

---

# Glossary

<b>ADS</b>	Appraisal Disclosure Statement
<b>AMN</b>	Abstract Machine Notation
<b>AQL</b>	Acceptable Quality Level
<b>ATM</b>	Automated Teller Machine
<b>BRS</b>	Business Requirements Specification
<b>BSC</b>	Balanced Score Card
<b>CAR</b>	Causal Analysis and Resolution
<b>CCB</b>	Change Control Board
<b>CCS</b>	Calculus Communicating Systems
<b>CICS</b>	Customer Information Control System
<b>CM</b>	Configuration Management
<b>CMM®</b>	Capability Maturity Model
<b>CMMI®</b>	Capability Maturity Model Integration
<b>COCOMO</b>	Constructive Cost Model
<b>COPQ</b>	Cost of Poor Quality
<b>COTS</b>	Customized Off the Shelf
<b>CSP</b>	Communicating Sequential Processes
<b>DAR</b>	Decision Analysis and Resolution
<b>DMAIC</b>	Define, Measure, Analyse, Improve, Control
<b>DMADV</b>	Define, Measure, Analyse, Design, Verify
<b>DOORS</b>	Dynamic Object-Oriented Requirements System
<b>DSDM</b>	Dynamic Systems Development Method
<b>EAF</b>	Effort Adjustment Factor
<b>ESA</b>	European Space Agency
<b>ESI</b>	European Software Institute
<b>FAR</b>	Functional Area Representatives
<b>FI</b>	Fully Implemented
<b>FMEA</b>	Failure Mode and Effects Analysis
<b>FSM</b>	Finite State Machine
<b>GG</b>	Generic Goal
<b>GP</b>	Generic Practice
<b>GQM</b>	Goal, Question, Metric
<b>GUI</b>	Graphical User Interface
<b>HP</b>	Hewlett Packard

---

<b>HR</b>	Human Resources
<b>HTML</b>	Hyper Text Mark-up Language
<b>IBM</b>	International Business Machines
<b>IDE</b>	Integrated Development Environment
<b>IDEAL</b>	Initiating, Diagnosing, Establishing, Acting and Learning
<b>IEC</b>	International Electro technical Commission
<b>IEEE</b>	Institute of Electrical and Electronic Engineers
<b>IPM</b>	Integrated Project Management
<b>ISEB</b>	Information System Examination Board
<b>ISO</b>	International Standards Organization
<b>JAD</b>	Joint Application Development
<b>KLOC</b>	Thousand Lines of Code
<b>LCL</b>	Lower Control Limit
<b>LDRA</b>	Liverpool Data Research Associates
<b>LI</b>	Largely Implemented
<b>LOC</b>	Lines of Code
<b>MA</b>	Measurement and Analysis
<b>MOD</b>	Ministry of Defence
<b>MTTF</b>	Mean Time to Failure
<b>MTTR</b>	Mean Time to Repair
<b>NATO</b>	North Atlantic Treaty Organization
<b>NI</b>	Not Implemented
<b>ODC</b>	Orthogonal Defect Classification
<b>OID</b>	Organization Innovation and Deployment
<b>OMT</b>	Object Modelling Technique
<b>OOSE</b>	Object-Oriented Software Engineering
<b>OPD</b>	Organization Process Definition
<b>OPF</b>	Organization Process Focus
<b>OPP</b>	Organization Process Performance
<b>OSSP</b>	Organization Set of Standard Processes
<b>OT</b>	Organization Training
<b>PB</b>	Project Board
<b>PCE</b>	Phase Containment Effectiveness
<b>PCMM</b>	People Capability Maturity Model
<b>PDCA</b>	Plan, Do, Check, Act
<b>PI</b>	Product Integration
<b>PI</b>	Partially Implemented
<b>PIID</b>	Practice Implementation Indicator Description
<b>PL/1</b>	Programming Language 1
<b>PMBOK</b>	Project Management Book of Knowledge
<b>PMI</b>	Project Management Institute
<b>PMC</b>	Project Monitoring and Control
<b>PMP</b>	Project Management Professional
<b>PP</b>	Project Planning

