

# REQUIREMENTS

---

for fun and profit

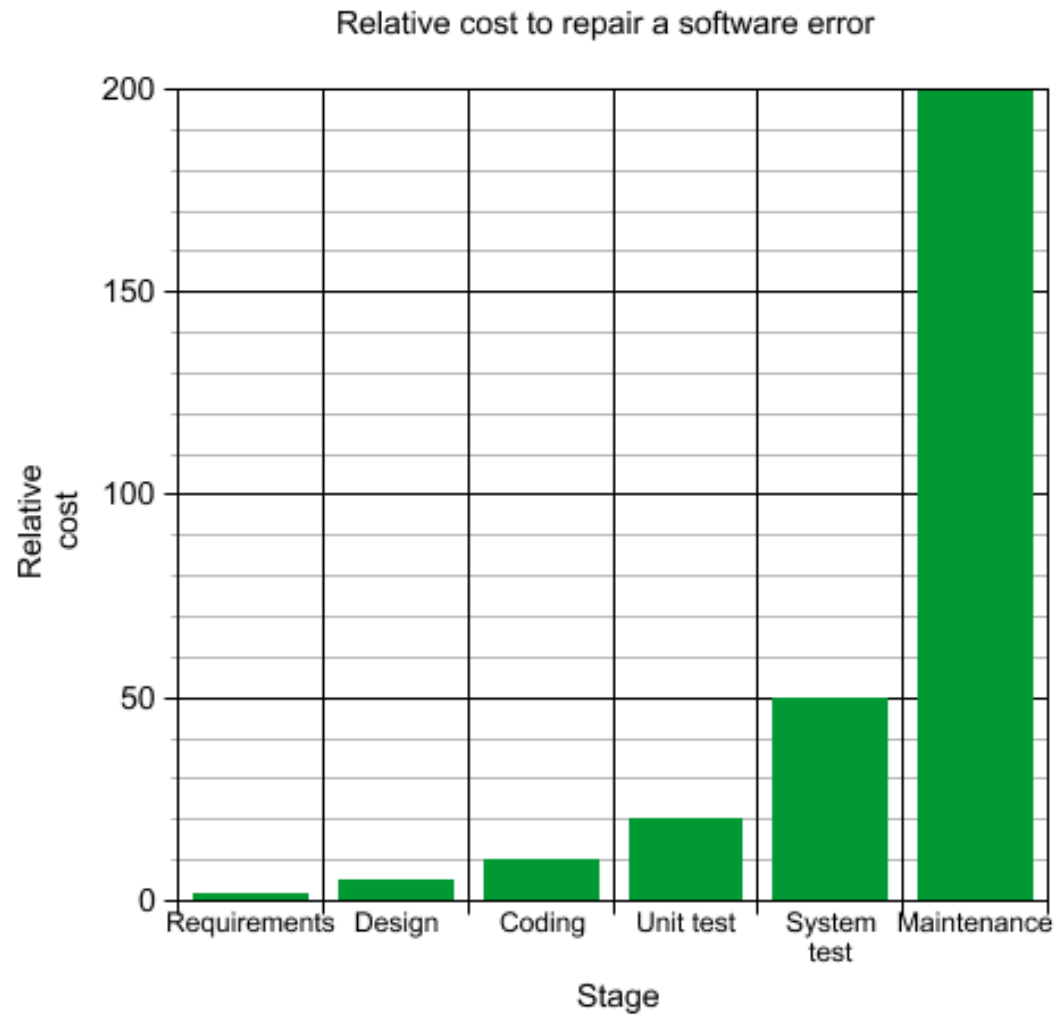
# Requirements document

- Due 10/8 (Monday!)
- Includes:
  - High level product specification
  - Users & use cases
  - Feature specification
  - UI diagrams
  - Product deliverables

# Approach to requirements

- Specify the problem, not the solution
- Divorced from implementation!
- Very user-centric

