Nathan White
Aron Kageza
Hanquing Guo

20180424
Rev 1.0

Notes:

Evaluating Software Models

1) {'relative error': ['actual development effort', 'estimated development effort', 'software model', 'predictive value']} : pg. 1
2) {'mean relative error': ['mean', 'sum', 'predict', 'criteria']} : pg. 1
3) {'magnitude of the relative error': ['bias', 'absolute', 'adjust', 'magnitude']} : pg. 1
4) {'Mean magnitude of relative error': ['mean', 'effort estimation', 'average', 'acceptable']} : pg. 1

Object-Oriented Analysis and Design

5) {'OO Analysis': ['investigation', 'describe', 'concept', 'problem domain']} : pg. 1
6) {'OO Design': ['define ', 'logical software objects', 'implement', 'Object']} : pg. 1
7) {'Conceptual model': ['abstract', 'object', 'block', 'attributes']} : pg. 2
8) {'collaboration diagram': ['message', 'data flow', 'illustrating', 'function']} : pg. 2
9) {'class diagrams': ['components', 'object block', 'combined model', 'function and attribute']} : pg. 3
10) {'UML': ['object-oriented analysis', 'methodology', 'visualizing', 'development life cycle']} : pg. 3
11) {'multiplicity': ['relationship', 'interaction', 'objects', 'number']} : pg. 8
12) {'navigation': ['direction', 'multiplicity', 'arrow', 'association']} : pg. 8
13) {'inheritance': ['relationship', 'super-class', 'sub-class', 'common attributes']} : pg. 8
14) {'include relationship': ['function', 'reuse', 'use case', 'consequence']} : pg. 8
15) {'misuse case': ['privacy and security', 'hostile', 'hacker', 'threat']} : pg. 10
16) {'actors': ['interact', 'system', 'player', 'relationship']} : pg. 5
17) {'attributes': ['class', 'definition', 'requirement', 'variable']} : pg. 7
18) {'operations': ['function', 'interaction', 'behavior', 'parameters']} : pg. 7
19) {'class': ['Types', 'Primitives', 'Inheritance', 'School']} : pg. 4

Risk Management

20) {'Risk Strategy': ['Reactive', 'Proactive', 'Plan', 'Minimize']} : pg. 1
21) {'Software Risks': ['Project', 'Technical', 'Business', 'Categories']} : pg. 2
22) {'Risk Identification': ['Size', 'Impact', 'Probability', 'Quantify']} : pg. 2

Introduction to Software Engineering

23) {'Waterfall Model': ['Feedback Loops', 'Documentation-driven', 'Easier Maintenance', 'Lacking Client Feedback', 'Tiered']} : pg. 2

Introduction to software engineer

24) {'Reuse Model': ['Refine', 'Object', 'Copy', 'Extract']} : pg. 5

Introduction to software engineer:

25) {'Confidentiality': ['Employer', 'Secret', 'Respect ', 'Leak']} : pg. 8

What is project planning:

26) {'Code Modification': ['Change', 'Reuse', 'Modify software', 'Assessment']} : pg. 6

Design Principles

27) {'Abstraction': ['Module', 'Object', 'Class', 'Design']} : pg. 1

28) {'Software Design': ['Computer Science', 'Structure', 'System', 'Analysis']} : pg. 1

29) {'Fan-out': ['Module', 'Fan-in', 'Call', 'Number']} : pg. 3

30) {'Fan-in': ['Module', 'Fan-out', 'Number', 'Call']} : pg. 3

31) {'Cohesion': ['Module', 'Association', 'Function', 'Coupling']} : pg. 3

32) {'Coupling': ['Module', 'Class', 'Interdependence', 'Cohesion']} : pg. 3

33) {'Functional Cohesion': ['Module', 'All', 'Task', 'Function']} : pg. 4

34) {'Sequential Cohesion': ['Module', 'Input', 'Output', 'Function']} : pg. 4

35) {'Communication Cohesion': ['Input', 'Output', 'Same', 'Function']} : pg. 4

36) {'Procedural Cohesion': ['Different', 'Control', 'Function', 'Unrelated']} : pg. 4

37) {'Temporal Cohesion': ['Function', 'Time', 'Module', 'Permanent']} : pg. 4

38) {'Logical Cohesion': ['Function', 'Outside', 'Flag', 'Execute']} : pg. 4

39) {'Coincidental Cohesion': ['Function', 'Random', 'Accidental', 'Unexpected']} : pg. 4

