

## APPENDIX C

### CLASSIFICATION OF TOPICS ACCORDING TO BLOOM’S TAXONOMY

#### INTRODUCTION

Bloom’s taxonomy is the best known and most widely used classification of cognitive educational goals. In order to help all audiences in that field who wish to use the Guide as a tool in designing course material, programs or accreditation criteria, the project was mandated to provide a first draft evaluation of the topics included in the Knowledge Areas breakdowns according Bloom’s Taxonomy. This should only be seen as a jump-start document to be further developed by other steps in other, related projects.

Knowledge Area Specialists were asked to provide an Appendix that states for each topic at which level of Bloom’s taxonomy a “graduate plus four years experience” should “master” this topic. The resulting table could also be used by the specialists themselves as a guide to choose the amount and level of reference material appropriate for each topic.

This appendix contains, for each Knowledge Area<sup>1</sup>, a table identifying the topics and the associated Bloom’s taxonomy level of understanding on each topic for a graduate with four years experience. The levels of understanding from lower to higher are: knowledge, comprehension, application, analysis, synthesis, and evaluation. The version used can be found at <http://www.valdosta.peachnet.edu/~whuitt/psy702/cogsys/bloom.html>

#### SOFTWARE REQUIREMENTS

TOPIC	Bloom Level
<b>Requirements engineering process</b>	
Process models	Knowledge
Process actors	Knowledge
Process support	Knowledge
Process quality and improvement	Knowledge
<b>Requirements elicitation</b>	
Requirements sources	Comprehension
Elicitation techniques	Application
<b>Requirements analysis</b>	
Requirements classification	Comprehension
Conceptual modeling	Comprehension
Architectural design and requirements allocation	Analysis
Requirements negotiation	Analysis
<b>Requirement specification</b>	
The requirements definition document	Application
The software requirements specification (SRS)	Application
Document structure	Application
Document quality	Analysis
<b>Requirements validation</b>	
The conduct of requirements reviews	Analysis
Prototyping	Application
Model validation	Analysis
Acceptance tests	Application
<b>Requirements management</b>	
Change management	Analysis
Requirement attributes	Comprehension
Requirements tracing	Comprehension

<sup>1</sup> Ratings for the Software Construction Area and the Software Maintenance Knowledge Area have been omitted for this edition.

## SOFTWARE DESIGN

Software Design Topic	Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
<b>I. SOFTWARE DESIGN BASIC CONCEPTS</b>						
General design concepts		X				
The context of software design		X				
The software design process				X		X
Enabling techniques for software design				X		
<b>II. Key issues in Software Design</b>						
Concurrency			X			
Control and handling of events			X			
Distribution			X			
Exception handling			X			
Interactive systems			X			
Persistence			X			
<b>III. SOFTWARE STRUCTURE AND ARCHITECTURE</b>						
Architectural structures and viewpoints			X			
Architectural styles (macro-architecture)				X		X
Design patterns (micro-architecture)				X		X
Families of programs and frameworks			X			
<b>IV. SOFTWARE DESIGN QUALITY ANALYSIS AND EVALUATION</b>						
Quality attributes				X		
Quality analysis and evaluation tools			X	X		
Measures			X	X		
<b>V. SOFTWARE DESIGN NOTATIONS</b>						
Structural descriptions (static view)			X	X		
Behavioral descriptions (dynamic view)			X	X		

Software Design Topic	Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
<b>VI. SOFTWARE DESIGN STRATEGIES AND METHODS</b>						
General strategies			X			
Function-oriented design			X			
Object-oriented design				X		X
Data-structure centered design		X				
Other methods		X	X			

Note: As mentioned in the URL used as reference for “Bloom’s et al.’s Taxonomy of the Cognitive Domain”, Evaluation has been considered to be at the same level as Synthesis, but using different cognitive processes.

## SOFTWARE CONSTRUCTION

Rating has been omitted for this edition.