---

<b>PPM</b>	Project Portfolio Management
<b>PPQA</b>	Process and Product Quality Assurance
<b>Prince</b>	Projects In a Controlled Environment
<b>PSP</b>	Personal Software Process
<b>PVCS</b>	Polytron Version Control System
<b>QA</b>	Quality Assurance
<b>QCC</b>	Quality Control Circle
<b>QMS</b>	Quality Management System
<b>QPM</b>	Quantitative Project Management
<b>RAD</b>	Rapid Application Development
<b>RAG</b>	Red, Amber, Green
<b>RCA</b>	Root Cause Analysis
<b>RD</b>	Requirements Development
<b>RFP</b>	Request for Proposal
<b>RM</b>	Requirements Management
<b>ROI</b>	Return on Investment
<b>RPM</b>	Rational Portfolio Manager
<b>RSM</b>	Rational Software Modeller
<b>RSKM</b>	Risk Management
<b>RUP</b>	Rational Unified Process
<b>SAM</b>	Supplier Agreement Management
<b>SCAMPI</b>	Standard CMMI Appraisal Method for Process Improvement
<b>SEI</b>	Software Engineering Institute
<b>SEPG</b>	Software Engineering Process Group
<b>SG</b>	Specific Goal
<b>SLA</b>	Service Level Agreement
<b>SLOC</b>	Source lines of code
<b>SOW</b>	Statement of Work
<b>SP</b>	Specific Practice
<b>SPC</b>	Statistical Process Control
<b>SPI</b>	Software Process Improvement
<b>SPICE</b>	Software Process Improvement Capability determination
<b>SQA</b>	Software Quality Assurance
<b>SSADM</b>	Structured Systems Analysis and Design Method
<b>TQM</b>	Total Quality Management
<b>TS</b>	Technical Solution
<b>TSP</b>	Team Software Process
<b>UAT</b>	User Acceptance Testing
<b>UCL</b>	Upper Control Limit
<b>UK</b>	United Kingdom
<b>UML</b>	Unified Modelling Language
<b>VAL</b>	Validation
<b>VDM</b>	Vienna Development Method
<b>VDM*</b>	Irish School of VDM

<b>VER</b>	Verification
<b>VOB</b>	Version Object Base
<b>VSS</b>	Visual Source Safe
<b>Y2K</b>	Year 2000

---

## References

1. Appraisal Requirements for CMMI V1.2. (ARC V1.2) (2006) SCAMPI Upgrade Team. TR CMU/SEI-2006-TR-011. August, 2006
2. Basili V, Rombach H (1988) The TAME project. Towards improvement-oriented software environments. *IEEE Trans Softw Eng* 14(6):758–773
3. Beck K (2000) *Extreme Programming explained. Embrace change.* Addison-Wesley, Reading
4. Bhandari I (1993) A case study of software process improvement during development. *IEEE Trans Softw Eng* 19(12):1157–1170
5. Bjorner D, Cliff J (1978) *The Vienna development method. The meta language, vol 61, Lecture notes in computer science.* Springer, New York
6. Bjorner D, Jones C (1982) *Formal specification and software development,* Prentice Hall International series in computer science. Prentice Hall International, Englewood Cliffs
7. Boehm B (1981) *Software engineering economics.* Prentice Hall, Englewood Cliffs
8. Boehm B (1988) A spiral model for software development and enhancement. *Computer* 21:61–72
9. Brassard M, Ritter D (1994) *The Memory Jogger. A pocket guide of tools for continuous improvement and effective planning.* Goal/QPC, Methuen
10. Brooks F (1975) *The mythical man month.* Addison Wesley, Reading
11. Brooks F (1986) No silver bullet. Essence and accidents of software engineering. In: *Information processing.* Elsevier, Amsterdam
12. Buxton JN, Naur P, Randell B (1975) *Software Engineering.* Petrocelli. Report on two NATO conferences held in Garmisch, Germany (October 1968) and Rome, Italy (October 1969)
13. Chrissis MB, Conrad M, Shrum S (2011) *CMMI for development. Guidelines for process integration and product improvement, 3rd edn, SEI series in software engineering.* Addison Wesley, Boston
14. Crosby P (1979) *Quality is free. The art of making quality certain.* McGraw Hill, New York
15. Deming WE (1986) *Out of crisis.* MIT Press, Cambridge, MA
16. Dijkstra EW (1972) *Structured programming.* Academic, London
17. Dijkstra EW (1976) *A disciple of programming.* Prentice Hall, Englewood Cliffs
18. Diller A (1990) *Z. An introduction to formal methods.* Wiley, England
19. Dunaway DK, Masters S (1996) *CMM Based Appraisal for Internal Process Improvement (CBA IPI): method description.* Technical report CMU/SEI-96-TR-007. Software Engineering Institute, Pittsburgh
20. Fagan M (1976) Design and code inspections to reduce errors in software development. *IBM Syst J* 15(3):182–210
21. Fenton N (1995) *Software metrics: A rigorous approach.* Thompson Computer Press, Boston
22. Gerhart S, Craighen D, Ralston T (1994) Experience with formal methods in critical systems. *IEEE Software*, January 1994
23. Gilb T (1976) *Software metrics.* Winthrop Publishers, Inc., Cambridge
24. Gilb T, Graham D (1994) *Software inspections.* Addison-Wesley, Reading
25. Gries D (1981) *The science of programming.* Springer, Berlin

