Best practices

• DOD
• Airlie software Council (Virginia)
• 1994 initiative
Nine Best practices

- Formal risk management
- User manual as specification
- Inspections and peer reviews
- Metric-based scheduling and tracking
- Binary gates at the inch-pebble level
- Program-wide visibility of project plan and progress versus plan
Nine best practices

• Defect tracking against quality targets
• Separate specification of hardware and software functionality
• People-aware management accountability
Worst practices

• Don’t use schedule compression to justify usage of new technology on any time critical project
• Don’t specify implementation technology in the RFP
• Don’t advocate use of unproven silver bullet approaches
Worst Practices

• Don’t expect to recover from any substantial schedule slip (10% or more) without making more than corresponding reductions in functionality to be delivered

• Don’t put items out of project control on the critical path

• Don’t expect to achieve large, positive improvements (10% or more over past observed performance
Worst practices

• Don’t bury all project complexity in the software (as opposed to the hardware)

• Don’t conduct the critical system engineering tasks without software expertise

• Don’t believe that formal reviews provide an accurate picture of the project. Usefulness inversely proportional to number beyond five
Microsoft

- Level 5 can’t compete against Microsoft?
- 17 million copies of Word?
- Legal problems
- Bozo explosion?
- 2000 unsolicited resumes/week
MS peopleware policies

• Hire smart people
• On project team right away (IBM - 6 months training)
• Weekly education sessions
• Mentor
• Kick back
MS managers

• Induce uncertainty, don’t swallow it
• The manager is the greatest expert on when you will finish - rely on QA for opinion
• Prevent someone from going dark
• Don’t micromanage
  – rely on interdependence among team members
• Small milestones
• Don’t trade one bad date for another
MS development practices

- Case tools, formal analysis and design - not
- Formal specification - not
- Daily build
- Testing
  - Development not allowed to being until testing signs off on specifications
  - 1-1 ratio of testers to developers
  - Quick and dirty tests before build