Maintenance & Evolution

• Origin: Real-world user needs
  – Late ‘40s and early ‘50s
  – Example

Maintenance & Evolution: Where we are now

• Several good models of software lifecycle
  – Staged, Versioned Staged, (others?)
• Tons of substantial research challenges
  – Improved Architecture for unanticipated change
  – Raising abstraction levels
    • Of code
    • By component reuse or assembly
  – Program Comprehension
Software Analysis

• Origins:
  – Formal Methods
    • Modeling
    • Specification
    • “program proving”
  – Pragmatic Engineering
    • Tools to support testing, maintenance, debugging, etc
  – Compiler Research
    • Optimizing compiler alums repurpose their toolkits

Software Analysis:
Where we are now

• Good work on
  – models
  – model checking
  – Code Slicing/Dicing

• Challenges
  – Linking models with code
  – Composable analyses
  – Support for discovering/maintaining/making evident the “truth” that is (or should be) in the code