Use Case-based Requirements

40) {'Use case': ['Object Oriented', 'Actors', 'Actions', 'Oval']} : pg. 2

41) {'Scenario': ['Use case', 'Actions', 'Sequence', 'Behavior']} : pg. 2

42) {'Actor': ['User', 'Use Case', 'Information', 'Stick']} : pg. 3

43) {'Use Case Diagram': ['Use Case', 'Actors', 'Visual', 'Picture']} : pg. 5

44) {'Preconditions': ['Main Flow', 'Use Case', 'Before', 'Start']} : pg. 6

45) {'Main Flow': ['Use Case', 'Actor', 'Steps', 'Sub flow']} : pg. 6

46) {'extend relationship': ['condition', 'optional behavior', 'scenario', 'exceptional']} : pg. 9

What is Project Planning

47) {'BRAK': ['Code', 'Requirements', 'Estimation', 'Volatility', 'Text']} : pg. 7

Introduction to software engineering

48) {'Extreme Programming': ['Stories', 'Features', 'Builds', 'Test Cases', 'Pair', 'Continuous']} : pg. 3

49) {'Synchronize and Stabilize': ['Microsoft', 'Requirements', 'Specifications', 'End of day']} : pg. 4

50) {'Fountain Model': ['Waterfall', 'Cycles', 'Serial', 'Dependency']} : pg. 4

51) {'Spiral Model': ['Risk', 'Precede', 'Evaluate', 'Plan']} : pg. 4

52) {'Object Oriented Programming': ['Class', 'Object', 'Inheritance', 'Encapsulation']} : pg. 4

53) {'Unit test': ['Program', 'Part', 'Code', 'Function']} : pg. 2

54) {'Maintenance': ['Cost', 'Time', 'During', 'Bug']} : pg. 2

55) {'Documentation': ['Text', 'Customer', 'Tutorial', 'Guideline ']} : pg. 6

56) {'Maintainability': ['Needs', 'Software', 'Technical support', 'Time']} : pg. 7

57) {'Dependability': ['Reliable', 'Robust', 'Availability', 'System']} : pg. 7

58) {'Efficiency': ['Trustworthy', 'Unused resources', 'Efficient', 'Useful output']} : pg. 7

59) {'Usability': ['Discover-able', 'Usable', 'Satisfaction', 'Errors']} : pg. 7

The Structured Specification

60) {'Data Dictionary': ['Document', 'Meaning', 'Description', 'Terms']} : pg. 3

61) {'DFD': ['Design', 'Data', 'Diagram', 'Flow']} : pg. 1

62) {'Functional Specification': ['Analyst', 'Requirements', 'User', 'Document']} : pg. 1

63) {'Data Store': ['DFD', 'Keep', 'Diagram', 'Design']} : pg. 1

64) {'Terminators': ['DFD', 'External', 'Sources', 'Sinks']} : pg. 1

65) {'Contextual Level': ['DFD', 'Zero', 'Top', 'Design']} : pg. 2

66) {'Decision Trees': ['Design', 'Diagram', 'Branch', 'Picture']} : pg. 4

67) {'ERDs': ['Entity', 'Diagram', 'Design', 'Data']} : pg. 4

68) {'Functional Primitive': ['DFD', 'Last', 'Design', 'Break']} : pg. 4

69) {'Structured Specification': ['functional specification', 'readings', 'user', 'document']} : pg. 1

70) {'Terminator': ['DFD', 'external data sources', 'diagrams', 'destinations']} : pg. 1

71) {'Component Variations': ['flow type', 'process', 'store', 'operations']} : pg. 2

Notes on Software Testing

72) {'Black Box Testing': ['Specifications', 'API', 'Inputs ', 'Outputs']} : pg. ?

73) {'White Box Testing': ['Method', 'Known', 'Source', 'Inside']} : pg. ?

74) {'Naive Testing': ['Simplistic', 'Execution', 'Cases', 'Guessing']} : pg. ?

75) {'Exhaustive Path': ['Brute', 'All', 'Combinations', 'Traverse']} : pg. ?

