Pre-Project Activities

Text Chapters 5 and 6

Pre-Project Activities

1. Contract Review
2. Development Plan
3. Quality Plan

Reality Check...

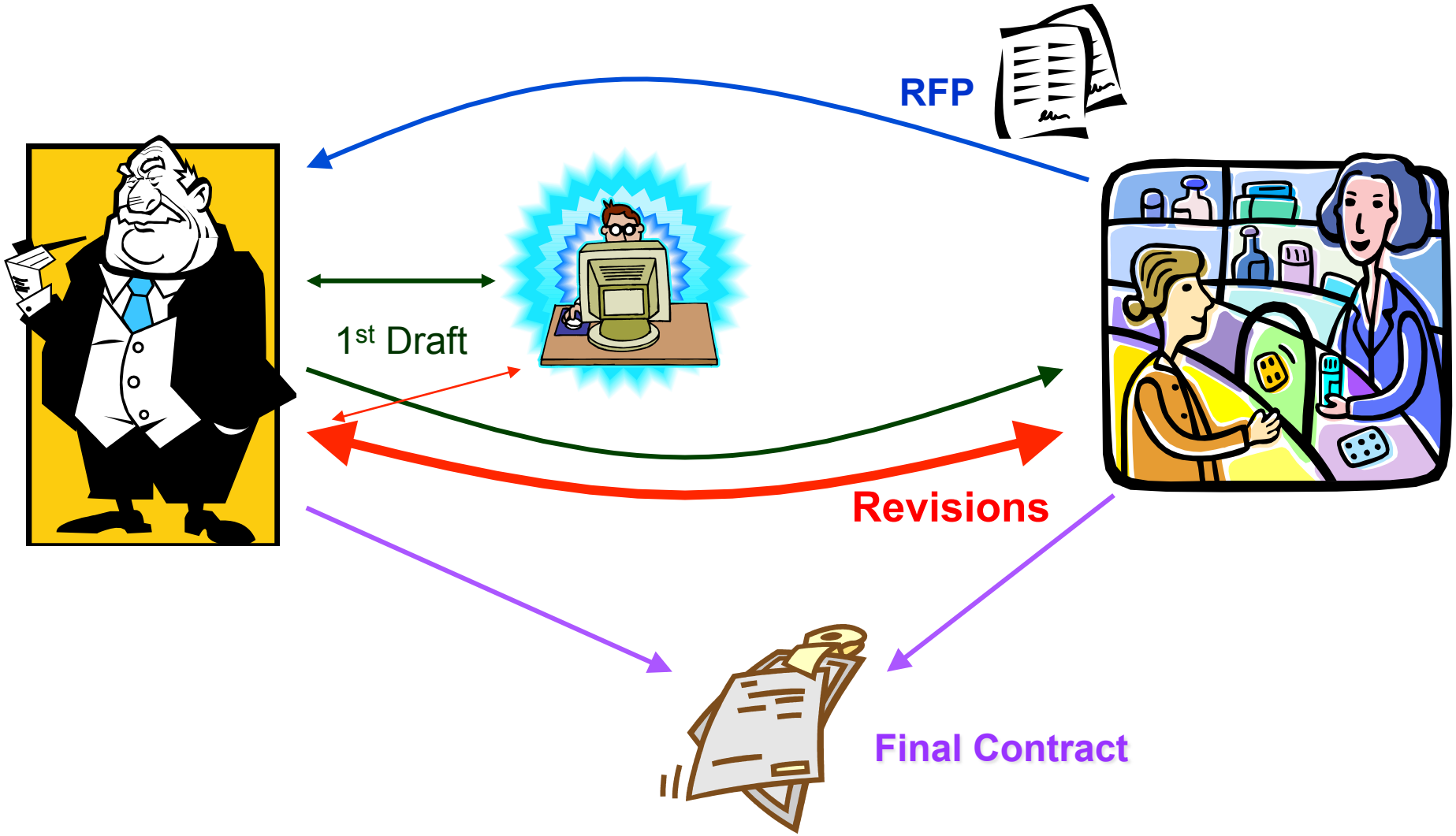
Q: Why should the software geeks worry about the contract?

A: Because the software team must do the work and assure the product's quality.

- ☐ loosely defined requirements
- ☐ unrealistic budgets
- ☐ unrealistic schedules

A: Contract review is required by ISO 9001

Contract Review Process



What to look for in 1st Draft

- customer reqs clarified and documented?
- alternative approaches examined?
- risks identified?
- costs and time estimates reasonable?
- both customer and creator have capacity?
- subcontractor participation clear?
- proprietary rights?
- relationship between customer and creator specified? *(see next slide)*

text section 5.3

Customer - Developer Interface

- communication channels
- project deliverables and acceptance criteria
- formal phase approval process
- customer design and test follow-up method
- customer change request orders