26. Hinchey M, Bowen J (eds) (1995) Applications of formal methods, Prentice Hall International series in computer science. Prentice Hall, New York
27. Hoare CAR (1985) Communicating sequential processes, Prentice Hall International series in computer science. Prentice Hall International, Englewood Cliffs
28. Hoare JP (1995) Application of the B-method to CICS. In: Hinchey M, Bowen JP (eds) Applications of formal methods, Prentice Hall International series in computer science. Prentice Hall, New York
29. Humphry W (1989) Managing the software process. Addison-Wesley, Reading
30. Ince D, Andrews D (1991) Practical formal methods with VDM, McGraw Hill International series in software engineering. McGraw Hill, New York
31. ISO/IEC (1991) ISO/IEC 9126: Information Technology. Software Product Evaluation: quality characteristics and guidelines for their use. ISO/IEC, Geneva
32. Jacobson I, Booch G, Rumbaugh J (1999) The unified software modelling language user guide. Addison-Wesley, Reading
33. Juran J (1951) Juran's quality handbook. McGraw Hill, New York
34. Kaplan RS, Norton DP (1996) The balanced scorecard. Translating strategy into action. Harvard Business School Press, Boston
35. Keeni G et al (2000) The evolution of quality processes at Tate Consultancy Services. IEEE Software 17(4):79–88
36. Kuhn T (1970) The structure of scientific revolutions. University of Chicago Press, Chicago
37. Lakatos I (1976) Proof and refutations. The logic of mathematical discovery. Cambridge University Press, Cambridge
38. Lions JL (1996) Ariane 5. Flight 501. Failure report by enquiry board, The MathWorks, Inc., Natick
39. Mac An Airchinnigh M (1990) Conceptual models and computing. PhD thesis, Department of Computer Science, University of Dublin. Trinity College, Dublin
40. Manley E (1995) Taurus: how I lived to tell the tale. Am Program Softw Fail 8:17
41. Manns T, Coleman M (1996) Software quality assurance. Macmillan Press Ltd., Hampshire
42. McDonnell E (1994) The development of a multiplex bundling system using formal methods, vols I & II. MSc thesis. Department of Computer Science, Trinity College, Dublin
43. Milner R et al (1989) A calculus of mobile processes (Part 1). LFCS report series. ECS-LFCS-89-85. Department of Computer Science. University of Edinburgh, Edinburgh
44. Ministry of Defence (1991) 00-55 (PART 1)/Issue 1, The procurement of safety critical software in defence equipment, PART 1: Requirements. Ministry of Defence, Interim Defence Standard, UK
45. Ministry of Defence (1991) 00-55 (PART 2)/Issue 1, The procurement of safety critical software in defence equipment, PART 2: Guidance. Ministry of Defence, Interim Defence Standard, UK
46. O'Hara F (1998) Peer reviews – the key to cost effective quality. European SEPG, Amsterdam
47. O'Regan G (2002) A practical approach to software quality. Springer, New York
48. O'Regan G (2006) Mathematical approaches to software quality. Springer, London
49. O'Regan G (2010) Introduction to software process improvement. Springer, London
50. O'Regan G (2013) Mathematics in computing. Springer, London
51. Office of Government Commerce (2004) Managing successful projects with PRINCE2. TSO, London
52. Parnas D (1972) On the criteria to be used in decomposing systems into modules. Commun ACM 15(12):1053–1058
53. Peterson B (1995) The IDEAL model. Software Engineering Institute. Software process improvement and practice (Pilot Issue), Wiley, New York, August. 1995
54. Polya G (1957) How to solve it. A new aspect of mathematical method. Princeton University Press, Princeton
55. Rational for the development of the U.K. defence standards for safety critical software. COMPASS conference, June 1990. Ministry of Defence

