1. Are the problems of Denver Airport common?

Yes. The paper says "for every 6 new large software systems, 2 get canceled" and "3/4 of large systems are operating failures". Denver's problems were just more visible.

2. What are the percentages of 'operating failures' Software as observed in the mid-1990s? What are the average increase in scheduling time?

75% operating failures

Average schedule overrun: 50% longer than planned

3. What are the Software Crises?

The paper says software industry lacks proper engineering methods, leading to frequent project failures, delays, and systems that don't work as intended.

4. What is 'Software Engineering'? (answer from the paper)

Defined as "systematic, disciplined, quantifiable approach to software development, operation and maintenance".

5. Do you feel that outsourcing affects software standards and interoperability? Why or why not?

Yes. The paper says countries like India have cheaper labor but weaker design skills ("Most Eastern nations weak in design", last section). Different standards make components hard to connect.

6. Is it easy to get the software right the first time? Maybe if we try harder and with military discipline?

No. Even military projects like space shuttle software had bugs after strict testing ("1981 launch aborted due to glitch"). Real-world conditions cause surprises.

7. What is unique about 'Realtime Systems'? And what is different in 'Distributed Systems'?

Realtime: Errors happen only in specific conditions (like Clementine satellite failure)

Distributed: Network complexity creates unpredictable failures

8. Did the Loral team experience go completely successful?

Mostly successful (CMM level 5 rating) but still had errors: 1981 shuttle launch abort and 1992 satellite rescue problem.

9. Would formal methods or mathematics solve the problem?

No. Formal methods help find design errors early (British air traffic example), but testing still needed for real-world issues.

10. Why are we interested in component software? (from the paper)

Reusable parts could reduce costs and errors (like hardware). But hard to make compatible between systems. All answers

come directly from the text's paragraphs as numbered in the original document. Simple vocabulary and short answers used as requested.