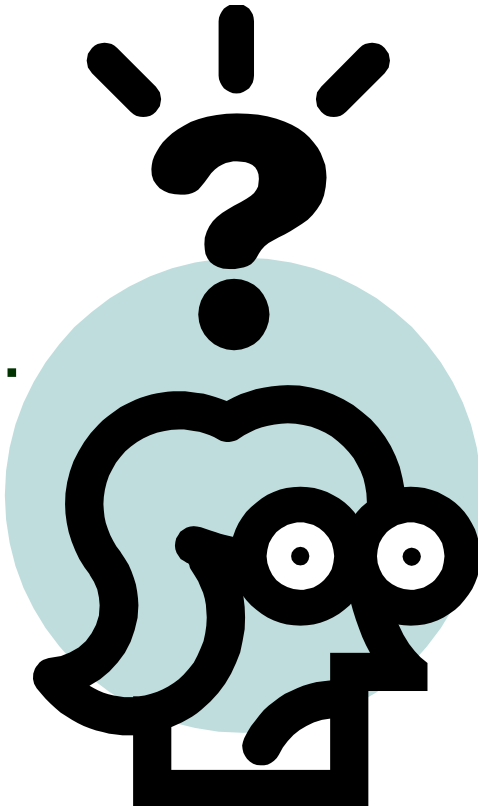
text pages 80-81

Subsequent Draft Reviews

- no unclarified issues remain
- no new additions or changes
- all understandings are documented

Difference in SRS and Contract

- **SRS** = What it must do
  - not How to create it
- **Contract** = How to get paid, …



??? noticeably absent from text ???

Items in the software contract

- **Contract Attachments**
  - Systems specifications
  - All responses and other materials completed from the Request for Proposal (RFP), including the completed system requirements
  - An Implementation Plan identifying the tasks to be completed, the assigned responsibilities and the scheduled completion dates
- **Deliverables**
  - Design
  - Hardware
  - Networking provision, connectivity, ISP, portal connectivity
  - Source code
  - Documentation
  - Training
  - Initial support and maintenance
  - Continuing support and maintenance

http://www.epmbook.com/resources/contractchecklist.doc

- **Delivery**
  - Timetable
  - Price reduction or penalty for delays
  - Trial period
- **Acceptance Criteria**
- **Use and Ownership of Software, Hardware and Services**
- **Confidentiality**
  - Client data
  - Client's business methods and trade secrets
  - Vendor-related information that is subject to non-disclosure

- **Installation and Training**
  - Timeframe of installation
  - Amount of disruption to client's operations
  - Minimum hardware and software configuration to be provided by client for vendor's hardware and software to operate upon or in conjunction with
  - All appropriate education required by client to successfully implement and operate system
  - Period of time that training will be available
  - Training location
  - Facilities required to provide training

- **Support and Maintenance**
  - Amount and nature of implementation support at no additional cost
  - Cost of annual maintenance
  - Starting time for maintenance (eg after warranty period)
  - Support the vendor can provide in the event of a disaster

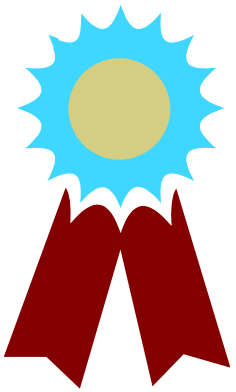
- **Events Constituting Default**
  - Failure to deliver
  - Failure of software or hardware to perform according to specifications
  - Unreliability of software or hardware
  - Failure of vendor to correct malfunctions within an agreed-on time period
  - Failure of vendor to provide support services
  - Bankruptcy of vendor

- **Default and Malfunction Remedies**
  - Termination of agreement
  - Refund of money paid and costs incurred
  - Replacement of software or hardware by vendor
  - Repair of software or hardware by vendor
  - Downtime credits
  - Backup facility in the event of malfunction
  - Time to correct malfunctions, which extends the warranty period

Purpose of **SRS** (IEEE 830)

- Functionality
  - What is the software supposed to do?
- External Interfaces
  - How does the software interact with people, the system's hardware, other hardware, and other software?
- Performance
  - What is the speed, availability, response time, recovery time of various software functions, etc.?
- Attributes
  - What are the portability, correctness, maintainability, security, etc. considerations?
- Design constraints
  - Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment, etc.?

A Good SRS (IEEE 830)



- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

Example Quality Requirement

- Bad

  no mention of usability

- Poor

  The system will be user friendly.

- Fair

  The system's graphical user interface will be designed so that current Supply Office workers will be able to effectively perform their routine tasks after one day of user training.

- Better

  After 8 hours of training, 80% of workers with average domain knowledge will be able to perform 70% of daily tasks (defined in section 5.1.4 of this document) in less than 15 minutes.

Example Quality Requirement

- Bad

  no mention of performance

- Poor

  The system will not slow down noticeably if more than 20 users are using the system.

- Good

  Given 1 to 20 users, the system will always respond within 3 seconds for 80% of operations.

  Given 20 to 50 users, …

Components of the **Development Plan**

- Work Schedule
  - PERT charts and Gantt Charts
  - Deliverables
  - Milestones
- Staff Organization
- Risk Management Actions
- Development Tools
- Development Standards

see CSCI621 for details

text section 6.2

Components of the **Quality Plan**

- Quality Goals
  - hopefully the SRS is some help
- Review Activities
  - schedule, type of review, scope, responsible person
- Software Test Plan
  - type of unit tests and coverage, integration plan
- Acceptance Tests for Sub-contract Software
- Configuration Tools and Procedures

text section 6.3

When a Quality Plan isn't necessary

- What about small projects?
- What about internal projects?

Next Topic…

- Life Cycles
  - waterfall++