What requirements do you have when buying a new car?

- Gets good gas mileage
- Has a good safety record
- Drives well in the snow
- Can carry a lot of cargo
- Looks good
- Is red

CS445 / ECE451 / CS645 / SE463

Software Requirements Specification & Analysis

Quality Requirements

Quality Requirements

- Functional requirements describe what the software is supposed to do
  - What services or tasks the software should provide
  - Black box input/output behaviour
- Quality requirements describe (extra) constraints on what constitutes an acceptable software solution

  e.g., How fast system should respond

  Which deployment platforms should be supported

  How strong should security be

Example: On an exam, you have 20 minutes to write a 300-word essay on the history of Canadian citizenship.

Quality Requirements

- Performance
  - execution speed
  - response time
  - throughput
    - e.g., “up to 30 simultaneous calls”
- Reliability
  - fault-tolerant
  - mean-time to failure
  - data backups
- Robustness
  - tolerates invalid input
  - fault-tolerant
  - fail-safe / -secure
  - degrades gracefully under stress
- Adaptability
  - ease of adding new functionality
  - reusable in other environments
  - self-optimizing, self-healing
- Security
  - controlled access to system, data
  - isolation of data, programs
  - protect against theft, vandalism
- Usability
  - how easy to learn / use
  - user productivity
- Scalability
  - workload
  - number of users
  - size of data sets
  - peak use
- Efficiency (capacity)
  - utilization of resources
- Accuracy / precision
  - tolerance of computation errors
  - precision of computation results

Other Nonfunctional Requirements

Design constraints

- interfaces to other systems
- required components, libraries
- programming language

Operating Constraints

- location
- size, power consumption
- temperature, humidity
- operating costs
- accessibility (for maintenance)

Product-family requirements

- modifiability
- portability
- reusability
- configurable UI

Process Requirements

- Resources
  - personnel development
  - costs
  - development schedule
- Documentation
  - audience
  - conventions
  - readability
- Complexity (of code)
  - comments / KLOC
  - coupling / cohesion
  - cyclomatic complexity
  - use of multiple inherit. overloading, templates
- Standards compliance

“Motherhood” requirements

Expressions such as “reliable”, “user-friendly”, and “maintainable” are motherhood requirements.

- No one would explicitly ask their opposite

e.g., *slow, unreliable, user-hostile, unmaintainable*, …

(Virtually) every software system must have attributes such as “reliable”, “user-friendly”, and “maintainable”; what differs from product to product is:

- the degree to which each attribute is required, and
- the relative importance of one attribute over another.

Good Enough

"Anyone can build a bridge. It takes an engineer to build a bridge that barely stands."

[unknown source]

Fit criteria

A fit criterion quantifies the extent to which a quality requirement must be met.

Example:

- *System shall be down, on average, no more than 5 minutes a year*
- *75% of users shall judge the system to be as usable as the existing system*
- *After training, 90% of users shall be able to process a new account within 4 minutes*
- *Computation errors shall be fixed within 3 weeks of being reported*

Measurable Criteria

Quality Attribute	Metric
Performance	response time throughput capacity
Efficiency	maximum allowed load on resouces
Reliability	mean-time to failure (MTTF)
Security	categories of users percentage of attacks that are successful
Robustness	percentage of failures on invalid input minimum performance under heavy loads
Scalability	size of input data sets number of users
Cost	maximum costs to buy, install, or operate
Portability	collection of target platforms
Readability	Flesh Reading Ease Score
Maintainability	mean-time to fix bugs, add features
Usability	amount of training needed to perform tasks on own time to perform tasks at expected speed number of calls to help desk rate at which users adopt software approval rating user error rates

Rich Fit Criteria

Requirement	Outstanding	Target	Minimum
Response time	0.1s	0.5s	1s
CPU utilization	20%	25%	30%
Usability	40 tasks/hr	30 tasks/hr	20 tasks/hr

When you can't test before delivery

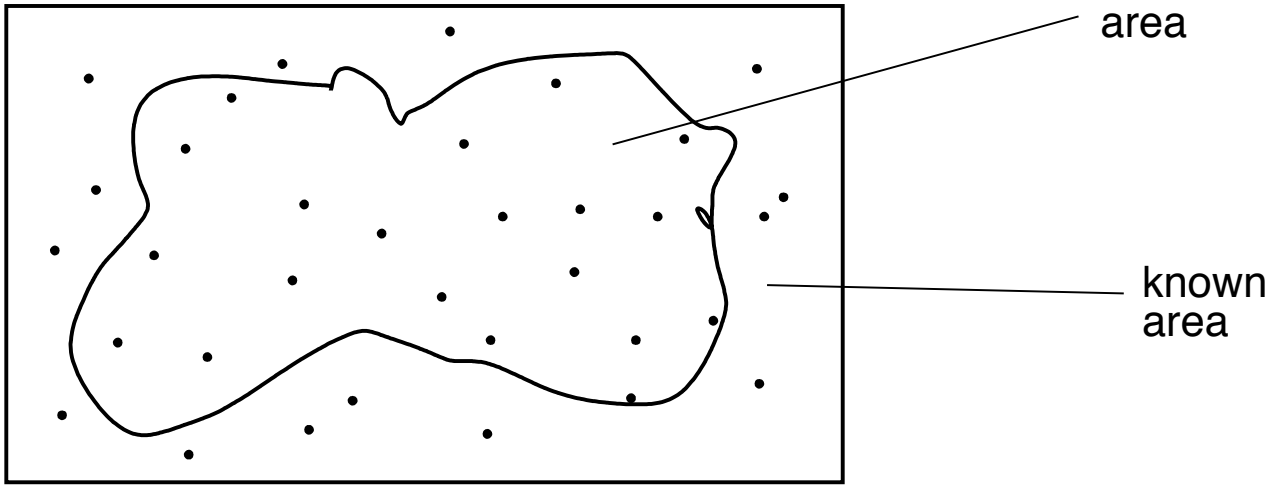
- Fit criteria that cannot be evaluated before the final product is delivered are harder to assess. For example:
  - *The system shall not be unavailable for more than a total of 3 minutes each year*
  - *The mean-time-to-failure shall be no less than 1 year*

