Barry Boehm:
A Prioritized Top-Ten List of Software Risk Items

- source: I believe May 1988 *IEEE Computer*, Table 4, p. 70, available 2-23-12 at:
  weblog.erenkrantz.com/~jerenk/phase-ii/Boe88.pdf

Top-Ten List of Software Risk Items

1. Personnel shortfalls
2. Unrealistics schedules and budgets
3. Developing the wrong software functions
4. Developing the wrong user interface
5. Gold plating
6. Continuing stream of requirement changes
7. Shortfalls in externally furnished components
8. Shortfalls in externally performed tasks
9. Real-time performance shortfalls
10. Straining computer-science capabilities

Risk-management techniques for each:

1. Personnel shortfalls
   - Staffing with top talent
   - Job matching
   - Teambuilding
   - Morale building
   - Cross-training
   - Pre-scheduling key people

2. Unrealistics schedules and budgets
   - Detailed, multisource cost and schedule estimation
   - Design to cost
   - Incremental development
   - Software reuse
   - Requirements scrubbing

3. Developing the wrong software functions
   - Organizational analysis
4. Developing the wrong user interface
   – Task analysis
   – Prototyping
   – Scenarios
   – User characterization (functionality, style, workload)

5. Gold plating
   – [note: Jalote course text, p. 83: "Gold plating refers to adding features to the software that are only marginally useful"]
   – Requirements scrubbing
   – Prototyping
   – Cost-benefit analysis
   – Design to cost

6. Continuing stream of requirement changes
   – High change threshold
   – Information hiding
   – Incremental development (defer changes to later increment)

7. Shortfalls in externally furnished components
   – Benchmarking
   – Inspections
   – Reference checking
   – Compatibility analysis

8. Shortfalls in externally performed tasks
   – Reference checking
   – Pre-award audits
   – Award-fee contracts
   – Competitive design or prototyping
   – Teambuilding

9. Real-time performance shortfalls
– Simulation
– Benchmarking
– Modeling
– Prototyping
– Instrumentation
– Tuning

10. **Straining computer-science capabilities**
    – Technical analysis
    – Cost-benefit analysis
    – Prototyping
    – Reference checking