76) {'Test Plan': ['Management', 'Coverage', 'Verify', 'Box']} : pg. 3

77) {'Random Input Testing': ['Bad', 'Poor', 'Stochastic', 'Noise']} : pg. 4

78) {'Equivalence Partition': ['Sets', 'Intersection', 'Union', 'Divide']} : pg. 4

79) {'Boundary Value Analysis': ['One', 'Limits', 'Ranges', 'Condition']} : pg. 4

80) {'Error Guessing': ['Experience', 'Analysis', 'Intuition', 'Knowledge']} : pg. 4

81) {'Logic Coverage': ['Statement ', 'Path', 'Branch ', 'Condition']} : pg. 11

82) {'Infeasible Path': ['Unreachable', 'Traversal', 'Execution', 'Impossible']} : pg. 12

83) {'Statement Coverage': ['Verify', 'Once', 'Logic', 'Strategy']} : pg. 12

84) {'Path Coverage': ['Potential', 'Conditions', 'All', 'Combinations']} : pg. 13

85) {'Condition Coverage': ['Test', 'Each', 'Logical', 'Boolean']} : pg. 14

86) {'Defect Density Model': ['Testing', 'Time', 'Effort', 'Minimized']} : pg. 18

87) {'Saturation Effect': ['False', 'Reliability', 'Estimates', 'Misled']} : pg. 18

Design Techniques

88) {'Data Flow Oriented Design': ['process', 'DFD', 'structure', 'program']} : pg. 1

89) {'Top-Down Function Design': ['function', 'module', 'software', 'information driven']} : pg. 1

What is project planning

90) {'Planning': ['Task', 'Organization', 'Design', 'Dependency']} : pg. 1

91) {'Metrics': ['Models', 'Predicts', 'Measures', 'System']} : pg. 2

92) {'Critical Path': ['Plan', 'Complete', 'Longest', 'Project']} : pg. 2

93) {'Function Points': ['Measure', 'Compute', 'LOC', 'Cost']} : pg. 4

94) {'Empirical Validation': ['Analyze', 'Data Collection', 'Proof', 'Experiment ']} : pg. 2

95) {'Effort': ['Cost', 'Duration', 'Work time', 'Measure']} : pg. 2

96) {'Risk': ['Avoid', 'Estimate', 'Evaluate', 'Analyze']} : pg. 1

97) {'Estimates': ['Approximation', 'Calculation', 'Guess', 'Analyze']} : pg. 3

98) {'Size Metrics': ['Operators', 'Operands', 'Occurrences', 'LOC']} : pg. 3

99) {'Macro-Model': ['Scale', 'Productivity', 'Big', 'Size']} : pg. 4

100) {'Early Planning': ['Estimate', 'Schedule', 'Tasks', 'Dependency']} : pg. 1

101) {'Emperical': ['Verified', 'Science', 'Theoretical', 'Data']} : pg. 2

102) {'Code Metric': ['Code', 'Data structure', 'Complexity ', 'Logic']} : pg. 3

103) {'Effort estimation': ['Past experience ', 'Data', 'Formal model', 'Average methods']} : pg. 4

104) {'Total effort': ['Development cost', 'Theoretical', 'Statistical', 'Mathematical']} : pg. 4

105) {'Criteria': ['Validity', 'Judge', 'Model', 'Objectivity']} : pg. 5

106) {'COCOMO': ['Tool', 'Estimate', 'Schedule', 'Cost']} : pg. 5

107) {'Product Attributes': ['Software Reliability', 'Data base', 'Complexity ', 'Cost driver']} : pg. 6

108) {'Personnel Attributes': ['Analyst Capability', 'Application Experience', 'Programmer Capability', 'Programming Language ']} : pg. 6

109) {'Computer Attributes': ['Execution Time', 'Storage Constraint', 'Turnaround time', 'Virtual Machine']} : pg. 6

110) {'Estimating Techniques': ['Algorithmic', 'Top-Down', 'Bottom-Up', 'Expert Judgment', 'Analogy']} : pg. 8

Software Quality Assurance

111) {'Reliability': ['Crash', 'Intended', 'Test', 'Quality']} : pg. 1

112) {'Process Specific Activities': ['KPA', 'Maturity', 'Levels', 'Testing']} : pg. 3

113) {'Quality Assurance': ['Review', 'Methods', 'Procedures', 'Guaratee']} : pg. 3

114) {'Prevention Costs': ['Quality', 'Planning', 'Reviews', 'Training']} : pg. 2

115) {'Failure Cost': ['Analysis', 'Project', 'Retest', 'Postmortem']} : pg. 2

Notes on software testing

116) {'Path Testing': ['Input', 'Output', 'Execution', 'Follow']} : pg. 11

117) {'Symbolic Evaluation': ['Methods', 'Dynamic', 'Global', 'Testing']} : pg. 10

Book:

Software Engineering

118) {'Mongolian Horde': ['Add', 'Behind', 'Mythical Man Month', 'Schedule', 'Productivity']} : pg. 24

Project Management Concepts

119) {'Steakholders': ['Managers', 'Practitioners', 'Customers', 'End User']} : pg. 687