## SOFTWARE TESTING

Topic	Bloom's level
<b>A. Testing Basic Concepts and definitions</b>	
Definitions of testing and related terminology	Analysis
Faults vs. failures	Analysis
Test selection criteria/Test adequacy criteria (or stopping rules)	Application
Testing effectiveness/Objectives for testing	Comprehension
Testing for defect identification	Comprehension
The oracle problem	Comprehension
Theoretical and practical limitations of testing	Application
The problem of infeasible paths	Comprehension
Testability	Comprehension
Testing vs. Static Analysis Techniques	Application
Testing vs. Correctness Proofs and Formal Verification	Knowledge
Testing vs. Debugging	Comprehension
Testing vs. Programming	Application
Testing within SQA	Application
Testing within CMM	Knowledge
Testing within Cleanroom	Knowledge
Testing and Certification	Comprehension
<b>B. Test Levels</b>	
Unit testing	Application
Integration testing	Application
System testing	Application
Acceptance/qualification testing	Application
Installation testing	Application
Alpha and Beta testing	Application
Conformance testing/Functional testing/Correctness testing	Application
Reliability achievement and evaluation by testing	Comprehension
Regression testing	Application
Performance testing	Comprehension
Stress testing	Comprehension
Back-to-back testing	Knowledge
Recovery testing	Comprehension
Configuration testing	Comprehension
Usability testing	Comprehension
<b>C. Test Techniques</b>	
Ad hoc	Synthesis
Equivalence partitioning	Application
Boundary-value analysis	Application
Decision table	Knowledge
Finite-state machine-based	Knowledge
Testing from formal specifications	Knowledge
Random testing	Application
Reference models for code-based	Application

Topic	Bloom's level
testing (flow graph, call graph)	
Control flow-based criteria	Application
Data flow-based criteria	Comprehension
Error guessing	Application
Mutation testing	Knowledge
Operational profile	Comprehension
SRET	Knowledge
Object-oriented testing	Application
Component-based testing	Comprehension
GUI testing	Knowledge
Testing of concurrent programs	Knowledge
Protocol conformance testing	Knowledge
Testing of distributed systems	Knowledge
Testing of real-time systems	Knowledge
Testing of scientific software	Knowledge
Functional and structural	Synthesis
Coverage and operational/Saturation effect	Knowledge
<b>D. Test related measures</b>	
Program measurements to aid in planning and designing testing.	Synthesis
Types, classification and statistics of faults	Application
Remaining number of defects/Fault density	Application
Life test, reliability evaluation	Comprehension
Reliability growth models	Knowledge
Coverage/thoroughness measures	Application
Fault seeding	Knowledge
Mutation score	Knowledge
Comparison and relative effectiveness of different techniques	Comprehension
<b>E. Managing the Test Process</b>	
Attitudes/Egoless programming	Application
Test process	Synthesis
Test documentation and workproducts	Synthesis
Internal vs. independent test team	Comprehension
Cost/effort estimation and other process metrics	Application
Termination	Application
Test reuse and test patterns	Application
Planning	Application
Test case generation	Application
Test environment development	Application
Execution	Application
Test results evaluation	Application
Problem reporting/Test log	Application
Defect tracking	Application

## SOFTWARE MAINTENANCE

Rating has been omitted for this edition.

