

**4. Software Development and Management:** The process of designing, writing, testing, debugging/troubleshooting, and maintaining the source code of computer programs and of managing and maintaining software in an organization.

**Critical Work Functions:**

Software Development

- Recognize steps in common Software Development Life Cycle (SDLC) models, e.g.:
  - Analyze user needs and software requirements, including security and accessibility functions
  - Design a secure software solution that fits within time and cost constraints
  - Implement (develop) software based on design
  - Test software, make necessary modifications and assure its quality, accessibility, and usability
  - Deploy and integrate software
  - Maintain and administer software
- Explain the importance of integrating security requirements into SDLC
- Understand and explain the importance of integrating user accessibility concepts and standards into the software development life cycle
- Understand common software architectures, including layered and distributed architecture models
- Describe modern tools for modeling software
- Recognize the existence of platform-specific developmental requirements, e.g., embedded systems, mobile computing, specialized devices, augmented reality, wearable computing
- Understand basic Web development functions and processes
- Recognize the importance of successful collaboration between software developers and designers, i.e., a developer’s ability to translate an “artistic” design into a functioning piece of software
- Describe the principles of user-centered design and universal design to increase usability
- Understand how software developers document their work
- Be able to interact effectively with a software development team

Programming

- Understand the importance of creating and understanding flowcharts, logic models, and other models that depict software logic and function
- Know what an algorithm is and how it works
- Understand relationships between software programs and the hardware and operating systems they run on
- Understand the difference between machine languages and higher level languages and how compilers translate between them
- Recognize common programming and scripting languages and what they are used for
- Understand what object-oriented programming is

- Describe secure coding practices and defensive programming techniques
- Understand and explain the relationships between databases and programming
- Describe common business processes for collecting information and feedback on software functionality
- Understand how software can include built-in accessibility features for people with disabilities and also can be designed in a way that is compatible with assistive technology devices
- Recognize the importance of maintenance and testing to the continued functioning of software
- Recognize major software security concerns (buffer overflow, X-site scripting, SQL Injection, etc.) and coding and management techniques to mitigate them
- Understand how programmers document their work
- Be able to interact effectively with programmers

#### Software Acquisition, Management and Maintenance

- Recognize and have familiarity with common operating systems, e.g., Windows, Mac, Linux, Unix, iOS, Android
- Recognize and have familiarity with accessibility features of all common operating systems
- Recognize and have familiarity with common enterprise software applications, e.g., Office Productivity Suites, Customer Relationship Management, Enterprise Resource Planning, Accounting/Finance, Database, Human Resource Management, Email, Online Collaboration
- Understand and be able to evaluate systems requirements for software
- Understand differences between enterprise hosted and outsourced software solutions, e.g., cloud
- Be able to evaluate business needs, software solutions and justify decisions for software solution acquisition, including build versus buy options
- Understand how 3<sup>rd</sup> party software is augmented with other solutions
- Be able to interact with software vendors and manage vendor relationships
- Understand common software purchasing, licensing, and maintenance agreements
- Understand common business processes for installing, managing and maintaining enterprise software
- Explain the process of software evolution
- Understand the purpose of, and differences between, updates, patches, and third-party modifications
- Recognize the impacts changes in software and interruptions to systems will have on end users and know how to minimize negative impacts, e.g., use of maintenance windows
- Be able to develop and maintain high quality software documentation (e.g., installation and update history, compatibility issues, license assignments) and communicate this information effectively to stakeholders
- Recognize the security risks and accessibility implications inherent in updating or

modifying software

**Technical Content Areas:**

Application Architecture

- Configuration and adaptation
- Deployment
- Design Patterns
- Risk management
- Scalability
- Standards
- Strategies

Development/Programming Fundamentals

- Algorithms (sorting, searching, automating and improving efficiency)
- Application Program Interface (API)
- Basic programming constructs (assignment, arithmetic expressions, loops, conditions, input/output, error handling)
- Data structures (list, vector, array, stack, queue, tree, graph, maps)
- Event-driven programming
- Object oriented programming
- Programming concurrent processes
- Secure coding standards
- Testing/Quality Assurance
- User interface/user experience (UIUX)

Development/Programming Technologies

- Database
- Integrative coding
- Inter-systems communications
- Machines languages and compilers
- Parallel systems development/programming
- Programming and scripting languages
- Software security practices

Software Development Life Cycle Models, such as

- Agile model
- Evolutionary model
- Incremental model

- Spiral model
- Waterfall model

#### Web Development

- Quality assurance
- Technical content
- Web site accessibility
- Web site design and usability
- Web site development/programming and maintenance
- Web site/Internet security