smith, Karl, *Project Management and Teamwork*, McGraw Hill, 2000.

# Decision type vs. Time needed



Smith, Karl, *Project Management and Teamwork*, McGraw Hill, 2000.

# Family Vacation Exercise

- The husband wants to take his family out to lake country to go camping and fishing. He has been planning the trip all year. He's rented a cottage, arranged to rent a boat.
- The wife wants to use the vacation time to visit her ailing mother (some 250 miles away from lake country).
- The two sons are really excited about going fishing.

# Family Vacation Exercise

**Husband:** “The plans are set. The boys are excited. We should go on the fishing trip.”

**Wife:** “But we don’t know how much longer my mother will be around, and I want to be by her. This is our only opportunity to have enough time to do that.”

**Husband:** “The boys will be miserable sitting around grandmother’s house for a week. They’d drive everybody crazy. Besides, your mother is not that sick. And she has your sister nearby to take care of her.”

**Wife:** “She’s my mother too. I want to be with her.”

**Husband:** “You could phone her every night. And we’re planning to spend time with her at Christmas.”

**Wife:** “That’s not for five more months. We don’t even know if she’ll still be here by then. Besides, she wants me and needs me.”

**Husband:** “The boys and I need you, too.”

**Wife:** “My mother is more important than fishing.”

**Husband:** “Your husband and sons are more important than your mother.”

# Win/win solutions

**Idea:** Find solution that mutually benefits all parties.

- Often not anyone's original proposal
- Asks all parties to think about *why* their proposal is important to them
  - E.g., what are their underlying requirements
- Looks for solution that satisfies the parties' fundamental requirements

# Student Records System Exercise

- Universities and employers want each student's transcript to report his or her ranking in each class (e.g., 1st, 2nd) and the number of students in each class — to help them compare students' performances.
- Students don't want such detailed performance indicators on the transcript — they wouldn't mind knowing how their marks to compare to those of other students, but they feel that making such information public would breed over-zealous competition among students over marks.

# Points to Remember

- It's a lot easier to solve the problem than it is to identify the problem.
- The stated problem is often not the real problem - ask lots of questions and confirm assumptions.
- Win/win solutions meet essential requirements - not necessarily the stated needs

# Assuming what the software should do

If we don't know that we don't know – we **assume**

*“It’s surprisingly easy to think that you understand a description that you’re reading or have just written when in fact you don’t really understand it at all.... Software practitioners get a lot of practice guessing what descriptions mean.”*

*Michael Jackson*



# Obfuscating Descriptions

Requirements writers can artificially complicated descriptions.

**Black** **Yellow** **Green** **Blue** **Red** **Black**

**Red** **Blue** **Yellow** **Green** **Black** **Yellow**

**Green** **Blue** **Red** **Black** **Blue** **Green**

**Yellow** **Red** **Black** **Green** **Red** **Blue**

# Objectives vs. Constraints

Software designers often fail to distinguish between requirements objectives and constraints.

- Objectives are to be optimized

*E.g., vacation should be fun*

*E.g., exam essay on the history Canadian citizenship*

- Constraints are either satisfied or not

*E.g., vacation dates*

*E.g., answer exam question in 1 page, in 20 minutes*

# Furniture Design Exercise

Working in groups of three and four, design an innovative piece of furniture that can be manufactured by individuals who have physical or mental impairments, disabilities, or conditions.

## Constraints:

- cost (must sell for under \$30)

- shipping (ship unpackaged without fear of damage)

## Objectives:

- innovative

# Furniture Design Exercise

In previous versions of this exercise, how the functional requirements were named affected both the quality of the solutions and the behaviours of the design groups.

- Social behaviour among the Name Groups is contentious, compared to that of the Function Groups who look like they have more fun
- Tend to have more Function Groups volunteer to present their solution for judgement
- The Function Group solutions tend to be evaluated as being more innovative.

# Points to Remember

- It's a lot easier to solve the problem than it is to identify the problem.
- The stated problem is often not the real problem - ask lots of questions and confirm assumptions.
- Win/win solutions meet essential requirements - not necessarily the stated needs.
- Descriptions of the problem that aren't in terms of essential requirements artificially constrain the solution space.

# Useful Life Skills

- It's a lot easier to solve the problem than it is to identify the problem.
- The stated problem is often not the real problem - ask lots of questions and confirm assumptions.
- Win/win solutions meet essential requirements - not necessarily the stated needs.
- Descriptions of the problem that aren't in terms of essential requirements artificially constrain the solution space.