56. Royce W (1970) The software lifecycle model (Waterfall Model). In: Proceedings of the WESTCON, Los Angeles, August, 1970
57. Rumbaugh J et al (1999) The unified software development process. Addison-Wesley, Reading
58. Shewhart W (1931) The economic control of manufactured products. Van Nostrand, New York
59. Software Engineering Institute (2006) CMMI executive overview. Presentation by the SEI. Software Engineering Institute
60. Software Engineering Institute (2009) CMMI impact. Presentation by Anita Carleton. Software Engineering Institute. August 2009
61. Sommerville I (2011) Software engineering, 9th edn. Pearson, Boston
62. Spivey JM (1992) The Z notation. A reference manual, Prentice Hall International series in computer science. Prentice Hall, Englewood Cliffs
63. Standard CMMI Appraisal Method for Process Improvement (2006) CMU/SEI-2006-HB-002. V1.2. August 2006
64. Standish Group (1999) Estimating: art or science. Featuring Morotz Cost Expert. Standish Group Research Note
65. Tierney M (1991) The evolution of Def Stan 00-55 and 00-56: an intensification of the "Formal Methods debate" in the UK. Research Centre for Social Sciences, University of Edinburgh, Edinburgh
66. Wichmann BA (2000) A personal view of formal methods. National Physical Laboratory, March 2000

---

# Index

## A

Agile development, 46–47  
Analogy method, 63  
Appraisals, 29  
  plan, 257–258  
Ariane 5, 4  
Ariane-5 disaster, 41  
Audit escalation, 149  
Audit meeting, 147–148  
Audit planning, 146  
Audit reporting, 148  
Axiomatic approach, 296

## B

Balanced scorecard (BSC), 154–156  
Baldrige, Malcom, 53  
Barriers to success, 207–208  
Baseline, 90  
Basili, Victor, 26, 152  
Booch method, 327  
Breakthrough and control, 12  
Brooks, Fred, 5  
BSC. *See* Balanced scorecard (BSC)  
Business case, 61

## C

Calculus communicating systems (CCS), 303  
Capability maturity model integration (CMMI), 35–57, 101–134, 211–232  
  appraisals, 243–244  
  categories of, 222  
  kick-off session, 245–246  
  maturity levels, 217–220  
  process areas, 222–223  
  project manager, 236, 242  
  project plan, 244–245  
  project schedule, 245  
  representations, 220–222

