## Khung phân loại trong lĩnh vực Software

Khung phân loại này được tham khảo tại ACM (The 1998 ACM Computing Classification System ).

* S.0 GENERAL
* S.1 PROGRAMMING TECHNIQUES
  + S.1.0 General
  + S.1.1 Applicative (Functional) Programming
  + S.1.2 Automatic Programming
  + S.1.3 Concurrent Programming
    - Distributed programming
    - Parallel programming
  + S.1.4 Sequential Programming
  + S.1.5 Object-oriented Programming
  + S.1.6 Logic Programming
  + S.1.7 Visual Programming
  + S.1.8 Miscellaneous
* S.2 SOFTWARE ENGINEERING
  + S.2.0 General
    - Protection mechanisms
    - Standards
  + S.2.1 Requirements/Specifications
    - Elicitation methods (e.g., rapid prototyping, interviews, JAD)
    - Languages
    - Methodologies (e.g., object-oriented, structured)
    - Tools
  + S.2.2 Design Tools and Techniques
    - Computer-aided software engineering (CASE)
    - Decision tables
    - Evolutionary prototyping
    - Flow charts
    - Modules and interfaces
    - Object-oriented design methods
    - Petri nets
    - Programmer workbench
    - Software libraries
    - State diagrams
    - Structured programming
    - Top-down programming
    - User interfaces
  + S.2.3 Coding Tools and Techniques
    - Object-oriented programming
    - Pretty printers
    - Program editors
    - Reentrant code
    - Standards
    - Structured programming
    - Top-down programming
  + S.2.4 Software/Program Verification
    - Assertion checkers
    - Class invariants
    - Correctness proofs
    - Formal methods
    - Model checking
    - Programming by contract
    - Reliability
    - Statistical methods
    - Validation
  + S.2.5 Testing and Debugging
    - Code inspections and walk-throughs
    - Debugging aids
    - Diagnostics
    - Distributed debugging
    - Dumps
    - Error handling and recovery
    - Monitors
    - Symbolic execution
    - Testing tools (e.g., data generators, coverage testing)
    - Tracing
  + S.2.6 Programming Environments
    - Graphical environments
    - Integrated environments
    - Interactive environments
    - Programmer workbench
  + S.2.7 Distribution, Maintenance, and Enhancement
    - Corrections
    - Documentation
    - Enhancement
    - Extensibility
    - Portability
    - Restructuring, reverse engineering, and reengineering
    - Version control
  + S.2.8 Metrics
    - Complexity measures
    - Performance measures
    - Process metrics
    - Product metrics
    - Software science
  + S.2.9 Management
    - Copyrights
    - Cost estimation
    - Life cycle
    - Productivity
    - Programming teams
    - Software configuration management
    - Software process models (e.g., CMM, ISO, PSP)
    - Software quality assurance (SQA)
    - Time estimation
  + S.2.10 Design
    - Methodologies
    - Representation
  + S.2.11 Software Architectures
    - Data abstraction
    - Domain-specific architectures
    - Information hiding
    - Languages (e.g., description, interconnection, definition)
    - Patterns (e.g., client/server, pipeline, blackboard)
  + S.2.12 Interoperability
    - Data mapping
    - Distributed objects
    - Interface definition languages
  + S.2.13 Reusable Software
    - Domain engineering
    - Reusable libraries
    - Reuse models
  + S.2.m Miscellaneous
    - Rapid prototyping
    - Reusable software
* S.3 PROGRAMMING LANGUAGES
  + S.3.0 General
    - Standards
  + S.3.1 Formal Definitions and Theory
    - Semantics
    - Syntax
  + S.3.2 Language Classifications
    - Applicative (functional) languages
    - Concurrent, distributed, and parallel languages
    - Constraint and logic languages
    - Data-flow languages
    - Design languages
    - Extensible languages
    - Macro and assembly languages
    - Microprogramming languages
    - Multiparadigm languages
    - Nondeterministic languages
    - Nonprocedural languages
    - Object-oriented languages
    - Specialized application languages
    - Very high-level languages
  + S.3.3 Language Constructs and Features
    - Abstract data types
    - Classes and objects
    - Concurrent programming structures
    - Constraints
    - Control structures
    - Coroutines
    - Data types and structures
    - Dynamic storage management
    - Frameworks
    - Inheritance
    - Input/output
    - Modules, packages
    - Patterns
    - Polymorphism
    - Procedures, functions, and subroutines
    - Recursion
  + S.3.4 Processors
    - Code generation
    - Compilers
    - Debuggers
    - Incremental compilers
    - Interpreters
    - Memory management (garbage collection)
    - Optimization
    - Parsing
    - Preprocessors
    - Retargetable compilers
    - Run-time environments
    - Translator writing systems and compiler generators
  + S.3.m Miscellaneous
* S.4 OPERATING SYSTEMS
  + S.4.0 General
  + S.4.1 Process Management
    - Concurrency
    - Deadlocks
    - Multiprocessing/multiprogramming/multitasking
    - Mutual exclusion
    - Scheduling
    - Synchronization
    - Threads
  + S.4.2 Storage Management
    - Allocation/deallocation strategies
    - Distributed memories
    - Garbage collection
    - Main memory
    - Secondary storage
    - Segmentation
    - Storage hierarchies
    - Swapping
    - Virtual memory
  + S.4.3 File Systems Management
    - Access methods
    - Directory structures
    - Distributed file systems
    - File organization
    - Maintenance
  + S.4.4 Communications Management
    - Buffering
    - Input/output
    - Message sending
    - Network communication
    - Terminal management
  + S.4.5 Reliability
    - Backup procedures
    - Checkpoint/restart
    - Fault-tolerance
    - Verification
  + S.4.6 Security and Protection
    - Access controls
    - Authentication
    - Cryptographic controls
    - Information flow controls
    - Invasive software (e.g., viruses, worms, Trojan horses)
    - Security kernels
    - Verification
  + S.4.7 Organization and Design
    - Batch processing systems
    - Distributed systems
    - Hierarchical design
    - Interactive systems
    - Real-time systems and embedded systems
  + S.4.8 Performance
    - Measurements
    - Modeling and prediction
    - Monitors
    - Operational analysis
    - Queueing theory
    - Simulation
    - Stochastic analysis
  + S.4.9 Systems Programs and Utilities
    - Command and control languages
    - Linkers
    - Loaders
    - Window managers