### Requirements

1. Elicitation
2. Analysis
3. Specification
4. Modeling
5. Formalization
6. Validation
7. IEEE Std 830
8. Natural language specifications
9. Use-case
10. Actor
11. Classes
12. Data model
13. Non-functional requirements

### Design and architecture

1. Box-and-line diagram
2. Time development
3. Run-time deployment
4. Coupling - Cohesion
5. Scale up - scale out
6. Architecture styles
7. Pipes and filters
8. Layers
9. Client-server
10. Design patterns
11. Observer
12. Façades
13. UML: sketching, blueprinting, programming language
14. Class diagram: attributes, associations, composition, generalization
15. Object diagram: sequence diagram
16. State machine diagrams
17. Activity diagrams
18. Deployment diagrams

### Testing

1. Error, Fault, Failure
2. Black-box testing
3. White-box testing
4. Oracle
5. Equivalence class testing
6. Boundary value testing
7. Control graph testing coverage
8. Data-flow testing coverage
9. Unit testing
10. Regression testing
11. Integration testing
12. System testing
13. Acceptance testing
14. Benchmarking
15. Pilot testing
16. Alpha test
17. Beta test
18. Installation testing
19. Parallel testing

### Planning and processes

1. Project, Process
2. Stakeholders
3. SMART goals
4. Milestone, Tollgate
5. Critical path
6. COCOMO
7. Agile
8. Extreme programming (XP)
9. Scrum
10. Models
11. Waterfall
12. V-model
13. Iterative development
14. Incremental development
15. Time-boxing
16. Processes and methodologies
17. RUP: Inception, Elaboration, Construction, Transition
18. Agile
19. Extreme programming (XP)
20. Configuration Management

### Quality factors

1. Usability engineering
2. Relevance
3. Efficiency
4. Attitude
5. Learnability
6. Reliability engineering
7. Failure intensity
8. Safety
9. Hazard, Incident, Accident
10. Inspection
11. Roles
12. Process
13. Inspection record
14. Weaker methods
15. Management
16. Software metrics
17. Software quality factors
18. ISO 9000
19. Total Quality management
20. CMMI or CMM levels 2 and 3