

References

- Barr Z: Earned value analysis: a case study, *PM Network* 10:33, 1996.
- Baumert JH et al: *Software measures and the capability maturity model*, CMU/SEI 92-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1992.
- Boehm B: A spiral model of software development and enhancement, *IEEE Computer* 21(5), 1988.
- Brooks FP, Jr: No silver bullet: essence and accidents of software engineering, *IEEE Computer* 15(1):10, 1987.
- Carleton AD et al: *Software measurement for DoD systems: recommendations for initial core measures*, CMU/SEI-92-TR-19, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1992.
- Chen M, Han J, Yu P: Data mining: an overview from a database perspective, *IEEE Transactions on Knowledge and Data Engineering* 8(6), 1996.
- Daich, GT et al: Metrics tools: Size, *CrossTalk* April 1995.
- DBMS client/server computing*, An M & T Publication, January 1993.
- Druyun DA: Acquisition policy 93M-017, Software Metrics Policy, Office of the Deputy Assistant Secretary of the Air Force (acquisition), February 16, 1994, *CrossTalk* April 1994.
- Easterbrook et al: *Experiences using formal methods for requirements modeling*.
- Erickson DR et al: Metrics tools: effort and schedule, *CrossTalk* March 1995.
- Fellenstein SJ: Year 2000 predictions for ISO 9000, *QM*, July/August 1999, p. 12.
- Firth R et al: *A guide to the classification and assessment of software engineering tools*, CMU/SEI-87-TR-10.
- Fleming QW: *Cost/schedule control systems criteria: the management guide to C/CSCSC*, Chicago, 1992, Probus Publishing.
- Fleming QW, Koppelman JM: The earned value body of knowledge, *PM Network* 10:11, 1996b.
- Florac WA et al: *Software quality measurement: a framework for counting problems and defects*, CMU/SEI-92-TR-22, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, September 1992.
- Ganska R, Grotzky J, Rubinstein J, Van Buren J: *Requirements engineering and design technology report*, software technology support center, Hill Air Force Base, October 1995.
- Gartner Group, *ADA Research Note*, November 1998.
- Gause DC, Weinberg GM: *Exploring requirements quality before design*, New York, 1989, Dorset House Publishing.

References

- Giles AE et al: Metrics tools, *CrossTalk* February 1995.
- Government Computer News*, April 3, 2000, p. 6.
- Hammer T, Huffman L, Rosenberg L, Wilson W, Hyatt L: *Requirement metrics for risk identification*, Software Engineering Laboratory Workshop, GSFC, 1996.
- Hammer T: *Measuring requirement testing*, 18th International Conference on Software Engineering, 1997.
- Hammer T: *Automated requirements management beware how you use tools*, 19th International Conference on Software Engineering, 1998.
- Hammond E: Making order out of chaos, *Federal Computer Week*, November 15, 1999, p. 26.
- Hansen GW, Hansen JV: *Database management and design*, New York, 1992, Prentice Hall.
- Hatfield MA: The case for earned value, *PM Network* 10:25, 1996.
- Hayden D: Picking a wireless data provider, *Field Force Automation*, April 2000, p. 32.
- IEEE Std 729–1983: *Standard glossary of software engineering terminology*, February 1983.
- IEEE Std 830–1993: *Recommended practice for software requirements specifications*, December 1993.
- Kerzner H: *Project management: a systems approach to planning, scheduling, and controlling*, New York, 1995, Van Nostrand Reinhold.
- Kitchenham B, Pfleeger SL: Software quality: the elusive target, *IEEE Software* 13(1):12, 1996.
- Krasner N: *Homing in on wireless location*, *Communication*, June 1999, p. 19.
- Maintainability and Quality Assurance*, 1989.
- Marsan CD: SuperNet set to launch, *Network World*, August 16, 1999.
- Martin J: *Information Manifesto*.
- NASA–STD–2100–91: *NASA software documentation standard*, NASA Headquarters Software Engineering Program, July 29, 1991.
- NASA: Software assurance guidebook, NASA Goddard Space Flight Center Office of Safety, Reliability. In Ewald A, Roy M: Distributed objects, *Object Magazine* Jan 1997.
- North K: Database developer, *WEB–Techniques*, June 1999, p. 18.
- Object Management Group, *What is CORBA?* August 1997.
- Palmer JF: *Integrating the structured techniques with JAD: leveled systems development*. A working paper presented at 12th Structured Methods Conference, August 1987.