- Possible approaches:
  - Measure the attributes of a prototype.
  - Measure secondary indicators

    e.g., number of user errors to assess usability
  - Estimate a system's quality attributes
  - Deliver system and pay penalty if requirements are not met

Monte Carlo techniques

Monte Carlo techniques: estimate an unknown quantity using a known quantity.

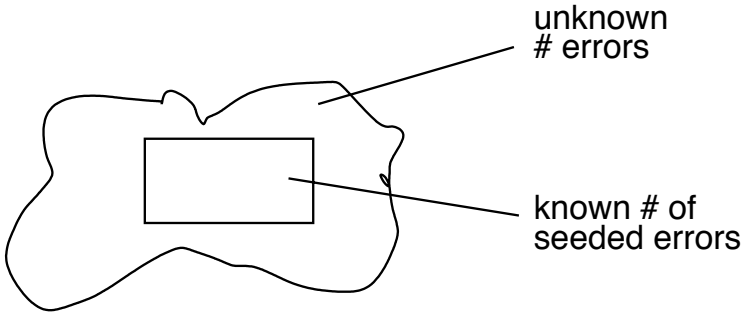


$$\frac{\text{Number of points in blob}}{\text{Total number of points}} \approx \frac{\text{Area of Shape}}{\text{Known Area of Rectangle}}$$

Monte Carlo techniques

We can use Monte Carlo techniques to estimate number of bugs remaining in a program (reliability).

- Plant a known number of errors into the program, which the testing team does not know about.
- Compare the number of seeded errors the team detects with the total number of errors it detects, to arrive at an estimate of the total number of bugs in the program.



$$\frac{\text{\# detected seeded errors}}{\text{\# seeded errors}} \approx \frac{\text{\# detected errors}}{\text{\# errors in the program}}$$

What quality requirements do you strive for when building a paper airplane?

Paper Airplane Quality Requirements

- Flies a long way (3 metres or more)
- Stays aloft a long time (3 sec or more)
- Flies straight (deviates no more than 10^{o})
- Looks sleek (75% onlookers say so)
- Is durable (still flies after 5 flights)

Paper Airplane Quality Requirements

✓ Flies a long way	(3 metres or more)
✓ Stays aloft a long time	(3 sec or more)
• Flies straight	(deviates no more than 10^{o})
✓ Looks sleek	(75% onlookers say so)
✓ Is durable	(still flies after 5 flights)

Quality Attributes Conflict

	Availability	Efficiency	Installability	Integrity	Interoperability	Modifiability	Performance	Portability	Reliability	Reusability	Robustness	Safety	Scalability	Security	Usability	Verifiability
Availability									+		+					
Efficiency	+				−	−	+	−			−		+		−	
Installability	+								+					+		
Integrity			−		−		−			−		+		+	−	−
Interoperability	+		−	−			−	+	+		+	−		−		
Modifiability	+		−				−		+	+			+			+
Performance		+			−	−		−			−		−			−
Portability		−			+	−	−			+				−	−	+
Reliability	+	−		+		+	−				+	+		+	+	+
Reusability		−		−	+	+	−	+						−		+
Robustness	+	−	+	+	+		−		+			+	+	+	+	
Safety		−		+	+		−				+			+	−	−
Scalability	+	+		+			+	+	+		+					
Security	+			+	+		−	−	+		+	+			−	−
Usability		−	+				−	−	+		+	+				−
Verifiability	+		+	+		+			+	+	+	+		+	+	

FIGURE 14-2 Positive and negative relationships among selected quality attributes.

Karl Wiegers, *Software Requirements*, (MS Press)

Summary

Requirement	Outstanding	Target	Minimum
Response time	0.1s	0.5s	1s
CPU utilization	20%	25%	30%
Usability	40 tasks/hr	30 tasks/hr	20 tasks/hr

References

Karl E Wiegers and Joy Beatty. *Software Requirements, 3ed*. Microsoft Press, 2013.

Chapter 14: "Beyond Functionality”

Soren Lauesen. 2001. *Software Requirements: Styles and Techniques, 1ed.* Pearson Education, 2001.

Chapter 6 “Quality Requirements”