**SOFTWARE CONFIGURATION MANAGEMENT**

SCM TOPIC	Bloom Level
I. Management of the SCM Process	Knowledge
A. Organizational Context for SCM	Knowledge
B. Constraints and Guidance for SCM	Knowledge
C. Planning for SCM	Knowledge
1. SCM Organization and Responsibilities	Knowledge
2. SCM Resources and Schedules	Comprehension
3. Tool Selection and Implementation	Knowledge
4. Vendor/Subcontractor Control	Knowledge
5. Interface Control	Comprehension
D. Software Configuration Management Plan	Knowledge
E. Surveillance of SCM	Comprehension
1. SCM Metrics and Measurement	Comprehension
2. In-Process Audits of SCM	Knowledge
II. Software Configuration Identification	Comprehension
A. Identifying Items to be controlled	Comprehension
1. Software Configuration	Comprehension
2. Software Configuration Items	Comprehension
3. Software configuration item relationships	Comprehension
4. Software Versions	Comprehension
5. Baselines	Comprehension
6. Acquiring Software Configuration Items	Knowledge
B. Software Library	Comprehension
III. Software Configuration Control	Application
A. Requesting, Evaluating, and Approving Software Changes	Application
1. Software Configuration Control Board	Application
2. Software Change Request Process	Application
B. Implementing Software Changes	Application
C. Deviations & Waivers	Comprehension
IV. Software Configuration Status Accounting	Comprehension
A. Software Configuration Status Information	Comprehension
B. Software Configuration Status Reporting	Comprehension
V. Software Configuration Auditing	Knowledge
A. Software Functional Configuration Audit	Knowledge
B. Software Physical Configuration Audit	Knowledge
C. In-process Audits of a Software Baseline	Knowledge
VI. Software Release Management & Delivery	Comprehension
A. Software Building	Comprehension
B. Software Release Management	Comprehension

**SOFTWARE ENGINEERING MANAGEMENT**

Topic	Level
<b>A. Organizational Management</b>	
Policy management	Comprehension
Personnel management	Analysis
Communication management	Analysis
Portfolio management	Comprehension
Procurement management	Knowledge
<b>B. Process/project Management</b>	
Determination and negotiation of requirements	Comprehension
Feasibility analysis	Application
Review/revision of requirements	Comprehension
Process planning	Analysis
Project planning	Application
Determine deliverables	Comprehension
Effort, schedule and cost estimation	Analysis
Resource allocation	Application
Risk management	Synthesis
Quality management	Synthesis
Plan management	Application
Implementation of plans	Application
Implementation of measurement process	Application
Monitor process	Application
Control process	Application
Reporting	Application
Determining satisfaction of requirements	Comprehension
Reviewing and evaluating performance	Application
Determining closure	Application
Closure activities	Comprehension
<b>C. Software Engineering Measurement</b>	
Organizational objectives	Synthesis
Software process improvement goals	Synthesis
Goal-driven measurement selection	Application
Measurement validity	Comprehension
Size measurement	Analysis
Structure measurement	Analysis
Resource measurement	Analysis
Quality measurement	Analysis
Survey techniques and form design	Knowledge
Automated and manual data collection	Knowledge
Model building, calibration and evaluation	Application
Implementation, interpretation and refinement of models	Analysis

## SOFTWARE ENGINEERING PROCESS

Topic	Level
<b>Software Engineering Process Concepts</b>	
Themes	Comprehension
Terminology	Knowledge
<b>Process Infrastructure</b>	
The Software Engineering Process Group	Comprehension
The Experience Factory	Comprehension
<b>Process Measurement</b>	
Methodology in Process Measurement	Comprehension
Process Measurement Paradigms	Comprehension
Analytic Paradigm	Comprehension
Benchmarking Paradigm	Comprehension
<b>Process Definition</b>	
Types of Process Definitions	Application
Life Cycle Framework Models	Application
Software Life Cycle Process Models	Application
Notations for Process Definitions	Application
Process Definition Methods	Application
Automation	Knowledge
<b>Qualitative Process Analysis</b>	
Process Definition Review	Comprehension
Root Cause Analysis	Comprehension
<b>Process Implementation and Change</b>	
Paradigms for Process Implementation and Change	Comprehension
Guidelines for Process Implementation and Change	Comprehension
Evaluating the Outcome of Process Implementation and Change	Comprehension