References

- Parnas DL: On the criteria to be used in decomposing systems into modules, *Communications of the ACM* 15, December 1972.
- Parnas DL, Clements PC, Weiss DM: The modular structure of complex systems, *IEEE Transactions on Software Engineering* SE-11(3), March 1985.
- Peterson JA: *Design for Windows managers*, Technical Report, MCT, 1986.
- Peterson S: Coming to terms with software reuse terminology: a model-based approach, SEI, *ACM Software Engineering Notes*, April 1991.
- Porter AA, Votta LG Jr, Basili VR: Comparing detection methods for software requirements Inspections: a replicated experiment, *IEEE Transactions on Software Engineering* 21(6):563, 1995.
- Radosevich L: Measuring up, *CIO Magazine*, September 15, 1999a, p. 52.
- Radosevich L: Tale of a turnaround, *CIO Magazine*, September 15, 1999b, p. 56.
- Roedler G: Applying PSM to Systems Engineering, *Insight: The Army's Software Metrics Newsletter*, Winter 1999.
- Ross DT: Douglas Ross talks about structured analysis, *IEEE Computer* 18(7):80, 1985.
- Royce WW: Managing the development of large software systems, Proceedings of the 9th International Conference, *Software Engineering, IEEE Computer Society* 328, 1987.
- Sathi A, Morton T, Roth S: Callisto: an intelligent project management system, *AI Magazine* 7(5):34, 1986.
- Scacchi W: Models of software evolution, life cycle and process, *SEI Curriculum Module SEI-CM-10-1.0*, October 1987.
- Shlaer S, Mellor SJ: *Object-oriented systems analysis, modeling the world in data*, Englewood Cliffs, NJ, 1988, Yourdon Press.
- Shlaer S, Mellor S: *Object-lifecycles: modeling the world in states*, New York, 1992, Prentice-Hall.
- Siegel J: *CORBA fundamental and programming*, Philadelphia, 1996, John Wiley & Sons.
- Singletary N: Whats the value of earned value? *PM Network* 10(28):30, 1996.
- Sodhi J, Sodhi P: *Software reuse*, New York, 1999, McGraw-Hill.
- Sodhi J, Sodhi P: *Object-oriented methods for software development*, New York, 1996, McGraw-Hill.
- Sodhi J: *Software requirements, analysis, and specification*, New York, 1992, McGraw-Hill.
- Sodhi J: *Software engineering methods, management, and CASE tools*, New York, 1991, McGraw-Hill.
- Sodhi J: *Computer systems techniques: development, implementation, and software maintenance*, PA, 1990, TAB Books.

References

- Sodhi J: *Managing ADA projects using software engineering*, PA, 1990, TAB Books.
- Sodhi J: *Evaluation of teaching SERA*, Seventh Annual National Conference for ADA Technology, 1989.
- Sodhi J, George KM: *Objects with multiple representations in ADA*, Seventh Annual National Conference for ADA Technology, 1989.
- Sodhi J: Overview of ADA features for real-time systems, *Defense Science*, November 1988.
- Software*, A Sentry Publication, January 1992.
- Software*, A Sentry Publication, May 1992.
- Software*, A Sentry Publication, September 1992.
- Software Development*, A Miller Publication, 1997
- Software Development*, A Miller Publication, January 1998
- Software Engineer's Reference Book 16*, 1991.
- Software Magazine*, Entering ORB-it, January 1997.
- Sommerville I: *Software engineering*, ed 4, Wokingham, England, 1992, Addison-Wesley.
- Sommerville I, Sawyer P: *Requirements engineering a good practice guide*, Philadelphia, 1997, John Wiley & Sons.
- Spurr K et al: *Software assistance for business re-engineering*, Philadelphia, 1994, John Wiley & Sons.
- Stikeleather J, Clarke J, Fingar P: *Distributed computing for business*, Technical Resource Connection.
- Stokes DA: Requirements analysis, *Computer Weekly Software Engineers Reference Book 16*, 1991.
- Stokes DA: Requirements analysis, *Computer Weekly*.
- The Sunday Oklahoman*, April 23, 2000.
- USA Today*, January 2000.
- Van Warren L, Beckman BC: *System for retrieving reusable software*, 1993, NASA Tech Briefs.
- Ward TP, Bracket WJ: *Object-oriented requirements definition and software design*, Boston, 1991, Boston University.
- Ward TP, Stephen JM: *Structured development for real-time systems*, vols I, II, III, Englewood Cliffs, NJ, 1985, Prentice-Hall.
- Warnier JD, Kenneth T: ORR, *structured systems development*, New York, 1977, Yourdon Press.