120) {'Organizational Paradigms': ['Closed', 'Random', 'Open', 'Synchronous']} : pg. 690

121) {'Agile': ['Satisfaction', 'Incremental', 'Informal', 'Simplicity']} : pg. 691

122) {'Software Scope': ['Context', 'Information', 'Function', 'Features']} : pg. 694

Understanding Requirements

123) {'requirement': ['Customer', 'Model', 'Change', 'Design']} : pg. 131

Software Process Structure

124) {'Software Engineering': ['Process Management', 'System', 'Development', 'Discipline']} : pg. 15

125) {'Process Flow': ['Iterative', 'Linear', 'Parallel', 'Umbrella']} : pg. 31

126) {'Umbrella Activities': ['tracking', 'risk', 'quality', 'configuration', 'technical']} : pg. 31

127) {'Communication': ['Talk', 'Speak', 'Telephone', 'Internet']} : pg. 34

Product Metrics

128) {'External Inputs': ['User', 'Outside', 'Function', 'Output']} : pg. 659

129) {'External Outputs': ['User', 'Information', 'Report', 'Input']} : pg. 659

130) {'Measure': ['Size', 'Dimension', 'Extent', 'Capacity']} : pg. 655

User Interface Design

131) {'Analysis': ['Design', 'Implementation', 'Document', 'Critical']} : pg. 317

Quality Managementt

132) {'Quality of Conformance': ['Degree', 'Requirements', 'End-user', 'Comply']} : pg. 414

Modeling

133) {'Requirements Engineering': ['Communication', 'Needs', 'Process', 'Project']} : pg. 132

134) {'Inception': ['Start', 'Beginning', 'Product', 'Idea']} : pg. 133

135) {'Ubiquitous Computing': ['Processors', 'Distributed', 'Everywhere', 'Commonplace']} : pg. 133

136) {'Elicitation': ['Ask', 'Objective', 'Accomplish', 'Use']} : pg. 134

137) {'Problem of Scope': ['Boundary', 'Space', 'Domain', 'Context']} : pg. 134

138) {'Problem of Understanding': ['Unsure', 'Limitations', 'Communication', 'Partial']} : pg. 134

139) {'Problem of Volatility': ['Requirements', 'Change', 'Time', 'Quickly']} : pg. 134

140) {'Goal-Oriented': ['Long-Term', 'Purpose', 'Achieve', 'Requirements']} : pg. 134

141) {'Elaboration': ['Expand', 'Elicitation', 'Inception', 'Refined']} : pg. 134

142) {'User Scenario': ['Describe', 'Actors', 'Scene', 'Workflow']} : pg. 135

143) {'Negotiation': ['Compromise', 'Reconcile', 'Argue', 'Requirements']} : pg 135

144) {'Iterative Approach': ['Sequential', 'Technique', 'Systematic', 'Requirements']} : pg 135

145) {'Specification': ['Document', 'Model', 'Prototype', 'Description']} : pg. 135

146) {'Validation': ['Verify', 'Assess', 'Quality', 'Review']} : pg. 136

147) {'Requirements Managment': ['Activities', 'Life Cycle', 'System', 'Track']} : pg. 138

148) {'Software Engineer': ['Design', 'Manage', 'System', 'Programmer']} : pg. 138

149) {'Collaboration': ['Together', 'Group', 'Coordinate', 'Work']} : pg. 140

150) {'Questions': ['Ask', 'Gather', 'Query', 'Inquire']} : pg. 140

151) {'Business Need': ['Identify', 'Recognize', 'Project', 'Inception']} : pg. 133

152) {'Task Analysis': ['Perform', 'Circumstance', 'Sequence', 'Workflow']} : pg. 326

153) {'Task Elaboration': ['Function', 'List', 'Breakdown', 'Design']} : pg. 328

154) {'Workflow Analysis': ['Multiple', 'Process', 'User', 'Technique']} : pg. 328

155) {'Swimlane': ['Diagram', 'Pool', 'Water', 'Users']} : pg. 329

156) {'Design Pattern': ['Abstraction', 'Solution', 'Interface', 'Common']} : pg. 334

157) {'Response Time': ['Delay', 'Lag', 'Bad', 'Variable']} : pg. 335

158) {'Accessibility': ['Impaired', 'Disability', 'Usable', 'Guidelines']} : pg. 336

Design Concepts

159) {'Pseudocode': ['Design', 'Program', 'Flow', 'Language']} : pg. 247

Process and Project Metrics

160) {'Defect': ['Error', 'Bug', 'Crash', 'Failure']} : pg. 450

Risk Management

161) {'Performance Risk': ['Requirements', 'Uncertainty', 'Cost', 'Support']} : pg. 782