CCB. *See* Change control board (CCB)  
CCS. *See* Calculus communicating systems (CCS)  
Change control, 95–97  
Change control board (CCB), 71, 82  
Change request, 71  
CICS. *See* Customer Information Control System (CICS)  
Clarity PPM, 274  
Class diagrams, 331  
Clearcase, 92  
Clearquest, 92  
CMMI. *See* Capability maturity model integration (CMMI)  
COCOMO. *See* Constructive Cost Model (COCOMO)  
Communicating sequential processes (CSP), 303  
Commuting diagram property, 323  
Conducting the appraisal, 261–267  
Configuration control, 91  
Configuration identification, 91  
Configuration management, 89  
  audits, 98  
  plan, 95  
Constructive Cost Model (COCOMO), 272  
Continuous improvement cycle, 236  
Continuous representation, 220  
Cost of poor quality (COPQ), 24, 103  
Cost predictor models, 63  
Crosby, Philip, 5, 15  
CSP. *See* Communicating sequential processes (CSP)  
Customer care metrics, 168–170  
Customer Information Control System (CICS), 293  
Customer satisfaction, 26–27  
  metrics, 157–158



**D**

Darlington Nuclear power plant, 293  
 Data reification, 323  
 Dedicated improvement sub-teams, 241–242  
 Def Stan 00-55, 292  
 Deliver appraisal results, 267  
 Deming, W. Edwards, 10  
 Development quality metrics, 164–167  
 Dynamic Object-Oriented Requirements System (DOORS), 276

**E**

E-commerce testing, 131–133  
 Enterprise Architect, 280  
 ESA. *See* European Space Agency (ESA)  
 Escrow agreement, 139  
 Estimation, 62  
 European Space Agency (ESA), 41  
 Expert judgment, 63

**F**

Fagan inspections, 38, 53, 108–116  
   guidelines, 109  
 Fagan, Michael, 20  
 Fishbone diagrams, 23, 175–177  
 Formal methods, 55–56  
 Formal specification, 289  
 Functional requirement, 78  
 Function points, 63

**G**

Generate appraisal results, 266–267  
 Generic goals, 226  
 Generic practices, 226  
 Gilb, Tom, 26  
 Goal Question Metric (GQM), 26, 152

**H**

Histograms, 23, 177–178  
 Human Resources and Training Metrics, 160–162  
 Humphrey, Watts, 17, 212

**I**

IDE. *See* Integrated development environment (IDE)  
 IEEE standards, 43  
 Information hiding, 84, 305

Inspection meeting, 112  
 Integrated development environment (IDE), 280  
 Irish School of VDM (VDM\*), 300  
 ISO 9000, 185  
 ISO 9001, 187, 205  
   implementation, 195  
 ISO 9004, 187  
 ISO 9126, 5  
 ISO 9000 Certification Process, 195–197

**J**

Juran, Joseph, 12

**L**

LDRA Tools, 117, 283

**M**

Maintenance, 52  
 Mathematical proof, 297, 324  
 Measurement, 151  
 Microsoft Project, 273  
 Model, 43  
 Model-oriented approach, 294  
 Mongolian Hordes Approach, 35

**N**

Non-functional requirements, 78

**O**

Object diagram, 331  
 Object Modelling Technique (OMT), 327  
 Object-Oriented Software Engineering (OOSE), 327  
 OMT. *See* Object Modelling Technique (OMT)  
 OOSE. *See* Object-Oriented Software Engineering (OOSE)

**P**

Pareto chart, 23, 178–179  
 Parnas, 39, 49, 86, 305  
 Partial correctness, 303  
 Partial function, 316  
 PCE. *See* Phase containment effectiveness (PCE)  
 Performance testing, 51  
 Personal Software Process (PSP), 205