References

- Wasserman AI, Pircher PA, Muller RJ: *An object-oriented structured design method for code generation*, *SIGSOFT Software Engineering Notes* 14(1):32, 1989.
- Wilson W, Rosenberg L, Hyatt L: Automated analysis of requirement specifications, Fourteenth Annual Pacific Northwest Software Quality Conference, 1996.
- Wirfs-Brock R et al: *Designing object-oriented software*, Englewood Cliffs, NJ, 1990, Prentice Hall.
- Wirth N: Program development by stepwise refinement, *Communications of the ACM* 14(4):221, 1971.
- Wood W, Long G, Wood P, David P: *Classifying software design methods*, Technical Report, CMU/SEI-89-TR-25.

List of Figures

Chapter 1: Information Technology Project Initiation

Figure 1–1: Various phases of the IT project

Figure 1–2: Cost of hardware and software trends with time

Figure 1–3: Context diagram

Figure 1–4: Relationship among users, customers, and stakeholders

Figure 1–5: Major phases of an IT project

Figure 1–6: Generation air–ticket relationship among customers, users, stakeholders, and management

Chapter 2: Guidelines for Successful Project Planning

Figure 2–1: Modeling and simulation process

Figure 2–2: Sample WBS model

Figure 2–3: Sample master schedule

Figure 2–4: Budget model

Figure 2–5: Functional organizational model

Figure 2–6: Pure organizational model

Figure 2–7: Matrix organizational model

Figure 2–8: Hybrid organizational model

Figure 2–9: Staffing concept for a typical IT project

Figure 2–10: Generating ticket systems prototype model

Figure 2–11: WBS, manpower, and budget allocation

Chapter 3: Project Estimating Techniques and Tools

Figure 3–1 A: Symbols used in the Gantt chart;

Figure 3–1 B: Gantt chart showing project planning versus actual activity

Figure 3–2: A typical PERT network showing activities that must be performed and their sequence. Note that the dummy activity has no duration.

Figure 3–3: A, Activity; B, Event; C, Dummy; D, Restrictions

Figure 3–4: A, Another example of restrictions; B, Merge; C, Burst

Figure 3–5: A PERT network

Figure 3–6: Project activities

Figure 3–7: Project activities CPM

Figure 3–8: Project activities and Gantt chart

Chapter 4: Project Management Quality Control Mechanism

Figure 4–1: Capability maturity model levels

Figure 4–2: P–CMM model

Chapter 5: IT Project Risk Management

Figure 5–1: Safety risk constraints

Figure 5–2: Risk management process

Figure 5–3: Risk management plan activities

Figure 5–4 A: A well–managed IT project

Figure 5–4 B: A poorly managed IT project

Figure 5–5: Risk–monitoring process

Chapter 7: System Measurement Techniques

Figure 7–1: Quantitative project measurement process

Figure 7–2: Project measurement process

Figure 7–3: Sample cost, schedule, and project progress

Figure 7–4 A: Sample cost, schedule, and project progress

Figure 7–4 B: Sample funding and personnel resources

Figure 7–5 A: Bathtub curve (hardware defect);