162) {'Cost Risk': ['Budget', 'Maintain', 'Price', 'Uncertainty']} : pg. 782

163) {'Support Risk': ['Uncertainty', 'Correct', 'Help', 'Enhance']} : pg. 782

164) {'Schedule Risk': ['Uncertainty', 'Project', 'Time', 'Deliver']} : pg. 782

165) {'Risk Estimation': ['Projection', 'Likelihood', 'Impact', 'Determine']} : pg. 782

166) {'RMMM': ['Plan', 'Probability', 'Strategy', 'Minimize']} : pg. 785

167) {'Cutoff Line': ['Risk', 'Terminate', 'Estimate', 'Identification']} : pg. 784

168) {'Fire-fighting': ['Crisis', 'Panic', 'Project', 'Problem']} : pg. 778

169) {'Uncertainty': ['Unknown', 'Unsure', 'Minimize', 'Risk']} : pg. 778

Quality Management

170) {'User Interface': ['Input', 'Output', 'Keyboard', 'Screen', 'Website']} : pg. 672

171) {'Cyclomatic Complexity': ['Linear', 'Independent', 'Metric', 'Testing']} : pg. 455

172) {'Validation Testing': ['Requirements', 'Software', 'Verify', 'Develop']} : pg. 484

173) {'System Testing': ['Mesh', 'Integrate', 'High', 'Performance']} : pg. 471

174) {'Smoke Testing': ['Integration', 'Development', 'Time-critical', 'Assessment']} : pg. 479

175) {'Alpha Tests': ['Developer', 'End-User', 'Process', 'Integrated']} : pg. 485

176) {'Beta Test': ['End-user', 'Developer', 'Live', 'Environment']} : pg. 485

177) {'Stress Testing': ['Load', 'Volume', 'Resources', 'Demand']} : pg. 487

178) {'Configuration Testing': ['Environment', 'Deployment', 'Documentation', 'Installed']} : pg 487

179) {'Security Testing': ['Penetration', 'Vulnerability', 'Attack', 'Hack']} : pg. 486

180) {'Recovery Testing': ['Failure', 'Automatic', 'Human-intervention', 'Resilience']} : pg. 486

Managing Software Projects

181) {'Resources': ['Human', 'Software', 'Environmental', 'Hardware']} : pg. 732

182) {'Decomposition': ['Break', 'Parts', 'Smaller', 'Problem']} : pg. 734

The Nature Of Software

183) {'Embedded Software': ['Firmware', 'Hardware', 'Controller', 'Processor', 'PROM']} : pg. 7

184) {'Cloud Computing': ['Internet', 'Remote', 'Resource', 'Azure/Google/AWS']} : pg. 10

185) {'Process Pattern': ['Solution', 'Resolution', 'Template', 'Planning']} : pg. 35

186) {'Modeling': ['Sketch', 'Mock-up', 'Miniature', 'Scale']} : pg. 17

187) {'Customer': ['Request', 'Pay', 'Requirements', 'Person']} : pg. 17

188) {'Obsolete': ['Outdated', 'Expired', 'Extinct', 'Deprecated']} : pg. 22

189) {'HTML': ['Hypertext', 'HTTP', 'Protocol', 'Browser']} : pg. 10

190) {'Algorithm': ['Code', 'Program', 'Processor', 'Computer']} : pg. 21

191) {'Compiler': ['Assemble', 'Execute', 'Program', 'Code']} : pg. 7

Software Testing Strategies

192) {'Independent Test Group': ['Separate', 'Complete', 'Procedure', 'Correct']} : pg. 469

Software Quality Assurance

193) {'Standards': ['Ensure', 'Conform', 'Voluntary', 'SQA']} : pg. 450

194) {'Audit': ['Review', 'Conformity', 'Quality', 'Certification']} : pg. 450
195) {'Six Sigma': ['Standard', 'High', 'Define', 'Measure']} : pg. 458
196) {'Failures-in-time': ['Statistical', 'Measure', 'Reliability', 'Billion']} : pg. 460
197) {'Mean-time-to-failure': ['Average', 'Duration', 'Program', 'Crash']} : pg. 450

Process Models

198) {'Prescriptive Model': ['Structure', 'Order', 'Sequential', 'Process']} : pg. 41
199) {'Process Model': ['Activity', 'Action', 'Flow', 'Software']} : pg. 41
200) {'Prototyping': ['Software', 'Requirement', 'Model', 'Evolutionary']} : pg. 45