- Phase containment effectiveness (PCE), 115  
 PIIDs. *See* Practice Implementation Indicator Descriptions (PIIDs)  
 Piloting the process, 249  
 Planview, 274  
 Planview Enterprise, 274  
 PMBOK. *See* Project management body of knowledge (PMBOK)  
 Polytron Version Control System (PVCS), 92, 282  
 Postcondition, 301  
 Practice Implementation Indicator Descriptions (PIIDs), 256  
 Precondition, 301, 302  
 Predicate transformer, 303  
 Prepare participants, 261–262  
 Prince 2, 38, 53, 61, 74  
 Problem-solving techniques, 174–182  
 Process calculi, 303  
 Process improvement metrics, 158–160  
 Process mapping, 206, 246–248  
 Process maturity models, 54–55  
 Process model, 204  
 Professional Engineering Association, 36  
 Professional engineers, 40  
 Project, 59  
 Project board, 61, 72  
 Project closure, 73–74  
 Project management, 53–54, 59 metrics, 162–164  
 Project management body of knowledge (PMBOK), 61  
 Project manager, 61  
 Project monitoring and control, 70–71  
 Project reporting, 73  
 Prototyping, 48, 78  
 PSP. *See* Personal Software Process (PSP)  
 PVCS. *See* Polytron Version Control System (PVCS)
- Q**  
 Quality audit metrics, 167–168  
 Quality center, 28, 130  
 Quality management (system), 69–70, 187–189
- R**  
 Rational software modeler, 278  
 Rational unified process, 43, 45, 336  
 Refinement, 290  
 Reporting the results, 267–268  
 Request for proposal (RFP), 137  
 Requirements, 78  
   analysis, 81  
   elicitation, 79–81  
   engineering, 78–79  
   validation, 81, 290  
   verification, 82  
   traceability, 82  
 RequisitePro, 277  
 RFP. *See* Request for proposal (RFP)  
 Risk management, 67–70  
 Rolling out process, 249
- S**  
 SCAMPI  
   appraisals, 229–230, 253–269  
   methodology, 229  
 Scatter graphs, 180–181  
 Schema, 311  
   calculus, 301, 311  
   composition, 320–322  
   inclusion, 320  
 Scientific revolutions, 294  
 SEI. *See* Software Engineering Institute (SEI)  
 SEPG team, 236, 237, 240–242, 248, 249, 340  
 Sequence diagram, 333, 334  
 Shewhart model, 8  
 Shewhart, Walter, 8, 202  
 Six Sigma (6 $\sigma$ ), 53, 206  
 Software crisis, 36, 56  
 Software design, 77  
 Software engineering, 36, 38, 40  
 Software Engineering Institute (SEI), 203  
 Software Engineering Tools, 271–287  
 Software failures, 41  
 Software inspections, 101  
 Software metrics, 151–183  
 Software process, 200–202  
   improvement, 25, 199–209  
 Software quality assurance, 143–150  
 Software reuse, 50  
 Software testing, 50  
 SOW. *See* Statement of work (SOW)  
 SPC. *See* Statistical process control (SPC)  
 Specific goals, 223  
 Specific practices, 225  
 Spiral model, 44  
 Sprint planning, 47  
 Standish group, 2, 36, 56  
 State diagrams, 335  
 Statement of work (SOW), 139  
 Statistical process control (SPC), 8

Steering group, 236, 238, 240–242,  
246, 250, 340  
Story, 46  
Structured walkthrough, 104  
Supplier selection, 135  
System testing, 51

**T**

Tacoma Narrows bridge, 4  
Taurus project, 3  
Team Software Process (TSP), 206  
Test cases, 123  
    design, 126–127  
Test director, 284  
Test driven development, 51  
Test environment, 121  
Testing, 119  
Test planning, 121, 125–126  
Test process, 121–125  
Test reporting, 127–128  
Test tools, 124, 130–131  
Total quality management (TQM), 6  
Traceability, 49, 147, 277  
Trend graph, 23  
TSP. *See* Team Software Process (TSP)

**U**

UAT testing, 51  
UML. *See* Unified modelling language (UML)  
UML activity diagram, 334

UML diagrams, 330–335  
Unified modelling language (UML),  
327, 337  
Unit testing, 50  
Use-case diagram, 333  
User requirements, 77

**V**

Vienna Development Method (VDM),  
291, 298  
VIPER, 297  
Visual Source Safe (VSS), 92, 282

**W**

Waterfall model, 43  
Weakest precondition, 303  
Work breakdown structure, 63

**Y**

Y2K, 37, 41  
Y2K bug, 4, 41

**Z**

Z, 291  
Zermelo set theory, 302  
Z specifications, 301, 312  
    language, 301