Figure 7–5 B: Software defect

Figure 7–6: Tailoring process

Chapter 8: Commercial Items

Figure 8–1 A: Project performance efficiency

Figure 8–1 B: Tasks completed

Figure 8–1 C: Defects by phases

Chapter 9: Customer, Customer, and Customer

Figure 9–1: Knowing your customers parameters

Figure 9–2: Customer expectations

Figure 9–3: Customer acceptance scenario

Chapter 11: Internet Applications

Figure 11–1: WAP server

Chapter 12: Distributed Object Technology

Figure 12–1: A sample of distributed object environment

Figure 12–2: An overview of CORBA

Figure 12–3: DCOM client server application overview

Figure 12–4: Logical flow with distributed object technology

Figure 12–5: Before information access

Figure 12–6: After information access

Chapter 13: Distributed Objects

Figure 13–1: Window flow diagram

Figure 13–2: Client–server application

Figure 13–3: Various phases

Figure 13–4: Class interaction diagram

Figure 13–5: Customer contact window

Figure 13–6: Flight information

Chapter 14: Wireless Practical Case Study

Figure 14–1: How WAP works

Figure 14–2: Wireless tasks

Chapter 15: System Requirements

Figure 15–1: IT system requirements determination process

Figure 15–2: IT system requirement specification

Chapter 16: Managing System Requirements

Figure 16–1: Identification of defects in system requirements

Figure 16–2: Types of system requirement errors

Figure 16–3: Caliber–RM user interface screen

Figure 16–4: Caliber–RM requirement traceability screen

Figure 16–5: Caliber–RM project requirement baseline

Figure 16–6: Caliber–RM requirement grid screen

Chapter 17: System Design Process

Figure 17–1: IT system design process

Figure 17–2: A system architecture

Figure 17–3: A sample of a generic system architecture model

Figure 17–4: Financial accounting domain and subdomains

Chapter 18: Software Requirements

Figure 18–1: IT system design process

Figure 18–2: A sample of the incremental model

Chapter 19: Software Design Process

Figure 18–3: A sample of the waterfall model

Figure 18–4: A sample of the rapid prototyping model

Figure 18–5: A sample of the spiral life–cycle model

Figure 18–6: A sample of the domain prototype model

Figure 18–7: A sample context diagram

Figure 18–8: A sample data flow diagram. That data flow diagram consists of rectangles that are the source or destination of data outside the system. (Customarily, rectangles are not shown in the data flow diagrams because they are already present in the context diagram.) A data flow symbol (arrow) represents a path where data moves into, around, and out of the system. A process symbol (circle) represents a function of the system that logically transforms data. A data store is a symbol (open–ended rectangle) that shows a place in the system where data is stored.

DF, Data flow; P, process (data process); ST, store (data store); DS, data source; TM, terminator (information destination).

Figure 18–9: A context diagram

Figure 18–10: A sample of level 1 partitioning

Figure 18–11: A sample of level 2 partitioning

Chapter 19: Software Design Process

Figure 19–1: IT software design process

Figure 19–2: An OAM diagram

Figure 19–3: An OIM diagram

Figure 19–4: OIM table

Figure 19–5: OIM table and object relationship

Figure 19–6: OIM notations

Figure 19–7: Supertype and/or subtype constructs

Figure 19–8: Correlation symbol

Figure 19–9: Object lifetime diagram

Figure 19–10: OBM sample

Figure 19–11: Software–structured design

Chapter 20: Software Implementation Process

Figure 20–1: IT software implementation process

Figure 20–2: Integration of software systems with reuse assets

Figure 20–3: A sample solutions

Figure 20–4: Functional model (context diagram)

Figure 20–5: Functional model (data flow diagram)

Figure 20–6: STD sample solution

Chapter 21: System Integration and Evaluation

Figure 21–1: IT system integration process

Figure 21–2: A well–managed IT project

Figure 21–3: A poorly managed IT project

Chapter 22: Practical Case Study

Figure 22–1: Organizational chart

Figure 22–2: IT project schedule

Figure 22–3: IT project costs

List of Tables

Chapter 1: Information Technology Project Initiation

Table 1–1: Allocation Of It Project Resources

Chapter 2: Guidelines for Successful Project Planning

Table 2–1: Criteria For Selecting An Organization Model

Chapter 3: Project Estimating Techniques and Tools

Table 3–1: Baseline Plan Work Units

Table 3–2: Schedule Variance Work Units

Table 3–3: Cost Variance Work Units

Table 3–4: Spend Comparison Approach Work Units

Table 3–5: Case Study Date

Chapter 5: IT Project Risk Management

Table 5–1: Risk Analysis Criteria

Chapter 9: Customer, Customer, and Customer

Table 9–1: Customer Confidence Level

Chapter 13: Distributed Objects

Use Case Design Information

List of Boxes

Chapter 1: Information Technology Project Initiation

Box 1–1: List of Support Personnel

Chapter 6: Commercial Best Practices Standards

Box 6–1: Characteristics of a Good System Development Standard

Box 6–2: List of Data Item Descriptions

Chapter 9: Customer, Customer, and Customer

Box 9–1: Features to Help the Manager Understand the Customer

Box 9–2: Customer Expectation Factors

Chapter 15: System Requirements

Box 15–1: SysRS Outline