# Why they matter



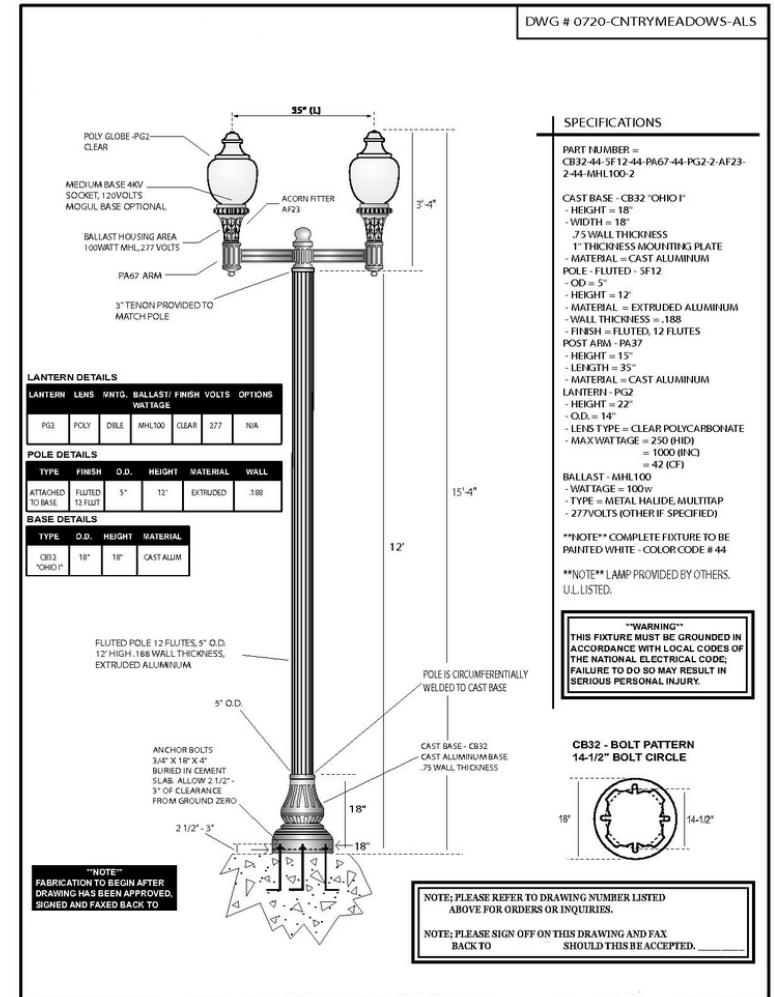
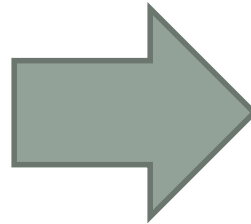
# Types of requirements

- Behavioral
  - Functionality
  - Performance
  - Security
  - Fault-tolerance
- Developmental quality attributes
  - Testability
  - Changeability
  - Maintainability

# Good requirements are:

- Complete
- Implementation independent
- Unambiguous and consistent
- Precise
- Verifiable

# Making requirements



# Preliminary requirements

- User can choose lamp parts
- App generates a schematic
- App shows price comparisons for competitors with similar lamp parts
- App is compatible with all browsers and fits into existing website
- App uses AutoCAD files the designers already have
- Process is easy for users



# Problems

- What does “all” browsers mean?
- Do all lamp parts go together?
- How do users choose parts?
- What about incomplete lamps?
- How do we compare to competitors?
- What does “easy for users” mean?

# Exercise

- Remote with 3 buttons (B1, B2, B3)
- Car detects press at 30 yards
- Car detects remote within 4 feet
- Keyless ignition system
- Key opens driver's side door



# The problem

- Define requirements for
  - Locking the doors
  - Unlocking the doors
  - Opening the trunk
- Specify four additional features for the remote
- Submit this **individually** to dropbox by 10/5 (Friday!)