SWEBOK KA #9: Software Engineering Tools and Methods

The Software Engineering Body of Knowledge (SWEBOK) features 11 knowledge areas (KAs). The ninth KA is Software Engineering Tools and Methods. The Software Engineering Tools and Methods KA is focused on the tools and methods used throughout the complete life cycle processes. It includes two topics, as shown in Figure 1. These topics are Software Engineering Tools and Software Engineering Methods.

The Software Engineering Tools topic covers the tools used for software engineering. Software requirement tools deal with software requirements and are classified as modeling and traceability tools. Software design tools are used for creating and checking software designs. Software construction tools are used to produce source code and include program editors, compilers and code generators, interpreters, and debuggers. Software testing tools include test generators, test execution frameworks, test evaluation tools, test management tools, and performance analysis tools. Software maintenance tools are used for modification of existing software and include comprehension tools and reengineering tools. Software configuration management tools include defect tracking tools, version management tools, and release and build tools. Software engineering management tools are divided into project planning and tracking, risk management, and measurement. Software engineering process tools include modeling tools, management tools, and software development environments. Software quality tools are divided into inspection and analysis tools. Finally, some miscellaneous tool issues include tool integration techniques, meta-tools, and tool evaluation.

The Software Engineering Methods topic covers the methods used for software engineering. Heuristic methods deal with informal approaches. The four categories of heuristic methods are structured, data-oriented, object-oriented, and domain-specific. Formal methods deal with mathematically based approaches. The categories of formal methods are specification languages and notations, refinement, and verification/proving properties. Prototyping methods deal with approaches based on various forms of prototyping. Prototyping styles are various approaches such as throwaway, evolutionary, and executable specification. Prototyping targets include requirements, architectural design, and user interface. Prototyping evaluation techniques are ways in which the results of a prototype exercise are used.



Figure 1. Breakdown of Topics for the Software Engineering Tools and Methods Knowledge Area