## SOFTWARE ENGINEERING TOOLS AND METHODS

Topic	Bloom Level
<b>Software Tools</b>	
<b>Software Requirements Tools</b>	Application
Requirements Modeling Tools	Application
Traceability Tools	Comprehension
<b>Software Design Tools</b>	Application
<b>Software Construction Tools</b>	
Program Editors	Application
Compilers and Code Generators	Application
Interpreters	Application
Debuggers	Application
<b>Software Testing Tools</b>	
Test Generators	Comprehension
Test Execution Frameworks	Application
Test Evaluation Tools	Application
Test Management Tools	Comprehension
Performance Analysis Tools	Comprehension
<b>Software Maintenance Tools</b>	
Comprehension Tools	Application
Re-engineering Tools	Knowledge
<b>Software Engineering Process Tools</b>	
Process Modeling Tools	Knowledge
Process Management Tools	Knowledge
Integrated CASE Environments	Application
Process-centered Software Engineering Environments	Comprehension
<b>Software Quality Tools</b>	
Inspection Tools	Comprehension
Static Analysis Tools	Application
<b>Software Configuration Management Tools</b>	
Defect, Enhancement, Issue and Problem Tracking Tools	Application
Version Management Tools	Application
Release and Build Tools	Application
<b>Software Engineering Management Tools</b>	
Project Planning and Tracking Tools	Application
Risk Management Tools	Comprehension
Measurement Tools	Application
<b>Infrastructure Support Tools</b>	
Interpersonal Communication Tools	Application
Information Retrieval Tools	Application
System Administration and Support Tools	Comprehension

Topic	Bloom Level
<b>Miscellaneous Tools Issues</b>	
Tool Integration Techniques	Knowledge
Meta Tools	Comprehension
Tool Evaluation	Application
<b>Software Methods</b>	
<b>Heuristic Methods</b>	Application
Structured Methods	Application
Data-oriented Methods	Application
Object-oriented Methods	Application
Domain-specific Methods	Comprehension
<b>Formal Methods</b>	
Specification Languages	Comprehension
Refinement	Knowledge
Validation/Proving Properties	Comprehension
<b>Prototyping Methods</b>	
Styles	Comprehension
Prototyping Targets	Application
Evaluation	Comprehension
<b>Miscellaneous Method Issues</b>	
Method Evaluation	Application

## SOFTWARE QUALITY

All software engineers are responsible for the quality of the products they build. We consider that the knowledge requirements for topics in Software Quality vary depending on the role of the software engineer. We use the roles of programmer, SQA/VV specialist, and project manager. The programmer will design and build the system, possibly be involved in inspections and reviews, analyze his work products statically, and possibly perform unit test. This person may turn over the products to others who will conduct integration and higher levels of testing, and may be asked to submit data on development tasks, but will not conduct analyses on faults or on measurements. The SQA/VV specialist will plan and implement the processes for software quality analysis, verification, and validation. The project manager of the development project will use the information from the software quality analysis processes to make decisions. Of course, in a small project, the software engineer may have to assume all of these roles, in which case, the highest of the three is appropriate.

Software Quality Topic (Numbered as to Section in this KA)	Bloom Level*, By Job Responsibility		
	<i>Programmer</i>	<i>SQA/VV Spec.</i>	<i>Project Manager</i>
Software Quality Concepts			
Measuring the Value of Quality	Comprehension	Comprehension	Analysis
ISO 9126 Quality Description	Comprehension	Comprehension	Comprehension
Dependability	Comprehension	Comprehension	Comprehension
Special Types of Systems and Quality Needs	Comprehension	Comprehension	Comprehension
Purpose and Planning of SQA and V&V			
Common Planning Activities			
The SQA Plan	Application	Synthesis	Evaluation
The V&V Plan	Application	Synthesis	Evaluation
Activities and Techniques for SQA and V&V			
Static Techniques			
Audits, Reviews, and Inspections	Application	Evaluation	Analysis
Analytic Techniques	Application	Evaluation	Analysis
Dynamic Techniques	Application	Evaluation	Analysis
Measurement Applied to SQA and V&V			
Fundamentals of Measurement	Application	Evaluation	Analysis
Metrics	Application	Evaluation	Analysis
Measurement Techniques	Application	Evaluation	Analysis
Defect Characterization	Application	Evaluation	Analysis
Additional uses of SQA and V&V data	Application	Evaluation	Analysis

\*The levels, in ascending order: Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation